Fast Sequential Parameter Inference for Dynamic State Space Models

A thesis submitted to University of Dublin, Trinity College
in partial fulfilment of the requirements for the degree of
Doctor of Philosophy

Department of Statistics, School of Computer Science and Statistics,
University of Dublin, Trinity College

February 2012

Arnab Bhattacharya

This thesis has not been submitted as an exercise for a degree at any other University. Except where otherwise stated, the work described herein has been carried out by the author alone. This thesis may be borrowed or copied upon request with the permission of the Librarian, University of Dublin, Trinity College. The copyright belongs jointly to the University of Dublin and Arnab Bhattacharya.

Arnab Bhattacharya

Dated: February 9th, 2012

Abstract

Many problems in science require estimation and inference on systems that generate data over time. Such systems, quite common in statistical signal processing, time series analysis and econometrics, can be stated in a state-space form. Estimation is made on the state of the state-space model, using a sequence of noisy measurements made on the system. This difficult problem of estimating the parameters in real time, has generated a lot of interest in the statistical community, especially since the latter half of the last century.

One area that is particularly important is the estimation of parameters which do not evolve over time. The parameters in the dynamic state-space model generally have a non-Gaussian posterior distribution and holds a nonlinear relationship with the data. Sequential inference of these static parameters requires novel statistical techniques. Addressing the challenges of such a problem provides the focus for the research contributions presented in this thesis. A functional approximation update of the posterior distribution of the parameters is developed. The approximate posterior is explored at a sufficient number of points on a grid which is computed at good evaluation points. The grid is re-assessed at each time point for addition/reduction of grid points. Bayes Law and the structure of the state-space model are used to sequentially update the posterior density of the model parameters as new observations arrive. These approximations rely on already existing state estimation techniques such as the Kalman filter and its nonlinear extensions, as well as integrated nested Laplace approximation. However, the method is quite general and can be used for any existing state estimation algorithm.

This new methodology of sequential updating makes the calculation of posterior both fast and accurate, while it can be applied to a wide class of models existing in literature. The above method is applied to three different state-space models namely, linear model with Gaussian errors, nonlinear model and model with non-Gaussian errors, and comparison with some other existing methods has been discussed.

Acknowledgements

Firstly I would like to express my thanks and gratitude to my supervisor Professor Simon Wilson. After spending a few years away from academia, I was extremely slow to start with. Simon has always been extremely patient in dealing with me and my questions and has always been there when I needed direction. He has been an excellent supervisor, always encouraging me with the topics that interest me and also giving me space to make my own observations. His knowledge of statistics is astounding and his continuing enthusiasm for my work has never ceased to amaze me. He introduced me to the world of Bayesian statistics and I greatly appreciate the opportunities he provided for me in these years. I thank my stars that I have someone as supervisor, who would allow the bad routine of work I follow. I sincerely hope that our collaboration will continue into the future.

Next a big thanks to three people in our department, Brett Houlding, Louis Aslett and Ji Won Yoon. The beginning of our work has been possible due to the assistance I got from Brett and his vision and clarity on the subject as a whole. Louis has been an excellent friend and partner for discussing anything related to statistics that would come to my mind. Also a big thank you for introducing me to the world of efficient and intelligent programming. And finally, it is due to Ji Won that my knowledge in the specific area of my topic increased many fold. He always had an answer to any question I had in mind. Another big thank you to Susanne, Tiep, James, Jason, Thinh and everybody else in our office, room 108.

I would like to thank my parents and my brother who have always been encouraging and supportive of my studies. It is because of their support that I could leave what I was not enjoying and take up what interests me. A thank you must go to the great group of friends, both here and in Kolkata who had always been there no matter what. Thank you Deb, Aurab, Ani and Vee for been there, high or low.

Finally and most importantly, I would like to say a big and special thank you to dearest Anindita Lahiri. Your patience and support is what drove me along. You waited patiently in India while I have been away ten thousand kms in far-away Dublin.

Contents

Decleration **v**

Summary **v**

Acknowledgements **v**

1 Introduction **1**

1.1 Motivation 1

1.1.1 Motivation for Statistical Research 3

1.2 Research Contributions 4

1.3 Overview of Chapters 5

2 Time Series and Sequential Estimation **7**

2.1 Time Series and Forecasting 7

2.2 Time Series Models 9

2.2.1 Classical Time Series 9

2.2.2 Inference and Prediction for ARIMA 10

2.3 State space models 11

2.3.1 Dynamic Models 14

2.3.2 Dynamic Linear Models 15

2.3.3 Nonlinear/Non-Gaussian Models 16

2.4 Inference for State Space models 16

2.4.1 Kalman Filter 18

2.4.2 Extensions of the Kalman Filter 19

2.4.2.1 Extended Kalman Filter 20

2.4.2.2 Unscented Kalman filter 21

2.4.2.3 Other Filters 23
2.4.2.4 Divergence 24
2.4.3 Sequential Monte Carlo Methods 24
2.4.3.1 Sequential Importance Sampling Filter (SIS) 25
2.4.4 Grid Based Filters 30
2.5 Off-line methods 32
2.5.1 Expectation Maximization 32
2.5.2 Variational Bayes 32
2.5.3 Expectation Propagation 33
2.6 Parameter Estimation in State-Space Models 34
2.6.1 Kalman filter based methods: 35
2.6.2 Sequential Monte Carlo based methods: 36
2.7 Conclusion 38

3 Statistical Methodology 39

3.1 Statistical Inference 39
3.1.1 Interpretation of Probability 40
3.1.2 Prior Distribution 41
3.1.3 Likelihood Function 41
3.1.4 Posterior Distribution 42
3.1.5 Prior Elicitation 43
3.1.6 Predictive Distribution 45
3.2 Numerical Problems in Bayesian Inference 45
3.3 Markov Chain Monte Carlo 47
3.3.1 Markov Chains and Detailed Balance 47
3.3.2 The Metropolis-Hastings Algorithm 49
3.4 Integrated Nested Laplace Approximation (INLA) 50
3.4.1 Gaussian Markov Random Field (GMRF) 50
3.4.2 Gaussian Approximation 51
3.4.3 INLA 52
3.5 Kalman Filter 54
3.6 Conclusion 57

4 Sequential Parameter Estimation in State Space Models 58

4.1 Introduction 58

4.2 Initial Steps with INLA 60

4.3 Sequential Bayesian Estimation of θ 61

4.3.1 Crude Solution 63

4.3.2 Computation of $\mathbb{P}(\mathbf{X}_t \mid \mathbf{Y}_{1:t-1}, \theta)$ and $\mathbb{P}(\mathbf{X}_t \mid \mathbf{Y}_{1:t}, \theta)$ for Equation (4.3.4) 64

4.4 Algorithm 65

4.5 Updating the Grid 65

4.5.1 Motivation 65

4.5.2 New Internal Grid point 68

4.5.2.1 Identifying when and where to add a new internal point . . 68

4.5.2.2 Calculating the posterior and other necessary values at new grid point(s) 69

4.5.2.3 The Generalized Additive Model 72

4.5.2.4 Multivariate Adaptive Regression Splines 73

4.5.2.5 Problem with Smoothing methods 75

4.5.2.6 Interpolation 76

4.5.3 New External Grid point 79

4.5.3.1 Method I 83

4.5.3.2 Method II 85

4.5.4 Combination of adding points internally and externally 88

4.5.5 Computing other statistics at the new grid points 89

4.6 Final Algorithm for Updating the Posterior Density of $\theta \mid \mathbf{Y}_{1:t}$ 90

4.7 Conclusion 91

5 Correction to Improve Posterior Density of the Parameters 92

5.1 Problem encountered in the current method 92

5.1.1 Explanation of problem 92

5.1.2 Cause of the problem 94

5.2 Solution to this problem/Dealing with outliers 96

5.2.1 Explaining Cseke and Heske's method 96

5.2.2 Application of Cseke and Heske's method in our algorithm 98

5.3 Implementation 99

5.4 Conclusion 101

6 Application of Sequential Algorithm to Simulated Examples 102

6.1 Goal of the Application 102

6.2 Performance Measures 103

6.2.1 Mode and Probability intervals 103

6.2.1.1 Estimation of Mode 104

6.2.1.2 Estimation of 95% marginal probability limits 104

6.2.2 Mahalanobis Distance 107

6.2.3 Coverage Proportion 108

6.2.4 Computation Time 108

6.3 Application of different algorithms in different models 109

6.4 Linear Model with Additive Gaussian Errors 109

6.4.1 Method 1: Crude Method of Chapter 4 111

6.4.1.1 Computation of $\mathbb{P}(X_t \,|\, \mathbf{Y}_t, \theta)$ 111

6.4.1.2 Application of the Crude Method 111

6.4.2 Method 2: Sequential Update of $\mathbb{P}(\theta \,|\, \mathbf{Y}_{1:t})$ 113

6.4.2.1 Computation of $\mathbb{P}(X_t \,|\, \mathbf{Y}_{1:t-1}, \theta)$ and $\mathbb{P}(X_t \,|\, \mathbf{Y}_{1:t}, \theta)$ for Sequential Update 113

6.4.2.2 Results after implementing the algorithm 115

6.4.3 Particle filter 119

6.4.4 Comparison of the two methods 121

6.5 Nonlinear Model 126

6.5.1 Method Used: Sequential Update 126

6.5.1.1 Computation of $\mathbb{P}(X_t \,|\, \mathbf{Y}_{1:t-1}, \theta)$ and $\mathbb{P}(X_t \,|\, \mathbf{Y}_{1:t}, \theta)$ for Sequential Update 126

6.5.1.2 Performance of the Sequential Algorithm 130

6.5.2 Particle filter 132

6.5.3 Comparison of the two methods 132

6.6 Non-Gaussian Model 135

6.6.1 Method Used: Sequential Update . . . 136

6.6.1.1 Computation of $\mathbb{P}(X_t \mid \mathbf{Y}_{1:t-1}, \theta)$ and $\mathbb{P}(X_t \mid \mathbf{Y}_{1:t}, \theta)$ for Sequential Update . . . 137

6.6.1.2 Performance of the Sequential Algorithm . . . 138

6.6.2 Particle filter . . . 138

6.6.3 Comparison of the two methods . . . 141

6.7 Conclusion . . . 144

7 Conclusion and Further Work **147**

7.1 Conclusion . . . 147

7.2 Future Work . . . 149

7.2.1 Restriction on the number of static parameters . . . 149

7.2.2 Convergence of the filter . . . 150

7.2.3 Computation of log-posterior and necessary statistics at new grid points . . . 151

7.2.4 Application on real life problem . . . 151

7.3 Final Comments . . . 151

A Linear models **153**

A.1 Example 1 . . . 153

A.2 Example 2 . . . 153

A.3 Example 3 . . . 156

A.4 Example 4 . . . 156

B Nonlinear models **159**

B.1 Example 1 . . . 159

B.2 Example 2 . . . 159

C Non-Gaussian models **162**

C.1 Example 1 . . . 162

C.2 Example 2 . . . 164

C.3 Example 3 . . . 164

Bibliography **166**

List of Figures

2.1 DAG representing the general state space model where the parameters are not shown. 12

2.2 Fit of filtered level x_t over data recorded from Three Rock antenna, south of Dublin. The fit is done using a state space model with three unknown parameters. 13

4.1 This plot shows the posterior marginal density of a parameter from some toy example. The posterior density denoted by the points on the plot clearly shows that grid points on the tail have negligible posterior marginal values, which can be dropped to facilitate computation. Also a refinement of grid is required, since the supporting grid is shrinking and new points are required to be added internally. 67

4.2 The posterior marginal density in this plot clearly shows that new points need to be added to the left of the existing grid. Also some existing grid points from the right tail can be dropped since the value of the marginal density at those points are very small. 68

4.3 Single point added between two points for a single variable. 69

4.4 The $\times$'s and the $\odot$ in this plot shows possible combinations of new internal points for each of the two parameters. The points $(\theta_1^\times, \theta_2^1), (\theta_1^\times, \theta_2^2), (\theta_1^1, \theta_2^\times)$ and $(\theta_1^2, \theta_2^\times)$ require interpolation on a single variable. Bivariate rule is needed for the combination $(\theta_1^\times, \theta_2^\times)$. 69

4.5 This plot has blue circles denoting the posterior marginal density and black circles denote the grid points representing the support of the density. The red circles are new points which need to be added to the existing grid and are added using the rule explained in Section 4.5.2.1. 70

4.6 This plot shows the approximate posterior marginal density at the new points added internally to the toy example shown in Figure 4.5. In this toy example, the approximate posterior at these new points are calculated using linear interpolation. 72

4.7 This plot also shows the approximate posterior marginal density of the new internal points in the original grid as in Figure 4.5. The red circles denoting the approximate posterior at internal points are computed using GAM. Note the over fit in the density for the new points. 73

4.8 This plot shows the approximate posterior marginal density of the new internal points calculated using MARS. Note that the smoothing is better for this methodology than GAM, for the toy example that we have used in Figure 4.5. 75

4.9 This plot is exactly same as Figure 4.4. This is provided here once more to better explain the bivariate interpolation rule of calculating the log-posterior density. 77

4.10 Surface plot of bivariate normal log-density. The black points visible on the blue surface reflect the log-density values of the existing grid and the red points reflect the interpolated log-density values at the newly added internal points. 80

4.11 In this plot we have compared the performance of linear interpolation technique in computation of the log-density value of a new added internal point. The computed approximate log-density value at this point is compared with the true conditional log-density of $X_2 \,|\, X_1 = x_1$. Note that there is only a very small difference between the values. 80

4.12 Single point added outside the support of the grid for a single variable. . . . 81

4.13 This plot shows all possible combinations of new external points for each of the two parameters $\theta_1^\times$ and $\theta_2^\times$ respectively. The four points for which log-posterior density is calculated using linear interpolation are $(\theta_1^\times, \theta_2^1)$, $(\theta_1^\times, \theta_2^2)$, $(\theta_1^1, \theta_2^\times)$ and $(\theta_1^2, \theta_2^\times)$. Bivariate interpolation is used on the single point $(\theta_1^\times, \theta_2^\times)$. 81

4.14 This plot is same as Figure 4.2 and is shown here for the sake of convenience in understanding Section 4.5.3. 82

4.15 This plot shows what we are talking about in Section 4.5.3.1. The line segment passing through the two grid points at the edge intersects the x-axis. The intersection is chosen as the new edge of the grid and is assigned a very low value as approximate posterior marginal. Here the density at the mid point is also calculated via interpolation. 83

4.16 This plot represents a situation using a toy example where the true support of a parameter "moves" too much to the left and the existing grid needs to be updated to cover the true support. However it is not possible to use the method detailed in Section 4.5.3.1. In this case, the straight line joining the last two points at the edge will never intersect the x-axis. Hence the algorithm will fail. 84

4.17 This is another situation explained in Section 4.5.3.1 where the extrapolated point is too far away from the extreme point at the left of the support and is not really representative of the real situation. There is a lot of gap between the posterior density points at the left of the grid. A lot of points will be added in subsequent iterations making the grid size very big which in turn will slow the algorithm considerably. 85

4.18 This pot shows a toy example where the support is expanding and point are added on both sides of the existing support. Linear extrapolation is used to calculate the approximate posterior marginals at these new grid points. . 87

4.19 This figure is same as Figure 4.13. This plot is provided here for the sake of clarity. 87

4.20 Section of the grid which shows the combination of internal and external points for two parameters. 88

5.1 Grid of posterior marginal for parameter θ_3 at time T. It can be seen that nothing seems wrong with the shape of the posterior density until this time point. Figure 5.1 however points out to a problem. 93

5.2 Plot showing marginal posterior of the same parameter θ_3 at time $T + 1$. Note the concentration of marginal posterior on a single grid point. This collapse of the posterior density in a single update has been examined in this chapter. 94

5.3 Change in support of the marginal density of some parameter of a toy example in a single change in time point from T to $T + 1$ for some T. The scale of the x-axes in the two graphs are different because certain grid points were dropped from the first plot and hence the existing support has shrunk. This is a situation where the grid updating algorithm will work perfectly and the algorithm will not be stopped. 95

5.4 Plot of the exact likelihood computed at time $T+1$ for a given mode of the filtering density and a given grid value of the posterior of θ in the 4 parameter model discussed in this chapter. The value of the new data point, y_{T+1} has also been plotted on the same plotting area, to show its position with respect to the exact likelihood. One can see that it is located far into the tails, making the value of $\mathbb{P}(y_t \mid x_t, \theta)$ very close to 0. 96

5.5 The posterior distribution of $\mathbb{P}(\theta \mid \mathbf{Y}_{1:T})$. This plot is an exact replica of Figure 5.1. 100

5.6 The collapse of the posterior density of $\mathbb{P}(\theta \mid Y_{1:T})$ into a single grid point. Again this figure is same as that in Figure 5.2. 100

5.7 This plot shows the posterior of $\mathbb{P}(\theta \mid Y_{1:T+1})$ after the correction factor has been applied in the sequential update at every time point. The correction factor results in a non-zero value of the sequential multiplier term, which in turn stops the collapse to happen. 101

6.1 Fit of Nadaraya-Watson estimator over the posterior marginal values defined on the grid. One can generate a lot more number of values over the domain of the parameter along with their predictions, normalize them and then use them to get the 95% probability bounds. 105

6.2 Fit of Nadaraya-Watson estimator over the grid values plotted with respect to the CDF values.The CDF values being on the x-axis, inverse probability distribution is applied to get the 2.5% and 97.5% percentiles. 106

6.3 This is a pictorial illustration of the radio antenna and the transmitters under consideration. 110

6.4 Plot of the mode and 95% probability interval for all the three parameters of the linear model, over time. The actual value of the parameters do not fall within bounds for any of the cases, symbolizing poor performance for our algorithm when used with the crude method. 112

6.5 The posterior marginal distribution for each of the parameters are plotted along with the true values given by the red line. Note that INLA is successful in identifying the true value of the parameter for some specific data set from the linear model. 114

6.6 Posterior marginal density defined over the grid for the observation precision parameter ρ_y as computed by INLA. The true value lies outside the probability limits, signifying a poor performance for INLA. 114

6.7 Posterior marginal density adding more grid points to the right tail and dropping points at the left tail. Thus from the starting grid provided by the INLA, the grid has moved considerably to accommodate for the true value. 115

6.8 Plot of the mode and 95% probability interval for all the three parameters over time for some particular data set simulated from the linear model. The performance of the algorithm is quite good for this particular case. 116

6.9 The actual value of the state precision parameter, ρ_x of the linear model lies outside the probability bounds after 5000 updates to the posterior distribution. 117

6.10 In this example the starting grid does not cover the actual value of the observation precision parameter, ρ_y of the linear model. However, the posterior density adds points externally over time to shift and finally the actual value falls within the probability bounds. 117

6.11 Plot of the averaged mode and 95% probability interval for all the three parameters over time. The average is done over all the data sets simulated from the linear Gaussian model. Hence on an average the performance of our approach has been excellent for this particular model. 118

6.12 Plot of the mean and 95% probability interval of all the three parameters over time using particle filter for the linear model. Note that the intervals and the mean merge with each other at around time point 400. 120

6.13 The gradual decrease of unique particles of the parameters over time. This happens since there is no sampling-resampling procedure associated with this filter. 121

6.14 Plot of the mean of the three parameters calculated over time using particle filter, with each trace denoting a certain data set simulated from the linear model. 122

6.15 Plot of the average of the mean of the three parameters over time, along with the average of the 2.5% and 97.5% quantile of the particles, calculated from samples generated by particle filter on the linear Gaussian model. . . . 123

6.16 Box-plot of Mahalanobis distance computed at regular intervals of time for the two methods implemented for the linear model. The starting box plot on the extreme left of the figure is the Mahalanobis distance computed on the grid provided by INLA, after 25 time points. 124

6.17 Coverage proportion for each of the parameter values plotted over time for the linear Gaussian model. We start with plotting the coverage proportion for INLA based on the starting 20 observations. 125

6.18 Error bar of log-computation time (secs) of each algorithm plotted against Mahalanobis distance computed for time 500 and 1000 for the linear Gaussian model. The limit of the bars are the 25^{th} and 75^{th} percentile respectively. 125

6.19 The posterior marginal distribution for each of the parameters are plotted along with the true values of the parameters for the nonlinear model, given by the red line. INLA has successfully identified the true parameter values in this specific data set simulated from the model. 127

6.20 This is an example, where the starting grid provided by INLA is not able to identify the true value of the parameter ϕ, for a specific data set simulated from the nonlinear model. 128

6.21 Posterior distribution of one of the parameters at different time points of the algorithm. Note the change in support of the posterior and also the incredible change in shape of the density. 129

6.22 Plot over time, of the mode and 95% probability interval for all four parameters from the nonlinear model. The performance of SINLA has been good overall, with failure to identify the state precision parameter completely. . . 130

6.23 An example of the sequential algorithm identifying the true value of the state precision parameter, ρ_x, from the nonlinear model. 131

6.24 Trace plot of the averaged mode and 95% probability interval for all the four parameters from the nonlinear model. The average is done over all the data sets simulated from this particular model. 131

6.25 Plot showing mean of the particles generated using SMC, at each time point for all the data sets simulated from the nonlinear model. Some of the runs have produced divergence, where the trace plot has moved away and continue to do so after several iterations. 133

6.26 Plot showing grand mean of the particles for all the parameters of the nonlinear model along with 95% probability bounds. The particle filter fails to identify the actual value of nearly all the parameters. 134

6.27 Box plot of log of Mahalanobis distance for the two methods applied on the data sets simulated from the nonlinear model. There are many extreme values using SMC, for which the distance is calculated on a log scale. 134

6.28 Coverage proportion for each of the parameter values of the nonlinear model plotted over time. We start with plotting the coverage proportion for INLA based on the first 50 observations. The performance of both the methods is unsatisfactory as shown in this plot. 135

6.29 Error bar of log-computation time for each algorithm applied on the data sets simulated from the nonlinear model, plotted against Mahalanobis distance computed for time 500 and 1000. The limit of the bars are the 25^{th} and 75^{th} percentile respectively. 136

6.30 The posterior marginal distribution for each of the parameters of the non-Gaussian model are plotted along with the true values given by the red line. INLA performs very poorly in identifying the transformed parameters for this specific simulated data set. 137

6.31 Good identification of the actual value of the transformed parameter, κ as seen from the plot of the posterior marginal density, computed using INLA. 138

6.32 Plot of mode and 95% probability interval for both of the transformed parameters of the non-Gaussian model over time for a certain simulated data set. 139

6.33 Plot of the averaged mode and 95% probability interval for both of the transformed parameters from the non-Gaussian model, over time. The average is done over estimates of the statistics calculated over all the simulated data sets. 140

6.34 Plot of the mean of the two transformed parameters of the non-Gaussian model, over time, with each trace denoting a certain simulated data set. Particle filter is quite successful in identifying the true values of the parameters. 142

6.35 Trace plot of the average of the mean of the two transformed parameters of the non-Gaussian model. The average has been done over all the data sets simulated from this model. 143

6.36 Box plot of log of Mahalanobis distance for the two methods applied on the data sets simulated from the non-Gaussian model. The values are plotted on a log-scale because of the presence of outliers for the sequential INLA method. It has been observed that our algorithm diverged for a specific simulated data set, resulting in extreme values of the mode. 144

6.37 Coverage proportion for each of the parameter values plotted over time for the non-Gaussian model. We start with plotting the coverage proportion for INLA based on the first 50 observations. 145

6.38 Error bar of log-computation time of each algorithm applied on the data sets simulated from the non-Gaussian model, plotted against Mahalanobis distance computed for time 500 and 1000. The limit of the bars are the 25^{th} and 75^{th} percentile respectively. 145

A.1 Plot of the mode and 95% probability interval for all the three parameters over time for the linear model in Example A.1. The actual value of the parameters perfectly fall within the bounds indicating successful identification of the parameter by our approach. 154

A.2 Plot of the mode and 95% probability interval for all the three parameters over time for the linear model in Example A.2. The actual value of the parameters perfectly fall within the bounds indicating successful identification of the parameter by our approach. 155

A.3 Plot of the mode and 95% probability interval for all the three parameters over time for the linear model in Example A.3. The actual value of the parameters perfectly fall within the bounds indicating successful identification of the parameter by our approach. 157

A.4 Plot of the mode and 95% probability interval for all the three parameters over time for the linear model in Example A.4. The actual value of the parameters perfectly fall within the bounds indicating successful identification of the parameter by our approach. 158

B.1 Plot over time, of the mode and 95% probability interval for all four parameters from the nonlinear model. The performance of SINLA has been good overall, with failure to identify the theta parameter completely. 160

B.2 Plot over time, of the mode and 95% probability interval for all four parameters from the nonlinear model. The performance of SINLA has been quite poor, with complete failure to identify most of the parameters. 161

C.1 Plot of mode and 95% probability interval for both the transformed parameters of the non-Gaussian model over time for the non-Gaussian model presented in Example Appendix Non-Gaussian models. The performance of our approach is very good, considering the fact that it has successfully identified the true values of the unknown parameters for this model. 163

C.2 Plot of mode and 95% probability interval for both the transformed parameters of the non-Gaussian model over time for our non-Gaussian model presented in this section. SINLA has successfully identified the true values of the unknown parameters for this model as is quite evident from the plots. 164

C.3 Plot of mode and 95% probability interval for both the transformed parameters of the non-Gaussian model over time for our non-Gaussian model. SINLA has successfully inferred about the true value of the log-variance parameter but failed to identify the κ parameter which is a transformation of the AR parameter. 165

List of Tables

4.1 Array in which the parameter values and approximate log-posterior values are stored 71

4.2 Updated matrix where the internal grid is being updated. 71

4.3 Storage matrix at time t before internal points are added. 78

4.4 Storage matrix at time t after internal points are added and their log-posterior calculated using interpolation. 79

4.5 Storage matrix at time t before external points are added. 86

4.6 Storage matrix at time t after external points are added and the log-posterior values at the new points are computed using extrapolation. 86

Chapter 1

Introduction

Sequential Bayesian estimation of parameters which do not change over time forms the basis of this thesis. In time series, machine learning and signal processing, there exists a large array of problems which require estimation and forecasting in real time, using noisy observations. In such areas, data is modeled using a dynamic state space model in which estimation of the unknown parameters are necessary. In most applications, however, the parameters of the model are independent of time, and remain fixed (also known as *static*). While some work has been done on estimation of time-dependent parameters, the area of estimation of static parameters in the model requires special attention. Hence, more specifically, this thesis introduces a new approach for static parameter estimation in general dynamic state-space models. This research idea was motivated by a problem involving estimation of signal-power of cognitive radio devices in wireless telecommunications.

1.1 Motivation

The main motivation for this research is to make accurate real-time estimation of unknown parameters in a dynamic state-space model. It started as trying to address a problem related to analyzing signal strength in wireless networks which is an ongoing research under Prof. Linda Doyle in the Center for Telecommunications Value-Chain Research (CTVR) at Trinity College, Dublin. The principal factor in this research is for cognitive radio devices to efficiently analyze the signal in real time and detect the lack of transmission power at a given frequency.

The electromagnetic radio spectrum is a natural resource, the use of which using transmitters and receivers is licensed by governments. In general, accessing the spectrum band is a greater problem than physical availability of the spectrum (Haykin 2012). Moreover, direct control by management of the available spectrum usage causes artificial spectrum scarcity. Scanning the radio spectrum licensed to commercial use, even in busy areas, like cities, shows the following pattern:

a) some frequency bands are largely unoccupied most of the time;

b) some bands are only partially occupied;

c) the remaining bands are very heavily used, causing problems with accessing it.

Thus if communicators use these interleaved and fragmented bands temporally, the spectrum-usage will be more effective and the cost of the spectrum now used will be much lower. A device (inclusive of software) meant for efficient use of the spectrum by exploiting the existence of unoccupied frequency bands (or spectrum holes) would thus assist this new idea. The cognitive radio (Haykin 2005) is such a device and is defined as an intelligent wireless communication system that is aware of its environment and uses the methodology of 'understanding-by-building' to learn from the environment and adapt to variations in the input stimuli. The main objectives of such a device are the following:

- highly reliable communication whenever and wherever needed;
- efficient utilization of the radio spectrum.

An important objective of cognitive radio is to develop a spectrum accessing system whereby, once the cognitive radio detects a spectrum hole, it will be permitted to transmit at that frequency and make use of the lack of primary user activity.

In order for cognitive radio devices to conform to any specified frequency usage etiquette, and to ensure they do not interfere at frequencies that are indeed being transmitted upon by other primary users, knowledge of the prevailing radio environment must be constantly available. As such, and because any radio environment of interest is likely to be subject to rapid fluctuations due to changes in the power being transmitted along any given frequency, the recordings made by the radio device must be efficiently analysed in real or virtually-real time, and must result in a highly accurate approximation of current frequency usage. This task is more difficult due to the requirement that it be accomplished in light of residual uncertainty concerning background noise and interference from additional frequency users.

In engineering literature, different models exist which make use of the physical characteristics of the environment to analyze and predict the power spectrum. Full-wave channel models or directional channel models are different methods proposed for this purpose (Lotze et al. 2006). Such models make use of different physical properties of radio frequency, e.g. angle of arrival, angle of departure, distance travelled etc. to make inferences about the spectrum usage. The concept of dynamic spectrum access control through determining some form of measure for interference in radio spectrum has also been considered (FCC 2003). However, as mentioned earlier, these models make use of physical theory and ignores the residual uncertainty concerning background noise, thus requiring greater detail on the environment to be successful.

Dynamic state-space models are a rich class well suited in capturing the uncertainty associated with the spectrum along with incorporating the physical properties, which in turn would help in better prediction for signal strength and identifying spectrum holes. We discuss this idea in more detail in the next section.

1.1.1 Motivation for the Statistical Research

Generally, two main styles of statistical modelling can be distinguished for the previously discussed wireless network usage problem. The recorded signal of a single user having a receiver of some kind forms a time dependent data set. On one hand, one can fit a model for the whole data set and estimate the unknown parameters. Classical time series developed by Box et al. (1970) and inference using general state-space models are examples of such inference method. A different methodology for dealing with the dynamic spectrum problem discussed above is processing received data sequentially rather than in a batch and this idea is best utilized through *dynamic state-space models* approach. Such models are very well suited for providing a solution to the spectrum problem, since prediction will be provided in real time. Two distinct elements are usually unknown in such models. One is the unobserved *state process* and the other are the *parameters* associated with both the noise and the state process. Sequential estimation requires speedy inference while maintaining accuracy in estimating the unknowns. A major driving force behind this, is the increase in computational ability which has made the Bayesian approach more popular. In this thesis, we are dealing with sequential Bayesian estimation of these unknowns.

While a lot of research has been done on state estimation, the area of parameter estimation has got much less attention in comparison. Sequential parameter estimation is not straight forward, because not only do they have a non-Gaussian distribution (in general), they also have a strong nonlinear relationship with the observations. This makes Bayesian inference of the model unknowns in real time very complex since one has to achieve both speed up and accuracy.

There are three distinct solutions for sequential parameter estimation. Ever since the Kalman filter was introduced for exact and closed form solutions for linear Gaussian models, it has been extended to provide sub-optimal solutions to parameter estimation. Almost always, in such cases, the distribution of the parameters are assumed to be Gaussian. Sequential Monte Carlo (SMC) is another very strong and useful numerical method which estimates the unknowns in a general state-space models by generating a large number of samples. Monte Carlo based methods produce quite accurate estimates but compromise on speed. A third method approaches the sequential inference method by construction of a discrete grid to approximate the posterior of the parameters. These methods will be discussed in greater detail in Chapter 2 and will come to know of their short comings much better.

The key observations noted in the methods discussed above that lead us to this work are the following:

1. There is a need for real time *static* parameter estimation. Very little work on this problem has been attempted and this problem requires developing a good methodology which maintains a balance between computation speed and accuracy;
2. Bayesian estimation of parameter by identity would make an easily implementable and efficient algorithm for estimation of static parameters on a grid. Moreover, this idea would also increase the computation speed of the algorithm when compared to sampling based techniques, and
3. In many problems we find that the parameter space is often of low dimension and this allows grid based estimation, which again is very fast and accurate.

These three points form the basis of this thesis. The problems of the category of real time sequential estimation such as the problem of dynamic spectrum discussed in the previous section is an area which demands a solution which is both accurate enough and speedy, and hence fits into this problem.

1.2 Research Contributions

The main research contribution of this thesis is listed below:

- A new real-time parameter estimation method for general state space models has been discussed in this thesis. This new method is inspired from the work of Rue et al. (2009) on the Integrated Nested Laplace Approximation (INLA) for Gaussian Markov random fields (GMRF) and is deterministic, rather than sampling based in nature. The new method is grid based, which can be viewed as an extension to INLA and has been named Sequential INLA (SINLA), even though it is not quite the sequential version of INLA. This thesis provides the algorithm for this method, which is easy to use and also gives examples of its possible usage along with the limitations. There are two advantages to this method which are listed below.
- Firstly, the new method can be applied to a diverse range of state space models. The new method can be implemented on all those models for which INLA can be used and many more as will be discussed in the latter chapters.
- The second and most important strength of this methodology is that it can be implemented with all of the existing filters that estimates the state process. There is no underlying assumption of the posterior density being Gaussian. Also this method does not require an artificial evolution transformation to be introduced for the static

parameters. Hence this method has the strength to out-perform most of the existing parameter estimation algorithms, in line with the possible restrictions that we have already mentioned. As far as we are concerned, work on this has not been done earlier.

1.3 Overview of Chapters

The rest of the thesis is organized as follows:

Chapter 2: Time Series and Sequential Estimation

In this chapter we discuss the preliminaries associated with classical time series models and then moves on to general state space models. We also explain the difference between off-line and on-line estimation methods. Classical time series modelling methods like those using Box-Jenkins models namely ARMA, ARIMA etc are given. References to off-line estimation of parameters are also included in this chapter. Parameter and state estimation methods involving state space models, viz. , DLM, Kalman filter and its extensions, SMC and other well known methods are also discussed in greater detail.

Chapter 3: Statistical Methodology

Statistical methods that are used for analysis are detailed in this chapter. This includes a short discussion on classical and Bayesian inference. Since almost all of the modelling is done in the Bayesian framework it receives more attention. The motivational idea for our method is INLA. The assumptions associated with INLA and the construction of the grid for the parameter space has been discussed broadly. Our method has been compared to existing methodologies like MCMC and SMC. Hence the Markov chain theory associated with both these methods and the different algorithms are briefed in this chapter. We also provide a proof of Kalman filter, based on Bayes theorem, at the end to give an idea of deterministic filtering methodology that has been used extensively for both linear and nonlinear models. This is particularly important since all our examples use the Kalman filter or sub-optimal extensions of it.

Chapter 4: Sequential Parameter Estimation in Time series models

This chapter contains the main contribution of this thesis. A deterministic sequential estimation for static parameters in a general dynamic state-space model is provided in a Bayesian framework. The chapter starts with providing a crude solution to this problem, developed at the beginning of our research. After this the basic concept underlying our sequential inference method is introduced and issues associated with its implementation are discussed in detail. A working algorithm for this method is also provided. Since our method is grid based, the important problem regarding how to make our grid adopt to changes in data by adding and dropping grid points are discussed in great detail. This chapter further discusses the limitations of our algorithm and puts down situations under which SINLA gives poor inferences.

Chapter 5: Correction to Improve Posterior Density of the Parameters

The sequential parameter estimation methodology, introduced in Chapter 4 can have a problem similar to the degeneracy problem (Doucet, Godsill & Andrieu 2000) in SMC. In this chapter, we provide a correction factor that can be put in the algorithm for complex models with outliers, which is when the problem arises mostly. This correction factor helps in better estimation of the parameters. However this may make the model slow, a reason why it is not used for all the models. We show with an example how the inference is improved with this correction factor.

Chapter 6: Application of Sequential Algorithm to Simulated Examples

The developed algorithm, SINLA is applied to different models in this chapter. The performance of the algorithm is being tested with respect to different models with different complexity levels. Static parameters of both linear and nonlinear models and also models with non-Gaussian error are being estimated and their accuracy checked using existing statistical measures. The performance of SINLA has also been compared to SMC and INLA itself, to get a view of the performance accuracy with respect to computation speed.

Chapter 7: Conclusion

This is the last chapter which concludes this thesis and provides a short discussion on possible future work related to this method.

In the next chapter, named *Time Series and Sequential Estimation*, we discuss the existing statistical techniques to model data of type time series. Different modelling techniques related to time series data and also off-line and on-line methods of state and parameter estimation are detailed in this chapter.

Chapter 2

Time Series and Sequential Estimation

In this chapter, modelling of time series data is discussed in the general setting of classical time series methodology and general state-space models. A short overview of time series methods is provided, followed by generalized state space methods. The state space method is presented both from an engineering and a statistical point of view. Subsequently, both the off-line and on-line estimation techniques used for state-space models are explained. More importance has been given to the one-line estimation technique, better known as sequential estimation and basic examples are used to better understand this methodology. Special emphasis has been given to the Bayesian framework of estimating the unknowns, since our developed method is based on it. Refer to Box et al. (1970) for an insight into the classical time series modelling and Durbin & Koopman (2001) for general state-space models. Sequential inference of state-space models in a linear set up is explained in great detail in West & Harrison (1997). Grewal & Andrews (2011) gives an adequate account of the problems and modelling techniques from an engineering standpoint.

2.1 Time Series and Forecasting

Much of statistical methodology deals with models in which the observations are assumed to be independent. In planned experiments, techniques like *randomization* are introduced to ensure that independent observations are collected, since dependence is considered a nuisance for construction of the model and makes inference a very complex exercise. However, several fields of research like business, economics, engineering and natural sciences produce data that are dependent and are ordered. Such observations are called *time series* data and main interest lies in studying the nature of the dependence. Primary interest often lies in prediction or *forecasting* future values from current and past values.

It is assumed that observations are available at discrete, equally spaced intervals of time. Let $\mathbf{Y}_t$ be the random variable at time t, denoting some time-dependent physical process, and $\mathbf{Y}_{t-1}, \mathbf{Y}_{t-2}, \dots$ be that from previous time points. The basis of investigation for different scientists or economists then becomes prediction of the values of the physical process under consideration at some future time point $t + l$, where $l > 0$ and to find an estimate $\hat{\mathbf{Y}}_{t+l}$.

The classical time series technique uses the Box and Jenkins' methodology to model and forecast time series data. A subsequent development was *state space* modelling which is considered to be more generalized for treatment of a wider range of problems that include both non-stationarity and non-Gaussian property. The rigidity of the classical models can be avoided by the structure of the state space models. This is mainly because state space models allow a natural interpretation of a time series as the result of several components, namely, trend, seasonality and regression covariates. Moreover, the classical ARMA models can be shown to be a special case of the state space approach (Durbin & Koopman 2001). The Bayesian framework has been used in both types of inference and prediction ((Monahan 1983) and (West & Harrison 1997)). This will be discussed in more detail in the future sections of this chapter.

Time series inference is generally of two different types. A dynamic model methodology does real-time inference as observations are recorded sequentially in time. The problems of estimation and forecasting are solved by recursively computing the conditional distribution of the quantities of interest, given the available information (Arulampalam et al. 2002). A very important feature of this method is that past data need not be stored as only new data are used to update the conditional posterior. A static model methodology on the other hand is not a recursive procedure. At any given point in time, it uses all the data to estimate the parameters (Brockwell & Davis 2002) and data needs to be stored for this purpose. Hence it makes use of all the data to model and forecast in a static framework.

The aim of this thesis is to investigate fast sequential Bayesian inference of time series data based on a dynamic model. Dynamic models, both linear and nonlinear have been studied extensively by engineers for a long time. The seminal paper by Kalman (1960) has been a reference for inference on dynamic models to this day. Several advancements are being made to take into account nonlinearity and non-Gaussian property and are extensions to the Kalman filter (Haykin 2001). A major limitation, as perceived by statisticians in these methods, is the assumption that errors are Gaussian. This assumption is not required by Markov Chain Monte Carlo based methods. Doucet et al. (2001) is a good reference for this important subject with a good overview of theory as well as examples. Different modelling schemes developed over time will be discussed in the next few sections followed by different inference techniques.

2.2 Time Series Models

In this section we discuss classical modelling of time series observations as proposed by Box et al. (1970). This discussion includes model fitting and estimation of model parameters. This is followed by discussing state space modelling and we explain why this modelling technique is more generalized. Finally we discuss the application of state space models in dynamic inference of real time data.

