Sequential Bayesian Inference for Dynamic Linear Models of Sensor Data

Yingying Lai

Thesis submitted for the degree of
Doctor of Philosophy

School of Mathematics, Statistics & Physics
Newcastle University
Newcastle upon Tyne
United Kingdom

May 2019

Abstract

We develop a spatio-temporal model to analyse pairs of observations on temperature and humidity. The data consist of six months of observations at five locations collected from a sensor network deployed in North East England. The model for the temporal component takes the form of two coupled dynamic linear models (DLMs), specified marginally for temperature and conditionally for humidity given temperature. To account for dependence at nearby locations, the governing system equations include spatial effects, specified using a Gaussian process. To understand the stochastic nature of the data, we perform fully Bayesian estimation for the model parameters and check the model fit via posterior distributions. The intractability of the posterior distribution necessitates the use of computationally intensive methods such as Markov chain Monte Carlo (MCMC). The main disadvantage of MCMC is computational inefficiency when dealing with large datasets. Therefore, we exploit a class of sequential Monte Carlo (SMC) algorithms known as particle filters, which sequentially approximate the posterior through a series of reweighting and resampling steps. The tractability of the observed data likelihood under the DLM admits the implementation of an iterated batch importance sampling (IBIS) scheme, which additionally uses a resample-move step to circumvent the particle degeneracy problem. To alleviate the computational burden brought from the resample-move step of IBIS, we develop a novel online version of IBIS by modifying the resample-move step through approximating the posterior over an observation window whose pre-specified length trades off accuracy and computational cost. Furthermore, performing the resampling step independently for batches of parameter samples allows a parallel implementation of the algorithm to be performed on a powerful multi-core high performance computing system. A comparison of observed measurements with their one-step and two-step forecast distributions shows that the model provides a good description of the underlying process and provides reasonable forecast accuracy.

Acknowledgements

I would like to express my sincere gratitude to my supervisors Prof Richard Boys and Dr Andrew Golightly for their guidance and encouragement. I would like to thank Prof Phil Taylor and the Newcastle Urban Observatory team for giving advice on data exploration. I would like to thank the School of Mathematics, Statistics and Physics for providing a pleasant environment, and the Faculty of Science, Agriculture and Engineering for funding this research. Finally, I would like to pass on my heartfelt appreciation to my families and friends for their selfless love and support.

Declaration

Parts of this thesis have been published by the author:

- Lai, Y., Golightly, A. and Boys, R. J. (2018). Sequential Bayesian inference for spatio-temporal models of temperature and humidity data. *Submitted*.
  (Available at https://arxiv.org/abs/1806.05424)

Contents

Contents i

List of Figures v

List of Tables ix

1 Introduction 1

1.1 Thesis aims 3

1.2 Outline of thesis 4

2 Dynamic linear models 7

2.1 Introduction 7

2.2 Seasonality 10

2.2.1 Sinusoidal form DLM 10

2.2.2 Fourier form DLM 11

2.3 Bayesian inference 13

2.3.1 Gibbs sampler 14

2.3.2 Marginal Metropolis-Hastings (MH) algorithm 19

2.3.3 MCMC diagnostics 22

2.4 Posterior predictive checks 24

2.4.1 Within-sample predictions 25

2.4.2 Out-of-sample forecasts 25

2.5 Simulation studies 26

2.5.1 Local level model 27

2.5.2 Sinusoidal form DLM 32

3 Sequential Monte Carlo **39**

3.1 Importance (re)sampling 39

3.1.1 Importance sampling 39

3.1.2 Importance resampling 42

3.2 State filtering 43

3.2.1 Sequential importance sampling (SIS) 43

3.2.2 Bootstrap particle filter (BPF) 45

3.2.3 Auxiliary particle filter (APF) 47

3.3 State and parameter filtering 49

3.3.1 Liu-West algorithm 49

3.3.2 Storvik algorithm 51

3.3.3 Particle learning (PL) 54

3.3.4 Iterated batch importance sampling (IBIS) 54

3.3.5 Adaptive iterated batch importance sampling (aIBIS) 57

3.3.6 Online IBIS 58

4 Simulation studies **61**

4.1 Comparison between fully adapted auxiliary particle filter and bootstrap particle filter 61

4.2 Comparison between the Liu-West algorithm, the Storvik algorithm, particle learning, IBIS and aIBIS 62

4.2.1 Local level model 62

4.2.2 Sinusoidal form DLM 67

5 Application to temperature and humidity data **75**

5.1 Data collection 75

5.2 Spatial DLMs 76

5.2.1 Additional harmonics 79

5.2.2 Spatial humidity DLM 80

5.3 Inference 81

5.4 Within-sample predictions and out-of-sample forecasts 86

5.5 Model selection 87

5.6 Parallel computing 89

5.6.1 System frameworks for parallelisation 89

5.6.2 Parallelisation for resampling 91

6 Results **97**

6.1 Simulation study 97

6.1.1 Comparison of full IBIS with serial resampling and parallelised local resampling 98

6.1.2 Comparison of full IBIS and online IBIS 98

6.2 Real data study 98

6.2.1 Posterior output 101

6.2.2 Predictive checks 102

7 Conclusions **109**

Bibliography **113**

List of Figures

2.1 A variety of time series data. 8

2.2 Diagram of the state space model. 9

2.3 Trace plots of MCMC samples by using different uniform proposal distributions. 23

2.4 Left: simulated data; right: 10 marginal posterior realisations of $\boldsymbol{\theta}_{1:n}$. 28

2.5 Gibbs sampler diagnostics: (1). trace plot by taking burn-in=100 and thinning=20; (2). autocorrelation function for the thinned chain after the burn-in period; (3). the posterior distribution (black) and the prior distribution (grey). The true parameter values are indicated by the solid circles. 29

2.6 Mean and 95% credible interval of within-sample predictions against the data. . 30

2.7 Simulated data (-o-) with mean and 95% credible interval of the samples for 1-step, 2-step, 3-step and 4-step ahead forecast respectively (error bars). 30

2.8 MH diagnostics: (1). trace plot by taking burn-in=100 and thining=20; (2). autocorrelation function for the thinned chain after the burn-in period; (3). the posterior distribution (black) and the prior distribution (grey). The true parameter values are indicated by the solid circles. 31

2.9 Comparison of the posterior distributions with the posterior means for V and W through the Gibbs sampler (red) and the MH algorithm (blue). The true parameter values are indicated by the solid circles. 32

2.10 Left: simulated data; right: 10 marginal posterior realisations of $\boldsymbol{F}_{1:n}\boldsymbol{\theta}_{1:n}$. . . . 33

2.11 MCMC diagnostics (the Gibbs sampler): (1). trace plot by taking burn-in=100 and thining=20; (2). autocorrelation function for the thinned chain after the burn-in period; (3). the posterior distribution (black) and the prior distribution (grey). The true parameter values are indicated by the solid circles. 34

2.12 Mean and 95% credible interval of the differences between within-sample predictions and the data. 35

2.13 Simulated data (-o-) with mean and 95% credible interval of the samples for 1-step, 2-step, 3-step and 4-step ahead forecast respectively (error bars). 36

2.14 MCMC diagnostics (MH algorithm): 1. trace plot by thining=20; 2. autocorrelation function; 3. the posterior distribution (black) with the prior distribution (grey). The true parameter values are indicated by the solid circles. 37

2.15 Comparison of the posterior distributions with the posterior means for V and W through the Gibbs sampler (red) and the MH algorithm (blue). The true parameter values are indicated by the solid circles. 38

3.1 Effective sample size. 45

4.1 Left plot: comparison of the posterior means of the state through FA-APF, BPF and MCMC over time. The true values of the state are indicated by the grey circles. Right plot: Comparison of the difference of posterior means with 95% credible intervals through FA-APF against MCMC and BPF against MCMC. . 63

4.2 Comparison of the posterior distribution of θ via BPF (blue) and FA-APF (red) and the MCMC output (histograms) at $t = 1,\ldots,6$ respectively. 64

4.3 Sequential posterior means with 95% credible intervals of W and V over time, calculated from the output of different SMC schemes. Top row: 3×10^3 particles; middle row: 5×10^3 particles; bottom row: 10^4 particles. The true parameter values are indicated by the horizontal grey lines. 65

4.4 Comparison of the posterior distributions of W and V through different SMC schemes given all the data with the MCMC output. Top row: 3×10^3 particles; middle row: 5×10^3 particles; bottom row: 10^4 particles. The true parameter values are indicated by the solid circles. 66

4.5 Sequential posterior means with 95% credible intervals of W_1, W_2, W_3 and V through different SMC schemes using 3×10^3 particles over time. The true parameter values are indicated by the horizontal grey lines. 69

4.6 Sequential posterior means with 95% credible intervals of W_1, W_2, W_3 and V through different SMC schemes using 5×10^3 particles over time. The true parameter values are indicated by the horizontal grey lines. 69

4.7 Sequential posterior means with 95% credible intervals of W_1, W_2, W_3 and V through different SMC schemes using 10^4 particles over time. The true parameter values are indicated by the horizontal grey lines. 70

4.8 Comparison of the posterior distributions of W_1, W_2, W_3 and V through different SMC schemes using 3×10^3 particles given all the data with the MCMC output. The true parameter values are indicated by the solid circles. 70

4.9 Comparison of the posterior distributions of W_1, W_2, W_3 and V through different SMC schemes using 5×10^3 particles given all the data with the MCMC output. The true parameter values are indicated by the solid circles. 71

4.10 Comparison of the posterior distributions of W_1, W_2, W_3 and V through different SMC schemes using 10^4 particles given all the data with the MCMC output. The true parameter values are indicated by the solid circles. 71

4.11 Comparison of the posterior distributions of W_1, W_2, W_3 and V through different SMC schemes using 10^6 particles given all the data with the MCMC output. The true parameter values are indicated by the solid circles. 72

5.1 Temperature and relative humidity data streams over time at each location. Periods of missingness are indicated just above the x-axis. 76

5.2 Scatter plots of temperature against relative humidity at each location. 77

5.3 Mean and 95% credible interval of the log Bayes factor comparing temperature sDLM against FDLM2 and FDLM1 against FDLM2, over time. 89

5.4 Mean and 95% credible interval of the log Bayes factor comparing humidity sDLM against FDLM2 and FDLM1 against FDLM2, over time. 90

5.5 Left: work flow of a parallel program in a shared memory system; right: work flow of a parallel program in a distributed memory system. 92

5.6 Actual speed-up by running parallel jobs through different numbers of cores in a distributed memory system. 92

6.1 Marginal parameter posterior densities obtained from the output of the full IBIS scheme with a standard serial resampling step (histograms) and a parallelised local resampling step (red). The true parameter values are shown as solid circles. 99

6.2 Marginal parameter posterior densities obtained from the output of the full IBIS scheme (histograms) and the online IBIS scheme with window widths $T = 100$ (yellow), $T = 300$ (blue) and $T = 500$ (red). The true parameter values are shown as solid circles. 100

6.3 Map showing site locations and a 10 km radius from each site, within which the spatial correlation for temperature is at least 0.76, and for humidity, is at least 0.64. 103

6.4 Mean (——) and 95% credible interval for the difference between the within-sample predictive and the observations, at each location (1–5) over time. The observation period is from 8th July 2017 04:00:00 to 29th July 2017 00:00:00. . 105

6.5 One-step ahead forecast mean (——) and 95% credible interval, at each location (1–5) over time. The observations are indicated (•). The observation period is from 12th July 2017 08:00:00 to 14th July 2017 00:00:00. 106

6.6 Two-step ahead forecast mean (——) and 95% credible interval, at each location (1–5) over time. The observations are indicated (•). The observation period is from 12th July 2017 08:00:00 to 14th July 2017 00:00:00. 107

List of Tables

2.1 ESS and ESS/sec for the parameters obtained by the Gibbs sampler and the MH algorithm. 31

2.2 ESS and ESS/sec for the parameters obtained by the Gibbs sampler and the MH algorithm 35

4.1 Comparison of the performance by LW, ST, PL, IBIS, aIBIS: CPU time (in seconds); bias (and RMSE in parentheses) of estimators of the posterior expectations $\widehat{E}(W|\boldsymbol{x}_{1:n})$, $\widehat{E}(V|\boldsymbol{x}_{1:n})$ and standard deviations $\widehat{SD}(W|\boldsymbol{x}_{1:n})$, $\widehat{SD}(V|\boldsymbol{x}_{1:n})$. All results are obtained by averaging over 100 runs of each SMC scheme. . . . 68

4.2 CPU time (in seconds) by averaging over 100 runs of LW, ST, PL, IBIS and aIBIS respectively. 72

4.3 Comparison of the performance by LW, ST, PL, IBIS, aIBIS: bias (and RMSE in parentheses) of estimators of the posterior expectations $\widehat{E}(W_1|\boldsymbol{x}_{1:n})$, $\widehat{E}(W_2|\boldsymbol{x}_{1:n})$, $\widehat{E}(W_3|\boldsymbol{x}_{1:n})$, $\widehat{E}(V|\boldsymbol{x}_{1:n})$ and standard deviations $\widehat{SD}(W_1|\boldsymbol{x}_{1:n})$, $\widehat{SD}(W_2|\boldsymbol{x}_{1:n})$, $\widehat{SD}(W_3|\boldsymbol{x}_{1:n})$, $\widehat{SD}(V|\boldsymbol{x}_{1:n})$. All results are obtained by averaging over 100 runs of each SMC scheme. 74

5.1 A summary of hourly average temperature and humidity data over the period 8th July 2017 to 31st December 2017 at five locations in North East England. . . . 77

6.1 Marginal parameter posterior medians and quantile-based 95% credible intervals obtained from the output of the online IBIS scheme. 102

Chapter 1

Introduction

Recent advances in sensor technology and data management mean that it is now possible to reliably and affordably collect data with respect to city operations from different locations. The data collection is merely the starting point. The essential part is to be able to analyse this spatial data and build accurate models which can be used to understand the stochastic nature, interdependencies and correlations between these data sets. This will require the combination of stochastic modelling for the non-stationary dynamic processes, efficient inference procedures and advanced computing architectures for big data.

Climate is one of the most important environmental factors which plays a critical role in the global mission of urban sustainability. Consequently, it has attracted tremendous attention from academic scientists and industrial experts in recent decades. The literature contains several temporal models for temperature at a single location. For example, Campbell and Diebold (2005) proposed an autoregressive (AR) model with Fourier components to account for seasonality, a polynomial deterministic trend and a generalised autoregressive conditional heteroscedasticity (GARCH) error process. Further AR modelling approaches have been proposed by Härdle and Cabrera (2012), Benth et al. (2007) and Benth and Benth (2012), with the latter adopting a continuous-time approach. Generic approaches for spatial data sets at multiple locations are widely available (see e.g. Cressie (1993), Stein (1999), Ripley (2004), Diggle and Ribeiro (2004), Gelfand et al. (2010), Cressie and Wikle (2011) and Banerjee et al. (2014)). By considering multiple variables jointly, Hu et al. (2013, 2015) use a stochastic partial differential equation (SPDE) to model yearly temperature and humidity data at 120 locations and perform fully Bayesian inference via an integrated nested Laplace approximation (Rue et al., 2009). Gamerman and Migon (1993) gave a list of hierachical dynamic linear models (DLMs) used for the state

evolution, smoothing and filtering through the stages of the hierarchy. A Bayesian hierachical model with the state process regressing on a spatial effect was discussed by Nott et al. (2001). Shaddick and Wakefield (2002) proposed a multivariate hierachical DLM with space-time regression structures on pollutant data studies.

Fully Bayesian estimation of the model parameters is necessary for appropriate forecasts and the model fit can be checked through posterior predictive distributions, which allow for uncertainty within the stochastic model. Markov chain Monte Carlo (MCMC) methods are primarily used for the offline learning of static parameters. For a DLM, a forward filter backward sampler (FFBS) (Carter and Kohn, 1994; Frühwirth-Schnatter, 1994) coupled with a Gibbs sampling approach, provides a strategy for state and parameter learning. When interest lies solely in the marginal parameter posterior, a marginal Metropolis-Hasting scheme is possible. Computational efficiency is one of the main barriers for use of MCMC in an online phase, as a new Markov chain has to be simulated when a new observation becomes available.

Sequential Monte Carlo (SMC) methods known as particle filters (Del Moral, 1996; Liu and Chen, 1998) are a set of online posterior density estimation algorithms allowing the posterior density of the model parameters to be updated sequentially through a series of weighted samples (known in this context as particles). A basic SMC scheme consists of a sequence of propagation and reweighting steps, which can be performed independently for each particle. The methods therefore permit the use of parallel computing. The main issue of the SMC schemes is particle degeneracy. That is, as time increases, only a small number of particles remain with non-zero weights. This problem can be alleviated (somewhat) through the resampling step to remove the particles with small weights and multiply the particles with large weigths. The bootstrap particle filter (Gordon et al., 1993) and the auxiliary particle filter (Pitt and Shephard, 1999; Pitt et al., 2012) were proposed based on this idea. However these methods are only applicable for fixed and known parameter values. Liu and West (2001) introduced a kernel smoothing method which adds a jitter to each particle of static parameters before particle propagation to the next time point. Storvik (2002), Fearnhead (2002), Carvalho et al. (2010) and Lopes et al. (2011) also proposed the algorithms that exploit the tractability of the conditional parameter posterior. Unfortunately, these methods do not overcome the particle degeneracy issue entirely (Chopin et al., 2010; Rios and Lopes, 2013). The iterated batch importance sampling (IBIS) algorithm (Chopin, 2002; Chopin et al., 2013) uses a resample-move step (referred to here as the rejuvenation step) (Gilks and Berzuini, 2001) in order to circumvent the degeneracy problem, where resampled paticles are mutated through a MCMC kernel. Additionally, Fearnhead and Taylor (2013) discussed an extension of the IBIS scheme by allowing the scaling of the tuning parameter in the rejuvenation steps to be changed adaptively. Unfortunately, the computational cost of the rejuvenation step

increases as the IBIS algorithm runs, precluding its use in real time.

1.1 Thesis aims

In this thesis we focus on understanding the relationship between temperature and humidity, as these are two of the most important factors in driving other climate processes. Our primary objective is the development of dynamic models which can be used to understand the stochastic nature of temperature and humidity, as well as quantify their spatial dependencies. Moreover, in order to facilitate accurate forecasts in real time, we focus on developing algorithms which allow inferences to be made sequentially in real time.

The modelling approach developed here is motivated by the fine scale temporal nature of the available data. Dynamic linear models (DLMs) are widely used for system evolution learning and short term forecasting due to their simple and practical structures; see, for example, West and Harrison (1999) for an introduction. We exploit these properties here by specifying a marginal DLM for temperature and a conditional DLM for humidity given temperature. We account for spatial dependence at nearby locations by adding a spatial Gaussian process to the system equations, thereby smoothing spatial deviations from the underlying temporal model.

We perform fully Bayesian inference for the model parameters as each observation becomes available. Since the posterior distribution is intractable, we use sequential Monte Carlo (SMC) methods that approximate the posterior distribution at each time point through a set of weighted samples; see Fearnhead and Künsch (2018) for a recent review of SMC methods. Although the posterior is intractable, the observed data likelihood is available in closed form, allowing the implementation of the iterated batch importance sampling (IBIS) scheme of Chopin (2002) (see also Chopin et al. (2013)). Essentially, parameter particles are incrementally weighted by the observed data likelihood contribution of the currently available observation. Particle degeneracy is mitigated via a resample-move step (Gilks and Berzuini, 2001) which 'moves' each parameter particle through a Metropolis-Hastings kernel that leaves the target invariant. This step can be executed subject to the fulfilment of some degeneracy criterion e.g. small effective sample size. As noted above, the computational cost of the resample-move step increases as the algorithm includes more data, as it requires calculation of the observed data likelihood of all available information. To obtain a novel online IBIS algorithm, where the computational cost of assimilating a single observation is bounded, we modify the resample-move step by basing the observed data likelihood on an observation window whose length is a tuning parameter,

chosen to balance accuracy and computational efficiency. We use a simulation study to formulate practical advice on how to choose the size of this window.

Further computational savings can be made by employing a high performance computing system. Whilst the weighting and move steps can be performed independently for each particle, a basic implementation of the resampling step requires collective operations, such as adding up the particle weights. Our approach is to use a simple strategy which performs the resampling step independently for batches of parameter samples, thus allowing a fully parallel (per parameter batch) implementation of the algorithm to be performed. We quantify the effect of the approximation induced by this approach using synthetic data. Finally, we apply the online IBIS scheme (with parallel implementation) to the observed dataset and examine the model reliability and forecast accuracy through comparison of observed measurements with their posterior predictive distribution.

1.2 Outline of thesis

The remainder of this thesis is organised as follows. In Chapter 2, we start with a review of the generic structure of the DLM. To account for the seasonality of the data, we introduce the sinusoidal form DLM and the Fourier form DLM that allows for multiple harmonics. A review of Markov chain Monte Carlo (MCMC) is given, with the details of the Gibbs sampler and the Metropolis-Hastings algorithm, followed by a brief discussion of some commonly used MCMC diagnostics. We give the within-sample predictive and out-of-sample forecast distributions for posterior predictive checks, which can be used to investigate the model fit. The chapter is concluded with comparison between the Gibbs sampler and the Metropolis-Hastings algorithm by fitting two DLMs to synthetic data.

In Chapter 3, we comprehensively review several state-of-the-art sequential Monte Carlo (SMC) methods. We initially introduce the fundamental concept of importance (re)sampling. Following that, we introduce sequential importance sampling (SIS), the bootstrap particle filter (BPF) and the auxiliary particle filter (APF), which are the sequential Bayesian schemes that can be applied to the problem of state filtering when the parameters are fixed and known. We also detail the schemes which are applicable to the presence of unknown parameters. They include the Liu-West algorithm, the Storvik algorithm and particle learning. The main issue of these schemes is particle degeneracy, where only a few particles with reasonable weights remain. We introduce the iterated batch importance sampling (IBIS) algorithm and its adaptive version

(aIBIS) that effectively circumvent particle degeneracy through the use of MCMC steps for particle rejuvenation. Finally, we provide a novel online version of IBIS which allows a bounded computational cost by running IBIS through separated observation windows.

A series of simulation studies for the comparison of different SMC schemes is presented in Chapter 4. We first compare the fully adapted APF and BPF by examining their posterior estimations of the state. Both schemes are implemented by fitting a simple DLM, the local level model to a synthetic data set. In the remainder of the chapter, we compare the Liu-West algorithm, the Storvik algorithm, particle learning, IBIS and aIBIS by fitting the local level model and the sinusoidal DLM respectively by assuming the model parameters are unknown. We find that the IBIS scheme offers the best performance, in terms of the accuracy of the posterior estimation and the computational cost.

In Chapter 5, the background of the real data collection is briefly introduced followed by a discussion of the structures of the spatial DLMs for temperature and humidity. A general Bayesian inference approach for handling missing data is then outlined. We give details of the IBIS and online IBIS schemes for the spatial temperature DLM. We access the seasonality of temperature data by fitting the sinusoidal form DLM and the Fourier form DLMs separately, and compare these models by using the concept of the Bayes factor. Due to the independence of operations on each particle, we can take advantage of parallel computing technics for implementing the SMC schemes. The details of conducting the parallel jobs in shared and distributed memory system are discussed. Resampling steps preclude full parallelisation of SMC schemes. We therefore consider a local resampling method, where the particles are partitioned into disjoint subsets and the algorithm can be run for each subset independently.

The effect of local resampling on posterior accuracy is considered in Chapter 6, using synthetic data. Furthermore, online IBIS is compared with full IBIS, before the former is applied to the real data. Posterior predictive checks are presented.

Conclusions are drawn in Chapter 7. In addition, we discuss some potential future work.

Chapter 2

Dynamic linear models

2.1 Introduction

A time series is a sequence of data points obtained at successive times. Due to advanced measurement techniques, time series data can be recorded on most aspects of our lives to capture the changes of nature and human behaviour, such as, weather, environment, energy consumption, stock price, etc; see Figure 2.1.

State space models, developed over last few decades, provide an abundant family of interpretable models for analysing time series data (see e.g. Harvey, 1989; West and Harrison, 1999). The hidden state is described by a latent component and is used to drive the stochastic process of a data stream. The state is typically unobserved, and evolves itself over time. Figure 2.2 shows the evolution of a simple univariate state space in which the continuous valued latent state process $\{\theta_0, \theta_1, \ldots, \theta_{t-1}, \theta_t, \ldots\}$ evolves according to a first order Markov chain with transition density $\pi(\theta_t|\theta_{t-1})$. The continuous-valued observation process $\{x_1, x_2, \ldots, x_{t-1}, x_t, \ldots\}$ is linked to the latent state process at an arbitrary time t via the density $\pi(x_t|\theta_t)$ and here it is assumed that the observations are conditionally independent given the state process.

The dynamic linear model (DLM) is a special form of the state space model, with x_t and θ_t evolving according to a Gaussian distribution with linear dependence on θ_{t-1}. Due to its simple and practical structure, the DLM is widely used for system evolution learning and short term

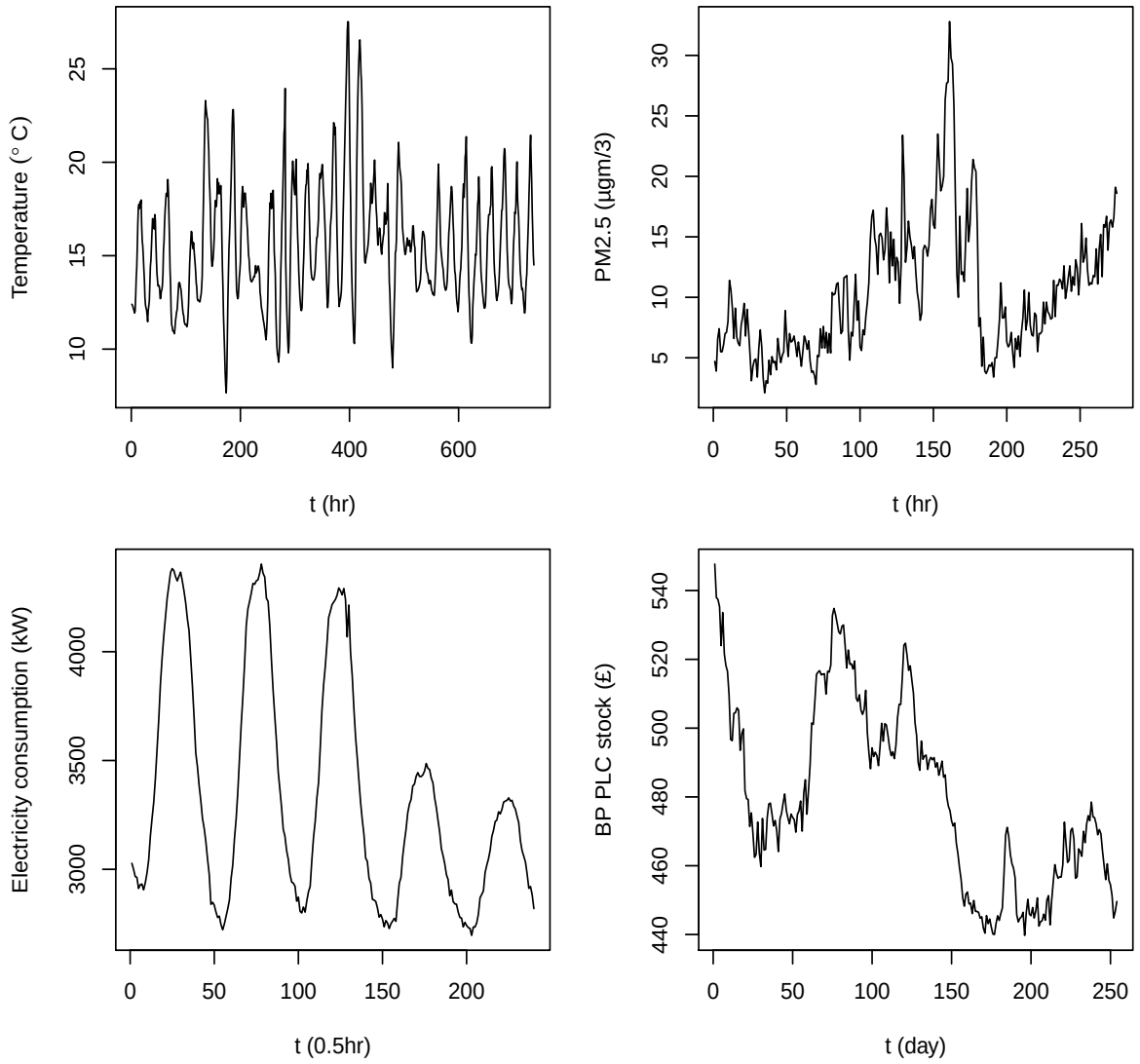

Figure 2.1: A variety of time series data.

forecasting (West and Harrison, 1999). For the univariate case, a DLM may be written as

$$X_t|\theta_t \sim N(F_t\theta_t, V_t),$$
$$\theta_t|\theta_{t-1} \sim N(G_t\theta_{t-1}, W_t)$$

where F_t, G_t, V_t and W_t are deterministic functions of t which may be constant in practice. Taking $\theta_0 \sim N(m_0, C_0)$ for suitably chosen hyper-parameters m_0 and C_0 completes the specification of

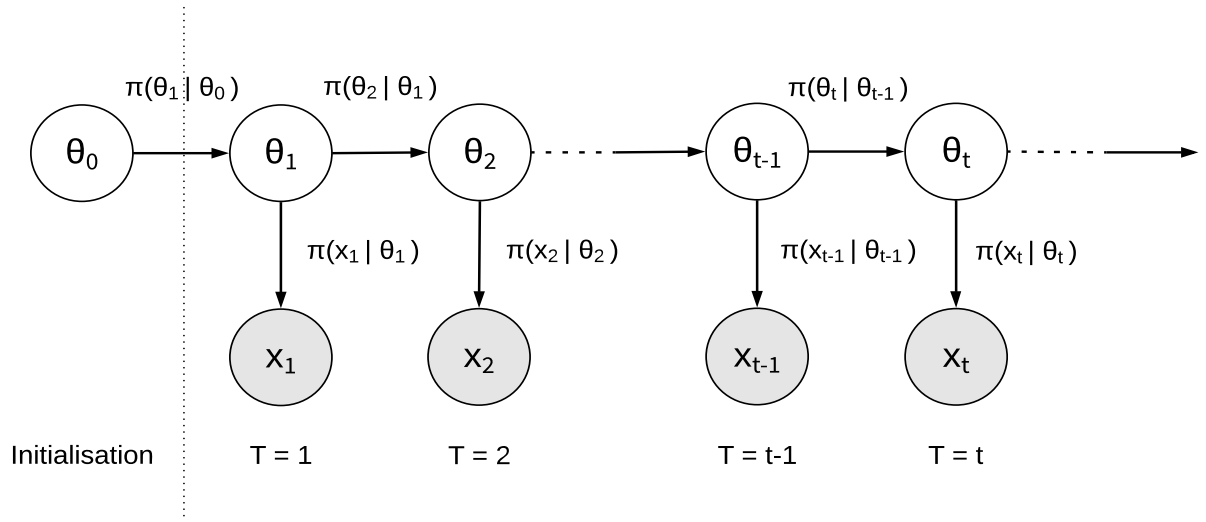

Figure 2.2: Diagram of the state space model.

the DLM. In practice, the DLM is usually written as

$$X_t = F_t\theta_t + v_t, \qquad v_t \overset{indep}{\sim} N(0, V_t),$$
$$\theta_t = G_t\theta_{t-1} + w_t, \qquad w_t \overset{indep}{\sim} N(0, W_t),$$

and these equations are known as the observation and system equation respectively. The observation error v_t and the system error w_t are white noise processes with the mean 0 and the variances V_t and W_t respectively.

The simplest univariate DLM is the locally constant DLM, also known as the local level model. This model has constant functions $F_t = 1$, $G_t = 1$ and variance componsents $V_t = V$, $W_t = W$. The observation and system equations are

$$X_t = \theta_t + v_t, \qquad v_t \overset{indep}{\sim} N(0, V), \tag{2.1}$$
$$\theta_t = \theta_{t-1} + w_t, \qquad w_t \overset{indep}{\sim} N(0, W). \tag{2.2}$$

Hence, the state process evolves according to a simple random walk from which noisy observations are taken. We will use this simple DLM to benchmark the performance of various competing inference schemes in Chapter 3.

For most practical applications, we will require θ_t and X_t to be vectors. Extending the univariate DLM to accommodate multivariate time series data is straightforward. The multivariate DLM

can be written generically as

$$\boldsymbol{X}_t = \boldsymbol{F}_t\boldsymbol{\theta}_t + \boldsymbol{v}_t, \qquad \boldsymbol{v}_t \overset{indep}{\sim} N(\boldsymbol{0}, \boldsymbol{V}_t), \tag{2.3}$$

$$\boldsymbol{\theta}_t = \boldsymbol{G}_t\boldsymbol{\theta}_{t-1} + \boldsymbol{w}_t, \qquad \boldsymbol{w}_t \overset{indep}{\sim} N(\boldsymbol{0}, \boldsymbol{W}_t) \tag{2.4}$$

where

- $\boldsymbol{X}_t$ is an $l \times 1$ vector;
- $\boldsymbol{\theta}_t$ is an $m \times 1$ vector;
- $\boldsymbol{F}_t$ is the $l \times m$ observation matrix;
- $\boldsymbol{G}_t$ is the $m \times m$ system matrix;
- $\boldsymbol{V}_t$ is the $l \times l$ observation variance;
- $\boldsymbol{W}_t$ is the $m \times m$ system variance.

Taking an initial distribution $\boldsymbol{\theta}_0 \sim \boldsymbol{N}(\boldsymbol{m}_0, \boldsymbol{C}_0)$, where $\boldsymbol{m}_0$ and $\boldsymbol{C}_0$ are known hyper-parameters, completes specification of the multivariate DLM.

2.2 Seasonality

Cyclical behaviour or seasonality is a significant feature of many time series. Due to the periodic patten, it is natural to incorporate harmonic functions within the DLM. We consider two approaches for modelling seasonality: (i) a single harmonic in the observation equation and (ii) multiple harmonics in the state equation. For ease of notation in what follows, we consider a univariate observation process.

2.2.1 Sinusoidal form DLM

The observation equation is given by

$$X_t = \boldsymbol{F}_t\boldsymbol{\theta}_t + v_t, \qquad v_t \overset{indep}{\sim} N(0, V_t) \tag{2.5}$$

where $\boldsymbol{\theta}_t = (\theta_{t,1}, \theta_{t,2}, \theta_{t,3})^T$ and the observation matrix is given by

$$\boldsymbol{F}_t = (\cos(2\pi t/P_x), \sin(2\pi t/P_x), 1)$$

where P_x is the time corresponding to one complete period. The system equation takes the form of (2.4) with $\boldsymbol{G}_t = \mathbb{I}_3$. For our applications, we take $V_t = V$ and

$$\boldsymbol{W}_t = \boldsymbol{W} = \text{diag}(W_1, W_2, W_3) = \begin{pmatrix} W_1 & 0 & 0 \\ 0 & W_2 & 0 \\ 0 & 0 & W_3 \end{pmatrix}.$$

We refer to this model as the sinusoidal form DLM. Note that the observation equation can be written as

$$X_t = \tilde{\theta}_{t,2} \cos\left(\frac{2\pi t}{P_x} - \tilde{\theta}_{t,1}\right) + \theta_{t,3} + v_t \tag{2.6}$$

where the parameters in (2.5) and (2.6) are related using

$$\tilde{\theta}_{t,1} = \sqrt{\theta_{t,1}^2 + \theta_{t,2}^2}, \qquad \tilde{\theta}_{t,2} = \tan^{-1}\left(\frac{\theta_{t,2}}{\theta_{t,1}}\right). \tag{2.7}$$

Hence, the sinusoidal form DLM captures seasonality (about a time varying basal level $\theta_{t,3}$) via a sinusoid whose amplitude $\tilde{\theta}_{t,2}$ and phase $\tilde{\theta}_{t,1}$ vary according to two independent random walk processes.

2.2.2 Fourier form DLM

When time series data exhibit more complex cyclical behaviour, additional harmonics may be used. The Fourier form DLM captures seasonality through linear combinations of periodic functions; see, for example, West and Harrison (1999) and Petris et al. (2009). By definition, a Fourier series is written as

$$X(t) = \sum_{r=1}^{q} \left\{ A_r \cos\left(\frac{2\pi r t}{P_x}\right) + B_r \sin\left(\frac{2\pi r t}{P_x}\right) \right\} + A_0, \tag{2.8}$$

that is, the summation of q harmonic functions and a mean level term. To construct a Fourier form DLM, let the observation matrix be the $1 \times (2q + 1)$ matrix partitioned as

$$\boldsymbol{F}_t = (1, 0 | 1, 0 | \ldots | 1), \tag{2.9}$$

so that the state vector $\boldsymbol{\theta}_t$ is of length $2q + 1$ and satisfies a system equation of the form (2.4) with system matrix

$$\boldsymbol{G}_t = \boldsymbol{G} = \text{diag}(\boldsymbol{H}_1, \ldots, \boldsymbol{H}_q, 1). \tag{2.10}$$

Here, $\boldsymbol{H}_r$ is a harmonic matrix given by

$$\boldsymbol{H}_r = \begin{pmatrix} \cos(2\pi r/P_x) & \sin(2\pi r/P_x) \\ -\sin(2\pi r/P_x) & \cos(2\pi r/P_x) \end{pmatrix}, \quad r = 1, \ldots, q.$$

The Fourier form DLM is then given by (2.3) and (2.4) with $\boldsymbol{F}_t$ and $\boldsymbol{G}_t$ as in (2.9) and (2.10).

To understand the model structure, consider the simple case in which the system variance $\boldsymbol{W} = \boldsymbol{0}$. Suppose that $\boldsymbol{\theta}_0^r = \left(\theta_{0,1}^r, \theta_{0,2}^r\right)$ for $r = 1, \ldots, q$ so that the initial state is

$$\boldsymbol{\theta}_0 = \left(\boldsymbol{\theta}_0^1, \ldots, \boldsymbol{\theta}_0^q, \theta_{0,3}\right)^T.$$

The state at $t = 1$ is

$$\boldsymbol{\theta}_1 = \boldsymbol{G}\boldsymbol{\theta}_0 = \left(\boldsymbol{H}_1\boldsymbol{\theta}_0^1, \ldots, \boldsymbol{H}_q\boldsymbol{\theta}_0^q, \theta_{0,3}\right)^T.$$

Substituting $\boldsymbol{\theta}_1$ into the observation equation (2.3), we obtain

$$X_1 = \sum_{r=1}^{q} \left(\theta_{0,1}^r \cos\left(\frac{2\pi r}{P_x}\right) + \theta_{0,2}^r \sin\left(\frac{2\pi r}{P_x}\right)\right) + \theta_{0,3} + v_1.$$

Similarly, when $t = 2$, by ignoring w_t we have that

$$\boldsymbol{\theta}_2 = \boldsymbol{G}^2\boldsymbol{\theta}_0 = \left(\boldsymbol{H}_1^2\boldsymbol{\theta}_0^1, \ldots, \boldsymbol{H}_q^2\boldsymbol{\theta}_0^q, \theta_{0,3}\right)^T.$$

Due to the property of the harmonic matrix, that is $(\boldsymbol{H}_r)^t = \boldsymbol{H}_{tr}$, the state can be rewritten as

$$\boldsymbol{\theta}_2 = \left(\boldsymbol{H}_2\boldsymbol{\theta}_0^1, \ldots, \boldsymbol{H}_{2q}\boldsymbol{\theta}_0^q, \theta_{0,3}\right)^T.$$

Therefore the observation equation at $t = 2$ is

$$X_2 = \sum_{r=1}^{q} \left(\theta_{0,1}^r \cos\left(\frac{2\pi r}{P_x} \times 2\right) + \theta_{0,2}^r \sin\left(\frac{2\pi r}{P_x} \times 2\right)\right) + \theta_{0,3} + v_2,$$

Recursive use in this way of the property of the harmonic matrix gives

$$X_t = \sum_{r=1}^{q} \left(\theta_{0,1}^r \cos\left(\frac{2\pi rt}{P_x}\right) + \theta_{0,2}^r \sin\left(\frac{2\pi rt}{P_x}\right)\right) + \theta_{0,3} + v_t \tag{2.11}$$

making clear the link between the Fourier form DLM and the Fourier series in (2.8). Note that when using the $q = 1$ harmonic, the observation equation of the Fourier form DLM coincides with that of the sinusoidal form in (2.5) given by

$$X_t = \theta_{0,1} \cos\left(\frac{2\pi t}{P_x}\right) + \theta_{0,2} \sin\left(\frac{2\pi t}{P_x}\right) + \theta_{0,3} + v_t.$$

However, in the case of non-zero $\boldsymbol{W}$, the error structures differ due to the use of the harmonic in the system equation for the Fourier form DLM, and in the observation equation for the sinusoidal form DLM.

Note that the number of harmonics q must be specified by the practitioner. Consequently, Fourier models with $q = 1, 2$ are typically used in practice to limit the size of the parameter space (Petris et al., 2009).

2.3 Bayesian inference

Without loss of generality, consider equally spaced data $\boldsymbol{x}_{1:n} = (x_1, \ldots, x_n)^T$ to which we wish to fit a DLM with latent states $\boldsymbol{\theta}_{0:n} = (\theta_0, \ldots, \theta_n)^T$ and static parameters $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_{n_{par}})^T$. For ease of notation, we consider a univariate DLM with variance components $V_t = V$ and $W_t = W$ but note that extension of the methodology introduced here to multivariate DLMs with arbitrary variance components is straightforward in principle. Upon ascribing a prior density $\pi(\boldsymbol{\phi})$ to $\boldsymbol{\phi}$, Bayesian inference may proceed via the joint posterior density

$$\pi(\boldsymbol{\phi}, \boldsymbol{\theta}_{0:n} | \boldsymbol{x}_{1:n}) \propto \pi(\boldsymbol{\phi}) \pi(\theta_0) \pi(\boldsymbol{\theta}_{1:n} | \boldsymbol{\phi}) \pi(\boldsymbol{x}_{1:n} | \boldsymbol{\theta}_{1:n}, \boldsymbol{\phi}) \tag{2.12}$$

where

$$\pi(\boldsymbol{x}_{1:n} | \boldsymbol{\theta}_{1:n}, \boldsymbol{\phi}) = \prod_{t=1}^{n} N\left(x_t; F_t \theta_t, V\right),$$

and

$$\pi(\boldsymbol{\theta}_{1:n} | \boldsymbol{\phi}) = \prod_{t=1}^{n} N\left(\theta_t; G_t \theta_{t-1}, W\right).$$

Here $N(\cdot; m, V)$ denotes a normal density with mean (vector) m and variance (matrix) V. If primary interest lies solely in the static parameter vector $\boldsymbol{\phi}$ then we may consider the marginal posterior

$$\pi(\boldsymbol{\phi} | \boldsymbol{x}_{1:n}) \propto \pi(\boldsymbol{\phi}) \pi(\boldsymbol{x}_{1:n} | \boldsymbol{\phi}) \tag{2.13}$$

where $\pi(\boldsymbol{x}_{1:n}|\boldsymbol{\phi})$ is the observed data likelihood given by

$$\pi(\boldsymbol{x}_{1:n}|\boldsymbol{\phi}) = \pi(x_1|\boldsymbol{\phi}) \prod_{t=2}^{n} \pi(x_t|\boldsymbol{x}_{1:t-1}, \boldsymbol{\phi}). \tag{2.14}$$

In practice, (2.12) and (2.13) are intractable, that is, closed form expressions for these densities cannot be obtained, necessitating the use of stochastic simulation algorithms such as Markov chain Monte Carlo (MCMC) and sequential Monte Carlo (SMC). In this chapter, we focus on the former, and provide two MCMC schemes for generating draws from (2.12) and (2.13). These are: the Gibbs sampler and (marginal) Metropolis-Hastings algorithm. For an in-depth discussion of MCMC, we refer the reader to Gamerman and Lopes (2006).

2.3.1 Gibbs sampler

The Gibbs sampler was first introduced by Geman and Geman (1984) for generating draws from Gibbs distributions, by alternately sampling from full conditional distributions. Gelfand and Smith (1990) were the first to point out to the wider statistics community that the approach could be used within the Bayesian paradigm, for scenarios where full conditional (posterior) distributions are available for sampling from. As with other MCMC schemes, the Gibbs sampler simulates from a Markov chain whose stationary distribution is the target (posterior) distribution of interest. The transition density is formed by the full conditional distributions.

Suppose now that interest lies in the marginal posterior $\pi(\boldsymbol{\phi}|\boldsymbol{x}_{1:n})$ given by (2.13). The Gibbs sampler is given by Algorithm 1. The Gibbs sampler defines a homogeneous Markov chain with transition density

$$\pi(\boldsymbol{\phi}|\tilde{\boldsymbol{\phi}}, \boldsymbol{x}_{1:n}) = \prod_{i=1}^{n_{par}} \pi(\phi_i|\phi_1, \ldots, \phi_{i-1}, \tilde{\phi}_{i+1}, \ldots, \tilde{\phi}_{n_{par}}, \boldsymbol{x}_{1:n}),$$

where n_{par} is the number of parameters.

Checking stationarity of $\pi(\boldsymbol{\phi}|\boldsymbol{x}_{1:n})$ can be achieved by showing that

$$\pi(\boldsymbol{\phi}|\boldsymbol{x}_{1:n}) = \int \pi(\boldsymbol{\phi}|\tilde{\boldsymbol{\phi}}, \boldsymbol{x}_{1:n}) \pi(\tilde{\boldsymbol{\phi}}|\boldsymbol{x}_{1:n}) d\tilde{\boldsymbol{\phi}} \tag{2.15}$$

using induction (Gamerman and Lopes, 2006). Formal convergence conditions can be found in Roberts and Smith (1994).

Algorithm 1 Gibbs sampler

1. Initialise the chain with $\boldsymbol{\phi}^{(0)} = \left(\phi_1^{(0)}, \ldots, \phi_{n_{par}}^{(0)}\right)^T$. For $j = 1, \ldots, N$:
2. Obtain a new value $\boldsymbol{\phi}^{(j)}$ from $\boldsymbol{\phi}^{(j-1)}$ by continuous sampling from the full conditional distributions:
   - $\phi_1^{(j)} \sim \pi(\phi_1 | \phi_2^{(j-1)}, \ldots, \phi_{n_{par}}^{(j-1)}, \boldsymbol{x}_{1:n})$
   - $\phi_2^{(j)} \sim \pi(\phi_2 | \phi_1^{(j)}, \phi_3^{(j-1)}, \ldots, \phi_{n_{par}}^{(j-1)}, \boldsymbol{x}_{1:n})$

     $\vdots$
   - $\phi_{n_{par}}^{(j)} \sim \pi(\phi_{n_{par}} | \phi_1^{(j)}, \ldots, \phi_{n_{par}-1}^{(j)}, \boldsymbol{x}_{1:n})$
3. Set $j := j + 1$. Return to step 2.

Upon discarding a number of sampled parameter values as "burn-in", the remaining samples of $\boldsymbol{\phi}^{(j)}$, $j = 1, \ldots, N$ are treated as realisations from $\pi(\boldsymbol{\phi} | \boldsymbol{x}_{1:n})$. The Gibbs sampler provides an efficient sampling mechanism provided that the full conditional distributions $\pi(\phi_i | \boldsymbol{\phi}_{-i}, \boldsymbol{x}_{1:n})$, $i = 1, \ldots, n_{par}$, have standard forms and are easy to sample from. However, for the DLMs considered here, the parameter full conditionals are likely to be intractable, after marginalising out the dynamic states. Neverthless, it is often the case that conditional on the data and the dynamic states, parameter (component) full conditionals will be available for sampling from (for a suitable choice of prior). Moreover, the full conditional density of the dynamic states, $\pi(\boldsymbol{\theta}_{1:n} | \boldsymbol{\phi}, \boldsymbol{x}_{1:n})$, is tractable. Consequently, the most natural implementation of the Gibbs sampler targets the joint posterior $\pi(\boldsymbol{\phi}, \boldsymbol{\theta}_{0:n} | \boldsymbol{x}_{1:n})$ by alternating between the following steps:

1. simulate $\boldsymbol{\theta}_{0:n}$ from $\pi(\boldsymbol{\theta}_{0:n} | \boldsymbol{\phi}, \boldsymbol{x}_{1:n})$,
2. simulate $\boldsymbol{\phi}$ from $\pi(\boldsymbol{\phi} | \boldsymbol{\theta}_{0:n}, \boldsymbol{x}_{1:n})$.

Performing step 1 is achieved using a forward filtering backward sampling algorithm, which we now describe in detail.

Forward filtering backward sampling (FFBS)

As above, for clarity of exposition we consider a univariate DLM with static variance components $V_t = V$ and $W_t = W$.

Consider the task of generating $\boldsymbol{\theta}_{0:n} \sim \pi(\boldsymbol{\theta}_{0:n}|\phi, \boldsymbol{x}_{1:n})$. Note that the parameter vector ϕ remains fixed throughout this section and we therefore drop it from the notation where possible. A widely used method is known as forward filtering backward sampling (FFBS) (Carter and Kohn, 1994; Frühwirth-Schnatter, 1994). The basic idea is to first calculate the filtering distributions $\pi(\theta_t|\boldsymbol{x}_{1:t})$ recursively, for $t = 1, \ldots, n$. This is achieved using a forward filter also known as a Kalman filter (Kalman, 1960). Upon reaching $t = n$, we may draw θ_n from the marginal posterior $\pi(\theta_n|\boldsymbol{x}_{1:n})$. The remaining dynamic states are then drawn in a backward sweep by exploiting the decomposition

$$\pi(\boldsymbol{\theta}_{0:n}|\boldsymbol{x}_{1:n}) = \pi(\theta_n|\boldsymbol{x}_{1:n})\pi(\theta_0|\theta_1)\prod_{t=1}^{n-1}\pi(\theta_t|\theta_{t+1}, \boldsymbol{x}_{1:t}). \tag{2.16}$$

Note that due to the linear Gaussian structure of the DLM, both the filtering densities $\pi(\theta_t|\boldsymbol{x}_{1:t})$ and the densities $\pi(\theta_t|\theta_{t+1}, \boldsymbol{x}_{1:t})$ required in the backward sweep are Gaussian densities.

In detail, consider the forward filter and suppose that at time $t-1$, having observed $\boldsymbol{x}_{1:t-1}$, we have the currently available posterior density $\pi(\theta_{t-1}|\boldsymbol{x}_{1:t-1}) = N(\theta_{t-1}; m_{t-1}, C_{t-1})$. Using the system equation (2.4) it should be clear that at time t the prior density of θ_t is

$$\pi(\theta_t|\boldsymbol{x}_{1:t-1}) = N(\theta_t; G_t m_{t-1}, R_t)$$

where $R_t = G_t C_{t-1} G_t^T + W$. Then using the observation equation (2.3), we obtain the density of the one step forecast as

$$\pi(x_t|\boldsymbol{x}_{1:t-1}) = N\left(x_t; F_t G_t m_{t-1}, F_t R_t F_T^T + V\right). \tag{2.17}$$

Note that after observing x_t, (2.17) gives the observed data likelihood increment. To update the currently available posterior density, we first construct the joint distribution of θ_t and X_t. After noting that $Cov(\theta_t, X_t) = Cov(\theta_t, F_t\theta_t + v_t) = R_t F_t^T$, we obtain

$$\begin{pmatrix}\theta_t \\ X_t\end{pmatrix} \Bigg| \boldsymbol{x}_{1:t-1} \sim N\left\{\begin{pmatrix}G_t m_{t-1} \\ F_t G_t m_{t-1}\end{pmatrix}, \begin{pmatrix}R_t & R_t F_t^T \\ F_t R_t & F_t R_t F_t^T + V\end{pmatrix}\right\}.$$

Conditioning on $X_t = x_t$ using standard multivariate normal results yields the updated posterior, i.e. the required filtering distribution, as

$$\pi(\theta_t|\boldsymbol{x}_{1:t}) = N(\theta_t; m_t, C_t)$$

where

$$m_t = G_t m_{t-1} + R_t F_t^T (F_t R_t F_t^T + V)^{-1}(x_t - F_t G_t m_{t-1}),$$
$$C_t = R_t - R_t F_t^T (F_t R_t F_t^T + V)^{-1} F_t R_t.$$

These steps can be executed for $t = 1, \ldots, n$ to give the forward filter.

To obtain the constituent terms in (2.16) necessary for executing the backward sweep, note that the joint density of θ_t and θ_{t+1} (given $\boldsymbol{x}_{1:t}$) is

$$\begin{pmatrix} \theta_t \\ \theta_{t+1} \end{pmatrix} \Bigg| \, \boldsymbol{x}_{1:t} \sim N\left\{ \begin{pmatrix} m_t \\ G_{t+1} m_t \end{pmatrix}, \begin{pmatrix} C_t & C_t G_{t+1}^T \\ G_{t+1} C_t & G_{t+1} C_t G_{t+1}^T + W \end{pmatrix} \right\}.$$

Conditioning on θ_{t+1} gives

$$\pi(\theta_t | \theta_{t+1}, \boldsymbol{x}_{1:t}) = N(\theta_t; h_t, H_t)$$

where

$$h_t = m_t + C_t G_{t+1}^T (G_{t+1} C_t G_{t+1}^T + W)^{-1}(\theta_{t+1} - G_{t+1} m_t),$$
$$H_t = C_t - C_t G_{t+1}^T (G_{t+1} C_t G_{t+1}^T + W)^{-1} G_{t+1} C_t.$$

Algorithm 2 gives the full form of the FFBS scheme. Note the convention that $\boldsymbol{x}_{1:0}$ denotes the empty set.

Example: local level model

Consider the local level model given by (2.1) and (2.2). Let $\phi = (V, W)^T$. The Gibbs sampler iterates over the following steps:

1. simulate $\boldsymbol{\theta}_{0:n}$ from $\pi(\boldsymbol{\theta}_{0:n} | \phi, \boldsymbol{x}_{1:n})$,
2. simulate ϕ_1 from $\pi(\phi_1 | \phi_2, \boldsymbol{\theta}_{0:n}, \boldsymbol{x}_{1:n}) = \pi(\phi_1 | \boldsymbol{\theta}_{0:n}, \boldsymbol{x}_{1:n})$,
3. simulate ϕ_2 from $\pi(\phi_2 | \phi_1, \boldsymbol{\theta}_{0:n}, \boldsymbol{x}_{1:n}) = \pi(\phi_2 | \boldsymbol{\theta}_{0:n})$.

Algorithm 2 FFBS scheme

Forward Filtering:

1. Initial distribution: $\theta_0 \sim N(m_0, C_0)$. Store the values of m_0 and C_0.
2. For $t = 1, \ldots, n$,
   (a) Prior at t. Using the system equation, we have that
   $$\theta_t | \boldsymbol{x}_{1:t-1} \sim N(G_t m_{t-1},\, G_t C_{t-1} G_t^T + W).$$
   Store $R_t = G_t C_{t-1} G_t^T + W$.
   (b) One step forecast. Using the observation equation, we have that
   $$X_t | \boldsymbol{x}_{1:t-1} \sim N\left(F_t G_t m_{t-1},\, F_t R_t F_t^T + V\right). \tag{2.18}$$
   Store the marginal likelihood contribution
   $$\pi(x_t | \boldsymbol{x}_{1:t-1}) = N\left(x_t; F_t G_t m_{t-1}\,,\, F_t R_t F_T^T + V\right).$$
   (c) Posterior at t: $\theta_t | \boldsymbol{x}_{1:t} \sim N(m_t,\, C_t)$, where
   $$\begin{aligned} m_t &= G_t m_{t-1} + R_t F_t^T (F_t R_t F_t^T + V)^{-1} (x_t - F_t G_t m_{t-1}), \\ C_t &= R_t - R_t F_t^T (F_t R_t F_t^T + V)^{-1} F_t R_t. \end{aligned}$$
   Store the values of m_t and C_t.

Backward Sampling:

3. Sample $\theta_n | \boldsymbol{x}_{1:n} \sim N(m_n,\, C_n)$.
4. For $t = n-1, \ldots, 0$,
   (a) Backward distribution: $\theta_t | \theta_{t+1}, \boldsymbol{x}_{1:t} \sim N(h_t,\, H_t)$, where
   $$\begin{aligned} h_t &= m_t + C_t G_{t+1}^T (G_{t+1} C_t G_{t+1}^T + W)^{-1} (\theta_{t+1} - G_{t+1} m_t), \\ H_t &= C_t - C_t G_{t+1}^T (G_{t+1} C_t G_{t+1}^T + W)^{-1} G_{t+1} C_t. \end{aligned}$$
   (b) Sample $\theta_t | \theta_{t+1}, \boldsymbol{x}_{1:t} \sim N(h_t,\, H_t)$.

Step 1 can be executed using the FFBS algorithm described in Algorithm 2. It now remains to derive full conditional distributions for the components of ϕ. Specifying independent inverse gamma prior distributions for ϕ_1 and ϕ_2 permits a semi-conjugate analysis (that is, leads to tractable full conditional distributions). We take $\phi_1 \sim IG(\alpha_v, \beta_v)$ and $\phi_2 \sim IG(\alpha_w, \beta_w)$ with

prior hyper-parameters α_v, β_v, α_w, β_w. The full conditional density of ϕ_1 is

$$\begin{aligned}
\pi(\phi_1|\phi_2, \theta_{0:n}, x_{1:n}) &\propto \pi(\boldsymbol{\phi}, \theta_{0:n}, x_{1:n}) \\
&\propto \pi(x_{1:n}|\theta_{1:n}, \phi_1)\pi(\theta_{1:n}|\phi_2)\pi(\theta_0)\pi(\boldsymbol{\phi}) \\
&\propto \underbrace{\prod_{t=1}^{n} \pi(x_t|\theta_t, \phi_1)}_{\text{observation equation}} \underbrace{\prod_{t=1}^{n} \pi(\theta_t|\theta_{t-1}, \phi_2)}_{\text{system equation}} \pi(\theta_0)\pi(\phi_1)\pi(\phi_2) \\
&\propto \pi(\phi_1) \prod_{t=1}^{n} \pi(x_t|\theta_t, \phi_1) \\
&\propto \phi_1^{-(\alpha_v + \frac{n}{2})-1} \exp\left(- \left[\beta_v + \frac{1}{2} \sum_{t=1}^{n} (x_t - \theta_t)^2 \right] \Big/ \phi_1 \right)
\end{aligned}$$

Therefore, we obtain

$$\phi_1|\theta_{1:n}, x_{1:n} \sim IG\left(\alpha_v + \frac{n}{2}, \beta_v + \frac{1}{2} \sum_{t=1}^{n} (x_t - \theta_t)^2 \right).$$

Similar arguments lead to

$$\phi_2|\theta_{0:n} \sim IG\left(\alpha_w + \frac{n}{2}, \beta_w + \frac{1}{2} \sum_{t=1}^{n} (\theta_t - \theta_{t-1})^2 \right).$$

2.3.2 Marginal Metropolis-Hastings (MH) algorithm

The Gibbs sampler provides a natural mechanism for sampling the joint posterior $\pi(\boldsymbol{\phi}, \boldsymbol{\theta}_{0:n}|\boldsymbol{x}_{1:n})$, provided that full conditional distributions are available for sampling from. Retaining only draws of $\boldsymbol{\phi}$ gives a sample from the marginal parameter posterior $\pi(\boldsymbol{\phi}|\boldsymbol{x}_{1:n})$. However, in scenarios where there is high dependence between the static parameters and dynamic states, it can be beneficial to directly target the marginal parameter posterior via MCMC. Since $\pi(\boldsymbol{\phi}|\boldsymbol{x}_{1:n})$ is typically intractable, we consider a Metropolis-Hastings (MH) algorithm, which can be used for sampling a generic target density that is only known up to a multiplicative constant.

The Metropolis-Hastings algorithm, first introduced by Metropolis et al. (1953) and then generalised by Hastings (1970), constructs a reversible Markov chain whose stationary distribution is the target of interest. Two key ingredients are the proposal distribution with density $q(\boldsymbol{\phi}^*|\boldsymbol{\phi})$ (which should be easy to sample from) and an acceptance probability $A(\boldsymbol{\phi}^*|\boldsymbol{\phi})$ whose form ensures that the target is stationary. Since we are targeting the marginal parameter posterior, we

Algorithm 3 Marginal MH algortithm

1. Initialise the chain with $\boldsymbol{\phi}^{(0)}$. Set $j = 1$.
2. Propose $\boldsymbol{\phi}^* \sim q(\boldsymbol{\phi}^*|\boldsymbol{\phi}^{(j-1)})$.
3. Calculate the acceptance probability $\alpha(\boldsymbol{\phi}^*|\boldsymbol{\phi}^{(j-1)})$ of the proposed move, where
$$\begin{aligned}\alpha(\boldsymbol{\phi}^*|\boldsymbol{\phi}^{(j-1)}) &= \min\left\{1,\ A(\boldsymbol{\phi}^*|\boldsymbol{\phi}^{(j-1)})\right\} \\ &= \min\left\{1,\ \frac{\pi(\boldsymbol{\phi}^*|\boldsymbol{x}_{1:n})q(\boldsymbol{\phi}^{(j-1)}|\boldsymbol{\phi}^*)}{\pi(\boldsymbol{\phi}^{(j-1)}|\boldsymbol{x}_{1:n})q(\boldsymbol{\phi}^*|\boldsymbol{\phi}^{(j-1)})}\right\}.\end{aligned}$$
4. With probability $\alpha(\boldsymbol{\phi}^*|\boldsymbol{\phi}^{(j-1)})$, set $\boldsymbol{\phi}^{(j)} = \boldsymbol{\phi}^*$; otherwise set $\boldsymbol{\phi}^{(j)} = \boldsymbol{\phi}^{(j-1)}$.
5. Set $j := j + 1$. Return to step 2.

refer to the resulting MH algorithm as marginal Metropolis-Hastings; see Algorithm 3.

The MH algorithm defines a homogeneous Markov chain with transition density (assuming that the chain moves) given by

$$\pi(\boldsymbol{\phi}|\tilde{\boldsymbol{\phi}}, \boldsymbol{x}_{1:n}) = q(\boldsymbol{\phi}|\tilde{\boldsymbol{\phi}})\alpha(\boldsymbol{\phi}|\tilde{\boldsymbol{\phi}}).$$

It can then be easily checked that

$$\pi(\boldsymbol{\phi}|\tilde{\boldsymbol{\phi}}, \boldsymbol{x}_{1:n})\pi(\tilde{\boldsymbol{\phi}}|\boldsymbol{x}_{1:n}) = \pi(\tilde{\boldsymbol{\phi}}|\boldsymbol{\phi}, \boldsymbol{x}_{1:n})\pi(\boldsymbol{\phi}|\boldsymbol{x}_{1:n})$$

which is known as the detailed balance equation. Integrating both sides of this equation with respect to $\tilde{\boldsymbol{\phi}}$ gives equation (2.15) and so we see that the target $\pi(\boldsymbol{\phi}|\boldsymbol{x}_{1:n})$ is stationary. Formal convergence conditions can be found in Roberts and Smith (1994) and Tierney (1994).

Note that the acceptance probability simplifies to

$$A(\boldsymbol{\phi}^*|\boldsymbol{\phi}^{(j-1)}) = \frac{\pi(\boldsymbol{\phi}^*)\pi(\boldsymbol{x}_{1:n}|\boldsymbol{\phi}^*)q(\boldsymbol{\phi}^{(j-1)}|\boldsymbol{\phi}^*)}{\pi(\boldsymbol{\phi}^{(j-1)})\pi(\boldsymbol{x}_{1:n}|\boldsymbol{\phi}^{(j-1)})q(\boldsymbol{\phi}^*|\boldsymbol{\phi}^{(j-1)})}$$

so that only the unnormalised posterior density is required. Furthermore, assuming that the components of $\boldsymbol{\phi}$ are independent *a priori* gives

$$A(\boldsymbol{\phi}^*|\boldsymbol{\phi}^{(j-1)}) = \frac{\pi(\boldsymbol{x}_{1:n}|\boldsymbol{\phi}^*)q(\boldsymbol{\phi}^{(j-1)}|\boldsymbol{\phi}^*)\prod_{i=1}^{n_{par}}\pi(\phi_i^*)}{\pi(\boldsymbol{x}_{1:n}|\boldsymbol{\phi}^{(j-1)})q(\boldsymbol{\phi}^*|\boldsymbol{\phi}^{(j-1)})\prod_{i=1}^{n_{par}}\pi(\phi_i^{(j-1)})}.$$

The observed data likelihood $\pi(\boldsymbol{x}_{1:n}|\boldsymbol{\phi})$ can be factorised according to (2.17) so that the accep-

tance probability becomes

$$A(\boldsymbol{\phi}^*|\boldsymbol{\phi}^{(j-1)}) = \frac{q(\boldsymbol{\phi}^{(j-1)}|\boldsymbol{\phi}^*)}{q(\boldsymbol{\phi}^*|\boldsymbol{\phi}^{(j-1)})} \frac{\pi(x_1|\boldsymbol{\phi}^*)}{\pi(x_1|\boldsymbol{\phi}^{(j-1)})} \prod_{t=2}^{n} \frac{\pi(x_t|x_{1:t-1}, \boldsymbol{\phi}^*)}{\pi(x_t|x_{1:t-1}, \boldsymbol{\phi}^{(j-1)})} \prod_{i=1}^{n_{par}} \frac{\pi(\phi_i^*)}{\pi(\phi_i^{(j-1)})}.$$

The constituent terms in (2.17) can be calculated using step 2(b) of the forward filter in Algorithm 2. Some common choices of proposal distribution are now discussed.

Choices of proposal distribution

Symmetric chains. If the proposal distribution is symmetric, then

$$q(\boldsymbol{\phi}^{(j-1)}|\boldsymbol{\phi}^*) = q(\boldsymbol{\phi}^*|\boldsymbol{\phi}^{(j-1)}).$$

In this case, the acceptance probability simplifies as

$$A(\boldsymbol{\phi}^*|\boldsymbol{\phi}^{(j-1)}) = \frac{\pi(\boldsymbol{\phi}^*|x_{1:n})}{\pi(\boldsymbol{\phi}^{(j-1)}|x_{1:n})} \tag{2.19}$$

which only requires the (unnormalised) posterior distribution.

Random walk chains. A common choice of the proposal distribution is the random walk proposal, where

$$\boldsymbol{\phi}^* = \boldsymbol{\phi}^{(j-1)} + \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim N(0, \boldsymbol{\Sigma}).$$

Using different values of the innovation variance $\boldsymbol{\Sigma}$ will result in different mixing properties of the chain. If $\boldsymbol{\Sigma}$ is large, large jumps are proposed, but are likely to be rejected. Conversely, If $\boldsymbol{\Sigma}$ is small, small jumps are proposed that are likely to be accepted. This suggests an optimal tuning parameter that balances acceptance rate with exploration of the parameter space. Therefore, we usually multiply the innovation variance by a scaling parameter, say γ, to optimise performance. A standard rule of thumb due to Roberts et al. (1997) and Roberts and Rosenthal (2001) is

$$\boldsymbol{\Sigma} = \frac{2.38^2}{n_{par}} Var(\boldsymbol{\phi}|\boldsymbol{x}_{1:n})$$

which leads to an acceptance rate of 0.234 for certain target distirbutions, as $n_{par} \to \infty$. In practice, values of the acceptance rate between 0.1 and 0.4 usually give reasonable mixing, as measured by for example effective sample size; see Section 2.3.3 for further details. The

posterior variance $Var(\phi|\boldsymbol{x}_{1:n})$ can be estimated via a pilot run of the MH scheme.

Note that if the innovation distribution is symmetric about zero, then we have a symmetric random walk chain with the acceptance probability defined by (2.19).

In many applications, parameter values must be strictly positive necessitating the use of a log normal random walk proposal, that is

$$\log \phi^* = \log \phi^{(j-1)} + \tilde{\epsilon}, \qquad \tilde{\epsilon} \sim N(0, \tilde{\Sigma})$$

where $\tilde{\Sigma}$ is a scaled estimate of the posterior variance of $\log \phi$. The acceptance probability for the marginal MH algorithm can be written as

$$A(\phi^*|\phi^{(j-1)}) = \frac{\pi(x_1|\phi^*)}{\pi(x_1|\phi^{(j-1)})} \prod_{t=2}^{n} \frac{\pi(x_t|x_{1:t-1}, \phi^*)}{\pi(x_t|x_{1:t-1}, \phi^{(j-1)})} \prod_{i=1}^{n_{par}} \frac{\pi(\phi_i^*)}{\pi(\phi_i^{(j-1)})} \prod_{i=1}^{n_{par}} \frac{\phi_i^*}{\phi_i^{(j-1)}}.$$

Independence chains. As the name suggests, the proposed value is generated independently of the current position of the chain, so that $q(\phi^*|\phi^{(j-1)}) = f(\phi^*)$ for some density $f(\cdot)$. The acceptance probability becomes

$$A(\phi^*|\phi^{(j-1)}) = \frac{\pi(\phi^*|x_{1:n})}{\pi(\phi^{(j-1)}|x_{1:n})} \frac{f(\phi^{(j-1)})}{f(\phi^*)}.$$

When the prior is selected as the proposal distribution, the acceptance probability becomes

$$A(\phi^*|\phi^{(j-1)}) = \frac{\pi(x_{1:n}|\phi^*)}{\pi(x_{1:n}|\phi^{(j-1)})}$$

and hence only depends on the ratio of the likelihoods of the proposed value and the current value. Ideally, the acceptance probability should be as close to one as possible and so care must be taken when choosing the proposal density. For example, if the prior is used, significant difference in posterior and prior support can lead to an inefficient MH scheme.

2.3.3 MCMC diagnostics

The basic idea of MCMC methods is to construct a Markov chain whose equilibrium distribution is the target distribution. Therefore, investigating convergence of the Markov chain (or lack thereof) is essential when applying MCMC approaches. Informal visual checks are common

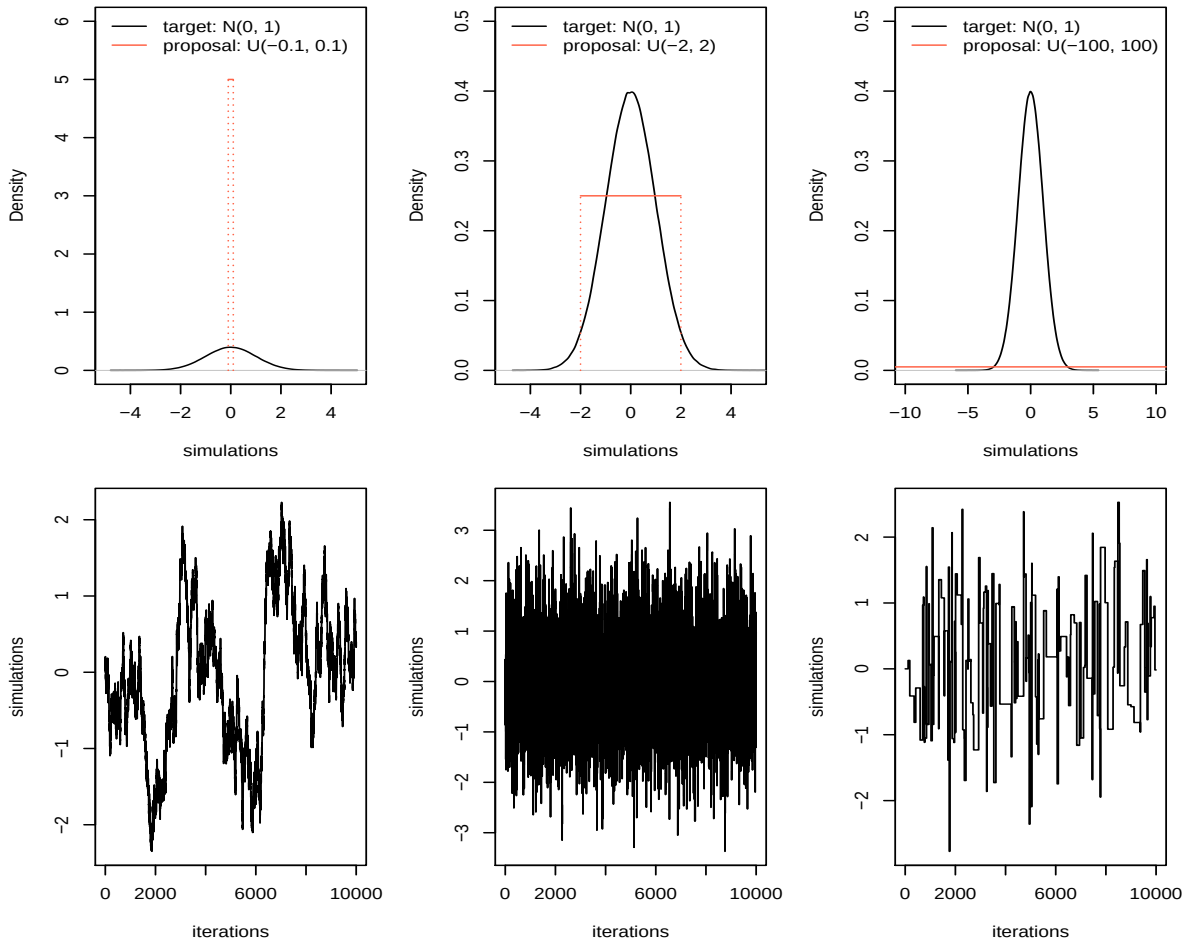

Figure 2.3: Trace plots of MCMC samples by using different uniform proposal distributions.

place, and usually involve inspection of trace plots, which monitor the trajectory of MCMC samples. Applying the MH algorithm by taking the standard normal distribution $N(0,1)$ as the target distribution and proposing the candidate samples through a random walk proposal with uniform innovations, Figure 2.3 shows the trace plots corresponding to different uniform innovations, i.e. sampled from $U(-0.1,0.1)$, $U(-2,2)$ and $U(-100,100)$ respectively, from left to right. When the random walk proposal has $U(-0.1,0.1)$, there is a high accepted rate and therefore the simulated samples are highly correlated. In the right column, the proposal has $U(-100,100)$ which is too wide compared with the target distribution and causes most proposed samples being rejected. The proposal distribution $U(-2,2)$ has the most comparable shape as the target distribution, hence the trace plot in the middle shows a good mixing.

As the initialisation of a chain is usually arbitrary and most likely not from a high density part of the stationary distribution, it is customary to remove a number of samples from the beginning of a chain. This number of iterations is the so called burn-in period (Gilks et al., 1996). In addition

to the visual examination of the trace plot, some formal methods on how to choose the length of burn-in period and assess convergence have been discussed by Heidelberger and Welch (1983), Geweke (1992), Gelman and Rubin (1992) and Raftery and Lewis (1996).

Successive MCMC draws are auto-correlated. If they exhibit a high degree of dependency, the chain will be slow to explore the parameter space. Moreover, posterior summaries based on such highly auto-correlated samples will generally have large variances. In order to get near independent MCMC samples, a simple method is to thin a chain by taking the value at every ith iterate. This procedure may also be important when long runs are required and memory storage is limited. However, to avoid information loss, we need to carefully select the frequency for thinning.

The autocorrelation function is used to quantify the dependency of samples at different lags. Assuming the chain is in equilibrium, let $\phi^{(j)}$ be the value at jth iterate. The autocorrelation between $\phi^{(j)}$ and $\phi^{(j+k)}$ is

$$\rho_k = \frac{E(\phi^{(j)}\phi^{(j+k)}) - E(\phi^{(j)})E(\phi^{(j+k)})}{Var(\phi^{(j)})}.$$

For independent samples, the autocorrelation ρ_k is zero. For MCMC output, the (sample) autocorrelation is 1 at $k = 0$ and gradually decreases as k increases. A plot of the autocorrelation function can be useful in deciding an appropriate level of thinning. The autocorrelation function can further be used to determine the effective sample size (ESS) of a set of MCMC samples. The ESS of a run of length N can be roughly interpreted as the equivalent number of independent samples. It is given by

$$ESS = \frac{N}{1 + 2\sum_{k=1}^{\infty} \rho_k}.$$

The estimation of the ESS requires estimating the spectral density at frequency zero and can be computed using the `R` package CODA (Plummer et al., 2006).

2.4 Posterior predictive checks

Posterior predictive checks are used to assess model validity by comparing suitably chosen predictive summaries to the observed data (Gelman and Hill, 2007). If the model is appropriate, there are not systematic discrepancies between the observed data and predictive summaries (Gelman et al., 2013). The predictive checks are usually conducted via calculation of the

within-sample predictive and out-of-sample forecast distributions.

2.4.1 Within-sample predictions

The within-sample predictive density is given by

$$\pi(\tilde{\boldsymbol{x}}_{1:n}|\boldsymbol{x}_{1:n}) = \int\int \pi(\tilde{\boldsymbol{x}}_{1:n}|\boldsymbol{\theta}_{1:n},\phi)\pi(\boldsymbol{\theta}_{1:n},\phi|\boldsymbol{x}_{1:n})d\boldsymbol{\theta}_{1:n}d\phi \tag{2.20}$$

where

$$\pi(\boldsymbol{\theta}_{1:n},\phi|\boldsymbol{x}_{1:n}) = \pi(\boldsymbol{\theta}_{1:n}|\phi,\boldsymbol{x}_{1:n})\pi(\phi|\boldsymbol{x}_{1:n}). \tag{2.21}$$

Although the within-sample predictive density in (2.20) is intractable, draws from $\pi(\boldsymbol{\theta}_{1:n},\phi|\boldsymbol{x}_{1:n})$ are readily available (via the Gibbs sampler, or marginal MH in conjunction with the backwards sampler) and therefore $\pi(\tilde{\boldsymbol{x}}_{1:n}|\boldsymbol{x}_{1:n})$ can be sampled via Monte Carlo. Given draws $(\phi^{(j)},\boldsymbol{\theta}_{1:n}^{(j)})$, $j = 1,\ldots,N$ from (2.21), we can simulate

$$\tilde{X}_t^{(j)}|\theta_t^{(j)},\phi^{(j)} \sim N(F_t\theta_t^{(j)}, V^{(j)}), \qquad j = 1,\ldots,N, \qquad t = 1,\ldots,n.$$

2.4.2 Out-of-sample forecasts

The system and observation forecast distributions can be obtained by exploiting the linear Gaussian structure of the DLM. The one-step ahead system forecast density is given by

$$\begin{aligned}\pi(\theta_{n+1}|\boldsymbol{x}_{1:n}) &= \int\int \pi(\theta_{n+1}|\theta_n,\phi,\boldsymbol{x}_{1:n})\pi(\theta_n|\phi,\boldsymbol{x}_{1:n})\pi(\phi|\boldsymbol{x}_{1:n})d\theta_n d\phi \\ &= \int \pi(\theta_{n+1}|\phi,\boldsymbol{x}_{1:n})\pi(\phi|\boldsymbol{x}_{1:n})d\phi\end{aligned}$$

where

$$\pi(\theta_{n+1}|\phi,\boldsymbol{x}_{1:n}) = N\left(\theta_{n+1}; G_{n+1}m_n, G_{n+1}C_nG_{n+1}^T + W\right).$$

Similarly, the one-step ahead observation forecast density is given by

$$\pi(x_{n+1}|\boldsymbol{x}_{1:n}) = \int \pi(x_{n+1}|\phi,\boldsymbol{x}_{1:n})\pi(\phi|\boldsymbol{x}_{1:n})d\phi$$

where

$$\pi(x_{n+1}|\boldsymbol{\phi}, \boldsymbol{x}_{1:n}) = N\left(x_{n+1}; F_{n+1}G_{n+1}m_n\,,\, F_{n+1}(G_{n+1}C_nG_{n+1}^T + W)F_{n+1}^T + V\right).$$

Hence, given N posterior summaries $(m_n^{(j)}, C_n^{(j)}), j = 1, \ldots, N$ from $\pi(\theta_n|\boldsymbol{\phi}, \boldsymbol{x}_{1:n})$ and $\boldsymbol{\phi}^{(j)}$ from $\pi(\boldsymbol{\phi}|\boldsymbol{x}_{1:n})$, the one-step ahead state and observation forecast distributions can be sampled via Monte Carlo, by drawing

$$\begin{aligned}
\theta_{n+1}^{(j)}|\boldsymbol{\phi}^{(j)}, \boldsymbol{x}_{1:n} &\sim N\left(G_{n+1}m_n^{(j)}\,,\, G_{n+1}C_n^{(j)}G_{n+1}^T + W^{(j)}\right), \\
X_{n+1}^{(j)}|\boldsymbol{\phi}^{(j)}, \boldsymbol{x}_{1:n} &\sim N\left(F_{n+1}G_{n+1}m_n^{(j)}\,,\, F_{n+1}(G_{n+1}C_n^{(j)}G_{n+1}^T + W^{(j)})F_{n+1}^T + V^{(j)}\right).
\end{aligned}$$

For the general k-step ahead forecast, the above draws are replaced by

$$\begin{aligned}
\theta_{n+k}^{(j)}|\boldsymbol{\phi}, \boldsymbol{x}_{1:n} &\sim N\left\{\left(\prod_{i=1}^{k} G_{n+i}\right)m_n^{(j)}\,,\, R_{n+k}^{(j)}\right\}, \\
x_{n+k}^{(j)}|\boldsymbol{\phi}^{(j)}, \boldsymbol{x}_{1:n} &\sim N\left\{F_{n+k}\left(\prod_{i=1}^{k} G_{n+i}\right)m_n^{(j)}\,,\, F_{n+k}R_{n+k}^{(j)}F_{n+k}^T + V^{(j)}\right\},
\end{aligned}$$

where

$$R_{n+k}^{(j)} = \left(\prod_{i=1}^{k} G_{n+i}\right)C_n^{(j)}\left(\prod_{i=1}^{k} G_{n+i}^T\right) + \sum_{j=0}^{k-2}\left(\left(\prod_{i=0}^{j} G_{n+k-i}\right)W^{(j)}\left(\prod_{i=0}^{j} G_{n+k-i}^T\right)\right) + W^{(j)}.$$

2.5 Simulation studies

In this section, we apply the Gibbs sampler and the marginal MH algorithm with a random walk proposal to synthetic data generated from the local level model and the sinusoidal form DLM considered in Sections 2.1 and 2.2.1.

2.5.1 Local level model

The simulated data stream consists of $n = 200$ observations with the error variances $V = 2$ and $W = 1$. Thus the local level model is

$$X_t = \theta_t + v_t, \qquad v_t \overset{indep}{\sim} N(0,2),$$
$$\theta_t = \theta_{t-1} + w_t, \qquad w_t \overset{indep}{\sim} N(0,1),$$

for $t = 1,\ldots,n$. The initial state is randomly drawn from $N(10,9)$. The data are shown in Figure 2.4 (left panel). We take $V,W \overset{indep}{\sim} IG(1,1)$ and $\theta_0 \sim N(10,16)$ *a priori* for both of the Gibbs sampler and the MH algorithm.

Results: Gibbs sampler

We consider the task of both state and parameter estimation. After discarding the first 100 iterations as burn-in, the Gibbs sampler was run for a further 10^5 iterations, with the samples further thinned by taking every 20th iterate. Gibbs output diagnostics (trace plots and autocorrelation functions) can be found in Figure 2.5 alongside kernel desnity estimates of the marginal posterior densities of V and W. The sampler mixes well with both traceplots suggesting convergence of the chain. Comparing the marginal posterior densities to the marginal prior densities shows that the data have been informative. Moreover, sampled values of V and W are consistent with the true values that produced the data.

Realisations from the within-sample state predictive distribution are shown in Figure 2.4 and, unsurprisingly given the relatively small value of the observation variance, are consistent with the data. The within-sample observation predictive distribution is summarised in Figure 2.6 by the mean and 95% credible interval at each observation time. It is clear that there is no systematic difference between the predictive mean and the observations. Moreover, all observations lie within the 95% credible interval. Figure 2.7 shows the comparison between k-step forecasts ($k = 1,2,3,4$) and the observations at each corresponding time points. The mean of the 1-step ahead forecast is very closed to the observed measurement at $t = 201$. As the number of forecast steps gets larger, forecast uncertainty increases. We can see less accuracy for the 2-step, 3-step and 4-step ahead forecasts, although the observed measurements at $t = 202$ and $t = 203$ still stay within the 95% credible intervals of the samples corresponding to 2-step and 3-step ahead forecasts.

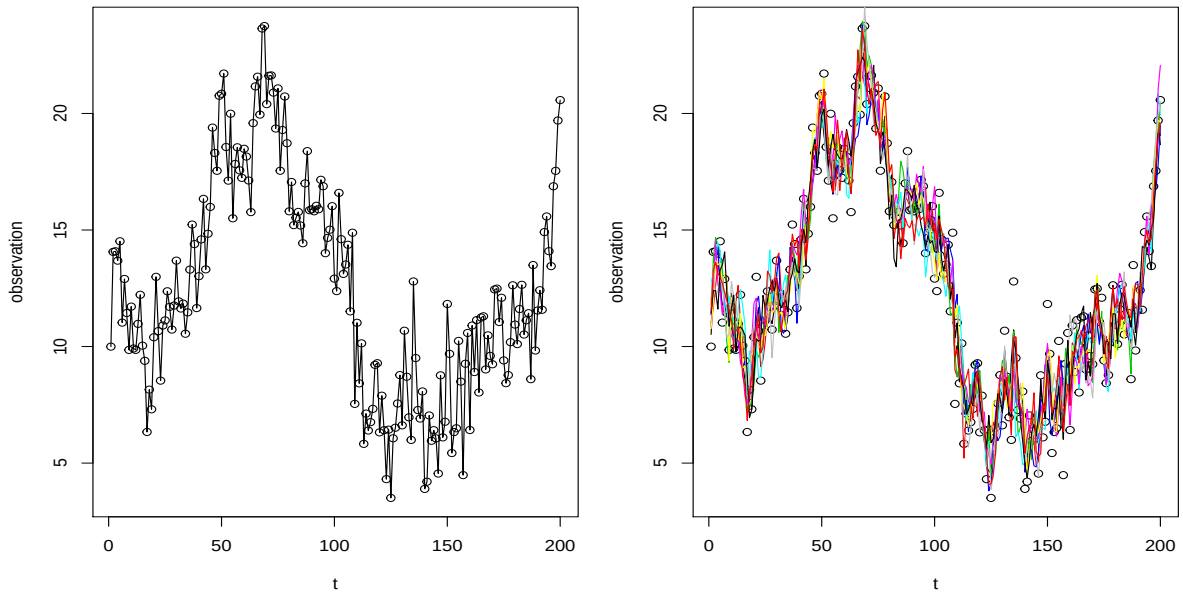

Figure 2.4: Left: simulated data; right: 10 marginal posterior realisations of $\boldsymbol{\theta}_{1:n}$.

Results: MH algorithm

We now apply the MH algorithm to the same simulated data set. We chose the random walk proposal to generate proposed samples. First we conducted a pilot run with 2000 iterations and a small innovation variance for the random walk proposal: $\Sigma = \text{diag}(0.01, 0.01)$. In the full run, we used the same number of iterations, burn-in and thinning factor as for the Gibbs sampler for comparison. The innovation variance was calculated as the product of the scaling $2.38^2/2$ and the variance of the samples from the pilot run. Figure 2.8 summarised the output of the MH scheme. Again, the trace plots and autocorrelation functions suggest good mixing of the parameter chains. The posterior distributions of the parameters are both consistent with the true values that produced the data. Of course, both the marginal MH algorithm and Gibbs sampler target the same parameter posterior and we can see from Figure 2.9 that consistent output from both schemes is obtained. Forecasts are easily generated using the output of the marginal MH scheme (not shown). Within sample predictive summaries can be generated by repeatedly running the backward sampler for each sampled value of (V, W) from the parameter posterior (not shown).

The Gibbs sampler requires a full forward and backward sweep per iteration whereas the marginal MH scheme only requires the forward sweep. Consequently, the computational cost for running the Gibbs sampler is around 4 minutes ands running the MH algorithm takes 1.5 minutes. Plainly, when interest lies in the marginal parameter posterior, the marginal MH scheme provides an

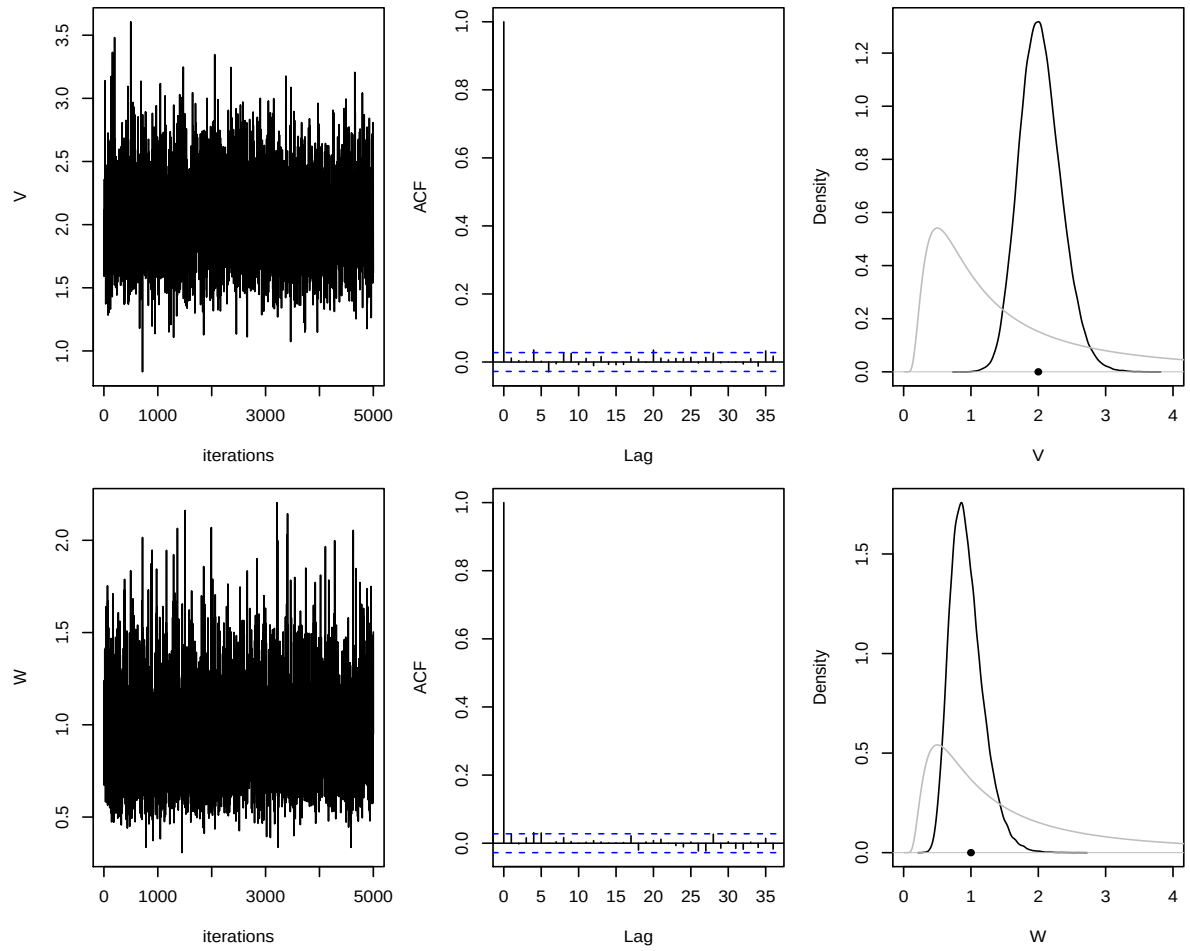

Figure 2.5: Gibbs sampler diagnostics: (1). trace plot by taking burn-in=100 and thinning=20; (2). autocorrelation function for the thinned chain after the burn-in period; (3). the posterior distribution (black) and the prior distribution (grey). The true parameter values are indicated by the solid circles.

efficient inference scheme. Moreover, draws from the marginal posterior of the dynamic states can be obtained post hoc, by appliying the backward sampler for parameter samples obtained from the (thinned) chain. Table 2.1 shows the numbers of ESS and ESS generated per second by the Gibbs sampler and the MH algorithm. Both schemes generated sufficient numbers of ESS, although the number of ESS for the parameter of system variance by the Gibbs sampler is relatively smaller than others. Because of less computational cost, the MH algorithm has a better performance than the Gibbs sampler in terms of ESS per second.

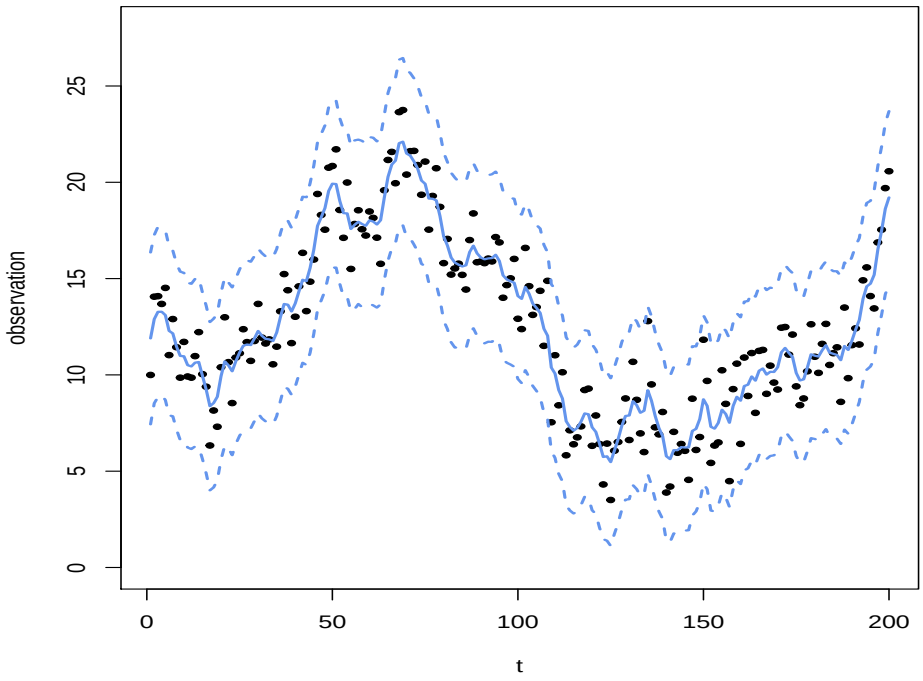

Figure 2.6: Mean and 95% credible interval of within-sample predictions against the data.

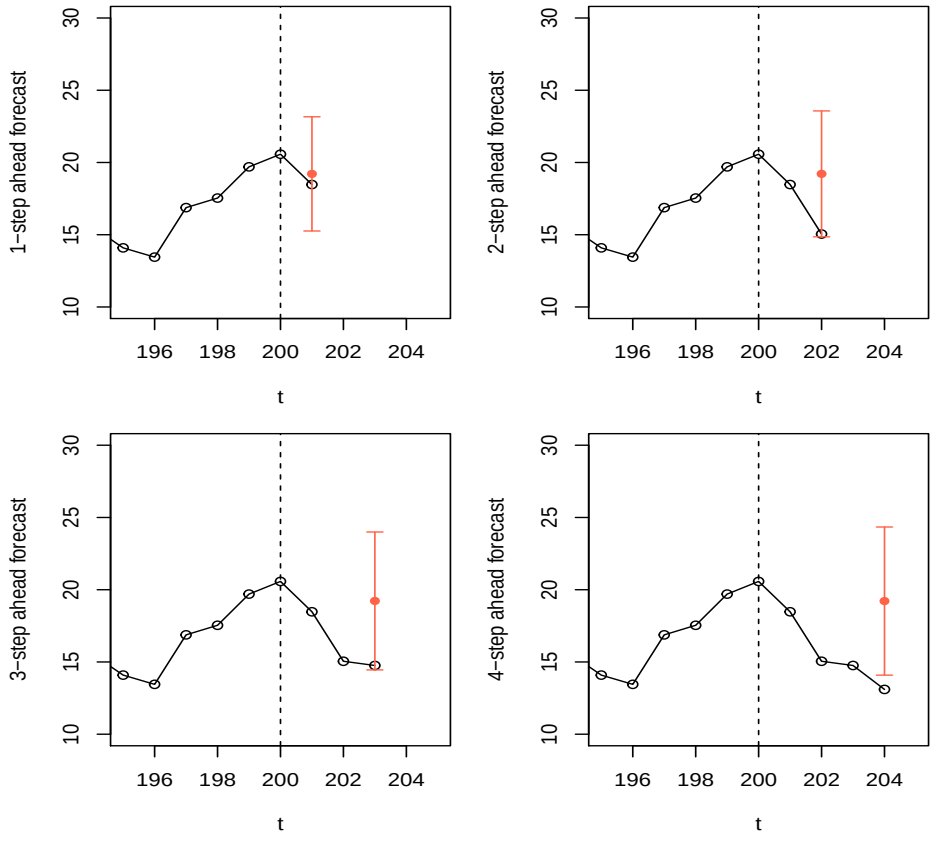

Figure 2.7: Simulated data (-o-) with mean and 95% credible interval of the samples for 1-step, 2-step, 3-step and 4-step ahead forecast respectively (error bars).

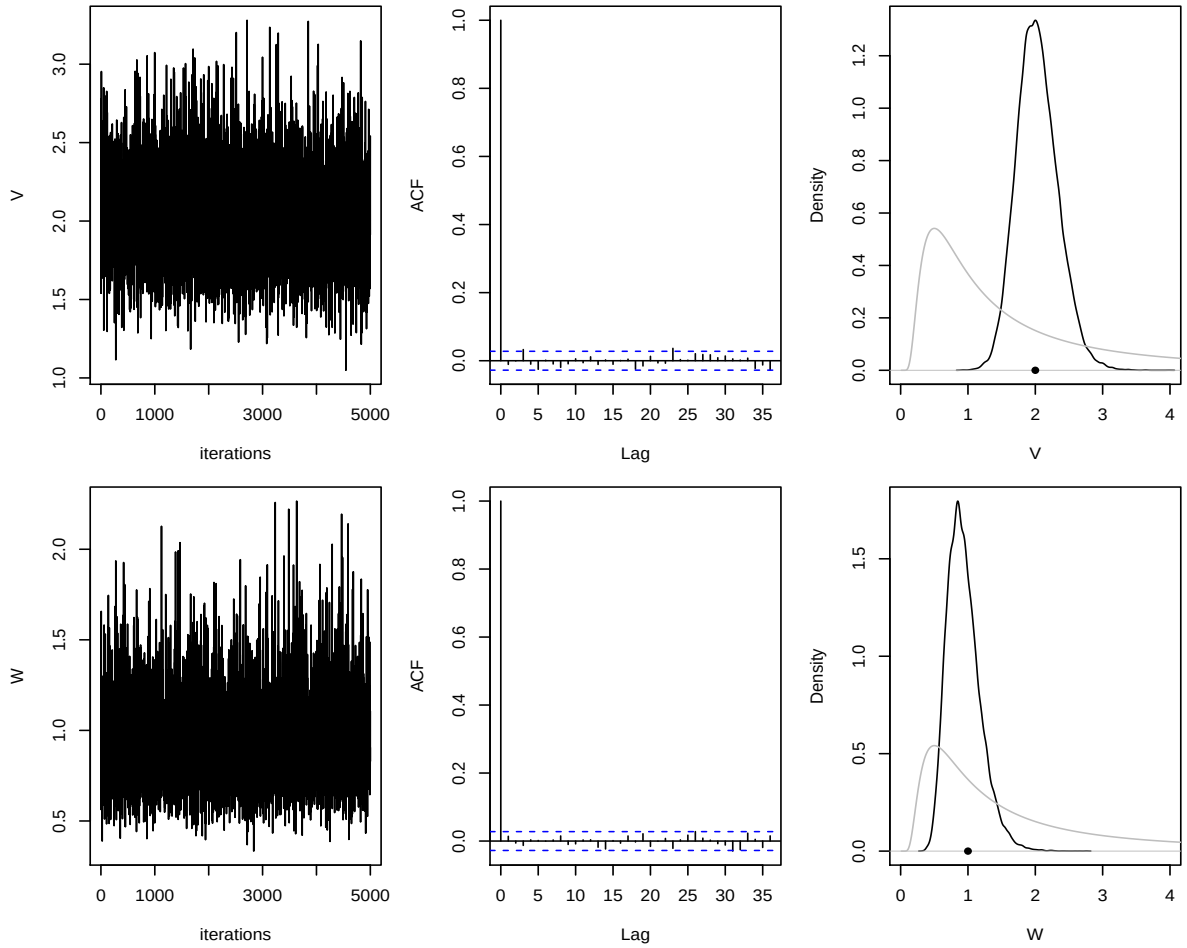

Figure 2.8: MH diagnostics: (1). trace plot by taking burn-in=100 and thining=20; (2). autocorrelation function for the thinned chain after the burn-in period; (3). the posterior distribution (black) and the prior distribution (grey). The true parameter values are indicated by the solid circles.