2.2.1 Classical Time Series

The most basic model used in time series is the linear additive model, also known as the classical decomposition model:

$$y_t = T_t + S_t + \epsilon_t \quad t = 1, 2, \ldots, n.$$

Here, T_t is a slowly changing component called the *trend*, S_t a periodical component with a fixed period referred to as the *seasonal component* and ϵ_t is the *random error component*, assumed to be Gaussian. The error component needs to be stationary for many inference methods under classical time series to work successfully. This assumption, if not true, requires transformation on the data to ensure stationarity. In many applications, these components are multiplicative (Cowpertwait & Metcalfe 2009) and the model is converted to the additive model by a logarithmic transformation. The general framework for studying stationary processes is based on linear time series models, namely Box-Jenkins autoregressive moving-average (ARMA) models (Box et al. 1970). A process $\{Y_t\}$ is said to be an $\mathbf{ARMA(p, q)}$ process with mean μ if $\{Y_t - \mu\}$ is an $ARMA(p, q)$ process denoted as

$$\phi(B)Y_t = \theta(B)Z_t,$$

where $\phi(\cdot)$ and $\theta(\cdot)$ are p^{th} and q^{th} degree polynomials

$$\phi(x) = 1 - \phi_1 x - \cdots - \phi_p x^p,$$

and

$$\theta(x) = 1 - \theta_1 x - \cdots - \theta_q x^q.$$

B is the backward shift operator ($B^j X_t = X_{t-j}, j = 0, \pm 1, \cdots$). The time series $\{Y_t\}$ is said to be an ARMA model of order (p, q), where p is the number of AR parameters and q is the number of MA parameters. Z_t is a random component which follows a Gaussian distribution with mean 0 and variance σ^2. It is important to keep in mind that an essential part of this definition is that Y_t needs to be stationary, for which the two important properties of **causality** and **invertibility** (Brockwell & Davis 2002) are strictly maintained. These properties ensure the existence of a stationary solution from the equation and also make sure that the solution is unique. Determining the appropriate model involves several steps. Determining the values of (p, q), also known as order selection is done using graphical procedures like the Auto-correlation function (ACF) plot or the Partial Auto-correlation (PACF) plot. A variety of goodness of fit tests or minimization of AIC statistic (Brockwell & Davis 2002) are other methods of determining the order of time series model.

A generalization of the ARMA class of models is done by modelling nonstationary time series. Autoregressive integrated moving-average (ARIMA) processes provide this generalization. This is done by introducing an extra non negative parameter into the model, known as the difference parameter (d). A time series, $\{Y_t\}$ is an $\mathbf{ARIMA(p, d, q)}$ process if $X_t := (1 - B)^d Y_t$ is a causal ARMA(p,q) process. The definition just states that the process reduces to an ARMA series when differenced finitely many times. The difference operator (d) also needs to be determined while the model selection is done. A further extension to ARIMA models is the seasonal ARIMA (SARIMA) model. Once differencing is applied to the data in these models, the analysis is same as that of ARMA models. The only unknown component is the differencing parameter d.

2.2.2 Inference and Prediction for ARIMA

Maximum likelihood estimation has been the most popular estimation technique for the parameters $\phi = (\phi_1, \cdots, \phi_p)$, $\theta = (\theta_1, \cdots, \theta_q)$ and σ^2. Some preliminary estimation techniques like Yule-Walker estimation or Burg's algorithm are used to get good initial values (Brockwell & Davis 2002). From an application point of view, the most important question one asks is regarding prediction; prediction of series h-steps ahead in time where we have data until time t. Thus one would be interested to make predictions about Y_{t+h} given that we have some data $y_1, \cdots, y_t$, for some $h > 0$. *Durbin-Levinson algorithm* and the *Innovations algorithm* (this is applicable to all series with finite second moments, regardless of stationarity) (Brockwell & Davis 2002) are usually used for predicting h-step ahead. Once reduced to ARMA models, all these estimation and prediction methods can also be used for ARIMA or SARIMA models. Note that prediction is usually carried out by using extensions of the methods developed for ARMA models.

A Bayesian approach to ARMA models was taken up after the limitations of classical approaches came into light and Bayesian approach gained popularity. Classical methods of inferring about parameters in ARMA models are based on conditional maximum likelihood estimation (Brockwell & Davis 2002). One of the biggest limitations of the classical approach is that the computation of unconditional likelihood function for the general stationary and invertible ARMA(p,q) model is quite complicated and produce numerical difficulties (Chib & Greenberg 1994). The Bayesian paradigm ensured that many of the computation difficulties effectively disappear and also present us with powerful tools to simulate from intractable joint distributions. There were of course other problems, for example initial applications debated over the choice of prior, a primary example of which is the controversy over the choice of prior for the coefficients of models (Steel 2008). Box and Jenkins devoted a section to Bayesian estimation and proposed the use of Jeffrey's prior (Jeffreys 1961). Subsequently, a 'fully' Bayesian analysis of ARMA models is provided by Monahan (1983). For a very good source of extensive study of Bayesian time series, the interested reader should look into Prado & West (2010).

2.3 State space models

State space representations have had a profound impact on time series analysis in the last decade or so. This is an extremely rich class of models for time series, including and going well beyond the linear models like ARMA and the variations thereof. The rigidity of the classic time series models are avoided by allowing the trend and seasonal components to evolve randomly rather than deterministically as was the case of the classical decomposition models (Brockwell & Davis 2002). In other words, state space models consider a time series as the output of a dynamic system perturbed by random disturbances. This structure allows a natural interpretation of the time series in terms of the components. At the same time, these models have an elegant and powerful probabilistic structure, allowing a natural treatment using the Bayesian approach. These models can be used to model both univariate and multivariate time series and include non-stationarity and irregularity in its formulation. State space models are very widely used in economics, biology, sociology, engineering and several other fields.

Real world time dependent processes produce observable outputs which can be characterized either in discrete time e.g. digital signals, characters from a finite alphabet, or continuous time e.g. speech samples, temperature measurements etc. In the state space approach it is assumed that there is a hidden system $\mathbf{X}_t \in \mathbb{R}^{n_x}$, $t \in \mathbb{N}$, also known as the latent variable. It is better known as the *state process* and it has an initial probability $\mathbb{P}_{\mathbf{X}_0}(\mathbf{x}_0)$ at time $t = 0$. It is further assumed that the state process evolves over time t as a first order Markov process (the Markov property is discussed in detail in the next chapter) according to the transition density $\mathbb{P}_{\mathbf{X}_t \,|\, \mathbf{X}_{t-1}, \Theta_t^1}(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}, \theta_t^1)$, where θ_t^1 represents a set of parameters. The observations are denoted by $\mathbf{Y}_t \in \mathbb{R}^{n_y}$ and are assumed to be conditionally independent given the states and some set of parameters Θ_t^2. They are generated from the conditional probability density $\mathbb{P}_{\mathbf{Y}_t \,|\, \mathbf{X}_t, \Theta_t^2}(\mathbf{y}_t \,|\, \mathbf{x}_t, \theta_t^2)$. This is also known as the *measurement process*. It is of course possible that some of the parameters do not depend on time. As already discussed in Chapter 1, such parameters are known as static parameters and denoted by θ. For simplicity, we denote all the unknown parameters as θ and drop the subscript of $\mathbb{P}_A(a)$ to denote it as $\mathbb{P}(a)$. Thus to sum up, the model is described by

$$\mathbb{P}(\mathbf{x}_0|\theta), \tag{2.3.1}$$

$$\mathbb{P}(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}, \theta) \quad t \geq 1 \text{ and} \tag{2.3.2}$$

$$\mathbb{P}(\mathbf{y}_t \,|\, \mathbf{x}_t, \theta) \quad t \geq 1. \tag{2.3.3}$$

The probability distributions can be continuous or discrete, and t is strictly discrete. This pattern is best illustrated by the following *directed acyclic graph* (DAG):

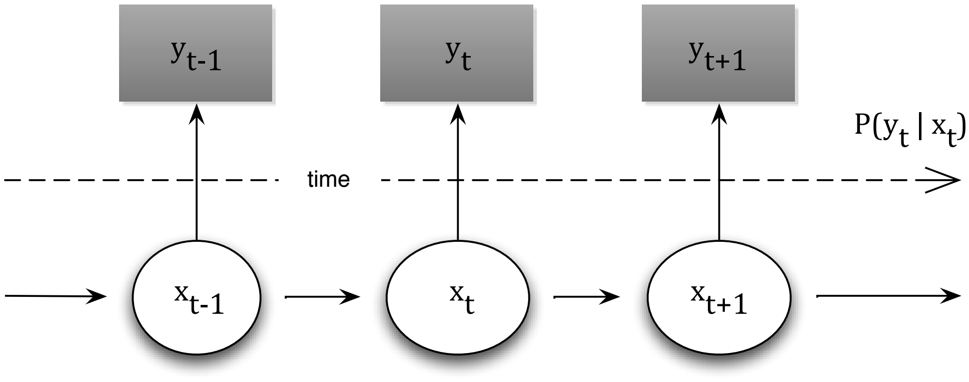


Figure 2.1: DAG representing the general state space model where the parameters are not shown.

An equivalent representation of the above model is the general state space model, in which the state variable is the same as the latent variable. The representation for a continuous time process is

$$\frac{\partial \mathbf{x}_t}{\partial t} = g_t(\mathbf{x}_{t-1}, \mathbf{w}_t, \theta_2), \tag{2.3.4}$$

$$\mathbf{y}_t = f_t(\mathbf{x}_t, \mathbf{u}_t, \mathbf{v}_t, \theta_1), \tag{2.3.5}$$

where in Equation (2.3.4) $g_t : \mathrm{R}^{n_x} \times \mathrm{R}^{n_w} \times \mathrm{R}^{n_{\theta_2}} \to \mathrm{R}^{n_x}$ is a possibly nonlinear function, $\mathbf{w}_t$ is the *system/state error* and n_x, n_w and n_{θ_2} are the dimensions of state, state error and parameter vector θ_2, respectively. In Equation (2.3.5), $f_t : \mathrm{R}^{n_x} \times \mathrm{R}^{n_u} \times \mathrm{R}^{n_v} \times \mathrm{R}^{n_{\theta_1}} \to \mathrm{R}^{n_y}$ is also a possibly nonlinear function, $\mathbf{u}_t$ is some covariate process that is fully known and n_y, n_u, n_v and n_{θ_1} are the dimensions of measurement process, $\mathbf{u}_t$, the measurement error and parameter vector θ_1, respectively. The likelihood is fully specified by f, $\mathbf{v}_t$ and θ_1, while the continuous transition density $\mathbb{P}(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}, \theta_2)$ is completely specified by g and $\mathbf{w}_t$. We denote by $\theta = \{\theta_1, \theta_2\}$.

For a discrete time series, the differential equation gets replaced by a difference equation and is of the following form:

$$\mathbf{y}_t = f(\mathbf{x}_t, \mathbf{u}_t, \mathbf{v}_t, \theta_1), \tag{2.3.6}$$

$$\mathbf{x}_t = g(\mathbf{x}_{t-1}, \mathbf{w}_t, \theta_2). \tag{2.3.7}$$

The state-space model, together with the prior distributions of the hyperparameters and the system states, defines stochastically how the system evolves over time and how we inaccurately observe the hidden state process. We provide an example here from the cognitive radio example we discussed in Chapter 1. For this purpose, recordings were analysed from a signal transmitted by the Three Rock Antenna south of Dublin. The following state space model was fitted to the collected data:

$$y_t = x_t + v_t \quad v_t \sim \mathcal{N}(0, \sigma_u^2), \tag{2.3.8}$$

$$x_t = (1+\phi)x_{t-1} - \phi x_{t-2} + w_t \quad w_t \sim \mathcal{N}(0, \sigma_w^2). \tag{2.3.9}$$

Parameter ϕ and the variance parameters σ_u^2 and σ_w^2 are unknown and need to be estimated. This is an example of a discrete state space model, where we assumed that there is an underlying process, x_t which signifies the true signal whereas the data we record are noisy realisations of the true process. A plot of 100 recording of the data along with the fitted filtered level of the state process x_t of the above model is shown in Figure 2.2 It can be seen how the filtered values closely follow the data, and visually the model seems to

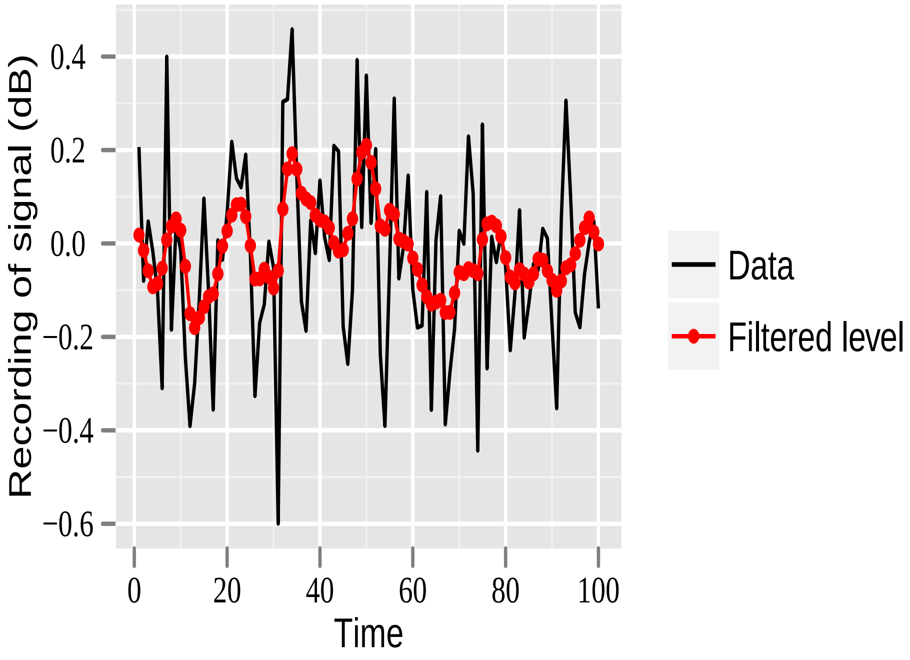


Figure 2.2: Fit of filtered level x_t over data recorded from Three Rock antenna, south of Dublin. The fit is done using a state space model with three unknown parameters.

have fitted well to the data. If we try to fit an ARMA or ARIMA model though, the model will be difficult to interpret in terms of the physical process in hand. It is easy to see the advantages one has when fitting a state space model, not only in terms of interpreting the model but also in terms of construction of the model. The rigidity of ARMA models is absent since one can go on adding layers to the model, with each layer depicting a physical process. This is discussed more broadly in the remaining part of this section.

State space techniques have several advantages over the Box-Jenkins time series models. The model building process is very structural and hence methodical. The different components of the classical model, such as trend or seasonal variations can be modelled separately and then put together to form a single state space model. Extra effects of explanatory variables are also easy to put in the same model. The following equations provide an example to explain what we mean:

$$\begin{aligned} y_t &= T_t + s_t + u_t + v_t, \\ T_t &= T_{t-1} + w_t, \\ s_t &= s_{t-1} + z_t, \end{aligned}$$

where one can think of T_t, s_t and u_t as trend, seasonal and fixed covariate input respectively. v_t, w_t and z_t are the error components. In addition to this structural formation, they are also very general, in the sense that they comprise all ARIMA models (Durbin & Koopman 2001). Multivariate observations can be handled by straightforward extensions of univariate theory. It is also possible to allow for missing observations. Furthermore, because of the Markovian structure of state space models, the calculations needed to implement them can readily be implemented in recursive form. The models in which the recursive Bayesian technique is applied are generally known as *dynamic models*. These models enable large data sets to be handled without a huge increase in computational burden, since these methods work sequentially and it is not necessary to store the data. Thus assuming we have some data set $Y_{1:T} : Y_1, \cdots, Y_T$; at time T, dynamic models would only require Y_T to make any inference about the parameters or state process. Box-Jenkins type models on the other hand will need to store the whole data set, $Y_{1:T}$. As T becomes large, computational issues set in and the algorithm fails. This idea will be discussed in more detail in the next section.

2.3.1 Dynamic Models

A dynamic model is a forecasting model which can be expressed in a recursive form. Dynamic or sequential inference is necessary when one needs to update the posterior in real time, from $\mathbb{P}(x_t \mid y_{1:t-1})$ to $\mathbb{P}(x_t \mid y_{1:t})$. It is well known that state space models are ideal for dynamic inference (West & Harrison 1997). In sequential inference, linearity and Gaussian errors (or the lack of both) plays a big role in estimating the ‘parameters’ of interest. While linear Gaussian models produce exact inference, which requires updating the first two moments at each time t, this is not the case for nonlinear or non-Gaussian models. Hence dynamic models are generally divided into linear models and nonlinear/non-Gaussian models, the estimation procedures of which are treated differently. We will be describing the two different types of models in the following sections.

2.3.2 Dynamic Linear Models

The most widely known and used subclass of dynamic models is known as the *Normal Dynamic Linear Models*, more popularly known as *Dynamic Linear Models* (DLM) (West & Harrison 1997). The fundamental principles used by a forecaster in dealing with problems through dynamic linear models comprise:

- sequential modelling;
- structural state space model;
- probabilistic representation of the parameters and forecasts.

These properties form the basis of any sequential inference algorithm.

The general DLM for a vector observation $\mathbf{Y}_t$ of the time series $\mathbf{Y}_1, \mathbf{Y}_2, \ldots$ is given by:

$$\mathbf{y}_t = \mathbf{F}_t^{'}\mathbf{x}_t + \mathbf{v}_t \quad \mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{V}_t), \tag{2.3.10}$$

$$\mathbf{x}_t = \mathbf{G}_t\mathbf{x}_{t-1} + \mathbf{w}_t \quad \mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{W}_t), \tag{2.3.11}$$

where for each time t:

1. $\mathbf{y}_t$ represents a $(n \times 1)$ measurement process;
2. $\mathbf{x}_t$ represents a $(p \times 1)$ state process;
3. $\mathbf{F}_t$ represents a $(p \times n)$ matrix, probably representing explanatory variables;
4. $\mathbf{G}_t$ represents a $(p \times p)$ matrix, also known as the evolutionary matrix;
5. $\mathbf{V}_t$ represents a $(n \times n)$ covariance matrix;
6. $\mathbf{W}_t$ represents a $(p \times p)$ covariance matrix.

At $t = 0$, the initial information is that the mean and variance of $\mathbf{X}_0 \,|\, \mathbf{Y}_0$ are assumed to be known to be $\mathbf{m}_0$ and $\mathbf{C}_0$ respectively. It is further assumed that the error sequences $\mathbf{v}_t$ and $\mathbf{w}_t$ are uncorrelated with each other in all time periods and uncorrelated with the initial state $\mathbf{X}_0$, i.e. $\mathbb{E}(\mathbf{v}_s\mathbf{w}_t^{'}) = 0$, for all s and $t = 0, 1, \cdots$; and $\mathrm{E}(\mathbf{v}_t\mathbf{X}_0^{'}) = \mathrm{E}(\mathbf{w}_t\mathbf{X}_0^{'}) = 0$ for all t. For models with correlated errors, the inference procedure is very different which has been discussed in Nieto & Guerrero (1995). Serial correlation of errors can also be allowed for generalization of the problem, a partial treatment to which has been provided by Gelb et al. (1974). In real life situations, the elements $\mathbf{F}_t$, $\mathbf{G}_t$, $\mathbf{V}_t$ and $\mathbf{W}_t$ can be assumed to be unknown and dependent on a vector of parameters θ_t, which need to be estimated.

2.3.3 Nonlinear/Non-Gaussian Models

Several physical systems, such as stochastic volatility are better modeled as nonlinear or non-Gaussian or both. There are situations, for example data generated from ecological process which cannot be modeled using Gaussian distribution, even after transformations. Here it is more plausible to have non-Gaussian distribution, for example a skewed distribution (Frühwirth-Schnatter 1994). Also with discrete data, approximations using continuous Gaussian distributions may be inappropriate. Binary series occur as indicators of presence or absence of sequence of events, say for example daily rainfall indicators. Generalized models of this type are represented as the equations stated in (2.3.6) and (2.3.7), where the functions $f(\cdot)$ and $g(\cdot)$ are possibly nonlinear and the errors $\mathbf{v}_t$ and $\mathbf{w}_t$ are possibly non-Gaussian. Such models make filtering, smoothing or prediction much more complex since the posterior integrals are usually not obtainable in closed form, as in linear Gaussian models. Since analytical solution is hard to find, numerical methods are being developed to solve for these models (Arulampalam et al. 2002).

2.4 Inference for State Space models

In sequential estimation, a statistician is mainly interested in the three concepts of filtering, smoothing and prediction.

Filtering is the recovery of information, at time t, of the state process from the noisy observations up to time t. So if $\mathbf{X}_t$ denotes the non-measurable state process at time t as defined in Equation (2.3.7), and $\mathbf{Y}_{1:t} = \mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_t$ be the noisy observations until and including time t, then we are interested in the distribution $\mathbb{P}(\mathbf{x}_t \,|\, \mathbf{y}_{1:t})$, also known as the filtering density. If there are unknown parameters, say θ, then one might be interested in either $\mathbb{P}(\mathbf{x}_t \,|\, \mathbf{y}_{1:t}, \theta)$ or the unconditional (on θ) distribution $\mathbb{P}(\mathbf{x}_t \,|\, \mathbf{y}_{1:t})$. The latter is using the following integral:

$$\mathbb{P}(\mathbf{x}_t \,|\, \mathbf{y}_{1:t}) = \int \mathbb{P}(\mathbf{x}_t \,|\, \mathbf{y}_{1:t}, \theta)\mathbb{P}(\theta \,|\, \mathbf{y}_{1:t}) \, \mathrm{d}\theta. \tag{2.4.1}$$

A common example of filtering in real life is extraction of the "true" signal from a noisy transmitted or recorded radio signal, that has been discussed in the earlier chapter as a motivational example. A transmitter sends signals to a radio receiver which is inevitably corrupted by noise, and filtering is done to recover the original signal.

Smoothing differs from filtering in the sense that observations both before and after time t can be used to get information about the state at time t. This means unlike filtering, which tries to infer the distribution of the state in real time, there is a delay in smoothing regarding the inference. But this disadvantage can be remedied by the fact that since more observations are used, the inference will be more accurate. In smoothing we are actually interested in the distribution $\mathbb{P}(\mathbf{x}_t \,|\, \mathbf{y}_{1:s})$ where $s > t$. This can be explained intuitively by how the human brain tackles hastily written handwriting (Anderson & Moore 1979). When one word is difficult to interpret, words before and after it are used for understanding the difficult word.

Prediction is the same as forecasting. As explained earlier in Section 2.2.2, h-step ahead predictions, $\mathbb{P}(\mathbf{y}_{t+h} \,|\, \mathbf{y}_{1:t})$ are to be calculated. In time series modeling, the area of prediction is the one in which statisticians are mostly interested in. Ample examples of prediction are available in literature (Durbin & Koopman 2001).

The joint posterior $\mathbb{P}(\mathbf{x}_{1:T} \,|\, \mathbf{y}_{1:T})$ is also of interest, but it is not always possible to calculate it directly like the filtering process. The following decomposition is useful (Doucet & Johansen 2009):

$$\begin{aligned}\mathbb{P}(\mathbf{x}_{1:T} \,|\, \mathbf{y}_{1:T}) &= \mathbb{P}(\mathbf{x}_T \,|\, \mathbf{y}_{1:T}) \prod_{t=1}^{T-1} \mathbb{P}(\mathbf{x}_t \,|\, \mathbf{x}_{t+1}, \mathbf{y}_{1:T}), \\ &= \mathbb{P}(\mathbf{x}_T \,|\, \mathbf{y}_{1:T}) \prod_{t=1}^{T-1} \mathbb{P}(\mathbf{x}_t \,|\, \mathbf{x}_{t+1}, \mathbf{y}_{1:t}). \qquad (2.4.2)\end{aligned}$$

It is possible to further break the term $\mathbb{P}(\mathbf{x}_t \,|\, \mathbf{x}_{t+1}, \mathbf{y}_{1:t})$ in (2.4.2) down as

$$\mathbb{P}(\mathbf{x}_t \,|\, \mathbf{x}_{t+1}, \mathbf{y}_{1:t}) = \frac{\mathbb{P}(\mathbf{x}_{t+1} \,|\, \mathbf{x}_t)\mathbb{P}(\mathbf{x}_t \,|\, \mathbf{y}_{1:t})}{\mathbb{P}(\mathbf{x}_{t+1} \,|\, \mathbf{y}_{1:t})}. \qquad (2.4.3)$$

All the terms in (2.4.3) are computed using either the model equations or the filtering densities. Note that all the filtering and prediction densities computed up to T need to be stored for the calculation of Equation (2.4.2). It also possible to obtain a recursive formula for the joint posterior density (Doucet et al. 2001),

$$\mathbb{P}(\mathbf{x}_{1:T} \,|\, \mathbf{y}_{1:T}) = \mathbb{P}(\mathbf{x}_{1:T-1} \,|\, \mathbf{y}_{1:T-1}) \frac{\mathbb{P}(\mathbf{x}_T \,|\, \mathbf{x}_{T-1})\mathbb{P}(\mathbf{y}_T \,|\, \mathbf{x}_T)}{\mathbb{P}(\mathbf{y}_T \,|\, \mathbf{y}_{1:T-1})}. \qquad (2.4.4)$$

The denominator term in (2.4.4) is not easy to compute, but is a constant with respect to $\mathbf{X}_{1:T}$. The rest of the terms can be computed from sequential filtering equations. As the different methods developed in this thesis are discussed, it will be clear which of these decompositions can be used under what circumstances.

Sequential inference techniques for filtering, smoothing and prediction vary from being deterministic approximations to sampling based methods. The basis for deterministic inference is based on the seminal paper by Kalman (Kalman 1960, Kalman & Bucy 1961), which introduces the famous *Kalman filter*. Numerous extensions of Kalman filters exist, each of which tries to improve on the fundamental idea by solving for nonlinear and/or non-Gaussian models. A relatively new but more general class of numerical methods are the *Sequential Monte Carlo* algorithms. Other forms of smoothing and estimation techniques like EM algorithm or Expectation Propagation can also be used. The interested reader should look into Durbin & Koopman (2000) and Shumway & Stoffer (1982).

2.4.1 Kalman Filter

Kalman (Kalman 1960) developed a linear filter which produces an estimate by minimizing the conditional mean square error $\mathbb{E}\big[(\mathbf{x}_t - \hat{\mathbf{x}}_t)^T[(\mathbf{x}_t - \hat{\mathbf{x}}_t) \,|\, \mathbf{y}_{1:t}\big]$, where $\hat{\mathbf{x}}_t$ is the estimate of $\mathbf{x}_t \,|\, \mathbf{y}_{1:t}$. The Kalman filter is the optimum filter in the class of linear filters as given in Equations (2.3.10) and (2.3.11), in the sense that it is the minimum mean squared error estimator (MMSE). The Kalman filter propagates the first two moments of the distribution of $\mathbf{X}_t \,|\, \mathbf{Y}_{1:t}$. An interesting feature of this filter is that the model has to be linear while the errors can be non-Gaussian as has been shown in Kalman (1960). So derivation and the working principle for this filter works even if we drop the distributional assumption in Equations (2.3.10) and (2.3.11).

Let us assume at time $t-1$, the mean $\bar{\mathbf{x}}_{t-1\,|\,t-1}$ and variance $\mathbf{P}_{t-1\,|\,t-1}$ of $\mathbf{X}_{t-1} \,|\, \mathbf{Y}_{1:t-1}$ are known. The Kalman filter updates to the moments when new data $\mathbf{Y}_t$ arrive, are shown by the sequential equations below:

1. *Prior mean and variance of* $\mathbf{X}_t \,|\, \mathbf{Y}_{1:t-1}$:

$$\bar{\mathbf{x}}_{t\,|\,t-1} = \mathbf{G}_t \bar{\mathbf{x}}_{t-1\,|\,t-1}, \tag{2.4.5}$$

$$\mathbf{P}_{t\,|\,t-1} = \mathbf{G}_t \mathbf{P}_{t-1\,|\,t-1} \mathbf{G}_t^T + \mathbf{W}_t. \tag{2.4.6}$$

2. *1-step ahead forecast*:

$$\bar{\mathbf{y}}_{t\,|\,t-1} = \mathbf{F}_t^T \bar{\mathbf{x}}_{t\,|\,t-1}, \tag{2.4.7}$$

$$\mathbf{Q}_{t\,|\,t-1} = \mathbf{F}_t^T \mathbf{P}_{t\,|\,t-1} \mathbf{F}_t + \mathbf{V}_t. \tag{2.4.8}$$

3. *Posterior mean of* $\mathbf{X}_t \,|\, \mathbf{Y}_{1:t}$:

$$\bar{\mathbf{x}}_{t\,|\,t} = \bar{\mathbf{x}}_{t\,|\,t-1} + \mathbf{K}_t(\mathbf{y}_t - \bar{\mathbf{y}}_{t\,|\,t-1}), \tag{2.4.9}$$

$$\begin{aligned} \mathbf{P}_{t\,|\,t} &= \mathbf{P}_{t\,|\,t-1} - \mathbf{K}_t \mathbf{Q}_{t\,|\,t-1} \mathbf{K}_t^T, \\ \mathbf{K}_t &= \mathbf{P}_{t\,|\,t-1} \mathbf{F}_t \mathbf{Q}_{t\,|\,t-1}^{-1}. \end{aligned} \tag{2.4.10}$$

$\mathbf{K}_t$ is known as the *Kalman gain*. It ensures that the posterior means and covariances adapt to the new data. Kalman gain ensures the propagation of MMSE estimates over time.

The convergence of the Kalman filter depends on the dual concept of *observability* and *reachability*. We would not be providing the formal definitions for them (for more details see Walrand (2005)), but only explain their implications on the filter. Reachability ensures convergence of the filter. The observability condition guarantees that the estimation error remains bounded. These conditions ensure asymptotic stability of the filter, i.e. $\lim_{t\to\infty} \mathbf{P}_{t\,|\,t} \to \bar{\mathbf{P}}$ and $\lim_{t\to\infty} \mathbf{K}_t \to \mathbf{K}$, for some limiting values $\bar{\mathbf{P}}$ and $\mathbf{K}$ (West & Harrison 1997).

West & Harrison (1997) provides detailed proof of derivation of Kalman filter using two different methodology. Standard Bayesian calculations are used to deduce the posterior mean and variance of the filtering and prediction density. A second method uses the more intuitive properties of additivity, linearity and distributional closure properties belonging to a Gaussian distribution, where the model has Gaussian distribution for the error terms and state process at time 0, $\mathbf{X}_0 \,|\, \mathbf{Y}_0$.

2.4.2 Extensions of the Kalman Filter

In the previous section, we have mentioned that the Kalman filter, ideally, can be used for linear models with non-Gaussian errors. While implementing it for non-Gaussian errors, however, it is possible only if one could have closed form expressions for estimates of:

$$\bar{\mathbf{x}}_{t\,|\,t-1} = \mathbb{E}(\mathbf{x}_t \,|\, \mathbf{y}_{1:t-1}), \tag{2.4.11}$$

and

$$\bar{\mathbf{x}}_{t\,|\,t} = \mathbb{E}(\mathbf{x}_t \,|\, \mathbf{y}_{1:t}). \tag{2.4.12}$$

Thus the theoretical solution for the filter is more of a conceptual kind and cannot always be determined analytically in practice. For the observation vector originating from an exponential family distribution, conjugate methods exist, the predictive density $\mathbb{P}(\mathbf{Y}_t \,|\, \mathbf{Y}_{1:t-1})$ and the filtering density $\mathbb{P}(\mathbf{X}_t \,|\, \mathbf{Y}_{1:t-1})$ can be analytically shown to be same as a known probability distribution (West & Harrison 1997). Chen & Singpurwalla (1994) describe the non-Gaussian Kalman filter in a more general form, resorting to numerical techniques like Markov chain simulation (Gelfand & Smith 1990) to get solution for the expectations listed in Equations (2.4.11) and (2.4.12). Extensions of Kalman filter that provide suboptimal solutions have been developed over the years to work with such models. Such solutions are suboptimal since approximate assumptions are made about the nonlinear and non-Gaussian dynamic state space models. A good reference for such methods is Lefebvre et al. (2004). We compare a few of the extensions of the Kalman filter in the next section.

2.4.2.1 Extended Kalman Filter

The Extended Kalman Filter (EKF) gives an approximation to the optimal estimate (Haykin 2001). It does so by linearizing the nonlinear model around the last state estimate. So rather than propagating the nonlinearity, the EKF considers, at each iteration, a linearization of the nonlinear dynamics around the last predicted and filtered estimates of the state, and for the new linearized dynamics, it uses the Kalman filter. The state space model used with the EKF is given as:

$$\mathbf{y}_t = f(\mathbf{x}_t, \mathbf{u}_t) + \mathbf{v}_t, \quad \mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{V}_t), \tag{2.4.13}$$

$$\mathbf{x}_t = g(\mathbf{x}_{t-1}) + \mathbf{w}_t, \quad \mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{W}_t), \tag{2.4.14}$$

where $f(\cdot)$ and $g(\cdot)$ are nonlinear functions.
As before, we assume at time $t-1$, the moments $\bar{\mathbf{x}}_{t-1\,|\,t-1}$ and $\mathbf{P}_{t-1\,|\,t-1}$ are known. The filter dynamics with respect to the general equations are as follows:

1. *Prior mean and variance of* $\mathbf{X}_t \,|\, \mathbf{Y}_{1:t-1}$:

$$\bar{\mathbf{x}}_{t\,|\,t-1} = g(\bar{\mathbf{x}}_{t-1\,|\,t-1}, \mathbf{u}_t), \tag{2.4.15}$$

$$\mathbf{P}_{t\,|\,t-1} = \hat{\mathbf{G}}_t \mathbf{P}_{t-1\,|\,t-1} \hat{\mathbf{G}}_t^T + \mathbf{W}_t. \tag{2.4.16}$$

2. *1-step ahead forecast*:

$$\bar{\mathbf{y}}_{t\,|\,t-1} = f(\mathbf{x}_{t-1}), \tag{2.4.17}$$

$$\mathbf{Q}_{t\,|\,t-1} = \hat{\mathbf{F}}_t \mathbf{P}_{t\,|\,t-1} \hat{\mathbf{F}}_t^T + \mathbf{V}_t. \tag{2.4.18}$$

3. *Posterior mean of* $\mathbf{X}_t \,|\, \mathbf{Y}_{1:t}$:

$$\bar{\mathbf{x}}_{t\,|\,t} = \bar{\mathbf{x}}_{t\,|\,t-1} + \mathbf{K}_t(\mathbf{y}_t - \bar{\mathbf{y}}_{t\,|\,t-1}), \tag{2.4.19}$$

$$\mathbf{P}_{t\,|\,t} = \mathbf{P}_{t\,|\,t-1} - \mathbf{K}_t \hat{\mathbf{F}}_t \mathbf{Q}_{t\,|\,t-1} \mathbf{K}_t^T, \tag{2.4.20}$$

$$\mathbf{K}_t = \mathbf{P}_{t\,|\,t-1} \mathbf{F}_t \mathbf{Q}_{t\,|\,t-1}^{-1},$$

where,

$$\hat{\mathbf{G}}_t = \left.\frac{\partial g\left(\mathbf{x}\right)}{\partial \mathbf{x}}\right|_{\mathbf{x}=\bar{\mathbf{x}}_{t-1|t-1}},$$

$$\hat{\mathbf{F}}_t = \left.\frac{\partial f\left(\mathbf{x}\right)}{\partial \mathbf{x}}\right|_{\mathbf{x}=\bar{\mathbf{x}}_{t|t-1}}.$$

It is obvious that the EKF utilizes the first term of the Taylor's series expansion of the nonlinear functions f and g to estimate $\hat{\mathbf{F}}_t$ and $\hat{\mathbf{G}}_t$, respectively (Arulampalam et al. 2002). If the functions are highly nonlinear, the performance of the EKF can be very poor. One would expect these filters to do well in high signal-to-noise ratio (SNR) situations. It is easy to see that high SNR essentially implies low measurement error, which in turn means accurate measurement resulting in large reduction in estimate's uncertainty. It can be shown that as $SNR \to 0$, $K \to 0$, which means no correction to the posterior means and variance in Equations (2.4.19) and (2.4.20).

Several variations to the EKF have been proposed to improve its performance and these methods fundamentally try to relax some of the assumptions made in EKF. One can include higher order terms of the Taylor's series expansion of the nonlinear terms. A filter which includes two terms from the Taylor series is known as a *second order EKF* (Arulampalam et al. 2002). However these extensions come with the price of higher complexity and increased computation, which particularly is a problem for higher dimensional systems. Also there are other nonlinear filters like the *Gaussian sum filter* (GSF), which uses a finite sum Gaussians and in the process involve a collection of EKF for each of the approximate Gaussians. This the main reason why GSF is more powerful than a single EKF Terejanu et al. (2011).

2.4.2.2 Unscented Kalman filter

There exist a number of *derivative-free filters* which have been developed to improve the performance of the EKF. These include the *Unscented Kalman filter* (UKF), the *Central Difference filter* (CDF) (Ito & Xiong 1999) and the *Divided Difference filter* (DDF) (Nørgaard et al. 2000*a*,*b*). It can be shown that the latter two are equivalent Lefebvre et al. (2004). These filters usually outperform EKF at an equal computational complexity of $\mathcal{O}(n^3)$, where . A careful analysis of Taylor's series expansion of the nonlinear functions, shows that both UKF and CDF are essentially the same. Both filters calculate the posterior mean in exactly the same way. The only difference lies in the form of approximation to the posterior covariance. The CDF ensures positive semi-definiteness of the posterior covariance matrix as opposed to the fact that UKF may produce non-positive semi-definiteness.

Unlike EKF, the Unscented Kalman Filter does not make any approximation to the nonlinear process (Julier & Uhlmann 1997, Julier et al. 1995). It uses the "true" nonlinear system and the only approximation is done on the distribution of the process. The nonlinearity is propagated through deterministically chosen points which attempt to capture the "true" mean and covariance of the variables. The posterior mean and covariance are also calculated up to second order using these chosen points.

As always, let us assume that at time $t-1$, the mean and covariance of $\mathbf{X}_{t-1} \mid \mathbf{Y}_{1:t-1}, \theta$ are given by $\bar{\mathbf{x}}_{t-1\,|\,t-1}$ and $\mathbf{P}_{t-1\,|\,t-1}$ respectively. Remembering that the state vector is of dimension p, a set of $2p+1$ weighted samples or *sigma points* are deterministically chosen.

The procedure of choosing points is being shown in the following equations:

$$\left.\begin{array}{lll} \mathcal{X}_0 & = \bar{\mathbf{x}}_{t-1\,|\,t-1} & \\ \mathcal{X}_i & = \bar{\mathbf{x}}_{t-1\,|\,t-1} + \left(\sqrt{(p+\lambda)\mathbf{P}_{t-1\,|\,t-1}}\right)_i & i = 1, \ldots, p \\ \mathcal{X}_i & = \bar{\mathbf{x}}_{t-1\,|\,t-1} - \left(\sqrt{(p+\lambda)\mathbf{P}_{t-1\,|\,t-1}}\right)_i & i = p+1, \ldots, p \end{array}\right\} \begin{array}{l}\textbf{Sigma} \\ \textbf{Points},\end{array} \qquad (2.4.21)$$

$$\left.\begin{array}{ll} \mathcal{W}_0^m & = \frac{\lambda}{p+\lambda} \\ \mathcal{W}_0^c & = \frac{\lambda}{p+\lambda} + \left(1 - \alpha^2 + \beta\right) \\ \mathcal{W}_i^m & = \mathcal{W}_i^c = \frac{1}{2(p+\lambda)} \quad i = 1, \ldots, 2p \end{array}\right\} \textbf{Weights}, \qquad (2.4.22)$$

and

$$\lambda = \alpha^2 \left(p + \kappa\right) - p.$$

The $\left(\sqrt{\mathbf{P}_{t-1\,|\,t-1}}\right)_i$ in (2.4.21) is the i^{th} row or column of the matrix square-root of $\mathbf{P}_{t-1\,|\,t-1}$. The square root of matrix is performed by well known linear algebra techniques, for e.g. Cholesky decomposition or QR decomposition Merwe & Wan (2001).