	Gibbs Sampler		MH Algorithm	
	V	W	V	W
ESS	16702	7642	13487	13517
ESS/sec	63	29	139	139

Table 2.1: ESS and ESS/sec for the parameters obtained by the Gibbs sampler and the MH algorithm.

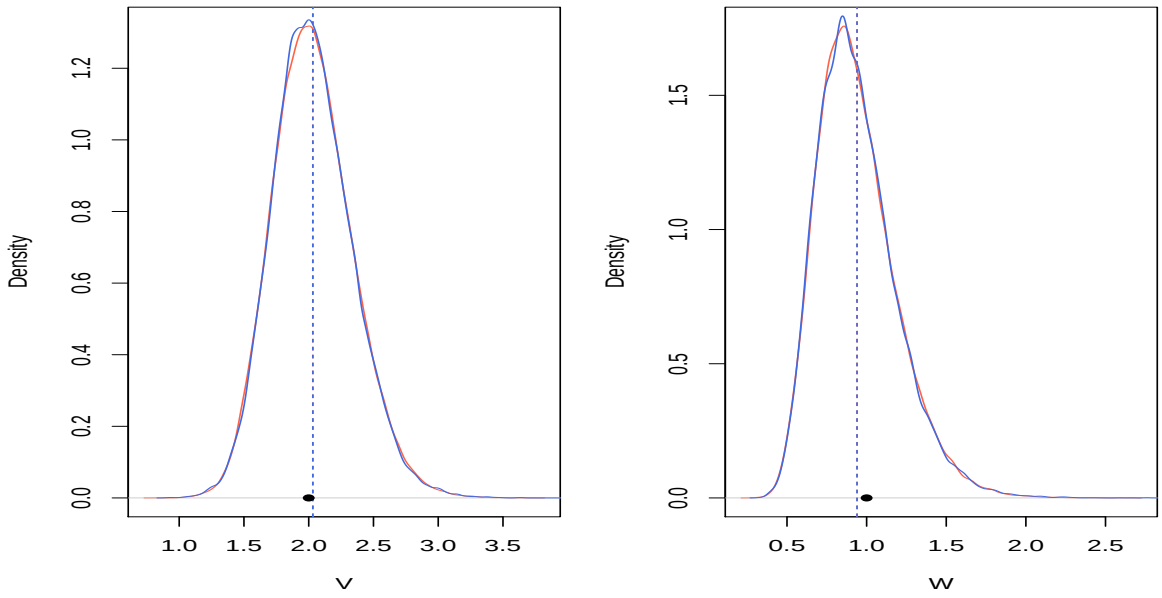

Figure 2.9: Comparison of the posterior distributions with the posterior means for V and W through the Gibbs sampler (red) and the MH algorithm (blue). The true parameter values are indicated by the solid circles.

2.5.2 Sinusoidal form DLM

In this example, we generated $n = 200$ synthetic observations from the sinusoidal form DLM with time units in hours. We took the initial state vector to be $\boldsymbol{\theta}_0 = (10, 0, 0)$ and the error variances $V = 2$ and $\boldsymbol{W} = \text{diag}(W_1, W_2, W_3) = 2\mathbb{I}_3$. The model then takes the form, for $t = 1, \ldots, n$

$$X_t = \boldsymbol{F}_t \boldsymbol{\theta}_t + v_t, \qquad v_t \overset{indep}{\sim} N(0,\, 2),$$
$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} + \boldsymbol{w}_t, \qquad \boldsymbol{w}_t \overset{indep}{\sim} N(\boldsymbol{0},\, 2\mathbb{I}_3).$$

where $\boldsymbol{F}_t = (\cos(\pi t/12), \sin(\pi t/12), 1)$. We take $V, W_1, W_2, W_3 \overset{indep}{\sim} IG(1,1)$ *a priori*. The data are shown in Figure 2.10 (left panel).

Results: Gibbs sampler

We ran the Gibbs sampler for 10^5 iterations, after discarding the first 100 values as burn-in. The output was then thinned by a factor of 20 to give 5000 values as the main monitoring run. Figure 2.11 summarises the output of the Gibbs sampler. All traceplots and autocorrelation functions suggest reasonable mixing. The marginal parameter posteriors are consistent with the

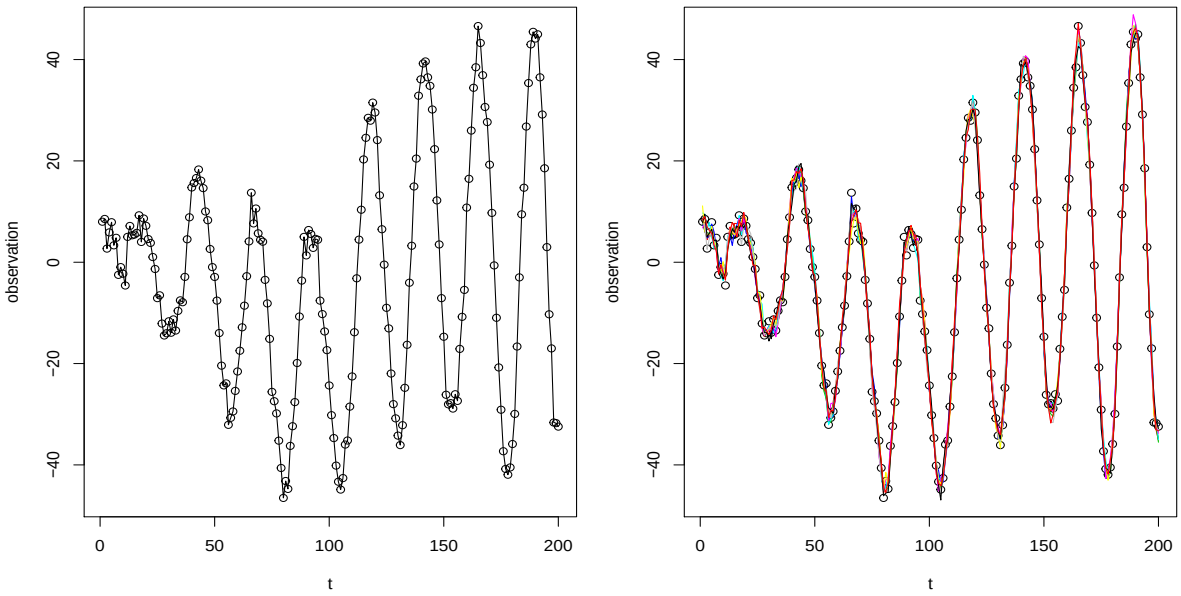

Figure 2.10: Left: simulated data; right: 10 marginal posterior realisations of $\boldsymbol{F}_{1:n}\boldsymbol{\theta}_{1:n}$.

true values that produced the data.

Realisations from the within-sample state predictive distribution are shown in Figure 2.4. Figure 2.10 shows 10 realisations of $\boldsymbol{F}_t\boldsymbol{\theta}_t$ (right panel) which are broadly consistent with the observations. In order to see any potential discrepancies between the within-sample observation predictions and the data more clearly, Figure 2.12 shows the mean and 95% credible interval of the difference between the samples generated from the within-sample predictive and the observation at each time point. Not surprisingly, this difference is plausibly zero at all time points.

Finally, Figure 2.13 shows k-step ahead forecasts ($k = 1,2,3,4$) against the corresponding observations. It is clear to that the forecast distributions are consistent with the observations, although the uncertainty increases as the number of forecast steps increases.

Results: MH algorithm

In the MH algorithm, we again chose the random walk proposal to generate proposed samples. A pilot run was conducted with 5000 iterations and a small innovation variance for the random walk proposal: $\Sigma = \text{diag}(0.01, 0.01, 0.01, 0.01)$. In the full run, we ran the marginal MH scheme with the same burn-in period, thinning frequency and total number of iterations as for the Gibbs sampler. We constructed the innovation variance by multiplying the scaling $2.38^2/4$ with the

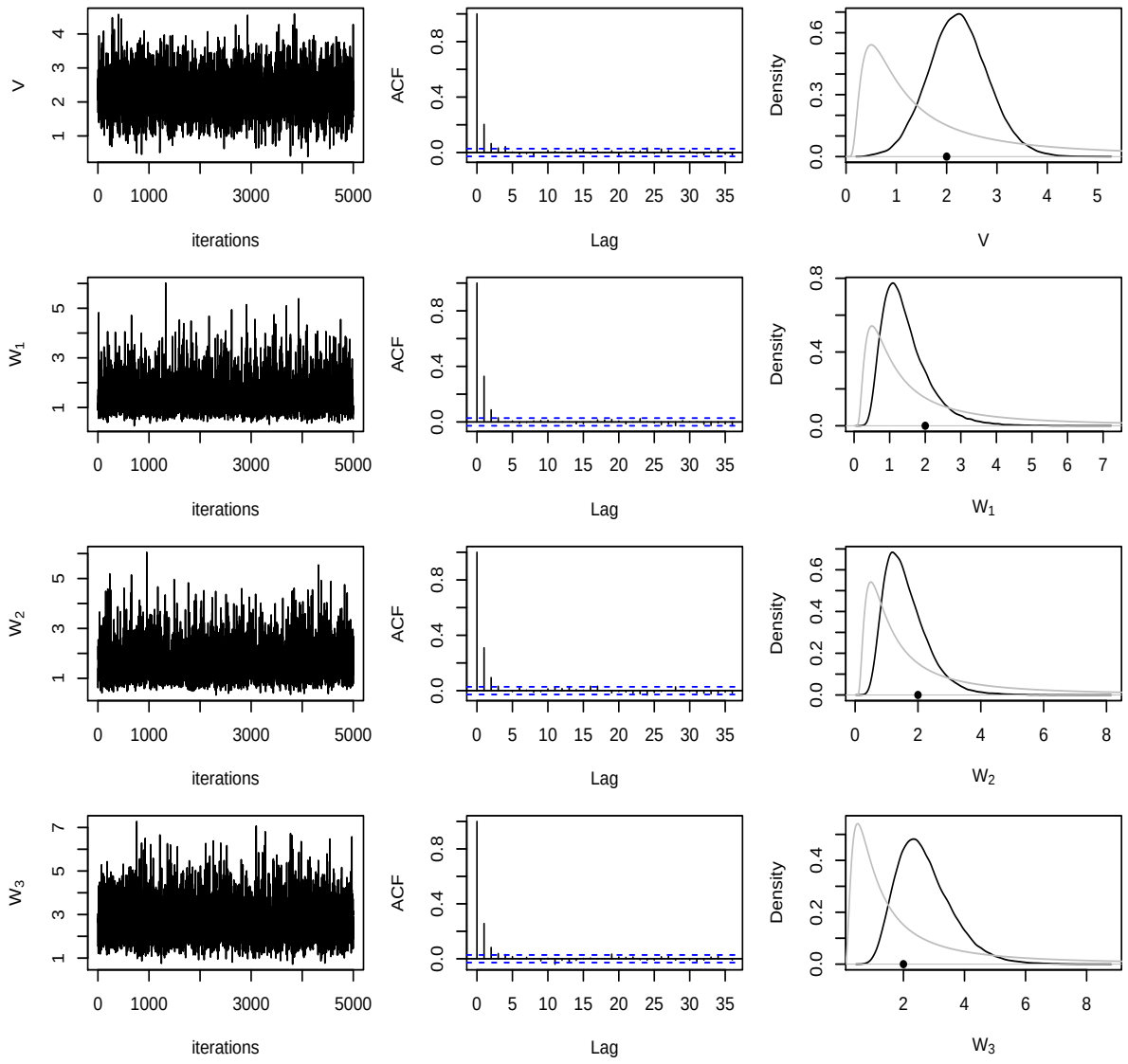

Figure 2.11: MCMC diagnostics (the Gibbs sampler): (1). trace plot by taking burn-in=100 and thining=20; (2). autocorrelation function for the thinned chain after the burn-in period; (3). the posterior distribution (black) and the prior distribution (grey). The true parameter values are indicated by the solid circles.

variance of the samples from the pilot run. Figure 2.14 summarises the MH output for each parameter chain. The traceplots and autocorrelation plots suggest convergence. The marginal posterior distributions are consistent with the true values and are also consistent with those obtained from the Gibbs sampler (see Figure 2.15). Similar results of within-sample predictions and out-of-sample forecasts are obtained for the MH scheme (not shown) as found in the previous section.

The run times for the Gibbs sampler and the MH algorithm are 70 minutes and 11 minutes

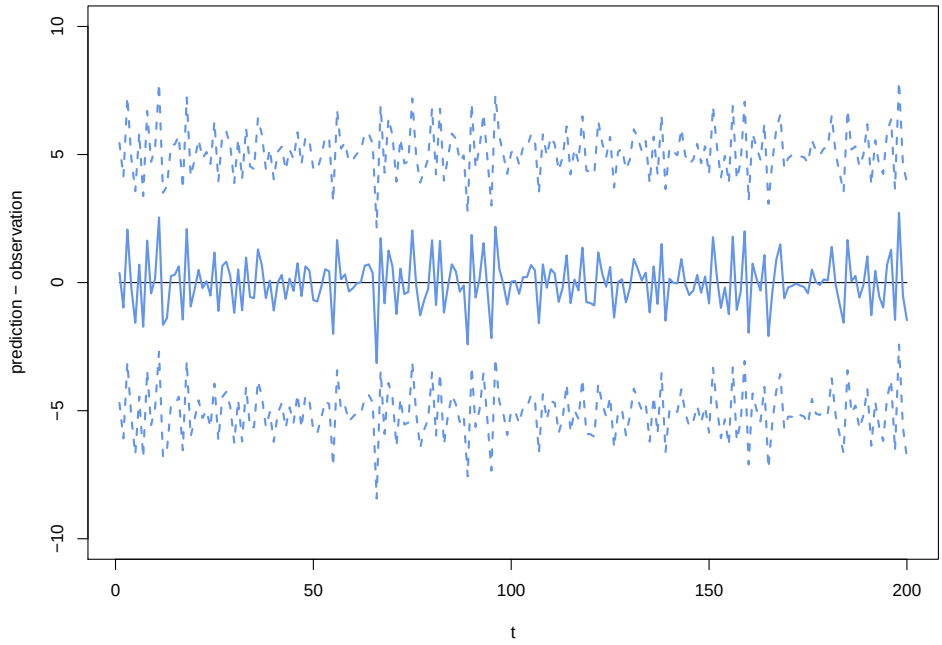

Figure 2.12: Mean and 95% credible interval of the differences between within-sample predictions and the data.

	Gibbs Sampler				MH Algorithm			
	V	W_1	W_2	W_3	V	W_1	W_2	W_3
ESS	4214	2668	3061	3218	7230	5479	8408	6590
ESS/sec	2	1	2	2	24	18	28	22

Table 2.2: ESS and ESS/sec for the parameters obtained by the Gibbs sampler and the MH algorithm

respectively. The relative computational cost (for Gibbs:MH) increases from 2.7:1 for the local level model to 6.4:1 for the sinusoidal DLM. This is expected, since for the sinusoidal DLM, the backward sweep requires matrix operations (as opposed to scalar operations for the local level model). Table 2.2 shows the numbers of ESS and ESS generated per second by the Gibbs sampler and the MH algorithm. Unsurprisingly the MH algorithm presents a much better performance than the Gibbs sampler in terms of efficiency.

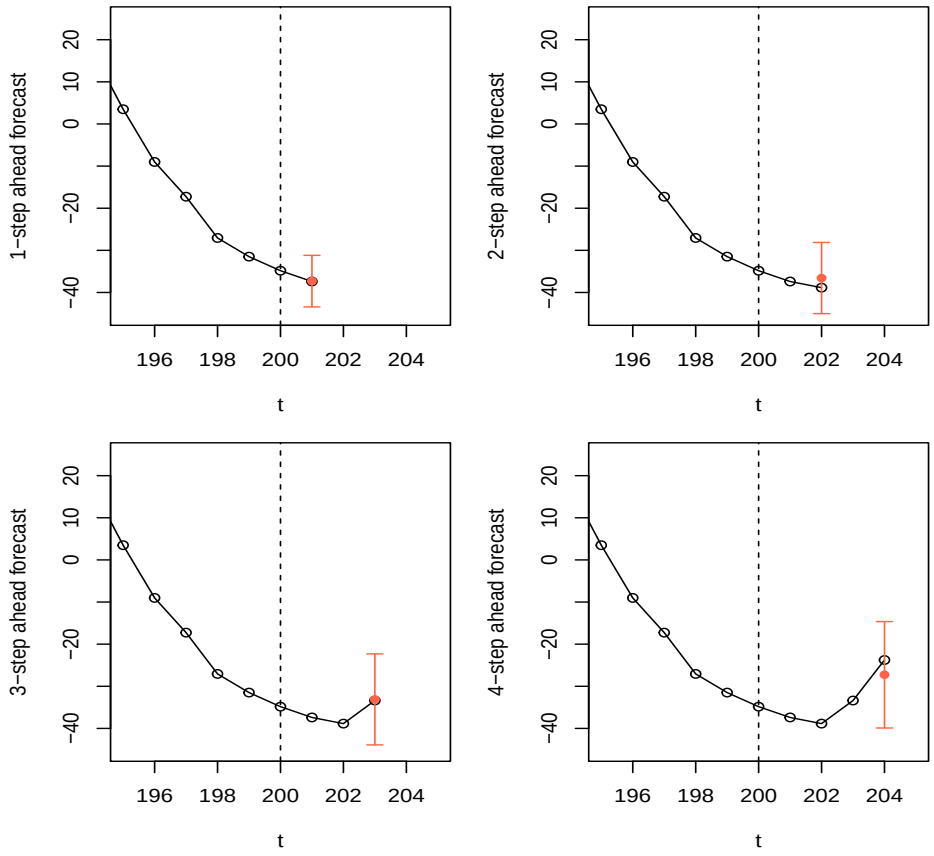

Figure 2.13: Simulated data (-o-) with mean and 95% credible interval of the samples for 1-step, 2-step, 3-step and 4-step ahead forecast respectively (error bars).

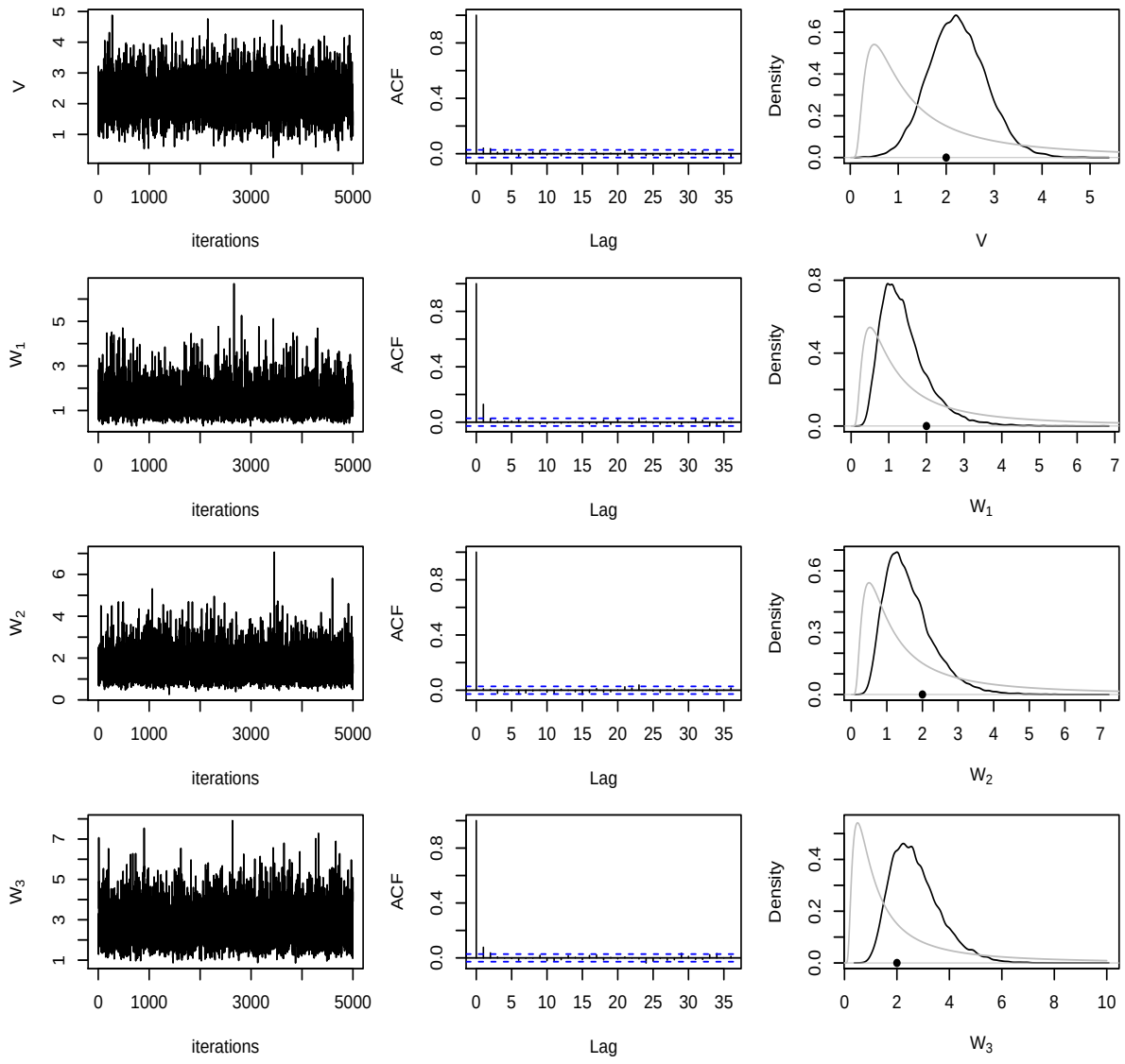

Figure 2.14: MCMC diagnostics (MH algorithm): 1. trace plot by thining=20; 2. autocorrelation function; 3. the posterior distribution (black) with the prior distribution (grey). The true parameter values are indicated by the solid circles.

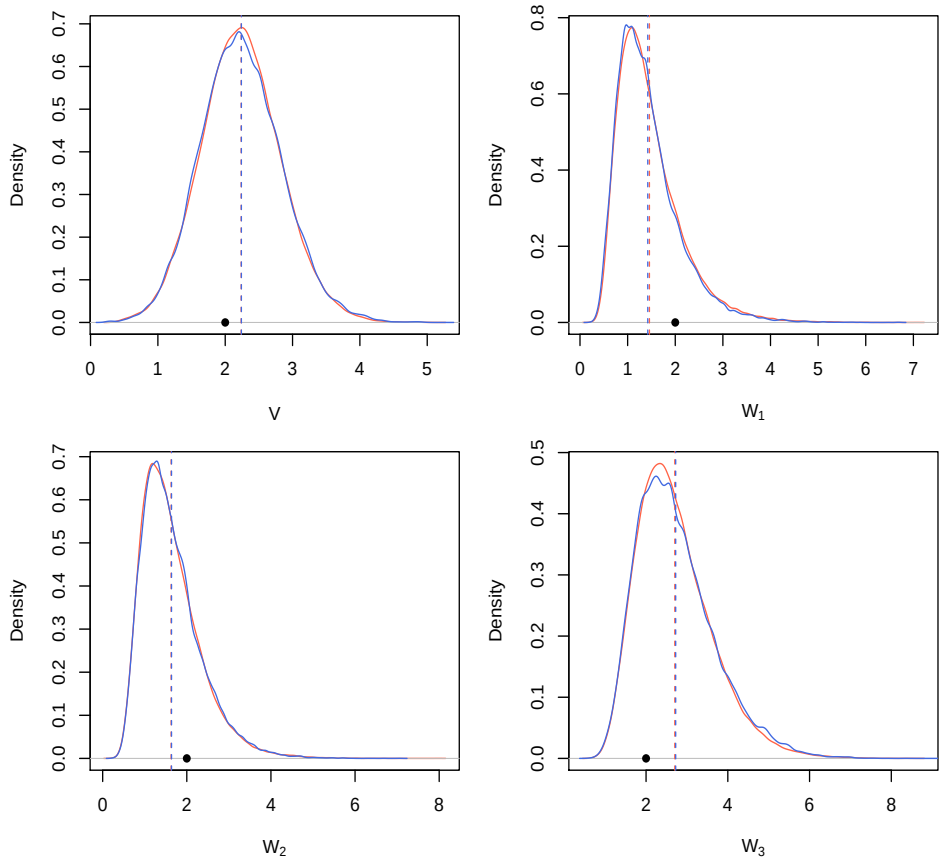

Figure 2.15: Comparison of the posterior distributions with the posterior means for V and W through the Gibbs sampler (red) and the MH algorithm (blue). The true parameter values are indicated by the solid circles.

Chapter 3

Sequential Monte Carlo

One of the bottlenecks of MCMC methods is computational efficiency, as they are of limited use for online inference since at any time a new observation becomes available, the MCMC scheme must be started from scratch to include the new observation. Sequential Monte Carlo (SMC) methods known as particle filters (Del Moral, 1996; Liu and Chen, 1998), are a set of online posterior density estimation algorithms that update the posterior density of the state space model parameters by applying Bayes' theorem sequentially; see Fearnhead and Künsch (2018) for a recent review of SMC methods. SMC methods are fundamentally based on importance (re)sampling methods which we now review.

3.1 Importance (re)sampling

3.1.1 Importance sampling

Consider a probability density $\pi(\boldsymbol{\theta})$, $\boldsymbol{\theta} \in D \subseteq \mathbb{R}^d$, and suppose we want to compute an expectation

$$\mu = \mathrm{E}_\pi(f(\boldsymbol{\theta})) = \int_D f(\boldsymbol{\theta})\pi(\boldsymbol{\theta})d\boldsymbol{\theta}.$$

If direct sampling from $\pi(\cdot)$ is straightforward then the Monte Carlo estimate of μ is

$$\hat{\mu} = \frac{1}{N}\sum_{j=1}^{N} f(\boldsymbol{\theta}^{(j)}),$$

where $\left\{\boldsymbol{\theta}^{(1)},\ldots,\boldsymbol{\theta}^{(N)}\right\}$ is a random sample from $\pi(\cdot)$. It is easily checked that $\hat{\mu}$ corresponds to an unbiased and consistent estimate of μ. However, direct sampling from $\pi(\boldsymbol{\theta})$ is not always straightforward. If $g(\boldsymbol{\theta})$ is another probability density function on $D \subseteq \mathbb{R}^d$ and it is easy to sample from, we may proceed by rewriting the expectation as

$$\begin{aligned}\mu = \mathrm{E}_\pi(f(\boldsymbol{\theta})) &= \int_D f(\boldsymbol{\theta})\frac{\pi(\boldsymbol{\theta})}{g(\boldsymbol{\theta})}g(\boldsymbol{\theta})d\boldsymbol{\theta} \\ &= \mathrm{E}_g\left(f(\boldsymbol{\theta})\frac{\pi(\boldsymbol{\theta})}{g(\boldsymbol{\theta})}\right).\end{aligned}$$

Suppose $\left\{\boldsymbol{\theta}^{(1)},\ldots,\boldsymbol{\theta}^{(N)}\right\}$ is a random sample drawn from $g(\boldsymbol{\theta})$. Then, the Monte Carlo estimate of μ is

$$\hat{\mu} = \frac{1}{N}\sum_{j=1}^{N} f(\boldsymbol{\theta}^{(j)})\frac{\pi(\boldsymbol{\theta}^{(j)})}{g(\boldsymbol{\theta}^{(j)})}. \tag{3.1}$$

Note that as before, $\hat{\mu}$ corresponds to an unbiased and consistent estimator of the true value $\mu = \mathrm{E}_\pi(f(\boldsymbol{\theta}))$. This method is known as importance sampling. The ratio which describes the difference between two probability densities $\pi(\boldsymbol{\theta})$ and $g(\boldsymbol{\theta})$ is the so called importance weight

$$\tilde{\omega}^{(k)} = \frac{\pi(\boldsymbol{\theta}^{(k)})}{g(\boldsymbol{\theta}^{(k)})}, \qquad k = 1,\ldots,N,$$

and $g(\boldsymbol{\theta})$ is called the importance density.

We can derive the variance of $\hat{\mu}$ as $\mathrm{Var}(\hat{\mu}) = \sigma_g^2/N$, where

$$\begin{aligned}\sigma_g^2 &= \int_D \left[\frac{f(\boldsymbol{\theta})\pi(\boldsymbol{\theta})}{g(\boldsymbol{\theta})}\right]^2 g(\boldsymbol{\theta})d\boldsymbol{\theta} - \mu^2 \\ &= \int_D \frac{[f(\boldsymbol{\theta})\pi(\boldsymbol{\theta}) - \mu g(\boldsymbol{\theta})]^2}{g(\boldsymbol{\theta})}d\boldsymbol{\theta} \\ &= \mathrm{E}_g\left(\frac{[f(\boldsymbol{\theta})\pi(\boldsymbol{\theta}) - \mu g(\boldsymbol{\theta})]^2}{g(\boldsymbol{\theta})^2}\right).\end{aligned}$$

Therefore, a good importance density $g(\boldsymbol{\theta})$ can be selected when $f(\boldsymbol{\theta})\pi(\boldsymbol{\theta}) - \mu g(\boldsymbol{\theta})$ approaches to zero, i.e. the variance σ_g^2 will be close to zero.

By the strong law of large numbers, we have

$$\int \frac{\pi(\boldsymbol{\theta})}{g(\boldsymbol{\theta})}g(\boldsymbol{\theta})d\boldsymbol{\theta} \simeq \frac{1}{N}\sum_{j=1}^{N}\frac{\pi(\boldsymbol{\theta}^{(j)})}{g(\boldsymbol{\theta}^{(j)})} \to 1, \quad \text{as } N \to \infty. \tag{3.2}$$

Therefore, taking the form of (3.1) divided by (3.2), we obtain the estimate

$$\begin{aligned}\tilde{\mu} &= \frac{\frac{1}{N}\sum_{j=1}^{N} f(\boldsymbol{\theta}^{(j)})\left[\pi(\boldsymbol{\theta}^{(j)})/g(\boldsymbol{\theta}^{(j)})\right]}{\frac{1}{N}\sum_{j=1}^{N}\left[\pi(\boldsymbol{\theta}^{(j)})/g(\boldsymbol{\theta}^{(j)})\right]} \\ &= \sum_{j=1}^{N} f(\boldsymbol{\theta}^{(j)})\omega^{(j)}\end{aligned}$$

where

$$\omega^{(k)} = \frac{\pi(\boldsymbol{\theta}^{(k)})/g(\boldsymbol{\theta}^{(k)})}{\sum_{k=1}^{N}\pi(\boldsymbol{\theta}^{(k)})/g(\boldsymbol{\theta}^{(k)})}, \qquad k = 1,\ldots,N,$$

are the normalised importance weights. This is called the self-normalised importance sampling estimate, which corresponds to an asymptotically unbiased and consistent estimator of μ. Note that the self-normalised importance sampling estimate only requires the availability of the target up to a multiplicative constant, since replacing $\pi(\boldsymbol{\theta}^{(k)})$ with $c\pi(\boldsymbol{\theta}^{(k)})$ for some constant $c > 0$ leaves the weight unchanged.

According to (3.2), we have

$$\Pr\left(\lim_{N\to\infty}\tilde{\mu} = \mu\right) = 1.$$

By using the delta method (Doob, 1935), the approximate variance of $\tilde{\mu}$ can be written as $\widetilde{\mathrm{Var}}(\tilde{\mu}) = \sigma^2_{g,sn}/N$, where

$$\begin{aligned}\sigma^2_{g,sn} &= \frac{\mathrm{E}_g\left(\left[f(\boldsymbol{\theta})\tilde{\omega} - \mu\tilde{\omega}\right]^2\right)}{\mathrm{E}_g(\tilde{\omega})^2} \\ &= \mathrm{E}_g\left(\tilde{\omega}^2\left[f(\boldsymbol{\theta}) - \mu\right]^2\right).\end{aligned}$$

A sketch proof can be found in (Owen, 2013).

3.1.2 Importance resampling

The importance samples $\left\{\boldsymbol{\theta}^{(1)},\ldots,\boldsymbol{\theta}^{(N)}\right\}$ and associated weights $\left\{\omega^{(1)},\ldots,\omega^{(N)}\right\}$ can be used to give a discrete approximation to $\pi(\boldsymbol{\theta})$. This approximation is given by

$$\hat{\pi}(\boldsymbol{\theta}) = \sum_{j=1}^{N} \omega^{(j)} \delta_{\boldsymbol{\theta}^{(j)}}$$

where $\delta_{\boldsymbol{\theta}^{(j)}}$ takes the value 1 if $\boldsymbol{\theta} = \boldsymbol{\theta}^{(j)}$ and 0 otherwise.

To obtain an equally weighted sample of size N, approximately distributed according to $\pi(\boldsymbol{\theta})$, we sample with replacement amongst $\left\{\boldsymbol{\theta}^{(1)},\ldots,\boldsymbol{\theta}^{(N)}\right\}$ using the normalised weights as probabilities. That is, $\Pr(\boldsymbol{\theta} = \boldsymbol{\theta}^{(j)}) = \omega^{(j)}$.

Note that as $N \to \infty$, $\hat{\pi}(\boldsymbol{\theta})$ approximates $\pi(\boldsymbol{\theta})$ increasingly well. To see this, consider the distribution function $\tilde{F}(\cdot)$ of a univariate θ generated by the algorithm. We have that

$$\begin{aligned}
\tilde{F}(a) &= \sum_{\{k:\, \theta^{(k)} \le a\}} \omega^{(k)} \\
&= \frac{\sum_{k=1}^{N} \tilde{\omega}^{(k)} \mathbb{I}(\theta^{(k)} \le a)}{\sum_{j=1}^{N} \tilde{\omega}^{(j)}}
\end{aligned}$$

where $\mathbb{I}(\theta^{(k)} \le a)$ takes the value 1 if $\theta^{(k)} \le a$ and 0 otherwise. Now

$$\tilde{F}(a) = \frac{\frac{1}{N}\sum_{k=1}^{N} \left[\pi(\theta^{(k)})/g(\theta^{(k)})\right] \mathbb{I}(\theta^{(k)} \le a)}{\frac{1}{N}\sum_{j=1}^{N} \left[\pi(\theta^{(j)})/g(\theta^{(j)})\right]}.$$

Thus, as $N \to \infty$,

$$\begin{aligned}
\tilde{F}(a) &\to \frac{\mathrm{E}_g\left(\left[\pi(\theta)/g(\theta)\right] \mathbb{I}(\theta \le a)\right)}{\mathrm{E}_g\left(\pi(\theta)/g(\theta)\right)} \\
&= \frac{\int_D \left[\pi(\theta)/g(\theta)\right] \mathbb{I}(\theta \le a) g(\theta) d\theta}{\int_D \pi(\theta) d\theta} \\
&= \Pr(\theta \le a).
\end{aligned}$$

Finally, we note that weighted resampling only requires the target to be available up to a multiplicative constant.

3.2 State filtering

Consider now the general form of the DLM in Section 2.1. Here, we will introduce two classic SMC methods which are used for tracking the state evolution sequentially when a new observation becomes available based on the estimation of the target posterior distribution. These methods are implemented under the assumption that all the static parameters ϕ are known and fixed. Consequently, we drop ϕ from the notation where possible.

3.2.1 Sequential importance sampling (SIS)

Consider the target posterior density $\pi(\boldsymbol{\theta}_{0:t}|\boldsymbol{x}_{1:t})$ at time t, which is not available in closed form. By Bayes' theorem, the posterior distribution can be factorised as

$$\pi(\boldsymbol{\theta}_{0:t}|\boldsymbol{x}_{1:t}) \propto \underbrace{\pi(\boldsymbol{x}_t|\boldsymbol{\theta}_t)}_{\text{likelihood}} \underbrace{\pi(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1})\pi(\boldsymbol{\theta}_{0:t-1}|\boldsymbol{x}_{1:t-1})}_{\text{prior}}. \tag{3.3}$$

Note that $\pi(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1})$ is the transition density of the state deduced from the system equation and $\pi(\boldsymbol{x}_t|\boldsymbol{\theta}_t)$ is the density of the observed measurement deduced from the observation equation.

Suppose that $\{\boldsymbol{\theta}^{(k)}_{0:t-1}, \omega^{(k)}_{t-1}\}^N_{k=1}$ is a weighted sample from $\pi(\boldsymbol{\theta}_{0:t-1}|\boldsymbol{x}_{1:t-1})$. Then $\pi(\boldsymbol{\theta}_{0:t}|\boldsymbol{x}_{1:t})$ can be approximated by

$$\hat{\pi}(\boldsymbol{\theta}_{0:t}|\boldsymbol{x}_{1:t}) \propto \pi(\boldsymbol{x}_t|\boldsymbol{\theta}_t)\pi(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1})\hat{\pi}(\boldsymbol{\theta}_{0:t-1}|\boldsymbol{x}_{1:t-1}), \tag{3.4}$$

where $\hat{\pi}(\boldsymbol{\theta}_{0:t-1}|\boldsymbol{x}_{1:t-1}) = \sum_{k=1}^N \omega^{(k)}_{t-1}\delta_{\boldsymbol{\theta}^{(k)}_{0:t-1}}$, and recall that $\delta_{\boldsymbol{\theta}^{(k)}_{0:t-1}}$ is a point mass on $\boldsymbol{\theta}_{0:t-1} = \boldsymbol{\theta}^{(k)}_{0:t-1}$.

Suppose we take

$$g(\boldsymbol{\theta}_{0:t}|\boldsymbol{x}_{1:t}) = g(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}, \boldsymbol{x}_t)g(\boldsymbol{\theta}_{0:t-1}|\boldsymbol{x}_{1:t-1})$$

as the importance density and let $\{\boldsymbol{\theta}^{(1)}_{0:t}, \ldots, \boldsymbol{\theta}^{(N)}_{0:t}\}$ be a sample drawn randomly from the impor-

Algorithm 4 SIS scheme

1. Initialisation. For $k = 1, \ldots, N$, sample $\boldsymbol{\theta}_0^{(k)} \sim \pi(\boldsymbol{\theta}_0)$ and set $\tilde{\omega}_0^{(k)} = 1$
2. For $t = 1, \ldots, n$:
   (a) Propagation. Sample $\boldsymbol{\theta}_t^{(k)} \sim g(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{t-1}^{(k)}, \boldsymbol{x}_t)$.
   (b) Update and normalize the importance weights via

$$\tilde{\omega}_t^{(k)} = \tilde{\omega}_{t-1}^{(k)} \frac{\pi(\boldsymbol{\theta}_t^{(k)} | \boldsymbol{\theta}_{t-1}^{(k)}) \pi(\boldsymbol{x}_t | \boldsymbol{\theta}_t^{(k)})}{g(\boldsymbol{\theta}_t^{(k)} | \boldsymbol{\theta}_{t-1}^{(k)}, \boldsymbol{x}_t)}, \qquad \omega_t^{(k)} = \frac{\tilde{\omega}_t^{(k)}}{\sum_{j=1}^{N} \tilde{\omega}_t^{(j)}}.$$

tance density at time t, then the importance weight $\tilde{\omega}_t^{(k)}$ can be written as

$$\begin{aligned}
\tilde{\omega}_t^{(k)} &= \frac{\pi(\boldsymbol{\theta}_{0:t}^{(k)} | x_{1:t})}{g(\boldsymbol{\theta}_{0:t}^{(k)} | \boldsymbol{x}_{1:t})} \\
&\propto \frac{\pi(\boldsymbol{\theta}_t^{(k)} | \boldsymbol{\theta}_{t-1}^{(k)}, \boldsymbol{x}_t) \hat{\pi}(\boldsymbol{\theta}_{0:t-1}^{(k)} | \boldsymbol{x}_{1:t-1})}{g(\boldsymbol{\theta}_t^{(k)} | \boldsymbol{\theta}_{t-1}^{(k)}, \boldsymbol{x}_t) g(\boldsymbol{\theta}_{0:t-1}^{(k)} | \boldsymbol{x}_{1:t-1})} \\
&\propto \tilde{\omega}_{t-1}^{(k)} \frac{\pi(\boldsymbol{\theta}_t^{(k)} | \boldsymbol{\theta}_{t-1}^{(k)}) \pi(\boldsymbol{x}_t | \boldsymbol{\theta}_t^{(k)})}{g(\boldsymbol{\theta}_t^{(k)} | \boldsymbol{\theta}_{t-1}^{(k)}, \boldsymbol{x}_t)}.
\end{aligned}$$

Hence the particles can be propagated through $g(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{t-1}, \boldsymbol{x}_t)$. This method is called sequential importance sampling and was initially introduced by Kong et al. (1994) (see Algorithm 4).

As a special case, suppose we take the prior as the importance density, that is, $g(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{t-1}, \boldsymbol{x}_t) = \pi(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{t-1})$. The importance weights become

$$\tilde{\omega}_t^{(k)} = \tilde{\omega}_{t-1}^{(k)} \pi(\boldsymbol{x}_t | \boldsymbol{\theta}_t^{(k)}), \qquad k = 1, \ldots, N.$$

Although SIS is easy to implement, as time t increases, only a small number of particles will have significant weights to dominate the posterior density. We call this problem degeneracy. The effective sample size is introduced as a useful criterion to monitor this problem, and is defined as

$$N_{ESS} = \frac{1}{\sum_{k=1}^{N} (\omega_t^{(k)})^2}$$

with $1 \leq N_{ESS} \leq N$. Here, as an example, we consider synthetic data simulated from the local

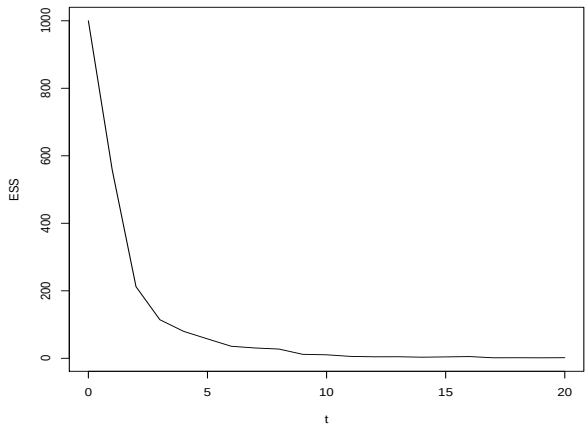

Figure 3.1: Effective sample size.

level model given by equations (2.1) and (2.2). The data were simulated with the error variances $V = 2$ and $W = 1$. The initial state was randomly drawn from $N(10,9)$. The data set consists of 20 observations and we take 1000 particles for running the SIS scheme. In Figure 3.1, the effective sample size starts from 1000 particles at time point 0. As time increases, the number of distinct particles shrinks quickly. After time point 10, we see that the target posterior collapses to a point mass, where only one particle remains to dominate the density and all other particles have negligible weights.

3.2.2 Bootstrap particle filter (BPF)

Using a resampling step inside SIS is a simple and efficient way to mitigate the problem of particle degeneracy by removing the less informative particles at the end of each time point before the next propagation. This method is called sequential importance sampling with resampling (SIR) (Smith and Gelfand, 1992). It is also referred to as a bootstrap particle filter (BPF) if $g(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}, \boldsymbol{x}_t) = \pi(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1})$ (Gordon et al., 1993). The BPF scheme is given by Algorithm 5. Note that we take the multinomial resampling approach by default for the illustrations of the SMC schemes in this Chapter. We write the multinomial distribution with integer outcomes $1,\ldots,N$ based on the normalised weights as the associated probabilities as $\mathcal{M}(\boldsymbol{\omega}^{1:N})$.

Notice that in the particle propagation process, the latest observation is not taken into account. Therefore, unless the variance of the observation process is large, few state particles will have reasonable weights, resulting in particle degeneracy. An improved method is to make particle propagation consistent with the arrival of new information, that is, sample the new particles

Algorithm 5 BPF scheme

1. Initialisation. For $k = 1, \ldots, N$, sample $\boldsymbol{\theta}_0^{(k)} \sim \pi(\boldsymbol{\theta}_0)$ and set $\tilde{\omega}_0^{(k)} = 1$.
2. For $t = 1, \ldots, n$:
   (a) Propagation. Sample $\boldsymbol{\theta}_t^{(k)} \sim \pi(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{t-1}^{(k)})$.
   (b) Update and normalize the importance weights via

$$\tilde{\omega}_t^{(k)} = \tilde{\omega}_{t-1}^{(k)} \pi(\boldsymbol{x}_t | \boldsymbol{\theta}_t^{(k)}), \qquad \omega_t^{(k)} = \frac{\tilde{\omega}_t^{(k)}}{\sum_{j=1}^{N} \tilde{\omega}_t^{(j)}}.$$

   (c) Sample indices $a_k \sim \mathcal{M}(\boldsymbol{\omega}^{1:N})$ and set $\{\boldsymbol{\theta}_t^{(k)}, \tilde{\omega}_t^{(k)}\} := \{\boldsymbol{\theta}_t^{(a_k)}, 1\}$.

conditional on both $\boldsymbol{\theta}_{t-1}$ and $\boldsymbol{x}_t$. Recall that the importance weights can be written as

$$\tilde{\omega}_t^{(k)} \propto \tilde{\omega}_{t-1}^{(k)} \frac{\pi(\boldsymbol{\theta}_t^{(k)} | \boldsymbol{\theta}_{t-1}^{(k)}, \boldsymbol{x}_t) \pi(\boldsymbol{x}_t | \boldsymbol{\theta}_{t-1}^{(k)})}{g(\boldsymbol{\theta}_t^{(k)} | \boldsymbol{\theta}_{t-1}^{(k)}, \boldsymbol{x}_t)}.$$

Taking the optimal importance density $g(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{t-1}, \boldsymbol{x}_t) = \pi(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{t-1}, \boldsymbol{x}_t)$, first introduced by Zaritskii et al. (1976), the new particles will be generated from the conditional distribution including the information of the observation $\boldsymbol{x}_t$, thus avoiding 'blind' forward simulation as used by the BPF scheme. We call this adjusted method the optimal BPF scheme. Under the structure of the DLM, we have that

$$\begin{pmatrix} \boldsymbol{\theta}_t \\ \boldsymbol{x}_t \end{pmatrix} \Bigg| \boldsymbol{\theta}_{t-1} \sim N \left\{ \begin{pmatrix} \boldsymbol{G}_t \boldsymbol{\theta}_{t-1} \\ \boldsymbol{F}_t \boldsymbol{G}_t \boldsymbol{\theta}_{t-1} \end{pmatrix}, \begin{pmatrix} \boldsymbol{W} & \boldsymbol{W} \boldsymbol{F}_t^T \\ \boldsymbol{F}_t \boldsymbol{W} & \boldsymbol{F}_t \boldsymbol{W} \boldsymbol{F}_t^T + \boldsymbol{V} \end{pmatrix} \right\}.$$

Using multivariate normal theory (Gamerman and Lopes, 2006), we obtain $\pi(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{t-1}, \boldsymbol{x}_t)$ as a Gaussian density with

$$E(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{t-1}, \boldsymbol{x}_t) = \boldsymbol{G}_t \boldsymbol{\theta}_{t-1} + \boldsymbol{W} \boldsymbol{F}_t^T (\boldsymbol{F}_t \boldsymbol{W} \boldsymbol{F}_t^T + \boldsymbol{V})^{-1} (\boldsymbol{x}_t - \boldsymbol{F}_t \boldsymbol{G}_t \boldsymbol{\theta}_{t-1})$$

and

$$Var(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{t-1}, \boldsymbol{x}_t) = \boldsymbol{W} - \boldsymbol{W} \boldsymbol{F}_t^T (\boldsymbol{F}_t \boldsymbol{W} \boldsymbol{F}_t^T + \boldsymbol{V})^{-1} \boldsymbol{F}_t \boldsymbol{W}.$$

A summary of the optimal BPF scheme is provided in Algorithm 6.

Algorithm 6 Optimal BPF scheme

1. Initialisation. For $k = 1, \ldots, N$, sample $\boldsymbol{\theta}_0^{(k)} \sim \pi(\boldsymbol{\theta}_0)$ and set $\tilde{\omega}_0^{(k)} = 1$.
2. For $t = 1, \ldots, n$:
   (a) Propagation. Sample $\boldsymbol{\theta}_t^{(k)} \sim \pi(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{t-1}^{(k)}, \boldsymbol{x}_t)$.
   (b) Update and normalize the importance weights via

$$\tilde{\omega}_t^{(k)} = \tilde{\omega}_{t-1}^{(k)} \pi(\boldsymbol{x}_t | \boldsymbol{\theta}_{t-1}^{(k)}), \qquad \omega_t^{(k)} = \frac{\tilde{\omega}_t^{(k)}}{\sum_{j=1}^N \tilde{\omega}_t^{(j)}}.$$

   (c) Sample indices $a_k \sim \mathcal{M}(\boldsymbol{\omega}^{1:N})$ and set $\{\boldsymbol{\theta}_t^{(k)}, \tilde{\omega}_t^{(k)}\} := \{\boldsymbol{\theta}_t^{(a_k)}, 1\}$.