Two different types of weighting sequence are explained in (2.4.22). It will be seen later that $\mathcal{W}^m$ is used in the computation of mean, whereas $\mathcal{W}^c$ is similarly used to compute the covariance (see, for example, Equations (2.4.23) and (2.4.24)). The parameters α, κ and β have to be chosen according to the following constraints. Positive semi-definivity of prior/posterior covariance matrix is guaranteed by choosing κ greater than 0. The algorithm in general however, is quite robust to different choices of κ, hence a good default choice is 0. The "size" of the sigma point distribution is controlled by α $(0 \leq \alpha \leq 1)$ and should ideally be chosen to be small to avoid sampling non-local effects when nonlinearity is strong. The third parameter β, is a non-negative weighing parameter which allows for minimization of higher order errors if prior knowledge of the distribution of $\mathbf{X}_{t-1}$ is available. In other words, it is used to pool in knowledge about higher order moments of the distribution. For a Gaussian prior, the optimal choice is 2.

The rest of the algorithm follows as a normal KF filter algorithm doing updates to the mean and covariance of $\mathbf{X}_t$. If we have additive Gaussian errors, for example Equations (2.4.13) and (2.4.14), the UKF algorithm can be simplified in the sense that computation of the variance terms become easier than that in non-additive errors. We present the simplified case here. Thus at time $t-1$, assuming that the sigma points are defined as $\mathcal{X}_{i,t-1}, i = 1, \ldots, 2p$, they are first passed through the nonlinear system equation and the prior mean and covariance computed in (2.4.23) and (2.4.24).

$$\begin{aligned}
\mathcal{X}_{i,t-1\,|\,t-1} &= g\left(\mathcal{X}_{i,t-1}\right) \quad i = 1,\ldots,2p \\
\bar{\mathbf{x}}_{t\,|\,t-1} &= \sum_{i=1}^{2p} \mathcal{W}_i^m \mathcal{X}_{i,t-1\,|\,t-1} && (2.4.23) \\
\mathbf{P}_{t\,|\,t-1} &= \sum_{i=1}^{2p} \mathcal{W}_i^c \left\{\mathcal{X}_{i,t-1\,|\,t-1} - \bar{\mathbf{x}}_{t\,|\,t-1}\right) \left(\mathcal{X}_{i,t-1\,|\,t-1} - \bar{\mathbf{x}}_{t\,|\,t-1}\right)^T + \mathbf{W}_t && (2.4.24)
\end{aligned}$$

Subsequently, the sigma points are then propagated through the nonlinear observation equation $f(\cdot)$.

$$\begin{aligned}
\mathcal{Y}_{i,t-1\,|\,t-1} &= f\left(\mathcal{X}_{i,t-1\,|\,t-1}\right) + \mathbf{0} \quad i = 1,\ldots,2p \\
\bar{\mathbf{y}}_{t\,|\,t-1} &= \sum_{i=1}^{2p} \mathcal{W}_i^m \mathcal{Y}_{i,t-1\,|\,t-1} && (2.4.25) \\
\mathbf{P}_{y_{t\,|\,t-1}} &= \sum_{i=1}^{2p} \mathcal{W}_i^c \left\{\mathcal{Y}_{i,t-1\,|\,t-1} - \bar{\mathbf{y}}_{t\,|\,t-1}\right) \left(\mathcal{Y}_{i,t-1\,|\,t-1} - \bar{\mathbf{y}}_{t\,|\,t-1}\right)^T + \mathbf{V}_t && (2.4.26) \\
\mathbf{P}_{x_{t-1}y_{t-1}} &= \sum_{i=1}^{2p} \mathcal{W}_i^c \left\{\mathcal{X}_{i,t-1\,|\,t-1} - \bar{\mathbf{x}}_{t\,|\,t-1}\right) \left(\mathcal{Y}_{i,t-1\,|\,t-1} - \bar{\mathbf{y}}_{t\,|\,t-1}\right)^T + \mathbf{V}_t \\
\mathbf{K}_t &= \mathbf{P}_{x_{t-1}y_{t-1}} \mathbf{P}_{y_{t\,|\,t-1}}^{-1} \\
\bar{\mathbf{x}}_{t\,|\,t} &= \bar{\mathbf{x}}_{t\,|\,t-1} + \mathbf{K}_t \left(\mathbf{y}_t - \bar{\mathbf{y}}_{t\,|\,t-1}\right) && (2.4.27) \\
\mathbf{P}_{t\,|\,t} &= \mathbf{P}_{t\,|\,t-1} + \mathbf{K}_t \mathbf{P}_{y_{t\,|\,t-1}} \mathbf{K}_t^T && (2.4.28)
\end{aligned}$$

The UKF, through its propagation of sigma points retaining the nonlinearity, results in approximations of mean and covariance that are correct up to third order for Gaussian errors for all nonlinearities and up to second order for non-Gaussian error inputs. Also, since the distribution of the state and measurement process are approximated rather than the nonlinearities, this filter partially incorporates higher order moments information and hence performs better than the EKF Ito & Xiong (1999). The UKF has been shown to be a superior alternative to the EKF in a variety of applications including state estimation for road vehicle navigation (LaViola 2003), parameter estimation for time series modelling (Wan & Merwe 2000), and neural network training (Wan et al. 2000).

2.4.2.3 Other Filters

There exist other extensions to the Kalman filter which, like the previous ones, assume a Gaussian distribution for the measurement and state process, but calculate the posterior mean and posterior covariance matrix via different methods. The *Ensemble Kalman Filter* (EnKF) (Evensen 2009) was proposed as a stochastic or Monte Carlo alternative to deterministic methods, while still assuming that the probability distributions involved are

Gaussian. Efficient numerical integration of the Gaussian filter based on Gauss-Hermite integration rule has resulted in the *Quadrature Kalman Filter* (QKF) (Ito & Xiong 1999, Arasaratnam et al. 2007). Even more numerical stability is provided by the *Cubature Kalman Filter* (CKF) (Arasaratnam & Haykin 2009) than those filters just listed. This filter can work for higher dimensions of state process than that of Quadrature filters.

While all these filters have always approximated the distribution by a Gaussian distribution, there are a few which have tried to avoid doing so. The Gram-Charlier or Edgeworth expansion, a series of Hermite polynomials, can be used to approximate a wide class of density functions (Sorenson & Stubberud 1968, Srinivasan 1970). These methods can approximate mostly unimodal non-Gaussian density functions but they are known to suffer from the fact that a large number of terms are required for the approximation. Another approach is the Gaussian sum representation, which uses the fact that any density can be approximated by a finite mixture of Gaussian densities up to a desired accuracy (Alspach & Sorenson 1972). The basic idea is to approximate the non-Gaussian densities by a mixture of Gaussians, where the mean and covariance of the Gaussian distributions are propagated through parallel EKF's.

2.4.2.4 Divergence

The prediction performance of an algorithm in the context of real data can sometimes be poor. This can happen due to several factors namely: (a) inaccurate or incomplete model, (b) wrongly assuming a Gaussian or unimodal density, (c) a very high degree of nonlinearity and (d) numerical instability errors (Arasaratnam & Haykin 2009). If the bias associated with the filter becomes large but bounded then the divergence is known as *apparent* divergence. However, the problem becomes serious if the bias eventually becomes infinite (Gelb et al. 1974). An example is provided in Perea et al. (2007) where divergence happens while determining the orbit of a satellite formation using Global Positioning System and signals.

2.4.3 Sequential Monte Carlo Methods

Ever since the introduction of this method by Gordon et al. (1993), *Sequential Monte Carlo* (SMC) algorithms have been a very useful and popular class of numerical methods for implementing estimation in sequential nonlinear non-Gaussian problems. In comparison to the methods which are extensions of the Kalman filter, these methods do not rely on any local linearization technique or distributional assumption, and are also known as particle filters. The aim of SMC methods is more general than that of non-Monte Carlo based algorithms. The fundamental objective is to estimate recursively in time the *joint posterior distribution* $\mathbb{P}(\mathbf{x}_{1:t} \,|\, \mathbf{y}_{1:t})$, and its associated features, namely the filtering distribution $\mathbb{P}(\mathbf{x}_t \,|\, \mathbf{y}_{1:t})$ and expectations:

$$\mathbb{E}_{(\mathbf{X}_{1:t} \,|\, \mathbf{Y}_{1:t})}[f_t(\mathbf{x}_{1:t})] = \int f_t(\mathbf{x}_{1:t})\mathbb{P}(\mathbf{x}_{1:t} \,|\, \mathbf{y}_{1:t})\, \mathrm{d}\mathbf{x}_{1:t},$$

for some function of interest $f_t(\cdot)$, integrable with respect to $\mathbb{P}(\mathbf{x}_{1:t} \,|\, \mathbf{y}_{1:t})$.

A recursive formula for the joint posterior distribution has been formulated in (2.4.4) and is restated here for clarity:

$$\mathbb{P}(\mathbf{x}_{1:t} \,|\, \mathbf{y}_{1:t}) = \mathbb{P}(\mathbf{x}_{1:t-1} \,|\, \mathbf{y}_{1:t-1})\frac{\mathbb{P}(\mathbf{x}_t \,|\, \mathbf{x}_{t-1})\mathbb{P}(\mathbf{y}_t \,|\, \mathbf{x}_t)}{\mathbb{P}(\mathbf{y}_t \,|\, \mathbf{y}_{1:t-1})}.$$

Also note that,

$$\mathbb{P}(\mathbf{y}_{1:t}) = \mathbb{P}(\mathbf{y}_{1:t-1})\, \mathbb{P}(\mathbf{y}_t \,|\, \mathbf{y}_{1:t-1}), \tag{2.4.29}$$

where

$$\mathbb{P}(\mathbf{y}_t \,|\, \mathbf{y}_{1:t-1}) = \int \mathbb{P}(\mathbf{x}_t \,|\, \mathbf{x}_{t-1})\mathbb{P}(\mathbf{y}_t \,|\, \mathbf{x}_t)\mathbb{P}(\mathbf{x}_{t-1} \,|\, \mathbf{y}_{1:t-1})\, \mathrm{d}\mathbf{x}_{t-1:t}. \tag{2.4.30}$$

The marginal distribution $\mathbb{P}(\mathbf{x}_t \,|\, \mathbf{y}_{1:t})$ also follows the following recursion:

$$\mathbb{P}(\mathbf{x}_t \,|\, \mathbf{y}_{1:t-1}) = \int \mathbb{P}(\mathbf{x}_t \,|\, \mathbf{x}_{t-1})\mathbb{P}(\mathbf{x}_{t-1} \,|\, \mathbf{y}_{1:t-1})\, \mathrm{d}\mathbf{x}_{t-1}, \tag{2.4.31}$$

$$\mathbb{P}(\mathbf{x}_t \,|\, \mathbf{y}_{1:t}) = \frac{\mathbb{P}(\mathbf{y}_t \,|\, \mathbf{x}_t)\mathbb{P}(\mathbf{x}_t \,|\, \mathbf{y}_{1:t-1})}{\int \mathbb{P}(\mathbf{y}_t \,|\, \mathbf{x}_t)\mathbb{P}(\mathbf{x}_t \,|\, \mathbf{y}_{1:t-1})\, \mathrm{d}\mathbf{x}_t}. \tag{2.4.32}$$

Note that, because no assumptions are made about the distributions involved, the integrals are not available in closed form and hence evaluation of these integrals is non-trivial.

The idea behind these methods is that if one has a large number of samples drawn from a particular posterior distribution, then it is possible to solve for integrals involving marginals or expectations quite easily. However, obtaining samples directly from the posterior distribution is often not possible. Hence one has to resort to Monte Carlo based methods, such as *Importance sampling.*

2.4.3.1 Sequential Importance Sampling Filter (SIS)

Importance Sampling:

We start by explaining a classical solution to solving Monte Carlo integrals known as importance sampling, for example see Geweke (1989). Let us assume that we have N independent and identically distributed random samples, also known as particles, $\mathbf{x}_{1:t}^{(i)}$; $i = 1, \cdots, N$ drawn from $\mathbb{P}(\mathbf{x}_{1:t} \,|\, \mathbf{y}_{1:t})$. An empirical estimate of this distribution is given by:

$$\mathbb{P}_N(\mathbf{x}_{1:t} \,|\, \mathbf{y}_{1:t}) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\mathbf{x}_{1:t}^{(i)}}(\mathbf{x}_{1:t}).$$

where $\delta_{\mathbf{x}_{1:t}^{(i)}}(\mathbf{x}_{1:t})$ is the Dirac-delta mass located in $\mathbf{x}_{1:t}^{(i)}$. The estimate of the expectation is

$$\mathbb{E}_N[f_t(\mathbf{x}_{1:t})] = \frac{1}{N}\sum_{i=1}^{N} f_t(\mathbf{x}_{1:t}^{(i)}).$$

The general Monte Carlo integration method is essentially an applied form of the law of large numbers Gilks et al. (1996*a*). It has very good convergence properties, which are based on CLT, and also the rate of convergence of the estimate is independent of the dimension of the integrand (see Geweke (1989) and Roberts & Rosenthal (2004) for a detailed account of the convergence theorems). In contrast, any deterministic numerical integration method has a rate of convergence that decreases as the dimension of the integrand increases (Doucet et al. 2001).

A classical solution for sampling from a posterior density is *importance sampling* . Suppose we want to sample from $\mathbb{P}(\mathbf{x}_{1:t} \,|\, \mathbf{y}_{1:t})$. Let there be an arbitrary distribution $\pi(\mathbf{x}_{1:t} \,|\, \mathbf{y}_{1:t})$, known as importance sampling distribution or proposal distribution from which it is easy to sample. $\pi(\mathbf{x}_{1:t} \,|\, \mathbf{y}_{1:t})$ includes the support of $\mathbb{P}(\mathbf{x}_{1:t} \,|\, \mathbf{y}_{1:t})$, and the expectation is derived as:

$$\mathbb{E}[f_t(\mathbf{x}_{1:t})] = \frac{\int f_t(\mathbf{x}_{1:t}) w(\mathbf{x}_{1:t}) \pi(\mathbf{x}_{1:t} \,|\, \mathbf{y}_{1:t})\, \mathrm{d}\mathbf{x}_{1:t}}{\int w(\mathbf{x}_{1:t}) \pi(\mathbf{x}_{1:t} \,|\, \mathbf{y}_{1:t})\, \mathrm{d}\mathbf{x}_{1:t}}, \tag{2.4.33}$$

where $w(\mathbf{x}_{1:t})$ is known as the importance weight and is defined as,

$$w(\mathbf{x}_{1:t}) = \frac{\mathbb{P}(\mathbf{x}_{1:t} \,|\, \mathbf{y}_{1:t})}{\pi(\mathbf{x}_{1:t} \,|\, \mathbf{y}_{1:t})}. \tag{2.4.34}$$

So, if we assume that we have N i.i.d. particles $\mathbf{x}_{1:t}^{(i)}$; $i = 1, \cdots, N$ sampled from $\pi(\mathbf{x}_{1:t} \,|\, \mathbf{y}_{1:t})$, a Monte Carlo approximation for $\mathbb{P}(\mathbf{x}_{1:t} \,|\, \mathbf{y}_{1:t})$ is given by:

$$\mathbb{P}_N(\mathbf{x}_{1:t} \,|\, \mathbf{y}_{1:t}) = \sum_{i=1}^{N} \tilde{w}_t^{(i)} \delta_{\mathbf{x}_{1:t}^{(i)}}(\mathbf{x}_{1:t}),$$

where the normalized importance weights are given as

$$\tilde{w}_t^{(i)} = \frac{w(\mathbf{x}_{1:t}^{(i)})}{\sum_{j=1}^{N} w(\mathbf{x}_{1:t}^{(j)})}. \tag{2.4.35}$$

Also an estimate for the expectation in (2.4.33) is

$$\mathbb{E}_N[f_t(\mathbf{x}_{1:t})] = \frac{\frac{1}{N}\sum_{i=1}^{N} f_t(\mathbf{x}_{1:t}^{(i)}) w(\mathbf{x}_{1:t}^{(i)})}{\frac{1}{N}\sum_{i=1}^{N} w(\mathbf{x}_{1:t}^{(i)})} = \sum_{1=1}^{N} f_t(\mathbf{x}_{1:t}^{(i)}) \tilde{w}_t^{(i)}, \tag{2.4.36}$$

Importance sampling in its simplest form cannot be applied recursively as is evident from the equations above. One needs to get all the data before estimating the filtering density. Hence, in general, each time a new data, $\mathbf{y}_{t+1}$ say, become available, the importance weights need to be recomputed all over again. This becomes computationally infeasible and hence SIS overcomes this problem by defining a recursive formula for the weights.

The proposal distribution at time $t-1; \pi(\mathbf{x}_{1:t-1} \,|\, \mathbf{y}_{1:t-1})$ has to be a marginal for the proposal distribution at time $t; \pi(\mathbf{x}_{1:t} \,|\, \mathbf{y}_{1:t})$, i.e.

$$\pi(\mathbf{x}_{1:t} \,|\, \mathbf{y}_{1:t}) = \pi(\mathbf{x}_{1:t-1} \,|\, \mathbf{y}_{1:t-1})\, \pi(\mathbf{x}_t \,|\, \mathbf{x}_{1:t-1}, \mathbf{y}_{1:t}). \tag{2.4.37}$$

So if one has samples $\mathbf{x}_{1:t-1}^{(i)}$; $i = 1, \cdots, N$ from the proposal $\pi(\mathbf{x}_{1:t-1} \,|\, \mathbf{y}_{1:t-1})$ and new samples $\mathbf{x}_t^{(i)}$; $i = 1, \cdots, N$ are generated from the density $\pi(\mathbf{x}_t \,|\, \mathbf{x}_{1:t-1}, \mathbf{y}_{1:t})$, then they can be augmented so that the new set, $\mathbf{x}_{1:t}^{(i)}$; $i = 1, \cdots, N$ becomes a sample from the proposal at time t.

To get an updating equation for the weights, first note that

$$\mathbb{P}(\mathbf{x}_{1:t} \,|\, \mathbf{y}_{1:t}) \propto \mathbb{P}(\mathbf{y}_t \,|\, \mathbf{x}_t)\, \mathbb{P}(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}).\, \mathbb{P}(\mathbf{x}_{1:t-1} \,|\, \mathbf{y}_{1:t-1}) \tag{2.4.38}$$

Using (2.4.37) and (2.4.38), the weight recursion is given as,

$$\tilde{w}_t^{(i)} = \tilde{w}_{t-1}^{(i)} \frac{\mathbb{P}(\mathbf{y}_t \,|\, \mathbf{x}_t^{(i)})\, \mathbb{P}(\mathbf{x}_t^{(i)} \,|\, \mathbf{x}_{t-1}^{(i)})}{\pi(\mathbf{x}_t^{(i)} \,|\, \mathbf{x}_{1:t-1}^{(i)}, \mathbf{y}_{1:t})}. \tag{2.4.39}$$

Furthermore, if the denominator in (2.4.39) can be approximated as $\pi(\mathbf{x}_t \,|\, \mathbf{x}_{1:t-1}, \mathbf{y}_{1:t}) \approx \pi(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}, \mathbf{y}_t)$, then the importance density only depends on $\mathbf{x}_{t-1}$ and $\mathbf{y}_t$. For a situation where only the filtered estimate is needed at each time point, this is quite helpful since one needs to store only $\mathbf{x}_{t-1}^{(i)}$ and can throw away $\mathbf{x}_{1:t-2}^{(i)}$. The modified weight is then

$$\tilde{w}_t^{(i)} \propto \tilde{w}_{t-1}^{(i)} \frac{\mathbb{P}(\mathbf{y}_t \,|\, \mathbf{x}_t^{(i)})\, \mathbb{P}(\mathbf{x}_t^{(i)} \,|\, \mathbf{x}_{t-1}^{(i)})}{\pi(\mathbf{x}_t^{(i)} \,|\, \mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t)}. \tag{2.4.40}$$

From the recursion in Equation 2.4.40, its is possible to define the weights also in the following way:

$$w(\mathbf{x}_{t-1:t}) = \frac{\mathbb{P}(\mathbf{y}_t \,|\, \mathbf{x}_t)\, \mathbb{P}(\mathbf{x}_t \,|\, \mathbf{x}_{t-1})}{\pi(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}, \mathbf{y}_t)}. \tag{2.4.41}$$

Using Equation 2.4.41 an estimation of $\mathbb{P}(\mathbf{y}_t \,|\, \mathbf{y}_{1:t-1})$ is given by

$$\mathbb{P}_N(\mathbf{y}_t \,|\, \mathbf{y}_{1:t-1}) = \frac{1}{N} \sum_{i=1}^{N} w(\mathbf{x}_{t-1:t}^{(i)}).$$

This provides us with an estimate of the marginal likelihood, as defined in Equation 2.4.29, and is given by

$$\mathbb{P}_N(\mathbf{y}_{1:t}) = \mathbb{P}_N(\mathbf{y}_1) \prod_{j=1}^{t} \mathbb{P}_N(\mathbf{y}_i \,|\, \mathbf{y}_{1:i-1}). \tag{2.4.42}$$

There are problems associated with SIS though. One has to remember that it is only an extension of the standard IS algorithm and it carries all the problems associated with IS, for example choice of importance density and ensuring bounded variance for the weights. Also carrying forward all the particles is a handicap for high-dimensional state variables.

A major problem that SIS suffers from is *degeneracy* where, after a few iterations, all but one particle will have negligible normalised weights. It is a known fact that in importance sampling the variance of the weights increases, typically exponentially with the number of observations (Doucet, Godsill & Andrieu 2000). This is the same with SIS (Doucet & Johansen 2009). As the variance of the weights increase, it is easy to see from Equation (2.4.35) that degeneracy will set in. A suitable measure of degeneracy is the *effective sample size* N_{eff} introduced in Liu & Che (1998) and defined as:

$$N_{eff} = \frac{N}{1 + Var_{\pi(\cdot\,|\,\mathbf{y}_{1:t})}(w(\mathbf{x}_{1:t}))}, \tag{2.4.43}$$

$$= \frac{N}{\mathbb{E}_{\pi(\cdot\,|\,\mathbf{y}_{1:t})}[(w(\mathbf{x}_{1:t}))^2]}. \tag{2.4.44}$$

It is not possible to estimate it exactly, but an estimate $\hat{N}_{eff}$ of N_{eff} is given as

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N}(\tilde{w}_t^{(i)})^2}.$$

Severe degeneracy is indicated when the value of N_{eff} is small, typically below some threshold N_{thres}. Several ways have been prescribed in a view to limit the degeneracy problem. Some of them are listed below:

Resampling

Resampling is a very nice and intuitive idea to stabilize the SIS. The basic idea behind resampling is to eliminate the particles that have small weights and give more importance to those which have higher weights. First consider that $\hat{\mathbb{P}}(\mathbf{x}_{1:t}\,|\,\mathbf{y}_{1:t})$ is an IS approximation to the actual posterior $\mathbb{P}(\mathbf{x}_{1:t}\,|\,\mathbf{y}_{1:t})$. This approximation is based on weighted samples from $\pi(\mathbf{x}_{1:t}\,|\,\mathbf{y}_{1:t})$. So to obtain approximate samples from $\mathbb{P}(\mathbf{x}_{1:t}\,|\,\mathbf{y}_{1:t})$, one just samples from $\hat{\mathbb{P}}(\mathbf{x}_{1:t}\,|\,\mathbf{y}_{1:t})$, i.e. draws samples $\mathbf{x}_{1:t-1}^{(i)}$ with probability $w_t^{(i)}$. This is called resampling since sampling is done from an approximate distribution which was itself obtained by sampling. The resulting sample is hence an i.i.d. sample from the approximate discrete density where the weights are reset to $\frac{1}{N}$. It is possible to sample here in $\mathcal{O}(N)$ operations. Improved and efficient methods of resampling have been proposed, namely systematic sampling (Kitagawa 1996) and residual sampling (Liu & Che 1998). It should be noted that other problems are introduced with the introduction of resampling. Particles that have high weights are selected many times leading to loss of diversity among the particles, also known as *sample impoverishment*. The essentially leads smoothing problems to degeneracy. See Arulampalam et al. (2002) and Doucet & Johansen (2009) for a detailed view on this topic.

Choice of Importance Density

The choice of the proposal distribution is quite important in importance sampling and obviously in SIS too. The optimum proposal distribution that minimizes the variance of the true weights conditioned on $\mathbf{x}_{t-1}^{(i)}$ and $\mathbf{y}_t$ has been shown to be,

$$\pi(\mathbf{x}_t^{(i)} \,|\, \mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t) = \mathbb{P}(\mathbf{x}_t^{(i)} \,|\, \mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t). \tag{2.4.45}$$

Substituting the optimum proposal density (2.4.45) into (2.4.40) gives us the weight recursion:

$$\tilde{w}_t^{(i)} \propto \tilde{w}_{t-1}^{(i)} \, \mathbb{P}(\mathbf{y}_t \,|\, \mathbf{x}_{t-1}^{(i)}). \tag{2.4.46}$$

However it has the major drawback that one requires the ability to draw samples from the density $\mathbb{P}(\mathbf{x}_t^{(i)} \,|\, \mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t)$ which is not possible for many models. Another important approach is to adopt the prior distribution as the proposal density (see Handschin & Mayne (1969)) i.e.

$$\pi(\mathbf{x}_{1:t} \,|\, \mathbf{y}_{1:t}) = \mathbb{P}(\mathbf{x}_{1:t}) = \mathbb{P}(\mathbf{x}_1) \prod_{i=2}^{t} \mathbb{P}(\mathbf{x}_i \,|\, \mathbf{x}_{i-1}). \tag{2.4.47}$$

In this case the modified weight recursion becomes

$$\tilde{w}_t^{(i)} \propto \tilde{w}_{t-1}^{(i)} \, \mathbb{P}(\mathbf{y}_t \,|\, \mathbf{x}_t^{(i)}). \tag{2.4.48}$$

The method is often inefficient in simulations as the state space is explored without any knowledge of the observations. It is especially sensitive to outliers. However, it does have the advantage that the importance weights are easily evaluated. So one can see that the design of the importance density among other things heavily influences the performance of a particle filter.

Several other versions of particle filters exist. Some have been found to outperform others in specific situations. But nearly all the particle filters are regarded as special cases or extensions of the SIS algorithm. The special cases are derived by an appropriate choice of proposal density and/or a modification to the sampling-resampling scheme. Other particle filters that are popular and use the resampling scheme at each step are: *Sampling Importance Resampling* (SIR) filter (Gordon et al. 1993), *Auxiliary Sampling Importance Resampling* ASIR filter (also known as Auxiliary particle filter), which uses an auxiliary variable (Pitt & Shephard 1999), *Regularized Particle Filter* (RPF) where the resampling is done from a continuous approximation to the posterior (Doucet et al. 2001). It should be mentioned here that even though these methods make use of the optimal proposal density, the algorithm is not efficient. For example if the variability of $\mathbb{P}(\mathbf{y}_t \,|\, \mathbf{x}_{t-1})$ is high, then resampling will have to be used very frequently and the particle approximation to the posterior will be unreliable. Other filters like the *Resample-Move* filter (Doucet et al. 2001) use MCMC algorithms like the Gibbs sampler (Gelfand & Smith 1990) or the Metropolis Hastings (MH) algorithm (Metropolis et al. 1953, Hastings 1970) in SMC filtering; the *Rao-Blackwellised Particle Filter* (RPBF), which samples some of the variables and marginalizes the rest exactly using some finite dimensional optimal filter, are used to overcome these problems (Doucet, de Freitas, Murphy & Russell 2000).

The advantages and disadvantages of SMC filters have been listed in the above discussion. It is easy to see that, while on one hand SMC can handle arbitrary models with arbitrary densities without making any approximation related to them, the computational complexity involved in the process can be high, in addition to problems like degeneracy and choice of proposal density. One more problem is to determine the optimum number of particles suitable for the particular density. There is no general methodology that is followed in this context since it can be shown that no matter how many particles are chosen, the degeneracy problem will set in for large t (Andrieu et al. 2005). Attempts have been made to provide a remedy to this problem by Poyiadjis et al. (2011).

2.4.4 Grid Based Filters

Grid based filtering (GBF) is yet another methodology of recursive Bayesian estimation. These methods approximate the posterior density by a discrete set of points, also known as *discrete grid.* We provide a detailed explanation of this filter in the remaining part of this section.

For a discrete distribution of the state process, these methods provide the optimal recursion for the filter density. Suppose that the state space at time $t-1$, consists of N discrete states $\mathbf{X}^i_{t-1}; \quad i = 1, \cdots, N$. For each state $\mathbf{X}^i_{t-1}$, let the posterior density $\mathbb{P}(\mathbf{x}_{t-1} = \mathbf{x}^i_{t-1} \,|\, \mathbf{y}_{1:t-1})$ be denoted by $w^i_{t-1\,|\,t-1}$. So the posterior density at time $t-1$ can be written as

$$\mathbb{P}(\mathbf{x}_{t-1} \,|\, \mathbf{y}_{1:t-1}) = \sum_{i=1}^{N} w^i_{t-1\,|\,t-1} \delta(\mathbf{x}_{t-1} - \mathbf{x}^i_{t-1}), \tag{2.4.49}$$

where $\delta(\cdot)$ as usual is the Dirac-delta function. Using the density in (2.4.49) as the prior at time t when a new observation is recorded, the posterior of the prediction and filtering density are can be approximated by:

$$\mathbb{P}(\mathbf{x}_t \,|\, \mathbf{y}_{1:t-1}) = \sum_{i=1}^{N} w^i_{t\,|\,t-1} \delta(\mathbf{x}_t - \mathbf{x}^i_t), \tag{2.4.50}$$

$$\mathbb{P}(\mathbf{x}_t \,|\, \mathbf{y}_{1:t}) = \sum_{i=1}^{N} w^i_{t\,|\,t} \delta(\mathbf{x}_t - \mathbf{x}^i_t), \tag{2.4.51}$$

where the weights are defined as

$$w^i_{t\,|\,t-1} = \sum_{j=1}^{N} w^j_{t-1\,|\,t-1}\,\mathbb{P}(\mathbf{x}^i_t\,|\,\mathbf{x}^j_{t-1}),$$

$$w^i_{t\,|\,t} = \frac{w^i_{t\,|\,t-1}\,\mathbb{P}(\mathbf{y}_t\,|\,\mathbf{x}^i_t)}{\sum_{j=1}^{N} w^j_{t\,|\,t-1}\,\mathbb{P}(\mathbf{y}_t\,|\,\mathbf{x}^j_t)}.$$

It is easy to see that in these methods one only needs to assume that the transition density and the likelihood, given by $\mathbb{P}(\mathbf{x}^i_t\,|\,\mathbf{x}^j_{t-1})$ and $\mathbb{P}(\mathbf{y}_t\,|\,\mathbf{x}^j_t)$ respectively, are known and nothing more is needed.

For a continuous state space, the grid based method can be used if the density can be "decomposed" into N 'cells' or discrete points. The representation of the densities remain the same as (2.4.49), (2.4.50) or (2.4.51) except for the fact that those are now discrete approximations and the weights are computed differently. Let $\bar{\mathbf{x}}^i_t$ denote the centre of the ith cell at time t. The weights that are calculated at the centre of each grid are written as

$$w^i_{t\,|\,t-1} = \sum_{j=1}^{N} w^j_{t-1\,|\,t-1}\,\mathbb{P}(\bar{\mathbf{x}}^i_t\,|\,\bar{\mathbf{x}}^j_{t-1}),$$

$$w^i_{t\,|\,t} = \frac{w^i_{t\,|\,t-1}\,\mathbb{P}(\mathbf{y}_t\,|\,\bar{\mathbf{x}}^i_t)}{\sum_{j=1}^{N} w^j_{t\,|\,t-1}\,\mathbb{P}(\mathbf{y}_t\,|\,\bar{\mathbf{x}}^j_t)}.$$

It should be noted that the grid must be sufficiently dense to yield a good approximation. However, if the dimension of the state process is large, the computational cost increases dramatically since this method also suffers from the curse of dimensionality. Hence these methods work well only in low dimensions.

Kitagawa (1987) provides a starting point with approximation to integrals using finite sums on a grid. The *Hidden Markov Model* (HMM) filter is an application of grid based filters (Rabiner 1989) and has been used extensively in speech recognition. Other grid based methods include that of Pole & West (1990), which use the Gauss-Hermite quadrature scheme in their grid based methodology. Grid based algorithms are very important for our thesis, since our new method is also a grid based method. Although our method does not use any of the existing grid-based methodologies, the basic idea explained here in this section also applies to our method. We will be showing in Chapter 4 how we use our grid-based method in a way different to those already mentioned here.

So for we have been discussing sequential estimation techniques (not including classical time series models), in which estimation of parameters is done by sequential updates to the posterior. In the next section we discuss some methods which base their inference on the whole set of data available to them. For a set of observations, say, $\mathbf{Y}_{1:T}$, these methods provide inference for posterior distribution the like of $\mathbb{P}(\mathbf{x}_{1:T}, \theta\,|\,\mathbf{y}_{1:T})$. Such methods are known a *Off-line methods.*

2.5 Off-line methods

Several methods for off-line inference exist which are frequently used to estimate the parameters in dynamic state space models. It should be kept in mind that in most of the cases, the parameters in a dynamic state space models are unknown and need to be estimated. Data from pilot study or otherwise are used to provide an estimate $\hat{\theta}$ of θ after which the previously discussed methods are used to infer about the posterior of the state process: $\mathbf{X}_t \,|\, \mathbf{Y}_{1:t}, \hat{\theta}$. Three popular off-line estimation techniques are discussed in this section, namely Expectation Maximization (Dempster et al. 1977), Variational Bayes (Jaakkola & Jordan 2000) and Expectation Propagation (Minka 2001a). Note that there exists other methods which can be used for off-line estimation, for example Laplace approximation (Tierney et al. 1989), MCMC (Gilks et al. 1996*b*) among others but we do not discuss them here. We will refer to the methods discussed here, later in Section 2.6.

2.5.1 Expectation Maximization

The EM algorithm is a two step iterative computation technique for calculation of the MLE of unknowns when the data can be viewed (or actually is) as incomplete data (Dempster et al. 1977). If there is a latent model with an observable variable $\mathbf{Y}$ and unobservable latent variable $\mathbf{X}$, along with unknown set of parameters Θ, the EM algorithm computes at the $(k+1)^{th}$ iteration:

- **Expectation step**:
  Compute $\mathbb{E}[logP(\mathbf{X}, \mathbf{Y} \,|\, \Theta) \,|\, \mathbf{Y}, \Theta^k](= Q(\Theta^k))$ where $P(\cdot)$ is the *complete* log-likelihood;
- **Maximization step**:
  Choose Θ^{k+1} such that $Q(\Theta^{k+1}) \geq Q(\Theta^k)$.

In a time series context, the estimation of the unknown parameters in a model as defined in Equations (2.3.6) and (2.3.7) is done using the EM algorithm, while the smoothing part involving the conditional expectation of the log-likelihood is done using variants of Kalman filter (Shumway & Stoffer 1982).

2.5.2 Variational Bayes

Variational bounding is a deterministic method which is more general than that of Laplace method. We start by assuming that we are seeking an approximation to the posterior $f(\mathbf{x})$. Then we assume some arbitrary function $q(\mathbf{x})$ and define:

$$I = \int_{\mathbf{x}} q(\mathbf{x}) \frac{f(\mathbf{x})}{q(\mathbf{x})} \,\mathrm{d}\mathbf{x}. \tag{2.5.1}$$

For convex functions $p(\cdot)$ and $g(\cdot)$, Jensen's equality states (in case of logarithm) that,

$$\log \int_x p(x)\, g(x) \mathrm{d}x \geq \int_x p(x)\, \log(g(x))\, \mathrm{d}x, \tag{2.5.2}$$

$$\text{if} \int_x p(x) \mathrm{d}x = 1. \tag{2.5.3}$$

Combining Equations (2.5.1) and (2.5.2), we get the following lower bound on I:

$$I \geq \exp\left(\int_{\mathbf{x}} q(\mathbf{x})\, \log \frac{f(\mathbf{x})}{q(\mathbf{x})}\, \mathrm{d}\mathbf{x} \right). \tag{2.5.4}$$

This of course, requires that $f(\mathbf{x})$ is positive, a condition that is satisfied for most integrals in Bayesian inference. For non-negative functions a similar idea to that explained in (Tierney et al. 1989) is used. We are free to choose $q(\mathbf{x})$ to provide the tightest bound, which corresponds to maximising the right hand side of Equation (2.5.4). Note that this is equivalent to minimising the reverse Kullback-Leibler divergence:

$$D(q||f) = \int_{\mathbf{x}} q(\mathbf{x})\, \log \frac{f(\mathbf{x})}{q(\mathbf{x})}\, \mathrm{d}\mathbf{x},$$

subject to the constraint in Equation (2.5.2). Even though this method has been used for several problems, it is computationally intense and more often than not, infeasible.

A less accurate but simpler way to obtain a variational bound is to bound the integrand and then integrate the bound:

$$f(\mathbf{x}) \geq g(\mathbf{x}), \tag{2.5.5}$$

$$I \geq \int_{\mathbf{x}} g(\mathbf{x})\, \mathrm{d}\mathbf{x}. \tag{2.5.6}$$

This method, popularly known as the Variational Bayes method, has been used by several authors in a variety of problems, see Titterington (2011) for its use in the field mixture modelling. For a good reference to use of variational Bayes method in filtering problems see Smídl & Quinn (2008) and Vermaak et al. (2003) for application in a tracking problem.

2.5.3 Expectation Propagation

Expectation Propagation (Minka 2001a) is a recursive approximation technique that approximates the posterior by a known family of distribution and then minimizes the Kullback-Leibler divergence between the two. To start with, note that the posterior distribution $\mathbb{P}(\mathbf{x}_{1:t} \,|\, \mathbf{y}_{1:t})$ can be written as a product of several factors like:

$$\mathbb{P}(\mathbf{x}_{1:t} \,|\, \mathbf{y}_{1:t}) = \frac{1}{Z_q} \mathbb{P}(\mathbf{x}_{1:t}) \prod_{i=1}^{t} t_i(\mathbf{x}_i, \mathbf{y}_i). \tag{2.5.7}$$

There are different ways to choose the terms $t_i(\cdot,\cdot)$. As a rule of thumb fewer terms are better, since that means fewer approximations; a popular choice being the marginal likelihood:

$$t_i(\mathbf{x}_i, \mathbf{y}_i) = \mathbb{P}(\mathbf{y}_i \,|\, \mathbf{x}_i).$$

The next step is to identify a parametric approximating distribution $q(\mathbf{x}_{1:t} \,|\, \mathbf{y}_{1:t})$ and it is necessary that it is chosen from an exponential family so that only a fixed number of expectations (the sufficient statistics) can be propagated. Such an approximation can be written as:

$$q(\mathbf{x}_{1:t} \,|\, \mathbf{y}_{1:t}) = \frac{1}{Z_q} \mathbb{P}(\mathbf{x}_{1:t}) \prod_{i=1}^{t} \tilde{t}_i(\mathbf{x}_i, \mathbf{y}_i).$$

The basic idea is to approximate $t_i(\cdot,\cdot)$ by $\tilde{t}_i(\cdot,\cdot)$ and then construct the posterior using these $\tilde{t}_i(\cdot,\cdot)$. The factors, $t_i(\cdot,\cdot)$, making up $q(\cdot)$ are updated one at a time, updating step for the i^{th} factor being as follows. Define,

$$p_i(\mathbf{x}_{1:t} \,|\, \mathbf{y}_{1:t}) = \frac{1}{Z_q} \mathbb{P}(\mathbf{x}_{1:t}) t_i(\mathbf{x}_i, \mathbf{y}_i) \prod_{j \neq i} \tilde{t}_j(\mathbf{x}_j, \mathbf{y}_j),$$

which is the same as $q(\cdot)$ but with the factor $\tilde{t}_i(\mathbf{x}_i, \mathbf{y}_i)$ replaced by $t_i(\mathbf{x}_i, \mathbf{y}_i)$. A **new** approximation, $t^*(\cdot \,|\, \cdot)$, results in the following:

$$q^*(\mathbf{x}_{1:t} \,|\, \mathbf{y}_{1:t}) = \frac{1}{Z_q} \mathbb{P}(\mathbf{x}_{1:t}) t_i^*(\mathbf{x}_i, \mathbf{y}_i) \prod_{j \neq i} \tilde{t}_j(\mathbf{x}_j, \mathbf{y}_j),$$

such that $q^*(\cdot \,|\, \cdot)$ is as close to $p_i(\cdot \,|\, \cdot)$ as possible. This is usually done by minimizing the Kullback-Leibler divergence between $q^*(\cdot)$ and $p_i(\cdot)$ $(KL(p_i,\, q^*))$, or some other measure. This step is repeated for all other i until convergence is reached.

Not that EP also allows one to approximate the smoothing distribution (for $T > t$):

$$\mathbb{P}(\mathbf{x}_t \,|\, \mathbf{y}_{1:T}) = \frac{1}{Z_q} t_i(\mathbf{x}_i, \mathbf{y}_i) \int \mathbb{P}(\mathbf{x}_{1:t}) \prod_{j \neq i}^{t} t_j(\mathbf{x}_j, \mathbf{y}_j) \, \mathrm{d}\mathbf{x}_{-i}, \tag{2.5.8}$$

where $\mathbf{x}_{-i} = (x_0, \cdots, x_{i-1}, x_{i+1}, \cdots, x_t)$. Recursive versions of this method, based on KL and used in sequential inference can be found in Ypma & Heskes (2005).

2.6 Parameter Estimation in State-Space Models

In all the procedures discussed in Sections 2.1 or 2.4, the parameters associated with the model are assumed to be known. Box and Jenkins' ARMA model methodology do provide maximum likelihood estimates for the variance and the AR or MA parameter, but they do not provide any sequential inference as is already known. Offline methods like those detailed in Section 2.5 are also used to estimate the parameters, again not sequentially.

Most of the dynamic state space based methods assume that the parameters are known or estimated off-line, and inference is done on the state process.

In a majority of the real life problems however, parameters in a state space model are typically not known, except maybe for situations for which previous work already exists. These parameters could be static or time dependent. Parameters which evolve with time need to be estimated at each time point, quite similar to the state estimation that all the filters perform. We concentrate on the estimation of static parameters in this section. Maximum likelihood techniques like Newton-Raphson or scoring algorithms have been used to estimate parameters (Gupta & Mehra 1974). However, such methods have their own problems. Firstly, these methods require the computation of second order derivatives and the inverse of the Hessian at each iteration, which becomes a demanding task if the dimension of the parameter space is high. Methods based on maximizing penalized likelihood criterion (Fahrmeir & Künstler 1998) also suffer from the same drawback. Alternative approaches like the EM algorithm have been proposed (Shumway & Stoffer 1982) which do not suffer from such problems. However, since the second order derivatives are not computed in the EM algorithm one cannot estimate the standard errors. Another known problem is that the EM algorithm may converge very slowly at the latter stages of the iteration (Shumway & Stoffer 1982). Optimization based algorithms tend to be quite slow in several applications which causes development of faster algorithms that are discussed in the next few paragraphs.

2.6.1 Kalman filter based methods:

Dual estimation and joint estimation methods for estimation of parameters have been developed which are based on extensions of Kalman filters (Wan et al. 2000). However, these techniques require the static parameters to be by an artificial evolution equation. Supposing the actual model is given by:

$$\mathbf{y}_t = f(\mathbf{x}_t, \mathbf{u}_t, \mathbf{v}_t, \theta), \tag{2.6.1}$$

$$\mathbf{x}_t = g(\mathbf{x}_{t-1}, \mathbf{w}_t, \theta). \tag{2.6.2}$$

We represent the set of parameters by a vector θ. In these methods, there is one extra equation denoting evolution of the parameters, approximating the above model. Thus the new model is given as:

$$\mathbf{y}_t = f(\mathbf{x}_t, \mathbf{u}_t, \mathbf{v}_t, \theta_t), \tag{2.6.3}$$

$$\mathbf{x}_t = g(\mathbf{x}_{t-1}, \mathbf{w}_t, \theta_t), \tag{2.6.4}$$

$$\theta_t = h(\theta_{t-1}) + \epsilon_t, \tag{2.6.5}$$

for some error term ϵ_t, usually Gaussian, and for some function $h(\cdot)$ which is generally linear, thus causing a random walk evolution on the parameters.

Thus we can see that some parameters, which are actually not dependent on time, are now represented as time dependent. Pretending that some static parameters are time dependent implies a "loss of information", resulting in posteriors that are more diffuse than what the actual posterior might be (Liu & West 2000). Methods based on extensions of Kalman filters however require this representation, and the way these filters function is discussed below.

The *dual estimation* technique used by extensions of Kalman filter first breaks up the main model as separate state space models for the state and the parameters. So assuming that the state space model is represented by Equations (2.6.1) and (2.6.2), one can re-write the state space model to represent the evolution equation as:

$$\mathbf{y}_t = f((g(\mathbf{x}_{t-1}, \mathbf{w}_t, \theta)), \mathbf{u}_t, \mathbf{v}_t, \theta), \tag{2.6.6}$$

$$\theta_t = \theta_{t-1} + \epsilon_t \quad \epsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbf{B}_t). \tag{2.6.7}$$

It can be seen that this formulation allows approximate KF techniques to be applied to the equations (2.6.6) and (2.6.7). At each time step, the current estimates of the parameters are used in the state estimation and while the state estimates are used in the parameter model (Equations (2.6.6) and (2.6.7)). Note that, the state equation involving the parameters is linear and Gaussian, while the measurement equation is potentially nonlinear.

An alternative to dual estimation is the *joint estimation* technique, in which the state and the parameter vector are concatenated into a single "joint" vector: $\mathbf{z}_t = \{\mathbf{x}_t^T, \theta_t^T\}^T$. The estimation of $\mathbf{z}_t$ can be done in the usual recursive way by writing the state-space model as:

$$\mathbf{y}_t = [1 \quad \mathbf{0}]\mathbf{z}_t + \mathbf{v}_t,$$
$$\mathbf{z}_t = \begin{bmatrix} g(\mathbf{x}_{t-1}, \theta_{t-1}) \\ \mathbf{I} \cdot \theta_{t-1} \end{bmatrix} + \begin{bmatrix} \mathbf{w}_t \\ \mathbf{u}_t \end{bmatrix}$$

The joint estimation provides *maximum a-posteriori probability estimate* (MAP) by maximizing the joint density of the states and the parameters given the data. A MAP estimate is the mode of the posterior joint distributions of parameter and state given the data. Any extension of the Kalman filter like EKF or UKF can be used in these methods.

2.6.2 Sequential Monte Carlo based methods:

Sequential Monte Carlo (SMC) methods combine several different procedures to estimate the unknown parameters. There exist very sophisticated off-line as well as online techniques, each of which have their own problems associated with them. A good review of parameter estimation using SMC based methods is provided in Kantas et al. (2009). There are two broad types: maximum likelihood based methods, in which a prior for θ is not required and Bayesian approaches, which explicitly requires a prior to be set for θ.

SMC methods can approximate the marginal likelihood as shown in Equation 2.4.42. So they can be used to for maximum likelihood parameter estimation, where the maximizing value of θ for the marginal likelihood of the observations is given by

$$\hat{\theta} = \arg\max_{\theta} l_t(\theta),$$

where

$$l_t(\theta) = \log \mathbb{P}_{\theta}(\mathbf{y}_{1:t}).$$

A gradient based approach or EM algorithm can be used to estimate θ. Both these methods are locally optimal. They are thus sensitive to initialization and the algorithm may stop at a local maximum. An alternative approach to offline ML estimation has been proposed in Ionides et al. (2011) and is known as *Iterated filtering.* Like the Kalman filter based methods, this procedure also requires artificial evolution for the parameters, while using SMC for state estimation. However this is an off-line method, and is not suitable for on-line estimation of the parameters. This method has certain advantages over gradient based or EM based methods, for example it does not require the computation of analytical derivatives, which may not be available and numerical derivatives, which could be unstable Ionides et al. (2006).

Both the gradient descent and EM algorithm can also be used in on-line parameter estimation. In the steepest descent method, upon arrival of a new observation, the parameter is updated in the direction of the descent of the negative predictive density of this new observation. This has been discussed in detail in Poyiadjis et al. (2005) and Doucet & Tadic (2003). As an alternative to this method, an on-line version of the EM algorithm has also been used in Andrieu et al. (2005).

Bayesian estimation of parameters can similarly be both off-line and on-line. A suitable prior density for θ is chosen and attempts are made to approximate the joint density $\mathbb{P}(\mathbf{x}_{1:t}, \theta \,|\, \mathbf{y}_{1:t})$. Ordinary MCMC sampling algorithms are quite difficult to design for these models. With the increase in time, the volume of data becomes very large for the usage of MCMC to be computationally feasible. Particle MCMC (PMCMC) are a new class of MCMC techniques which rely on the SMC algorithm. Andrieu et al. (2010) discuss the algorithm in detail, along with the theoretical properties.

The idea of joint on-line estimation in a Bayesian setting with state variables and static parameters using SMC methods seems easily applicable at first sight. In fact, this idea has been implemented by Berzuini et al. (1997), for a specific problem. Using standard SMC methods by writing $\mathbf{Z}_t = (\mathbf{X}_t, \theta_t)$, with an initial prior density $\mathbb{P}(\theta_0)\mathbb{P}(\mathbf{x}_0)$ and transition density $\mathbb{P}(\mathbf{x}_t \,|\, \mathbf{x}_{t-1})\, \delta_{\theta_{t-1}}(\theta_t)$ (i.e. $\theta_t = \theta_{t-1}$), results in degeneracy for the parameter particles. We are left with only a few number of particles as t increases (Storvik 2002). As a result, artificial dynamics need to be introduced for on-line estimation based on SMC methods. It is obvious that these on-line techniques also suffer from the same problems associated with forced artificial evolution as in the case of algorithms based on Kalman filter. Several different types of algorithm exist and have been tried for sequential parameter estimation. A simple SMC can be applied on the parameters, after introducing a small artificial noise (maybe decreasing with t) to the parameter evolution equation (Kitagawa 1998). A kernel density estimation based method has also been proposed in Liu & West (2000). One possible way to avoid the artificial evolution of parameters is to use MCMC within a SMC step. The basic idea is to use an MCMC kernel with invariant density $\mathbb{P}(\mathbf{x}_{1:t}, \theta \,|\, \mathbf{y}_{1:t})$. Andrieu et al. (1999) used this concept for on-line parameter estimation. For a detailed overview of online parameter estimation, the interested reader should look in to Kantas et al. (2009). In a very recent paper Poyiadjis et al. (2011) has developed methods for static parameter estimation which are based on approximations to the score and observed information matrix. However, as noted by the authors, convergence of estimates often takes several thousand time steps and hence is very slow.

2.7 Conclusion

In this chapter we have discussed in great detail the whole subject of estimation and inference in time series models. Box & Jenkins ARMA models has been discussed along with dynamic state space models. We have also explored the topic of state and parameter estimation, both on-line and off-line. We hope that the advantages and disadvantages of existing methods in time series models should be clear from this chapter. In the next chapter, we discuss in detail the statistical techniques that are used by these models as well as those that will be used by us. Some definitions, for example Markov property, have been used in this chapter which will be talked about in greater detain in Chapter 3.

Chapter 3

Statistical Methodology

This chapter contains a brief overview of the Bayesian framework for parametric model fitting and other statistical methods used in this thesis. It also contains an outline of integrated nested Laplace approximation (INLA), which is a recently developed technique by Rue et al. (2009). It is a fast computational method providing a functional approximation to the posterior for a certain class of models that includes some state space models. A short introduction to MCMC has also been provided in this chapter since we will be using SMC filters in latter chapters to compare their performance with our methods. At the very end of the chapter, a proof for the derivation of Kalman filter is shown using Bayesian properties. This has been provided to bring some clarity about the workings of not only extensions of Kalman filter, but also filters which are based on conjugate distributions.

3.1 Statistical Inference

In this section we provide a philosophical view of the two main approaches prevalent in the subject area of statistics. It is well known that statistics as a tool is practiced over a wide range of diverse applications. While data exploration techniques can be informal, the concept of analysis of data follows some highly conventional rules or theories. The fundamental things that one tries to do in statistical inference is to learn from the data in the presence of uncertainty. Uncertainty in the real world is quantified using probability which is generally interpreted in two different ways:

1. the *frequentist* (or *classical*) approach, in which the probability is interpreted to mean (usually hypothetically) a relative frequency, and,

2. the *Bayesian* (or *inverse probability*) in which the notion of personal decision making is extended to cover assessment of any uncertain event or proposition.

3.1.1 Interpretation of Probability

The frequentist approach to inference in statistics is related to the frequentist approach to probability. The highly popular and much used concepts of hypothesis testing and confidence intervals are two of the most important applications of frequentist inference. Frequentist approaches are distinguished by two important features:

1. the information provided by the data is the sole quantitative form of relevant probabilistic information, and,
2. analysis and measure of uncertainty is done by long-run frequency under hypothetical repetitions (*long run relative frequency*).

In the frequentist approach, the unknown parameters of a model take unique values, i.e. they are fixed but unknown, therefore it is not meaningful to make probabilistic statements about them. The interested reader should consult the excellent book written by Casella & Berger (2001) for further details on frequentist approaches.

In contrast, Bayesian statistics offers the input of personalistic beliefs in contexts of uncertainty, with the aim of incorporating an individual's act of decision making in the particular event. Bayesian approaches base all the calculations on the laws of probability. Subjective probability, and not relative frequency, is used as the measure of uncertainty for any unknown in the model. This is essentially borrowed from a decision theoretic approach. A simple example illustrates the difference between the two approaches. If a person states that the probability that a head will turn up for a particular coin is 0.5, then it may mean that in many tosses of the coin, about half the time, heads will show up (a version of the law of large numbers). However it can also mean that if a person puts a bet on head - if a head comes he wins, or else he loses the same amount, then the gamble is fair to that person. The first version is frequentist while the second is (subjective) Bayesian. One can think of situations where only a Bayesian explanation for probability is feasible, for example election of a particular candidate, since here the idea of long-run frequency has little meaning given the situation. Bayesian analysis thus is firmly based on personal evaluation of uncertainty, or more generally speaking, some *prior* knowledge about the nature of the system, even before data have been collected. Since one person's prior beliefs do not necessarily agree with another's, the Bayesian view of probability is a subjective interpretation of probability. Some scientists and philosophers have argued that there may be a third kind of interpretation of probability which represents a shared belief, rather than a person's subjective uncertainty. It is called *objective* or *non-subjective* probability. The basis of Bayesian inference is rooted on a single principle: the quantification of uncertainty in the light of data through the use of *Bayes' theorem*. Fortunately the mathematics underlying the analysis remains the same whether one uses a subjective or an objective definition for prior belief.

From the above discussion, we can see that Bayesian inference treats the unknown parameters as random variables, which makes it natural to make probability statements about them, allowing interpretation to be more intuitive. An excellent book on the foundations of Bayesian analysis are by de Finetti (1972, 1974, 1975) and Bernardo & Smith (1996). For further reading refer to the books by Box & Tiao (2011), Ghosh et al. (2006), Gelman et al. (2004) and others.

3.1.2 Prior Distribution

In statistical inference, the experimental data are collected using a sampling procedure, and are denoted by some random variable $\mathbf{Y} = \mathbf{y}$, where $\mathbf{y}$ denote the realisations of the random variable. Usually we use a parametric model for assigning the probability law of $\mathbf{Y}$, and the quantity of interest lies in the vector $\theta = (\theta_1, \cdots, \theta_p)$ of parameters of the model. The parameters need to be determined, for example by a point estimate or an interval estimate in a classical sense, so that a complete formulation of the model is possible.

The Bayesian inference on θ acknowledges that solution to its estimation lies in the computation of its conditional distribution given the data. The Bayesian approach allows one to to explicitly introduce all the information we have about the parameters, be it from experts' opinions, from previous studies, from the theory and from the phenomenon itself, in the inferential process. A probability density function, $\mathbb{P}(\theta)$, is assigned to θ and is known as the *prior distribution* or simply the *prior*. The prior is also defined to be an honest expression of beliefs about θ, with no mathematical restrictions on its form, except that it is a probability distribution. Notable exception are improper priors (DeGroot 2005), which do not integrate to a finite value, but ensures that the posterior is a probability distribution. In other words, improper priors need not be probability distributions themselves. The prior density is further parameterized by some *hyperparameters* ψ, which also need to be specified so that the prior information is adequately summarized.

The process of selecting and specifying prior distributions $\mathbb{P}(\theta)$ and its hyperparameters is not always straightforward and is discussed in latter chapters.

3.1.3 Likelihood Function

A concept of fundamental importance is the *likelihood* function. The data can be modelled by $\mathbb{P}(\mathbf{Y}|\theta)$, a function of the data given some fixed parameter vector θ. Here $\mathbb{P}(\mathbf{Y}|\theta)$ is the probability density of all the data and thus properties such as integration to unity hold as $\int \mathbb{P}(\mathbf{Y}|\theta)\, d\mathbf{Y} = 1$. On the other hand, informally for fixed $\mathbf{Y}$, $\mathbb{P}(\mathbf{Y}|\theta)$ is regarded as a function of θ and is called the likelihood function:

$$\mathcal{L}(\theta|\mathbf{Y}) = \mathbb{P}(\mathbf{Y}|\theta).$$

It measures how likely it is to observe the data given that the parameters take the value θ. It is quite common to suppress $\mathbf{Y}$ and write the likelihood as $\mathcal{L}(\theta)$. The likelihood function is not unique in that for any $c(\mathbf{Y}) > 0$ that may depend on $\mathbf{Y}$ but not on θ, $c(\mathbf{Y})\mathcal{L}(\theta)$ is also the same likelihood function (Barnard et al. 1962).

Interpreted this way, a likelihood is a conditional probability density statement which does not have the same properties as a density, for example it does not have to integrate to 1. Note that the likelihood function is used both in frequentist and Bayesian inference [(Fisher 1922) and (Bernardo & Smith 1996)].

3.1.4 Posterior Distribution

In Bayesian inference, the conditional density $\mathbb{P}(\theta|\mathbf{Y})$ of θ given the data, $\mathbf{Y} = \mathbf{y}$, is called the posterior density, a quantification of our uncertainty about θ in the light of data. The transition from the prior $\mathbb{P}(\theta)$ to the posterior $\mathbb{P}(\theta|\mathbf{Y})$ is what we learn from the data. The posterior is thus obtained from two sources of information: the prior and the likelihood. The frequentist approach, in contrast, utilizes only the likelihood information. *Bayes' theorem* is the basic rule which takes a Bayesian from prior to posterior:

$$\begin{aligned}\mathbb{P}(\theta|\mathbf{Y}) &= \frac{\mathbb{P}(\theta)\mathbb{P}(\mathbf{Y}|\theta)}{\mathbb{P}(\mathbf{Y})}, \\ &\propto \mathbb{P}(\theta)\mathbb{P}(\mathbf{Y}|\theta),\end{aligned} \tag{3.1.1}$$

where

$$\mathbb{P}(\mathbf{Y}) = \begin{cases} \int \mathbb{P}(\theta)\mathbb{P}(\mathbf{Y}|\theta)\, d\theta & \text{in the continuous case,} \\ \sum_\theta \mathbb{P}(\theta)\mathbb{P}(\mathbf{Y}|\theta) & \text{in the discrete case.} \end{cases}$$

More succinctly, this can be written as:

$$\text{Posterior} \propto \text{Prior} \times \text{Likelihood}$$

In the Bayesian framework, all inference about the parameter of interest is derived from a single source: the posterior distribution. So the Bayesian can simply report the posterior, or could report summary descriptive statistics associated with the posterior distribution. For example, for a real valued parameter θ, one could report the posterior mean

$$\mathbb{E}(\theta|\mathbf{Y}) = \int \theta\, \mathbb{P}(\theta|\mathbf{Y})\, d\theta,$$

and the posterior variance

$$Var(\theta|\mathbf{Y}) = \int (\theta - \mathbb{E}(\theta|\mathbf{Y}))^2\, \mathbb{P}(\theta|\mathbf{Y})\, d\theta.$$

Furthermore, one could answer more structured problems like interval estimation or testing of hypothesis. In the case of estimation of θ, one would report the above summary statistics, or other moments, quantiles, highest posterior regions and credible intervals, see for example, (Ghosh et al. 2006) for more details. In contrast to a frequentist analysis, where subsequent to determining the likelihood, there is a choice of many inference rules (for example, a choice between unbiased estimators with uniformly equal variance), in the Bayesian analysis the derivation of posterior inference is uniquely determined once the prior and likelihood have been formulated, and according to needs, decide what to do with the posterior.

3.1.5 Prior Elicitation

Incorporating the prior information about the parameters in order to make inference about them can be a contentious issue as well as a potential strength of the Bayesian approach. However, full elicitation of subjective probability is quite rare. Most priors used in priors practice are partly non-subjective. However, prior expert opinion is particularly interesting, since there exists the possibility that the opinion it offers is more useful than not (Kadane & Wolfson 1998). Incorporation of prior opinion have a considerable effect on problems of statistical inference with small sample size or high or infinite dimensional parameter space (Gelman et al. 2004).

In the usage of prior, the use of objective Bayesian analysis has no conflict with the subjectivist approach. A noninformative or objective prior (Bernardo & Smith 1996, Berger 2006) is said to represent "prior ignorance" or in other words "let the data speak for themselves". Although it is much argued that there is no "objective" prior that represents ignorance, there is often a need for a prior to have minimal effect, relative to the data, on the final inference. A widely used example is *Jeffreys' prior* (Jeffreys 1946):

$$\mathbb{P}(\theta) \propto [\mathbb{J}(\theta)]^{1/2},$$

where $\mathbb{J}(\theta)$ is the *Fisher Information* (Fisher 1925) for θ:

$$\mathbb{J}(\theta) = -\mathbb{E}\left[\frac{\partial^2 \, log\mathbb{P}(\mathbf{Y}|\theta)}{\partial\theta^2} \middle| \theta \right].$$

In examples with low-dimensional θ, objective Bayesian analysis has some similarities with likelihood based methods, in that the estimate obtained or hypothesis accepted tends to be very close to what a frequentist would have done. The only difference is that objective Bayesian inference produces a posterior distribution and a data-based evaluation of the error associated with inference.

Objective priors are often improper. As have been discussed earlier, improper prior density is non-negative for all θ but

$$\int_\theta \mathbb{P}(\theta)\, d\theta = \infty.$$

Such an improper prior can be used in the Bayes' formula to compute the posterior provided that

$$\int_\theta \mathbb{P}(\theta)\mathbb{P}(\mathbf{Y}|\theta)\, d\theta < \infty.$$

Then the posterior density becomes a proper probability density function.

It is generally mathematically and computationally convenient to work with priors that yield posteriors in closed form. One way to ensure tractability of the posterior is to use *conjugate* priors (Gelman et al. 2004). These are distributions that belong to the same class of distributions as the posterior, and are used widely in practice. For example, assume that the data $\mathbf{Y} = \mathbf{y}$ are normally distributed with unknown mean θ and some known variance. So if the prior beliefs of the parameter θ can be expressed in terms of a normal distribution, then the resulting posterior density will be a product of two normal densities, which is also a normal distribution. Minimizing and maximizing posterior quantities then becomes an easy task. A crucial drawback of the conjugate class of priors is that it is usually "too small" to provide robustness. Further, tails of these prior densities are similar to those of the likelihood function, and hence prior moments greatly influence posterior inferences. Thus, even when the data are in conflict with the specified prior information the conjugate priors used can have very pronounced effect (which may be undesirable if data are to be trusted more). It must be added here that mixtures of conjugate priors, for example the Student's t prior, which is a scale mixture of normals having flat tails, on the other hand can provide robust inferences. Alternatively another approach that makes the conjugate distribution for robust is the use of hierarchical modeling. Interested readers should look at Robert (2001) for better insight into this area.

The hyperparameters also reflect the strength of the prior information and its relation to the data. Where genuine, substantial prior information exists, the choice of hyperparameters can be based on expert opinions of the properties of the parameters θ such as moments or quantiles. For a non-informative prior which is fairly flat over the support of θ, an useful example is a normal distribution with a large variance, or a uniform distribution. Thus the choice of hyperparameters decide the nature of the prior. Garthwaite et al. (2005) give more details on various methods for specifying prior information.

Bayes' theorem synthesizes information from both the prior and the likelihood in formulating the posterior density. As the amount of information increases, the model will increasingly rely on the data rather than the prior knowledge to obtain a posterior. A sensitivity analysis can be carried out to check whether the conclusions drawn from the posterior remain stable under changes in the assumptions made about the posterior.

3.1.6 Predictive Distribution

In sequential inference, it is of great interest to make predictions about future observations as well as making inferences about the parameters of the model. The belief about the next observation is based on the already observed data of size n (in time series data it could be of time t) $\mathbf{y}_1, \cdots, \mathbf{y}_n$and the posterior of the unknown parameters.

The predictive distribution for a new observation $\tilde{\mathbf{y}}_{n+1}$, given observed data up till n is given by:

$$\mathbb{P}(\tilde{\mathbf{y}}_{n+1}|\mathbf{y}_1, \cdots, \mathbf{y}_n) = \int \mathbb{P}(\mathbf{y}_{n+1}|\theta)\, \mathbb{P}(\theta|\mathbf{y}_1, \cdots, \mathbf{y}_n)\, \mathrm{d}\theta,$$

where $\mathbb{P}(\theta|\mathbf{y}_1, \cdots, \mathbf{y}_n)$ is the posterior density. The first term inside the integral in the above equation holds since the distribution of the future observations, $\tilde{\mathbf{y}}_{n+1}$, given θ does not depend on any of the past data, $\mathbf{y}_1, \cdots, \mathbf{y}_n$.

3.2 Numerical Problems in Bayesian Inference

A matter of substantial importance in Bayesian statistics is the computation of integrals. This is sometimes referred to as the *integration problem* (Evans & Swartz 1995). Exact analytical solutions do not exist in most of the cases and hence the integral needs to be approximated. A typical example would be the computation of the denominator $\mathbb{P}(\mathbf{y})$, in Equation (3.1.1), also known as the *normalizing constant* in a posterior distribution. The computation of $\mathbb{P}(\mathbf{y})$ involves solving the following integral:

$$\mathbb{P}(\mathbf{y}) = \int \mathbb{P}(\mathbf{y}|\theta)\, \mathbb{P}(\theta)\, \mathrm{d}\theta, \tag{3.2.1}$$

which is intractable in most applications.

The computation of the normalizing constant, or of any other term characteristic of the posterior or predictive density is generally very difficult. The difficulty increases with higher dimensions, but even in lower dimensions, solving the integrals become a problem. Approximations need to be made, either analytically or numerically to solve these integrals. In sequential inference, at each time point, the calculation of the predictive density and the one-step ahead prior density requires solving for integrals. So, if the known posterior of a state process is assumed to be $\mathbb{P}(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}, \theta)$ at time $t-1$, then the one-step ahead prior density is calculated as:

$$\mathbb{P}(\mathbf{x}_t|\mathbf{y}_{1:t-1}, \theta) = \int \mathbb{P}(\mathbf{x}_t|\mathbf{x}_{t-1}, \theta)\, \mathbb{P}(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}, \theta)\, \mathrm{d}\mathbf{x}_{t-1}.$$

Other integrals of high importance are the posterior and predictive distributions unconditional on θ:

$$\mathbb{P}(\mathbf{y}_t|\mathbf{y}_{1:t-1}) = \int \mathbb{P}(\mathbf{y}_t|\mathbf{y}_{1:t-1},\theta)\mathbb{P}(\theta|\mathbf{y}_{1:t-1})\,\mathrm{d}\theta \text{ and}$$
$$\mathbb{P}(\mathbf{x}_t|\mathbf{y}_{1:t}) = \int \mathbb{P}(\mathbf{x}_t|\mathbf{y}_{1:t},\theta)\mathbb{P}(\theta|\mathbf{y}_{1:t})\,\mathrm{d}\theta.$$

Any characteristic of the posterior density of the posterior of θ say, can also be difficult to compute, for example:

$$\mathbb{E}(g(\theta)) = \int g(\theta)\mathbb{P}(\theta|\mathbf{Y}_{1:t})\,\mathrm{d}\theta.$$

Normally these integrals do not have exact solutions, except for linear models with additive Gaussian error, parameters with Gaussian priors and models where conjugacy property can be applied (Doucet et al. 2001). Hence good and fast approximating schemes are required for their solutions.

Several methods have been proposed over the years to solve the this problem. While some are functional approximations, others are sampling based and each of these have their own advantages and disadvantages. Broadly speaking there are three distinct methodologies used in the literature to solve the above mentioned integrals. The three categories are listed below:

- Functional approximation methods provide a density approximation to the posterior, where a Gaussian approximation is used to solve the integral of the type defined by Equation (3.2.1). The Gaussian approximation technique will be discussed in detail in Section 3.4.2. This technique will be necessary to understand another approximating method known as integrated nested Laplace approximation which will be introduced in latter sections of this chapter. One of the most popular functional approximations is the *Laplace's approximation* which is an asymptotic method (Tierney & Kadane 1986) using a second order Taylor's expansion under certain assumptions and again uses the Gaussian approximation method. Other popular methods are *Expectation-Propagation* (Minka 2001b), *Variational Bayes* (Jaakkola & Jordan 2000) and others. For a good reference for this topic, the interested reader should see Minka (2001a).

- Sampling based methods like *Importance sampling* (Geweke 1989) and *Monte Carlo Markov Chain* (MCMC) methods are considered to be a very rich class of methodology and are widely used to integrals in high dimensional problems. These methods have very good convergence results associated with them. (Gilks et al. 1996*a*) provides a wonderful discussion on this subject.

- Other useful methods that have been tried successfully are the *Quadrature methods* and in certain special cases they are very efficient (Monahan 2011). However these methods suffer from the curse of dimensionality. We will not be discussing these methods any more since we will not be using them in our thesis. Nevertheless it should be mentioned that our method follows the quadrature methods, in certain aspects, quite closely.

In the next section, we discuss the first two methods in detail. First we explore MCMC and its relevant theory. Further on, we will be discussing INLA which will help us develop our methodology in a clear manner.

3.3 Markov Chain Monte Carlo

As has been explained in Section 3.2, any feature of posterior distribution may be needed to calculate and these include posterior moments or highest posterior density regions among a few. All these integration based summaries can be expressed in terms of posterior expectations of functions of θ, by drawing n independent samples $\{\theta_i, i = 1, \cdots, n\}$ from $\mathbf{P}(\theta \mid \mathbf{Y})$ and approximate:

$$\begin{aligned}\mathbb{E}[f(\theta) \mid \mathbf{Y}] &= \frac{\int f(\theta)\mathbf{P}(\theta)\mathbf{P}(\mathbf{Y} \mid \theta)\, \mathrm{d}\theta}{\int \mathbf{P}(\theta)\mathbf{P}(\mathbf{Y} \mid \theta)\, \mathrm{d}\theta}, && (3.3.1)\\ &\approx \frac{1}{n}\sum_{i=1}^{n} f(\theta_i). && (3.3.2)\end{aligned}$$

As have been mentioned earlier, posterior distributions are not known in closed in many situations, and hence direct sampling from posterior distributions is not possible and independent samples are not available. Markov chains provide a solution to this problem by generating dependent samples from the posterior of interest. These samples can be used for Monte Carlo integration purposes; this is then Markov chain Monte Carlo. Markov chain Monte Carlo (MCMC) methods provide algorithms for the drawing of (correlated) samples from highly complex or multi-dimensional distributions from which direct sampling is impossible. One such algorithm is the Metropolis-Hasting algorithm. A very simple summary of this method is provided in the following after a brief introduction to Markov chains and detailed balance theory. For a more comprehensive introduction see Gilks et al. (1996*a*).

3.3.1 Markov Chains and Detailed Balance

A key concept behind solving the integration problem as defined in Section 3.2 is that of *stochastic process* and *Markov chain*. In this section some definitions are provided:

Definition A **stochastic process** is a family of random variables $\{\mathbf{X}_t; t \in T, T \subseteq \mathbb{N}\}$, where T is called the index state. The possible values (i.e. **states**) of $\mathbf{X}_t$ form a set $\mathbb{S}$ known as the **state space**.

For simplicity, it will be assumed in the rest of the chapter that T is a countable set, hence known as *discrete time stochastic process.* This can be extended to stochastic processes with a continuous index.

Definition A stochastic process $\{\mathbf{X}_t; t \in T, T \subseteq \mathbb{N}\}$ is said to have the **Markov property**, and is known as a **Markov chain** with a countable state space $\mathbb{S}$ if the present state $\mathbf{X}_t = j$ is independent of all previous states except for $\mathbf{X}_{t-1} = i$. More formally:

$$\mathbb{P}(\mathbf{X}_t = j|\mathbf{X}_{t-1} = i, \ldots, \mathbf{X}_0 = i_0) = \mathbb{P}(\mathbf{X}_t = j|\mathbf{X}_{t-1} = i),$$

for all $t \geq 1$ and for all $\mathbf{X}_t, \ldots, \mathbf{X}_0 \in \mathbb{S}$, i.e. for Markov chains, the *past* and *present* are independent given the *present.*

Typical chain behaviour is specified by initial state X_0 and the *transition probabilities* $P_{ij} = \mathbb{P}(\mathbf{X}_t = j|\mathbf{X}_{t-1} = i)$.

A key concept is *detailed balance*, which is connected to reversibility of the Markov process. Reversibility just means that the joint distribution of the process at a series of times is unchanged if the direction of time is reversed. Clearly this only makes sense for a stationary process as for any other Markov process the convergence towards equilibrium reveals the direction of time.
For a discrete-time discrete-state-space Markov process reversibility entails

$$\mathbb{P}(\mathbf{X}_t = i, \mathbf{X}_{t+1} = j) = \mathbb{P}(\mathbf{X}_{t+1} = i, \mathbf{X}_t = j) = \mathbb{P}(\mathbf{X}_t = j, \mathbf{X}_{t+1} = i).$$

So if π_i is the stationary distribution,

$$\pi_i P_{ij} = \pi_j P_{ji},$$

for any transition matrix P_{ij}. This equation is known as detailed balance Grimmett & Stirzaker (2001).

If we know there is a unique stationary distribution, and we can show detailed balance for our distribution π, it can be shown that it is the unique stationary distribution. If we also know that the Markov process converges to its stationary distribution, then this is a valid MCMC sampling scheme.

The definitions presented will help to present the idea of Metropolis-Hastings algorithm in the next section and GMRF in a latter section. More about stochastic processes and Markov chains can be found in the classic book by Grimmett & Stirzaker (2001).

3.3.2 The Metropolis-Hastings Algorithm

A general way to construct a Markov chain with a given stationary distribution π was given by Metropolis et al. (1953) which was given added flexibility by Hastings (1970). These MCMC schemes start with a transition kernel $q(x; y)$ of a Markov process on the state space. Given a current state $\mathbf{Y}_t$ this is used to generate a candidate next state $\mathbf{Y}^*$. Then either the transition is accepted and $\mathbf{Y}_{t+1} = \mathbf{Y}^*$ or it is not when $\mathbf{Y}_{t+1} = \mathbf{Y}_t$. The probability that the move is accepted is $\alpha(\mathbf{Y}_t, \mathbf{Y}^*)$ where

$$\alpha(x; y) = min \left\{ 1, \frac{\pi(y)q(y,x)}{\pi(x)q(x,y)} \right\}.$$

It is a simple exercise to show that this satisfies detailed balance. For the stationary distribution to be also the limiting distribution the chain needs to be aperiodic: note that it will be aperiodic if there is a positive probability of rejecting a move (Roberts & Rosenthal 2004).

For generating samples from the stationary distribution, the algorithm is allowed to run from an initial starting point for a *sufficient* amount of time until convergence is achieved, and then on samples are collected. In other words, the algorithm is allowed to iterate until the samples increasingly appear to be dependent samples from a stationary distribution, and then it is allowed to run till we get the number of required samples.

It is important to choose the proposal distributions carefully, for while any choice of $q(\cdot|\cdot)$ will yield the correct stationary distribution of the Markov chain (Gilks et al. 1996*a*), the mixing and convergence properties of the algorithm are dependent on the proposal distribution chosen. If the proposed moves between states are small, the probability of acceptance of a candidate value will be relatively high and consequently the chain will take a long time to explore the target distribution. Conversely, if the proposed moves between states are too large, the acceptance rate will be quite low and the chain will fail to move, greatly reducing the number of effective samples available for inference. In both these extreme cases the algorithm can be said to “mix slowly, indicating that the chain moves slowly around the support of the target distribution. For the many choices on implementing a Metropolis-Hastings MCMC sampler see Chib & Greenberg (1995).

Another issue is to decide when convergence is achieved. The random walk can remain for several iterations in some region of the posterior space influenced by the initial starting value of the chain leading to the miss-conception that convergence is reached. A general solution to this problem, as proposed by Gilks et al. (1996*a*), is to run multiple parallel chains, each with difference starting values from the other. Although it increases the computational burden, it gives us a check of when the chain reaches convergence. Mixing and convergence issues, coupled with highly correlated variables causes the speed at which the algorithm explores the target distribution to be quite slow, sometimes running into weeks or even months. Alternative algorithms, for example functional approximation methods try to address the problem involving slow computational speed encountered by MCMC methods. In the following section, we provide a brief overview of one such method, namely, INLA.

3.4 Integrated Nested Laplace Approximation (INLA)

A recently proposed method, *Integrated Nested Laplace Approximation* (INLA) (Rue et al. 2009) is a fast Bayesian functional approximation technique which performs approximate inference in a subclass of structured additive regression models, namely *latent Gaussian models*. A latent variable model with an underlying latent variable that follows a *Gaussian Markov random field* (GMRF) is known as *latent Gaussian model* (Rue & Held 2005). Some preliminaries are defined in the following subsections which will help in the explanation of the INLA technique.

3.4.1 Gaussian Markov Random Field (GMRF)

A **Gaussian Markov Random Field** is a finite dimensional random vector following a multivariate Gaussian distribution, with additional *Markov* properties. Before a formal mathematical definition of GMRF is provided, a *neighborhood system* is introduced. Let $L = \{i; i = 1, 2, ..., M\}$ be a set of sites (or nodes). A collection of subsets of L,

$$\eta = \{\eta_i : i \in L, \eta_i \subset L\},$$

is a neighbourhood system of L if and only if the neighbourhood of node i, η_i, is such that

- $i \notin \eta_i$,
- if $j \in \eta_i$, then $i \in \eta_j$, $\forall i, j \in L$

A random vector $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N) \in \Re^N$ is called a GMRF with respect to a neighborhood system η with mean μ and precision matrix $Q > 0$ *iff* its density has the form

- $\mathbb{P}_{\mathbf{X}}(\mathbf{x}) = (2\pi)^{-N/2} \left|Q\right|^{\frac{1}{2}} exp \left(-\frac{1}{2} (\mathbf{x} - \mu)^T Q (\mathbf{x} - \mu) \right)$
- $Q_{ij} \neq 0 \Longleftrightarrow i \in \eta_j$ for all $i \neq j$

When Q is sparse, computational handling of the GMRF becomes easier by utilizing relevant matrix decomposition algorithms.

The covariance matrix is not used in GMRF since it is generally dense and hence imposes computational burden. The simple relationship between conditional independence and the zero structure of the precision matrix is not evident in the covariance matrix. While the precision matrix give information on conditional dependence structure, the covariance matrix provides information on marginal dependence. Hence in dealing with GMRFs, parameterization is based on the precision matrix. In most cases only $\mathcal{O}(N)$ of the N^2 entries of Q are non-zero, so Q is sparse. This means one can use numerical methods for sparse matrices, which are much quicker than dense matrix calculations (Rue & Held 2005).

The formulation of INLA includes the idea of *Gaussian approximation* of a probability density. which comes from *Laplace approximation* to integrals. This is discussed in detail in the next section, which will enable the reader to understand the implementation issues.