3.2.3 Auxiliary particle filter (APF)

A general form of SIR was proposed by Pitt and Shephard (1999). The method introduces an additional step that prunes out particles that are inconsistent with the next observation before the propagation by introducing an auxiliary variable. This is the auxiliary particle filter (APF). Recall that the approximate target is given by

$$\hat{\pi}(\boldsymbol{\theta}_{0:t} | \boldsymbol{x}_{1:t}) \propto \pi(\boldsymbol{x}_t | \boldsymbol{\theta}_t) \pi(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{t-1}) \hat{\pi}(\boldsymbol{\theta}_{0:t-1} | \boldsymbol{x}_{1:t-1}).$$

Now consider the target

$$\hat{\pi}(\boldsymbol{\theta}_{0:t}, k | \boldsymbol{x}_{1:t}) \propto \pi(\boldsymbol{x}_t | \boldsymbol{\theta}_t) \pi(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{t-1}^{(k)}) \hat{\pi}(\boldsymbol{\theta}_{0:t-1}^{(k)} | \boldsymbol{x}_{1:t-1}) \tag{3.5}$$

from which we obtain $\hat{\pi}(\boldsymbol{\theta}_{0:t} | \boldsymbol{x}_{1:t})$ by marginalising over k. Pitt and Shephard (1999) then sample (3.5) via the importance density

$$g(\boldsymbol{\theta}_{0:t}, k | \boldsymbol{x}_{1:t}) \propto \tilde{\boldsymbol{\omega}}_{t-1}^{(k)} g(k | \boldsymbol{x}_t) g(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{t-1}, \boldsymbol{x}_t)$$

where $\tilde{\boldsymbol{\omega}}_{t-1}^{(k)} \propto \hat{\pi}(\boldsymbol{\theta}_{0:t-1}^{(k)} | \boldsymbol{x}_{1:t-1})$. Hence the weighted resampling procedure first generates indices a_k, $k = 1, \ldots, N$, by drawing from the discrete distribution on $\{1, \ldots, N\}$ with probabilities proportional to $\tilde{\omega}_{t-1}^{(k)} g(k | \boldsymbol{x}_t)$. The particles are then propagated via

$$\boldsymbol{\theta}_t^{(k)} \sim g(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{t-1}^{(a_k)}, \boldsymbol{x}_t), \qquad k = 1, \ldots, N.$$

Algorithm 7 APF scheme

1. Initialisation. For $k = 1, \ldots, N$, sample $\boldsymbol{\theta}_0^{(k)} \sim \pi(\boldsymbol{\theta}_0)$, and set $\tilde{\omega}_0^{(k)} = 1$.
2. For $t = 1, \ldots, n$:
   (a) Update and normalize the importance weights via

$$\tilde{\omega}_{t-1}^{*(k)} = \tilde{\omega}_{t-1}^{(k)} g(k|\boldsymbol{x}_t), \qquad \omega_{t-1}^{*(k)} = \frac{\tilde{\omega}_{t-1}^{*(k)}}{\sum_{j=1}^{N} \tilde{\omega}_{t-1}^{*(j)}}.$$

   (b) Sample indices $a_k \sim \mathcal{M}(\boldsymbol{\omega}^{*1:N})$ and set $\{\boldsymbol{\theta}_{t-1}^{(k)}, \tilde{\boldsymbol{\theta}}_t^{(k)}\} := \{\boldsymbol{\theta}_{t-1}^{(a_k)}, \tilde{\boldsymbol{\theta}}_t^{(a_k)}\}$.
   (c) Propagation. Sample $\boldsymbol{\theta}_t^{(k)} \sim g(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}^{(k)}, \boldsymbol{x}_t)$.
   (d) Update and normalize the importance weights via

$$\tilde{\omega}_t^{(k)} = \frac{\pi(\boldsymbol{x}_t|\boldsymbol{\theta}_t^{(a_k)})\pi(\boldsymbol{\theta}_t^{(k)}|\boldsymbol{\theta}_{t-1}^{(a_k)})}{g(a_k|\boldsymbol{x}_t)g(\boldsymbol{\theta}_t^{(k)}|\boldsymbol{\theta}_{t-1}^{(a_k)}, \boldsymbol{x}_t)}, \qquad \omega_t^{(k)} = \frac{\tilde{\omega}_t^{(k)}}{\sum_{j=1}^{N} \tilde{\omega}_t^{(j)}}.$$

The updated unnormalised weights can be seen to be

$$\tilde{\omega}_t^{(k)} \propto \frac{\pi(\boldsymbol{x}_t|\boldsymbol{\theta}_t^{(a_k)})\pi(\boldsymbol{\theta}_t^{(k)}|\boldsymbol{\theta}_{t-1}^{(a_k)})}{g(a_k|\boldsymbol{x}_t)g(\boldsymbol{\theta}_t^{(k)}|\boldsymbol{\theta}_{t-1}^{(a_k)}, \boldsymbol{x}_t)}, \qquad k = 1, \ldots, N.$$

Pitt and Shephard (1999) suggest that $g(k|\boldsymbol{x}_t) = \pi(\boldsymbol{x}_t|\tilde{\boldsymbol{\theta}}_t)$ where $\tilde{\boldsymbol{\theta}}_t$ is the mean, median or mode of $\pi(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}^{(k)})$. The optimal choice of the importance density can be found by noting that

$$\pi(\boldsymbol{x}_t|\boldsymbol{\theta}_t)\pi(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}) = \pi(\boldsymbol{x}_t|\boldsymbol{\theta}_{t-1})\pi(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}, \boldsymbol{x}_t)$$

which immediately suggests taking $g(k|\boldsymbol{x}_t) = \pi(\boldsymbol{x}_t|\boldsymbol{\theta}_{t-1})$ and $g(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}, \boldsymbol{x}_t) = \pi(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}, \boldsymbol{x}_t)$. For a DLM we may compute $\pi(\boldsymbol{x}_t|\boldsymbol{\theta}_{t-1})$ as

$$\pi(\boldsymbol{x}_t|\boldsymbol{\theta}_{t-1}) = N(\boldsymbol{x}_t; \boldsymbol{F}_t\boldsymbol{G}_t\boldsymbol{m}_{t-1}, \boldsymbol{F}_t(\boldsymbol{G}_t\boldsymbol{C}_{t-1}\boldsymbol{G}_t^T + \boldsymbol{W})\boldsymbol{F}_t^T + \boldsymbol{V})$$

where we have used $\boldsymbol{\theta}_{t-1}|\boldsymbol{x}_{1:t-1} \sim N(\boldsymbol{m}_{t-1}, \boldsymbol{C}_{t-1})$. Hence in this case, the updated unnormalised weights are one, and the APF is known as fully adapted (Pitt et al., 2012). The generic APF is given by Algorithm 7. Note that the bootstrap PF is obtained as a special case of the APF by taking $g(k|\boldsymbol{x}_t) = 1$.

3.3 State and parameter filtering

The general APF algorithm often performs well when all the parameters are known. However in practice, the model contains unknown parameters which need to be estimated from the data. Hence the unknown components include the hidden state and the model parameters. Therefore the target distribution becomes $\pi(\boldsymbol{\theta}_{0:t}, \boldsymbol{\phi}|\boldsymbol{x}_{1:t})$ where $\boldsymbol{\phi}$ denotes the unknown parameters. A difficulty to implement SMC in the presence of unknown parameters is that, as the dimension of the unknown parameter vector increases, the inherent problem of particle degeneracy will be exacerbated due to the limitation of increasing particle size arbitrarily.

In this section, we consider SMC schemes which are used to sequentially update the joint density of the posterior for the state and the model parameters. Note that in what follows, the parameters are always assumed to be time-invariant.

3.3.1 Liu-West algorithm

Liu and West (2001) introduced an approach that combines a kernel smoothing method (KS) with the APF to effectively regenerate the unknown static parameters at each time point. They further amended the problem of information loss that occurs by adding an artificial evolution noise to the parameters.

By Bayes' theorem, the target density $\pi(\boldsymbol{\theta}_{0:t}, \boldsymbol{\phi}|\boldsymbol{x}_{1:t})$ at time t can be written as

$$\pi(\boldsymbol{\theta}_{0:t}, \boldsymbol{\phi}|\boldsymbol{x}_{1:t}) \propto \pi(\boldsymbol{x}_t|\boldsymbol{\theta}_{t-1}, \boldsymbol{\phi})\pi(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}, \boldsymbol{\phi}, \boldsymbol{x}_t)\pi(\boldsymbol{\theta}_{0:t-1}, \boldsymbol{\phi}|\boldsymbol{x}_{1:t-1}).$$

The particle filter then replaces $\pi(\boldsymbol{\theta}_{0:t-1}, \boldsymbol{\phi}|\boldsymbol{x}_{1:t-1})$ with the discrete approximation

$$\hat{\pi}(\boldsymbol{\theta}_{0:t-1}, \boldsymbol{\phi}|\boldsymbol{x}_{1:t-1}) = \sum_{k=1}^{N} \omega_{t-1}^{(k)} \delta_{(\boldsymbol{\theta}_{0:t-1}^{(k)}, \boldsymbol{\phi}^{(k)})},$$

where $\delta_{(\boldsymbol{\theta}_{0:t-1}^{(k)}, \boldsymbol{\phi}^{(k)})}$ is the point mass at $(\boldsymbol{\theta}_{0:t-1}, \boldsymbol{\phi}) = (\boldsymbol{\theta}_{0:t-1}^{(k)}, \boldsymbol{\phi}^{(k)})$. Similarly, the estimate of the marginal distribution of $\boldsymbol{\phi}$ is

$$\hat{\pi}(\boldsymbol{\phi}|\boldsymbol{x}_{1:t-1}) = \sum_{k=1}^{N} \omega_{t-1}^{(k)} \delta_{\boldsymbol{\phi}^{(k)}}.$$

Based on the particle values at time $t-1$, we denote the expectation and variance of $\phi|\boldsymbol{x}_{1:t-1}$ estimated by the particle filter as $\bar{\phi}$ and Σ. Diversity can be introduced into the particle set by replacing $\hat{\pi}(\phi|\boldsymbol{x}_{1:t-1})$ with a kernel density estimate, found by replacing the point mass $\delta_{\phi^{(k)}}$ by the density of some random variable following a normal distribution $N(\phi^{(k)}, \tilde{\Sigma})$, where $\tilde{\Sigma} = \gamma\Sigma$ for some tuning parameter γ, that is, to generate a new set of particles by adding a jitter to the previous ones. Plainly, this will result in the error for the approximating distribution increasing to $(1+\gamma)\Sigma$. As time goes on, the error will recursively increase and consequently the information loss will occur. The Liu-West algorithm overcomes this problem as follows.

The parameter particles conditional on the auxiliary variable indicating the mixture component has a normal distribution with

$$\phi|k \sim N(\boldsymbol{\eta}^{(k)}, \gamma\Sigma), \qquad k = 1, \ldots, N$$

where the kernel locations are specified using a shrinkage rule proposed by West (1993a) and West (1993b) as $\boldsymbol{\eta}^{(k)} = \kappa\phi^{(k)} + (1-\kappa)\bar{\phi}$, that corrects the particle over-dispersion. The shrinkage factor κ can be written as $\kappa = (3\delta - 1)/2\delta$ using a discount factor δ. The value of δ is typically chosen from $(0.95, 0.99)$, so the range for κ is $(0.974, 0.995)$. The tuning parameter γ is specified as $\gamma = 1 - \kappa^2$. With these kernel locations, $\mathrm{E}(\phi|k)$ and $\mathrm{Var}(\phi|k)$ remain same as $\bar{\phi}$ and Σ under the mixture approximation. The posterior under the Liu-West scheme is then given by

$$\hat{\pi}_{LW}(\boldsymbol{\theta}_{0:t}, \phi|\boldsymbol{x}_{1:t}) \propto \pi(\boldsymbol{x}_t|\boldsymbol{\theta}_t, \phi)\pi(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}, \phi)\hat{\pi}_{LW}(\boldsymbol{\theta}_{0:t-1}, \phi|\boldsymbol{x}_{1:t-1}),$$

where $\hat{\pi}_{LW}(\boldsymbol{\theta}_{0:t-1}, \phi|\boldsymbol{x}_{1:t-1}) = \sum_{k=1}^{N} \omega_{t-1}^{(k)} N(\phi; \boldsymbol{\eta}^{(k)}, \gamma\Sigma)\delta_{\boldsymbol{\theta}_{0:t-1}^{(k)}}$. In practice, we may take Σ as a diagonal matrix in order to speed up the algorithm calculation.

Following the auxiliary particle filter of Section 3.2.3, we may further write

$$\hat{\pi}_{LW}(\boldsymbol{\theta}_{0:t}, \phi, k|\boldsymbol{x}_{1:t}) \propto \pi(\boldsymbol{x}_t|\boldsymbol{\theta}_t, \phi^{(k)})\pi(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}^{(k)}, \phi^{(k)})\hat{\pi}_{LW}(\boldsymbol{\theta}_{0:t-1}^{(k)}, \phi^{(k)}|\boldsymbol{x}_{1:t-1}).$$

Since $\pi(\boldsymbol{x}_t|\boldsymbol{\theta}_t, \phi)\pi(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}, \phi) = \pi(\boldsymbol{x}_t|\boldsymbol{\theta}_{t-1}, \phi)\pi(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}, \boldsymbol{x}_t, \phi)$, a simple choice of the importance density is to take

$$g(\boldsymbol{\theta}_{0:t}, \phi, k|\boldsymbol{x}_{1:t}) \propto g(\boldsymbol{\theta}_{0:t-1}^{(k)}, \phi^{(k)}|\boldsymbol{x}_{1:t-1})\pi(\boldsymbol{x}_t|\boldsymbol{\theta}_{t-1}^{(k)}, \phi^{(k)})\pi(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}^{(k)}, \boldsymbol{x}_t, \phi^{(k)}).$$

As we have seen, due to linearity and normality of a DLM, $\pi(\boldsymbol{x}_t|\boldsymbol{\theta}_{t-1}^{(k)}, \phi^{(k)})$ and $\pi(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}^{(k)}, \boldsymbol{x}_t, \phi^{(k)})$ are both tractable. By taking the fully adapted APF, the updated unnormalised weights always equal one. A summary of Liu-West algorithm is provided in Algorithm 8.

Algorithm 8 Liu-West algorithm (FA-APF+KS)

1. Initialisation. For $k = 1, \ldots, N$, sample $\boldsymbol{\theta}_0^{(k)} \sim \pi(\boldsymbol{\theta}_0)$ and $\boldsymbol{\phi}^{(k)} \sim \pi(\boldsymbol{\phi})$, and set $\tilde{\omega}_0^{(k)} = 1$.
2. For $t = 1, \ldots, n$:
   (a) Kernel summaries. Calculate $\boldsymbol{\eta}^{(k)} = \kappa \boldsymbol{\phi}^{(k)} + (1 - \kappa)\bar{\boldsymbol{\phi}}_x$ and $\Sigma = \widehat{Var}(\boldsymbol{\phi}|\boldsymbol{x}_{1:t-1})$.
   (b) Update and normalize the importance weights via

$$\tilde{\omega}_t^{(k)} = \tilde{\omega}_{t-1}^{(k)} \pi(\boldsymbol{x}_t|\boldsymbol{\theta}_{t-1}^{(k)}, \boldsymbol{\phi}^{(k)}), \qquad \omega_t^{(k)} = \frac{\tilde{\omega}_t^{(k)}}{\sum_{j=1}^{N} \tilde{\omega}_t^{(j)}}.$$

   (c) Sample indices $a_k \sim \mathcal{M}(\boldsymbol{\omega}^{1:N})$ and set $\{\boldsymbol{\eta}^{(k)}, \boldsymbol{\phi}^{(k)}, \boldsymbol{\theta}_{t-1}^{(k)}, \tilde{\omega}_t^{(k)}\} := \{\boldsymbol{\eta}^{(a_k)}, \boldsymbol{\phi}^{(a_k)}, \boldsymbol{\theta}_{t-1}^{(a_k)}, 1\}$.
   (d) Propagation. Sample $\boldsymbol{\phi}^* \sim N(\boldsymbol{\eta}^{(k)}, \gamma\Sigma)$ and $\boldsymbol{\theta}_t^{(k)} \sim \pi(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}^{(k)}, \boldsymbol{x}_t, \boldsymbol{\phi}^*)$. Put $\boldsymbol{\phi}^{(k)} := \boldsymbol{\phi}^*$.

A main issue of the Liu-West algorithm is the sensitivity of the parameter estimates to the seemingly arbitrary selection of the shrinkage parameter κ. If the choice of κ is not appropriate, it may result in a high Monte Carlo error or even cause the inaccurate estimation of the parameters. Rios and Lopes (2013) gave empirical studies to compare different particle filtering schemes, where the result shows that the Liu-West algorithm is prone to particle degeneracy and underperformance in terms of model accuracy when comparing to the Storvik algorithm and particle learning which will be introduced in the next sections.

3.3.2 Storvik algorithm

A method introduced by Storvik (2002) and Fearnhead (2002) overcomes the problem of a random selection of the shrinkage parameter in the Liu-West algorithm. The scheme is able to deal with static parameters by considering recursive updates based on sufficient statistics for the parameters. This approach can be implemented in situations where the parameter posterior, conditional on the data and latent states, is tractable, and depends on some low dimensional vector $\boldsymbol{\xi}$. In reality the method can be considered as an extension of the BPF by embedding the update of sufficient statistics into the filter. For this scheme, it is assumed that the target posterior

of the state and parameters can be expanded as

$$\pi(\boldsymbol{\theta}_{0:t}, \boldsymbol{\phi}|\boldsymbol{x}_{1:t}) \propto \pi(\boldsymbol{x}_t|\boldsymbol{\theta}_t, \boldsymbol{\phi})\pi(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}, \boldsymbol{\phi})\pi(\boldsymbol{\phi}|\boldsymbol{\theta}_{0:t-1}, \boldsymbol{x}_{1:t-1})\pi(\boldsymbol{\theta}_{0:t-1}|\boldsymbol{x}_{1:t-1})$$

$$\propto \pi(\boldsymbol{x}_t|\boldsymbol{\theta}_t, \boldsymbol{\phi})\pi(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}, \boldsymbol{\phi})\pi(\boldsymbol{\phi}|\boldsymbol{\xi}_{0:t-1})\pi(\boldsymbol{\theta}_{0:t-1}|\boldsymbol{x}_{1:t-1}),$$

where the statistic $\boldsymbol{\xi}_{0:t-1}$ is sufficient for the parameters conditional on the observations and latent states up to time $t-1$. Hence we may write $\pi(\boldsymbol{\phi}|\boldsymbol{\theta}_{0:t-1}, \boldsymbol{x}_{1:t-1}) = \pi(\boldsymbol{\phi}|\boldsymbol{\xi}_{0:t-1})$. Therefore, the storvik filter uses the approximation

$$\hat{\pi}_{sto}(\boldsymbol{\theta}_{0:t}, \boldsymbol{\phi}|\boldsymbol{x}_{1:t}) \propto \pi(\boldsymbol{x}_t|\boldsymbol{\theta}_t, \boldsymbol{\phi})\pi(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}, \boldsymbol{\phi})\hat{\pi}_{sto}(\boldsymbol{\theta}_{0:t-1}, \boldsymbol{\phi}|\boldsymbol{x}_{1:t-1}) \tag{3.6}$$

where $\hat{\pi}_{sto}(\boldsymbol{\theta}_{0:t-1}, \boldsymbol{\phi}|\boldsymbol{x}_{1:t-1}) = \sum_{k=1}^{N} \omega_{t-1}^{(k)} \pi(\boldsymbol{\phi}|\boldsymbol{\theta}_{0:t-1}^{(k)}, \boldsymbol{x}_{1:t-1})\delta_{\boldsymbol{\theta}_{0:t-1}^{(k)}}$. We sample (3.6) by using the importance density

$$g(\boldsymbol{\theta}_{0:t}, \boldsymbol{\phi}|\boldsymbol{x}_{1:t}) \propto \pi(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}, \boldsymbol{\phi})g(\boldsymbol{\theta}_{0:t-1}, \boldsymbol{\phi}|\boldsymbol{x}_{1:t-1}).$$

Note that we adapt the convention that the parameter sampling step is performed after propagation.

Example: Local Level Model

To illustrate the Storvik filter, consider the local level model of Section 2.1. We choose a conjugate prior for the parameters of the observation variance and the system variance, that is, $\phi_1 = V$ and $\phi_2 = W$. We take $\phi_1 \sim IG(\alpha_{v,0}, \beta_{v,0})$ and $\phi_2 \sim IG(\alpha_{w,0}, \beta_{w,0})$. Thus the initial sufficient statistics is $\boldsymbol{\xi} = (\alpha_{v,0}, \beta_{v,0}, \alpha_{w,0}, \beta_{w,0})$. To update the sufficient statistics associated with $\phi_1^{(k)}|\boldsymbol{x}_{1:t}, \boldsymbol{\theta}_{0:t}^{(k)} \sim IG(\alpha_{v,t}^{(k)}, \beta_{v,t}^{(k)})$ and $\phi_2^{(k)}|\boldsymbol{x}_{1:t}, \boldsymbol{\theta}_{0:t}^{(k)} \sim IG(\alpha_{w,t}^{(k)}, \beta_{w,t}^{(k)})$, for $k = 1, \ldots, N$, we have

$$\pi(\phi_1^{(k)}|\boldsymbol{x}_{1:t}, \boldsymbol{\theta}_{0:t}^{(k)}) \propto \underbrace{\pi(x_t|\theta_t^{(k)}, \phi_1^{(k)})}_{\text{observation equation}} \underbrace{\pi(\phi_1^{(k)}|\boldsymbol{x}_{1:t-1}, \boldsymbol{\theta}_{0:t-1}^{(k)})}_{\text{prior}}$$

$$\propto (\phi_1^{(k)})^{-\frac{1}{2}} \exp\left[-\frac{(x_t - \theta_t^{(k)})^2}{2\phi_1^{(k)}}\right] (\phi_1^{(k)})^{-\alpha_v^*-1} \exp\left(-\frac{\beta_v^*}{\phi_1^{(k)}}\right)$$

$$\propto (\phi_1^{(k)})^{-(\frac{1}{2}+\alpha_v^*)-1} \exp\left\{-\left[\beta_v^* + \frac{(x_t - \theta_t^{(k)})^2}{2}\right] \Big/ \phi_1^{(k)}\right\}.$$

Algorithm 9 Storvik algorithm (BPF+SS)

1. Initialisation. For $k = 1, \ldots, N$, set $\boldsymbol{\xi}^{(k)}$ to be the prior hyperparameters and $\tilde{\omega}_0^{(k)} := 1$, sample $\boldsymbol{\theta}_0^{(k)} \sim \pi(\boldsymbol{\theta}_0)$, $\boldsymbol{\phi}^{(k)} \sim \pi(\boldsymbol{\phi}|\boldsymbol{\xi}^{(k)})$.
2. For $t = 1, \ldots, n$:
   (a) Propagation. Sample $\boldsymbol{\theta}_t^{(k)} \sim \pi(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}^{(k)}, \boldsymbol{\phi}^{(k)})$.
   (b) Update and normalize the importance weights via

$$\tilde{\omega}_t^{(k)} = \tilde{\omega}_{t-1}^{(k)} \pi(\boldsymbol{x}_t|\boldsymbol{\theta}_t^{(k)}, \boldsymbol{\phi}^{(k)}), \qquad \omega_t^{(k)} = \frac{\tilde{\omega}_t^{(k)}}{\sum_{j=1}^N \tilde{\omega}_t^{(j)}}.$$

   (c) Sample indices $a_k \sim \mathcal{M}(\boldsymbol{\omega}^{1:N})$ and set $\{\boldsymbol{\xi}^*, \boldsymbol{\theta}_{t-1}^{(k)}, \boldsymbol{\theta}_t^{(k)}, \tilde{\omega}_t^{(k)}\} := \{\boldsymbol{\xi}^{(a_k)}, \boldsymbol{\theta}_{t-1}^{(a_k)}, \boldsymbol{\theta}_t^{(a_k)}, 1\}$.
   (d) Update the sufficient statistics $\boldsymbol{\xi}^{(k)} = S(\boldsymbol{\xi}^*, \boldsymbol{\theta}_{t-1}^{(k)}, \boldsymbol{\theta}_t^{(k)}, \boldsymbol{x}_t)$.
   (e) Sample $\boldsymbol{\phi}^* \sim \pi(\boldsymbol{\phi}|\boldsymbol{\xi}^{(k)})$. Put $\boldsymbol{\phi}^{(k)} := \boldsymbol{\phi}^*$.

Therefore for the conditional distribution $\pi(\phi_1^{(k)}|\cdot)$, we can update its sufficient statistics through

$$\alpha_{v,t}^{(k)} = \alpha_v^* + \frac{1}{2},$$

$$\beta_{v,t}^{(k)} = \beta_v^* + \frac{(x_t - \theta_t^{(k)})^2}{2}.$$

Analogously we can update $\pi(\phi_2^{(k)}|\cdot)$ by storing the quantities

$$\alpha_{w,t}^{(k)} = \alpha_w^* + \frac{1}{2},$$

$$\beta_{w,t}^{(k)} = \beta_w^* + \frac{(\theta_t^{(k)} - \theta_{t-1}^{(k)})^2}{2}.$$

The Storvik algorithm has some of drawbacks which may affect its practicality. First, it is limited to models for which a prior can be chosen that is conjugate with respect to the conditional posterior. If the prior is not conjugate, the Storvik algorithm can not be implemented. Additionally, the particle degeneracy issue may easily arise since the scheme was introduced based on a blind propagation without considering observation information. Furthermore, as discussed by Chopin et al. (2010), the algorithm does not completely overcome degeneracy of the static parameter particle set, due to the particle representation of the sufficient statistic and the use of resampling steps.

3.3.3 Particle learning (PL)

Carvalho et al. (2010) and Lopes et al. (2011) proposed a similar particle filter by applying the sufficient statistics method for updating the static parameters, and avoiding the blind propagation issue by embedding the concept of the APF. The method is so called particle learning (PL), which can be thought of as an extension of APF. Particle learning uses the approximation

$$\hat{\pi}_{PL}(\boldsymbol{\theta}_{0:t}, \boldsymbol{\phi}, k|\boldsymbol{x}_{1:t}) \propto \pi(\boldsymbol{x}_t|\boldsymbol{\theta}_t)\pi(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}^{(k)})\hat{\pi}_{PL}(\boldsymbol{\theta}_{0:t-1}^{(k)}, \boldsymbol{\phi}|\boldsymbol{x}_{1:t-1}), \tag{3.7}$$

where

$$\hat{\pi}_{PL}(\boldsymbol{\theta}_{0:t-1}^{(k)}, \boldsymbol{\phi}|\boldsymbol{x}_{1:t-1}) = \sum_{k=1}^{N} \pi(\boldsymbol{\phi}|\boldsymbol{\theta}_{0:t-1}^{(k)}, \boldsymbol{x}_{1:t-1})\omega_{t-1}^{(k)}\delta_{\boldsymbol{\theta}_{0:t-1}^{(k)}}$$

and marginalising out k in (3.7) gives samples from $\hat{\pi}_{PL}(\boldsymbol{\theta}_{0:t}, \boldsymbol{\phi}|\boldsymbol{x}_{1:t})$. The importance density is taken as

$$g(\boldsymbol{\theta}_{0:t}, \boldsymbol{\phi}, k|\boldsymbol{x}_{1:t}) \propto g(\boldsymbol{\theta}_{0:t-1}^{(k)}, \boldsymbol{\phi}|\boldsymbol{x}_{1:t-1})\pi(\boldsymbol{x}_t|\boldsymbol{\theta}_{t-1}^{(k)}, \boldsymbol{\phi})\pi(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}^{(k)}, \boldsymbol{x}_t, \boldsymbol{\phi})$$

and hence uses the fully adapted APF, combined with the tractability of the conditional parameter posterior.

Comparing with the Storvik algorithm, particle degeneracy here can be alleviated to some extent as PL includes the information of the new observation into the model before the particle propagation. However, degeneracy may eventually still happen due to the resampling step unless the particle size N increases exponentially with time (Chopin et al., 2010). Additionally, PL has the same problem as the Storvik algorithm suffering from the limitation of the choice of the conjugate prior. A summary of the PL algorithm is provided in Algorithm 10.

3.3.4 Iterated batch importance sampling (IBIS)

The iterated batch importance sampling (IBIS) is another sequential Bayesian inference algorithm that approximates the target through recursive resampling from $\pi(\boldsymbol{\phi}|\boldsymbol{x}_{1:t})$, together with MCMC steps for rejuvenating parameter samples in order to circumvent particle degeneracy; see, for example, Chopin (2002) and Chopin et al. (2013).

Essentially, it is supposed that primary interest lies in the marginal parameter posterior (at time t) given by

$$\pi(\boldsymbol{\phi}|\boldsymbol{x}_{1:t}) \propto \pi(\boldsymbol{x}_t|\boldsymbol{x}_{1:t-1}, \boldsymbol{\phi})\pi(\boldsymbol{\phi}|\boldsymbol{x}_{1:t-1})$$

Algorithm 10 PL algorithm (APF+SS)

1. Initialisation. For $k = 1, \ldots, N$, set $\boldsymbol{\xi}^{(k)}$ to be the prior hyperparameters and $\tilde{\omega}_0^{(k)} = 1$, sample $\boldsymbol{\theta}_0^{(k)} \sim \pi(\boldsymbol{\theta}_0)$, $\boldsymbol{\phi}^{(k)} \sim \pi(\boldsymbol{\phi}|\boldsymbol{\xi}^{(k)})$.
2. For $t = 1, \ldots, n$:
   (a) Update and normalize the importance weights via
   $$\tilde{\omega}_t^{(k)} = \tilde{\omega}_{t-1}^{(k)} \pi(\boldsymbol{x}_t|\boldsymbol{\theta}_{t-1}^{(k)}, \boldsymbol{\phi}^{(k)}), \qquad \omega_t^{(k)} = \frac{\tilde{\omega}_t^{(k)}}{\sum_{j=1}^N \tilde{\omega}_t^{(j)}}.$$
   (b) Sample indices $a_k \sim \mathcal{M}(\boldsymbol{\omega}^{1:N})$ and set $\{\boldsymbol{\phi}^*, \boldsymbol{\xi}^*, \boldsymbol{\theta}_{t-1}^{(k)}, \tilde{\omega}_t^{(k)}\} := \{\boldsymbol{\phi}^{(a_k)}, \boldsymbol{\xi}^{(a_k)}, \boldsymbol{\theta}_{t-1}^{(a_k)}, 1\}$.
   (c) Propagation. Sample $\boldsymbol{\theta}_t^{(k)} \sim \pi(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}^{(k)}, \boldsymbol{\phi}^*, \boldsymbol{x}_t)$.
   (d) Update the sufficient statistics $\boldsymbol{\xi}^{(k)} = S(\boldsymbol{\xi}^*, \boldsymbol{\theta}_{t-1}^{(k)}, \boldsymbol{\theta}_t^{(k)}, \boldsymbol{x}_t)$.
   (e) Sample $\boldsymbol{\phi}^* \sim \pi(\boldsymbol{\phi}|\boldsymbol{\xi}^{(k)})$. Put $\boldsymbol{\phi}^{(k)} := \boldsymbol{\phi}^*$.

which immediately suggests a scheme where samples are drawn from $\pi(\boldsymbol{\phi}|\boldsymbol{x}_{1:t})$ and weighted by $\pi(\boldsymbol{x}_t|\boldsymbol{x}_{1:t-1}, \boldsymbol{\phi})$. Hence, given a weighted sample $\{\boldsymbol{\phi}^{(k)}, \omega_{t-1}^{(k)}\}_{k=1}^N$ from $\pi(\boldsymbol{\phi}|\boldsymbol{x}_{1:t-1})$, the weights are updated at time t via

$$\omega_t^{(k)} \propto \omega_{t-1}^{(k)} \pi(\boldsymbol{x}_t|\boldsymbol{x}_{1:t-1}, \boldsymbol{\phi}^{(k)}).$$

Note that for a DLM, the observed data likelihood increment is tractable and is obtained via the forward filter (see Algorithm 2).

Simply updating the incremental weights over the time will lead to particle degeneracy. To circumvent this issue, the IBIS scheme uses a resample-move step (Gilks and Berzuini, 2001) that first resamples parameter particles (by drawing indices from a multinomial $\mathcal{M}(\boldsymbol{\omega}^{1:N})$ distribution) and then moves each parameter sample through a Metropolis-Hastings kernel which leaves the target posterior invariant. Running a resample-move step is expensive, and it is only used if some degeneracy criterion is fulfilled, such as ESS $< \delta N$ for $\delta \in (0, 1)$, where a standard choice is $\delta = 0.5$. When the parameters must be strictly positive (as is the case for the model in Chapter 5), we take a proposal density

$$q(\boldsymbol{\phi}^*|\boldsymbol{\phi}) = \log N(\boldsymbol{\phi}^*; \log(\boldsymbol{\phi}), \gamma Var(\log(\boldsymbol{\phi})|\boldsymbol{x}_{0:t})) \tag{3.8}$$

where $\log N(\cdot; m, V)$ denotes the density associated with the exponential of a $N(m, V)$ random variable. We use the standard rule of thumb by taking the scaling parameter $\gamma = 2.38^2/n_{par}$.

Algorithm 11 IBIS scheme

1. Initialisation. For $k = 1, \ldots, N$ sample $\boldsymbol{\phi}^{(k)} \sim \pi(\cdot)$ and set $\tilde{\omega}_0^{(k)} = 1$. Store $\boldsymbol{m}_0^{(k)}$ and $\boldsymbol{C}_0^{(k)}$. For $t = 1, \ldots, n$:

2. Sequential importance sampling. For $k = 1, \ldots, N$:

   (a) Perform iteration i of the forward filter to obtain $\pi(\boldsymbol{x}_t|\boldsymbol{x}_{1:t-1}, \boldsymbol{\phi}^{(k)})$, $\boldsymbol{m}_t^{(k)}$ and $\boldsymbol{C}_t^{(k)}$. Note the convention that $\pi(\boldsymbol{x}_1|\boldsymbol{\phi}^{(k)}) = \pi(\boldsymbol{x}_1|\boldsymbol{x}_{1:0}, \boldsymbol{\phi}^{(k)})$.

   (b) Update and normalize the importance weights via

$$\tilde{\omega}_t^{(k)} = \tilde{\omega}_{t-1}^{(k)} \pi(\boldsymbol{x}_t|\boldsymbol{x}_{1:t-1}, \boldsymbol{\phi}^{(k)}), \qquad \omega_t^{(k)} = \frac{\tilde{\omega}_t^{(k)}}{\sum_{j=1}^N \tilde{\omega}_t^{(j)}}.$$

   (c) Update the observed data likelihood via

$$\pi(\boldsymbol{x}_{1:t}|\boldsymbol{\phi}^{(k)}) = \pi(\boldsymbol{x}_{1:t-1}|\boldsymbol{\phi}^{(k)}) \pi(\boldsymbol{x}_t|\boldsymbol{x}_{1:t-1}, \boldsymbol{\phi}^{(k)}).$$

3. If ESS $< \delta N$ resample and move. For $k = 1, \ldots, N$:

   (a) Sample indices $a_k \sim \mathcal{M}(\boldsymbol{\omega}^{1:N})$ and set $\{\boldsymbol{\phi}^{(k)}, \tilde{\omega}_t^{(k)}\} := \{\boldsymbol{\phi}^{(a_k)}, 1\}$ and $\pi(\boldsymbol{x}_{1:t}|\boldsymbol{\phi}^{(k)}) := \pi(\boldsymbol{x}_{1:t}|\boldsymbol{\phi}^{(a_k)})$, $\boldsymbol{m}_t^{(k)} := \boldsymbol{m}_t^{(a_k)}$ and $\boldsymbol{C}_t^{(k)} := \boldsymbol{C}_t^{(a_k)}$.

   (b) Propose $\boldsymbol{\phi}^* \sim q(\cdot|\boldsymbol{\phi}^{(k)})$. Perform iterations $1, \ldots, i$ of the forward filter to obtain $\pi(\boldsymbol{x}_{1:t}|\boldsymbol{\phi}^*)$, $\boldsymbol{m}_t^*$ and $\boldsymbol{C}_t^*$. With probability

$$\min\left\{1, \frac{\pi(\boldsymbol{\phi}^*)\pi(\boldsymbol{x}_{1:t}|\boldsymbol{\phi}^*)}{\pi(\boldsymbol{\phi}^{(k)})\pi(\boldsymbol{x}_{1:t}|\boldsymbol{\phi}^{(k)})} \times \frac{q(\boldsymbol{\phi}^{(k)}|\boldsymbol{\phi}^*)}{q(\boldsymbol{\phi}^*|\boldsymbol{\phi}^{(k)})}\right\},$$

   put $\boldsymbol{\phi}^{(k)} := \boldsymbol{\phi}^*$, and $\pi(\boldsymbol{x}_{1:t}|\boldsymbol{\phi}^{(k)}) := \pi(\boldsymbol{x}_{1:t}|\boldsymbol{\phi}^*)$, $\boldsymbol{m}_t^{(k)} := \boldsymbol{m}_t^*$ and $\boldsymbol{C}_t^{(k)} := \boldsymbol{C}_t^*$.

Therefore to evaluate the acceptance probability $A(\boldsymbol{\phi}^{(k)}|\boldsymbol{\phi}^*)$ of the proposed move, we have

$$A(\boldsymbol{\phi}^{(k)}|\boldsymbol{\phi}^*) = \frac{\pi(\boldsymbol{x}_1|\boldsymbol{\phi}^*)}{\pi(\boldsymbol{x}_1|\boldsymbol{\phi}^{(k)})} \prod_{j=2}^{t} \frac{\pi(\boldsymbol{x}_j|\boldsymbol{x}_{1:j-1}, \boldsymbol{\phi}^*)}{\pi(\boldsymbol{x}_j|\boldsymbol{x}_{1:j-1}, \boldsymbol{\phi}^{(k)})} \prod_{i=1}^{n_{par}} \frac{\pi(\phi_i^*)}{\pi(\phi_i^{(k)})} \prod_{i=1}^{n_{par}} \frac{\phi_i^*}{\phi_i^{(k)}}.$$

Note that if interest is in the full posterior $\pi(\boldsymbol{\theta}_{0:n}, \boldsymbol{\phi}|\boldsymbol{x}_{1:n})$, then samples can be generated post-hoc. Given equally weighted draws $\{\boldsymbol{\phi}^{(k)}\}_{k=1}^N$ from $\pi(\boldsymbol{\phi}|\boldsymbol{x}_{1:n})$, the backward sampler (see Algorithm 2) can be applied for each $\boldsymbol{\phi}^{(k)}$, to give draws $\boldsymbol{\theta}_{0:n}^{(k)}|\boldsymbol{x}_{1:n}, \boldsymbol{\phi}^{(k)}$ from $\pi(\boldsymbol{\theta}_{0:n}|\boldsymbol{x}_{1:n}, \boldsymbol{\phi})$. The full details of the IBIS scheme is summarised in Algorithm 11.

3.3.5 Adaptive iterated batch importance sampling (aIBIS)

The adaptive iterated batch importance sampling algorithm (aIBIS) algorithm introduced by Fearnhead and Taylor (2013) is an extension of IBIS. The main idea of aIBIS is to allow the scaling γ of the tuning parameter within the random walk proposal to be a random variable in each rejuvenation step and moreover, the proposed particles could be generated from different types of MCMC kernels (Fearnhead and Taylor, 2013). The purpose is to improve the efficiency of the resample-move step by selecting the best scaling γ and MCMC kernel for each particle, to further reduce the chance of performing particle rejuvenation. The density of the square root of the scaling $u_t = \sqrt{\gamma}$ is defined as

$$\pi(u_t) \propto \sum_{k=1}^{N} f(\tilde{\Lambda}(\phi^{(k)}, \phi^*)) R(u_t | u_{t-1}^{(k)}),$$

where $\tilde{\Lambda}(\phi^{(k)}, \phi^*) = A(\phi^{(k)}|\phi^*)\Lambda(\phi^{(k)}, \phi^*)$ is an adjusted value of the expected square jumping distance (ESJD) found by multiplying the acceptance probability with the ESJD. Here the ESJD is given by

$$\Lambda(\phi^{(k)}, \phi^*) = (\phi^{(k)} - \phi^*)^T Var(\phi | \boldsymbol{x}_{1:t})(\phi^{(k)} - \phi^*).$$

The ESJD is a computational measurement of the mixing of the Markov chain and maximising the ESJD is equivalent to minimising the autocorrelation of particles; see examples in Sherlock and Roberts (2009) and Pasarica and Gelman (2010). Here $R(\cdot|\cdot)$ is a density for u_t with centre $u_{t-1}^{(k)}$, where $\{u_{t-1}^{(k)}\}_{k=1}^{N}$ are the current scaling set, and $f(\cdot)$ is a function of the ESJD. In practice, $f(\cdot)$ should be increasing with $\tilde{\Lambda}$ so that more weight is assigned to a scaling which generates a larger ESJD. We follow Fearnhead and Taylor (2013) by taking a linear funtion $f(\tilde{\Lambda}) = \tilde{\Lambda} + \zeta$, $\zeta \geq 0$. A resampled set of scalings will be updated based on the probabilities proportional to the weights as $f(\cdot)$. If the rejuvenation step is not triggered then $u_t^{(k)} = u_{t-1}^{(k)}$. The initial collection of $\{u_0^{(k)}\}_{k=1}^{N}$ can be drawn from an arbitrary distribution, such as a uniform distribution.

A further improvement of aIBIS is that it allows choice of different MCMC kernels in the rejuvenation step. Suppose there are n_K MCMC kernels, each defined by a proposal distribution $q_{u,j}$, where $j \in \{1, \ldots, n_K\}$. Then each resampled particle $\phi^{(k)}$ is assigned a random kernel type and an associated scaling as a pair $(u_t^{(k)}, j_t^{(k)})$. The joint density of scaling and kernel type is defined as

$$\pi(u_t, j_t) \propto \sum_{k=1}^{N} f(\tilde{\Lambda}(\phi^{(k)}, \phi^*)) R(u_t | u_{t-1}^{(k)}) \delta_{j_{t-1}^{(k)}}(j_t),$$

where $\delta_{j_{t-1}^{(k)}}(j)$ is a point mass on $j_t = j_{t-1}^{(k)}$. For example, we can consider two different MCMC

kernels in practice: the random walk proposal and the Liu-West proposal (kernel smoothing) with

$$q_{rw}(\phi^*|\phi^{(k)}) = \log N(\phi^*;\ \log \phi^{(k)},\ (u_{t-1}^{(k)})^2 Var(\log \phi|\boldsymbol{x}_{1:t})), \tag{3.9}$$

$$q_{lw}(\phi^*|\phi^{(k)}) = \log N(\phi^*;\ \log \left[\alpha_{t-1}^{(k)}\phi^{(k)} + (1-\alpha_{t-1}^{(k)})\bar{\phi}\right],\ (u_{t-1}^{(k)})^2 Var(\log \phi|\boldsymbol{x}_{1:t})). \tag{3.10}$$

The prior distribution of u_t for the random walk proposal can be a uniform distribution $U(0,c)$, where c is a constant hyperparameter. The prior distribution of u_t for the Liu-West proposal should always be $U(0,1)$ due to the definition of $\alpha_t = \sqrt{1-u_t^2}$, $\alpha_t > 0$. Algorithm 12 summarises the aIBIS scheme.

3.3.6 Online IBIS

The main computational bottleneck of IBIS (or aIBIS) is the resample-move step. If this step is triggered at time t, then the observed data likelihood $\pi(\boldsymbol{x}_{1:t}|\phi^*)$ must be calculated for each proposed particle ϕ^*. Consequently, the computational cost grows with t, precluding the use of IBIS as an online scheme. To bound the computational cost of assimilating a single observation, we modify the resample-move step by basing the observed data likelihood on an observation window whose time length is chosen to balance accuracy and computational efficiency.

We follow a similar approach introduced by Del Moral et al. (2017) and define a sequence of windows with equal widths, say T, over the observation period. Hence, divide the observation period into b windows, $s \in \{1,\dots,b\}$ and denote by $\boldsymbol{x}_{t_i^s}$ the ith observation in window s, for $i = 1,\dots,n_s$. The observation times satisfy $t_i^s \in ((s-1)T, sT]$ when $s = 1,\dots,b-1$ and $t_i^s \in ((b-1)T, t_{n_b}^b]$ when $s = b$. The standard IBIS scheme is run over the first window. For windows $s = 2,\dots,b$, the resample-move step targets

$$\tilde{\pi}(\phi|\boldsymbol{x}_{1:t_i^s}) \propto \tilde{\pi}(\phi|\boldsymbol{x}_{1:(s-1)T})\pi(\boldsymbol{x}_{t_1^s:t_i^s}|\boldsymbol{x}_{1:(s-1)T}, \phi) \tag{3.11}$$

where

$$\tilde{\pi}(\phi|\boldsymbol{x}_{1:(s-1)T}) = \frac{1}{N}\sum_{k=1}^{N} \log N(\phi; \log \phi^{(k)}, h_s^2)$$

is a kernel density estimate (KDE) of $\pi(\phi|\boldsymbol{x}_{1:(s-1)T})$ and the bandwidth h_s^2 can be calculated using, for example, Silverman's rule of thumb (Silverman, 1986) as

$$h_s^2 = 1.06^2 N^{-2/5} \widehat{Var}(\phi^{(1:N)}|\boldsymbol{x}_{1:(s-1)T}).$$

Algorithm 12 aIBIS scheme

1. Initialisation. For $k = 1,\dots,N$ sample $(\boldsymbol{\phi}^{(k)}, u_0^{(k)}, j_0^{(k)}) \sim \pi(\cdot)$ and set $\tilde{\omega}_0^{(k)} = 1$. Store $\boldsymbol{m}_0^{(k)}$ and $\boldsymbol{C}_0^{(k)}$.
   For $t = 1,\dots,n$:

2. Sequential importance sampling. For $k = 1,\dots,N$:

   (a) Perform iteration i of the forward filter to obtain $\pi(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1}, \boldsymbol{\phi}^{(k)})$, $\boldsymbol{m}_t^{(k)}$ and $\boldsymbol{C}_t^{(k)}$. Note the convention that $\pi(\boldsymbol{x}_0|\boldsymbol{\phi}^{(k)}) = \pi(\boldsymbol{x}_0|\boldsymbol{x}_{0:0}, \boldsymbol{\phi}_x^{(k)})$.

   (b) Update and normalize the importance weights via

$$\tilde{\omega}_t^{(k)} = \tilde{\omega}_{t-1}^{(k)} \pi(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1}, \boldsymbol{\phi}^{(k)}), \qquad \omega_t^{(k)} = \frac{\tilde{\omega}_t^{(k)}}{\sum_{j=1}^{N} \tilde{\omega}_t^{(j)}}.$$

   (c) Update the observed data likelihood via

$$\pi(\boldsymbol{x}_{0:t}|\boldsymbol{\phi}^{(k)}) = \pi(\boldsymbol{x}_{0:t-1}|\boldsymbol{\phi}^{(k)}) \pi(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1}, \boldsymbol{\phi}^{(k)}).$$

3. If ESS $< \delta N$ resample and move. For $k = 1,\dots,N$:

   (a) Sample indices $a_k \sim \mathcal{M}(\boldsymbol{\omega}^{1:N})$ and set $\{\boldsymbol{\phi}^{(k)}, \tilde{\omega}_t^{(k)}\} := \{\boldsymbol{\phi}^{(a_k)}, 1\}$ and $\pi(\boldsymbol{x}_{0:t}|\boldsymbol{\phi}^{(k)}) := \pi(\boldsymbol{x}_{0:t}|\boldsymbol{\phi}^{(a_k)})$, $\boldsymbol{m}_t^{(k)} := \boldsymbol{m}_t^{(a_k)}$ and $\boldsymbol{C}_t^{(k)} := \boldsymbol{C}_t^{(a_k)}$.

   (b) Propose $\boldsymbol{\phi}^* \sim q(\cdot|\boldsymbol{\phi}^{(k)}, u_{t-1}^{(k)}, j_{t-1}^{(k)})$. Perform iterations $1,\dots,k$ of the forward filter to obtain $\pi(\boldsymbol{x}_{0:t}|\boldsymbol{\phi}^*)$, $\boldsymbol{m}_t^*$ and $\boldsymbol{C}_t^*$. With probability

$$\min\left\{1, \frac{\pi(\boldsymbol{\phi}^*)\pi(\boldsymbol{x}_{0:t}|\boldsymbol{\phi}^*)}{\pi(\boldsymbol{\phi}^{(k)})\pi(\boldsymbol{x}_{0:t}|\boldsymbol{\phi}^{(k)})} \times \frac{q(\boldsymbol{\phi}^{(k)}|\boldsymbol{\phi}^*)}{q(\boldsymbol{\phi}^*|\boldsymbol{\phi}^{(k)})}\right\},$$

   put $\boldsymbol{\phi}^{(k)} := \boldsymbol{\phi}^*$, and $\pi(\boldsymbol{x}_{0:t}|\boldsymbol{\phi}^{(k)}) := \pi(\boldsymbol{x}_{0:t}|\boldsymbol{\phi}^*)$, $\boldsymbol{m}_t^{(k)} := \boldsymbol{m}_t^*$ and $\boldsymbol{C}_t^{(k)} := \boldsymbol{C}_t^*$.

   (c) Sample indices $b_k \sim \mathcal{M}(f(\tilde{\Lambda})^{1:N})$ and set $\{u_t^{(k)}, j_t^{(k)}\} := \{u_{t-1}^{(b_k)}, j_{t-1}^{(b_k)}\}$.

Thus in order to evaluate (3.11), we only need to evaluate the observed data likelihood contribution from the beginning of the current window until the current time. Furthermore, by taking the proposal density to be $q(\boldsymbol{\phi}^*|\boldsymbol{\phi}) = \tilde{\pi}(\boldsymbol{\phi}^*|\boldsymbol{x}_{1:(s-1)T})$, the kernel density estimate need not be evaluated in the MH acceptance ratio. The choice of the window width has a direct influence on computational efficiency and posterior accuracy.

Note that with a smaller width for each of the windows, the number of rejuvenation steps in a fixed period becomes more due to the nature of IBIS (which is that the variance of the weights are always larger at the beginning of the process). Therefore it is not obviously beneficial to

Algorithm 13 Online IBIS scheme

1. Initialisation. Divide the observed period into b windows, $s \in \{1,\dots,b\}$. Denote by t_i^s the ith observation time in window s, $i = 1,\dots,n_s$. For $s = 1$, implement the IBIS scheme (Algorithm 11). For $s = 2,\dots,b$ and $i = 1,\dots,n_s$:

2. Sequential importance sampling. For $k = 1,\dots,N$:

   (a) Perform iteration i (corresponding to time t_i^s) of the forward filter to obtain $\pi(\boldsymbol{x}_{t_i^s}|\boldsymbol{x}_{1:t_{i-1}^s},\boldsymbol{\phi}^{(k)})$, $\boldsymbol{m}_{t_i^s}^{(k)}$ and $\boldsymbol{C}_{t_i^s}^{(k)}$.