3.4.2 Gaussian Approximation

A Gaussian approximation of a density is obtained by matching the modal configuration and curvature at the mode. This approximation has been applied earlier by Lindley (1980), and can be viewed as an application of the Laplace method for integrals Tierney & Kadane (1986). Let $\mathcal{L}$ denote the log of the likelihood and $\mathbb{P}(\theta)$ be the prior density of some random variable θ. Supposing one is interested in the integral

$$\int e^{\mathcal{L}(\theta)}\mathbb{P}(\theta)\,d\theta = \int e^{-nL}\,d\theta,$$

where $L = -log\mathbb{P} - \mathcal{L}/n$. Let $\hat{\theta}$ be the posterior mode. Then using Taylor's approximation, one can write

$$\begin{aligned}\int e^{-nL}\,d\theta &= \int e^{-n\{L(\hat{\theta})+\frac{1}{2}(\theta-\hat{\theta})'j(\hat{\theta})(\theta-\hat{\theta})+\cdots\}}\,d\theta,\\ &\approx e^{-nL(\hat{\theta})}\int e^{\frac{n}{2}(\theta-\hat{\theta})'j(\hat{\theta})(\theta-\hat{\theta})}\,d\theta,\\ &\approx e^{-nL(\hat{\theta})}\sqrt{2\mathbb{P}}n^{-\frac{1}{2}}\left|j(\hat{\theta})\right|^{-\frac{1}{2}},\end{aligned}$$

where $j(\hat{\theta}) = -1/L''(\hat{\theta})$. The Laplace approximation shows that the posterior, which is of the form $\mathbb{P}(\theta|\mathbf{Y}) = \mathbb{P}(\mathbf{Y}|\theta)\mathbb{P}(\theta)$, can be approximated as

$$\mathbb{P}(\theta|\mathbf{Y}) \propto e^{-\frac{1}{2}(\theta-\hat{\theta})'j(\hat{\theta})(\theta-\hat{\theta})}.$$

Then we have

$$\mathbb{P}(\theta|\mathbf{Y}) \sim \mathcal{N}(\hat{\theta}, j^{-1}(\hat{\theta})).$$

The Gaussian approximation will produce reasonable results as long as the posterior is unimodal or at least dominated by a single mode, and it is generally assumed that the sample size is large enough for this to be the case. This is because of the fact that this technique is the result of asymptotic approximations like the Laplace approximation (Tierney et al. 1989).

3.4.3 INLA

INLA is a recently introduced methodology which approximates the posterior distribution for a certain class of latent models, namely Gaussian latent models (Rue et al. 2009). Interest lies in the inference on the parameters $\mathbf{X}$ and θ. It is assumed that the hidden state process, $\mathbf{X}\,(\mathbf{X} \in \mathcal{X})$ is a GMRF, with an initial distribution $\mathbb{P}(\mathbf{X}_0)$. The response variable, $\mathbf{Y}\,(\mathbf{Y} \in \mathcal{Y})$ is assumed to be conditionally independent given the latent process $\mathbf{X}$ and set of parameters θ. There is no restriction on the distributional properties of the parameters θ. $\mathbf{Y}$ and $\mathbf{X}$ are related by the following model:

$$\mathbb{P}(\mathbf{Y}|\mathbf{X},\theta) = \prod_i \mathbb{P}(\mathbf{Y}_i|\mathbf{X}_i,\theta).$$

The posterior distribution of $\mathbf{X}$, conditional on the parameters and the data, is thus prior times the likelihood:

$$\mathbb{P}(\mathbf{X}|\mathbf{Y},\theta) \propto \mathbb{P}(\theta)\mathbb{P}(\mathbf{X}|\theta)\prod_i \mathbb{P}(\mathbf{Y}_i|\mathbf{X}_i,\theta). \tag{3.4.1}$$

However, the parameters need to be integrated out to get the posterior conditional on only the observations

$$\mathbb{P}(\mathbf{X}|\mathbf{Y}) = \int \mathbb{P}(\mathbf{X}|\mathbf{Y},\theta)\mathbb{P}(\theta|\mathbf{Y})\,d\theta.$$

In many problems, the real goal of inference is to calculate the marginal of the latent field given the data

$$\begin{aligned}\mathbb{P}(\mathbf{X}_i|\mathbf{Y}) &= \int \mathbb{P}(\mathbf{X}_i|\mathbf{Y},\theta)\mathbb{P}(\theta|\mathbf{Y})\,d\theta,\\ &\approx \int \tilde{\mathbb{P}}(\mathbf{X}_i|\mathbf{Y},\theta)\tilde{\mathbb{P}}(\theta|\mathbf{Y})\,d\theta,\\ &\approx \sum_j \tilde{\mathbb{P}}(\mathbf{X}_i|\mathbf{Y},\theta_j)\tilde{\mathbb{P}}(\theta_j|\mathbf{Y})\,\Delta\theta, \qquad (3.4.2)\end{aligned}$$

where $\tilde{\mathbb{P}}(\cdot|\cdot)$ is an approximated (conditional) density of its arguments. Approximations to $\mathbb{P}(\mathbf{X}_i|\mathbf{Y})$ are computed by first approximating $\mathbb{P}(\theta|\mathbf{Y})$ and $\mathbb{P}(\mathbf{X}|\mathbf{Y},\theta)$, and then using numerical integration techniques (for example finite sum) to integrate out θ. Many, but not all latent Gaussian models satisfy a basic property, which makes the above approximations both fast and accurate. Firstly the latent field is a GMRF. As has been noted earlier, this assumption results in working with a sparse precision matrix, making calculations faster, as compared to dense general matrices. Added to this is the assumption that the number of parameters is small, say ≤ 10. These two assumptions helps in very fast and accurate inference on the unknown parameters.

INLA is considered quite promising since it calculates the two terms $\tilde{\mathbb{P}}(\mathbf{X}|\mathbf{Y},\theta_j)$ and $\tilde{\mathbb{P}}(\theta|\mathbf{Y})$ in (3.4.2) very efficiently and quickly. The posterior of θ can be written as

$$\mathbb{P}(\theta|\mathbf{Y}) \propto \frac{\mathbb{P}(\theta)\mathbb{P}(\mathbf{X}|\theta)\mathbb{P}(\mathbf{Y}|\mathbf{X},\theta)}{\mathbb{P}(\mathbf{X}|\mathbf{Y},\theta)}. \tag{3.4.3}$$

In INLA, the posterior, as explained in Equation (3.4.3) is approximated by the following

$$\tilde{\mathbb{P}}(\theta|\mathbf{Y}) \propto \left.\frac{\mathbb{P}(\theta)\mathbb{P}(\mathbf{X}|\theta)\mathbb{P}(\mathbf{Y}|\mathbf{X},\theta)}{\tilde{\mathbb{P}}_{\mathbf{G}}(\mathbf{X}|\mathbf{Y},\theta)}\right|_{\mathbf{X}=\mathbf{X}^*(\theta)} \tag{3.4.4}$$

where $\tilde{\mathbb{P}}_{\mathbf{G}}(\cdot)$ is the *Gaussian approximation* to the full conditional density of $\mathbf{X}$, $\mathbb{P}_{\mathbf{X}|\mathbf{Y},\theta}(\cdot)$, and $\mathbf{X}^*$ is the mode of the full conditional density, for some given value of θ. The mode is computed iteratively using some Newton's algorithm type optimization technique. The Gaussian approximation is intuitively appealing for latent Gaussian models. For most real data sets, the conditional posterior of $\mathbf{X}$ is typically well behaved, and looks 'almost' Gaussian. This is due to the impact of the Gaussian prior of the latent field.

It is now required to explore $\tilde{\mathbb{P}}(\theta|\mathbf{Y})$ sufficiently well to select good evaluation points. It is assumed for simplicity that $\theta \in \Re^p$, which can always be obtained by reparameterisation. Subsequently the mode and negative-Hessian at the mode of $\tilde{\mathbb{P}}(\theta|\mathbf{y})$ are obtained by optimizing it with respect to θ. This again can be done by some Newton-based optimisation method, which builds up on an approximation to the second derivative of log-$\tilde{\mathbb{P}}(\theta|\mathbf{y})$ using finite difference methods. Let θ^* be the mode and H the negative Hessian at the mode. Thus $\Sigma = H^{-1}$ is the covariance matrix of θ. The approximate posterior of θ is then explored at a sufficient number of points on a grid, using the calculated mode and Hessian. To aid the exploration one uses standardized variables $\mathbf{z}$ instead of θ. The standardized variable has mean 0. Letting $\Sigma = \mathbf{V}\Delta\mathbf{V}^{-1}$ to be the eigen-decomposition, θ is defined via $\mathbf{z}$ as:

$$\theta(\mathbf{z}) = \theta^* + \mathbf{V}\Delta^{1/2}\mathbf{z}.$$

log-$\tilde{\mathbb{P}}(\theta|\mathbf{y})$ is explored using the $\mathbf{z}$-parameterization, by starting from 0 and moving along the axis in the direction of the eigenvectors. Let us assume $\theta = (\theta_1, \theta_2)$, which implies $\mathbf{z} = (z_1, z_2)$. Thus one starts at $\mathbf{z} = 0$ (which is the mode), and proceed in the positive direction of z_1 with step-length δ_z, as long as

$$\text{log-}\tilde{\mathbb{P}}(\theta(0)|\mathbf{y}) - \text{log-}\tilde{\mathbb{P}}(\theta(\mathbf{z})|\mathbf{y}) < \delta.,$$

for some $\delta > 0$. The value of δ controls the stopping rule for the exploration of the support. Subsequently the direction is switched and the same rule is applied. This is done for both the axes. Finally all possible combinations of grid points are considered and any given point is accepted or rejected based on the stopping rule defined above. Note that, for the interested user, posterior marginals for θ_i can be obtained from the joint posterior distribution by using the already calculated grid-points and numerical integration.

The next step is to provide accurate approximations to the posterior marginal of $\mathbf{X}_i$'s, conditioned on selected values of θ by using the Laplace approximation a second time, or a simplified form of Laplace approximation. These methods are not explained here; interested reader may want to look into Rue et al. (2009).

3.5 Kalman Filter

The Kalman filter plays a big role in this thesis for inference on the state parameters. In all our examples, state filtering density and the predictive density is calculated either using the Kalman filter or extensions of it. We present a detailed proof of the Kalman filter in this section, the rationale behind which is to understand other forms of filters, namely extensions of Kalman filter like UKF or EKF and also conjugate filters. This proof also helps the reader to understand filters of the type of non-Gaussian Kalman filter, introduced by Chen & Singpurwalla (1994).

A linear state space model with additive Gaussian errors is given as:

$$\left.\begin{aligned} \mathbf{y}_t &= \mathbf{F}_t^T\mathbf{x}_t + \mathbf{v}_t \quad \mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{V}_t),\\ \mathbf{x}_t &= \mathbf{G}_t\mathbf{x}_{t-1} + \mathbf{w}_t \quad \mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{W}_t), \end{aligned}\right\} \tag{3.5.1}$$

where for each time t:

1. $\mathbf{y}_t$ is a $(n \times 1)$ observation vector;
2. $\mathbf{x}_t$ is a $(p \times 1)$ state vector;
3. $\mathbf{F}_t$ is a known $(p \times n)$ matrix;
4. $\mathbf{G}_t$ is a known $(p \times p)$ matrix;
5. $\mathbf{V}_t$ is a $(n \times n)$ covariance matrix;
6. $\mathbf{W}_t$ is a $(p \times p)$ covariance matrix.

At $t = 0$, the initial information is that the mean and variance of $\mathbf{X}_0|\mathbf{Y}_0$ are assumed to be known to be $\bar{\mathbf{x}}_0$ and $\mathbf{P}_0$ respectively. The error sequences $\mathbf{v}_t$ and $\mathbf{w}_t$ are independent, and also mutually independent.

Deterministic inference on a system as defined in Equation (3.5.1) is provided with an optimal solution in the class of linear filters by the Kalman filter. This is done by minimizing the conditional mean square error $\mathbb{E}\big[\big(\mathbf{x}_t - \hat{\mathbf{x}}_t)^T[(\mathbf{x}_t - \hat{\mathbf{x}}_t)|\mathbf{y}_{1:t}\big]$, where $\hat{\mathbf{x}}_t)$ is the estimate of $\mathbf{x}_t|\mathbf{y}_{1:t}$. An alternate but more intuitive solution is provided by West & Harrison (1997) using a Bayesian approach.

Theorem 3.5.1 (Kalman Filter) *For time t, the one step-ahead forecast and the posterior distribution of the state process defined by the DLM is as follows:*

I Posterior at $t-1$:
For some mean $\bar{\mathbf{x}}_{t-1|t-1}$ and covariance matrix $\mathbf{P}_{t-1|t-1}$,

$$\mathbf{X}_{t-1}|\mathbf{Y}_{1:t-1} \sim \mathcal{N}(\bar{\mathbf{x}}_{t-1|t-1}, \mathbf{P}_{t-1|t-1}).$$

II Prior at t:

$$\mathbf{X}_{t}|\mathbf{Y}_{1:t-1} \sim \mathcal{N}(\bar{\mathbf{x}}_{t|t-1}, \mathbf{P}_{t|t-1}),$$

where

$$\bar{\mathbf{x}}_{t|t-1} = \mathbf{G}_t\bar{\mathbf{x}}_{t-1|t-1} \quad \textit{and} \quad \mathbf{P}_{t|t-1} = \mathbf{G}_t\mathbf{P}_{t-1|t-1}\mathbf{G}_t^T + \mathbf{W}_t.$$

III 1-step ahead forecast:

$$\mathbf{Y}_{t}|\mathbf{Y}_{1:t-1} \sim \mathcal{N}(\bar{\mathbf{y}}_{t|t-1}, \mathbf{Q}_{t|t-1}),$$

where,

$$\bar{\mathbf{y}}_{t|t-1} = \mathbf{F}_t^T\bar{\mathbf{x}}_{t|t-1} \quad \textit{and} \quad \mathbf{Q}_{t|t-1} = \mathbf{F}_t^T\mathbf{P}_{t|t-1}\mathbf{F}_t + \mathbf{V}_t.$$

IV Posterior at t:

$$\mathbf{X}_{t}|\mathbf{Y}_{1:t} \sim \mathcal{N}(\bar{\mathbf{x}}_{t|t}, \mathbf{P}_{t|t}),$$

with,

$$\bar{\mathbf{x}}_{t|t} = \bar{\mathbf{x}}_{t|t-1} + \mathbf{K}_t\mathbf{e}_t \quad \textit{and} \quad \mathbf{P}_{t|t} = \mathbf{P}_{t|t-1} - \mathbf{K}_t\mathbf{Q}_{t|t-1}\mathbf{K}_t^T,$$

where

$$\mathbf{e}_t = \mathbf{Y}_t - \bar{\mathbf{y}}_{t|t-1} \quad \textit{and} \quad \mathbf{K}_t = \mathbf{P}_{t|t-1}\mathbf{F}_t\mathbf{Q}_{t|t-1}^{-1}.$$

Proof Proof of the above is by induction and relies on the theory of multivariate normal distribution (Anderson 2003) and application of Bayes' theorem, which makes the proof quite general. It is assumed that *I* holds.

The *State evolution* equation in (3.5.1) along with the assumption of *I*, leads to the prior of *II*. Obviously $\mathbf{X}_t|\mathbf{Y}_{1:t-1}$ is normally distributed since it is a linear function of $\mathbf{X}_{t-1}$ and $\mathbf{w}_t$, which in turn are independently normally distributed. Hence from the state equation, it is seen that the mean and covariance matrix of $\mathbf{X}_t|\mathbf{Y}_{1:t-1}$ is $\mathbf{G}_t\bar{\mathbf{x}}_{t-1|t-1}$ and $\mathbf{G}_t\mathbf{P}_{t-1|t-1}\mathbf{G}_t^T + \mathbf{W}_t$ respectively.

The *Observation evaluation* equation provides the following conditional density

$$\mathbb{P}(\mathbf{Y}_t|\mathbf{X}_t) = \mathbb{P}(\mathbf{Y}_t|\mathbf{X}_t, \mathbf{Y}_{1:t-1}).$$

Hence using the above equation along with the prior distribution defined in *II*, one finds that $\mathbf{Y}_t$ and $\mathbf{X}_t$ are jointly normally distributed conditional on $\mathbf{Y}_{1:t-1}$, with variance term of $\mathbf{Y}_t$ computed below:

$$\begin{aligned} Var(\mathbf{Y}_t|\mathbf{Y}_{1:t-1}) &= Var(\mathbf{F}_t^T\mathbf{X}_t + \mathbf{v}_t|\mathbf{Y}_{1:t-1}), \\ &= \mathbf{F}_t^T Var(\mathbf{X}_t|\mathbf{Y}_{1:t-1})\mathbf{F}_t + \mathbf{0}^T + Var(\mathbf{v}_t) = \mathbf{F}_t^T\mathbf{P}_{t|t-1}\mathbf{F}_t + \mathbf{V}_t. \end{aligned}$$

The observation equation directly gives us the mean as $\mathbb{E}(\mathbf{Y}_t|\mathbf{Y}_{1:t-1}) = \mathbf{F}_t^T\bar{\mathbf{x}}_{t|t-1}$. Additionally, the covariance matrix can also be computed as,

$$\begin{aligned} Cov(\mathbf{Y}_t, \mathbf{X}_t|\mathbf{Y}_{1:t-1}) &= Cov(\mathbf{F}_t^T\mathbf{X}_t + \mathbf{v}_t, \mathbf{X}_t|\mathbf{Y}_{1:t-1}), \\ &= \mathbf{F}_t^T Var(\mathbf{X}_t, \mathbf{X}_t|\mathbf{Y}_{1:t-1}) + \mathbf{0}^T = \mathbf{F}_t^T\mathbf{P}_{t|t-1}. \end{aligned}$$

Thus the joint conditional distribution of $\mathbf{Y}_t$ and $\mathbf{X}_t$ is given as:

$$\begin{matrix} \mathbf{Y}_t \\ \mathbf{X}_t \end{matrix} \Big| \mathbf{Y}_{1:t-1} \sim \mathcal{N}\left[\begin{pmatrix} \bar{\mathbf{y}}_{t|t-1} \\ \bar{\mathbf{x}}_{t|t-1} \end{pmatrix}, \begin{pmatrix} \mathbf{Q}_{t|t-1} & \mathbf{F}_t^T\mathbf{P}_{t|t-1} \\ \mathbf{P}_{t|t-1}^T\mathbf{F}_t & \mathbf{P}_{t|t-1} \end{pmatrix} \right]$$

The above distribution can be used along with Multivariate Gaussian theory results to obtain the posterior in IV. However, a more general proof in a Bayesian set-up will be provided here.

Using Bayes' theorem, it is easy to see that,

$$\mathbb{P}(\mathbf{X}_t|\mathbf{Y}_{1:t}) \propto \mathbb{P}(\mathbf{X}_t|\mathbf{Y}_{1:t-1})\mathbb{P}(\mathbf{Y}_t|\mathbf{X}_t). \tag{3.5.2}$$

It has been shown that the first term in Equation (3.5.2) is given as:

$$\mathbb{P}(\mathbf{X}_t|\mathbf{Y}_{1:t-1}) \propto \exp\left\{ -\frac{1}{2}(\mathbf{X}_t - \bar{\mathbf{x}}_{t|t-1})^T\mathbf{P}_{t|t-1}^{-1}(\mathbf{X}_t - \bar{\mathbf{x}}_{t|t-1}) \right\},$$

and the second term as

$$\mathbb{P}(\mathbf{Y}_t|\mathbf{X}_t) \propto \exp\left\{ -\frac{1}{2}(\mathbf{Y}_t - \mathbf{F}_t^T\mathbf{x}_t)^T\mathbf{V}_t^{-1}(\mathbf{Y}_t - \mathbf{F}_t^T\mathbf{x}_t) \right\},$$

Taking the natural logarithms and multiplying by -2, Equation (3.5.2) can be written as,

$$-2\ln\{\mathbb{P}(\mathbf{X}_t|\mathbf{Y}_{1:t})\} = (\mathbf{X}_t - \bar{\mathbf{x}}_{t|t-1})^T\mathbf{P}_{t|t-1}^{-1}(\mathbf{X}_t - \bar{\mathbf{x}}_{t|t-1}) + (\mathbf{Y}_t - \mathbf{F}_t^T\mathbf{X}_t)^T\mathbf{V}_t^{-1}(\mathbf{Y}_t - \mathbf{F}_t^T\mathbf{X}_t) + \text{constant}, \tag{3.5.3}$$

where the constant term in Equation (3.5.3) does not any term involving $\mathbf{X}_t$. The quadratic function in the above equation can be rearranged with a new constant as

$$-2\ln\{\mathbb{P}(\mathbf{X}_t|\mathbf{Y}_{1:t})\} = \mathbf{X}_t^T(\mathbf{P}_{t|t-1}^{-1} + \mathbf{F}_t\mathbf{V}_t^{-1}\mathbf{F}_t^T)\mathbf{X}_t - 2\mathbf{X}_t^T(\mathbf{P}_{t|t-1}^{-1}\bar{\mathbf{x}}_{t|t-1} + \mathbf{F}_t\mathbf{Y}_t\mathbf{V}_t^{-1}) + constant. \tag{3.5.4}$$

By the definition of $\mathbf{P}_{t|t}$ in the statement of the theorem, it can be verified that

$$(\mathbf{P}_{t|t-1}^{-1} + \mathbf{F}_t\mathbf{V}_t^{-1}\mathbf{F}_t^T)\mathbf{P}_{t|t} = \mathbf{I},$$

so that

$$\mathbf{P}_{t|t-1}^{-1} + \mathbf{F}_t\mathbf{V}_t^{-1}\mathbf{F}_t^T = \mathbf{P}_{t|t}^{-1}.$$

Using this form of $\mathbf{P}_{t|t}^{-1}$ and the form of $\bar{\mathbf{x}}_{t|t}$ in the theorem statement, it follows that

$$\mathbf{P}_{t|t}^{-1}\bar{\mathbf{x}}_{t|t} = \mathbf{P}_{t|t-1}^{-1}\bar{\mathbf{x}}_{t|t-1} + \mathbf{F}_t\mathbf{Y}_t\mathbf{V}_t^{-1}.$$

Thus Equation (3.5.4) can now be written as

$$\begin{aligned}-2\ln\{\mathbb{P}(\mathbf{X}_t|\mathbf{Y}_{1:t})\} &= \mathbf{X}_t^T\mathbf{P}_{t|t}^{-1}\mathbf{X}_t - 2\mathbf{X}_t^T\mathbf{P}_{t|t}^{-1}\bar{\mathbf{x}}_{t|t} + constant \\ &= (\mathbf{X}_t - \bar{\mathbf{x}}_{t|t})^T\mathbf{P}_{t|t}^{-1}(\mathbf{X}_t - \bar{\mathbf{x}}_{t|t}) + constant\end{aligned}$$

Thus result IV follows.

3.6 Conclusion

The statistical methodologies discussed in this section will be used for different purposes in the following chapters where we will be presenting our main work. Firstly, INLA forms the basis of our work where we are interested in inferring about static parameters of a dynamic model. This idea will be clear in the following chapter when we present our work. Secondly, methods like MCMC will be necessary for comparing the performance of our work with sampling based ones. This will be more evident in the results chapter of this thesis.

Chapter 4

Sequential Parameter Estimation in State Space Models

4.1 Introduction

State space models, sequential Bayesian inference and related methods have been discussed in Chapter 2. A broad overview of statistical methods for state space models has been provided in Chapter 3. In this chapter, the first novel work of the thesis is presented. It discusses a new method for sequential parameter estimation developed as part of the research work.

For the purpose of this research work, the following has been assumed:

- The data are modelled by a discrete state space model with unknown parameters.
- The hidden variable $\mathbf{X}_t$ is assumed to be Gaussian or nearly Gaussian with a Markov structure.
- It is also assumed that the number of unknown parameters is relatively small, say less than 10.

Even though introduced before, we define the discrete time state space model here once again for the sake of clarity:

$$\mathbf{y}_t = f(\mathbf{x}_t, \mathbf{u}_t, \mathbf{v}_t, \theta_1), \tag{4.1.1}$$

$$\mathbf{x}_t = g(\mathbf{x}_{t-1}, \mathbf{w}_t, \theta_2), \tag{4.1.2}$$

where $\theta \in (\theta_1, \theta_2)$ is the vector of unknown parameters. As has been discussed in Chapter 1, it will be assumed from here on that θ is independent of t. Statistical interest related to state space models lies on making inference about the unknown state variable and parameters. Estimation of the parameters is not at all a straightforward process, since almost always their distributions are non-Gaussian and closed form expressions cannot be derived (Andrieu et al. 2005). A simple example is presented here to establish this fact.

Example: *Linear Gaussian Model*

$$y_t = x_t + v_t \quad v_t \sim \mathcal{N}(0, \sigma_u^2), \tag{4.1.3}$$

$$x_t = \phi x_{t-1} + w_t \quad w_t \sim \mathcal{N}(0, \sigma_w^2), \tag{4.1.4}$$

where $\theta = (\phi, \sigma_u^2, \sigma_w^2)$ and $\Theta = (-1, 1) \times \mathbb{R}^+ \times \mathbb{R}^+$. θ denotes the static parameter vector.

It is easy to see from the following that the posterior of θ does not belong to a known family of distributions, even for the linear Gaussian model. To clarify that, let us start with specifying the joint posterior for the unknown state process and the unknown parameters in the following:

$$\mathbb{P}(\theta, \mathbf{x}_{1:T} \,|\, \mathbf{y}_{1:T}) \propto \mathbb{P}(\mathbf{y}_{1:T} \,|\, \mathbf{x}_{1:T}, \theta)\, \mathbb{P}(\mathbf{x}_{1:T} \,|\, \theta)\, \mathbb{P}(\phi)\, \mathbb{P}(\sigma_u^2)\, \mathbb{P}(\sigma_w^2), \tag{4.1.5}$$

In Equation (4.1.5), the first term on the right is the likelihood term and depends on x_t for all $t = 1, \cdots, T$ and σ_u^2. Similarly the second term is a GMRF which depends on ϕ and σ_w^2. One can assume a truncated normal and inverse Gamma priors for the AR parameter and the variance parameters respectively. Simple mathematics show that the posterior distribution $\mathbb{P}(\theta \,|\, \mathbf{y}_{1:T})$, which involves integrating out the state variable from Equation (4.1.5), is not in closed form. Thus even for a linear Gaussian model, the construction of the posterior for the static parameters is non-trivial.

This chapter focuses on Bayesian inference of static parameters and not the state process. A lot of research activity in this area has either been done using non-sequential inference methods like Iterated filtering (Ionides et al. 2011) or through extensions of the Kalman filter (Wan et al. 2000) and sampling based methods (Poyiadjis et al. 2011). As is generally common in sequential inference techniques, there is always a trade off between accuracy and speed of the algorithm. Some methods like *particle filters* are generally quite accurate, but slow and also not suitable for high dimensional state vectors. On the other hand, algorithms based on extensions of Kalman filter are quite good in terms of speed, but have to compromise on accuracy (Arulampalam et al. 2002). Already discussed techniques like dual estimation and joint estimation are used for parameter estimation, but there remains a significant need for accurate but computationally cheaper methods.

The proposed method discretizes the posterior of the parameters on a grid of points. Grid based methods have also been proposed earlier by Bucy & Senne (1971) and Pole & West (1990). But heavy computation associated with such filters restricted any widespread use of these filters. Recently, with increased efficiency of computer hardware, there has been much interest in grid based methods through the development of the INLA method that combines speed and accuracy. It is INLA that motivates the work of the new method. We propose a new method for sequential Bayesian estimation technique called *Sequential INLA* used for inferencing about discrete state space parameters. This chapter is organized as follows, Section 4.2 explains how to initiate the algorithm including construction of starting grid. Section 4.3 evaluates the sequential algorithm for updating the posterior of θ. This section also details the actual implementation algorithm. It is necessary to understand that it is not possible to have a fixed grid in sequential updating of the posterior. Checking and updating of the grid at each time point is discussed in Section 4.5.

4.2 Initial Steps with INLA

Since SINLA is a grid based method, the algorithm starts with computation of a discrete grid over the support of $\mathbb{P}(\theta \mid \mathbf{Y}_{1:t})$. Let this grid be called the *starting grid.* If the model makes the following assumptions:

- The state process $\mathbf{X}_t$ is assumed to be Gaussian with a Markov structure and,
- It is assumed that the number of unknown parameters is ≤ 10,

then it conforms to the requirements of INLA method. Hence INLA can now be applied to the first few observations $\mathbf{Y}_1, \cdots, \mathbf{Y}_n$, where n is a small number for constructing the starting grid. However, it is worth mentioning here that in a wide variety of applications, the state process may not have a Gaussian distribution. INLA can still be used, although the approximation won't be very good. Moreover any other suitable approximation method like Expectation-Propagation can also be used, as long as it can provide us with a starting grid for the posterior of $\theta \mid \mathbf{Y}_{1:t}$.

Since a small set of data will be used, INLA will be very fast in creating the grid of points over the support of $\mathbb{P}(\theta \mid \mathbf{Y}_{1:n})$. Supposing the conditional posterior of θ until time $t-1$ has been computed on a grid of points using INLA. A new data point $\mathbf{Y}_t$ is recorded and INLA is re-applied using all the data $\mathbf{Y}_1, \cdots, \mathbf{Y}_t$. As soon as INLA reaches a threshold in terms of computation time, one should start the sequential algorithm described in the following section. As is the case with INLA, the mode and the Hessian at the mode is utilized for the construction the grid for the posterior of the parameter space at deterministically chosen points. There is no specific rule to set the value of n. It depends on the dimension, as well as the type (non-linear and/or non-Gaussian) of the problem. A pilot data set could also be used to compute the grid in advance. This would be particularly advantageous, since the algorithm can then start from the first available data. The time point n is chosen as the starting time (t_0) for the sequential method.

INLA also provides the mean and covariance matrix for the Gaussian approximation to the posterior of state variables, $\mathbb{P}(\mathbf{X}_n \mid \mathbf{Y}_n, \theta)$. The mean and the variance of the Gaussian approximation are μ and $\mathbf{V}$ respectively. Then one can choose μ_n and $\mathbf{V}_{n,n}$ as the initial information for the filtering algorithms. Any other statistic, for example the sufficient statistics, can also be computed and stored using INLA. Certain special methods, say for example linear Bayes methods (Hartigan 1969), use these statistics and not necessarily the mean and/or the variance.

4.3 Sequential Bayesian Estimation of θ

Parallel sequential Bayesian inference of $\theta \,|\, \mathbf{Y}_{1:t}$ is done to each point on the grid, and the posterior carried forward in time. So when a new datum $\mathbf{Y}_t$ is recorded at time t, the objective is to update the conditional posterior of θ from that at time $t-1$, $\mathbb{P}(\theta \,|\, \mathbf{Y}_{1:t-1})$, to that at time t, $(\mathbb{P}(\theta \,|\, \mathbf{Y}_{1:t}))$. This can be better represented by the following diagram:

$$\mathbb{P}(\theta \,|\, \mathbf{Y}_{1:t-1}) \xrightarrow{\quad \mathbf{Y}_t \quad} \mathbb{P}(\theta \,|\, \mathbf{Y}_{1:t}).$$

Supposing that one has an approximation to $\mathbb{P}(\theta \,|\, \mathbf{Y}_{1:t-1})$ at time $t-1$, which is obtained by some inferential method (e.g. INLA) and is denoted by $\tilde{\mathbb{P}}(\theta \,|\, \mathbf{Y}_{1:t-1})$. The question now arises as to finding a sequential expression say $g_t(\theta)$ at time t, which when multiplied to this approximation $\tilde{\mathbb{P}}(\cdot)$ gives a good approximation to the posterior at time t maybe up to a constant, i.e.

$$\tilde{\mathbb{P}}(\theta \,|\, \mathbf{Y}_{1:t}) = \tilde{\mathbb{P}}(\theta \,|\, \mathbf{Y}_{1:t-1}) \times g_t(\theta).$$

Using Bayes' law and the structure of the state-space models, the posterior density can be written in the following recursive form:

$$\begin{aligned}
\mathbb{P}\left(\theta \,|\, \mathbf{Y}_{1:t}\right) &\propto \mathbb{P}\left(\theta, \mathbf{Y}_{1:t}\right), \\
&\propto \mathbb{P}\left(\theta, \mathbf{Y}_t, \mathbf{Y}_{1:t-1}\right), \\
&\propto \mathbb{P}\left(\theta, \mathbf{Y}_{1:t-1}\right) \frac{\mathbb{P}\left(\mathbf{Y}_t, \mathbf{Y}_{1:t-1}, \theta\right)}{\mathbb{P}\left(\mathbf{Y}_{1:t-1}, \theta\right)}, && (4.3.1) \\
&\propto \mathbb{P}\left(\theta \,|\, \mathbf{Y}_{1:t-1}\right) \mathbb{P}\left(\mathbf{Y}_t \,|\, \mathbf{Y}_{1:t-1}, \theta\right), && (4.3.2) \\
&\propto \mathbb{P}\left(\theta \,|\, \mathbf{Y}_{1:t-1}\right) \frac{\mathbb{P}\left(\mathbf{Y}_t \,|\, \mathbf{Y}_{1:t-1}, \mathbf{X}_t, \theta\right) \mathbb{P}\left(\mathbf{X}_t \,|\, \mathbf{Y}_{1:t-1}, \theta\right)}{\mathbb{P}\left(\mathbf{X}_t \,|\, \mathbf{Y}_{1:t}, \theta\right)}, && (4.3.3) \\
&\propto \mathbb{P}\left(\theta \,|\, \mathbf{Y}_{1:t-1}\right) \frac{\mathbb{P}\left(\mathbf{Y}_t \,|\, \mathbf{X}_t, \theta\right) \mathbb{P}\left(\mathbf{X}_t \,|\, \mathbf{Y}_{1:t-1}, \theta\right)}{\mathbb{P}\left(\mathbf{X}_t \,|\, \mathbf{Y}_{1:t}, \theta\right)}, && (4.3.4)
\end{aligned}$$

where the transition from Equations (4.3.1) to (4.3.3) is via usage of Bayes' theorem, with the state process $\mathbf{X}_t$ being introduced in Equation (4.3.3). The assumption of conditional independence of $\mathbf{Y}_t$ results in the simplification from Equation (4.3.3) to Equation (4.3.4). So from Equation (4.3.4), one can see that an approximation to the posterior of θ at time

t can be made as:

$$\tilde{\mathbb{P}}\left(\theta \mid \mathbf{Y}_{1:t}\right) \approx \text{Constant} \times \mathbb{P}\left(\theta \mid \mathbf{Y}_{1:t-1}\right) \left.\frac{\mathbb{P}\left(\mathbf{Y}_t \mid \mathbf{X}_t, \theta\right) \mathbb{P}\left(\mathbf{X}_t \mid \mathbf{Y}_{1:t-1}, \theta\right)}{\mathbb{P}\left(\mathbf{X}_t \mid \mathbf{Y}_{1:t}, \theta\right)}\right|_{\mathbf{x}_t^*(\theta)}. \tag{4.3.5}$$

$\mathbf{x}_t^*(\theta)$ is any value of $\mathbf{X}_t$ for which the denominator in Equation (4.3.5) is positive.

The terms $\mathbb{P}\left(\mathbf{X}_t \mid \mathbf{Y}_{1:t-1}, \theta\right)$ and $\mathbb{P}\left(\mathbf{X}_t \mid \mathbf{Y}_{1:t}, \theta\right)$ have very complicated forms themselves. Assuming we have an approximation to the posterior at time $t-1$ given by $\tilde{\mathbb{P}}\left(\mathbf{X}_{t-1} \mid \mathbf{Y}_{1:t-1}, \theta\right)$, we need to project it forward in time to calculate the *prior distribution* $\mathbb{P}\left(\mathbf{X}_t \mid \mathbf{Y}_{1:t-1}, \theta\right)$. This is achieved using the transition probability,

$$\mathbb{P}\left(\mathbf{X}_t \mid \mathbf{Y}_{1:t-1}, \theta\right) = \int \mathbb{P}\left(\mathbf{X}_t \mid \mathbf{X}_{t-1}, \theta\right) \mathbb{P}\left(\mathbf{X}_{t-1} \mid \mathbf{Y}_{1:t-1}, \theta\right) \, d\mathbf{X}_{t-1},$$

which gives us the following approximation:

$$\tilde{\mathbb{P}}\left(\mathbf{X}_t \mid \mathbf{Y}_{1:t-1}, \theta\right) \approx \int \mathbb{P}\left(\mathbf{X}_t \mid \mathbf{X}_{t-1}, \theta\right) \tilde{\mathbb{P}}\left(\mathbf{X}_{t-1} \mid \mathbf{Y}_{1:t-1}, \theta\right) \, d\mathbf{X}_{t-1}. \tag{4.3.6}$$

If possible, one should try to compute this integral exactly. Mostly however, the above integral is approximated by some sub-optimal method like the EKF, for example. When a new observation arrives, it is incorporated in the calculation of the posterior using the likelihood function,

$$\mathbb{P}\left(\mathbf{X}_t \mid \mathbf{Y}_{1:t}, \theta\right) \propto \mathbb{P}\left(\mathbf{Y}_t \mid \mathbf{X}_t, \theta\right) \mathbb{P}\left(\mathbf{X}_t \mid \mathbf{Y}_{1:t-1}, \theta\right),$$

resulting in the approximation,

$$\tilde{\mathbb{P}}\left(\mathbf{X}_t \mid \mathbf{Y}_{1:t}, \theta\right) \propto \mathbb{P}\left(\mathbf{Y}_t \mid \mathbf{X}_t, \theta\right) \tilde{\mathbb{P}}\left(\mathbf{X}_t \mid \mathbf{Y}_{1:t-1}, \theta\right). \tag{4.3.7}$$

The recursion formula, sometimes involving high-dimensional integrals is only available in closed form in certain special situations. For a linear Gaussian model, with Gaussian priors for the hyperparameters, exact solutions are available. Closed form solutions are also available if conjugacy properties can be used. For the special case when the distribution of the state process belongs to an exponential family of distributions, Bather (1965) has shown that such integrals are solvable by updating a set of sufficient statistics. This has been extended to include the Kalman filter framework by Chen & Singpurwalla (1994).

Different methods to compute the integrals will be shown in the next couple of sections. While some are based on crude approximations, others make use of already existing theory in filtering. These inference techniques are discussed below.

4.3.1 Crude Solution

In this section, the problem of identifying and solving for $g_t(\theta)$ is approached in a manner which is different from that of Equation (4.3.4). At time $t-1$, assume that an approximation to $\mathbb{P}(\theta \,|\, \mathbf{Y}_{1:t-1})$ exists, say by INLA, and denoted by $\mathbb{P}_{INLA}(\theta \,|\, \mathbf{Y}_{1:t-1})$. So here, the formulation of the multiplier is:

$$\mathbb{P}_{INLA}(\theta \,|\, \mathbf{Y}_{1:t}) = \mathbb{P}_{INLA}(\theta \,|\, \mathbf{Y}_{1:t-1}) \times g_t(\theta). \tag{4.3.8}$$

The functional multiplier $g_{t-1}(\theta)$ can be approximated from Equation (4.3.8) in the following way:

$$\begin{aligned}
g_t(\theta) &= \frac{\mathbb{P}_{INLA}(\theta \,|\, \mathbf{Y}_{1:t})}{\mathbb{P}_{INLA}(\theta \,|\, \mathbf{Y}_{1:t-1})}, \\
&\propto \frac{\mathbb{P}(\mathbf{X}_{1:t} \,|\, \theta)\mathbb{P}(\mathbf{Y}_{1:t} \,|\, \mathbf{X}_{1:t}, \theta)}{\mathbb{P}_G(\mathbf{X}_{1:t} \,|\, \mathbf{Y}_{1:t}, \theta)} \Bigg/ \frac{\mathbb{P}(\mathbf{X}_{1:t-1} \,|\, \theta)\mathbb{P}(\mathbf{Y}_{1:t-1} \,|\, \mathbf{X}_{1:t-1}, \theta)}{\mathbb{P}_G(\mathbf{X}_{1:t-1} \,|\, \mathbf{Y}_{1:t-1}, \theta)}, & (4.3.9) \\
&\propto \frac{\mathbb{P}(\mathbf{X}_{1:t} \,|\, \theta)}{\mathbb{P}(\mathbf{X}_{1:t-1} \,|\, \theta)} \frac{\mathbb{P}(\mathbf{Y}_{1:t} \,|\, \mathbf{X}_{1:t}, \theta)}{\mathbb{P}(\mathbf{Y}_{1:t-1} \,|\, \mathbf{X}_{1:t-1}, \theta)} \frac{\mathbb{P}_G(\mathbf{X}_{1:t-1} \,|\, \mathbf{Y}_{1:t-1}, \theta)}{\mathbb{P}_G(\mathbf{X}_{1:t} \,|\, \mathbf{Y}_{1:t}, \theta)}, & (4.3.10) \\
&\approx \frac{\mathbb{P}(\mathbf{X}_{1:t} \,|\, \theta)\mathbb{P}(\mathbf{Y}_t \,|\, \mathbf{X}_t, \theta)}{\mathbb{P}(\mathbf{X}_{1:t-1} \,|\, \theta)\mathbb{P}_G(\mathbf{X}_t \,|\, \mathbf{Y}_t, \theta)}, & (4.3.11) \\
&\propto \frac{\mathbb{P}(\mathbf{X}_t \,|\, \mathbf{X}_{t-1}, \theta)\mathbb{P}(\mathbf{Y}_t \,|\, \mathbf{X}_t, \theta)}{\mathbb{P}_G(\mathbf{X}_t \,|\, \mathbf{Y}_t, \theta)}. & (4.3.12)
\end{aligned}$$

The approximations that are incorporated into the equations above are listed below:

- $P_G(\mathbf{X}_{1:t} \,|\, \mathbf{Y}_{1:t}, \theta) \approx P_G(\mathbf{X}_t \,|\, \mathbf{Y}_{1:t}, \theta) P_G(\mathbf{X}_{1:t-1} \,|\, \mathbf{Y}_{1:t}, \theta)$,
- $P_G(\mathbf{X}_t \,|\, \mathbf{Y}_{1:t}, \theta) \approx P_G(\mathbf{X}_t \,|\, \mathbf{Y}_t, \theta)$,
- $P_G(\mathbf{X}_{1:t-1} \,|\, \mathbf{Y}_{1:t}, \theta) \approx P_G(\mathbf{X}_{1:t-1} \,|\, \mathbf{Y}_{1:t-1}, \theta)$.

The multiplier defined above puts the idea of INLA into a sequential context up to a constant of normality, even though it is very crude in its approximation. At each time point, the sequential multiplier is computed at the mode of the Gaussian approximation term $\mathbb{P}_G(\mathbf{X}_t \,|\, \mathbf{Y}_t, \theta)$. If the mode is computed as $\mathbf{x}_t^*$, $g_t(\theta)$ is written as,

$$g_t(\theta) = \left. \frac{\mathbb{P}(\mathbf{X}_t \,|\, \mathbf{X}_{t-1}, \theta)\mathbb{P}(\mathbf{Y}_t \,|\, \mathbf{X}_t, \theta)}{\mathbb{P}_G(\mathbf{X}_t \,|\, \mathbf{Y}_t, \theta)} \right|_{\mathbf{X}_t = \mathbf{x}_t^*}. \tag{4.3.13}$$

A Gaussian approximation defined in Equation (4.3.13) should be quick to compute, since the density to be approximated is univariate in terms of time. This property of the sequential multiplier would make this inference procedure very fast. However there remain doubts over the validity of the approximations explained in this section. We will be discussing more about the performance of the sequential multiplier and the effect of the approximations on it in Chapter 6.

4.3.2 Computation of $\mathbb{P}(\mathbf{X}_t \,|\, \mathbf{Y}_{1:t-1}, \theta)$ and $\mathbb{P}(\mathbf{X}_t \,|\, \mathbf{Y}_{1:t}, \theta)$ for Equation (4.3.4)

In this section we concentrate on computing Equation (4.3.4) without making any crude approximations related to it. Equation (4.3.4) requires computation of the two terms $\mathbb{P}(\mathbf{X}_t \,|\, \mathbf{Y}_{1:t-1}, \theta)$ and $\mathbb{P}(\mathbf{X}_t \,|\, \mathbf{Y}_{1:t}, \theta)$ and also of the value $\mathbf{x}_t^*$. As discussed earlier, exact computations of these terms are possible for certain specific models. Linear models with additive Gaussian errors enables the usage of the Kalman filter. Kalman filter propagates the mean and covariance of the state process distribution with time. Since all distributions are Gaussian and the model is absolutely linear, the mean and the variance totally explain the posterior which is also Gaussian. INLA provides the starting mean and variance at each grid point. Thus at time t_0, the mean μ_n and variance $\mathbf{V}_{n,n}$ are where the algorithm starts from. The posterior mean of the filter is taken as the value of $\mathbf{x}_t^*$ at each iteration. For each value of θ on the grid, the posterior, the mean and the variance are calculated and stored. These stored values get updated at each iteration as new data comes in and sequential updates are made using them.

For models with nonlinearity and non-Gaussian property, extensions of the Kalman filter like EKF or UKF are used to calculate the integrals. In certain special cases, where the state distribution comes from the exponential family, certain sufficient statistics are propagated the same way as in Kalman filter. More general methods like EKF would linearize a nonlinear model, and provide an approximation to the mean and variance of the posterior distribution of the state process, assuming a Gaussian distribution of the same. UKF would allow the nonlinearity to be propagated by choosing deterministic points and then calculating the weighted mean and weighted variance from the non-linear model. Both the above mentioned methods make use of Kalman filter by approximating the non linearity and the non-Gaussian property. It is assumed here that only the Kalman filter or extensions of it are used to compute these terms because of the simplicity in their application and high computation speed. Primary reasons for using the EKF/UKF are that, other than the fact that they are very well known and easy to implement, they are fast and codes are available very easily from existing sources. However other methods are equally applicable. As long as there is a grid defining the posterior of θ, any general method can be used here.

It is known that INLA provides the starting mean and variance (or other sufficient statistics as required) at each grid point. Thus for an algorithm based on extensions to Kalman filter, the mean μ_n and variance $\mathbf{V}_{n,n}$ at time $t_0 = n$ is where the sequential algorithm starts from. At each value of θ on a grid, the posterior, the mean and the variance (or some other statistic) are calculated and stored. These stored values are updated at each iteration.

4.4 Algorithm

Thus an initial algorithm can be set up for the sequential inference procedure explained so far. If it is reasonable to assume that the grid will stay stable and no changes need to be made to it, then the following algorithm defines the proposed sequential Bayesian inference method.

Algorithm 1 Sequential INLA with a fixed grid

```
$t = 1$
**while** $t < n$ **do**
    Observe $\mathbf{Y}_{1:t}$,
    Compute $\mathbb{P}\left(\theta \mid \mathbf{Y}_{1:t}\right)$ by INLA.
    $t = t + 1$.
**end while**
**repeat**
    Observe $\mathbf{Y}_t$,
    Compute $g_t(\theta)$ by methods already discussed,
    Update $\mathbb{P}(\theta \mid \mathbf{Y}_{1:t}) = \mathbb{P}(\theta \mid \mathbf{Y}_{1:t-1}) \times g_t(\theta)$,
    $t = t + 1$
**until** $t = t_{end}$
```

Algorithm 1 updates the conditional posterior of θ at each of the grid points. In most of the problems however the grid needs to be updated i.e. points need to be added to the existing grid to cover the support of the posterior density or deleted to expedite the algorithm. This can be because of several reasons, the most prominent being the learning process that the algorithm goes through as more and more data affects the process and the grid has to mould to this. Also an outlier can also cause the grid to shift. The grid updating procedure is discussed in detail in the following section.

4.5 Updating the Grid

4.5.1 Motivation

A fixed grid over the support of $\mathbb{P}(\theta \mid \mathbf{Y}_{1:t})$ is not a reasonable assumption in sequential inference. The support of $\mathbb{P}(\theta \mid \mathbf{Y}_{1:t})$ may translate, shrink or expand, making any predefined grid a poor representation of the support, which leads to a poor approximation to the posterior. Some method of detecting when any of these happen, and updating the grid appropriately, is necessary. The grid should be able to add or drop grid points as and when necessary, thus making dynamic updates to the existing one. Ideally there can be one of the following possible situations:

- Grid points are dropped if the value of posterior density at those points is close to zero.

- The posterior expands and hence the grid needs to be made more coarse.
- The posterior shrinks, requiring a refinement of the grid.

The grid formed by INLA does not cover the support of a posterior distribution with heavy tails. The grid making process is terminated as soon as the grid points reach the tails of the density. In the sequential process, shrinking of the support of the posterior means that some of the grid points at the tails have very small values of the posterior. Since the support is shrinking, it is more sensible to add more points in the region of high density and drop the points where the density is very small. Such points with very low density values are dropped to facilitate a faster computation time.

It can also be necessary to make the support coarser or more refined. In other words, new grid points outside the range of the current grid may be required. Similarly, new points could be needed within the range of the grid. One possible reason for this to happen is that the starting grid can be inaccurate in the sense that it does not contain the true support of the parameter we are interested in. More will be discussed about this in Section 4.5.3. This section will explain different ways to add (both internally and externally) or delete grid points and also the calculation of the unknown quantities associated with a new grid point. It should also be mentioned here, that each new grid point requires estimates related to the algorithm, other than the posterior density. So for a linear Gaussian model, for example, the mean and the variance of the state process, along with the filtering density need to be computed for a new grid point, whether internal or external. All the examples stated here will be based on the computation of the log-posterior at a new point, but all other estimates are calculated in a similar manner.

Here we will elaborate a bit more on computations of other statistics estimated at the new grid points. It has been mentioned earlier that these estimates are particular to the filtering methodology that is used in the sequential inference procedure. In our examples, we have used either the Kalman filter or extensions of Kalman filter whichever suited the problem at hand. For all these methods, the first and second order moments of the filtering density $\mathbb{P}(\mathbf{X}_t \mid \mathbf{Y}_{1:t}, \theta)$ are propagated forward in time. So for any new grid point, that is to be added to the current support, the first and second moments of the filtering density for that new grid point also need to be computed. The grid updates will be explained in terms of the computation of the posterior density at the new points, but any computation related to these moments will follow the exact same methodology.

The most important property of a dynamic grid, from a computational perspective, is to add new points as infrequently as possible. Adding new grid points at each time point makes the algorithm slow, since even a single extra grid point caters to the *curse of dimensionality*. One possible way is to make the initial grid wide relative to the support of the distribution. The initial construction of the grid is based only on very few observations, hence the support of the prior will have a strong effect on the support of the posterior.

Thus a prior with very low variability will possibly make the support of the posterior too narrow. Such a posterior could be far from the posterior based on a large number of observations. As new data start coming in, this grid may now have to be translated (possibly quite a bit). A flat prior on the parameters in such cases could be very helpful. A very wide support will, hopefully, not require any external points. This keeps the algorithm fast.

Adding a new grid point internally, i.e. within the range of the existing grid, is easier. A new internal grid point will be necessary when the support shrinks as more and more data trains the process and hence the existing grid structure appears to be too sparse for the current support. Figure 4.1 is a toy example where it can be seen that some new internal grid points will be necessary for the parameter. The support has shrunk considerably and some points can also possibly be dropped along the edge of the current grid.

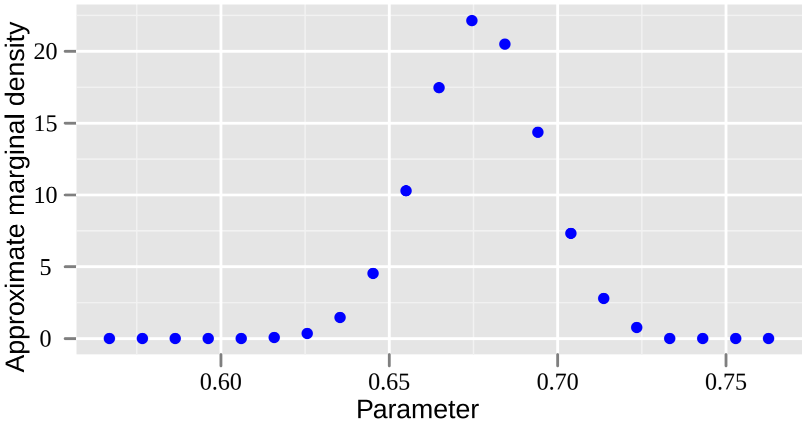


Figure 4.1: This plot shows the posterior marginal density of a parameter from some toy example. The posterior density denoted by the points on the plot clearly shows that grid points on the tail have negligible posterior marginal values, which can be dropped to facilitate computation. Also a refinement of grid is required, since the supporting grid is shrinking and new points are required to be added internally.

Figure 4.2 is another toy example where we see that a few grid points on the right tail of the current density can be dropped. It is easy to see that the approximate density in this example needs to be translated to the left side of the current support, meaning that external points need to be added to the left tail of the current grid. Thus we see that, it is very necessary to keep a constant track of grid points that has become redundant and can be dropped, in addition to identifying areas where new grid points need to be added. This idea of shedding unnecessary helps in keeping a high computation speed of the algorithm
.

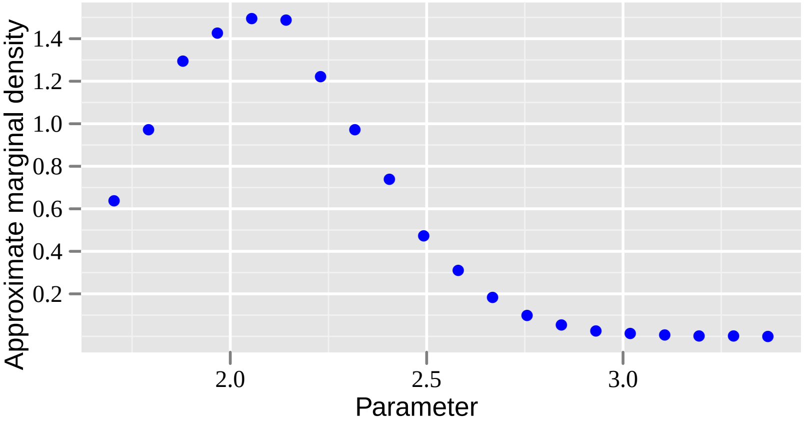


Figure 4.2: The posterior marginal density in this plot clearly shows that new points need to be added to the left of the existing grid. Also some existing grid points from the right tail can be dropped since the value of the marginal density at those points are very small.

4.5.2 New Internal Grid point

4.5.2.1 Identifying when and where to add a new internal point

The first question one needs to answer is how to determine the requirement for a new grid point. For each of the parameters, this is done using their marginal densities. The calculation of marginal densities, although it includes a lot of summation over the grid, can still be done very quickly for the dimensions that we are considering here. Some criteria can be defined based on the approximate densities of adjacent grid points which determines the requirement of a new point in between. We have used the Euclidean distance between the approximate density of two successive grid points scaled between 0 and 1 to implement this criterion. The scaling is done to ensure that the size of the support of the marginal distributions for each of the parameters do not affect the addition rule. If the distance is above a certain threshold, a new grid point is to be included in between the two grid points. Let θ_1 and θ_2 be two adjacent grid points with $\theta_1 < \theta_2$ for some one-dimensional parameter θ. The approximate posterior marginal density at these points are $\tilde{\mathbb{P}}(\theta_1)$ and $\tilde{\mathbb{P}}(\theta_2)$ respectively. If the absolute distance between the two densities is greater than some specific value, say δ, then a new point is chosen between θ_1 and θ_2. Thus the new internal grid point, θ_* is given as:

$$\theta_* = \theta_1 + \frac{\theta_2 - \theta_1}{2} \quad \text{if} \quad \left| \frac{\tilde{\mathbb{P}}(\theta_2) - \tilde{\mathbb{P}}(\theta_1)}{\max_\theta \tilde{\mathbb{P}}(\theta)} \right| > \delta.$$

In Figure 4.3, assuming the distance metric between the points θ_1 and θ_2 is greater than δ, then the new grid point θ_* is added in between the two existing points.

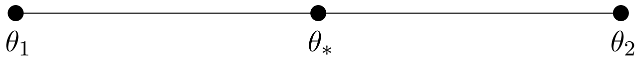


Figure 4.3: Single point added between two points for a single variable.

However for two parameters, both of which require the inclusion of a new point, the situation is more complex. A graphical representation of such a case is presented in Figure 4.4. Let θ_1 and θ_2 be two parameters, the joint posterior of which is defined on a grid. Assume also that a single internal point needed to be added for each of them, as computed with the help of the respective marginal densities. For the parameter θ_1, a new internal grid point is to be put between the already existing points θ_1^1 and θ_1^2. Similarly, θ_2^1 and θ_2^2 for the other parameter θ_2. The new points on the joint posterior grid, which require the posterior density to be calculated, can be shown by slicing a section out from the joint grid of the two parameters. The "$\otimes$" and the "$\oplus$" symbols represent the new points on the joint grid for each of which the posterior density needs to be calculated. Thus even though individual points for a particular parameter are identified from its marginal density, the joint posterior grid requires calculation on all possible combinations.

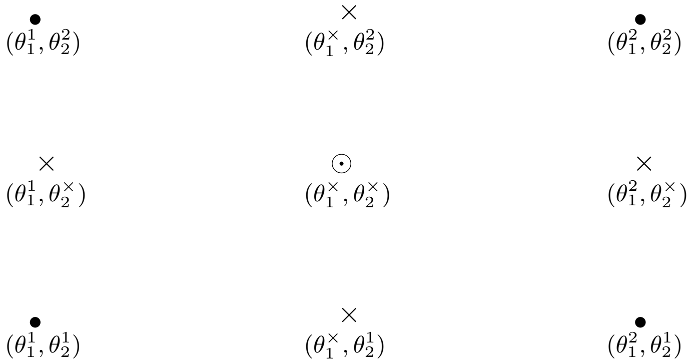

(θ_1^1, θ_2^2)
$(\theta_1^\times, \theta_2^2)$
(θ_1^2, θ_2^2)
$(\theta_1^1, \theta_2^\times)$
$(\theta_1^\times, \theta_2^\times)$
$(\theta_1^2, \theta_2^\times)$
(θ_1^1, θ_2^1)
$(\theta_1^\times, \theta_2^1)$
(θ_1^2, θ_2^1)

Figure 4.4: The $\times$'s and the $\odot$ in this plot shows possible combinations of new internal points for each of the two parameters. The points $(\theta_1^\times, \theta_2^1), (\theta_1^\times, \theta_2^2), (\theta_1^1, \theta_2^\times)$ and $(\theta_1^2, \theta_2^\times)$ require interpolation on a single variable. Bivariate rule is needed for the combination $(\theta_1^\times, \theta_2^\times)$.

The identification of a new internal grid point, as explained above, has been applied in a toy example. The points in black in Figure 4.5 resemble the existing grid with the blue circles being the approximate density values. The points in red denote the added internal points. Those points are chosen on the basis of the above algorithm.

4.5.2.2 Calculating the posterior and other necessary values at new grid point(s)

Ideally the calculation of the density at the new points should be done by the same approximating method by which the values on the existing grid were derived. Thus, assuming that at time point T, a new internal point θ is added to the grid structure, and assuming INLA was used to construct the starting grid, the density at the new point is

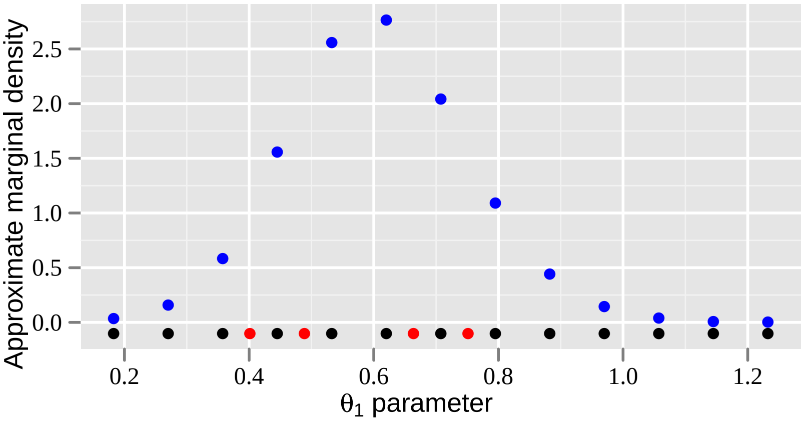


Figure 4.5: This plot has blue circles denoting the posterior marginal density and black circles denote the grid points representing the support of the density. The red circles are new points which need to be added to the existing grid and are added using the rule explained in Section 4.5.2.1.

given by $\mathbb{P}_{INLA}(\theta \mid \mathbf{Y}_{1:T})$. However, there are two major problems which do not allow this procedure to be carried out in general. First, data in a sequential procedure may not be stored over time as the size of the data-set becomes larger and storage becomes an issue in many applications. Hence all the observations up to and including $\mathbf{Y}_{1:T}$ are generally not available. Secondly, assuming that all the data are stored, the approximating procedure will take a long time to compute, which increases as T increases. Hence using the same approximation is not only impossible in certain cases, but also infeasible. Nevertheless the density calculation can be done in several ways that are feasible computationally. A smooth function can be used to fit the joint density and predictions can be made at the new grid points. Alternatively interpolation, which is straightforward to implement but could be inaccurate, can also be used here. We will discuss this issue thoroughly in Sections 4.5.2.3, 4.5.2.4 and 4.5.2.6.

The problem of calculation of the density is better understood through an example. The parameter values of the grid and the associated log-density are stored in an array, as in Table 4.1. There are three parameters on which the grid is computed: $\theta = (\theta_1, \theta_2, \theta_3)$. All possible combinations of parameter grid points are stored along with their log-posterior density values. This same example has produced Figure 4.5. After calculating the marginal density, it can be seen that new grid points need to be added to the θ_1 parameter between points $(0.3577293, 0.4452042)$ and $(0.4452042, 0.5326790)$. Actually four new points need to be added, but here only two are shown as an example. The log-posterior is calculated at the new points 0.4014667 and 0.4889416 using the joint density (and not their marginal densities) and the array now becomes that of Table 4.2. The new matrix will now contain all possible combinations of the new points and all the other parameter points along with their predictions. In this example, the density at the new grid point is calculated by linear

Table 4.1: Array in which the parameter values and approximate log-posterior values are stored

θ_1	θ_2	θ_3	Approx. log-posterior
0.2702545	−1.718234	−17.27443	−904.1742
0.3577293	−1.718234	−17.27443	−902.5360
0.4452042	−1.718234	−17.27443	−901.3573
0.5326790	−1.718234	−17.27443	−900.7947
0.6201539	−1.718234	−17.27443	−900.7706
⋮	⋮	⋮	⋮

Table 4.2: Updated matrix where the internal grid is being updated.

θ_1	θ_2	θ_3	Approx. log-posterior
0.2702545	−1.718234	−17.27443	−904.1742
0.3577293	−1.718234	−17.27443	−902.5360
0.4014667	−1.718234	−17.27443	−901.9466
0.4452042	−1.718234	−17.27443	−901.3573
0.4889416	−1.718234	−17.27443	−901.0760
0.5326790	−1.718234	−17.27443	−900.7947
0.6201539	−1.718234	−17.27443	−900.7706
⋮	⋮	⋮	⋮

interpolation, but of course, as discussed earlier it can be done by any other suitable method, like smoothing. The marginal density at the new grid which includes the new internal points is plotted in Figure 4.6. The red circles represent the log-posterior marginal density values that are calculated at the new grid points.

As has been said earlier, there are many ways to make predictions the log-posterior values at the new grid points. Curve fitting methods can be used instead of linear interpolation that has been used here. One small note here should be added, that the curve fitting for more than one parameter has to be done on the joint density. So multi-variate curve fitting methods are required. To fit a curve one has to make use of methods which are quick and moderately accurate. One major problem is that with increasing dimensions, the fitting mechanism also becomes very slow and inaccurate. Methods that can be used in this situation include nonparametric fitting (such as splines and different versions of it), generalized additive models, polynomial interpolation etc. In this thesis, Generalized Additive Models (Hastie & Tibshirani 1990) and Multivariate Adoptive Regression Splines Friedman (1991) are used to fit the joint density in our toy example because of their fast computation properties in high dimensions. GAM and MARS will be explained in short in the following sections.

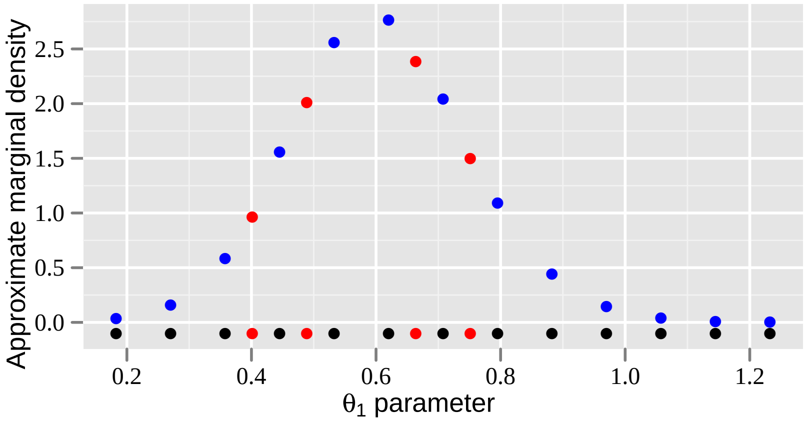


Figure 4.6: This plot shows the approximate posterior marginal density at the new points added internally to the toy example shown in Figure 4.5. In this toy example, the approximate posterior at these new points are calculated using linear interpolation.

4.5.2.3 The Generalized Additive Model

The *Generalized Additive Model* (GAM) (Hastie & Tibshirani 1990, Wood 2004) is essentially a *Generalized Linear Model* (GLM) in which part of the linear predictor is specified in terms of smooth functions of covariates,

$$g\left(\mathbb{E}(\mathbf{y}_i)\right) = \mathbf{X}_i\theta + f_1(\mathbf{x}_{1i}) + f_4(\mathbf{x}_{2i}, \mathbf{x}_{3i}) + f_3(\mathbf{x}_{4i}) + \cdots,$$

where $\mathbf{y} \sim$ ***exponential family distribution***, $\mathbf{X}$ is the matrix of covariates and $g(\cdot)$ is the link function. One starts with choosing a spline type basis for each of these $f_i(\cdot)$, thus defining it as $f(x) = \sum_{i=1}^{k} \beta_i b_i(x)$ where b_i are known but the β_i are to be estimated. If the model is estimated by likelihood maximization, then it will tend to overfit (i.e. under smooth). This tendency can be controlled during fitting using penalization (Wood 2006). So we choose a measure of smoothness for f, e.g. :

$$\mathbb{J}(f) = \int f''(x)^2 dx = \beta^T \mathbf{S} \beta.$$

And finally *penalized* MLE is used to estimate the unknowns:

$$\text{Minimize } 2\left(l(y) - l(\cdot)\right) + \lambda\beta^T\mathbf{S}\beta \text{ with respect to } (\beta, \lambda).$$

Note that $2\left(l(y) - l(\cdot)\right)$ is the *deviance* which actually is defined as 2[(Max. log likelihood) - (log likelihood for GAM)] at the MLE. The smoothing parameter λ controls the smoothness vs. fit trade-off and is chosen so as to minimize prediction error (predictive deviance). Model selection criteria such as GCV, AIC, etc. are used in this case to determine the number of predictors.

In case of higher dimension smooths, different basis functions such as thin plate regression splines or tensor product smooths can be used. A few extra properties are being kept in mind doing multivariate smoothing since it can be computationally very complex. For example, there is no need to choose knots in GAM unlike in splines. Also independence to linear rescaling of covariates is maintained for fast computation. The interested reader can look at Wood (2003) for an in depth account.

A fit with default bandwidth and thin plate regression splines as basis functions was tried. Figure 4.7 shows the marginal density of the parameter θ_1. The marginal posterior density values at two of those points shows some irregularity which indicates an under-smooth fit. This can possibly be fixed or improved with a better chosen bandwidth. But

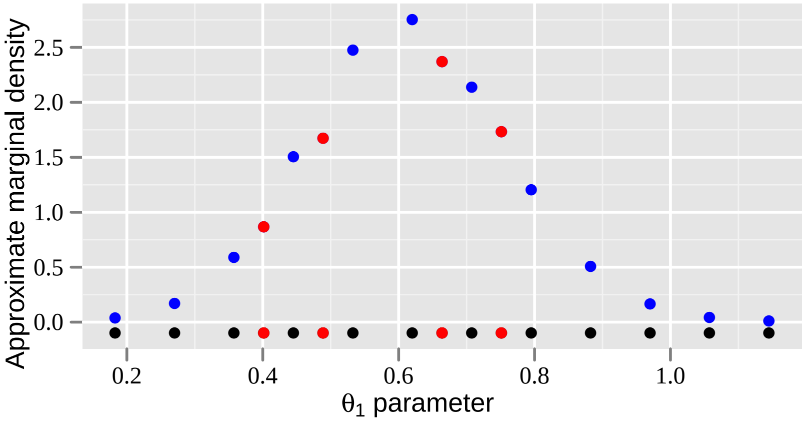


Figure 4.7: This plot also shows the approximate posterior marginal density of the new internal points in the original grid as in Figure 4.5. The red circles denoting the approximate posterior at internal points are computed using GAM. Note the over fit in the density for the new points.

this immediately shows the weakness of this method in the sequential setting. There has to be a control over the bandwidth, which can possibly change from one iteration to the other. But that is not practical in sequential inference. It is not at all feasible that one is forced to monitor and adjust a smoothing algorithm in a sequential inferential method since it will slow the algorithm down.

4.5.2.4 Multivariate Adaptive Regression Splines

Multivariate Adaptive Regression Splines or MARS is a form of regression analysis which solves the high dimensional smoothing problem geometrically (Friedman 1991). The starting point is to define the smoothing function $f(\cdot)$ as a linear combination of basis functions:

$$f(\mathbf{y}) = \sum_{m=1}^{M} a_m B_m(\mathbf{y}).$$

Now, the whole region (data) over which $f(\cdot)$ is defined as $\mathbb{D}$ and it has to be divided into non-overlapping subregions $\{R_m\}_1^M$ according to a specific rule. In each subregion the basis function is defined as:

$$B_m(\mathbf{y}) = \mathbb{I}[\mathbf{y} \in R_m].$$

At a given stage of partitioning, all the existing subregions are each optimally split into two (daughter) subregions. So given a region R, which can be split into two subregions R_l and R_r, and some $t \in [-\infty, \infty]$, the eligible splits takes the form,

$$\begin{aligned} &\text{If } \mathbf{y} \in R, \text{ and if } x_v \leq t, \\ &\qquad \text{then } \mathbf{y} \in R_l, \\ &\qquad \text{else } \mathbf{y} \in R_r. \end{aligned}$$

where v labels one of the covariates. The $\{a_m\}_1^M$ values are the coefficients which are jointly adjusted to give a best fit to the data based on some goodness-of-fit criteria. The recursive partitioning is done to adjust the coefficient values to best fit the data and also to derive a good set of basis functions. The partitioning is defined mathematically with the help of step functions like

$$H[\eta] = \begin{cases} 1 & : \eta \geq 0, \\ 0 & : \text{otherwise}, \end{cases}$$

for some η. Hence at some stage, given a predictor variable x_v and cut-off point t, the quantity being minimized is the lack-of-fit of a model with B_m being replaced by its product with the step function $H[+(x_v - t)]$ and added to it is the product of B_m and reflected step function (mirror image) $H[+(x_v - t)]$. This is equivalent to splitting the region R_m on variable v at split point t. In MARS however, to ensure continuity at the subregion boundaries, a two-sided truncated power basis function of order q ($[\pm(x-t)_+^q]$) is used.

Like most forward stepwise regression procedures, it makes sense to follow up a forward recursion by a backward stepwise procedure to remove basis functions that no longer contribute sufficiently to the accuracy of the fit. Generally the intention is to fit an excessively large model to the data in the forward process and then tone it down using the backward process. However just deleting the basis functions creates a problem, and hence deletion must happen in pairs. For that siblings are deleted in pairs and merged with their mother. To generalize the algorithm more, each region is not deleted after the subdivision in the forward step. So at each step, the parent and both her sibling basis functions are retained. The basis functions can be bivariate or trivariate or even of higher order taking into account the interaction between the variables. Thus the MARS algorithm produces a smoothing function of the type,

$$\tilde{f}(\mathbf{y}) = a_0 + \sum_{K_m=1} f_i(x_i) + \sum_{K_m=2} f_{ij}(x_i, x_j) + \sum_{K_m=3} f_{ijk}(x_i, x_j, x_k) + \cdots ,$$

where K_m denotes the number of splits. Each of those f's can be represented as a combination of basis functions.

The MARS algorithm was also implemented on the aforesaid example. In terms of a better smooth, it performs slightly better than GAM for this specific example. It is seen in our application to this toy example, that MARS is computationally faster than GAM when we use the R packages GAM and MARS (Hastie 2011, Milborrow et al. 2012). Figure 4.8 shows the marginal of the parameter θ_1 with the marginal posterior density values at the new internal grid points. However, do note that the MARS method suffers from the same drawbacks as that of the approach GAM. The stated drawbacks are discussed in more detail in the following section.

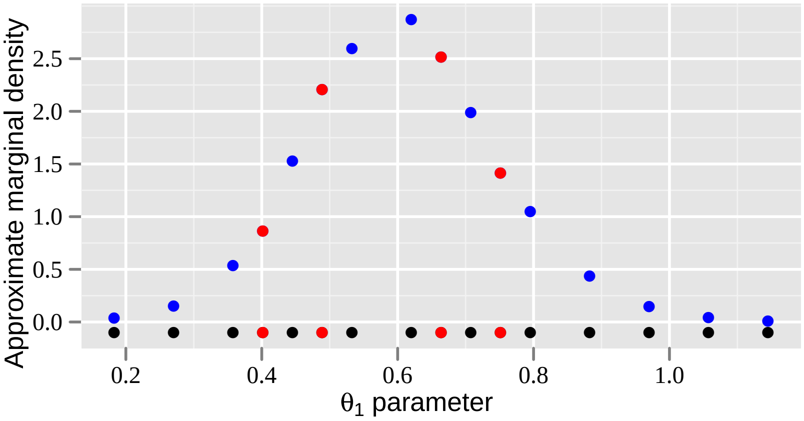


Figure 4.8: This plot shows the approximate posterior marginal density of the new internal points calculated using MARS. Note that the smoothing is better for this methodology than GAM, for the toy example that we have used in Figure 4.5.

4.5.2.5 Problem with Smoothing methods

High dimensional curve fitting methods, if properly tuned, will give the best predictions at the missing grid points. Since the density is a non-linear curve, linear interpolation will never give very accurate results. Theoretically, curve fitting methods can approximate the density up to an approximating constant. But these methods also bring have their negatives.

As discussed in brief earlier, there exist problems with the use of these methods in a sequential setting. Of course it is obvious that choosing the correct fit is very important. But other factors such as which basis functions to choose is also a major concern. Given a multi-dimensional data set, manual control of certain factors (for example choosing the basis function) is necessary. That would mean tweaking the internal parameters or basis functions to be used at each time point, since the posterior changes support and shape with time. This means monitoring the fit at each time point to check for over or under-fitting, which in turn means to stop the process and get the best smooth. But that is not feasible at all in a sequential process because of the increase it causes to the computation time at each iteration. These problems make these methods difficult to use in dynamic filtering.

4.5.2.6 Interpolation

Interpolation techniques provide the simplest way to predict at new points on a curve, given that the fitted curve passes through all the function values. A well-known estimation method, based on interpolation and extrapolation techniques is *kriging* (Diggle et al. 1998). Kriging computes the functional value at some unknown spatial point by a linear combination of the existing values. The weights are computed by minimizing the variance of the functional value subject to unbiased condition. But in high dimensions, kriging becomes extremely difficult. It also is computationally very demanding.

Local interpolation is easier and quicker. It can be both linear and non-linear. Non-linear interpolation provides a more accurate prediction for new points. But this comes with the trade-off, similar to the smoothing techniques already explained in Sections 4.5.2.3 and 4.5.2.4, that non-linear interpolation on high dimensions becomes quite complex. Also it is difficult to decide on the number of posterior-density-points, in the neighborhood of the missing grid point, that would be required for the non-linear interpolation method to work well.

Linear interpolation has been used in this thesis to compute the density at the internal point(s). Simplicity is its strength, while inaccuracy its weakness. Linear interpolation should work quite well if the grid is fine. This means that a large number of points will be required to define the posterior density, which is at the expense of a bigger storage matrix and posterior updates that take more time. A very coarse grid will produce skewed approximations at the new internal points. Hence again there is a trade-off. The grid point selection and dropping methodology should be such that the balance between grid size and computation is maintained. Taking note of the speed and ease of computation, linear interpolation is used in the algorithm.

Univariate Interpolation:

Interpolation on one dimension is quite straightforward. Like in the example explained graphically earlier, let θ be the parameter to which internal grid points need to be added. Let θ_1 and θ_2 be two successive points of θ and let $\log \tilde{\mathbb{P}}(\theta_1)$ and $\log \tilde{\mathbb{P}}(\theta_2)$ be the log-posterior density at those points. Let θ^* be the new grid point. The log-density at θ^*

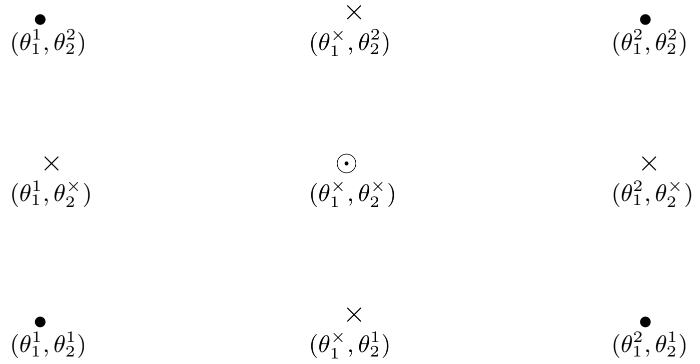
termed as $\log \tilde{\mathbb{P}}(\theta^*)$ is computed as,

$$\begin{aligned}\log \tilde{\mathbb{P}}(\theta^*) &\approx \log \tilde{\mathbb{P}}(\theta_1) + \frac{\partial \tilde{\mathbb{P}}(\theta)}{\partial \theta}(\theta^* - \theta_1), \\ &\approx \log \tilde{\mathbb{P}}(\theta_1) + \frac{\log \tilde{\mathbb{P}}(\theta_2) - \log \tilde{\mathbb{P}}(\theta_1)}{(\theta_2 - \theta_1)}(\theta^* - \theta_1).\end{aligned}$$

This is the building step of all interpolation that needs to be done in a multi-parameter setting. Table 4.2 is a an example where there are three parameters, but interpolation needed to be done to only one of them. The log-joint posterior density for parameter θ_1 was calculated using the formula above. Thus for the new internal grid point 0.401, the approximate density is computed in the following manner:

$$\begin{aligned}\log \tilde{\mathbb{P}}(0.401, -1.72, -17.274) \approx&\ \log \tilde{\mathbb{P}}(0.358, -1.72, -17.274) \\ &+ \frac{\log \tilde{\mathbb{P}}(0.445, -1.72, -17.274) - \log \tilde{\mathbb{P}}(0.358, -1.72, -17.274)}{(0.445 - 0.358)}(0.401 - 0.358).\end{aligned}$$

For new points added simultaneously to multiple parameters, the calculation becomes more complex. As an example, the calculations will be shown for two parameters, where each of them has one new internal grid point and calculations need to be done for a number of combinations. This can be better viewed using the geometric representation of the grid that has been explained earlier in this chapter. As before, let θ_1 and θ_2 be the two parameters. Also let $\theta_1^\times$ and $\theta_2^\times$ be the new points for each of the parameters respectively. The points are such that $\theta_1^1 < \theta_1^\times < \theta_1^2$ and $\theta_2^1 < \theta_2^\times < \theta_2^2$. This is represented in Figure 4.9. The log-posterior of the four points at the edges of Figure 4.9, namely $(\theta_1^\times, \theta_2^1), (\theta_1^\times, \theta_2^2), (\theta_1^1, \theta_2^\times)$ and $(\theta_1^2, \theta_2^\times)$ require interpolation on a single variable. However for the point at the center $(\theta_1^\times, \theta_2^\times)$, log-posterior needs to be computed using a bivariate rule, i.e. on two variables.



(θ_1^1, θ_2^2)
$(\theta_1^\times, \theta_2^2)$
(θ_1^2, θ_2^2)
$(\theta_1^1, \theta_2^\times)$
$(\theta_1^\times, \theta_2^\times)$
$(\theta_1^2, \theta_2^\times)$
(θ_1^1, θ_2^1)
$(\theta_1^\times, \theta_2^1)$
(θ_1^2, θ_2^1)

Figure 4.9: This plot is exactly same as Figure 4.4. This is provided here once more to better explain the bivariate interpolation rule of calculating the log-posterior density.