   (b) Update and normalise the importance weights via

$$\tilde{\omega}_{t_i^s}^{(k)} = \tilde{\omega}_{t_{i-1}^s}^{(k)} \pi(\boldsymbol{x}_{t_i^s}|\boldsymbol{x}_{1:t_{i-1}^s},\boldsymbol{\phi}^{(k)}), \qquad \omega_{t_i^s}^{(k)} = \frac{\tilde{\omega}_{t_i^s}^{(k)}}{\sum_{z=1}^{N} \tilde{\omega}_{t_i^s}^{(z)}}.$$

   (c) Update the observed data likelihood contribution in the current window via

$$\pi(\boldsymbol{x}_{t_1^s:t_i^s}|\boldsymbol{x}_{1:(s-1)T},\boldsymbol{\phi}^{(k)}) = \pi(\boldsymbol{x}_{t_1^s:t_{i-1}^s}|\boldsymbol{x}_{1:(s-1)T},\boldsymbol{\phi}^{(k)})\pi(\boldsymbol{x}_{t_i^s}|\boldsymbol{x}_{1:t_{i-1}^s},\boldsymbol{\phi}^{(k)}),$$

   with the convention that $\pi(\boldsymbol{x}_{t_1^s:t_i^s}|\boldsymbol{x}_{1:(s-1)T},\boldsymbol{\phi}^{(k)}) = \pi(\boldsymbol{x}_{t_1^s}|\boldsymbol{x}_{1:(s-1)T},\boldsymbol{\phi}^{(k)})$ for $i = 1$.

3. If ESS $< \delta N$ resample and move. For $k = 1,\dots,N$:

   (a) Sample indices $a_k \sim \mathcal{M}(\boldsymbol{\omega}^{1:N})$ and set $\{\boldsymbol{\phi}^{(k)},\tilde{\omega}_{t_i^s}^{(k)}\} := \{\boldsymbol{\phi}^{(a_k)},1\}$, $\boldsymbol{m}_{t_i^s}^{(k)} := \boldsymbol{m}_{t_i^s}^{(a_k)}$, $\boldsymbol{C}_{t_i^s}^{(k)} := \boldsymbol{C}_{t_i^s}^{(a_k)}$ and $\pi(\boldsymbol{x}_{t_1^s:t_i^s}|\boldsymbol{x}_{1:(s-1)T},\boldsymbol{\phi}^{(k)}) := \pi(\boldsymbol{x}_{t_1^s:t_i^s}|\boldsymbol{x}_{1:(s-1)T},\boldsymbol{\phi}^{(a_k)})$.

   (b) Propose $\boldsymbol{\phi}_x^* \sim \log N(\log(\boldsymbol{\phi}_x^{(k)}),h_s^2)$. Using $\boldsymbol{m}_{(s-1)T}^* = \boldsymbol{m}_{(s-1)T}^{(k)}$ and $\boldsymbol{C}_{(s-1)T}^* = \boldsymbol{C}_{(s-1)T}^{(k)}$, perform iterations $1,\dots,i$ (corresponding to times $t_1^s,\dots,t_i^s$) of the forward filter to obtain $\pi(\boldsymbol{x}_{t_1^s:t_i^s}|\boldsymbol{x}_{1:(s-1)T},\boldsymbol{\phi}^*)$. With probability

$$\min\left\{1, \frac{\pi(\boldsymbol{x}_{t_1^s:t_i^s}|\boldsymbol{x}_{1:(s-1)T},\boldsymbol{\phi}^*)}{\pi(\boldsymbol{x}_{t_1^s:t_i^s}|\boldsymbol{x}_{1:(s-1)T},\boldsymbol{\phi}^{(k)})}\right\},$$

   put $\boldsymbol{\phi}^{(k)} := \boldsymbol{\phi}^*$, $\pi(\boldsymbol{x}_{t_1^s:t_i^s}|\boldsymbol{x}_{1:(s-1)T},\boldsymbol{\phi}^{(k)}) := \pi(\boldsymbol{x}_{t_1^s:t_i^s}|\boldsymbol{x}_{1:(s-1)T},\boldsymbol{\phi}^*)$, $\boldsymbol{m}_{t_i^s}^{(k)} := \boldsymbol{m}_{t_i^s}^*$ and $\boldsymbol{C}_{t_i^s}^{(k)} := \boldsymbol{C}_{t_i^s}^*$.

apply online IBIS with a simple model even the observation period is large, as each rejuvenation step will be relatively easy to implement. A simulation study comparing IBIS and online IBIS for different window lengths on a complicated spatial DLM is given in 6.1.2. The online IBIS scheme is summarised by Algorithm 13.

Chapter 4

Simulation studies

In this chapter, we investigate the performance of various SMC schemes by fitting a local level model and a sinusoidal form DLM to synthetic data sets generated from each model. Recall the general structure of the DLM with a univariate observation equation given by

$$X_t = \boldsymbol{F}_t\boldsymbol{\theta}_t + v_t, \qquad v_t \overset{indep}{\sim} N(0,V),$$
$$\boldsymbol{\theta}_t = \boldsymbol{G}_t\boldsymbol{\theta}_{t-1} + \boldsymbol{w}_t, \qquad \boldsymbol{w}_t \overset{indep}{\sim} N(\mathbf{0}, \boldsymbol{W}).$$

The first synthetic data set is generated from the local level model with $\boldsymbol{F}_t = \boldsymbol{G}_t = 1$ and the true values for the parameters $V = 2$ and $\boldsymbol{W} = 1$. The initial state $\boldsymbol{\theta}_0$ is randomly drawn from a $N(10,9)$ distribution. For the second synthetic data set generated from the sinusoidal form DLM, we take with the initial state vector to be $\boldsymbol{\theta}_0 = (10,0,0)$ and the error variances as $V = 2$ and $\boldsymbol{W} = \text{diag}(W_1, W_2, W_3) = 2\mathbb{I}_3$. The observation matrix is $\boldsymbol{F}_t = (\cos(\pi t/12), \sin(\pi t/12), 1)$ and the system matrix is the three-dimensional identity matrix, that is, $\boldsymbol{G}_t = \mathbb{I}_3$. For each model, 200 observations were generated.

4.1 Comparison between fully adapted auxiliary particle filter and bootstrap particle filter

In this simulation study, a fully adapted auxiliary particle filter (FA-APF) and bootstrap particle filter (BPF) both with $N = 1000$ particles are applied to the simulated data set generated from the

local level model. We compare the state posterior estimated by both schemes with the MCMC output (the MH algorithm is applied here). The left plot in Figure 4.1 shows the posterior mean of θ_t over time, based on the output of FA-APF, BPF and MCMC, against the true values of the state used for generating the simulated data. As expected, the mean output via FA-APF and BPF is consistent with the output of the MCMC scheme since they all target the same posterior. Also the posterior mean is consistent with the true θ_t (circle points) over time, demonstrating accurate estimation of the state values. Furthermore, the right plot in Figure 4.1 shows the difference of the posterior mean values obtained by BPF against MCMC, and FA-APF against MCMC, with the corresponding 95% credible intervals. We can see both mean difference curves move around zero over time which shows no systematic difference between the particle filtering estimates and the MCMC estimates for the posterior of θ_t. Moreover the values obtained from the output of FA-APF against MCMC are less variable than that of BPF. For this data set, FA-APF performs better than BPF in dealing with the state estimation due to its improvement on the issues of degeneracy and sample impoverishment. In Figure 4.2, we compare the posterior distributions obtained through FA-APF and BPF with the MCMC output at time points $t = 1, \ldots, 6$ respectively. We see that the kernel density estimates of the particles obtained from the output of both particle filter schemes are consistent with the output from the MCMC scheme.

4.2 Comparison between the Liu-West algorithm, the Storvik algorithm, particle learning, IBIS and aIBIS

4.2.1 Local level model

To illustrate the performance of the various SMC schemes introduced in the previous chapter that are applicable in the context of unknown parameters, and to assess their accuracy and efficiency, we first fit the local level model to the simulated data set. We conduct 100 runs for each of the schemes with the particle size $N = 3 \times 10^3$, 5×10^3 and 10^4. The inverse gamma distribution $IG(1, 1)$ is taken as the independent prior for both unkown parameters V and W. For the Liu-West algorithm, we find that in practice, the approximation results are very sensitive to the chosen value of the shrinkage factor κ. Therefore we choose the largest value of 0.995 for κ, following the tuning advice in Liu and West (2001). We choose an ESS threshold of $\delta = 0.5$ for both IBIS and aIBIS schemes, which means, when the ESS drops below half the number of particles, the resample-move step will be triggered. The random walk proposal densities of (3.8) and (3.9) are taken for the IBIS and aIBIS scheme respectively for particle rejuvenation. Figure 4.3 presents

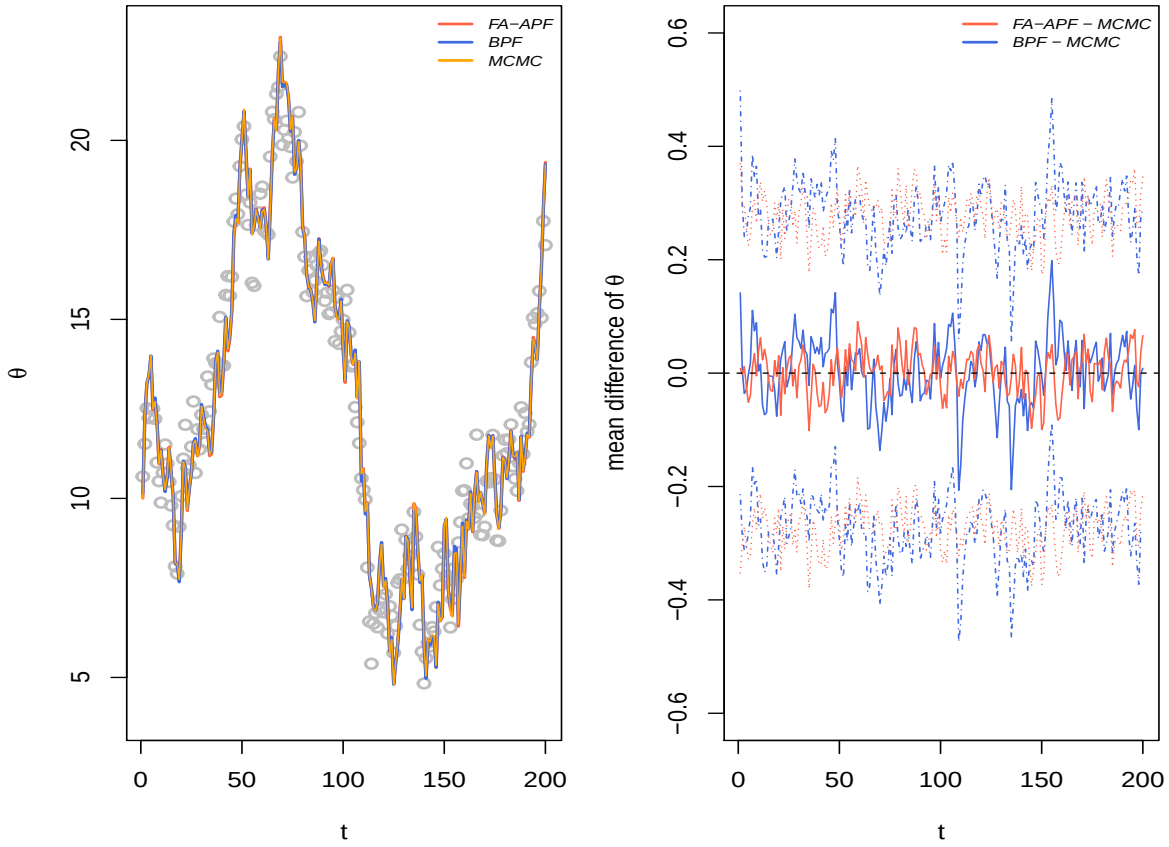

Figure 4.1: Left plot: comparison of the posterior means of the state through FA-APF, BPF and MCMC over time. The true values of the state are indicated by the grey circles. Right plot: Comparison of the difference of posterior means with 95% credible intervals through FA-APF against MCMC and BPF against MCMC.

the sequential posterior means of V and W with the 95% credible intervals over time by applying different SMC schemes from randomly selected runs corresponding to the particle choices above. In general, the posterior means of the static parameters approach the true values and posterior uncertainty reduces, as more data are observed. The output of the Liu-West algorithm appears to be inconsistent with that of the other filters. In Figure 4.4, we compare the posterior distributions obtained by the different SMC schemes and different numbers of particles given all the data (at $t = 200$) with the posterior distribution from the MCMC output (10^5 iterations). For the particle size $N = 3 \times 10^3$, the output from the Liu-West algorithm, the Storvik algorithm and particle learning (PL) exhibit distinct inconsistencies compared to the MCMC output, especially for the system variance W. However, satisfactory accuracy is obtained by applying IBIS and aIBIS, where the posterior distributions obtained by both schemes are very similar to the MCMC output. As the particle size increases to $N = 5 \times 10^3$, the posterior distribution obtained by PL is in reasonable agreement with the MCMC output, but the results from the Liu-West algorithm and the Storvik algorithm still have obvious differences. Most schemes perform well when the

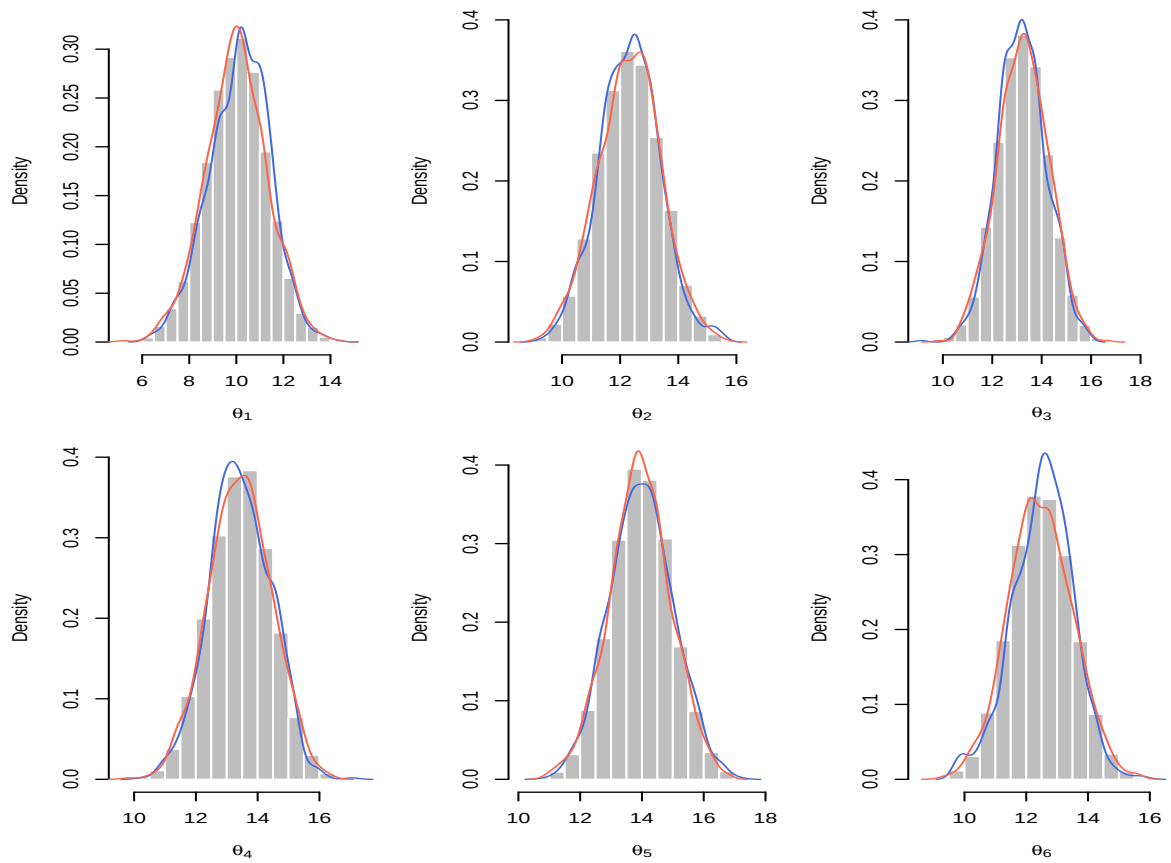

Figure 4.2: Comparison of the posterior distribution of θ via BPF (blue) and FA-APF (red) and the MCMC output (histograms) at $t = 1, \ldots, 6$ respectively.

particle size increases to $N = 10^4$, except the Liu-West algorithm, where a clear inconsistency still appears for the posterior distribution of W. When comparing the Liu-West algorithm with the other algorithms, it has the least accuracy even when using a carefully selected shrinkage ratio. The Storvik algorithm and PL are accurate if the particle size is large enough, but in practice, such methods may face the requirement of massive particle numbers which is likely to be a challenge for computer systems. IBIS and aIBIS demonstrate the best performances in terms of the required number of particles to obtain reasonable posterior accuracy. They manage to alleviate particle degeneracy and maintain a satisfactory level of the informative particle size (known as effective sample size) through the particle rejuvenation steps.

A more formal comparison of the different SMC schemes can be achieved by performing multiple independent runs of each algorithm. Table 4.1 summarises the average computational costs and the bias and root-mean-square error (RMSE) of estimators of the marginal posterior expectations and standard deviations of V and W by comparing the output of each SMC scheme (100 repeated times) and that of a long run with 10^5 iterations of MCMC. The bias and RMSE of estimators of

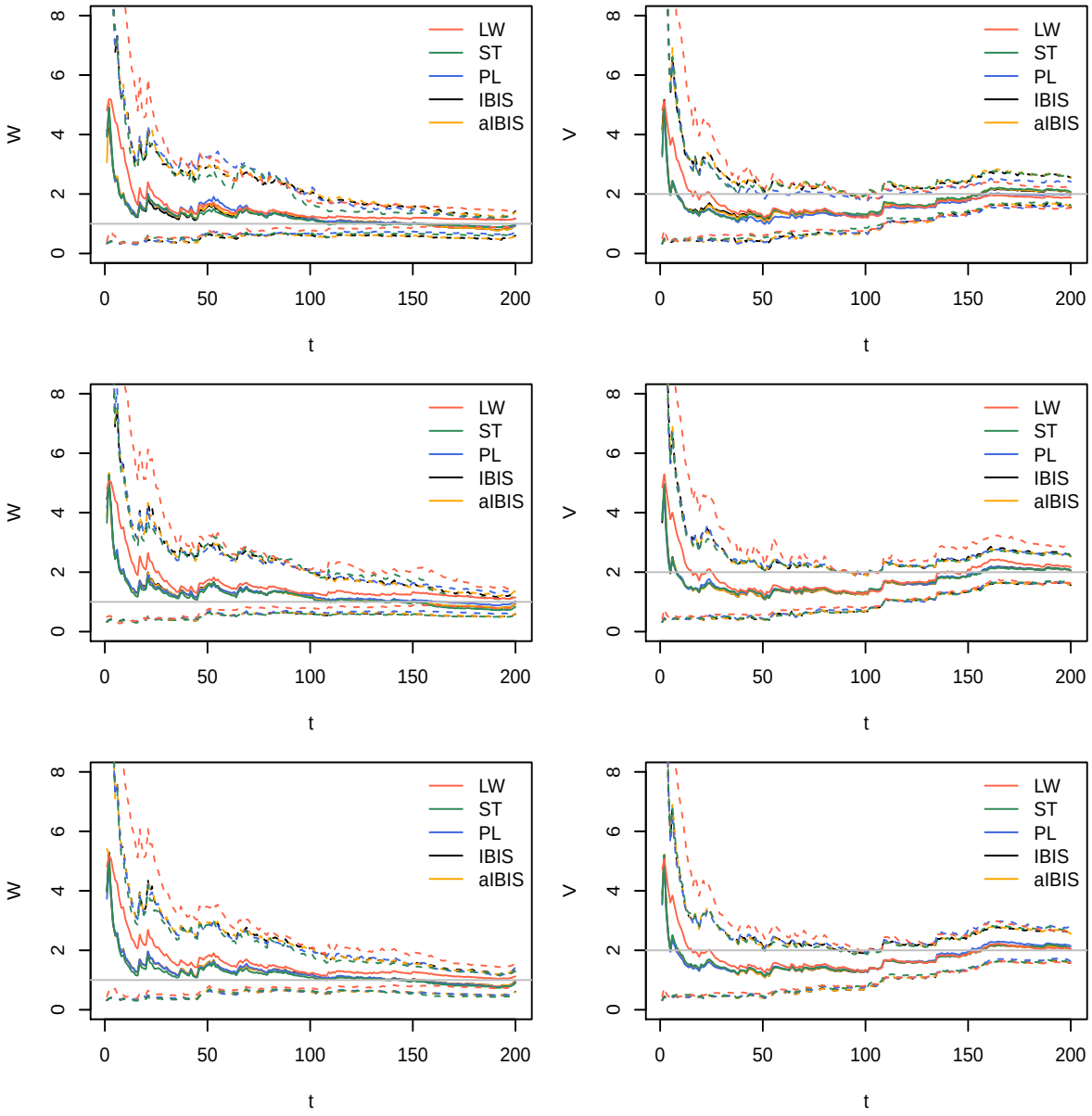

Figure 4.3: Sequential posterior means with 95% credible intervals of W and V over time, calculated from the output of different SMC schemes. Top row: 3×10^3 particles; middle row: 5×10^3 particles; bottom row: 10^4 particles. The true parameter values are indicated by the horizontal grey lines.

the marginal posterior expectations can be calculated using

$$\text{bias} = \frac{1}{n_{run}} \sum_{i=1}^{n_{run}} \left[E_i^{\text{SMC}}(\cdot|\boldsymbol{x}_{1:n}) - E^{\text{MCMC}}(\cdot|\boldsymbol{x}_{1:n}) \right]$$

$$\text{RMSE} = \left\{ \frac{1}{n_{run}} \sum_{i=1}^{n_{run}} \left[E_i^{\text{SMC}}(\cdot|\boldsymbol{x}_{1:n}) - E^{\text{MCMC}}(\cdot|\boldsymbol{x}_{1:n}) \right]^2 \right\}^{1/2}$$

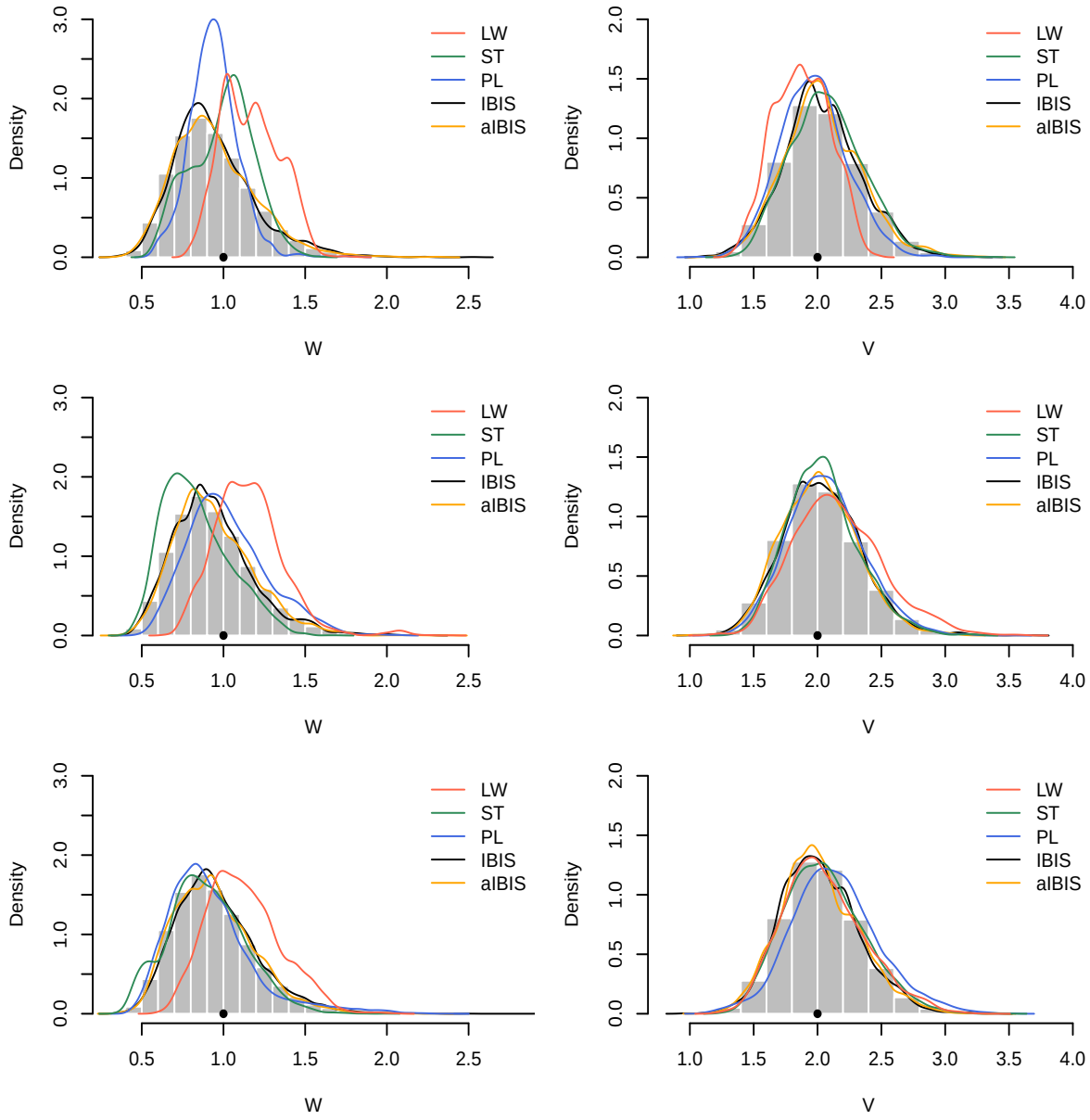

Figure 4.4: Comparison of the posterior distributions of W and V through different SMC schemes given all the data with the MCMC output. Top row: 3×10^3 particles; middle row: 5×10^3 particles; bottom row: 10^4 particles. The true parameter values are indicated by the solid circles.

where n_{run} equals the number of independent runs. Similarly we can update the bias and RMSE of estimators of the posterior standard deviations.

In terms of the computational performance, the Liu-West algorithm, the Storvik algorithm and PL give generally comparable results and their average costs are around twice as fast as IBIS. Unsurprisingly, aIBIS is the most time-consuming scheme of the five due to the extra computation needed to estimate the tuning parameter in the resample-move step. For the bias

of the estimators of the posterior expections and standard deviations, the values generated by IBIS and aIBIS are much smaller than the other three schemes in general, which demonstrate more accurate results are obtained by IBIS and aIBIS. Regarding the inspection of RMSE of the posterior means and standard deviations for the parameters, as the particle number increases, the RMSE values decrease for all schemes, i.e. the posterior density estimated by each SMC scheme and each run gets closer to the reference posterior obtained by MCMC. Based on the results, the Liu-West algorithm performs worst in terms of model accuracy, although it shows better computational efficiency than other schemes. PL has a slightly better performance than the Storvik algorithm because of the avoidance of the blind propagation issue. However, these two schemes are fragile when the particle size is small. The Liu-West algorithm, the Storvik algorithm and PL exhibit much higher RMSE values than IBIS and aIBIS with different sizes of particles. Therefore, even when the computational cost is taken into account, IBIS and aIBIS demonstrate better performance than the others. Generally speaking, the difference between the output obtained by IBIS and aIBIS is negligible, although aIBIS gives somewhat higher accuracy than IBIS due to the adptive optimal choice of the tuning parameter. Considering the computational costs, running IBIS is nearly twice as fast as aIBIS, which supports the better practicality of IBIS for large data sets.

4.2.2 Sinusoidal form DLM

In this section, we fit the sinusoidal form DLM (see Section 2.2.1) to the simulated data set described in Section 2.5.2 and apply various SMC schemes to estimate the unknown parameters by using different sizes of particles: $N = 3 \times 10^3$, 5×10^3 and 10^4. We take independent inverse gamma $IG(1,1)$ distributions *a priori* for the parameters V, W_1, W_2 and W_3. For the Liu-West algorithm, we choose 0.995 as the optimal value of the shrinkage factor κ (Liu and West, 2001). For the IBIS and aIBIS scheme, the ESS threshold is taken as $\delta = 0.5$, and we choose the random walk proposal densities of (3.8) and (3.9) for two schemes respectively. Figures 4.5 - 4.7 present the means and 95% credible intervals of the marginal posterior distributions of parameters W_1, W_2, W_3 and V updated sequentially by each SMC scheme over time. We can see that the Liu-West algorithm, the Storvik algorithm and PL are relatively unstable, especially for the credible intervals, even when the particle size is increased to 10^4. IBIS and aIBIS exhibit similar and more stable posterior summaries over time for different sizes of particles. Figures 4.8 - 4.10 show the comparison between the marginal posterior distributions of all four parameters at $t = 200$ through different SMC schemes with different particle sizes and the MCMC output obtained by a long run (10^5 iterations). Clearly the Liu-West algorithm, the Storvik algorithm

	N	CPU (s)	bias (RMSE)			
			$\widehat{E}(W\|\boldsymbol{x}_{1:n})$	$\widehat{SD}(W\|\boldsymbol{x}_{1:n})$	$\widehat{E}(V\|\boldsymbol{x}_{1:n})$	$\widehat{SD}(V\|\boldsymbol{x}_{1:n})$
LW	3k	0.33	0.2273 (0.2678)	0.0128 (0.0410)	0.0268 (0.1386)	0.0246 (0.0768)
	5k	0.58	0.2106 (0.2240)	0.0168 (0.0383)	0.0462 (0.1024)	0.0252 (0.0554)
	10k	1.13	0.1856 (0.1923)	0.0198 (0.0388)	0.0796 (0.1096)	0.0390 (0.0555)
ST	3k	0.34	0.0004 (0.1030)	-0.0310 (0.0571)	-0.0334 (0.1199)	-0.0199 (0.0402)
	5k	0.57	0.0045 (0.0837)	-0.0128 (0.0469)	-0.0063 (0.0745)	-0.0111 (0.0362)
	10k	1.15	0.0138 (0.0570)	-0.0057 (0.0353)	-0.0236 (0.0577)	-0.0041 (0.0275)
PL	3k	0.38	0.0066 (0.0844)	-0.0241 (0.0528)	-0.0096 (0.0754)	-0.0136 (0.0355)
	5k	0.66	0.0103 (0.0638)	-0.0108 (0.0421)	-0.0130 (0.0632)	-0.0087 (0.0310)
	10k	1.22	0.0126 (0.0520)	0.0004 (0.0274)	-0.0166 (0.0477)	-0.0038 (0.0201)
IBIS	3k	0.70	0.0001 (0.0104)	-0.0006 (0.0059)	0.0032 (0.0129)	0.0010 (0.0086)
	5k	1.19	0.0001 (0.0080)	0.0003 (0.0052)	-0.0072 (0.0118)	-0.0008 (0.0067)
	10k	2.16	0.0011 (0.0045)	-0.0010 (0.0036)	0.0005 (0.0064)	0.0006 (0.0044)
aIBIS	3k	1.29	0.0022 (0.0093)	-0.0005 (0.0068)	0.0036 (0.0112)	-0.0006 (0.0081)
	5k	2.18	0.0015 (0.0067)	0.0005 (0.0060)	-0.0064 (0.0115)	-0.0002 (0.0053)
	10k	4.04	0.0001 (0.0054)	-0.0010 (0.0038)	-0.0006 (0.0063)	0.0007 (0.0045)

Table 4.1: Comparison of the performance by LW, ST, PL, IBIS, aIBIS: CPU time (in seconds); bias (and RMSE in parentheses) of estimators of the posterior expectations $\widehat{E}(W|\boldsymbol{x}_{1:n})$, $\widehat{E}(V|\boldsymbol{x}_{1:n})$ and standard deviations $\widehat{SD}(W|\boldsymbol{x}_{1:n})$, $\widehat{SD}(V|\boldsymbol{x}_{1:n})$. All results are obtained by averaging over 100 runs of each SMC scheme.

and PL all suffer from the particle degeneracy problem as their results are highly inconsistent with the MCMC output. IBIS and aIBIS have similar performance for different particle sizes, and the results match up with the MCMC output. To check the correctness further of the Liu-West algorithm, the Storvik algorithm and PL, we increase the particle size to 10^6. Figure 4.11 shows the corresponding result by each scheme comparing with the MCMC output. With a sufficient number of particles, the marginal posteriors obtained from by the Storvik algorithm and PL now match up with the MCMC output. However the Liu-West algorithm still experiences the particle degeneracy issue and it produces the marginal posterior distribution which are least similar to the MCMC output.

Table 4.2 and 4.3 summarises the performance results by calculating the bias and root-mean-square error (RMSE) of estimators of the posterior expectations and standard deviations for the comparison between each SMC scheme and the MCMC output obtained from a long run (10^5

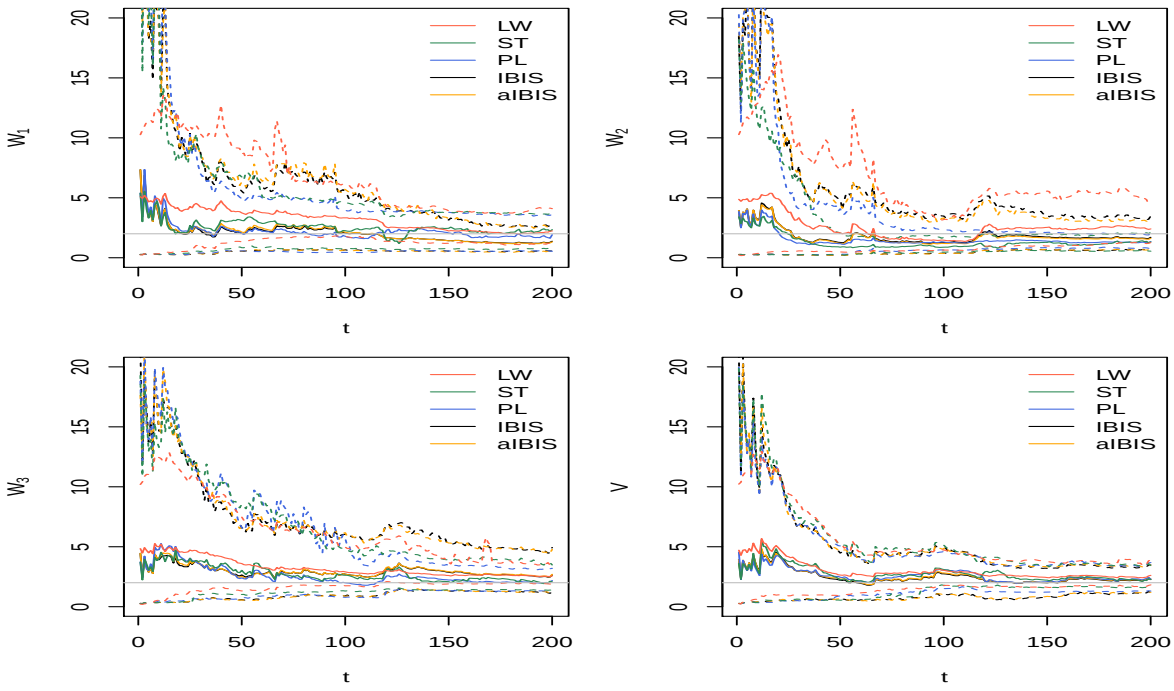

Figure 4.5: Sequential posterior means with 95% credible intervals of W_1, W_2, W_3 and V through different SMC schemes using 3×10^3 particles over time. The true parameter values are indicated by the horizontal grey lines.

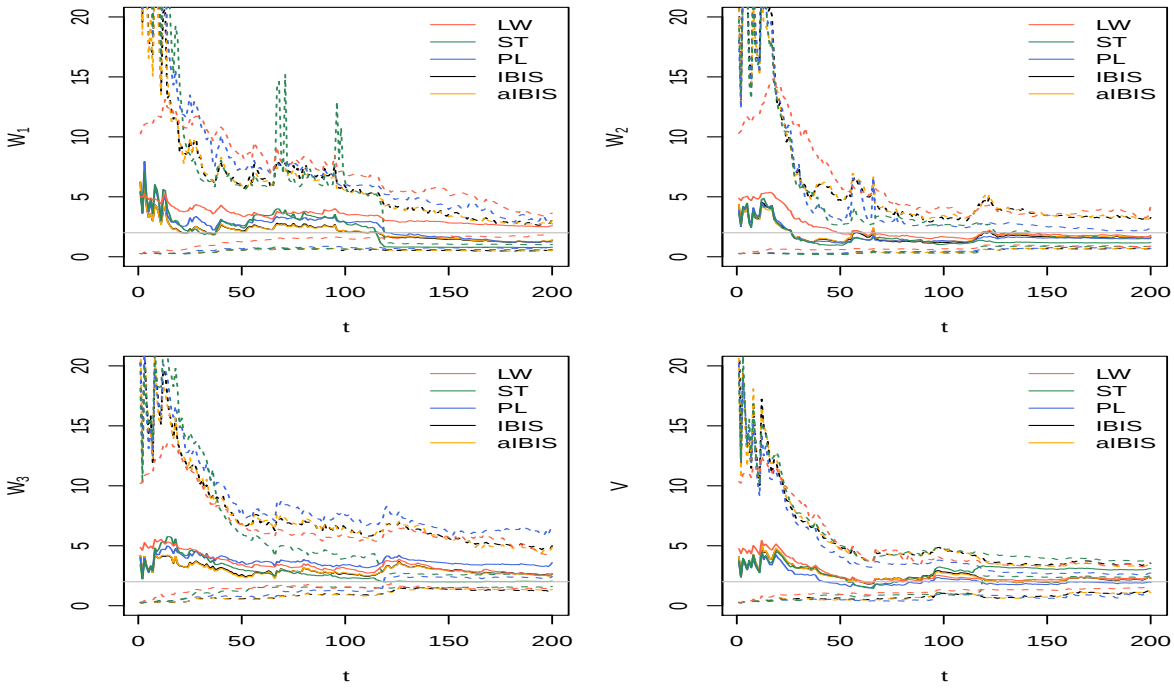

Figure 4.6: Sequential posterior means with 95% credible intervals of W_1, W_2, W_3 and V through different SMC schemes using 5×10^3 particles over time. The true parameter values are indicated by the horizontal grey lines.

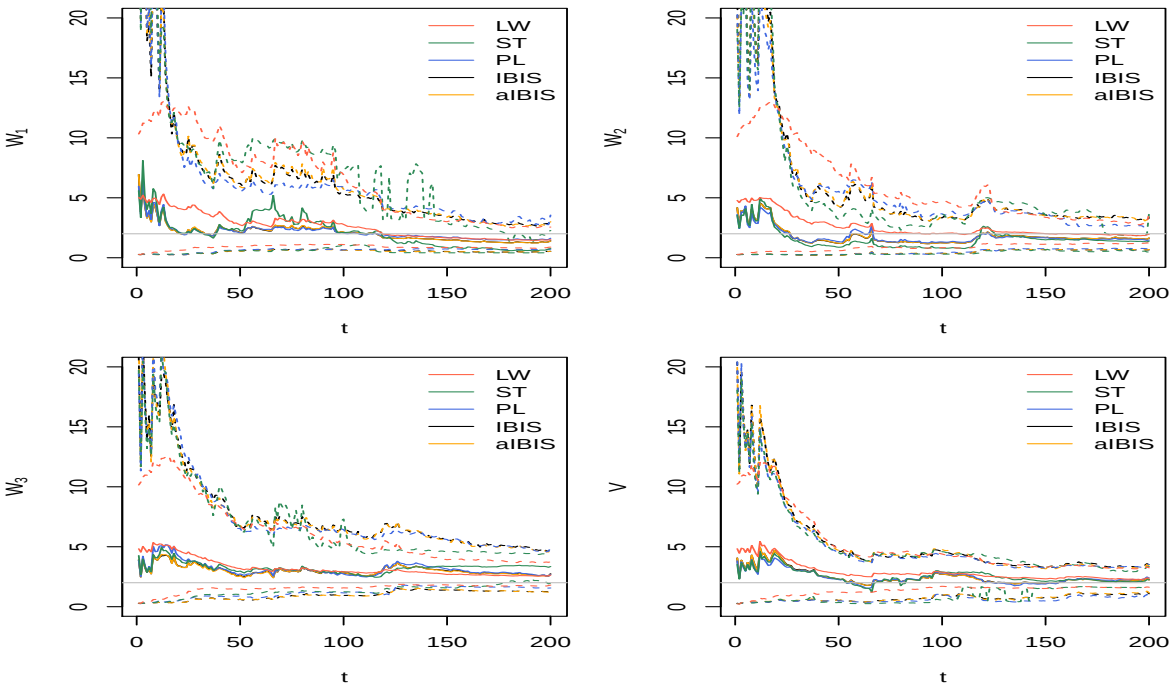

Figure 4.7: Sequential posterior means with 95% credible intervals of W_1, W_2, W_3 and V through different SMC schemes using 10^4 particles over time. The true parameter values are indicated by the horizontal grey lines.

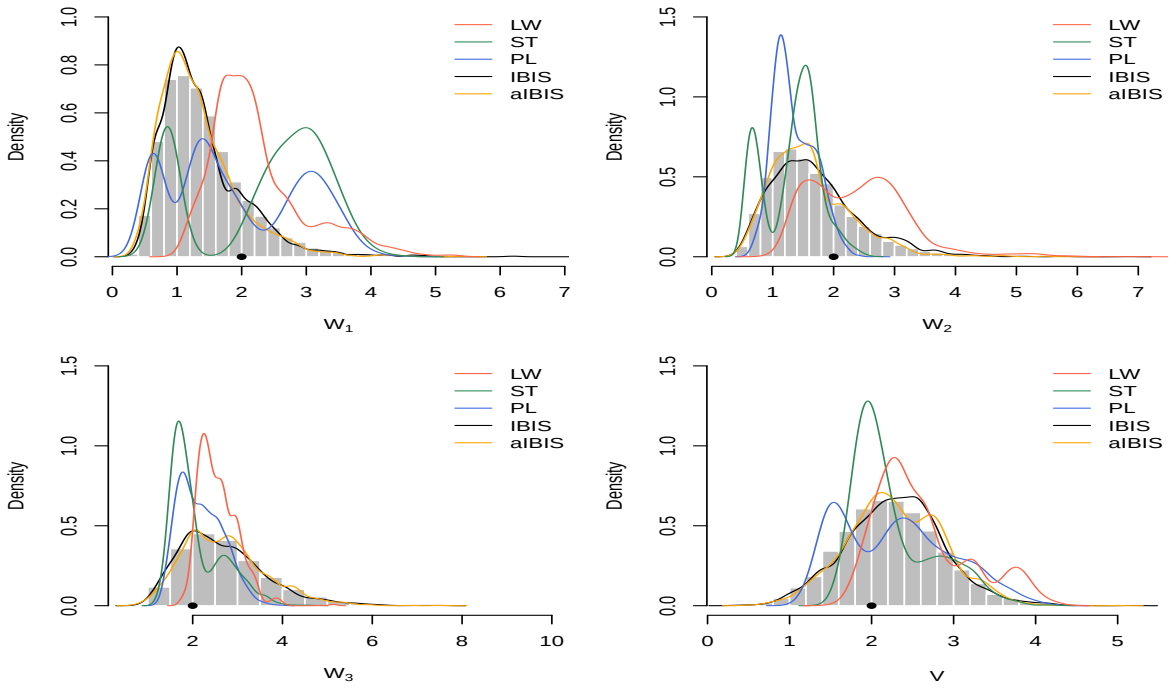

Figure 4.8: Comparison of the posterior distributions of W_1, W_2, W_3 and V through different SMC schemes using 3×10^3 particles given all the data with the MCMC output. The true parameter values are indicated by the solid circles.

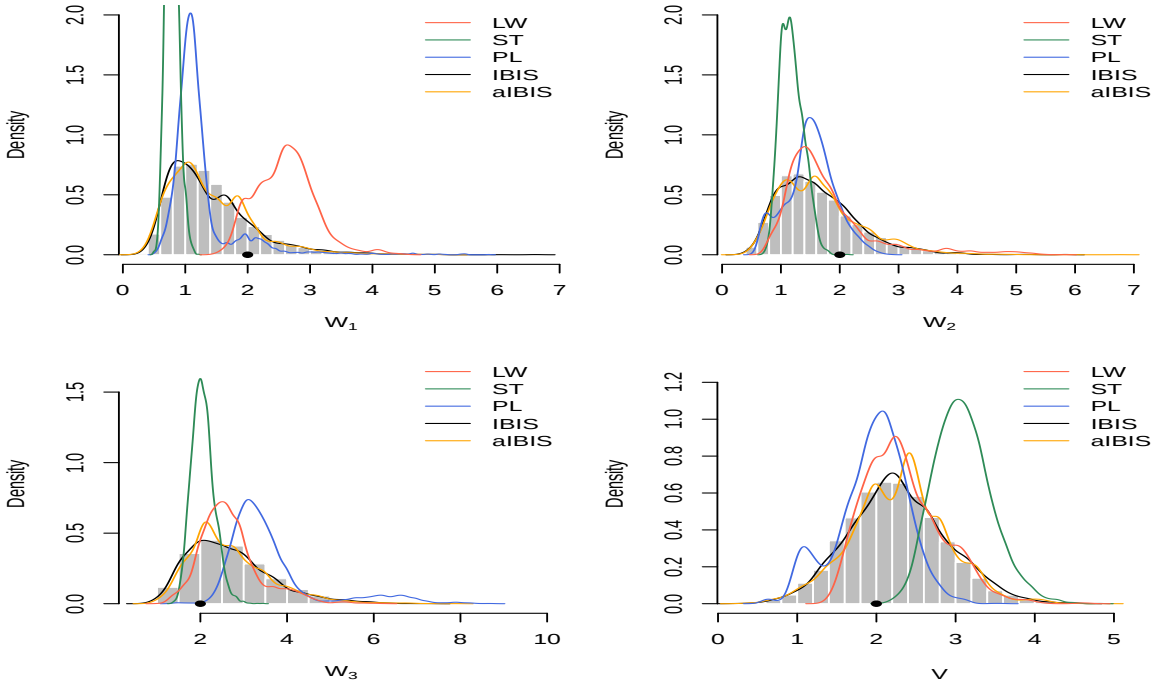

Figure 4.9: Comparison of the posterior distributions of W_1, W_2, W_3 and V through different SMC schemes using 5×10^3 particles given all the data with the MCMC output. The true parameter values are indicated by the solid circles.

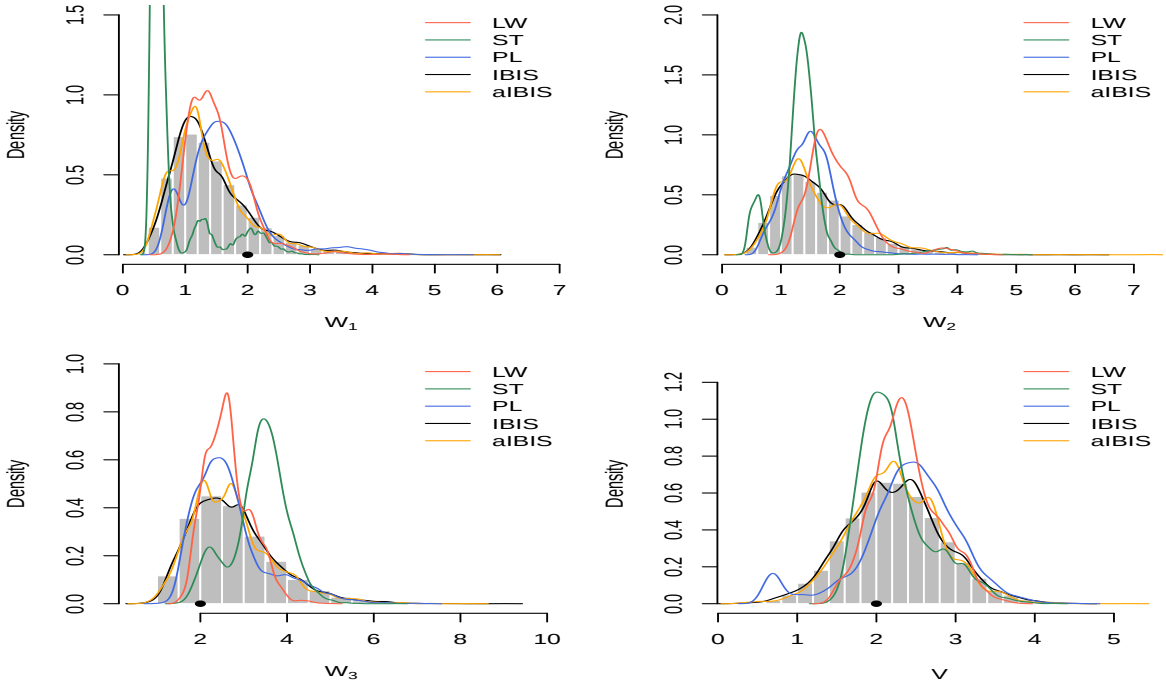

Figure 4.10: Comparison of the posterior distributions of W_1, W_2, W_3 and V through different SMC schemes using 10^4 particles given all the data with the MCMC output. The true parameter values are indicated by the solid circles.