Bivariate Interpolation:

Bivariate interpolation is defined in a similar manner as the linear one. So as before, the approximate log-posterior density at the new point (θ_1^*, θ_2^*), defined as $\log \tilde{\mathbb{P}}(\theta_1^\times, \theta_2^\times)$, is computed using the following formula:

$$\log \tilde{\mathbb{P}}(\theta_1^\times, \theta_2^\times) \approx \log \mathbb{P}(\theta_1^1, \theta_2^1) + \frac{\partial \log \mathbb{P}(\theta_1^1, \theta_2^1)}{\partial \theta_1}(\theta_1^\times - \theta_1^1) + \frac{\partial \log \mathbb{P}(\theta_1^1, \theta_2^1)}{\partial \theta_2}(\theta_2^\times - \theta_2^1), \tag{4.5.1}$$

$$\approx \log \mathbb{P}(\theta_1^1, \theta_2^1) + \frac{\log \mathbb{P}(\theta_1^2, \theta_2^1) - \log \mathbb{P}(\theta_1^1, \theta_2^1)}{(\theta_1^2 - \theta_1^1)}(\theta_1^\times - \theta_1^1) \tag{4.5.2}$$

$$+ \frac{\log \mathbb{P}(\theta_1^1, \theta_2^2) - \log \mathbb{P}(\theta_1^1, \theta_2^2)}{(\theta_2^2 - \theta_2^1)}(\theta_2^\times - \theta_2^1). \tag{4.5.3}$$

It should be mentioned here that the pivotal point for the interpolation in the above formula is $\log \mathbb{P}(\theta_1^1, \theta_2^1)$. The interpolation can also be based on $\log \mathbb{P}(\theta_1^2, \theta_2^2)$. A similar formula exists for the second option:

$$\log \tilde{\mathbb{P}}(\theta_1^*, \theta_2^*) \approx \log \mathbb{P}(\theta_1^2, \theta_2^2) - \frac{\log \mathbb{P}(\theta_1^2, \theta_2^1) - \log \mathbb{P}(\theta_1^1, \theta_2^1)}{(\theta_1^2 - \theta_1^1)}(\theta_1^2 - \theta_1^*) \tag{4.5.4}$$

$$- \frac{\log \mathbb{P}(\theta_1^1, \theta_2^2) - \log \mathbb{P}(\theta_1^1, \theta_2^2)}{(\theta_2^2 - \theta_2^1)}(\theta_2^2 - \theta_2^*). \tag{4.5.5}$$

Note however that these two formulae are not equivalent. Hence a choice of the pivotal density point becomes crucial. In thesis, we have always used Equation (4.5.3)

A section of the storage array will be shown here for one of the examples we used involving four parameters. This section required the inclusion of internal points for two parameters out of four, two grid points for each parameter. So at time t after the updates to the log-posterior are made, the point 0.6713617 needs to be added to parameter θ_1 and 2.290211 needs to be added to parameter θ_2. The section of the grid where new points would be added is shown in Table 4.3.

Table 4.3: Storage matrix at time t before internal points are added.

θ_1	θ_2	θ_3	θ_4	Approx. log-posterior
0.6704219	2.262800	0.3212924	9199.826	−1809.400
0.6723014	2.262800	0.3212924	9199.826	−1808.558
⋮	⋮	⋮	⋮	⋮
0.6704219	2.317622	0.3212924	9199.826	−1809.459
0.6723014	2.317622	0.3212924	9199.826	−1809.390

As the new points are added, the storage matrix now contains combinations of new points for the different parameters. These combinations and the interpolated log-posterior distribution value is shown in Table 4.4. The combinations in red required linear interpolation in one variable for the computation, while that in blue required interpolation in two variables. In this specific example, do note that interpolation up to four variables could be required.

Table 4.4: Storage matrix at time t after internal points are added and their log-posterior calculated using interpolation.

θ_1	θ_2	θ_3	θ_4	Approx. log-posterior
0.6704219	2.262800	0.3212924	9199.826	−1809.400
0.6713617	2.262800	0.3212924	9199.826	−1808.979
0.6723014	2.262800	0.3212924	9199.826	−1808.558
⋮	⋮	⋮	⋮	⋮
0.6704219	2.290211	0.3212924	9199.826	−1809.429
0.6713617	2.290211	0.3212924	9199.826	−1809.008
0.6723014	2.290211	0.3212924	9199.826	−1808.974
⋮	⋮	⋮	⋮	⋮
0.6704219	2.317622	0.3212924	9199.826	−1809.459
0.6713617	2.317622	0.3212924	9199.826	−1809.424
0.6723014	2.317622	0.3212924	9199.826	−1809.390

The complexity of course increases even more with higher dimensions. The complexity does not lie in the actual computation of the log-posterior value, but rather in determining all the possible combinations of points. Once all the combinations are identified and taken care of, the implementation of the linear interpolation is very straightforward. Using simulated data, we have found that the linear interpolation procedure is very fast and reasonably accurate. Also unlike other smoothing methods, constant monitoring is not required at all. This makes the linear interpolation our favored technique in this methodology.

A toy example is provided here to show the performance of the linear interpolation method in a bivariate density. A vector of two variables (X_1, X_2) follows a bivariate normal distribution. Figure 4.10 shows the plot of the log-density of (X_1, X_2) and overlayed on top of it is the log-density of the existing grid along with that of the added internal grid point for both the variables. Though the values of the interpolated log-densities at the new internal points seem to be correct, a closer look at Figure 4.11 reveals a small difference between the true log-density and the interpolated log-density at the new internal grid point of the density of X_2 conditional on $X_1 = x_1$.

4.5.3 New External Grid point

The need to add more points to the grid outside its support is another important problem. It has been explained in Section 4.5.1, that the starting grid may not be accurate. As more and more data come in, the posterior may need to change its support. In fact, the initial grid is based on some approximating method which uses only a small set of data. Such a grid is bound to be inaccurate in the sense that it can be far away from the support of the true posterior of the parameters. Once the sequential algorithm initiates with some starting-grid, one should not expect the same grid to be fixed over time with no new grid points added or deleted from it. This refinement of the grid happens as new data points

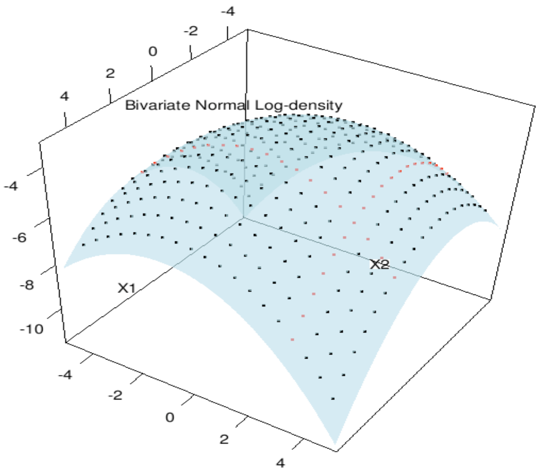


Figure 4.10: Surface plot of bivariate normal log-density. The black points visible on the blue surface reflect the log-density values of the existing grid and the red points reflect the interpolated log-density values at the newly added internal points.

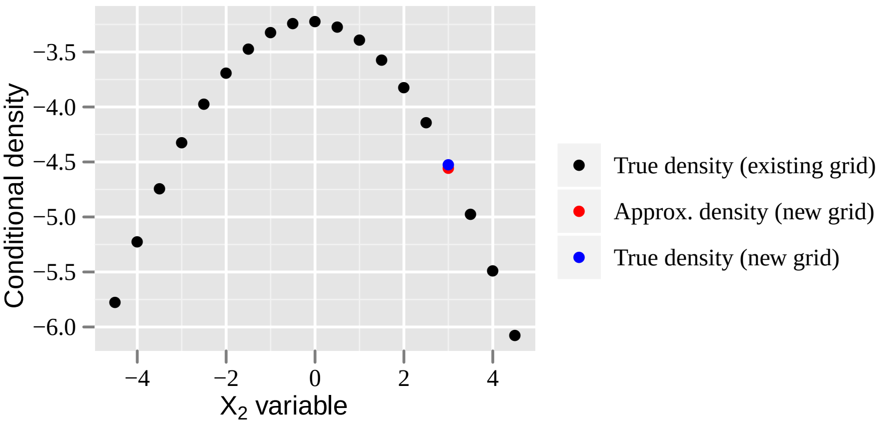


Figure 4.11: In this plot we have compared the performance of linear interpolation technique in computation of the log-density value of a new added internal point. The computed approximate log-density value at this point is compared with the true conditional log-density of $X_2 \,|\, X_1 = x_1$. Note that there is only a very small difference between the values.

are recorded, and the support of the grid moves with time towards the true support. This requires the inclusion of new grid points outside the support of existing grid points.

There is one more reason why adding a point external to the current support may be necessary. Sometimes it is required to make the grid coarse. This generally happens due to one of two cases. The first is when the approximating method for the starting grid does not extend the support of the grid to cover the tails of the posterior. It has been mentioned earlier that it will be attempted to make the initial grid wide relative to the support of the distribution. However, if this is not achieved, then the grid does not cover the whole support. The algorithm needs to add points externally. And secondly, sometimes the grid needs to be expanded in both directions. This especially happens during the starting of the sequential process, where the support of the grid shifts and changes quite frequently.

In adding an internal point, the number of points to be added for each parameter did not matter. This is because between two existing grid points in the marginal density, only one new grid point was added. Adding points external to the existing support is not that straightforward, as we will see in the rest of this section. The geometry of adding points externally is nearly the same as that of adding internal points. Allowing θ_1 and θ_2 to be two successive points right at the edge, let θ^* be an external point for which extrapolation is necessary where $\theta_1 < \theta_2 < \theta^*$. For extrapolation in one variable, Figure 4.12 shows what is being done on the grid. In Figure 4.12, it seems like the new external point is at a distance from the farthest point on the grid θ_2 which is same as that between θ_1 and θ_2. But in the actual algorithm this distance could be different. More in this will be detailed in Sections 4.5.3.1 and 4.5.3.2.

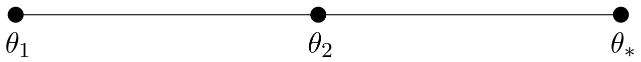


Figure 4.12: Single point added outside the support of the grid for a single variable.

For two or more parameters, where external points need to be added to more than one of them at the same time, again the issue of combination of points needs to be taken into account. This is somewhat similar to that of the addition of internal grid point, but not exactly the same. Again let θ_1 and θ_2 be the two parameters for which external grid points need to be added. Let $\theta_1^\times$ and $\theta_2^\times$ be the new points for each of the parameters respectively. The points are such that $\theta_1^2 > \theta_1^1 > \theta_1^\times$ and $\theta_2^1 < \theta_2^2 < \theta_2^\times$. This is shown in Figure 4.13.

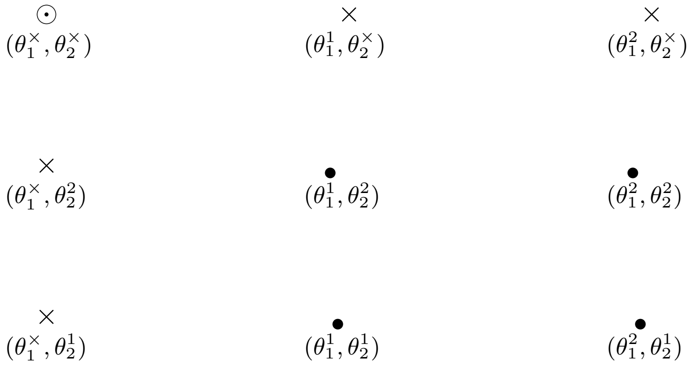

$(\theta_1^\times, \theta_2^\times)$
$(\theta_1^1, \theta_2^\times)$
$(\theta_1^2, \theta_2^\times)$
$(\theta_1^\times, \theta_2^2)$
(θ_1^1, θ_2^2)
(θ_1^2, θ_2^2)
$(\theta_1^\times, \theta_2^1)$
(θ_1^1, θ_2^1)
(θ_1^2, θ_2^1)

Figure 4.13: This plot shows all possible combinations of new external points for each of the two parameters $\theta_1^\times$ and $\theta_2^\times$ respectively. The four points for which log-posterior density is calculated using linear interpolation are $(\theta_1^\times, \theta_2^1)$, $(\theta_1^\times, \theta_2^2)$, $(\theta_1^1, \theta_2^\times)$ and $(\theta_1^2, \theta_2^\times)$. Bivariate interpolation is used on the single point $(\theta_1^\times, \theta_2^\times)$.

It is difficult to determine how many points one should add externally. Should the number of new points stretch up to the tails? This would ideally be the best solution, but then how can that be implemented? Two methods for extending the grid were tried. Both of them were based on linear extrapolation. As an example, Figure 4.2 is presented here once again so that the problem at hand can be better understood. One can quite easily figure out that the grid for the marginal posterior density needs to be extended to the left in Figure 4.14.

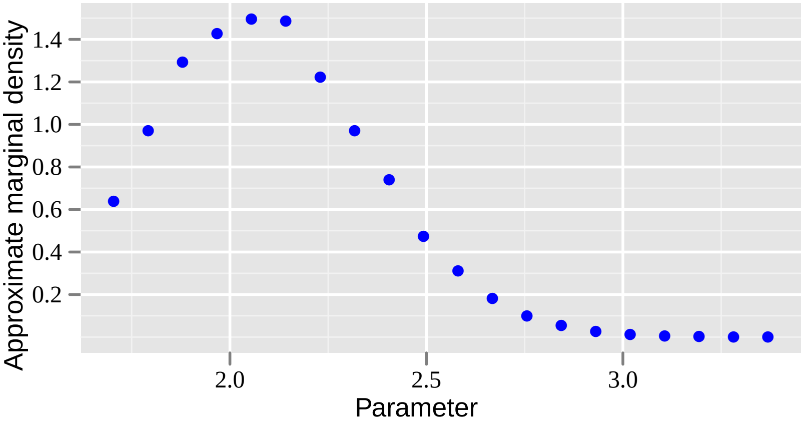


Figure 4.14: This plot is same as Figure 4.2 and is shown here for the sake of convenience in understanding Section 4.5.3.

Any decision to include a new point outside the current support of the grid is based on the approximate marginal distributions. This decision again can be based on some criterion involving the value of the marginals at the edge of the current grid. For example, in this thesis the ratio between the value of the marginal at the *approximate mode* to the value of the marginal at the edge grid point is taken as a statistic which has been used to determine the addition of a new external point. The word *approximate mode* has been used since it refers to the grid point with the highest posterior density from among the grid of points. If the ratio is greater than some pre-determined value, the decision to add new points is made as can be seen below:

$$\frac{\tilde{\mathbb{P}}(\theta_{Edge})}{\tilde{\mathbb{P}}(\theta_{Mode})} > \delta.$$

The value of δ is subjectively chosen, and that determines the frequency with which one adds points. A small value of δ means points will be added very frequently.

Already discussed curve fitting methods, such as GAM or MARS were not tried at all for extrapolation. Once the difficulties associated with them became known, there was no point in trying those algorithms in our method at all. We rather discuss two different methods that we tried to predict the posterior density for a point added outside the support of the current grid.

4.5.3.1 Method I

The first method tries to change the support of the grid to align with the support of the posterior distribution. This means new points need to be added until the grid can cover the tail of the distribution. So for a particular parameter, the basic idea here is to choose a grid point outside the edge of the existing grid such that the marginal density at that point is almost zero.

One way to do this is to join the marginal density values of the points on the edge of the grid of the marginal distribution by a straight line and allow it to intersect the x-axis. The point of intersection gives a new point which defines the edge for the new support of the grid defining the posterior. Figure 4.15 illustrates how the external grid point for this particular parameter is chosen. Once the new external point is chosen, some value of the approximate posterior density is assigned to the joint posterior density associated with that point. Let there be only one parameter, θ with $\theta^1, \cdots, \theta^n$ being the points defining the grid. Let θ^* be the new extreme point where the line joining the marginal densities of grid points at the edge intersect the axis. Though the true theoretical value one can assign to $\tilde{\mathbb{P}}(\theta^*)$ is 0, for mathematical stability a very low number, say 0.0001, is set.

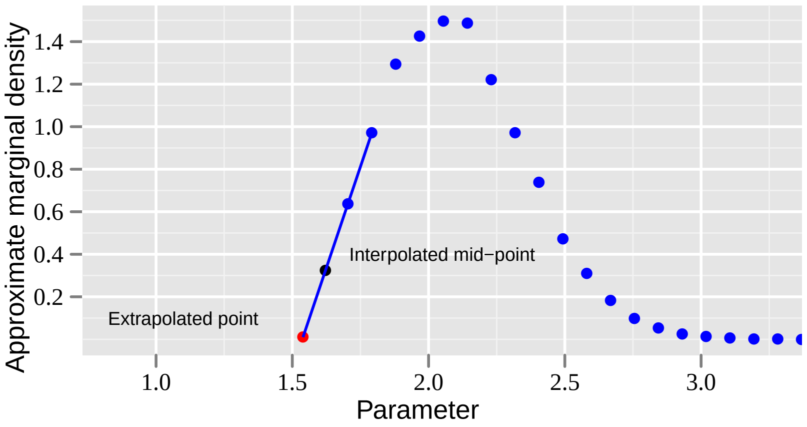


Figure 4.15: This plot shows what we are talking about in Section 4.5.3.1. The line segment passing through the two grid points at the edge intersects the x-axis. The intersection is chosen as the new edge of the grid and is assigned a very low value as approximate posterior marginal. Here the density at the mid point is also calculated via interpolation.

If the dimension of the parameter space is more than 1, the approximation can become quite crude. As before, let θ_1 and θ_2 be two parameters. Suppose, a new external point, θ_1^*, needs to be added to the grid of θ_1. Let the grid on parameter θ_2 be defined by the following set of points $\theta_2^1, \cdots, \theta_2^n$. So new approximate log-posterior values have to be set for all of the following: $\log \tilde{\mathbb{P}}(\theta_1^*, \theta_2^1), \cdots, \log \tilde{\mathbb{P}}(\theta_1^*, \theta_2^n)$. As has been done in the univariate case, a very low value is set for these approximate posterior values too. This can immediately be questioned, since the shape of the joint posterior at the tails is not exactly known. So putting the same value to the posterior of all possible combinations here would be wrong.

After the new external grid point has been selected, the gap can be filled in with more points. The number of points to fill in this gap again depends on the specific problem and also on the user. Ideally one would not choose too many points to fill this gap and rather add just one or two points. In this thesis, we have added only one extra point in between, the mid point between the edge of the existing grid and the new selected point. The internal grid algorithm will add more points in subsequent iterations if the gap between these three points are too high. Thus there is nothing rigid about this rule and is left to the users sense.

However, care is needed while applying this algorithm. One can think of a similar situation as described in Figure 4.14, but with a little variation. What happens when the line joining the last two points at the left edge never intersects the x-axis at the left, i.e. the slope of the line segment joining them is negative? Figure 4.16 gives an idea of this situation. In Figure 4.16, the line joining the marginal densities of grid points at the edge will never intersect the x-axis, creating a bug in the algorithm. Secondly, sometimes the intersection point is at a large distance from the existing support. This requires lots of internal points to be included in between. However, it is well known that adding too many grid points increases the computation time and slows down the algorithm considerably. Figure 4.17 gives an insight into such an example, where points will be repeatedly added in each subsequent iterations. As more and more internal points are added to the grid, the grid size becomes extremely large, again slowing down the algorithm.

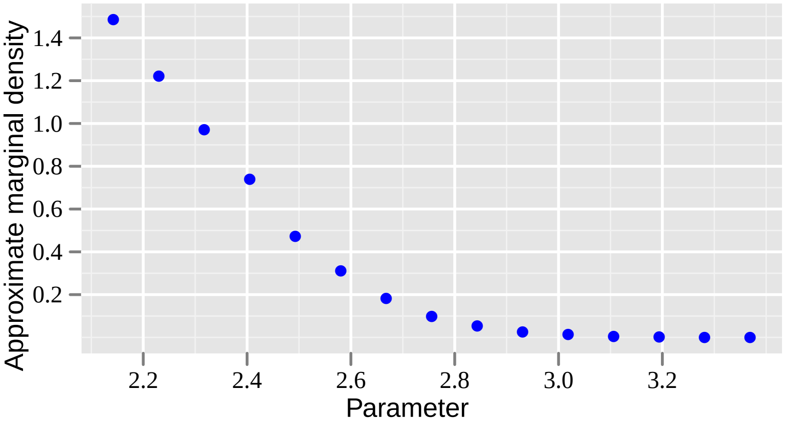


Figure 4.16: This plot represents a situation using a toy example where the true support of a parameter "moves" too much to the left and the existing grid needs to be updated to cover the true support. However it is not possible to use the method detailed in Section 4.5.3.1. In this case, the straight line joining the last two points at the edge will never intersect the x-axis. Hence the algorithm will fail.

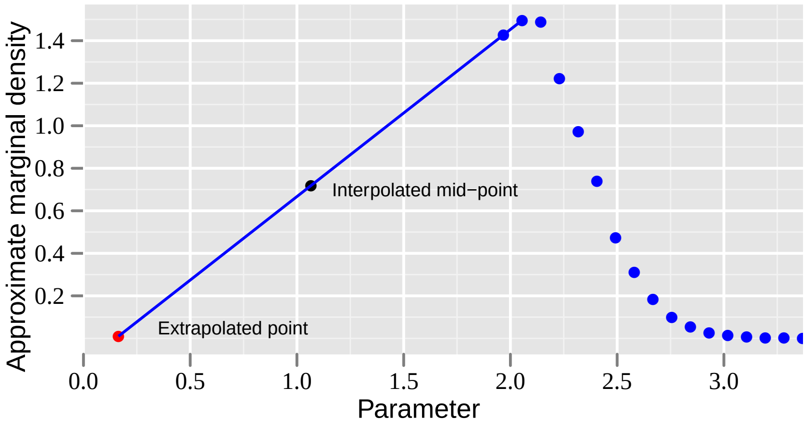


Figure 4.17: This is another situation explained in Section 4.5.3.1 where the extrapolated point is too far away from the extreme point at the left of the support and is not really representative of the real situation. There is a lot of gap between the posterior density points at the left of the grid. A lot of points will be added in subsequent iterations making the grid size very big which in turn will slow the algorithm considerably.

4.5.3.2 Method II

To get around the problems mentioned in the first method explained in the previous section, it was decided to use a simpler algorithm. It is better to use an algorithm which will not stop the iteration by producing non-feasible outcomes like the ones mentioned above. In our method, we chose to add a single external point whenever it was necessary. The extra grid point is added at a distance equivalent to the distance between the last two points on the edge of the grid defining a marginal posterior. Thus if the posterior of the parameter θ is currently defined by the grid $\theta^1, \theta^2, \cdots, \theta^n$, then the new external grid θ^* point to be added to the left of the current support of the grid is defined by:

$$\theta^* = \theta^1 - (\theta^2 - \theta^1).$$

A new external point at the right tail will similarly be defined as:

$$\theta^* = \theta^n + (\theta^n - \theta^{n-1}).$$

This way of adding small numbers of points close to the existing ones has an advantage. Firstly, this ensures that the extrapolated point is not added at distance far from the existing points, thus maintaining local accuracy to some extent. Also, unlike the previous method, we are constantly adding a maximum of two points per iteration, thus slowly increasing the burden on computation. Adding too many points in a single iteration can be costly, since there is generally a good possibility that over time they are dropped again. It has been observed in all our examples, that the support of the grid needs to add points externally at the initial phase. So for the first *few* observations new points are added outside the current support almost regularly. But then the process slows down and points are dropped because the posterior grid shrinks. This is the reason that we think it is a good idea to add points slowly and not lots of them in one single step.

The density at the external point is linearly extrapolated. The extrapolation process is almost the same as that of interpolation. Thus the approximate posterior density in one variable, as explained in the example above involving a new external point added to the left tail can be written as:

$$\log \tilde{\mathbb{P}}(\theta^*) = \log \tilde{\mathbb{P}}(\theta^1) - \frac{\log \tilde{\mathbb{P}}(\theta^2) - \log \tilde{\mathbb{P}}(\theta^1)}{\theta^2 - \theta^1}(\theta^* - \theta^1).$$

Linear extrapolation in one variable is better understood in terms of the matrix of stored parameters and the joint density. The example we choose needs points to be added to one of the parameters in a three parameter problem. Thus at time t, the posterior joint density at the grid has been stored in a matrix, after the updates are being made to the log-posterior values. Two points need to be added on either side of the grid of points defining the third parameter, say θ_3. The last two points at the edges for θ_3 are $(-12.49, -12.07)$ and $(-2.76, -2.34)$ respectively for the left and the right tail of the grid. A section of the matrix with the joint log-posterior values is shown below in Table 4.5: When the new external points are added to the grid and their log-posterior values extrapolated, the same section of the storage matrix is now shown in Table 4.6: The rows in red are the ones in which new external points are added and the log-posterior densities computed using linear extrapolation. Figure 4.18 is the marginal density of parameter θ_3 with the approximate

Table 4.5: Storage matrix at time t before external points are added.

θ_1	θ_2	θ_3	Approx. log-posterior
0.1472733	0.6930450	−12.493367	−711.0289
0.1472733	0.6930450	−12.070124	−710.9826
⋮	⋮	⋮	⋮
0.1472733	0.6930450	−2.758781	−712.1634
0.1472733	0.6930450	−2.335538	−712.5425

Table 4.6: Storage matrix at time t after external points are added and the log-posterior values at the new points are computed using extrapolation.

θ_1	θ_2	θ_3	Approx. log-posterior
0.1472733	0.6930450	−12.916610	−711.0753
0.1472733	0.6930450	−12.493367	−711.0289
0.1472733	0.6930450	−12.070124	−710.9826
⋮	⋮	⋮	⋮
0.1472733	0.6930450	−2.758781	−712.1634
0.1472733	0.6930450	−2.335538	−712.5425
0.1472733	0.6930450	−1.912295	−712.9215

marginal density of the new added points.

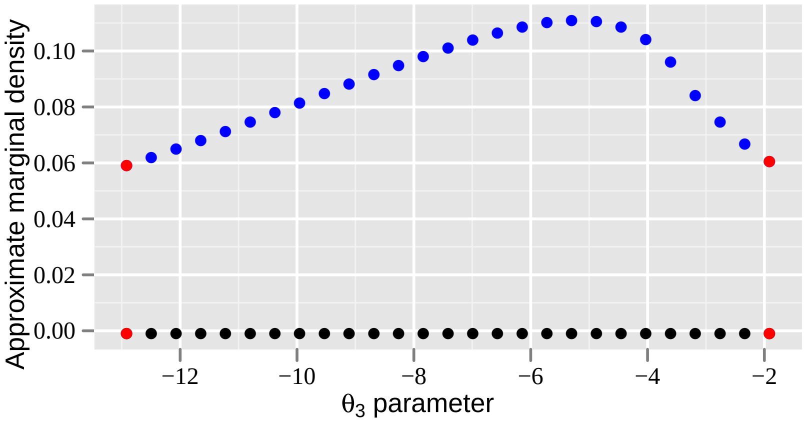


Figure 4.18: This pot shows a toy example where the support is expanding and point are added on both sides of the existing support. Linear extrapolation is used to calculate the approximate posterior marginals at these new grid points.

For higher dimensions, a similar situation to that of interpolation occurs. All possible combinations of new grid points for higher dimensions need to be considered, which means extrapolation for two variables or more is needed. A section of the grid for two variables where external points were added for both the variables has been explained geometrically earlier in Figure 4.13. Let us re-use the same graph here as Figure 4.13. Here the

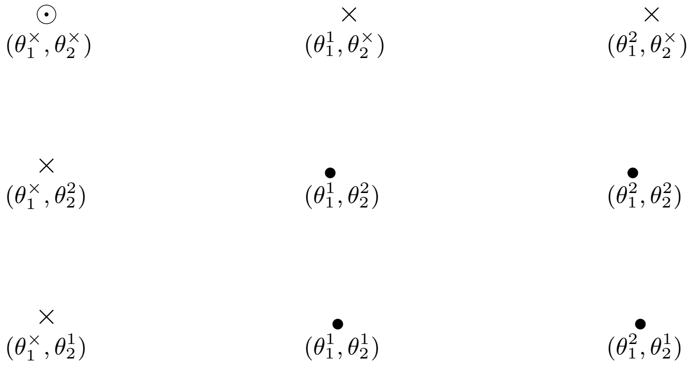

$(\theta_1^{\times}, \theta_2^{\times})$
$(\theta_1^{1}, \theta_2^{\times})$
$(\theta_1^{2}, \theta_2^{\times})$
$(\theta_1^{\times}, \theta_2^{2})$
$(\theta_1^{1}, \theta_2^{2})$
$(\theta_1^{2}, \theta_2^{2})$
$(\theta_1^{\times}, \theta_2^{1})$
$(\theta_1^{1}, \theta_2^{1})$
$(\theta_1^{2}, \theta_2^{1})$

Figure 4.19: This figure is same as Figure 4.13. This plot is provided here for the sake of clarity.

joint posterior density at the point $(\theta_1^{\times}, \theta_2^{\times})$ requires extrapolation on two variables. The approximate log-posterior density at this point is given by:

$$\begin{aligned}\log \tilde{\mathbb{P}}(\theta_1^{\times}, \theta_2^{\times}) &= \log \tilde{\mathbb{P}}(\theta_1^{1}, \theta_2^{2}) + \frac{\log \tilde{\mathbb{P}}(\theta_1^{1}, \theta_2^{2}) - \log \tilde{\mathbb{P}}(\theta_1^{1}, \theta_2^{1})}{\theta_2^{2} - \theta_2^{1}}(\theta_2^{2} - \theta_2^{\times}) \\ &+ \frac{\log \tilde{\mathbb{P}}(\theta_1^{2}, \theta_2^{2}) - \log \tilde{\mathbb{P}}(\theta_1^{1}, \theta_2^{2})}{\theta_1^{2} - \theta_1^{1}}(\theta_1^{\times} - \theta_1^{1}).\end{aligned}$$

The approximate log-posterior density of all the other combinations of points in the graph can be computed using extrapolation on one variable. So similarly as in interpolation, here too the complexity exists in determining all the possible combinations and not the computations at all. Once all the combinations are recognized, the procedure becomes quite straightforward and also is fast.

All the known disadvantages associated with techniques of the type local linearization are applicable here. With extrapolation, the possibility of an error is more, since prediction needs to be made on an unknown space outside the support. But as in all sequential techniques, a trade-off exists between accuracy and speed.

To conclude, however, our algorithm uses linear interpolation and extrapolation (Method II) for the simplicity that these methods offer. Linear interpolation gives reasonably accurate results in our simulated examples and the biggest contribution has been in terms of computation speed. There is no doubt that better methods need to be used for the computation of approximate posterior densities at new grid points, but for now we use the linear interpolation methods.

4.5.4 Combination of adding points internally and externally

This is an extension to the combination of grid points problem we have already explained. Previously, they are confined to only those points which were added internally to the support or external to the support. At a given time point it is very much possible that both happen together. So if θ_1 and θ_2 are two parameters, with an external point $\theta_1^{\times}$ added to the first parameter and an internal point θ_2^{+} added to the second parameter. Again we will be taking help of a graphical representation for a situation where we have an external point and an internal point added to a two parameter problem. It is quite confusing to compute the posterior density for the grid-point $(\theta_1^2, \theta_2^{+})$. First the log-posterior density for the rest of the points is computed using interpolation or extrapolation in one variable. For $(\theta_1^2, \theta_2^{+})$, the log-posterior of the new internal points are computed and then an extrapolation on one variable is applied to those values. To be more clear

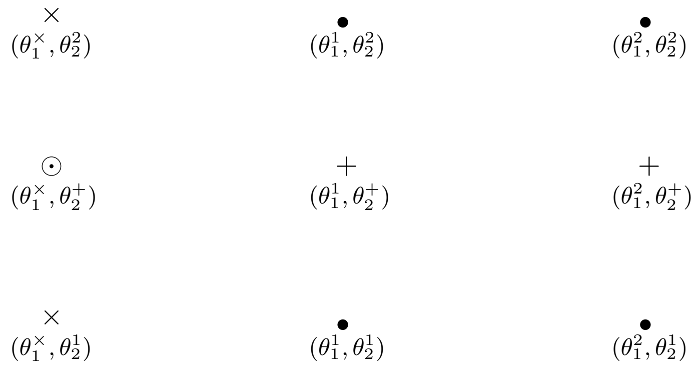

$(\theta_1^{\times}, \theta_2^2)$
(θ_1^1, θ_2^2)
(θ_1^2, θ_2^2)
$(\theta_1^{\times}, \theta_2^{+})$
$(\theta_1^1, \theta_2^{+})$
$(\theta_1^2, \theta_2^{+})$
$(\theta_1^{\times}, \theta_2^1)$
(θ_1^1, θ_2^1)
(θ_1^2, θ_2^1)

Figure 4.20: Section of the grid which shows the combination of internal and external points for two parameters.

about it, first the values of $\log \tilde{\mathbb{P}}(\theta_1^1, \theta_2^+)$ and $\log \tilde{\mathbb{P}}(\theta_1^2, \theta_2^+)$ are computed. Then these values are used for computing $\log \tilde{\mathbb{P}}(\theta_1^\times, \theta_2^+)$ using linear extrapolation. This could have also been done the other way round, i.e. by using the extrapolated log-posterior values $\log \tilde{\mathbb{P}}(\theta_1^\times, \theta_2^1)$ and $\log \tilde{\mathbb{P}}(\theta_1^\times, \theta_2^2)$ and using interpolation to get $\log \tilde{\mathbb{P}}(\theta_1^\times, \theta_2^+)$. These two different methods do not give equivalent answers for the log posterior density. We chose computing $\log \tilde{\mathbb{P}}(\theta_1^\times, \theta_2^+)$ using the interpolated values, since we believe it will have a higher level of accuracy, based on the fact that this method uses extrapolation only once rather than twice by the other method. Linear interpolation can have better accuracy than linear extrapolation, in general. However, we must also state that the difference won't be much for the two methods concerned.

For higher dimensions, this process surely gets more complex. But again, like before, the complications lie in figuring out the combinations and not the actual calculation. The modification of the grid thus stays fast and computationally easy. The user only has to make sure that calculations in all possible dimensions are taken into account. Once this has been done, the grid update is an effective and fast process.

4.5.5 Computing other statistics at the new grid points

In sequential updating algorithms, for any new point that is being added to the existing grid, calculating only the approximate posterior density at the new point does not suffice. The filtering algorithm which computes the values of the two densities, $\mathbb{P}\left(\mathbf{X}_t \,|\, \mathbf{Y}_{1:t-1}, \theta\right)$ and $\mathbb{P}\left(\mathbf{X}_t \,|\, \mathbf{Y}_{1:t}, \theta\right)$ as defined in Equation (4.3.4), almost always require to calculate different statistics, for example the first two moments. They also need to be computed for each of the grid points.

We have used the Kalman filter or the extensions of it in the examples in Chapter 6. Hence the first and second order moments need to be calculated for the new grid points. We explain the procedure through an example here. Let the filter density update requires the update of the first two moments of the filtering density. As before, let θ^* be a new grid point for some parameter θ, such that $\theta^1 < \theta^* < \theta^2$. The first two moments of $(\mathbf{X}_t \,|\, \mathbf{Y}_{1:t-1}, \theta)$ of course already exist for $\{\theta^1, \theta^2\}$. We need to calculate the terms $\mathbb{E}(\mathbf{X}_t \,|\, \mathbf{Y}_{1:t-1}, \theta^*)$ and $\mathrm{Var}(\mathbf{X}_t \,|\, \mathbf{Y}_{1:t-1}, \theta^*)$ for the new grid point. In our thesis we have again used linear interpolation to calculate the moments. The procedure is exactly same as that in the previous sections where we have computed the log-posterior density at a new grid point. The fact that better methods needs to be devised is obvious. We need to find some technique, which will provide good solutions to this problem. It is especially necessary for multivariate models.

4.6 Final Algorithm for Updating the Posterior Density of $\theta \mid \mathbf{Y}_{1:t}$

We provide the complete algorithm in this section for ease in understanding and implementation. The new algorithm is easy to understand and has a lot of scope for improvement. Adding a new point to the set of existing grid points is the real challenge. Not only there needs to be a "rule" which decides when and how new points are added, there is also the need to approximate the posterior density at the new point. New methods can be thought of to improve on the current algorithm. The existing methods are presented with full clarity in the algorithm below. The algorithm also provides flexibility to the user in choosing the methodology for calculating the prediction and the filtering densities at each time.

Algorithm 2 Sequential INLA with a check for updating current grid

```
$t = 1$
while $t < n$ do
    Observe $\mathbf{Y}_{1:t}$,
    Compute $\mathbb{P}\left(\theta \mid \mathbf{Y}_{1:t}\right)$ by INLA.
    $t = t + 1$.
end while
repeat
    Observe $\mathbf{Y}_t$,
    Compute $g_t(\theta) = \frac{\mathbb{P}(\mathbf{Y}_t \mid \mathbf{X}_t,\theta)\mathbb{P}(\mathbf{X}_t \mid \mathbf{Y}_{1:t-1},\theta)}{\mathbb{P}(\mathbf{X}_t \mid \mathbf{Y}_{1:t},\theta)}\Big|_{\mathbf{x}_t^*(\theta)}$, where $\mathbf{x}_t^*(\theta)$ is any value of $\mathbf{X}_t$ for
    which the denominator is positive,
    Compute $g_t(\theta)$ by sequential method, for all $\theta$ on the grid,
    Update $\mathbb{P}(\theta \mid \mathbf{Y}_{1:t}) = \mathbb{P}(\theta \mid \mathbf{Y}_{1:t-1}) \times g_t(\theta)$, for all $\theta$ on grid,
    if $\mathbb{P}(\theta_i \mid \mathbf{Y}_{1:t}) < \delta$ for each grid point, where $\delta$ is a very small number then
        Drop all those grid points such that they are not to be used anymore,
    end if
    if $\mathbb{P}(\theta_i^k \mid \mathbf{Y}_{1:t}) - \mathbb{P}(\theta_i^l \mid \mathbf{Y}_{1:t}) > \delta_1$ where $\theta_i^k$ and $\theta_i^l$ are adjacent grid points and $\delta_1$ is
    some user specific value then
        Add a grid point between $\theta_i^k$ and $\theta_i^l$ and compute the joint posterior density at that
        point using linear interpolation based on $\mathbb{P}(\theta_i^k \mid \mathbf{Y}_{1:t})$ and $\mathbb{P}(\theta_i^l \mid \mathbf{Y}_{1:t})$,
    end if
    if $\mathbb{P}(\theta_i^{Edge} \mid \mathbf{Y}_{1:t})/\mathbb{P}(\theta_i^{Mode} \mid \mathbf{Y}_{1:t}) > \delta_2$ where $\theta_i^{Edge}$ denote the last grid point at the
    tail while $\theta_i^{Mode}$ is the grid with highest density, and $\delta_2$ is an user specific value then
        Add a grid point more extreme to that of $\theta_i^{Edge}$ and calculate the joint posterior at
        that using linear extrapolation using the density at the last two grid points on the
        tails.
    end if
    Figure out all possible combinations of grid points in higher dimensions when multiple
    grid points need to be added across dimensions and then compute the joint posterior
    at those points.
    $t = t + 1$
until $t = t_{end}$
```

4.7 Conclusion

In this chapter, we have explained in detail the most important section of our work in this Ph.D. thesis. The sequential algorithm is presented first along with other methods that were treid out but did not work out well. Since Sequential INLA is a grid based method, the construction of the initial grid has also been discussed in detail. Given that we are working on real time inference requiring dynamic updates, a very important part of this work is to dynamically update the grid. We have discussed different methods along with toy examples to explain the difficulties associated with them and also justify our choice over existing methods. This new method have been tried on data sets simulated from different models. In certain casesit was seen that the result did not turn out to be satisfactory. This issue will be looked into and explained in the next chapter and possible solutions will also be provided.

Chapter 5

Correction to Improve Posterior Density of the Parameters

5.1 Problem encountered in the current method

In certain special situations the sequential updating scheme explained in Chapter 4 perform poorly. We have applied our algorithm on data simulated from different models each of which vary in linearity and Gaussianity assumptions. While the results have mostly been satisfactory, in certain special cases, such as strong nonlinearity, the sequential algorithm is found lacking. In the following section, we will be discussing the problem in greater detail, explaining why the method fails.