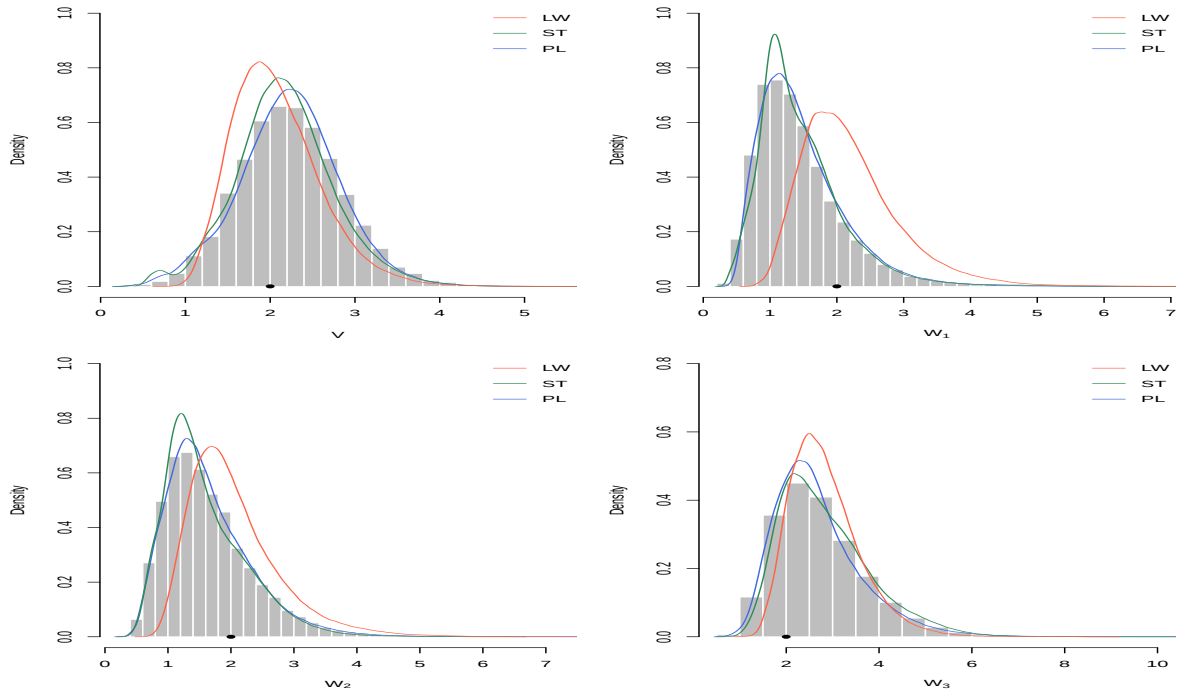

Figure 4.11: Comparison of the posterior distributions of W_1, W_2, W_3 and V through different SMC schemes using 10^6 particles given all the data with the MCMC output. The true parameter values are indicated by the solid circles.

N	CPU(s)				
	LW	ST	PL	IBIS	aIBIS
3k	41	14	42	91	171
5k	68	24	70	153	296
10k	132	46	137	308	610

Table 4.2: CPU time (in seconds) by averaging over 100 runs of LW, ST, PL, IBIS and aIBIS respectively.

iterations) and the average computational costs for various SMC schemes. Analogous to the previous results for the local level model, when the particle size increases, the absolute values of bias and RMSE gradually decrease due to the ease of particle degeneracy. Moreover, the results demonstrate further that the Liu-West algorithm, the Storvik algorithm and PL give a less accurate performance than IBIS and aIBIS when using the same number of particles. IBIS and aIBIS always have similar performance and increasing the particle size does not dramatically improve the performance for both of these schemes. Therefore $N = 3 \times 10^3$ particles is sufficient to estimate the posterior distribution accurately in this case when using IBIS and aIBIS. For the computational cost comparison, although the Liu-West algorithm, the Storvik algorithm and PL are much more efficient than IBIS and aIBIS, they are unable to provide accurate results

under such limited particle sizes. In fact, referring to the posterior distributions presented in Figure 4.11, the computational costs to run the Liu-West algorithm, the Storvik algorithm and PL with 10^6 particles are 3.7, 1.3, 3.8 hours respectively. Hence, these methods do not scale well as the number of data points or model complexity increases. For the efficiency comparison between IBIS and aIBIS, IBIS only takes half of the computational time of aIBIS. Therefore in real data applications, we will apply the standard IBIS scheme and assume a fixed tunning parameter for the resample-move step.

	N	bias (RMSE)							
		$\widehat{E}(W_1\|\boldsymbol{x}_{1:n})$	$\widehat{SD}(W_1\|\boldsymbol{x}_{1:n})$	$\widehat{E}(W_2\|\boldsymbol{x}_{1:n})$	$\widehat{SD}(W_2\|\boldsymbol{x}_{1:n})$	$\widehat{E}(W_3\|\boldsymbol{x}_{1:n})$	$\widehat{SD}(W_3\|\boldsymbol{x}_{1:n})$	$\widehat{E}(V\|\boldsymbol{x}_{1:n})$	$\widehat{SD}(V\|\boldsymbol{x}_{1:n})$
LW	3k	1.159 (1.276)	0.051 (0.242)	0.427 (0.617)	-0.078 (0.212)	0.225 (0.516)	-0.350 (0.394)	-0.179 (0.355)	-0.192 (0.221)
	5k	0.994 (1.044)	0.041 (0.188)	0.385 (0.522)	-0.063 (0.189)	0.119 (0.426)	-0.288 (0.325)	-0.116 (0.284)	-0.141 (0.174)
	10k	0.924 (0.966)	0.041 (0.130)	0.434 (0.524)	-0.010 (0.157)	0.092 (0.295)	-0.217 (0.255)	-0.144 (0.248)	-0.122 (0.145)
ST	3k	0.507 (0.817)	-0.208 (0.327)	0.060 (0.912)	-0.289 (0.414)	-0.091 (0.795)	-0.389 (0.507)	0.049 (0.462)	-0.213 (0.252)
	5k	0.208 (0.605)	-0.195 (0.340)	0.122 (0.593)	-0.230 (0.344)	0.071 (0.816)	-0.379 (0.448)	0.014 (0.480)	-0.204 (0.254)
	10k	0.139 (0.432)	-0.092 (0.258)	-0.022 (0.488)	-0.177 (0.262)	0.097 (0.633)	-0.107 (0.380)	-0.013 (0.332)	-0.061 (0.164)
PL	3k	0.193 (0.631)	-0.109 (0.292)	-0.066 (0.481)	-0.254 (0.331)	0.057 (0.744)	-0.259 (0.354)	0.006 (0.417)	-0.148 (0.209)
	5k	0.136 (0.428)	-0.021 (0.207)	0.016 (0.421)	-0.075 (0.310)	0.116 (0.566)	-0.099 (0.302)	-0.060 (0.340)	-0.060 (0.144)
	10k	0.079 (0.286)	-0.007 (0.178)	0.007 (0.269)	-0.089 (0.201)	-0.095 (0.333)	-0.103 (0.229)	0.046 (0.200)	-0.030 (0.125)
IBIS	3k	-0.032 (0.046)	0.017 (0.030)	0.037 (0.053)	-0.001 (0.026)	-0.045 (0.065)	-0.011 (0.031)	0.020 (0.032)	-0.017 (0.024)
	5k	-0.026 (0.035)	0.012 (0.023)	0.016 (0.032)	0.007 (0.022)	-0.023 (0.037)	-0.006 (0.024)	0.009 (0.022)	-0.013 (0.020)
	10k	-0.029 (0.035)	0.016 (0.021)	0.008 (0.023)	0.006 (0.015)	-0.027 (0.041)	0.001 (0.015)	0.014 (0.021)	-0.006 (0.011)
aIBIS	3k	-0.026 (0.038)	0.023 (0.032)	0.030 (0.046)	-0.007 (0.024)	-0.037 (0.059)	-0.011 (0.034)	0.015 (0.032)	-0.016 (0.024)
	5k	-0.020 (0.034)	0.018 (0.029)	0.025 (0.038)	0.011 (0.025)	-0.016 (0.040)	-0.003 (0.023)	-0.003 (0.026)	-0.018 (0.023)
	10k	-0.029 (0.037)	0.016 (0.023)	0.007 (0.021)	0.008 (0.018)	-0.024 (0.038)	0.002 (0.017)	0.014 (0.025)	-0.008 (0.012)

Table 4.3: Comparison of the performance by LW, ST, PL, IBIS, aIBIS: bias (and RMSE in parentheses) of estimators of the posterior expectations $\widehat{E}(W_1|\boldsymbol{x}_{1:n})$, $\widehat{E}(W_2|\boldsymbol{x}_{1:n})$, $\widehat{E}(W_3|\boldsymbol{x}_{1:n})$, $\widehat{E}(V|\boldsymbol{x}_{1:n})$ and standard deviations $\widehat{SD}(W_1|\boldsymbol{x}_{1:n})$, $\widehat{SD}(W_2|\boldsymbol{x}_{1:n})$, $\widehat{SD}(W_3|\boldsymbol{x}_{1:n})$, $\widehat{SD}(V|\boldsymbol{x}_{1:n})$. All results are obtained by averaging over 100 runs of each SMC scheme.

Chapter 5

Application to temperature and humidity data

5.1 Data collection

Recent advances in sensor technology and data management mean that it is now possible to reliably and affordably collect data on many aspects of city life. The temperature and relative humidity data analysed in this chapter were collected from the Urban Observatory (James et al., 2014), a big data hub providing smart-city data via a grid of sensors in North East England. The data are received in real time, and this requires efficient network transmission and data storage solutions. Temperature is measured in degrees Celsius and relative humidity is measured as the ratio of the amount of water vapour held in the air against the the maximum amount of water vapour the air can hold at a specific temperature. The data are captured and processed through a microprocessor inside a sensor and transmitted via a high speed network to the database (Galatioto et al., 2014). We consider data streams at five locations: Newcastle upon Tyne, Seaham, Peterlee, Whitley Bay and Consett. The observation period is from 8th July 2017 to 31st December 2017. Due to the different recording frequencies of some of the sensors, we take the average values of temperature and relative humidity over every consecutive hour, giving a total of 4239 time points at which at least one location has a measurement. Figure 5.1 shows the multiple data streams over time at different locations. Both temperature and relative humidity exhibit a clear sinusoidal pattern over each 24 hour period. Scatter plots of humidity against temperature for each location are shown in Figure 5.2 and reveal a strong negative linear correlation. Unfortunately, missing data are inevitable due to network disconnection or sensor

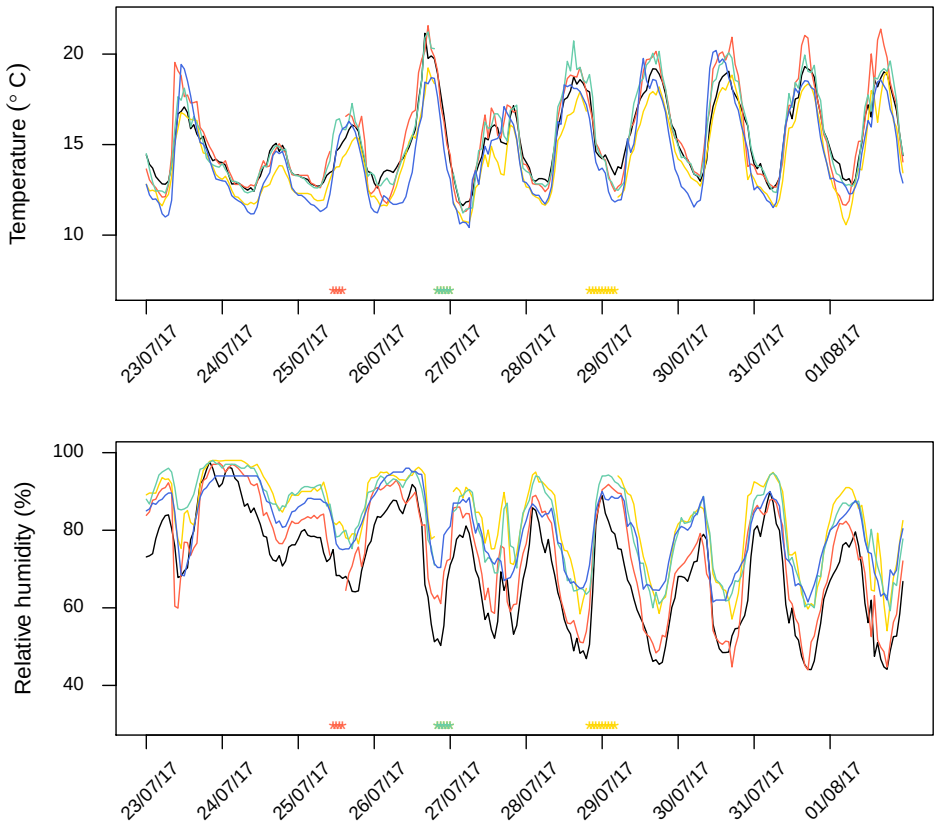

Figure 5.1: Temperature and relative humidity data streams over time at each location. Periods of missingness are indicated just above the x-axis.

failure. Figure 5.1 and Table 5.1 summarise and display the proportion of missing data at each location during the observation period.

5.2 Spatial DLMs

In Figure 5.1 we noted that the data show clear seasonality in both temperature and humidity measurements. This suggests that marginally each variable should be modelled by a sinusoidal form with a 24 hour period. In reality, the measurements are likely to be recorded irregularly over time. Therefore, we consider times as t_i, $i = 1, \ldots, n$ with $t_1 = 1$, and assume the initialisation of the state at $t_0 = 0$. In general we will consider data from ℓ locations but, for simplicity, we first consider data at a single location j. We propose a DLM for temperature with observation equation

$$X_{t_i}^j = \boldsymbol{F}_{t_i}^{x,j} \boldsymbol{\theta}_{t_i}^{x,j} + v_i^{x,j}, \qquad v_i^{x,j} \overset{indep}{\sim} N(0, V^{x,j}), \tag{5.1}$$

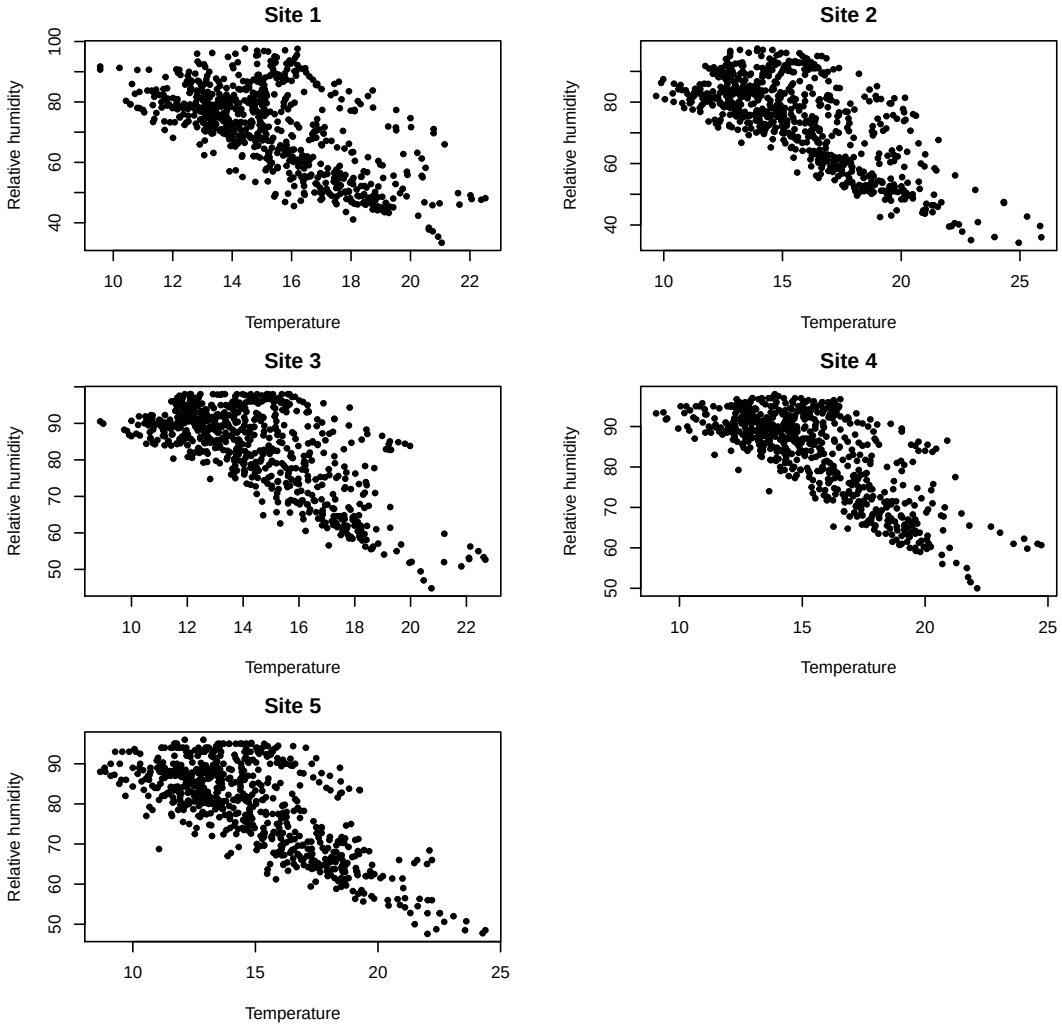

Figure 5.2: Scatter plots of temperature against relative humidity at each location.

Variable	Location	Missing	Prop.	Mean	Min.	25%	Median	75%	Max.
Temperature	Newcastle	392	9.25%	10.62	-9.10	6.70	11.70	14.88	27.53
(°C)	Seaham	54	1.27%	11.48	-2.17	8.12	12.30	15.07	25.90
	Peterlee	46	1.09%	10.49	-2.24	7.37	11.52	13.95	22.68
	Whitley Bay	6	0.14%	11.07	-4.62	7.72	12.10	14.73	24.73
	Consett	306	7.22%	10.40	-3.37	6.90	11.20	14.24	24.38
Humidity	Newcastle	392	9.25%	83.33	42.50	78.33	85.50	90.67	99.00
(%)	Seaham	54	1.27%	73.62	34.23	67.08	74.50	81.67	97.42
	Peterlee	46	1.09%	84.86	44.83	80.22	86.83	91.67	99.00
	Whitley Bay	6	0.14%	86.25	50.00	82.25	88.25	93.00	98.25
	Consett	306	7.22%	83.59	46.40	79.33	86.00	90.50	97.00

Table 5.1: A summary of hourly average temperature and humidity data over the period 8th July 2017 to 31st December 2017 at five locations in North East England.

where the observation matrix $\boldsymbol{F}_{t_i}^{x,j} = (\cos(\pi t_i/12), \sin(\pi t_i/12), 1)$ and $\boldsymbol{\theta}_{t_i}^{x,j} = (\theta_{t_i,1}^{x,j}, \theta_{t_i,2}^{x,j}, \theta_{t_i,3}^{x,j})^T$. Note that, after dropping the superscripts, the observation equation can be written as

$$X_{t_i} = \tilde{\theta}_{t_i,2} \cos\left(\frac{\pi t_i}{12} - \tilde{\theta}_{t_i,1}\right) + \theta_{t_i,3} + v_i \tag{5.2}$$

where the parameters in (5.1) and (5.2) are related using

$$\tilde{\theta}_{t_i,1} = \sqrt{\theta_{t_i,1}^2 + \theta_{t_i,2}^2}, \qquad \tilde{\theta}_{t_i,2} = \tan^{-1}\left(\frac{\theta_{t_i,2}}{\theta_{t_i,1}}\right). \tag{5.3}$$

We allow amplitude, phase shift and basal temperature to be time-varying, and take a system equation of the form

$$\boldsymbol{\theta}_{t_i}^{x,j} = \boldsymbol{G}_{t_i}^{x,j}\boldsymbol{\theta}_{t_{i-1}}^{x,j} + k_i \boldsymbol{w}_i^{x,j} + \boldsymbol{p}_i^{x,j}, \qquad \boldsymbol{w}_i^{x,j} \overset{indep}{\sim} N\left\{\boldsymbol{0}, \text{diag}(\boldsymbol{W}^{x,j})\right\} \tag{5.4}$$

where the system matrix $\boldsymbol{G}_{t_i}^{x,j} = \mathbb{I}_3$, the 3×3 identity matrix, and $\boldsymbol{W}^{x,j} = (W_1^{x,j}, W_2^{x,j}, W_3^{x,j})^T$. Note that including k_i, where $k_i^2 = t_i - t_{i-1}$, allows for measurements to be on an irregularly spaced temporal grid. Further the terms $\boldsymbol{p}_i^{x,j} = (p_{i,1}^{x,j}, p_{i,2}^{x,j}, p_{i,3}^{x,j})^T$ allow for spatial variability between amplitude, phase shift and basal temperature values at nearby locations. We model the components of the spatially smooth error process $\boldsymbol{p}_i^{x,j}$ using independent zero mean Gaussian process (GP) priors with covariance functions $f_m^x(\cdot), m = 1,2,3$, that is,

$$p_{i,m}^{x,j} \sim GP\{0, f_m^x(\cdot)\}, \quad m = 1,2,3.$$

We take these covariance functions to have a simple exponential form

$$f_m^x(d_{jj'}) = \text{Cov}(\theta_{t_i,m}^{x,j}, \theta_{t_i,m}^{x,j'}) = \sigma_{x,m}^2 \exp(-\psi_{x,m} d_{jj'}), \quad m = 1,2,3$$

and depend on parameters $\boldsymbol{\sigma}_x = (\sigma_{x,1}, \sigma_{x,2}, \sigma_{x,3})$ and $\boldsymbol{\psi}_x = (\psi_{x,1}, \psi_{x,2}, \psi_{x,3})$, with the latter determining the decay ratio of the correlation as the distance between two locations $d_{jj'}$ increases (Banerjee et al., 2014).

The full spatial DLM (over all locations) can be written as

$$\begin{aligned} \boldsymbol{X}_{t_i} &= \boldsymbol{F}_{t_i}^x \boldsymbol{\theta}_{t_i}^x + \boldsymbol{v}_i^x, \qquad \boldsymbol{v}_i^x \overset{indep}{\sim} N\{\boldsymbol{0}, \text{diag}(V^{x,1}, \ldots, V^{x,\ell})\}, \\ \boldsymbol{\theta}_{t_i}^x &= \boldsymbol{\theta}_{t_{i-1}}^x + k_i \boldsymbol{w}_i^x + \boldsymbol{p}_i^x, \qquad \boldsymbol{w}_i^x \overset{indep}{\sim} N\{\boldsymbol{0}, \text{diag}(\boldsymbol{W}^{x,1}, \ldots, \boldsymbol{W}^{x,\ell})\}, \end{aligned} \tag{5.5}$$

where

- $\boldsymbol{F}_{t_i}^x = \text{diag}(\boldsymbol{F}_{t_i}^{x,1}, \ldots, \boldsymbol{F}_{t_i}^{x,\ell})$;
- $\boldsymbol{\theta}_{t_i}^x = ((\boldsymbol{\theta}_{t_i}^{x,1})^T, \ldots, (\boldsymbol{\theta}_{t_i}^{x,\ell})^T)^T$;
- the 3ℓ-vector of spatial effects $\boldsymbol{p}_i^x = ((\boldsymbol{p}_i^{x,1})^T, \ldots, (\boldsymbol{p}_i^{x,\ell})^T)^T$ is normally distributed with zero mean and covariance matrix

$$\boldsymbol{K}^x = \begin{pmatrix} f^x(d_{11})\mathbb{I}_3 & \ldots & f^x(d_{1\ell})\mathbb{I}_3 \\ \vdots & \ddots & \vdots \\ f^x(d_{\ell 1})\mathbb{I}_3 & \ldots & f^x(d_{\ell\ell})\mathbb{I}_3 \end{pmatrix}. \tag{5.6}$$

5.2.1 Additional harmonics

As described in section 2.2.2, the Fourier form DLM can be an alternative form for modelling the seasonality. Therefore the full spatial Fourier form DLM (over all locations) can be written as

$$\begin{aligned} \boldsymbol{X}_{t_i} &= \boldsymbol{F}_{t_i}^x \boldsymbol{\theta}_{t_i}^x + \boldsymbol{v}_i^x, \qquad \boldsymbol{v}_i^x \overset{indep}{\sim} N\{\boldsymbol{0}, \text{diag}(V^{x,1}, \ldots, V^{x,\ell})\}, \\ \boldsymbol{\theta}_{t_i}^x &= \boldsymbol{G}_{t_i}^x \boldsymbol{\theta}_{t_{i-1}}^x + k_i \boldsymbol{w}_i^x + \boldsymbol{p}_i^x, \qquad \boldsymbol{w}_i^x \overset{indep}{\sim} N\{\boldsymbol{0}, \text{diag}(\boldsymbol{W}^{x,1}, \ldots, \boldsymbol{W}^{x,\ell})\}, \end{aligned}$$

where

- $\boldsymbol{F}_{t_i}^x = \text{diag}(\boldsymbol{F}_{t_i}^{x,1}, \ldots, \boldsymbol{F}_{t_i}^{x,\ell})$ with $\boldsymbol{F}_{t_i}^{x,j} = (1,0|1,0|\ldots|1)$, $j = 1, \ldots, \ell$;
- $\boldsymbol{G}_{t_i}^x = \text{diag}(\boldsymbol{G}_{t_i}^{x,1}, \ldots, \boldsymbol{G}_{t_i}^{x,\ell})$ with $\boldsymbol{G}_{t_i}^{x,j} = \text{diag}(\boldsymbol{H}_1, \ldots, \boldsymbol{H}_q, 1)$, $j = 1, \ldots, \ell$, and the harmonic matrice $\boldsymbol{H}_r$ has

$$\boldsymbol{H}_r = \begin{pmatrix} \cos(\pi r/12) & \sin(\pi r/12) \\ -\sin(\pi r/12) & \cos(\pi r/12) \end{pmatrix}, \quad r = 1, \ldots, q;$$

- $\boldsymbol{\theta}_{t_i}^x = ((\boldsymbol{\theta}_{t_i}^{x,1})^T, \ldots, (\boldsymbol{\theta}_{t_i}^{x,\ell})^T)^T$;
- $\boldsymbol{p}_i^x = ((\boldsymbol{p}_i^{x,1})^T, \ldots, (\boldsymbol{p}_i^{x,\ell})^T)^T$ has a normal distribution with zero mean and covariance matrix given by $\boldsymbol{K}^x$ in (5.6).

Note that for the full spatial Fourier form DLM, specifying q harmonics will give $2\ell(q+1)+6$ static parameters to be inferred and in practice $q = 1$ or 2 are typically used (Petris et al., 2009).

For the $q = 1$ harmonic and the trivial case of $\boldsymbol{W}^{x,j} = \mathbf{0}$, the observation equation of the Fourier form DLM coincides with that the sinusoidal form in (5.1) given by

$$X_{t_i}^j = \theta_{0,1}^{x,j} \cos(\pi t_i/12) + \theta_{0,2}^{x,j} \sin(\pi t_i/12) + \theta_{0,3}^{x,j} + v_i^{x,j}.$$

However, when $\boldsymbol{W}^{x,j} \neq \mathbf{0}$, the error structures differ due to the use of the harmonic in the system equation of the Fourier form DLM, and in the observation equation for the sinusoidal form DLM. The task of choosing between competing models is considered in Section 5.5.

5.2.2 Spatial humidity DLM

Due to the strong linear relationship between temperature and humidity, we specify a conditional DLM for humidity by regressing on temperature linearly in the observation equation. For a particular location j, the DLM takes the form

$$\begin{aligned} Y_{t_i}^j &= \boldsymbol{F}_{t_i}^{y,j} \boldsymbol{\theta}_{t_i}^{y,j} + v_i^{y,j}, & v_i^{y,j} &\overset{indep}{\sim} N\left(0, V^{y,j}\right) \\ \boldsymbol{\theta}_{t_i}^{y,j} &= \boldsymbol{\theta}_{t_{i-1}}^{y,j} + k_i \boldsymbol{w}_i^{y,j} + \boldsymbol{p}_i^{y,j}, & \boldsymbol{w}_i^{y,j} &\overset{indep}{\sim} N\{\mathbf{0}, \mathrm{diag}(\boldsymbol{W}^{y,j})\} \end{aligned}$$

where $\boldsymbol{F}_{t_i}^{y,j} = (X_{t_i}^j, 1)$, $\boldsymbol{\theta}_{t_i}^{y,j} = (\theta_{t_i,1}^{y,j}, \theta_{t_i,2}^{y,j})^T$ and $\boldsymbol{W}^{y,j} = (W_1^{y,j}, W_2^{y,j})^T$. As in Section 5.2, we assign the components of the spatial error process $\boldsymbol{p}_i^{y,j} = (p_{i,1}^{y,j}, p_{i,2}^{y,j})^T$ independent zero mean Gaussian process priors with covariance functions

$$f_m^y(d_{jj'}) = \mathrm{Cov}(\theta_{t_i,m}^{y,j}, \theta_{t_i,m}^{y,j'}) = \sigma_{y,m}^2 \exp(-\psi_{y,m} d_{jj'}), \quad m = 1, 2.$$

The spatial humidity DLM then takes the form

$$\begin{aligned} \boldsymbol{Y}_{t_i} &= \boldsymbol{F}_{t_i}^y \boldsymbol{\theta}_{t_i}^y + \boldsymbol{v}_i^y, & \boldsymbol{v}_i^y &\overset{indep}{\sim} N\{\mathbf{0}, \mathrm{diag}(V^{y,1}, \ldots, V^{y,\ell})\} \\ \boldsymbol{\theta}_{t_i}^y &= \boldsymbol{\theta}_{t_{i-1}}^y + k_i \boldsymbol{w}_i^x + \boldsymbol{p}_i^y, & \boldsymbol{w}_i^y &\overset{indep}{\sim} N\{\mathbf{0}, \mathrm{diag}(\boldsymbol{W}^{y,1}, \ldots, \boldsymbol{W}^{y,\ell})\} \end{aligned} \tag{5.7}$$

where

- $\boldsymbol{F}_{t_i}^y = \mathrm{diag}(\boldsymbol{F}_{t_i}^{y,1}, \ldots, \boldsymbol{F}_{t_i}^{y,\ell})$;
- $\boldsymbol{\theta}_{t_i}^y = ((\boldsymbol{\theta}_{t_i}^{y,1})^T, \ldots, (\boldsymbol{\theta}_{t_i}^{y,\ell})^T)^T$;
- the 2ℓ-vector of spatial effects $\boldsymbol{p}_i^y$ is distributed analogously to $\boldsymbol{p}_i^x$.

Note that the joint model given by (5.5) and (5.7) induces a marginal model for hourly average humidity with the sinusoidal pattern observed in the data. After integrating out $X_{t_i}^j$ in the observation equation for $Y_{t_i}^j$, we obtain

$$Y_{t_i}^j = \boldsymbol{F}_{t_i}^{x,j} \boldsymbol{\theta}_{t_i}^{x,j} \theta_{t_i,1}^{y,j} + \theta_{t_i,2}^{y,j} + v_i^{y,j} + \theta_{t_i,1}^{y,j} v_i^{x,j}$$

which exhibits the same sinusoidal structure of (5.1), albeit with a different amplitude, phase and basal level. It is clear that the joint model for $(X_{t_i}^j, Y_{t_i}^j)^T$ is not a DLM, as the marginal humidity model depends on $\boldsymbol{\theta}_{t_i}^{x,j}$ and $\boldsymbol{\theta}_{t_i}^{y,j}$ in a nonlinear way. Nevertheless, the factorisation of the joint model as marginal and conditional DLMs can be exploited when performing inference for the model parameters, and this is the subject of the next section.

5.3 Inference

Fitting the model for temperature and humidity described in Section 5.2 to data is complicated by the fact that in practice, sensor data is sometimes missing at one or more locations. To deal with this scenario, we let $\boldsymbol{X}_{t_i}^o$ and $\boldsymbol{Y}_{t_i}^o$ denote the observed temperature and humidity processes at time t_i. We assume that if temperature is missing at location j at time t_i, then so is humidity (and vice-versa), as is the case for our application. The observation model can then be written as

$$\boldsymbol{X}_{t_i}^o = \boldsymbol{P}_{t_i}^x \boldsymbol{X}_{t_i}, \qquad \boldsymbol{Y}_{t_i}^o = \boldsymbol{P}_{t_i}^y \boldsymbol{Y}_{t_i} \tag{5.8}$$

where the $n_i^{x/y} \times \ell$ incidence matrix $\boldsymbol{P}_{t_i}^{x/y}$ determines which components are observed at time t_i. For example, if we have data streams from 5 different locations and temperature data are missing at the second and third location at time t_i, then the incidence matrix is

$$\boldsymbol{P}_{t_i}^x = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Let ϕ_x denote the flattened vector of $V^{x,1}, \ldots, V^{x,\ell}$, $\boldsymbol{W}^{x,1}, \ldots, \boldsymbol{W}^{x,\ell}$, $\boldsymbol{\sigma}_x$ and ψ_x. Define ϕ_y similarly. Given observations $\boldsymbol{x}_{1:t_i}^o$ and $\boldsymbol{y}_{1:t_i}^o$ at times $0 = t_1 < t_2 < \ldots < t_i$, our primarily goal is sequential exploration of the marginal posterior density $\pi(\phi_x, \phi_y | \boldsymbol{x}_{1:t_i}^o, \boldsymbol{y}_{1:t_i}^o)$. We assume that ϕ_x and ϕ_y are independent *a priori* with prior density $\pi(\phi_x, \phi_y) = \pi(\phi_x)\pi(\phi_y)$. Bayes' theorem

gives the posterior density of interest as

$$\begin{aligned}\pi(\boldsymbol{\phi}_x, \boldsymbol{\phi}_y | \boldsymbol{x}^o_{1:t_i}, \boldsymbol{y}^o_{1:t_i}) &\propto \pi(\boldsymbol{\phi}_x)\pi(\boldsymbol{\phi}_y)\pi(\boldsymbol{x}^o_{1:t_i}, \boldsymbol{y}^o_{1:t_i} | \boldsymbol{\phi}_x, \boldsymbol{\phi}_y) \\ &= \pi(\boldsymbol{\phi}_x)\pi(\boldsymbol{\phi}_y)\pi(\boldsymbol{x}^o_{1:t_i} | \boldsymbol{\phi}_x)\pi(\boldsymbol{y}^o_{1:t_i} | \boldsymbol{x}^o_{1:t_i} \boldsymbol{\phi}_y) \\ &\propto \pi(\boldsymbol{\phi}_x | \boldsymbol{x}^o_{1:t_i})\pi(\boldsymbol{\phi}_y | \boldsymbol{x}^o_{1:t_i}, \boldsymbol{y}^o_{1:t_i}) \end{aligned} \tag{5.9}$$

and so the parameter sets $\boldsymbol{\phi}_x$ and $\boldsymbol{\phi}_y$ are independent *a posteriori*. Moreover, we have that

$$\begin{aligned}\pi(\boldsymbol{\phi}_x | \boldsymbol{x}^o_{1:t_i}) &\propto \pi(\boldsymbol{\phi}_x | \boldsymbol{x}^o_{1:t_{i-1}})\pi(\boldsymbol{x}^o_{t_i} | \boldsymbol{x}^o_{1:t_{i-1}}, \boldsymbol{\phi}_x) \\ \pi(\boldsymbol{\phi}_y | \boldsymbol{x}^o_{1:t_i}, \boldsymbol{y}^o_{1:t_i}) &\propto \pi(\boldsymbol{\phi}_y | \boldsymbol{x}^o_{1:t_{i-1}}, \boldsymbol{y}^o_{1:t_{i-1}})\pi(\boldsymbol{y}^o_{t_i} | \boldsymbol{x}^o_{1:t_i}, \boldsymbol{y}^o_{1:t_{i-1}}, \boldsymbol{\phi}_y) \end{aligned} \tag{5.10}$$

where the observed data likelihood contributions $\pi(\boldsymbol{x}^o_{t_i} | \boldsymbol{x}^o_{1:t_{i-1}}, \boldsymbol{\phi}_x)$ and $\pi(\boldsymbol{y}^o_{t_i} | \boldsymbol{x}^o_{1:t_i}, \boldsymbol{y}^o_{1:t_{i-1}}, \boldsymbol{\phi}_y)$ can be calculated using a forward filter.

To simplify notation we consider the spatial temperature model and drop the superscript x. Given the form of the observation model in (5.8), we have that

$$\begin{aligned}\boldsymbol{X}^o_{t_i} &= \tilde{\boldsymbol{F}}_{t_i}\boldsymbol{\theta}_{t_i} + \tilde{\boldsymbol{v}}_i, \qquad \tilde{\boldsymbol{v}}_i \overset{indep}{\sim} N(\boldsymbol{0}, \tilde{\boldsymbol{V}}), \\ \boldsymbol{\theta}_{t_i} &= \boldsymbol{\theta}_{t_{i-1}} + \tilde{\boldsymbol{w}}_i, \qquad \tilde{\boldsymbol{w}}_i \overset{indep}{\sim} N(\boldsymbol{0}, \tilde{\boldsymbol{W}}), \end{aligned} \tag{5.11}$$

where $\tilde{\boldsymbol{F}}_{t_i} = \boldsymbol{P}_{t_i}\boldsymbol{F}_{t_i}$, $\tilde{\boldsymbol{V}} = \boldsymbol{P}_{t_i}\text{diag}(V^1, \ldots, V^\ell)\boldsymbol{P}^T_{t_i}$ and $\tilde{\boldsymbol{W}} = k_i^2\text{diag}(\boldsymbol{W}^1, \ldots, \boldsymbol{W}^\ell) + \boldsymbol{K}$. Now suppose that $\boldsymbol{\theta}_{t_0} \sim N(\boldsymbol{m}_0, \boldsymbol{C}_0)$ *a priori* and recall that $t_0 = 0$. The observed data likelihood increments $\pi(\boldsymbol{x}^o_{t_i} | \boldsymbol{x}^o_{1:t_{i-1}}, \boldsymbol{\phi}_x)$, and hence the full observed data likelihood $\pi(\boldsymbol{x}^o_{1:t_n} | \boldsymbol{\phi}_x)$, can be obtained from the forward filter described in Algorithm 14.

According to the performance of different SMC schemes illustrated in the previous chapter, the iterated batch importance sampling (IBIS) algorithm of Chopin (2002) demonstrated the best performance, in terms of posterior accuracy and computational cost. Hence we will choose the IBIS scheme for the real data analysis. Recall that the IBIS scheme involves two steps: an incremental weighting step and a rejuvenation (resample-move) step. In the incremental weight step, the weight is updated for each particle through the observed data likelihood contribution of the current observation, that is, $\omega^{(k)}_{t_i} \propto \omega^{(k)}_{t_{i-1}}\pi(\boldsymbol{x}^o_{t_i} | \boldsymbol{x}^o_{0:t_{i-1}}, \boldsymbol{\phi}^{(k)}_x)$. In the resample-move step, the particles are rejuvenated through MCMC steps that effectively circumvents particle degeneracy. The IBIS scheme as appropriate for the spatial temperature model is given by Algorithm 15.

The online IBIS scheme discussed in Section 3.3.6 is an extension scheme of the IBIS scheme, which can boost computational efficiency by estimating the parameter posterior over separated observation windows with bounded CPU costs, and is therefore particularly well suited to

Algorithm 14 Forward filter

1. Initialisation ($i = 0$). Sample $\boldsymbol{\theta}_{t_0} \sim N(\boldsymbol{m}_0, \boldsymbol{C}_0)$. Store the values of $\boldsymbol{m}_0$, $\boldsymbol{C}_0$.
2. For $i = 1, \ldots, n$,
   (a) Prior at t_i. Using the system equation, we have that $\boldsymbol{\theta}_{t_i} | \boldsymbol{x}^o_{1:t_{i-1}}, \boldsymbol{\phi}_x \sim N(\boldsymbol{m}_{t_{i-1}}, \boldsymbol{C}_{t_{i-1}} + \tilde{\boldsymbol{W}})$.
   (b) One step forecast. Using the observation equation, we have that
   $$\boldsymbol{X}^o_{t_i} | \boldsymbol{x}^o_{1:t_{i-1}}, \boldsymbol{\phi}_x \sim N\{\tilde{\boldsymbol{F}}_{t_i} \boldsymbol{m}_{t_{i-1}}, \tilde{\boldsymbol{F}}_{t_i} (\boldsymbol{C}_{t_{i-1}} + \tilde{\boldsymbol{W}}) \tilde{\boldsymbol{F}}^T_{t_i} + \tilde{\boldsymbol{V}}\}.$$
   Compute the observed data likelihood increment
   $$\pi(\boldsymbol{x}^o_{t_i} | \boldsymbol{x}^o_{1:t_{i-1}}, \boldsymbol{\phi}_x) = N\{\boldsymbol{x}^o_{t_i} \,;\, \tilde{\boldsymbol{F}}_{t_i} \boldsymbol{m}_{t_{i-1}} \,,\, \tilde{\boldsymbol{F}}_{t_i} (\boldsymbol{C}_{t_{i-1}} + \tilde{\boldsymbol{W}}) \tilde{\boldsymbol{F}}^T_{t_i} + \tilde{\boldsymbol{V}}\}.$$
   (c) Posterior at t_i. Combining the distributions in (a) and (b) gives the joint distribution of $\boldsymbol{\theta}_{t_i}$ and $\boldsymbol{X}^o_{t_i}$ (conditional on $\boldsymbol{x}_{1:t_{i-1}}$ and $\boldsymbol{\phi}_x$) as
   $$\begin{pmatrix} \boldsymbol{\theta}_{t_i} \\ \boldsymbol{X}^o_{t_i} \end{pmatrix} \sim N\left\{ \begin{pmatrix} \boldsymbol{m}_{t_{i-1}} \\ \tilde{\boldsymbol{F}}_{t_i} \boldsymbol{m}_{t_{i-1}} \end{pmatrix}, \begin{pmatrix} \boldsymbol{C}_{t_{i-1}} + \tilde{\boldsymbol{W}} & (\boldsymbol{C}_{t_{i-1}} + \tilde{\boldsymbol{W}}) \tilde{\boldsymbol{F}}^T_{t_i} \\ \tilde{\boldsymbol{F}}_{t_i} (\boldsymbol{C}_{t_{i-1}} + \tilde{\boldsymbol{W}}) & \tilde{\boldsymbol{F}}_{t_i} (\boldsymbol{C}_{t_{i-1}} + \tilde{\boldsymbol{W}}) \tilde{\boldsymbol{F}}^T_{t_i} + \tilde{\boldsymbol{V}} \end{pmatrix} \right\}$$
   and therefore $\boldsymbol{\theta}_{t_i} | \boldsymbol{x}^o_{1:t_i}, \boldsymbol{\phi}_x \sim N(\boldsymbol{m}_{t_i}, \boldsymbol{C}_{t_i})$, where
   $$\begin{aligned} \boldsymbol{m}_{t_i} &= \boldsymbol{m}_{t_{i-1}} + (\boldsymbol{C}_{t_{i-1}} + \tilde{\boldsymbol{W}}) \tilde{\boldsymbol{F}}^T_{t_i} \{\tilde{\boldsymbol{F}}_{t_i} (\boldsymbol{C}_{t_{i-1}} + \tilde{\boldsymbol{W}}) \tilde{\boldsymbol{F}}^T_{t_i} + \tilde{\boldsymbol{V}}\}^{-1} (\boldsymbol{x}^o_{t_i} - \tilde{\boldsymbol{F}}_{t_i} \boldsymbol{m}_{t_{i-1}}), \\ \boldsymbol{C}_{t_i} &= \boldsymbol{C}_{t_{i-1}} + \tilde{\boldsymbol{W}} - (\boldsymbol{C}_{t_{i-1}} + \tilde{\boldsymbol{W}}) \tilde{\boldsymbol{F}}^T_{t_i} \{\tilde{\boldsymbol{F}}_{t_i} (\boldsymbol{C}_{t_{i-1}} + \tilde{\boldsymbol{W}}) \tilde{\boldsymbol{F}}^T_{t_i} + \tilde{\boldsymbol{V}}\}^{-1} \tilde{\boldsymbol{F}}_{t_i} (\boldsymbol{C}_{t_{i-1}} + \tilde{\boldsymbol{W}}) \,. \end{aligned}$$
   Store the values of $\boldsymbol{m}_{t_i}$, $\boldsymbol{C}_{t_i}$ and $\pi(\boldsymbol{x}^o_{t_i} | \boldsymbol{x}^o_{1:t_{i-1}}, \boldsymbol{\phi}_x)$.

large and complex models. Algorithm 16 summarises the online IBIS scheme for the spatial temperature model.

For the data considered here, temperature and humidity measurements are always observed or missing at the same time. However, if temperature is missing but humidity is observed, inference is complicated by the fact that the DLM for humidity is conditional on temperature. Essentially, the linear Guassian structure allows for integrating out the state process, but not both the missing temperature values and the state process. A simple but *ad-hoc* solution to deal with this problem is to just ignore the humidity observation when temperature is missing at the same time. The inference scheme described above can then be applied. A more principle approach is discussed below, where we create an imputation method that allows to infer the missing temperature data by maintaining a particle approximation in the IBIS scheme.

Algorithm 15 IBIS scheme for the spatial temperature model

1. Initialisation. For $k = 1, \ldots, N$ sample $\boldsymbol{\phi}_x^{(k)} \sim \pi(\cdot)$ and set $\tilde{\omega}_0^{(k)} = 1$. Store $\boldsymbol{m}_0^{(k)}$ and $\boldsymbol{C}_0^{(k)}$.

   For $i = 1, \ldots, n$:

2. Sequential importance sampling. For $k = 1, \ldots, N$:

   (a) Perform iteration i of the forward filter (Algorithm 14) to obtain $\pi(\boldsymbol{x}_{t_i}^o | \boldsymbol{x}_{1:t_{i-1}}^o, \boldsymbol{\phi}_x^{(k)})$, $\boldsymbol{m}_{t_i}^{(k)}$ and $\boldsymbol{C}_{t_i}^{(k)}$. Note the convention that $\pi(\boldsymbol{x}_1^o | \boldsymbol{\phi}_x^{(k)}) = \pi(\boldsymbol{x}_1^o | \boldsymbol{x}_{1:0}^o, \boldsymbol{\phi}_x^{(k)})$.

   (b) Update and normalise the importance weights using

$$\tilde{\omega}_{t_i}^{(k)} = \tilde{\omega}_{t_{i-1}}^{(k)} \pi(\boldsymbol{x}_{t_i}^o | \boldsymbol{x}_{1:t_{i-1}}^o, \boldsymbol{\phi}_x^{(k)}), \qquad \omega_{t_i}^{(k)} = \frac{\tilde{\omega}_{t_i}^{(k)}}{\sum_{j=1}^N \tilde{\omega}_{t_i}^{(j)}}.$$

   (c) Update the observed data likelihood using

$$\pi(\boldsymbol{x}_{1:t_i}^o | \boldsymbol{\phi}_x^{(k)}) = \pi(\boldsymbol{x}_{1:t_{i-1}}^o | \boldsymbol{\phi}_x^{(k)}) \pi(\boldsymbol{x}_{t_i}^o | \boldsymbol{x}_{1:t_{i-1}}^o, \boldsymbol{\phi}_x^{(k)}).$$

3. If ESS $< \delta N$ resample and move as follows. For $k = 1, \ldots, N$:

   (a) Sample indices $a_k \sim \mathcal{M}(\omega^{1:N})$ and set $\{\boldsymbol{\phi}_x^{(k)}, \tilde{\omega}_{t_i}^{(k)}\} := \{\boldsymbol{\phi}_x^{(a_k)}, 1\}$, $\pi(\boldsymbol{x}_{1:t_i}^o | \boldsymbol{\phi}_x^{(k)}) := \pi(\boldsymbol{x}_{1:t_i}^o | \boldsymbol{\phi}_x^{(a_k)})$, $\boldsymbol{m}_{t_i}^{(k)} := \boldsymbol{m}_{t_i}^{(a_k)}$ and $\boldsymbol{C}_{t_i}^{(k)} := \boldsymbol{C}_{t_i}^{(a_k)}$.

   (b) Propose $\boldsymbol{\phi}_x^* \sim q(\cdot | \boldsymbol{\phi}_x^{(k)})$. Perform iterations $1, \ldots, i$ of the forward filter (Algorithm 14) to obtain $\pi(\boldsymbol{x}_{1:t_i}^o | \boldsymbol{\phi}_x^*)$. With probability

$$\min \left\{ 1, \frac{\pi(\boldsymbol{\phi}_x^*) \pi(\boldsymbol{x}_{1:t_i}^o | \boldsymbol{\phi}_x^*)}{\pi(\boldsymbol{\phi}_x^{(k)}) \pi(\boldsymbol{x}_{1:t_i}^o | \boldsymbol{\phi}_x^{(k)})} \times \frac{q(\boldsymbol{\phi}_x^{(k)} | \boldsymbol{\phi}_x^*)}{q(\boldsymbol{\phi}_x^* | \boldsymbol{\phi}_x^{(k)})} \right\},$$

   put $\boldsymbol{\phi}_x^{(k)} := \boldsymbol{\phi}_x^*$, $\pi(\boldsymbol{x}_{1:t_i}^o | \boldsymbol{\phi}_x^{(k)}) := \pi(\boldsymbol{x}_{1:t_i}^o | \boldsymbol{\phi}_x^*)$, $\boldsymbol{m}_{t_i}^{(k)} := \boldsymbol{m}_{t_i}^*$ and $\boldsymbol{C}_{t_i}^{(k)} := \boldsymbol{C}_{t_i}^*$.

Dealing with missing temperature

Let $\boldsymbol{x}_{t_i}^m$ represent missing temperature at time t_i. Then the conditional observed data likelihood for humidity is

$$\pi(\boldsymbol{y}_{t_i} | \boldsymbol{x}_{1:t_i}^o, \boldsymbol{y}_{1:t_{i-1}}^o, \boldsymbol{\phi}_y, \boldsymbol{x}_{t_i}^m).$$

Further integrating out $\boldsymbol{x}_{t_i}^m$ is analytically intractable. We illustrate a principled approach that is to create an imputation method which allows us to infer the missing temperature data by maintaining a particle approximation of $\boldsymbol{x}_{t_i}^m$ in the IBIS scheme. Considering missing temperature as an unknown variable, the parameter space at time t_i becomes $(\boldsymbol{\phi}_x, \boldsymbol{\phi}_y, \boldsymbol{x}_{t_i}^m)$. Therefore, the

Algorithm 16 Online IBIS scheme for the spatial temperature model

1. Initialisation. Divide the observed period into b windows, $s \in \{1,\dots,b\}$. Denote by t_i^s the ith observation time in window s, $i = 1,\dots,n_s$. For $s = 1$, implement the IBIS scheme (Algorithm 15). For $s = 2,\dots,b$ and $i = 1,\dots,n_s$:

2. Sequential importance sampling. For $k = 1,\dots,N$:

   (a) Perform iteration i (corresponding to time t_i^s) of the forward filter (Algorithm 14) to obtain $\pi(\boldsymbol{x}^o_{t_i^s}|\boldsymbol{x}^o_{1:t^s_{i-1}},\boldsymbol{\phi}_x^{(k)})$, $\boldsymbol{m}^{(k)}_{t_i^s}$ and $\boldsymbol{C}^{(k)}_{t_i^s}$.

   (b) Update and normalise the importance weights using

$$\tilde{\omega}^{(k)}_{t_i^s} = \tilde{\omega}^{(k)}_{t^s_{i-1}}\pi(\boldsymbol{x}^o_{t_i^s}|\boldsymbol{x}^o_{1:t^s_{i-1}},\boldsymbol{\phi}_x^{(k)}), \qquad \omega^{(k)}_{t_i^s} = \frac{\tilde{\omega}^{(k)}_{t_i^s}}{\sum_{z=1}^N \tilde{\omega}^{(z)}_{t_i^s}}.$$

   (c) Update the observed data likelihood contribution in the current window using

$$\pi(\boldsymbol{x}^o_{t_1^s:t_i^s}|\boldsymbol{x}^o_{1:(s-1)T},\boldsymbol{\phi}_x^{(k)}) = \pi(\boldsymbol{x}^o_{t_1^s:t^s_{i-1}}|\boldsymbol{x}^o_{1:(s-1)T},\boldsymbol{\phi}_x^{(k)})\pi(\boldsymbol{x}^o_{t_i^s}|\boldsymbol{x}^o_{1:t^s_{i-1}},\boldsymbol{\phi}_x^{(k)}),$$

   with the convention that $\pi(\boldsymbol{x}^o_{t_1^s:t_i^s}|\boldsymbol{x}^o_{1:(s-1)T},\boldsymbol{\phi}_x^{(k)}) = \pi(\boldsymbol{x}_{t_1^s}|\boldsymbol{x}^o_{1:(s-1)T},\boldsymbol{\phi}_x^{(k)})$ for $i = 1$.

3. If ESS $< \delta N$ resample and move. For $k = 1,\dots,N$:

   (a) Sample indices $a_k \sim \mathcal{M}(\omega^{1:N})$ and set $\{\boldsymbol{\phi}_x^{(k)},\tilde{\omega}^{(k)}_{t_i^s}\} := \{\boldsymbol{\phi}_x^{(a_k)},1\}$, $\boldsymbol{m}^{(k)}_{t_i^s} := \boldsymbol{m}^{(a_k)}_{t_i^s}$, $\boldsymbol{C}^{(k)}_{t_i^s} := \boldsymbol{C}^{(a_k)}_{t_i^s}$ and $\pi(\boldsymbol{x}^o_{t_1^s:t_i^s}|\boldsymbol{x}^o_{1:(s-1)T},\boldsymbol{\phi}_x^{(k)}) := \pi(\boldsymbol{x}^o_{t_1^s:t_i^s}|\boldsymbol{x}^o_{1:(s-1)T},\boldsymbol{\phi}_x^{(a_k)})$.

   (b) Propose $\boldsymbol{\phi}_x^* \sim \log N(\log\boldsymbol{\phi}_x^{(k)}, h_s^2)$. Using $\boldsymbol{m}^*_{(s-1)T} = \boldsymbol{m}^{(k)}_{(s-1)T}$ and $\boldsymbol{C}^*_{(s-1)T} = \boldsymbol{C}^{(k)}_{(s-1)T}$, perform iterations $1,\dots,i$ (corresponding to times $t_1^s,\dots,t_i^s$) of the forward filter (Algorithm 14) to obtain $\pi(\boldsymbol{x}^o_{t_1^s:t_i^s}|\boldsymbol{x}^o_{1:(s-1)T},\boldsymbol{\phi}_x^*)$. With probability

$$\min\left\{1, \frac{\pi(\boldsymbol{x}^o_{t_1^s:t_i^s}|\boldsymbol{x}^o_{1:(s-1)T},\boldsymbol{\phi}_x^*)}{\pi(\boldsymbol{x}^o_{t_1^s:t_i^s}|\boldsymbol{x}^o_{1:(s-1)T},\boldsymbol{\phi}_x^{(k)})}\right\},$$

   put $\boldsymbol{\phi}_x^{(k)} := \boldsymbol{\phi}_x^*$, $\pi(\boldsymbol{x}^o_{t_1^s:t_i^s}|\boldsymbol{x}^o_{1:(s-1)T},\boldsymbol{\phi}_x^{(k)}) := \pi(\boldsymbol{x}^o_{t_1^s:t_i^s}|\boldsymbol{x}^o_{1:(s-1)T},\boldsymbol{\phi}_x^*)$, $\boldsymbol{m}^{(k)}_{t_i^s} := \boldsymbol{m}^*_{t_i^s}$ and $\boldsymbol{C}^{(k)}_{t_i^s} := \boldsymbol{C}^*_{t_i^s}$.

posterior of parameters and missing data can be expanded as

$$\pi(\boldsymbol{\phi}_x,\boldsymbol{\phi}_y,\boldsymbol{x}^m_{t_i}|\boldsymbol{x}^o_{1:t_{i-1}},\boldsymbol{y}^o_{1:t_i},\boldsymbol{x}^o_{t_i}) \propto \underbrace{\pi(\boldsymbol{\phi}_x,\boldsymbol{\phi}_y|\boldsymbol{x}^o_{1:t_{i-1}},\boldsymbol{y}^o_{1:t_{i-1}})\pi(\boldsymbol{x}^m_{t_i}|\boldsymbol{x}^o_{1:t_{i-1}},\boldsymbol{\phi}_x)}_{\text{the prior}} \times$$
$$\underbrace{\pi(\boldsymbol{y}^o_{t_i}|\boldsymbol{x}^o_{1:t_i},\boldsymbol{y}^o_{1:t_{i-1}},\boldsymbol{x}^m_{t_i},\boldsymbol{\phi}_y)\pi(\boldsymbol{x}^o_{t_i}|\boldsymbol{x}^o_{1:t_{i-1}},\boldsymbol{y}^o_{1:t_{i-1}},\boldsymbol{\phi}_x)}_{\text{the observed data likelihood}}.$$

The prior and the observed data likelihood for the joint model are indicated in the above expanded equation. Importance resampling may proceed by using the prior as the importance density. In the rejuvenation step, the proposed particles of $\boldsymbol{x}_{t_i}^{m,*}$ are generated based on the resampled set $\boldsymbol{x}_{t_i}^{m,(k)}$ through a joint random walk. The acceptance probability for the particle move can be written as

$$
\begin{aligned}
&A(\boldsymbol{\phi}_x^*, \boldsymbol{\phi}_y^*, \boldsymbol{x}_{t_i}^{m,*} | \boldsymbol{\phi}_x^{(k)}, \boldsymbol{\phi}_y^{(k)}, \boldsymbol{x}_{t_i}^{m,(k)}) \\
&= \prod_{j=1}^{n_{par}^x} \frac{\pi(\phi_{x,j}^*)}{\pi(\phi_{x,j}^{(k)})} \prod_{p=1}^{n_{par}^y} \frac{\pi(\phi_{y,p}^*)}{\pi(\phi_{y,p}^{(k)})} \prod_{j=1}^{n_{par}^x} \frac{\phi_{x,j}^*}{\phi_{x,j}^{(k)}} \prod_{p=1}^{n_{par}^y} \frac{\phi_{y,p}^*}{\phi_{y,p}^{(k)}} \times \\
&\frac{\pi(\boldsymbol{x}_{t_i}^{m,*} | \boldsymbol{x}_{1:t_{i-1}}^o, \boldsymbol{\phi}_x^*)}{\pi(\boldsymbol{x}_{t_i}^{m,(k)} | \boldsymbol{x}_{1:t_{i-1}}^o, \boldsymbol{\phi}_x^{(k)})} \frac{\pi(\boldsymbol{x}_{t_i}^o | \boldsymbol{x}_{1:t_{i-1}}^o, \boldsymbol{\phi}_x^*)}{\pi(\boldsymbol{x}_{t_i}^o | \boldsymbol{x}_{1:t_{i-1}}^o, \boldsymbol{\phi}_x^{(k)})} \prod_{r=1}^{i-1} \frac{\pi(\boldsymbol{x}_{t_r}^o | \boldsymbol{x}_{1:t_{r-1}}^o, \boldsymbol{\phi}_x^*)}{\pi(\boldsymbol{x}_{t_r}^o | \boldsymbol{x}_{1:t_{r-1}}^o, \boldsymbol{\phi}_x^{(k)})} \times \\
&\frac{\pi(\boldsymbol{y}_{t_i}^o | \boldsymbol{y}_{1:t_{i-1}}^o, \boldsymbol{\phi}_y^*, \boldsymbol{x}_{t_i}^{m,*}, \boldsymbol{x}_{t_i}^o)}{\pi(\boldsymbol{y}_{t_i}^o | \boldsymbol{y}_{1:t_{i-1}}^o, \boldsymbol{\phi}_y^{(k)}, \boldsymbol{x}_{t_i}^{m,(k)}, \boldsymbol{x}_{t_i}^o)} \prod_{r=1}^{i-1} \frac{\pi(\boldsymbol{y}_{t_r}^o | \boldsymbol{y}_{1:t_{r-1}}^o, \boldsymbol{\phi}_y^*, \boldsymbol{x}_{1:t_r})}{\pi(\boldsymbol{y}_{t_r}^o | \boldsymbol{y}_{1:t_{r-1}}^o, \boldsymbol{\phi}_y^{(k)}, \boldsymbol{x}_{1:t_r})}
\end{aligned}
$$

with the convention $\pi(\boldsymbol{x}_{t_1} | \boldsymbol{x}_{1:t_0}, \cdot) = \pi(\boldsymbol{x}_{t_1} | \cdot)$ and $\pi(\boldsymbol{y}_{t_1} | \boldsymbol{y}_{1:t_0}, \cdot) = \pi(\boldsymbol{y}_{t_1} | \cdot)$. Algorithm 17 summarises the details of the IBIS algorithm for the scenario when temperature is missing but humidity is observed.

A problem with this approach is that the number of missing temperature data points (in which case the corresponding humidity are observed) is likely to increase over time, and eventually, the missing variable space will become large. Moreover, a particle representation of all missing variables must be maintained throughout the run, increasing storage costs. In order to limit the number of unknown variables and maintain computational efficiency at the expense of some posterior accuracy, we can estimate the unobserved temperature by calculating the expectation(s) of the particles for the missing data. The details of the IBIS algorithm for missing temperature with expectation replacement is provided in Algorithm 18.

5.4 Within-sample predictions and out-of-sample forecasts

Recall the DLM given by (5.11). In order to compute within-sample predictions, the smoothing density $\pi(\boldsymbol{\theta}_{1:t_n} | \boldsymbol{x}_{1:t_n}^o, \boldsymbol{\phi}_x)$ is required. Draws from this density can be readily obtained by using a

backward sampler that recursively draws from

$$\pi(\boldsymbol{\theta}_{t_i}|\boldsymbol{\theta}_{t_{i+1}}, \boldsymbol{x}^o_{1:t_i}, \boldsymbol{\phi}_x) = N\left\{\boldsymbol{\theta}_{t_i};\, \boldsymbol{m}_{t_i} + \boldsymbol{B}_{t_i}(\boldsymbol{\theta}_{t_{i+1}} - \boldsymbol{m}_{t_i}),\, \boldsymbol{C}_{t_i} - \boldsymbol{B}_{t_i}\boldsymbol{R}_{t_{i+1}}\boldsymbol{B}^T_{t_i}\right\}, \tag{5.12}$$

where $\boldsymbol{B}_{t_i} = \boldsymbol{C}_{t_i}\boldsymbol{R}^{-1}_{t_{i+1}}$ and $\boldsymbol{R}_{t_{i+1}} = \boldsymbol{C}_{t_i} + \tilde{\boldsymbol{W}}$; see, for example, West and Harrison (1999). Hence, given an equally weighted sample $\{\boldsymbol{\phi}^{1:N}_x\}$ from the marginal posterior $\pi(\boldsymbol{\phi}_x|\boldsymbol{x}^o_{1:t_n})$, we can integrate over parameter uncertainty to generate draws from the within-sample system posterior predictive density $\pi(\boldsymbol{\theta}_{1:t_n}|\boldsymbol{x}^o_{1:t_n})$ by recursively drawing from (5.12) for each particle $\boldsymbol{\phi}^{(k)}_x$ (and the associated quantities $\boldsymbol{m}^{(k)}_{t_i}$, $\boldsymbol{C}^{(k)}_{t_i}$ generated by the forward filter). Subsequently, the within-sample observation posterior predictive density $\pi(\boldsymbol{x}_{1:t_n}|\boldsymbol{x}^o_{1:t_n})$ can be sampled by drawing

$$\boldsymbol{X}^{(k)}_{t_i}|\boldsymbol{\theta}^{(k)}_{t_i}, \boldsymbol{\phi}^{(k)}_x \sim N(\boldsymbol{F}_{t_i}\boldsymbol{\theta}^{(k)}_{t_i},\, \boldsymbol{V}^{(k)}), \qquad i = 1,\ldots,n, \quad k = 1,\ldots,N.$$

Out-of-sample system and observation forecast distributions can be obtained by again exploiting the linear Gaussian structure of the DLM. Given an equally weighted sample $\{\boldsymbol{\phi}^{1:N}_x\}$ from the marginal posterior $\pi(\boldsymbol{\phi}_x|\boldsymbol{x}^o_{1:t_n})$, samples from $\pi(\boldsymbol{\theta}_{t_{n+1}}|\boldsymbol{x}^o_{1:t_n})$ and $\pi(\boldsymbol{x}_{t_{n+1}}|\boldsymbol{x}^o_{1:t_n})$ can be obtained by recursively drawing

$$\begin{aligned}
\boldsymbol{\theta}^{(k)}_{t_{n+1}}|\boldsymbol{\phi}^{(k)}_x &\sim N(\boldsymbol{m}^{(k)}_{t_n},\, \boldsymbol{C}^{(k)}_{t_n} + \tilde{\boldsymbol{W}}^{(k)}), \qquad k = 1,\ldots,N \\
\boldsymbol{x}^{(k)}_{t_{n+1}}|\boldsymbol{\phi}^{(k)}_x &\sim N\{\boldsymbol{F}_{t_{n+1}}\boldsymbol{m}^{(k)}_{t_n},\, \boldsymbol{F}_{t_{n+1}}(\boldsymbol{C}^{(k)}_{t_n} + \tilde{\boldsymbol{W}}^{(k)})\boldsymbol{F}^T_{t_{n+1}} + \boldsymbol{V}^{(k)}\}, \qquad k = 1,\ldots,N.
\end{aligned}$$

5.5 Model selection

As noted in Section 5.2, seasonality in the marginal DLM can be accounted for in (at least) two ways. A sinusoid can be specified in the observation equation, with a system equation describing the evolution of the parameters governing the amplitude and phase. Alternatively, a Fourier form structure can be used in the system equation where the appropriate number of harmonics must be specified by the practitioner. Our joint model consists of a marginal DLM for temperature and a conditional DLM for humidity given tempertaure. This induces a marginal DLM for humidity with the same form as that for temperature. We therefore consider three candidate spatial DLMs for modelling temperature and humidity data marginally: 1. sinusoidal form DLM (sDLM); 2. Fourier form DLM with 1 harmonic (FDLM1); 3. Fourier form DLM with 2 harmonics (FDLM2).

Choosing between these competing models is possible via computation of the Bayes factor

(Kass and Raftery, 1995; Frühwirth-Schnatter, 1995), which, under the assumption of equal prior probability for two competing models, say $M1$ and $M2$, is defined as the ratio of the observed data likelihood given $M1$, and given $M2$. The Bayes factor based on temperature data is therefore

$$BF = \frac{\pi(\boldsymbol{x}^o_{1:t_n}|M1)}{\pi(\boldsymbol{x}^o_{1:t_n}|M2)}$$

with a similar form when using humidity data. Note that $BF < 1$ suggests the data support $M2$.

In this context, the observed data likelihood is known as the evidence and can be factorised as

$$\pi(\boldsymbol{x}^o_{1:t_n}) = \pi(\boldsymbol{x}^o_1) \prod_{i=2}^{n} \pi(\boldsymbol{x}^o_{t_i}|\boldsymbol{x}^o_{1:t_{i-1}}).$$

We note that it is straightforward to estimate the evidence using the output of the IBIS scheme, at virtually no additional computational cost. Each factor $L_{t_i} = \pi(\boldsymbol{x}^o_{t_i}|\boldsymbol{x}^o_{1:t_{i-1}}, \boldsymbol{\phi}^{(k)}_x)$ in the product above is estimated by

$$L_1 = \sum_{k=1}^{N} \frac{1}{N} \pi(\boldsymbol{x}^o_1|\boldsymbol{\phi}^{(k)}_x), \qquad L_{t_i} = \sum_{k=1}^{N} \omega^{(k)}_{t_{i-1}} \pi(\boldsymbol{x}^o_{t_i}|\boldsymbol{x}^o_{1:t_{i-1}}, \boldsymbol{\phi}^{(k)}_x), \quad i = 2, \ldots, n. \tag{5.13}$$

Unfortunately, the size of the real data set precludes calculation of the Bayes factor using all measurements at all sites. To guide our modelling approach we therefore consider 400 observations at three randomly chosen locations. We take the independent inverse gamma distribution $IG(1, 0.01)$ *a priori* for the parameters in each model. We implement the full IBIS scheme with a serial multinomial resampling step for each model, using $N = 10^7$ particles. To account for Monte Carlo error, we repeat this process 30 times. Taking FDLM2 as a baseline for comparison, we compute Bayes factors for sDLM vs FDLM2 and FDLM1 vs FDLM2. Figure 5.3 and 5.4 show the mean $\log -BF$ value (and 95% credible interval) based on data $\boldsymbol{x}^o_{1:t}$ and $\boldsymbol{y}^o_{1:t}$ against t. For the marginal temperature DLM it is clear that FDLM2 is the least favoured model. Furthermore, for $t > 80$, the log Bayes factors corresponding to sDLM against FDLM2 are always strictly greater than those corresponding to FDLM1 against FDLM2. For the marginal humidity DLM, there is little difference in overall fit between sDLM and FDLM1. Given that computational cost scales as $1 : 1.1 : 1.3$ for $sDLM : FDLM1 : FDLM2$, we conclude that sDLM offers the best compromise between model fit and computational efficiency.

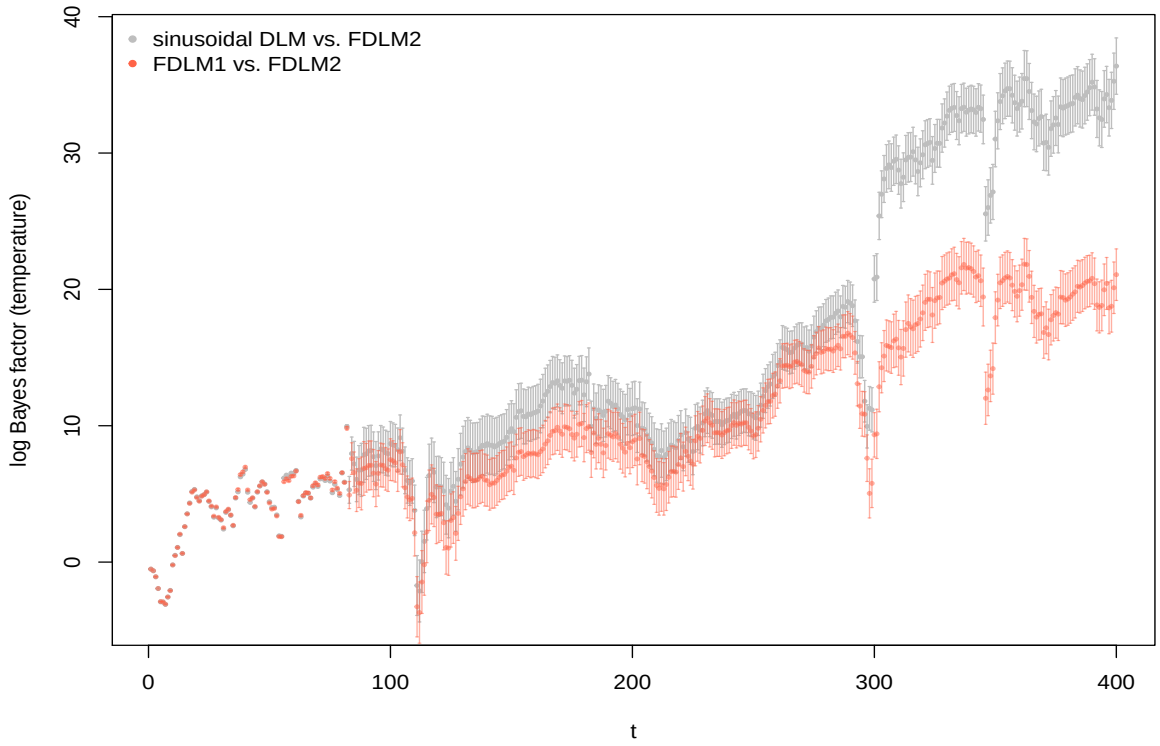

Figure 5.3: Mean and 95% credible interval of the log Bayes factor comparing temperature sDLM against FDLM2 and FDLM1 against FDLM2, over time.

5.6 Parallel computing

5.6.1 System frameworks for parallelisation

Parallel computing is an efficient way of computation that allows independent tasks to be processed simultaneously in order to minimise the computational time for a programme. Thanks to the rapid development of computing hardwares, a multi-core high performance computing cluster (HPCC) provides a powerful platform to carry out the parallel computing operations to process large data sets and big models. An HPCC consists of hundreds of nodes (analogous to a cluster of individual computers), and for each node, it has multiple codes within which to share a piece of own memory. Those cores are interconnected with each other to form a communication network. According to the nature of an individual job, the CPU parallelisation can be conducted in two different ways, under a shared memory system or a distributed memory system.

Within a shared memory system, the parallel tasks are only allocated to the multiple cores within one node and run as a batch of child threads. Once all the child threads are completed, the

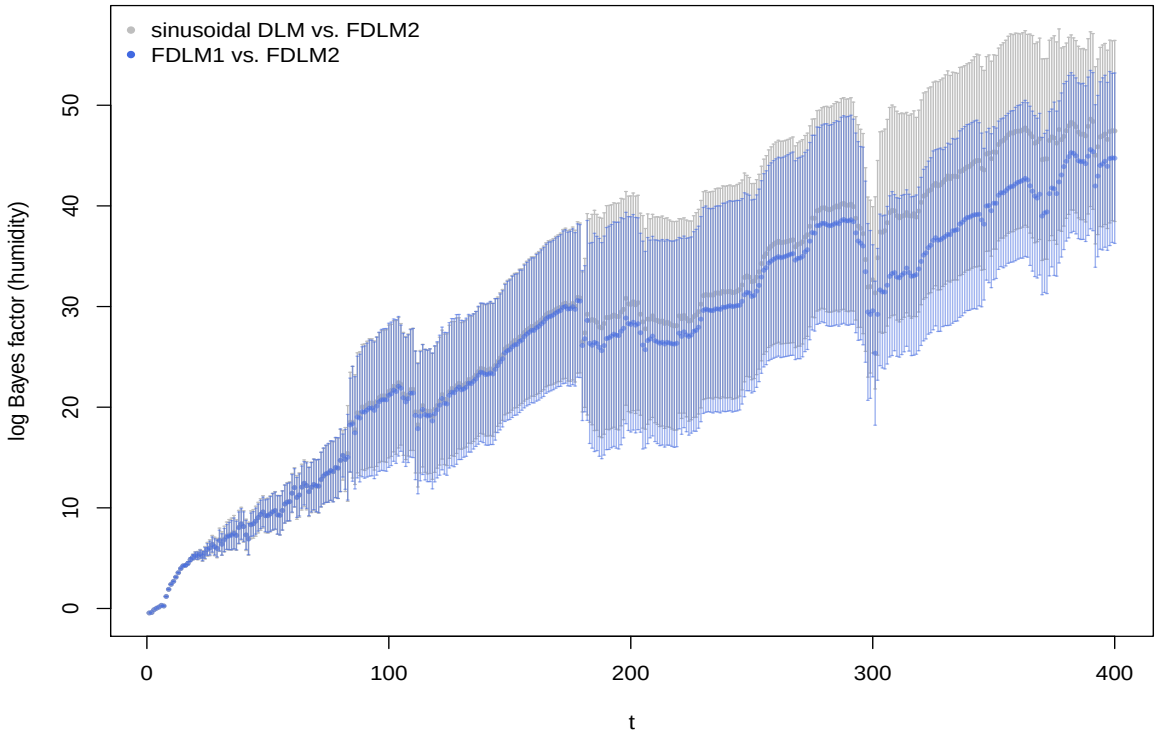

Figure 5.4: Mean and 95% credible interval of the log Bayes factor comparing humidity sDLM against FDLM2 and FDLM1 against FDLM2, over time.

information will be gathered into one core and the serial process starts again as a master thread. Shared memory systems offer an efficient way to deal with a moderate amount of parallel works by taking advantage of fast communication within a node. However this method will hit the bottleneck of memory limitation when the amount of parallel tasks is large. A distributed memory system, to a certain extent, addresses this problem by providing a more flexible environment, where trunks of jobs are randomly allocated and processed over multiple cores in different nodes. In the distributed memory system, the memory size is scalable and the information can be communicated through a high speed network. Figure 5.5 demonstrates the work flow of a parallel programme conducted in a shared memory system and distribution memory system respectively.

Most of programs cannot be fully parallelised in reality. Amdahl's Law (Amdahl, 1967) explains the potential speed-up of a parallel program, in which it is stated that

$$SU = \frac{1}{\frac{P}{N_c} + S}$$

where SU is speed-up, P is the proportion of the job that is parallelised, S is the proportion of the job that is serial, N_c is the number of cores, and $P + S = 1$. Therefore we can see

that by using parallel computing techniques the program would speed up the process, but the speed-up size never achieves the number of cores used if some part of the program cannot be parallelised. Amdahl's law only gives us a theoretical view on the relationship for the time spent on the serial process and parallel process. By considering together the effect of the time taken by the information exchange between the cores, it is clear that using more cores does not always result in an increase in computational efficiency. This phenomenon is especially reflected in a distributed memory system. To demonstrate this, we consider the example taken in Section 5.5 by fitting sDLM to the 400 observation data set for three sites and implement the full IBIS scheme using $N = 10^5$ particles. Due to the nature of independent particles, the IBIS scheme can be partially parallelised (more details will be discussed in the next section). We conduct several runs by running the parallel part of the IBIS scheme through different numbers of cores in a distributed memory system. Figure 5.6 shows the change of actual speed-up for the program as the number of cores increases. We can see, up to the number of 50 cores, the actual speed-up always increases, although the slope of speed-up gradually decreases after 10 cores. The speed-up eventually drops due to increasing communication time by adding an additional number of cores over 50. In practice, the number of cores needs to be carefully chosen based on the amount of workloads and the memory size requsted for each core.

OpenMP is an application programming interface (API) designed for parallel programming in a shared memory computing environment. An important concept for OpenMP is the preparatory definitions of shared variables and private variables at the beginning of the parallel part of a programme. The directive is sent to allocate a part of memory to store shared variables, which allows all the child threads to be able to access the shared variables. All the private variables are carried by the child threads and they cannot access each other. MPI is the message passing interface, a communication API that manages the parallel jobs in a distributed memory system. Unlike OpenMP, we do not need to define shared variables and private variables in advance under MPI, as all the variables taken to the parallel work are defined as private. When using MPI, we need to clarify the master core and child cores clearly to allow manually control the message communication between specific cores. However all those jobs will be automatically set up if OpenMP is applied. More details about OpenMP and MPI techniques have been discussed by Quinn (2003), Pacheco (1996), Gropp et al. (2014) and Chapman et al. (2007).

5.6.2 Parallelisation for resampling

The incremental weighting steps are readily parallelised in an SMC scheme. Additionally, for IBIS the move step can be performed independently for each particle. However, commonly used

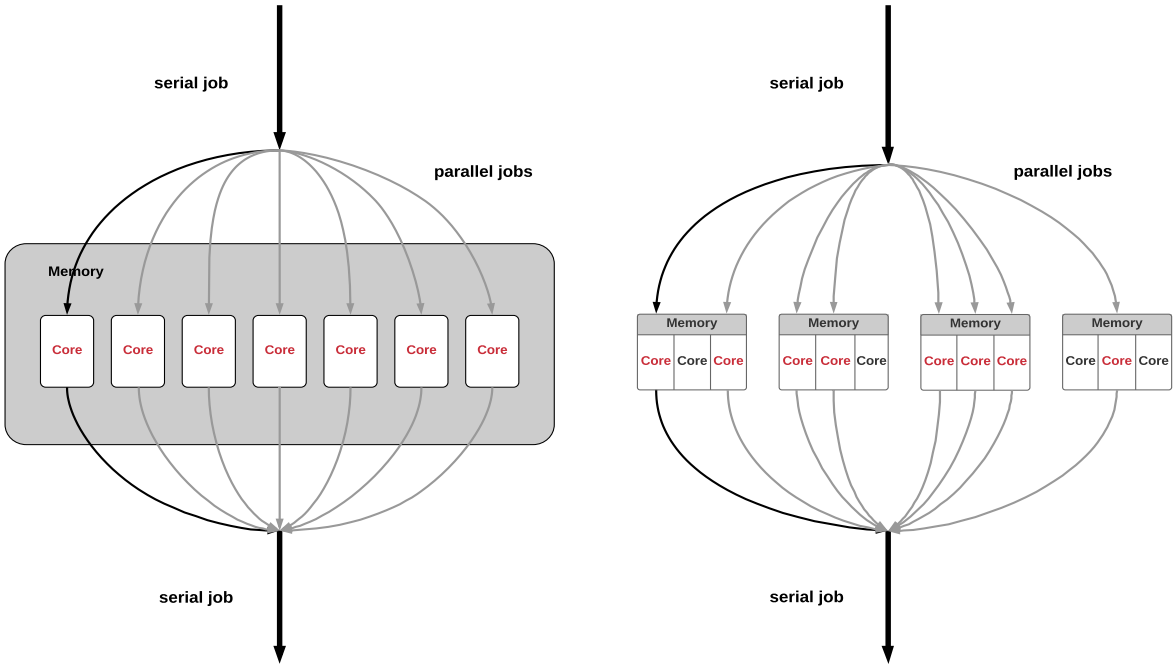

Figure 5.5: Left: work flow of a parallel program in a shared memory system; right: work flow of a parallel program in a distributed memory system.

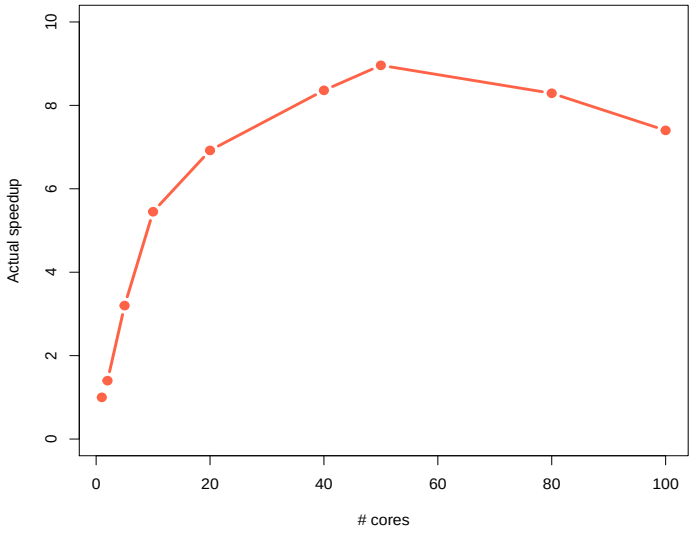

Figure 5.6: Actual speed-up by running parallel jobs through different numbers of cores in a distributed memory system.

resampling schemes, such as the multinomial approach considered here, involve a collective operation (summing the weights) precluding obvious parallelisation of the full IBIS scheme. Hendeby et al. (2010) and Gong et al. (2012) describe a forward adder tree method which parallelises the calculation of the cumulative weight. Murray et al. (2016) suggest parallel Metropolis resampling and rejection resampling schemes to mitigate numerical instabilities of summing the weights for a large number of particles. However, these methods still require information exchange and global operations and they are designed mainly for use on shared memory systems.

Distributed memory systems are naturally amenable to heavy parallelised jobs. In this context, a number of parallel resampling methods have been discussed in the literature; see, for example, Brun et al. (2002) and Bolić et al. (2004, 2005). We follow the local resampling method (Brun et al., 2002) by partitioning particles into disjoint subsets, within which resampling is performed. The algorithm proceeds by first calculating a local ESS for each subset of particles. If a local ESS is less than a threshold, then the rejuvenation step is triggered locally. The innovation variance for the MH proposal in the move step is also calculated locally based on the individual particle subset. To mitigate load-balance problems that can occur when the resample-move step is executed for some subsets but not others, we also carry out a rejuvenation step at regular time points, e.g. every 20 time points. This approach naturally fits within the distributed memory architecture and allows full parallelisation of the IBIS scheme. In principle, this approach should significantly improve computational efficiency of the inference scheme, as there is no need for task communication. However, in practice the number of informative particles may reduce significantly in some subsets as the algorithm runs. This in turn results in the rejuvenation step being executed more frequently. Therefore, a trade-off has to be considered carefully between the number of particle subsets and the number of particles in each subset. Section 6.1.1 describes a simulation study comparing a standard serial implementation with a fully parallelised version (with local resampling).

Algorithm 17 IBIS algorithm for missing temperature

1. Initialisation. For $k = 1, \ldots, N$ sample $\phi_x^{(k)}, \phi_y^{(k)} \sim \pi(\cdot)$ and set $\tilde{\omega}_0^{(k)} := 1$. Store $\boldsymbol{m}_0^{(k)}$ and $\boldsymbol{C}_0^{(k)}$. Denote t_g^m as time when temperature is missing but humidity observed, for $g = 1, \ldots, p$ and for $i = 1, \ldots, n$:

2. If $t_i \neq t_g^m$, implement the IBIS scheme (Algorithm 15), otherwise go to 2.

3. Sequential importance sampling. For $k = 1, \ldots, N$:

   (a) Sample $\boldsymbol{x}_{t_i}^{m,(k)} \sim \pi(\boldsymbol{x}_{t_i}^m | \boldsymbol{x}_{1:t_{i-1}}^o, \phi_x^{(k)})$.

   (b) Perform iteration i of the forward filter to obtain

   $$\pi(\boldsymbol{y}_{t_i}^o | \boldsymbol{x}_{1:t_i}^o, \boldsymbol{y}_{1:t_{i-1}}^o, \boldsymbol{x}_{t_1^m:t_g^m}^{m,(k)}, \phi_y^{(k)}) \pi(\boldsymbol{x}_{t_i}^o | \boldsymbol{x}_{1:t_{i-1}}^o, \boldsymbol{y}_{1:t_{i-1}}^o, \phi_x^{(k)}),$$

   $\boldsymbol{m}_{t_i}^{(k)}$ and $\boldsymbol{C}_{t_i}^{(k)}$. Note the convention that

   $$\pi(\boldsymbol{y}_1^o | \boldsymbol{x}_1^o, \boldsymbol{x}_1^{m,(k)}, \phi_y^{(k)}) \pi(\boldsymbol{x}_1^o | \phi_x^{(k)}) = \pi(\boldsymbol{y}_1^o | \boldsymbol{x}_{1:1}^o, \boldsymbol{y}_{1:t_0}^o, \boldsymbol{x}_1^{m,(k)}, \phi_y^{(k)}) \pi(\boldsymbol{x}_1^o | \boldsymbol{x}_{1:0}^o, \boldsymbol{y}_{1:0}^o, \phi_x^{(k)}).$$

   (c) Update and normalise the importance weights using

   $$\tilde{\omega}_{t_i}^{(k)} = \tilde{\omega}_{t_{i-1}}^{(k)} \pi(\boldsymbol{y}_{t_i}^o | \boldsymbol{x}_{1:t_i}^o, \boldsymbol{y}_{1:t_{i-1}}^o, \boldsymbol{x}_{t_1^m:t_g^m}^{m,(k)}, \phi_y^{(k)}) \pi(\boldsymbol{x}_{t_i}^o | \boldsymbol{x}_{1:t_{i-1}}^o, \boldsymbol{y}_{1:t_{i-1}}^o, \phi_x^{(k)})$$

   and $\omega_{t_i}^{(k)} = \tilde{\omega}_{t_i}^{(k)} / \sum_{j=1}^N \tilde{\omega}_{t_i}^{(j)}$.

   (d) Update the observed data likelihood using

   $$\begin{aligned}\pi(\boldsymbol{x}_{1:t_i}^o, \boldsymbol{y}_{1:t_i}^o | \phi_x^{(k)}, \phi_y^{(k)}, \boldsymbol{x}_{t_1^m:t_g^m}^{m,(k)}) =& \pi(\boldsymbol{x}_{1:t_{i-1}}^o, \boldsymbol{y}_{1:t_{i-1}}^o | \phi_x^{(k)}, \phi_y^{(k)}) \times \\ & \pi(\boldsymbol{y}_{t_i}^o | \boldsymbol{x}_{1:t_i}^o, \boldsymbol{y}_{1:t_{i-1}}^o, \boldsymbol{x}_{t_1^m:t_g^m}^{m,(k)}, \phi_y^{(k)}) \pi(\boldsymbol{x}_{t_i}^o | \boldsymbol{x}_{1:t_{i-1}}^o, \boldsymbol{y}_{1:t_{i-1}}^o, \phi_x^{(k)}).\end{aligned}$$

4. If ESS $< \delta N$ resample and move as follows. For $k = 1, \ldots, N$:

   (a) Sample indices $a_k \sim \mathcal{M}(\omega^{1:N})$ and set $\boldsymbol{m}_{t_i}^{(k)} := \boldsymbol{m}_{t_i}^{(a_k)}$ and $\boldsymbol{C}_{t_i}^{(k)} := \boldsymbol{C}_{t_i}^{(a_k)}$, $\{\phi_x^{(k)}, \phi_y^{(k)}, \boldsymbol{x}_{t_1^m:t_g^m}^{m,(k)}, \tilde{\omega}_{t_i}^{(k)}\} := \{\phi_x^{(a_k)}, \phi_y^{(a_k)}, \boldsymbol{x}_{t_1^m:t_g^m}^{m,(a_k)}, 1\}$ and $\pi(\boldsymbol{x}_{1:t_i}^o, \boldsymbol{y}_{1:t_i}^o | \phi_x^{(k)}, \phi_y^{(k)}, \boldsymbol{x}_{t_1^m:t_g^m}^{m,(k)}) = \pi(\boldsymbol{x}_{1:t_i}^o, \boldsymbol{y}_{1:t_i}^o | \phi_x^{(a_k)}, \phi_y^{(a_k)}, \boldsymbol{x}_{t_1^m:t_g^m}^{m,(a_k)})$.

   (b) Propose $\phi_x^*, \phi_y^*, \boldsymbol{x}_{t_1^m:t_g^m}^{m,*} \sim q(\cdot | \phi_x^{(k)}, \phi_y^{(k)}, \boldsymbol{x}_{t_1^m:t_g^m}^{m,(k)})$. Perform iterations $1, \ldots, i$ of the forward filter to obtain $\pi(\boldsymbol{x}_{1:t_i}^o, \boldsymbol{y}_{1:t_i}^o | \phi_x^*, \phi_y^*, \boldsymbol{x}_{t_1^m:t_g^m}^{m,*})$. With probability

   $$\min\left\{1, \frac{\pi(\phi_x^*, \phi_y^*, \boldsymbol{x}_{t_1^m:t_g^m}^{m,*}) \pi(\boldsymbol{x}_{1:t_i}^o | \phi_x^*, \phi_y^*, \boldsymbol{x}_{t_1^m:t_g^m}^{m,*})}{\pi(\phi_x^{(k)}, \phi_y^{(k)}, \boldsymbol{x}_{t_1^m:t_g^m}^{m,(k)}) \pi(\boldsymbol{x}_{1:t_i}^o | \phi_x^{(k)}, \phi_y^{(k)}, \boldsymbol{x}_{t_1^m:t_g^m}^{m,(k)})} \times \frac{q(\phi_x^{(k)}, \phi_y^{(k)}, \boldsymbol{x}_{t_1^m:t_g^m}^{m,(k)} | \phi_x^*, \phi_y^*, \boldsymbol{x}_{t_1^m:t_g^m}^{m,*})}{q(\phi_x^*, \phi_y^*, \boldsymbol{x}_{t_1^m:t_g^m}^{m,*} | \phi_x^{(k)}, \phi_y^{(k)}, \boldsymbol{x}_{t_1^m:t_g^m}^{m,(k)})}\right\},$$

   put $(\phi_x^{(k)}, \phi_y^{(k)}, \boldsymbol{x}_{t_1^m:t_g^m}^{m,(k)}) := (\phi_x^*, \phi_y^*, \boldsymbol{x}_{t_1^m:t_g^m}^{m,*})$, $\pi(\boldsymbol{x}_{1:t_i}^o, \boldsymbol{y}_{1:t_i}^o | \phi_x^{(k)}, \phi_y^{(k)}, \boldsymbol{x}_{t_1^m:t_g^m}^{m,(k)}) := \pi(\boldsymbol{x}_{1:t_i}^o, \boldsymbol{y}_{1:t_i}^o | \phi_x^*, \phi_y^*, \boldsymbol{x}_{t_1^m:t_g^m}^{m,*})$, $\boldsymbol{m}_{t_i}^{(k)} := \boldsymbol{m}_{t_i}^*$ and $\boldsymbol{C}_{t_i}^{(k)} := \boldsymbol{C}_{t_i}^*$.

Algorithm 18 IBIS algorithm for missing temperature with expectation replacement

1. Initialisation. For $k = 1,\dots,N$ sample $\phi_x^{(k)}, \phi_y^{(k)} \sim \pi(\cdot)$ and set $\tilde{\omega}_0^{(k)} := 1$. Store $\boldsymbol{m}_0^{(k)}$ and $\boldsymbol{C}_0^{(k)}$. For $i = 1,\dots,n$, implement the IBIS scheme (Algorithm 15). If temperature data are missing but the corresponding humidity are not, go to step 2.

2. Sequential importance sampling. For $k = 1,\dots,N$:

   (a) Sample $\boldsymbol{x}_{t_i}^{m,(k)} \sim \pi(\boldsymbol{x}_{t_i}^m | \boldsymbol{x}_{1:t_{i-1}}^o, \phi_x^{(k)})$.

   (b) Perform iteration i of the forward filter to obtain

   $$\pi(\boldsymbol{y}_{t_i}^o | \boldsymbol{x}_{1:t_i}^o, \boldsymbol{y}_{1:t_{i-1}}^o, \boldsymbol{x}_{t_i}^{m,(k)}, \phi_y^{(k)}) \pi(\boldsymbol{x}_{t_i}^o | \boldsymbol{x}_{1:t_{i-1}}^o, \boldsymbol{y}_{1:t_{i-1}}^o, \phi_x^{(k)}),$$

   $\boldsymbol{m}_{t_i}^{(k)}$ and $\boldsymbol{C}_{t_i}^{(k)}$. Note the convention that

   $$\pi(\boldsymbol{y}_1^o | \boldsymbol{x}_1^o, \boldsymbol{x}_1^{m,(k)}, \phi_y^{(k)}) \pi(\boldsymbol{x}_1^o | \phi_x^{(k)}) = \pi(\boldsymbol{y}_1^o | \boldsymbol{x}_{1:1}^o, \boldsymbol{y}_{1:t_0}^o, \boldsymbol{x}_1^{m,(k)}, \phi_y^{(k)}) \pi(\boldsymbol{x}_1^o | \boldsymbol{x}_{1:0}^o, \boldsymbol{y}_{1:0}^o, \phi_x^{(k)}).$$

   (c) Update and normalise the importance weights using

   $$\tilde{\omega}_{t_i}^{(k)} = \tilde{\omega}_{t_{i-1}}^{(k)} \pi(\boldsymbol{y}_{t_i}^o | \boldsymbol{x}_{1:t_i}^o, \boldsymbol{y}_{1:t_{i-1}}^o, \boldsymbol{x}_{t_i}^{m,(k)}, \phi_y^{(k)}) \pi(\boldsymbol{x}_{t_i}^o | \boldsymbol{x}_{1:t_{i-1}}^o, \boldsymbol{y}_{1:t_{i-1}}^o, \phi_x^{(k)})$$

   and $\omega_{t_i}^{(k)} = \tilde{\omega}_{t_i}^{(k)} / \sum_{j=1}^N \tilde{\omega}_{t_i}^{(j)}$.

   (d) Update the observed data likelihood using

   $$\begin{aligned}\pi(\boldsymbol{x}_{1:t_i}^o, \boldsymbol{y}_{1:t_i}^o | \phi_x^{(k)}, \phi_y^{(k)}, \boldsymbol{x}_{t_i}^{m,(k)}) =& \pi(\boldsymbol{x}_{1:t_{i-1}}^o, \boldsymbol{y}_{1:t_{i-1}}^o | \phi_x^{(k)}, \phi_y^{(k)}) \times \\ & \pi(\boldsymbol{y}_{t_i}^o | \boldsymbol{x}_{1:t_i}^o, \boldsymbol{y}_{1:t_{i-1}}^o, \boldsymbol{x}_{t_i}^{m,(k)}, \phi_y^{(k)}) \pi(\boldsymbol{x}_{t_i}^o | \boldsymbol{x}_{1:t_{i-1}}^o, \boldsymbol{y}_{1:t_{i-1}}^o, \phi_x^{(k)}).\end{aligned}$$

3. If ESS $< \delta N$ resample and move as follows. For $k = 1,\dots,N$:

   (a) Sample indices $a_k \sim \mathcal{M}(\omega^{1:N})$ and set $\boldsymbol{m}_{t_i}^{(k)} := \boldsymbol{m}_{t_i}^{(a_k)}$ and $\boldsymbol{C}_{t_i}^{(k)} := \boldsymbol{C}_{t_i}^{(a_k)}$, $\{\phi_x^{(k)}, \phi_y^{(k)}, \boldsymbol{x}_{t_i}^{m,(k)}, \tilde{\omega}_{t_i}^{(k)}\} := \{\phi_x^{(a_k)}, \phi_y^{(a_k)}, \boldsymbol{x}_{t_i}^{m,(a_k)}, 1\}$ and $\pi(\boldsymbol{x}_{1:t_i}^o, \boldsymbol{y}_{1:t_i}^o | \phi_x^{(k)}, \phi_y^{(k)}, \boldsymbol{x}_{t_i}^{m,(k)}) = \pi(\boldsymbol{x}_{1:t_i}^o, \boldsymbol{y}_{1:t_i}^o | \phi_x^{(a_k)}, \phi_y^{(a_k)}, \boldsymbol{x}_{t_i}^{m,(a_k)})$.

   (b) Propose $\phi_x^*, \phi_y^*, \boldsymbol{x}_{t_i}^{m,*} \sim q(\cdot | \phi_x^{(k)}, \phi_y^{(k)}, \boldsymbol{x}_{t_i}^{m,(k)})$. Perform iterations $1,\dots,i$ of the forward filter to obtain $\pi(\boldsymbol{x}_{1:t_i}^o, \boldsymbol{y}_{1:t_i}^o | \phi_x^*, \phi_y^*, \boldsymbol{x}_{t_i}^{m,*})$. With probability

   $$\min\left\{1, \frac{\pi(\phi_x^*, \phi_y^*, \boldsymbol{x}_{t_i}^{m,*}) \pi(\boldsymbol{x}_{1:t_i}^o | \phi_x^*, \phi_y^*, \boldsymbol{x}_{t_i}^{m,*})}{\pi(\phi_x^{(k)}, \phi_y^{(k)}, \boldsymbol{x}_{t_i}^{m,(k)}) \pi(\boldsymbol{x}_{1:t_i}^o | \phi_x^{(k)}, \phi_y^{(k)}, \boldsymbol{x}_{t_i}^{m,(k)})} \times \frac{q(\phi_x^{(k)}, \phi_y^{(k)}, \boldsymbol{x}_{t_i}^{m,(k)} | \phi_x^*, \phi_y^*, \boldsymbol{x}_{t_i}^{m,*})}{q(\phi_x^*, \phi_y^*, \boldsymbol{x}_{t_i}^{m,*} | \phi_x^{(k)}, \phi_y^{(k)}, \boldsymbol{x}_{t_i}^{m,(k)})}\right\},$$

   put $(\phi_x^{(k)}, \phi_y^{(k)}, \boldsymbol{x}_{t_i}^{m,(k)}) := (\phi_x^*, \phi_y^*, \boldsymbol{x}_{t_i}^{m,*})$, $\pi(\boldsymbol{x}_{1:t_i}^o, \boldsymbol{y}_{1:t_i}^o | \phi_x^{(k)}, \phi_y^{(k)}, \boldsymbol{x}_{t_i}^{m,(k)}) := \pi(\boldsymbol{x}_{1:t_i}^o, \boldsymbol{y}_{1:t_i}^o | \phi_x^*, \phi_y^*, \boldsymbol{x}_{t_i}^{m,*})$, $\boldsymbol{m}_{t_i}^{(k)} := \boldsymbol{m}_{t_i}^*$ and $\boldsymbol{C}_{t_i}^{(k)} := \boldsymbol{C}_{t_i}^*$.

4. Calculate the expectation $\bar{\boldsymbol{x}}_{t_i}^m = \sum_{j=1}^N \boldsymbol{x}_{t_i}^{m,(j)} / N$, replace missing temperature by $\bar{\boldsymbol{x}}_{t_i}^m$.

Chapter 6

Results

6.1 Simulation study

In order to assess the performance of the proposed online IBIS scheme and the effect of local resampling, we looked at results from synthetic data generated from the marginal model for temperature in (5.5). We consider two spatial locations (giving 14 parameters in total) and simulated $n = 1300$ observations at each location. The true parameter values used to produce the synthetic data are $W_k^j = 0.01$, $V^j = \sigma_k^2 = 1$ and $\psi_k = 0.01$ for $j = 1,2$ and $k = 1,2,3$. As this is a data-rich scenario, we assumed very weak independent inverse gamma $IG(1,0.01)$ prior distributions for all these parameter components, but truncated them above at 10 as values in excess of 10 are far from plausible. We also took the prior distribution for the initial system state as $\boldsymbol{\theta}_0 \sim N(\boldsymbol{m}, \boldsymbol{C})$, where $\boldsymbol{m} = (0,0,17,0,0,17)^T$ and $\boldsymbol{C} = \mathbb{I}_6$. We used 10^7 particles and an ESS threshold of $\delta = 0.5$ for triggering the resample-move step. All computer code was written in C and executed on a high performance cluster with Intel Xeon E5-2699 v4 processors (2.2 GHz, 55 MB cache), where each node has two processors and each processor has 22 cores (2.9 GB CPU memory per core).

6.1.1 Comparison of full IBIS with serial resampling and parallelised local resampling

We consider first two parallelised implementations of the full IBIS scheme: (i) weighting and move steps are performed in parallel over 22 cores through a shared memory system (within one processor) with the resampling step performed in serial; (ii) particles are divided over 200 cores and local resampling is used. Figure 6.1 shows the parameter marginal posterior densities obtained by using method 1 (IBIS with serial resampling) and method 2 (IBIS with parallelised local resampling) together with the true values. It is clear that both approaches give posterior output consistent with the true values (used to simulate the data). Moreover, the posterior densities from the fully parallelised method 2 match up well with those from the exact (simulation based) method 1. However the run time for method 1 (IBIS with serial resampling) is around 23 hours whereas that for method 2 (IBIS with parallelised local resampling) is around 4 hours, a six-fold speed-up.

6.1.2 Comparison of full IBIS and online IBIS

We now compare the full IBIS scheme with online IBIS and in both schemes we use the parallelised local resampling method. For online IBIS, we consider three widths for the fixed window: $T = 100$, 300 and 500. Figure 6.2 shows the output of the marginal posterior densities from the online IBIS scheme for each window size, together with the densities from the full IBIS scheme. As expected, as the larger window increases, so does posterior accuracy. The marginal posteriors from online IBIS using $T = 300$ and $T = 500$ almost overlay those from full IBIS. However, there are noticeable differences when using $T = 100$. In terms of computational efficiency, online IBIS with both $T = 300$ and $T = 500$ take roughly 2 CPU hours, that with $T = 100$ takes approximately 1 CPU hour. Consequently, for this example, online IBIS with $T = 300$ and local parallel resampling gives an overall reduction in computational cost of around a factor of 12 compared to full IBIS with serial resampling.

6.2 Real data study

Now we analyse the data on hourly average temperature and humidity values introduced in Section 5.1. Recall that these data are measurements recorded during the period 8th July 2017 to

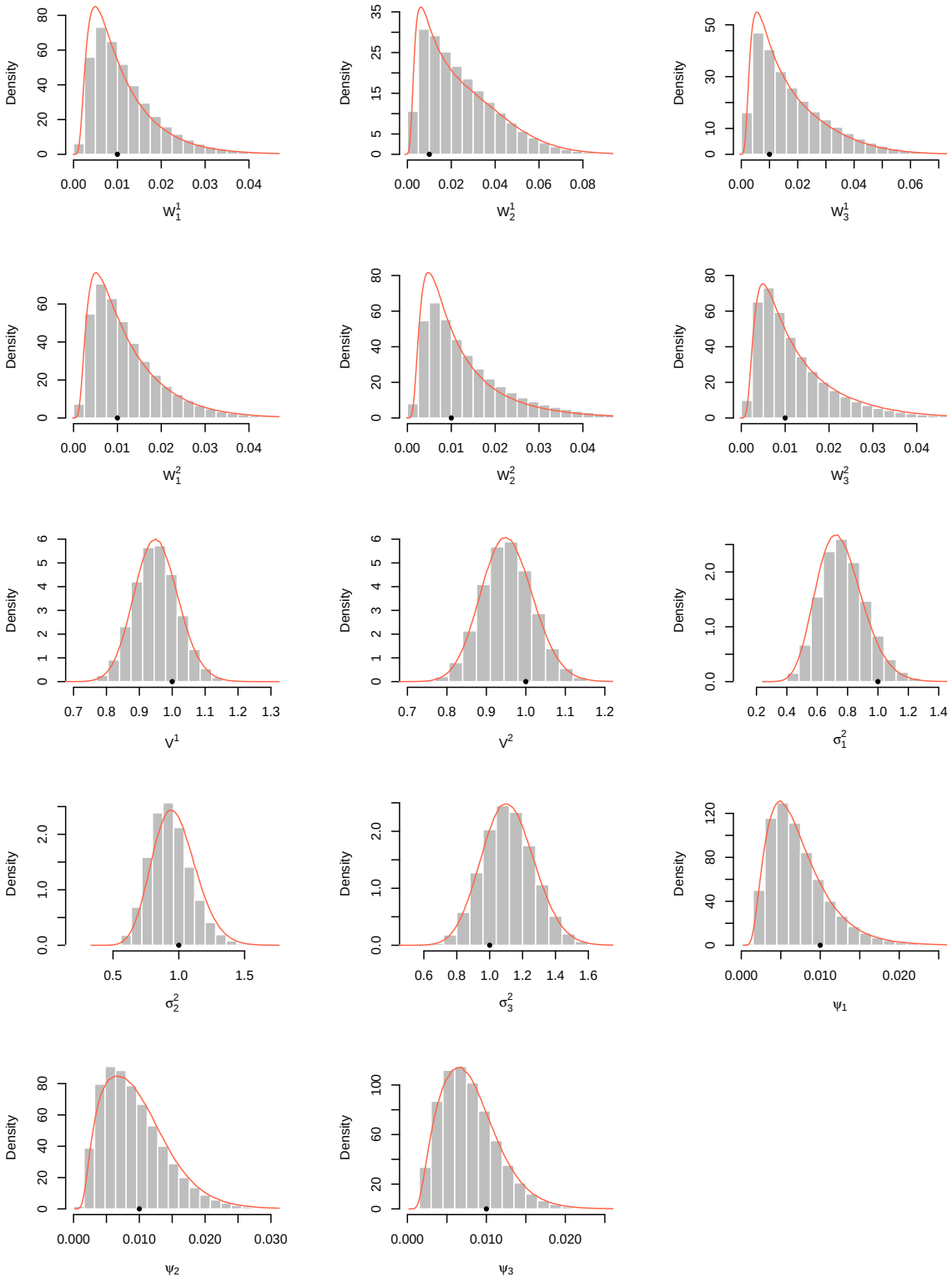

Figure 6.1: Marginal parameter posterior densities obtained from the output of the full IBIS scheme with a standard serial resampling step (histograms) and a parallelised local resampling step (red). The true parameter values are shown as solid circles.

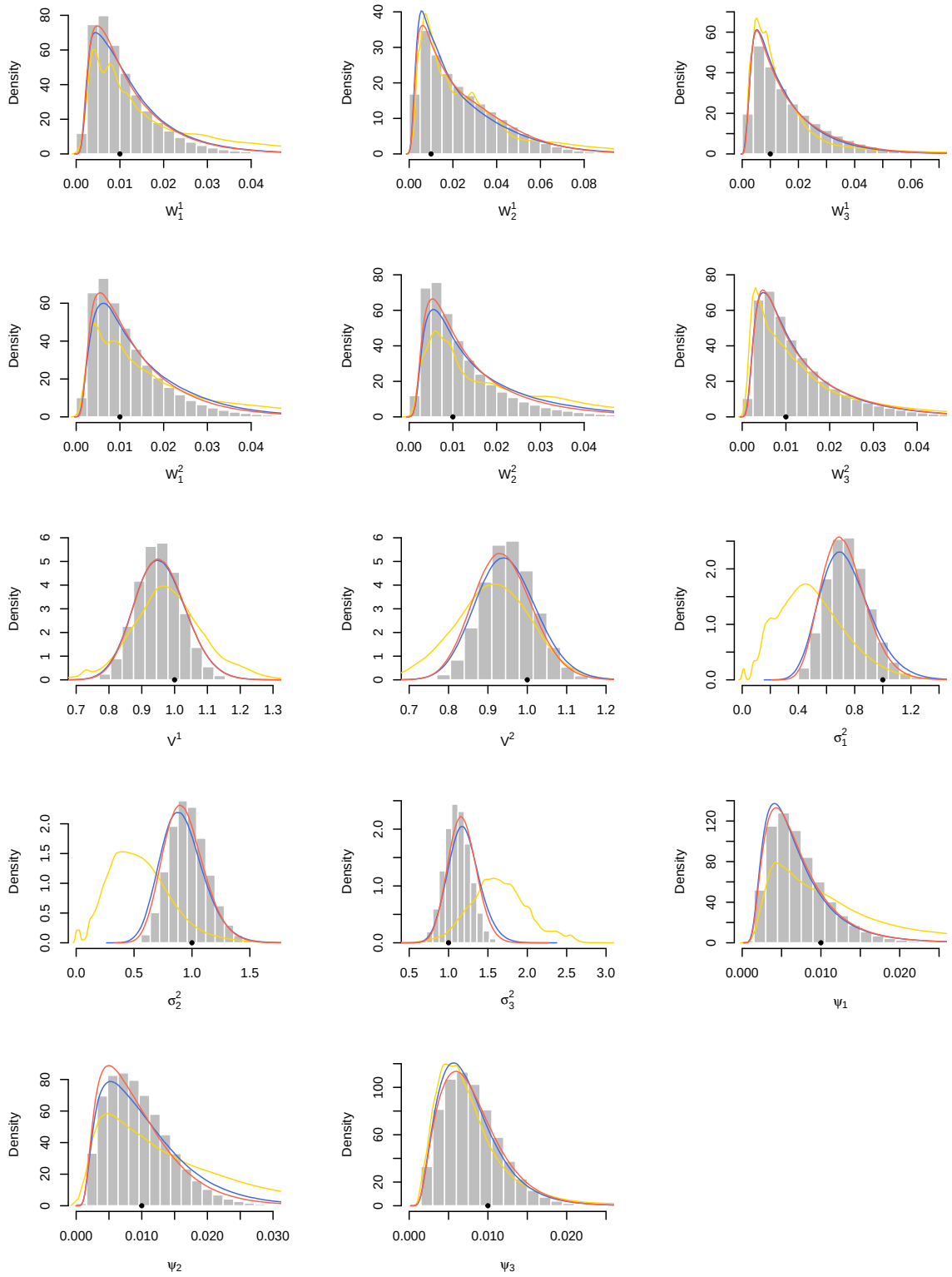

Figure 6.2: Marginal parameter posterior densities obtained from the output of the full IBIS scheme (histograms) and the online IBIS scheme with window widths $T = 100$ (yellow), $T = 300$ (blue) and $T = 500$ (red). The true parameter values are shown as solid circles.

31st December 2017 and that the observations are irregularly spaced due to network and sensor failures. We take independent inverse gamma $IG(1, 0.01)$ prior distributions, truncated above at 10, for all the static parameters in both temperature and humidity DLMs. To incorporate our prior belief that the underlying system should be smoother than the observation process, we also impose the constraint that at each location $j = 1, \ldots, 5$, $W_i^{x,j} < V^{x,j}$ $(i = 1, 2, 3)$ and $W_k^{y,j} < V^{y,j}$ $(k = 1, 2)$. We ran the online IBIS scheme with $N = 10^7$ particles, fully parallelised (with local resampling) over 200 cores using an ESS threshold of $\delta = 0.5$. Regular particle rejuvenation steps were set up for the process at every 20 time points, and the resample-move step was executed in any batch whose ESS fell below half the number of particles (in the batch). Finally, to balance accuracy and computational efficiency, we used a window width of $T = 1500$, and this gave a run time of approximately 9.5 days. On average, it took around 3 minutes to assimilate an observation with hourly frequency.

6.2.1 Posterior output

Table 6.1 shows the marginal posterior medians and quantile-based 95% credible intervals for the static parameters in the joint temperature and humidity model. These summaries were obtained from output of the online IBIS scheme. Inspection of the posterior medians for the system variances (governing both temperature and humidity models) reveals that these components are larger at location 1 (Newcastle) than at the other locations. This is perhaps not surprising given that location 1 has the largest fraction of missing data (see Table 5.1). Also sampled posterior values of the observation variance components $V^{x,j}$ and $V^{y,j}$ are generally very much larger at location 2 (Seaham), and this too is consistent with the simple data summaries in Table 5.1 – Seaham is the least spatially consistent location in terms of median temperature and humidity. Variation across sites is accounted for by the elements of $\boldsymbol{\sigma}^2$. The relatively large values of $\sigma^2_{x,3}$ and $\sigma^2_{y,2}$ suggest that there is some spatial inconsistency in the dynamically varying mean level components $\theta^{x,j}_{t_i,3}$ and $\theta^{y,j}_{t_i,2}$. Spatial consistency of these mean level components can be assessed further by noting that

$$\mathrm{Cor}(\theta^{x,j}_{t_i,3}, \theta^{x,j'}_{t_i,3}) = \exp(-\psi_{x,3} d_{jj'}), \qquad \mathrm{Cor}(\theta^{y,j}_{t_i,2}, \theta^{y,j'}_{t_i,2}) = \exp(-\psi_{y,2} d_{jj'}).$$

Hence, fixing $\psi_{x,3}$ and $\psi_{y,2}$ at their posterior medians gives a simple linear relationship between distance and log correlation. For example, within a 10km radius from each location, there is a spatial correlation of at least 0.76 for temperature and 0.64 for humidity. These areas are displayed in Figure 6.3. We note that it is not surprising that spatial correlation for humidity is lower than that for temperature, as the humidity records are also easily influenced by other

Temperature				Humidity			
ϕ_x	Median	2.5%	97.5%	ϕ_y	Median	2.5%	97.5%
$W_1^{x,1}$	0.0050	0.0011	0.0110	$W_1^{y,1}$	0.0156	0.0118	0.0208
$W_2^{x,1}$	0.0056	0.0013	0.0114	$W_2^{y,1}$	0.0074	0.0019	0.0183
$W_3^{x,1}$	0.0053	0.0014	0.0116	$W_1^{y,2}$	0.0071	0.0049	0.0102
$W_1^{x,2}$	0.0026	0.0008	0.0089	$W_2^{y,2}$	0.0072	0.0018	0.0183
$W_2^{x,2}$	0.0031	0.0008	0.0095	$W_1^{y,3}$	0.0024	0.0014	0.0038
$W_3^{x,2}$	0.0039	0.0009	0.0096	$W_2^{y,3}$	0.0048	0.0015	0.0144
$W_1^{x,3}$	0.0021	0.0006	0.0082	$W_1^{y,4}$	0.0032	0.0017	0.0054
$W_2^{x,3}$	0.0023	0.0006	0.0075	$W_2^{y,4}$	0.0050	0.0016	0.0156
$W_3^{x,3}$	0.0021	0.0006	0.0083	$W_1^{y,5}$	0.0020	0.0010	0.0035
$W_1^{x,4}$	0.0027	0.0007	0.0083	$W_2^{y,5}$	0.0049	0.0016	0.0148
$W_2^{x,4}$	0.0032	0.0007	0.0095	$V^{y,1}$	0.0265	0.0147	0.0826
$W_3^{x,4}$	0.0036	0.0009	0.0102	$V^{y,2}$	0.4520	0.3362	0.5822
$W_1^{x,5}$	0.0042	0.0008	0.0103	$V^{y,3}$	0.0201	0.0137	0.0382
$W_2^{x,5}$	0.0026	0.0007	0.0089	$V^{y,4}$	0.0199	0.0137	0.0383
$W_3^{x,5}$	0.0038	0.0007	0.0092	$V^{y,5}$	0.0190	0.0134	0.0331
$V^{x,1}$	0.0089	0.0047	0.0173	$\sigma^2_{y,1}$	0.0257	0.0209	0.0315
$V^{x,2}$	0.0230	0.0110	0.0419	$\sigma^2_{y,2}$	1.6054	1.4961	1.7228
$V^{x,3}$	0.0078	0.0044	0.0138	$\psi_{y,1}$	0.0016	0.0008	0.0029
$V^{x,4}$	0.0088	0.0049	0.0251	$\psi_{y,2}$	0.0447	0.0388	0.0511
$V^{x,5}$	0.0164	0.0061	0.0380				
$\sigma^2_{x,1}$	0.0423	0.0105	0.1611				
$\sigma^2_{x,2}$	0.0627	0.0250	0.1672				
$\sigma^2_{x,3}$	0.2310	0.0837	0.2706				
$\psi_{x,1}$	0.0014	0.0004	0.0496				
$\psi_{x,2}$	0.0013	0.0004	0.0606				
$\psi_{x,3}$	0.0274	0.0011	0.0354				

Table 6.1: Marginal parameter posterior medians and quantile-based 95% credible intervals obtained from the output of the online IBIS scheme.

factors, such as urban structure and distance from the sea, in addition to temperature.

6.2.2 Predictive checks

We assess the validity of the proposed model by comparing observed data with their model-based within-sample posterior predictive distributions and with model-based out-of-sample forecast distributions. Simulation methods can be used to construct these distributions and details on how

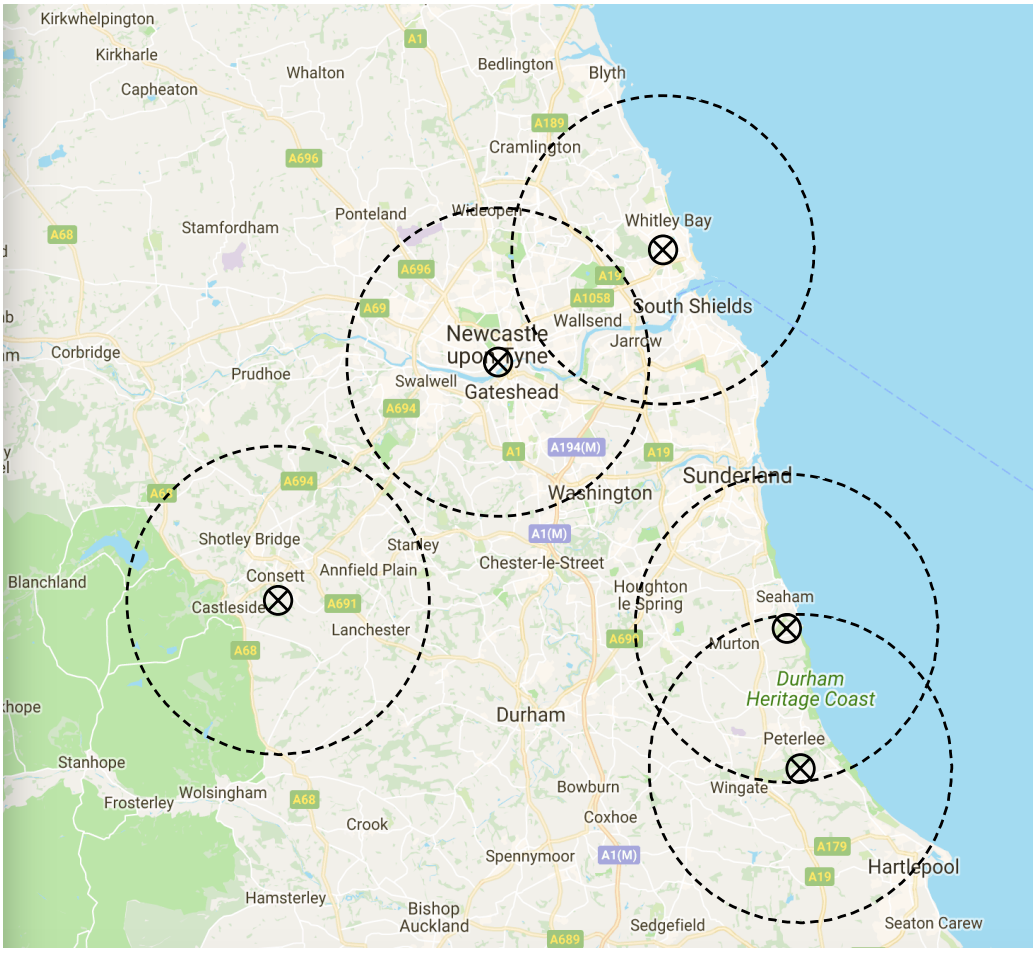

Figure 6.3: Map showing site locations and a 10 km radius from each site, within which the spatial correlation for temperature is at least 0.76, and for humidity, is at least 0.64.

to generate draws from them is provided in Section 5.4. Figure 6.4 shows discrepancies between observations and their within-sample predictive distribution over the first 500 hours at each of the 5 locations. These distributions are characterised by their mean and 95% credible interval. It is clear that the mean difference at each time-location combination is small and that a mean difference of zero is plausible (the 95% credible intervals include zero). Similar results were obtained for the full data set (not shown).

Figure 6.5 shows the mean and 95% credible interval at each location for the one-step ahead forecast. The times displayed were chosen at random over a two day period and, for comparison purposes, the observations at these times are also shown. Unsurprisingly forecast uncertainty grows during periods of prolonged missingness. The figure shows that observations typically lie within the forecast interval and that the model-based one-step forecast distribution is consistent with the observed data. Figure 6.6 shows the mean and 95% credible interval at each location for the two-step ahead forecast. Similar to the one-step forecasts, this figure shows that these

forecast distributions are consistent with the data but, of course, have larger uncertainty.

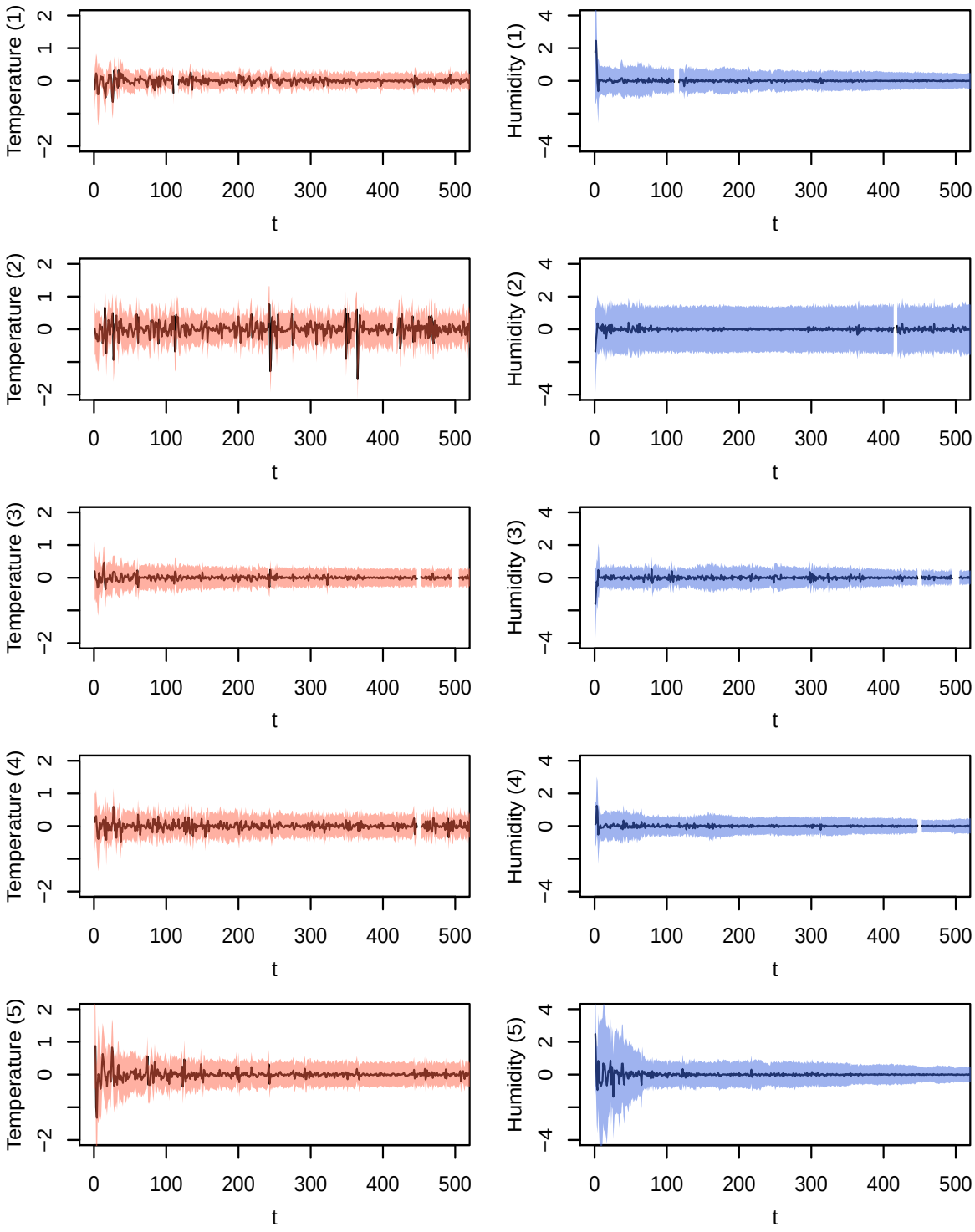

Figure 6.4: Mean (——) and 95% credible interval for the difference between the within-sample predictive and the observations, at each location (1–5) over time. The observation period is from 8th July 2017 04:00:00 to 29th July 2017 00:00:00.

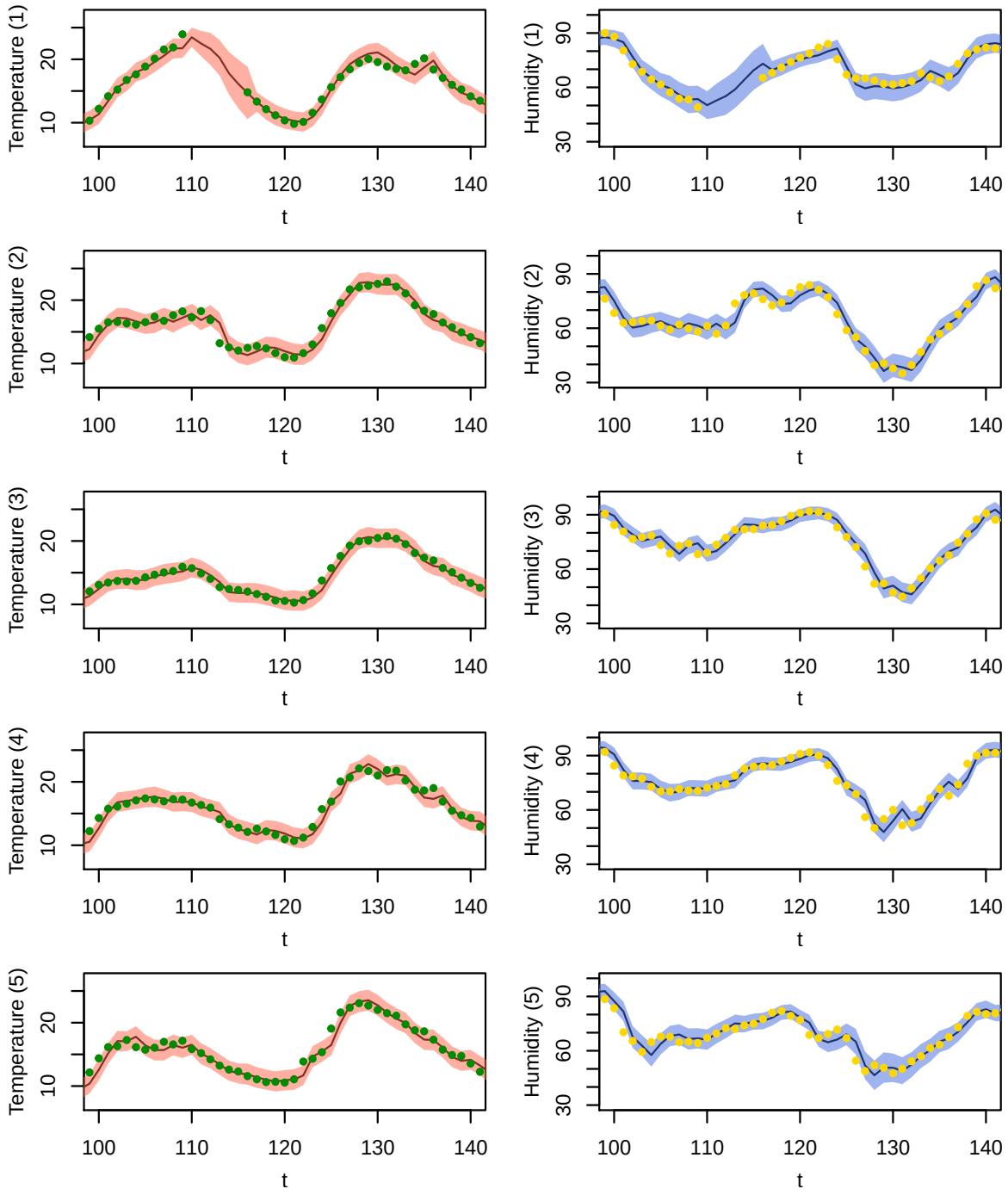

Figure 6.5: One-step ahead forecast mean (——) and 95% credible interval, at each location (1–5) over time. The observations are indicated (•). The observation period is from 12th July 2017 08:00:00 to 14th July 2017 00:00:00.

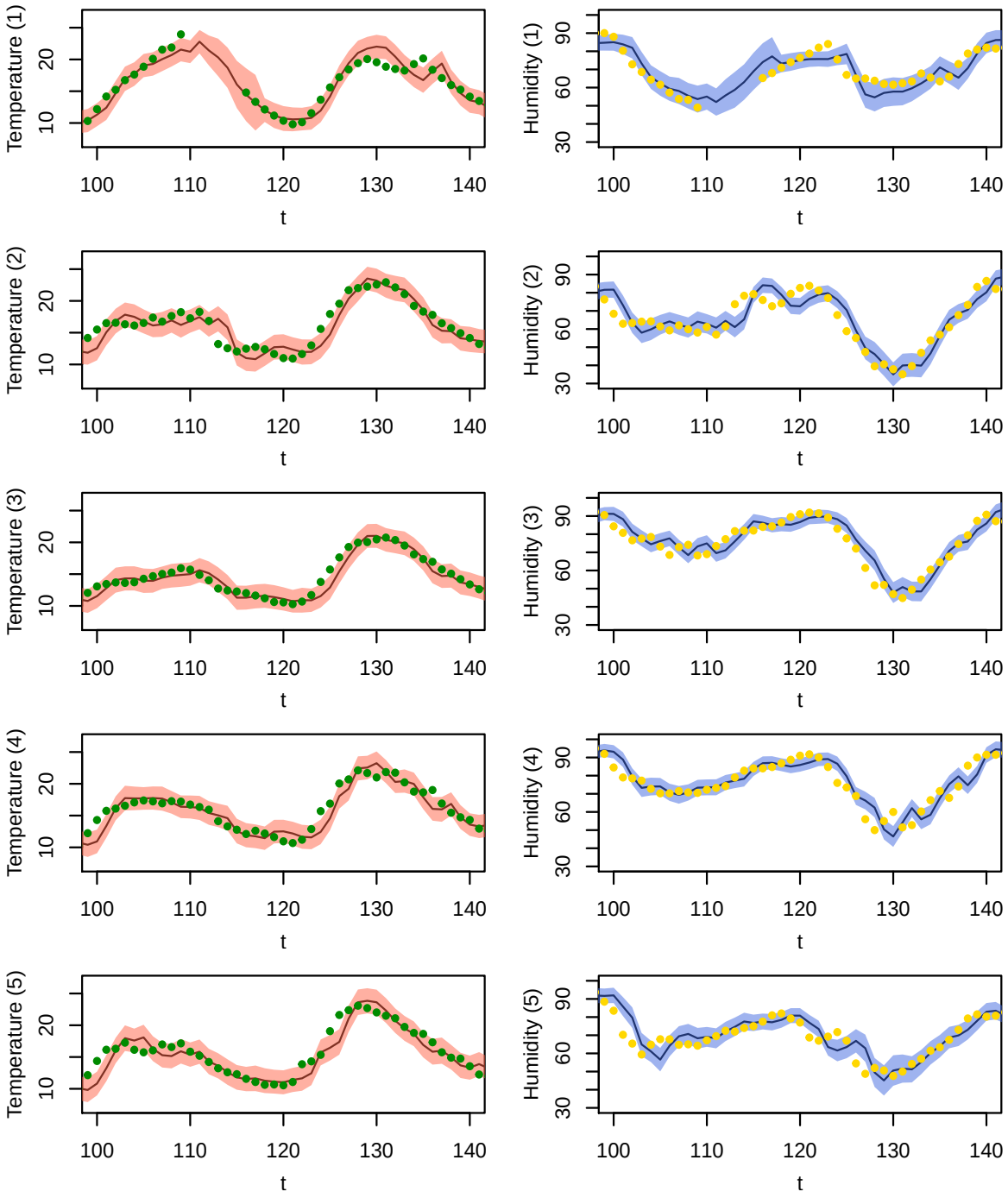

Figure 6.6: Two-step ahead forecast mean (——) and 95% credible interval, at each location (1–5) over time. The observations are indicated (•). The observation period is from 12th July 2017 08:00:00 to 14th July 2017 00:00:00.

Chapter 7

Conclusions

The objective of this thesis was to develop dynamic models for fitting to large spatial sensor streams and propose an efficient sequential Bayesian inference procedure for parameter learning in real time. Dynamic linear models (DLMs) are widely used in practice, not only because of their simple structure, but also their flexibility for handling different non-stationary time series. We reviewed Markov chain Monte Carlo (MCMC) methods in Chapter 2, which are a collection of computationally intensive algorithms typically used for posterior sampling. These methods are inefficient in the context of sequential learning, due to the requirement of having to restart the scheme upon receipt of new data. We therefore explored the use of sequential Monte Carlo (SMC), which allows for the posterior density to be updated sequentially through a sequence of propagation and reweighting steps. However, SMC schemes can easily suffer from a particle degeneracy issue. That is, the posterior sample can collapse to a point mass. We investigated this issue and compared the performance of different SMC schemes via simulation studies in Chapter 4. The Liu-West algorithm (Liu and West, 2001) deals with particle degeneracy by adding artificial noise to each parameter particle. The Storvik algorithm (Storvik, 2002) and particle learning (Carvalho et al., 2010; Lopes et al., 2011) exploit the tractability of the conditional parameter posterior to maintain a diverse particle set. However, as demonstrated by the simulation study, these methods are unable to completely overcome degeneracy. The tractability of the observed data likelihood allows us to construct the SMC algorithm using an iterated batch importance sampling (IBIS) scheme, first introduced by Chopin (2002). The IBIS scheme tries to deal with particle degeneracy by employing a resample-move (rejuvenation) step which allows the particle set to be rejuvenated by moving each particle through a Metropolis-Hastings kernel that leaves the target posterior invariant. Its adaptive version (Fearnhead and Taylor, 2013) allows the scaling of the tuning parameter within the random walk proposal to

be a random variable in each rejuvenation step. However, we discovered that the adaptive IBIS (aIBIS) scheme lacked practicality, as the CPU cost was almost doubled when comparing with the IBIS scheme, yet gains in posterior accuracy were minimal.

In Chapter 5 and 6, we developed and fitted a spatio-temporal model to around six months of data on hourly temperature and humidity values at five locations in the North East of England. The data were obtained from a sensor network providing streaming data on environmental variables such as climate, pollution and traffic flow, held at the Newcastle Urban Observatory. The model we used for observed seasonality in temperature is a dynamic linear model (DLM) whose observation equation takes the form of a sinusoid, with time varying amplitude and phase described by the system equation. We captured the observed linear relationship between humidity and temperature via a conditional DLM in which humidity is regressed on temperature. Spatial consistency at nearby sites is accounted for by adding a Gaussian process in the system equation. Our primary goal was real time forecasting of temperature and humidity. To this end, we applied the IBIS scheme for updating the parameter posterior as each measurement becomes available. An issue of IBIS is that the computational cost of the resample-move step increases as the algorithm runs, due to the time taken to calculate the observed data likelihood at each particle, as more data is included. This problem is made much more acute by the long length of the observed time series and the high dimension of the parameter space and this makes the algorithm unusable as an online algorithm. To circumvent this issue, we modified the resample-move step in two ways. First, we used a sequence of observation windows and calculated the observed data likelihood for the data within the window. As the data in each window is included, the parameter posterior (at the start of the window) is approximated using a kernel density estimate and then updated using the observed data likelihood for the window. This places an upper bound on the computational cost. We looked at the effect of the choice of window length on computational efficiency and posterior accuracy and found that reasonable posterior accuracy could be achieved for a modest window length. Finally, we increased the computational efficiency of the algorithm by using a fully parallel implementation which divides the particles into batches and performs the resampling step locally, for each batch. In this case, the parallelisation in a distributed memory system could be readily applied through a Message Passing Interface (MPI). We termed the resulting scheme *online IBIS* and found that for our data set, an observation (consisting of both temperature and humidity hourly averages at each of five locations) could be assimilated in around 3 minutes on average, with this average time dominated by the rejuvenation steps. One-step and two-step forecast distributions could then be determined very quickly. Given that observations arrive every hour, this made the scheme entirely feasible for use in real time.

This work can be extended in a number of ways. For example, covariate information such as

altitude, distance from the coast and wind direction/speed could be included in the spatial model, in order to improve the precision of the forecasts. Unfortunately this information is not currently available. For further enhancing the computational performance, we can investigate the feasibility of applying hybrid parallel computing technics (Rabenseifner et al., 2009) through MPI and OpenMP to the model simultaneously, i.e. in addition to the parallel runs for the disjoint particle subsets in a distributed memory system regarding the local resampling scheme, we could also run the algorithm for each particle subset in a shared memory system through the OpenMP directive at the same time. Moreover, the model can be generalised to incorporate spatial correlations for more sensor streams from multiple locations, such as pollution data and traffic data. The different variables may have different features, and that will prompt us to consider other model structures including the general state space models with non-linear structures to fit those data. We can consider the method of SMC^2 (Chopin et al., 2013) for updating the posterior of fixed parameters to adapt to dynamic models where the marginal likelihood is intractable. Finally, a potential future avenue of research is to visualise the modelling results in real time, so that the end users can use these system and forecasting results to monitor climate change or the city operations. This will require some visualization software to be written to fit the SMC methods to the data.

Bibliography

Amdahl, G. (1967). Validity of the single processor approach to achieving large-scale computing capabilities. In *AFIPS Conference Proceedings 1967*, page 483–485.

Banerjee, S., Carlin, B., and Gelfand, A. (2014). *Hierarchical Modeling and Analysis for Spatial Data*. CRC Press, 2nd edition.

Benth, F., Benth, J., and Koekebakker, S. (2007). Putting a price on temperature. *Scadinavian Journal of Statistics*, 34(4):746–767.

Benth, J. and Benth, F. (2012). A critical view on temperature modelling for application in weather derivatives markets. *Energy Economics*, 34(2):592–602.

Bolić, M., Djurić, P., and Hong, S. (2004). Resampling algorithms for particle filters: A computational complexity perspective. *EURASIP Journal on Advances in Signal Processing*, 15:2267–2277.

Bolić, M., Djurić, P., and Hong, S. (2005). Resampling algorithms and architectures for distributed particle filters. *IEEE Transactions on Signal Processing*, 53(7):2442–2450.

Brun, O., Teuliere, V., and Garcia, J. (2002). Parallel particle filtering. *Journal of Parallel and Distributed Computing*, 62(7):1186–1202.

Campbell, S. and Diebold, F. (2005). Weather forecasting for weather derivatives. *Journal of the American Statistical Association*, 100(469):6–16.

Carter, C. and Kohn, R. (1994). On Gibbs sampling for state space models. *Biometrika*, 81(3):541–553.

Carvalho, C., Johannes, M., Lopes, H., and Polson, N. (2010). Particle learning and smoothing. *Statistical Science*, 25(1):88–106.

Chapman, B., Jost, G., van der Pas, R., Gropp, W., and Lusk, E. (2007). *Using OpenMP: Portable Shared Memory Parallel Programming*. The MIT Press.

Chopin, N. (2002). A sequential particle filter method for static models. *Biometrika*, 89(3):539–551.

Chopin, N., Iacobucci, A., Marin, J., Mengersen, K., Robert, C., Ryder, R., and Schäfer, C. (2010). On particle learning. Available from arxiv: 1006.0554.

Chopin, N., Jacob, P., and Papaspiliopoulos, O. (2013). SMC^2: an efficient algorithm for sequential analysis of state space models. *J. R. Statist. Soc. B.*, 75(3):397–426.

Cressie, N. (1993). *Statistics for Spatial Data*. Wiley-Interscience.

Cressie, N. and Wikle, C. (2011). *Statistics for Spatio-Temporal Data*. Wiley-Interscience.

Del Moral, P. (1996). Non linear filtering: Interacting particle solution. *Markov Processes and Related Fields*, 2(4):555–580.

Del Moral, P., Jasra, A., and Zhou, Y. (2017). Biased online parameter inference for state-space models. *Methodology and Computing in Applied Probability*, 19(3):727–749.

Diggle, P. and Ribeiro, P. (2004). *Model-based Geostatistics*. Springer.

Doob, J. L. (1935). The limiting distributions of certain statistics. *The Annals of Mathematical Statistics*, 6(3):160–169.

Fearnhead, P. (2002). Markow chain Monte Carlo, sufficient statistics, and particle filters. *Journal of Computational and Graphical Statistics*, 11(4):848 –862.

Fearnhead, P. and Künsch, H. (2018). Particle filters and data assimilation. Available from `https://arxiv.org/abs/1709.04196`.

Fearnhead, P. and Taylor, B. (2013). An adaptive sequential Monte Carlo sampler. *Bayesian Analysis*, 8(2):411–438.

Frühwirth-Schnatter, S. (1994). Data augmentation and dynamic linear models. *J. Time Series Analysis*, 15(2):183–202.

Frühwirth-Schnatter, S. (1995). Bayesian model discrimination and Bayes factors for linear Gaussian state space models. *J. R. Statist. Soc. B.*, 57(1):237–246.

Galatioto, F., Bell, M., and Hill, G. (2014). Understanding the characteristics of the microenvironments in urban street canyons through analysis of pollution measured using a novel pervasive sensor array. *Environmental Monitoring and Assessment*, 186(11):7443–7460.

Gamerman, D. and Lopes, H. F. (2006). *Markov chain Monte Carlo: Stochastic Simulation for Bayesian Inference*. Chapman and Hall/CRC, 2 edition.

Gamerman, D. and Migon, H. (1993). Dynamic hierarchical models. *J. R. Statist. Soc. B.*, 55(3):629–642.

Gelfand, A., Diggle, P., Fuentes, M., and Guttorp, P. (2010). *Handbook of Spatial Statistics*. CRC Press.

Gelfand, A. and Smith, A. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85(410):398–409.

Gelman, A., Carlin, B., Stern, S., Dunson, B., Vehtari, A., and Rubin, B. (2013). *Bayesian Data Analysis*. Chapman and Hall/CRC.

Gelman, A. and Hill, J. (2007). *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press.

Gelman, A. and Rubin, D. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, 7(4):457–472.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741.

Geweke, J. (1992). Evaluating the accuracy of sampling-based approaches to calculating posterior moments. In Bernardo, J., Berger, J., Dawid, A., and Smith, A., editors, *Bayesian Statistics*, pages 169–193. Clarendon Press, Oxford.

Gilks, W., Richardson, S., and Spiegelhalter, D. (1996). Introducing Markov chain Monte Carlo. In Gilks, W., Richardson, S., and Spiegelhalter, D., editors, *Markov chain Monte Carlo in Practice*, pages 1–19. Chapman and Hall, London.

Gilks, W. R. and Berzuini, C. (2001). Following a moving target – Monte Carlo inference for dynamic Bayesian models. *J. R. Statist. Soc. B.*, 63(1):127–146.

Gong, P., Basciftci, Y., and Ozguner, F. (2012). A parallel resampling algorithm for particle filtering on shared-memory architectures. *2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops and PhD Forum*, pages 1477–1483.

Gordon, N., Salmond, D., and Simith, A. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEEE Proceedings*, F-140:107–113.

Gropp, W., Lusk, E., and Skjellum, A. (2014). *Using MPI: Portable Parallel Programming with the Message-Passing Interface*. The MIT Press.

Härdle, W. and Cabrera, B. (2012). The implied market price of weather risk. *Applied Mathematical Finance*, 19(1):59–95.

Harvey, A. (1989). *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge University Press.

Hastings, W. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109.

Heidelberger, P. and Welch, P. D. (1983). Simulation run length control in the presence of an initial transient. *Operations Research*, 31(6):1109–1144.

Hendeby, G., Karlsson, R., and Gustafsson, F. (2010). Particle filtering: The need for speed. *EURASIP Journal on Advances in Signal Processing*, pages 1–9.

Hu, X., Lindgren, F., Simpson, D., and Rue, H. (2013). Multivariate Gaussian random fields with oscillating covariance functions using systems of stochastic partial differential equations. Available from `https://arxiv.org/abs/1307.1384`.

Hu, X., Steinsland, I., Simpson, D., Martino, S., and Rue, H. (2015). Spatial modelling of temperature and humidity using systems of stochastic partial differential equations. Available from `https://arxiv.org/abs/1307.1402`.

James, P., Dawson, R., Harris, N., and Joncyzk, J. (2014). Urban Observatory Environment. Newcastle University. http://dx.doi.org/10.17634/154300-19.

Kalman, R. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45.

Kass, R. E. and Raftery, A. E. (1995). Bayes factors. *Journal of the American Statistical Association*, 90(430):773–795.

Kong, A., Liu, J. S., and Wong, W. H. (1994). Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association*, 89(425):278–288.

Liu, J. and West, M. (2001). Combined parameter and state estimation in simulation in simulation-based filtering. In Doucet, A., de Freitas, N., and Gordon, N., editors, *Sequential Monte Carlo Methods in Practice*, pages 197–223. Springer.

Liu, J. S. and Chen, R. (1998). Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93(443):1032–1044.

Lopes, H., Carvalho, C., Johannes, M., and Polson, N. (2011). Particle learning for sequential Bayesian computation. In Bernardo, J., Bayarri, M., Berger, J., Dawid, A., Heckerman, D., Smith, A., and West, M., editors, *Bayesian Statistics 9*, pages 317–360. Springer.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092.

Murray, L., Lee, A., and Jacob, P. (2016). Parallel resampling in the particle filter. *Journal of Computational and Graphical Statistics*, 25(3):789–805.

Nott, D., Dunsmuir, W., Kohn, R., and Woodcock, F. (2001). Statistical correction of a deterministic numerical weather prediction model. *Journal of the American Statistical Association*, 96(455):794–804.

Owen, A. B. (2013). *Monte Carlo theory, methods and examples*. Online.

Pacheco, P. (1996). *Parallel Programming with MPI*. Morgan Kaufmann.

Pasarica, C. and Gelman, A. (2010). Adaptively scaling the Metropolis algorithm using expected squared jumped distance. *Statistica Sinica*, 20(1):343–364.

Petris, G., Petrone, S., and Campagnoli, P. (2009). *Dynamic Linear Models with R*. Springer.

Pitt, M. and Shephard, N. (1999). Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599.

Pitt, M., Silva, R., Giordani, P., and Kohn, R. (2012). On some properties of Markov chain Monte Carlo simulation methods based on the particle filter. *Journal of Econometrics*, 171(2).

Plummer, M., Best, N., Cowles, K., and Vines, K. (2006). CODA: convergence diagnosis and output analysis for MCMC. *R News*, 6(1):7–11.

Quinn, M. (2003). *Parallel Programming in C with MPI and OpenMP*. McGraw-Hill.

Rabenseifner, R., Hager, G., and Jost, G. (2009). Hybrid MPI/OpenMP parallel programming on clusters of multi-core SMP nodes. *2009 17th Euromicro International Conference on Parallel, Distributed and Network-based Processing*, (c):427–436.

Raftery, A. and Lewis, S. (1996). Implementing MCMC. In Gilks, W., Richardson, S., and Spiegelhalter, D., editors, *Markov chain Monte Carlo in Practice*, pages 115–130. Chapman and Hall, London.

Rios, M. and Lopes, H. (2013). The extended Liu and West filter: Parameter learning in Markov switching stochastic volatility models. In Zeng, Y. and Wu, S., editors, *State-Space Models: Applications in Economics and Finance*, chapter 2, pages 23–61. Springer, New York.

Ripley, B. (2004). *Spatial Statistics*. Wiley-Interscience.

Roberts, G., Gelman, A., and Gilks, W. (1997). Weak convergence and optimal scaling of random walk Metropolis algorithms. *The Annals of Applied Probability*, 7(1):110–120.

Roberts, G. and Rosenthal, J. (2001). Optimal scaling for various Metropolis-Hastings algorithms. *Statistical Science*, 16(4):351–367.

Roberts, G. and Smith, A. (1994). Simple conditions for the convergence of the Gibbs sampler and Metropolis-Hastings algorithms. *Stochastic Processes and their Applications*, 49(2):207–216.

Rue, H., Martino, S., and Chopin, N. (2009). Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *J. R. Statist. Soc. B.*, 71(2):319–392.

Shaddick, G. and Wakefield, J. (2002). Modelling daily multivariate pollutant data at multiple sites. *J. R. Statist. Soc. C.*, 51(3):351–372.

Sherlock, C. and Roberts, G. (2009). Optimal scaling of the random walk Metropolis on elliptically symmetric unimodal targets. *Bernoulli*, 15(3):774–798.

Silverman, B. (1986). *Density Estimation for Statistics and Data Analysis*. Chapman & Hall/CRC.

Smith, A. and Gelfand, A. (1992). Bayesian statistics without tears: a sampling-resampling perspective. *American Statistician*, 46(2):84–88.

Stein, M. (1999). *Interpolation of Spatial Data: Some Theory for Kriging*. Springer.

Storvik, G. (2002). Particle filters in state space models with the presence of unknown static parameters. *IEEE: Trans. of Signal Processing*, 50(2):281–289.

Tierney, L. (1994). Markov chains for exploring posterior distributions (with discussion). *Annals of Statistics*, 22(4):1701–1762.

West, M. (1993a). Approximating posterior distribution by mixtures. *J. R. Statist. Soc. B.*, 55(2):409–422.

West, M. (1993b). Mixture models, Monte Carlo, Bayesian updating and dynamic models. *Computing Science and Statistics.*, 24:325–333.

West, M. and Harrison, J. (1999). *Bayesian Forecasting and Dynamic Models*. Springer, 2nd edition.

Zaritskii, V., Svetnik, V., and Šimelevič, L. (1976). Monte-Carlo technique in problems of optimal information processing. *Automation and Remote Control*, 36(12):2015–2022.