5.1.1 Explanation of problem

One problem that has been encountered in the proposed sequential method has been what we have called a "collapse" of the density to only a few grid points. And this phenomenon happens in one or two time steps. Suppose at time t, the posterior density of $\theta \mid \mathbf{y}_{1:t}$ is *well defined*, for some univariate parameter θ. By the phrase *well defined* we mean that each of the grid points have significant posterior density values. However, after one or two time points, i.e. as the grid is updated after data points $\mathbf{y}_{1:t+1}$ and $\mathbf{y}_{1:t+2}$are recorded, the posterior density is significant at only a few grid points while the rest have very low density values. Note from Chapter 4 that monitoring of our sequential algorithm is done through the marginal density of each of the parameters, when the dimension of θ is greater than 1. For higher dimensions, it has been found in certain cases that the whole grid defining the marginal density has negligible density values except for one or two grid points. This complication seems to be somewhat similar to the degeneracy problem of the particle filters. The interested reader should look in Arulampalam et al. (2002) for an introduction to the degeneracy problem in particle filters. The example below illustrates the problem.

Suppose data has been generated from the following 4 parameter model,

$$y_t = \theta_1 x_t^2 + v_t, \tag{5.1.1}$$

$$x_{t+1} = 4 + sin(\omega \pi t) + \theta_2 x_t + w_t. \tag{5.1.2}$$

Here $v_t \sim \mathcal{N}(0, \theta_3)$, $w_t \sim \mathcal{N}(0, \theta_4)$ and ω is assumed to be known. The vector of unknown parameters is given by the $\theta \equiv (\theta_1, \theta_2, \theta_3, \theta_4)$. As always, we are interested in the posterior of $\theta \mid \mathbf{y}_{1:t}$ for some time t. In this example, the posterior for $\theta \mid \mathbf{y}_{1:t}$ has been computed for some value of $t = T$, say. Figure 5.1 shows the approximate marginal density at time $t = T$ for one of the 4 parameters. At time $T + 1$, i.e. after a single iteration, the posterior marginal density for the same parameter is concentrated on single point while the other grid points have negligible posterior values. Figure 5.2 shows the situation we have just discussed involving a certain parameter. However, this situation does not happen for any data simulated from the above mentioned nonlinear model. We have simulated several data sets from the model, and the problem happens only in certain "special" cases. This problem thus narrows down to identifying the pattern for which a collapse of the grid is happening, causing the algorithm to come to a complete halt.

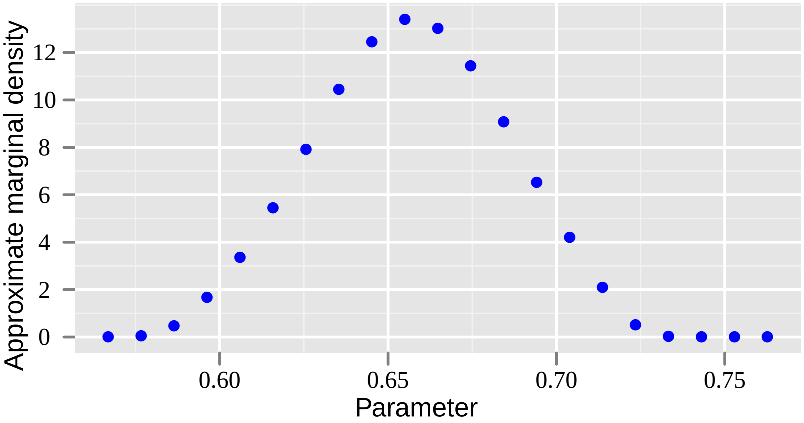


Figure 5.1: Grid of posterior marginal for parameter θ_3 at time T. It can be seen that nothing seems wrong with the shape of the posterior density until this time point. Figure 5.1 however points out to a problem.

Looking at the marginal density in Figure 5.2 one may think that this problem can possibly be avoided if there is a better methodology to adjust the support of the true posterior density, since the current dynamic grid approach does not work here. As per the current adaptive grid process, when grid points are dropped, new grid points should be added internally and/or externally and the grid should adopt to cover the true support of the posterior. However, even though expected, that is not always possible. In cases such as the one described above, except for one or two points, all other points in the grid are

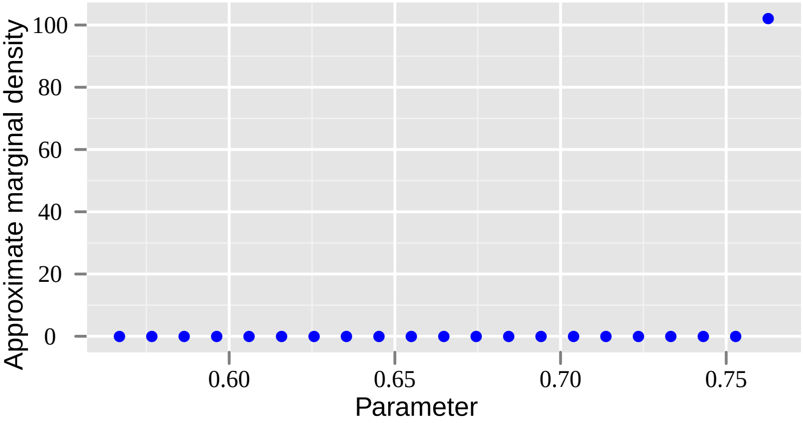


Figure 5.2: Plot showing marginal posterior of the same parameter θ_3 at time $T+1$. Note the concentration of marginal posterior on a single grid point. This collapse of the posterior density in a single update has been examined in this chapter.

dropped. Thus the posterior is defined by only a very few points (mostly only a single point), and the algorithm crashes. The current dynamic grid methodology cannot rectify the situation when there is only one grid point left. In our example, the algorithm at the start had to adjust the grid for support at every successive time point. The change in support was very frequent and quite extreme, which prompted the algorithm to stop.

A very interesting example is provided by Figure 5.3, which shows the performance of the algorithm for a different data-set. Note the change that occurred in the support of the marginal density from time t to $t+1$ for some value of t. The grid points that represented the region of high posterior density in the plot titled “At time 7”, have very low posterior values at the very next update, shown in the other plot. The grid points at the right tail of the posterior in the figure at top seems to represent the region of high posterior density in the figure situated below. Several grid points situated at the left tail of the posterior has been dropped completely. This example represents the abrupt shift in support of posterior and shows the amount of flexibility required by our algorithm to deal with such cases. For a situation like this, the grid updating will work perfectly fine and the algorithm will continue to perform. But for situations such as in Figure 5.2, the grid update will fail completely, and the algorithm has to stop. One can see from Figure 5.3 how much the support needs to be shifted.

5.1.2 Cause of the problem

No literature was found which could explain this problem. We tried to list down all possible cases which can justify the defined problem. One possible reason is that for the filtering problem we used UKF. UKF requires that certain tuning parameters need to be set. One of the tuning parameters (α) govern how much non-local effects are captured

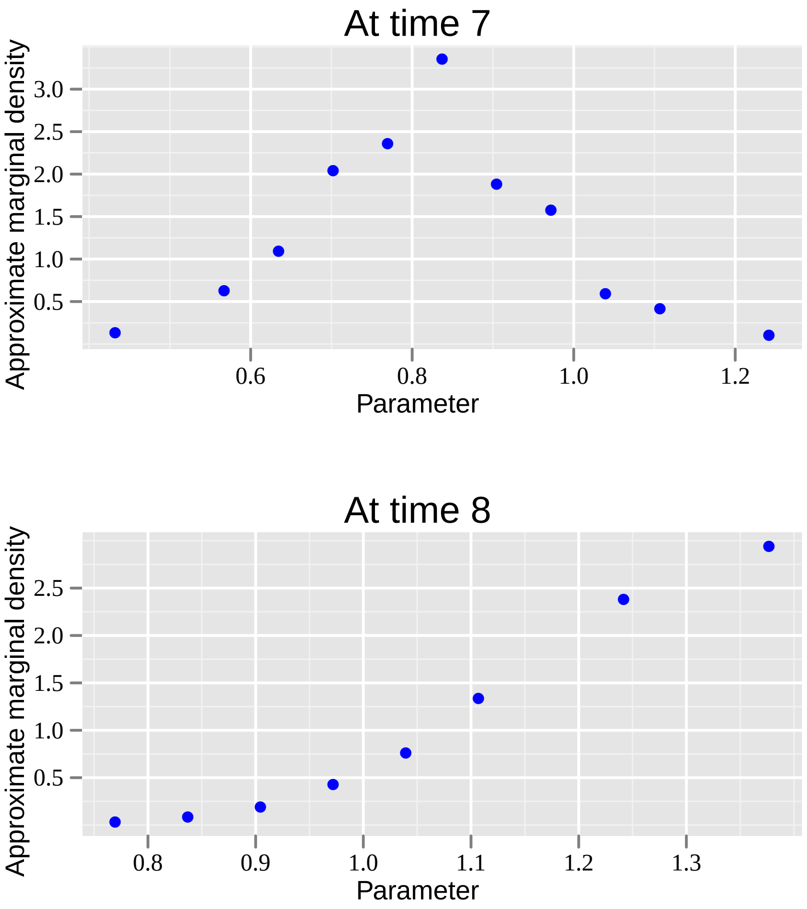


Figure 5.3: Change in support of the marginal density of some parameter of a toy example in a single change in time point from T to $T + 1$ for some T. The scale of the x-axes in the two graphs are different because certain grid points were dropped from the first plot and hence the existing support has shrunk. This is a situation where the grid updating algorithm will work perfectly and the algorithm will not be stopped.

by the sigma points. One should refer to Section 2.4.2.2 for a recap. Since the model in question is highly nonlinear, we tried different values for the tuning parameters, especially varying the value of α, but they did not have any effect on the problem at hand. Hence it was concluded that the tuning parameters in UKF cannot solve the collapse of the grid.

Since the collapse happened only for certain data sets, we decided to study the values of the data set at the time point of collapse. The value of the observation at that specific time point can be considered to be an outlier, since it was quite large when compared to the values before and after it. The likelihood, $\mathbb{P}(y_t \mid x_t, \theta)$, at that point have a very low value. In Figure 5.4 the likelihood density at time t is plotted with the observation y_t on the same plotting area, which gives us a good idea about the problem. It is easily seen that the likelihood value is very small at the data point. This in turn results in the

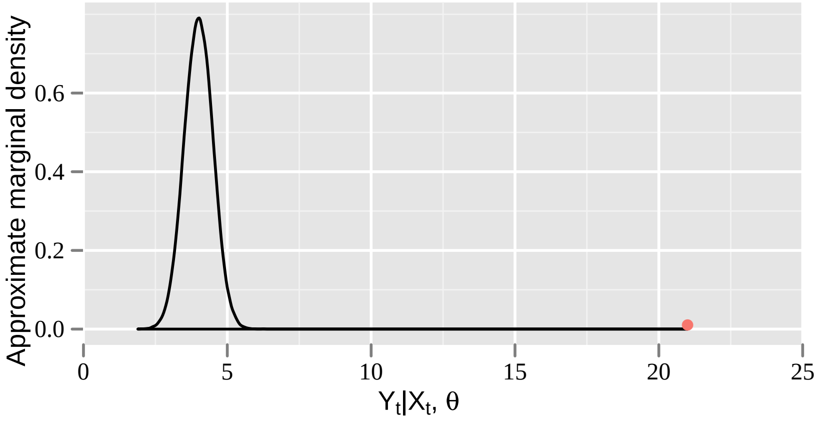


Figure 5.4: Plot of the exact likelihood computed at time $T+1$ for a given mode of the filtering density and a given grid value of the posterior of θ in the 4 parameter model discussed in this chapter. The value of the new data point, y_{T+1} has also been plotted on the same plotting area, to show its position with respect to the exact likelihood. One can see that it is located far into the tails, making the value of $\mathbb{P}(y_t \mid x_t, \theta)$ very close to 0.

sequential multiplier $g_t(\theta)$ as has been defined by:

$$g_t(\theta) = \frac{\mathbb{P}\left(\mathbf{Y}_t \mid \mathbf{X}_t, \theta\right) \mathbb{P}\left(\mathbf{X}_t \mid \mathbf{Y}_{1:t-1}, \theta\right)}{\mathbb{P}\left(\mathbf{X}_t \mid \mathbf{Y}_{1:t}, \theta\right)},$$

to have a small value. We also found the values of $\mathbb{P}\left(\mathbf{X}_t \mid \mathbf{Y}_{1:t-1}, \theta\right)$ and $\mathbb{P}\left(\mathbf{X}_t \mid \mathbf{Y}_{1:t}, \theta\right)$ are generally not extreme, in the sense of being very big or small.

Once the problem has been identified , some way to fix this problem is necessary. We took help from a recent paper by Cseke & Heskes (2010), which ideally aimed at improving inferences from the INLA, but have been found to solve our problem. We start off by explaining their method, followed by how it has been used to solve this problem.

5.2 Solution to this problem/Dealing with outliers

In this section we first explain the method proposed by Cseke & Heskes (2010) in their recent paper. Further on, we discuss the implementation of their method in view of the sequential setting.

5.2.1 Explaining Cseke and Heske's method

An improvement for the marginal posterior density calculation of $\mathbb{P}\left(\mathbf{X}_t \mid \mathbf{Y}_{1:t}, \theta\right)$ in INLA was provided by a recent paper by Cseke & Heskes (2010). One of the goals that INLA seeks to address is to approximate the marginal distribution given as,

$$\mathbb{P}\left(\mathbf{X}_i \mid \mathbf{Y}_{1:t}, \theta\right) \propto \mathbb{P}\left(\mathbf{Y}_i \mid \mathbf{X}_i, \theta\right) \int \mathbb{P}\left(\mathbf{X}_{1:t} \mid \theta\right) \prod_{j \neq i} \mathbb{P}\left(\mathbf{Y}_j \mid \mathbf{X}_j, \theta\right)\ d\mathbf{X}_{-i}, \tag{5.2.1}$$

where $\mathbf{X}_{-i} \equiv (\mathbf{X}_1, \ldots, \mathbf{X}_{i-1}, \mathbf{X}_{i+1}, \ldots, \mathbf{X}_n)$. Their paper seeks to improve on the approximation of $\mathbb{P}\left(\mathbf{X}_i \mid \mathbf{Y}_{1:t}, \theta\right)$.

A global approximation to the joint density $\mathbb{P}\left(\mathbf{X}_{1:t} \mid \mathbf{Y}_{1:t}, \theta\right)$ can be provided by different methods like a Gaussian approximation or Expectation - Propagation. First let us explain the terminologies for defining the posterior of the latent process. First let us recall the way we have expressed the joint posterior, $\mathbb{P}\left(\mathbf{X}_{1:t} \mid \mathbf{Y}_{1:t}, \theta\right)$, in Equation (2.5.7) of Chapter 2:

$$\mathbb{P}(\mathbf{x}_{1:t} \mid \mathbf{y}_{1:t}) = \frac{1}{Z_q} \mathbb{P}(\mathbf{x}_{1:t}) \prod_{i=1}^{t} t_i(\mathbf{x}_i, \mathbf{y}_i). \tag{5.2.2}$$

Let us assume that we have an approximation to the posterior in Equation (5.2.2) as:

$$\mathbb{P}\left(\mathbf{X}_{1:t} \mid \mathbf{Y}_{1:t}, \theta\right) \approx \mathbb{P}\left(\mathbf{X}_{1:t} \mid \theta\right) \prod_{i=1}^{t} \tilde{t}_i(\mathbf{X}_i, \mathbf{Y}_i, \theta), \tag{5.2.3}$$

where $\tilde{t}_i(\mathbf{X}_i, \mathbf{Y}_i, \theta)$ is some approximating function dependent on $\mathbf{X}_i$, $\mathbf{Y}_i$ and θ. The form of $\tilde{t}(\mathbf{X}_i, \mathbf{Y}_i, \theta)$ varies according as what method (Expectation-Propagation, Gaussian approximation etc.) is used.

Now this approximation can be defined quite easily in the case of Gaussian approximation which is used in INLA for computing $\mathbb{P}\left(\mathbf{X}_{1:t} \mid \mathbf{Y}_{1:t}, \theta\right)$. This approximation is based on the fact that a Taylor series approximation is done to the (log) likelihood and all terms after the second order are ignored. We present the representation in Equation 5.2.3 for computation of $\tilde{t}_i(\mathbf{X}_i, \mathbf{Y}_i, \theta)$ in Gaussian approximation:

$$\begin{aligned}
\mathbb{P}\left(\mathbf{Y}_t \mid \mathbf{X}_t, \theta\right) &= \exp(\log\left\{\mathbb{P}\left(\mathbf{Y}_t \mid \mathbf{X}_t, \theta\right)\right\}) \\
&\approx \exp\left(a_t + b_t x_t - \frac{1}{2} c_t x_t^2\right), \qquad (5.2.4) \\
&\propto \exp\left\{-\frac{1}{2} c_t \left(x_t - \frac{b_t}{c_t}\right)^2\right\}, \\
&= \mathcal{N}\left(\frac{b_t}{c_t}, \frac{1}{c_t}\right), \\
&= \tilde{t}_t\left(\mathbf{X}_t, \mathbf{Y}_t, \theta\right), \qquad (5.2.5)
\end{aligned}$$

where the constants a_t, b_t and c_t in Equation (5.2.4) are defined as follows. Denoting $\mathbb{P}\left(\mathbf{Y}_t \mid \mathbf{X}_t, \theta\right)$ as $f(\mathbf{X}_t)$, we have $b_t = f^{'}(\mathbf{X}_t^*) - f^{''}(\mathbf{X}_t^*)$ and $c_t = -f^{''}(\mathbf{X}_t^*)$. It easily follows from Equation (5.2.5) that the approximate likelihood can be written as a product of $\tilde{t}(\mathbf{X}_i, \mathbf{Y}_i, \theta), \quad \forall\, i$. So it can be seen that the Gaussian approximation can be represented in the form of Equation (5.2.3) which in turn uses Equation 5.2.5:

$$\mathbb{P}\left(\mathbf{X}_{1:t} \mid \mathbf{Y}_{1:t}, \theta\right) \propto \mathbb{P}\left(\mathbf{X}_{1:t} \mid \theta\right) \prod_{1}^{t} \mathcal{N}\left(\frac{b_i}{c_i}, \frac{1}{c_i}\right).$$

We denote the approximation in Equation (5.2.3) as $q(\mathbf{X}_{1:t})$,

$$q(\mathbf{X}_{1:t}) \propto \mathbb{P}\left(\mathbf{X}_{1:t} \mid \theta\right) \prod_{1}^{t} \tilde{t}_i\left(\mathbf{X}_i, \mathbf{Y}_i, \theta\right). \tag{5.2.6}$$

Using these approximations, Cseke and Heske have proposed the initial improvement:

$$\begin{aligned}
\mathbb{P}\left(\mathbf{X}_i \mid \mathbf{Y}_{1:t}\theta\right) &\propto \mathbb{P}\left(\mathbf{Y}_i \mid \mathbf{X}_i, \theta\right) \int \mathbb{P}\left(\mathbf{X}_{1:t} \mid \theta\right) \prod_{j \neq i} \mathbb{P}\left(\mathbf{Y}_j \mid \mathbf{X}_j, \theta\right) \, d\mathbf{X}_{-i} \\
&\propto \frac{\mathbb{P}\left(\mathbf{Y}_i \mid \mathbf{X}_i, \theta\right)}{\tilde{t}_i\left(\mathbf{X}_i, \mathbf{Y}_i, \theta\right)} \int q\left(\mathbf{X}_{1:t}\right) \prod_{j \neq i} \frac{\mathbb{P}\left(\mathbf{Y}_j \mid \mathbf{X}_j, \theta\right)}{\tilde{t}_j\left(\mathbf{X}_j, \mathbf{Y}_j, \theta\right)} \, d\mathbf{X}_{-i} \text{ using Equation (5.2.6)} \\
&\propto \frac{\mathbb{P}\left(\mathbf{Y}_i \mid \mathbf{X}_i, \theta\right)}{\tilde{t}_i\left(\mathbf{X}_i, \mathbf{Y}_i, \theta\right)} q\left(\mathbf{X}_i\right) \int q\left(\mathbf{X}_{-i} \mid \mathbf{X}_i\right) \prod_{j \neq i} \frac{\mathbb{P}\left(\mathbf{Y}_j \mid \mathbf{X}_j, \theta\right)}{\tilde{t}_j\left(\mathbf{X}_j, \mathbf{Y}_j, \theta\right)} \, d\mathbf{X}_{-i} \\
&\approx q\left(\mathbf{X}_i\right) \frac{\mathbb{P}\left(\mathbf{Y}_i \mid \mathbf{X}_i, \theta\right)}{\tilde{t}_i\left(\mathbf{X}_i, \mathbf{Y}_i, \theta\right)} \qquad (5.2.7)
\end{aligned}$$

The expression $q\left(\mathbf{X}_i\right)$ at the last step is the marginal distribution of the Gaussian approximation to the posterior density as explained earlier in Equation (5.2.6) and can be derived by using Equation (5.2.1) or otherwise. The authors have improved upon this approximation in their paper, but we will be using only this basic improvement for its simplicity. The implementation is explained in the following section.

5.2.2 Application of Cseke and Heske's method in our algorithm

Further approximation needs to be done when applying this "correction" in our sequential inferential setting. This is because at each time point t, an approximation for the full joint posterior distribution cannot be calculated, since that would mean storing all the data which in turn contradicts the purpose of a sequential estimator. In other words, $q\left(\mathbf{X}_t\right)$ is not derived as the marginal of a joint approximation $\mathbb{P}\left(\mathbf{X}_{1:t} \mid \mathbf{Y}_{1:t}, \theta\right)$ once INLA is no longer applicable. Thus in a sequential setting at time t, we replace $q\left(\mathbf{X}_t\right)$ by the approximation to $\mathbb{P}\left(\mathbf{X}_t \mid \mathbf{Y}_{1:t}, \theta\right)$ provided by some filtering algorithm, say UKF (or EKF). UKF (or EKF) can also provide a formal expression for $\tilde{t}_i\left(\mathbf{X}_t, \mathbf{Y}_t, \theta\right)$ since they assume normality for the likelihood and provides the mean and variance of the approximating Gaussian distribution. In our setting an approximate corrected posterior of $\mathbf{X}_t \mid \mathbf{Y}_{1:t}, \theta$ is given by:

$$\mathbb{P}_{\mathit{Corrected}}\left(\mathbf{X}_t \mid \mathbf{Y}_{1:t}, \theta\right) \approx \mathbb{P}_{\mathit{EKF/UKF}}\left(\mathbf{X}_t \mid \mathbf{Y}_{1:t}\theta\right) \frac{\mathbb{P}\left(\mathbf{Y}_i \mid \mathbf{X}_i, \theta\right)}{\tilde{t}_i\left(\mathbf{X}_i, \mathbf{Y}_i, \theta\right)}. \tag{5.2.8}$$

We propose that the denominator in Equation (5.2.8) be calculated using any of the filtering algorithms. An approximation to the density of $\mathbf{Y}_t \mid \mathbf{X}_t, \theta$ can still be provided by the EKF/UKF assuming a Gaussian distribution for it. The filtering algorithm can provide with a mean and a variance to the normal density approximating $\mathbf{Y}_t \mid \mathbf{X}_t, \theta$. Thus whatever the true distribution of $\mathbf{Y}_t \mid \mathbf{X}_t, \theta$ is, we can have:

$$\tilde{t}_i\left(\mathbf{X}_i, \mathbf{Y}_i, \theta\right) = \mathbb{P}_G\left(\mathbf{Y}_i \mid \mathbf{X}_i, \theta\right).$$

We propose a $\mathbb{P}_G\left(\mathbf{Y}_i \mid \mathbf{X}_i, \theta\right)$ whose mean and variance are computed by propagating $\mathbb{P}_{EKF/UKF}\left(\mathbf{X}_t \mid \mathbf{Y}_{1:t}\theta\right)$ through the observation equation $f(\cdot)$ in Equation (2.3.6) of the state-space representation in Chapter 2. This retains the nonlinearity between $\mathbf{X}_t$ and $\mathbf{Y}_t$. This is done by selecting sigma points $\mathcal{X}_1, \ldots, \mathcal{X}_n$ from $\mathbb{P}_{EKF/UKF}\left(\mathbf{X}_t \mid \mathbf{Y}_{1:t}\theta\right)$, and letting the mean and variance of $\mathbb{P}_G\left(\mathbf{Y}_i \mid \mathbf{X}_i, \theta\right)$ be the mean and variance of $f(\mathcal{X}_1, \theta), \ldots, f(\mathcal{X}_n, \theta)$. This must be repeated for each grid point in the posterior of θ.

5.3 Implementation

The implementation of the correction based on Cseke and Heske's method is applied on the 4 parameter problem given by Equations (5.1.1) and (5.1.2), for a simulated data set in which the collapsing problem was present. One can see that the variance parameter in the observation equation is unknown. In other words, one can say that the parameter(s) controlling the variability of the likelihood is unknown. This would mean that the exact likelihood will be computed based on the grid-values of the parameter. By our method, the marginal posterior of the variance parameter is defined on a grid. Hence the exact likelihood $\mathbb{P}(\mathbf{Y}_t \mid \mathbf{X}_t, \theta)$ is calculated based on grid-values of the parameter θ_3. The marginal posterior density of $\mathbb{P}(\theta_3 \mid \mathbf{Y}_{1:T})$ is shown in Figure 5.5 for some time, T. This plot of the marginal posterior is same as that of Figure 5.1 and is shown here again for clarity. In time $T+1$, a new observation y_{T+1} is recorded and the posterior density of the parameters is updated. It can be seen in Figure 5.6 that the whole density is concentrated on only one grid point. Again, as for Figure 5.5, this plot is identical to Figure 5.2. The algorithm while using the adaptive grid procedure, will drop all the grid points except for one in this case. Once the grid update process is implemented, it is absolutely not possible to add points internally or externally. Hence the algorithm does not proceed any longer, and the sequential updates are stopped.

We applied the correction, as discussed in Section 5.2.2 to this problem, once computation of the posterior using INLA shifted to our sequential updating methodology. It was seen that the collapse of the posterior density does not happen, after the application of the correction. The situation is being averted and the algorithm can work without any problem now. Figure 5.7 shows the posterior density of the parameter at time $T+1$, once

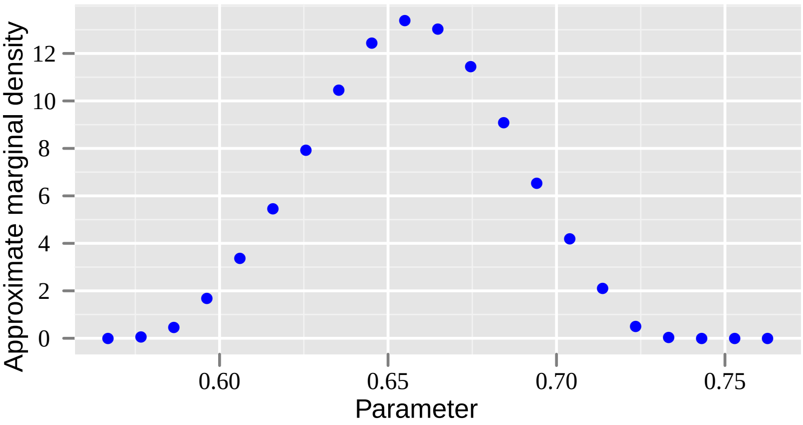


Figure 5.5: The posterior distribution of $\mathbb{P}(\theta \mid \mathbf{Y}_{1:T})$. This plot is an exact replica of Figure 5.1.

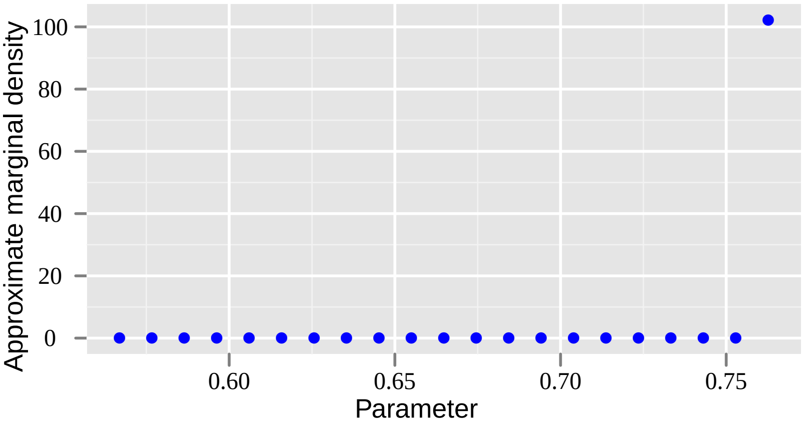


Figure 5.6: The collapse of the posterior density of $\mathbb{P}(\theta \mid Y_{1:T})$ into a single grid point. Again this figure is same as that in Figure 5.2.

the application of the correction to the algorithm is in place.

What has been noticed here is that because of the correction, the Gaussian approximation to the likelihood has higher variability than the exact likelihood, resulting in the approximate likelihood value of $\mathbb{P}(Y_{T+1} = y_{T+1} \mid X_{T+1} = x^*_{T+1}, \theta = \theta^*)$ to be non trivial. Here x^*_t is the mode of the filtering density at $T + 1$ and θ^* is a particular value of the posterior grid. The sequential multiplier $g_t(\theta)$ is not close to 0 anymore, and the collapsing problem is avoided.

It must be mentioned that even though the problem has been avoided, a thorough examination of this problem is necessary. This correction has been applied to several data-sets simulated from the same model as discussed in this chapter. However, it needs to be understood that this correction factor is not always necessary. For models with strong nonlinearity, the correction factor should be used, more as a precautionary measure than anything else. We do not implement it in all applications because of the extra computation

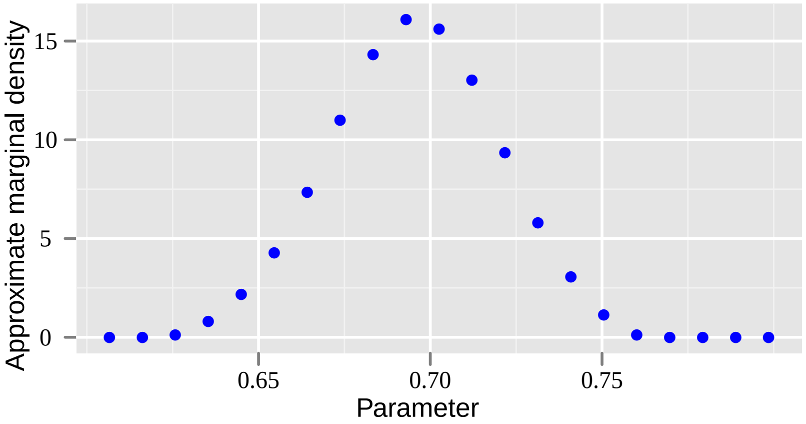


Figure 5.7: This plot shows the posterior of $\mathbb{P}(\theta \mid Y_{1:T+1})$ after the correction factor has been applied in the sequential update at every time point. The correction factor results in a non-zero value of the sequential multiplier term, which in turn stops the collapse to happen.

required in computing the approximate likelihood. Hence it has not been implemented for the other models that have been used in this thesis.

5.4 Conclusion

This chapter concludes the development of our new method as far as this thesis is concerned. This chapter coupled with Chapter 4 explains the algorithm in its completeness. The sequential method can now be implemented in different applications, where the posterior of the static parameters is constructed on a dynamic grid, assuming that the assumptions in Chapter 4 are met. In case of a situation where the model has strong nonlinearities, a correction factor has been provided which though slowing the algorithm, stops it from a degeneracy like phenomenon. In the next chapter we will be implementing our methodology on different data sets simulated from different models. The performance of our method has been compared with existing algorithms like article filter, in terms of speed and accuracy.

Chapter 6

Application of Sequential Algorithm to Simulated Examples

6.1 Goal of the Application

The novel work of this thesis, as introduced and explained in Chapters 4 and 5 is now applied to three distinctly different models, namely linear, nonlinear and non-Gaussian. Fundamentally in sequential estimation state space models can be divided into two broad groups, linear Gaussian and nonlinear and/or non-Gaussian. For linear Gaussian models, there exist exact solutions for the filtering density and the 1-step ahead prior density (Meinhold & Singpurwalla 1983). Hence, a linear model with additive Gaussian error is the best starting point for a first example. The availability of exact computations makes it a good choice to test the algorithm.

Nonlinear and/or non-Gaussian models require approximations for computation of the filtering and prediction density, and they are more rigorous in nature. We use these models in rigorous testing for our algorithm. These models are also used to test the grid updating process. This is mainly because of the fact that the support of the posterior distribution will be expected to change a lot under nonlinearity and non-Gaussian density conditions, once INLA provides the initial grid which depends only on the first few data points. Another important point which needs to be checked through these examples is the trade off between accuracy and speed. Our algorithm are tested against already existing algorithms like SMC and INLA on the basis of accuracy and computation speed.

The results of application of the sequential algorithm over all these models are presented in the following sections. To start with, in Section 6.2 we will explain some procedures which have been used to diagnose the performance of the filter and compare it with other existing methods. Section 6.4 shows the application of the algorithm on multivariate data simulated from a linear Gaussian model. Data are simulated from a nonlinear model and the algorithm is used to estimate the parameters in Section 6.5. Finally the algorithm is applied on data simulated from a model with non-Gaussian observation variable in Section 6.6.

6.2 Performance Measures

The performance of the filter has to be measured by appropriate diagnostics. To start with, some measure of central tendency is generally calculated at each time point in any filtering algorithm. The mean or the mode is normally computed in filtering algorithms (Arulampalam et al. 2002). On the other hand, the accuracy of an estimate of some central tendency computed from the filter is evaluated using statistics such as the *Mean Squared Error* (MSE) or the *Root Mean Squared Error* (RMSE). The MSE of an estimator $\hat{\theta}$ for some parameter θ is defined as

$$\text{MSE}(\hat{\theta}) = \mathbb{E}(\hat{\theta} - \theta)^2.$$

The square root of MSE yields the RMSE. However, the MSE (or RMSE) is applicable for univariate parameters. For higher dimensions one can compute a MSE matrix. The MSE matrix is given by the following formula:

$$\text{MSE}(\hat{\Theta}) = \mathbb{E}\left[(\hat{\Theta} - \Theta)^T(\hat{\Theta} - \Theta)\right],$$

where $\hat{\Theta}$ and Θ are vectors.

In our examples, we have provided an estimate of the posterior mode. We have also made use of graphical procedures, like plotting the estimated mode and probability bounds over time to evaluate the performance of the algorithm. This will be explained in greater detail in the next section. Evaluating the accuracy of the estimates is a bit more tricky, since we are always dealing with multi-dimensional parameters which are correlated. The MSE matrix is difficult to interpret when the parameters are correlated. Hence, instead of computing the MSE matrix, an alternative that can be used in higher dimensions is the *Mahalanobis distance* (Mahalanobis 1936). A plot of coverage probabilities is also provided to judge the performance of the algorithm over replications of data. We explain these concepts serially in these next few sections.

6.2.1 Mode and Probability intervals

A way of monitoring the performance of the sequential update is to plot the mode of the marginal posterior distribution for each time with 95% probability intervals. Since the true values of the parameters are known, our sequential method will be deemed successful in making inference about a certain parameter if the true parameter value lies within the probability intervals.

6.2.1.1 Estimation of Mode

At each iteration, the value on the grid for which the posterior is a maximum is plotted and referred to as the "approximate mode". Note that the "approximate mode" is that point on the grid at which the posterior distribution has a maximum value among all the other points, and is not necessarily the true mode of the posterior. The "approximate mode" can be computed in different ways in this case. We prefer to use the "approximate mode" of the joint posterior density, while that of the marginal posterior can also be used in the trace plots. The marginalization of the posterior is done using finite difference approximation. For each point on the grid of a particular parameter, the joint density values associated with that particular grid value is summed up and multiplied by the distance between grid points for the other parameters. So in mathematical terms, the marginal density for a certain parameter θ_i, belonging to a set of parameters θ, is given by

$$f(\theta_i) \approx \sum_{\theta_{-i}} f(\theta)\Delta(\theta_{-i}). \tag{6.2.1}$$

An improved approximation to the marginal mode of the parameters was also tried. A smooth curve was fitted over the points representing the posterior marginal density defined at the specific grid points. An unique point was chosen for which the posterior marginal was the maximum. Non-parametric methods of curve fitting like smoothing splines or kernel based fits can be used for constructing the density. However this was deemed tedious and difficult for reasons that will be explained in greater detail in the next section.

6.2.1.2 Estimation of 95% marginal probability limits

The 95% marginal probability bounds or credible limits for each of the parameters are also computed. For a particular parameter, to compute the bounds, the marginal density has to be calculated. For a posterior distribution defined on a grid, the joint credible interval is trivial to compute. However, it is very difficult to represent the performance of the algorithm using graphical measures with respect to the bounds, since in higher dimensions it becomes impossible to visualize. Thus for graphical representation, the 2.5% and 97.5% percentiles need to be calculated for the marginal posterior density.

The existing grid points for the marginal posterior provide a discrete approximation to the marginal density. Assuming that the grid points of a certain parameter θ are defined in ascending order $\theta^0, \theta^1, \cdots, \theta^n$, first we need to identify the four grid points $(\theta_{L_1}, \theta_{L_2})$ and $(\theta_{H_1}, \theta_{H_2})$ such that $\sum_{i=0}^{L_1} \mathbb{P}(\theta_i) < 0.025 < \sum_{i=0}^{L_2} \mathbb{P}(\theta_i)$ and $\sum_{i=0}^{H_1} \mathbb{P}(\theta_i) < 0.975 < \sum_{i=0}^{H_2} \mathbb{P}(\theta_i)$ respectively. If we define the lower and upper 95% probability limits to be θ_{L_*} and θ_{H_*} respectively, we can use different techniques to calculate them. We explain some of the

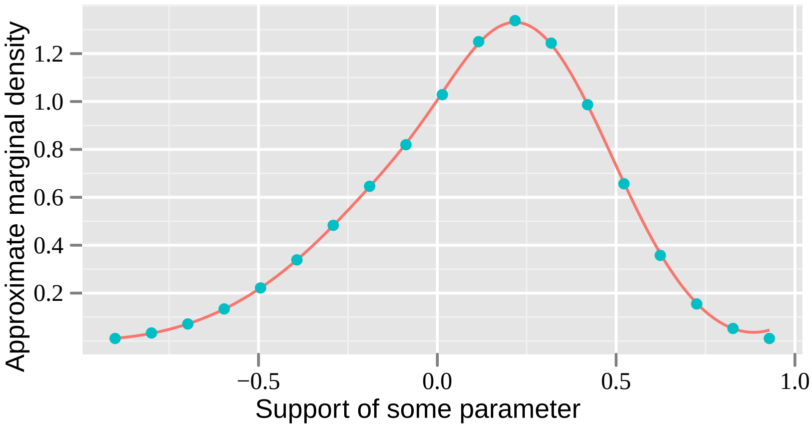


Figure 6.1: Fit of Nadaraya-Watson estimator over the posterior marginal values defined on the grid. One can generate a lot more number of values over the domain of the parameter along with their predictions, normalize them and then use them to get the 95% probability bounds.

techniques that were used to estimate the probability bounds from the posterior marginal.

A non parametric curve fitting technique can be used to fit a curve over the marginal posterior density grid. We used the Nadaraya - Watson (N-W) Estimator (Nadaraya 1989), which is a kernel based regression estimator. The N-W estimator is written as:

$$\hat{f}(x) = \frac{\sum_{i=1}^{n} f_i \mathbf{K}\left(\frac{x-X_i}{h}\right)}{\sum_{j=1}^{n} \mathbf{K}\left(\frac{x-X_j}{h}\right)},$$

where **K** is the kernel and h is the bandwidth. A user-defined number of values over the support of the current marginal of the parameter θ along with their predicted values can be generated using this technique. Thus in other words, while the size of the actual grid is 20, this method improves the accuracy of the approximation for the probability intervals by smoothing out the discrete density approximation. Figure 6.1 shows the N-W estimator fit over the posterior marginal values defined for each grid point.

However this method has its share of problems. For multiple parameters the support of each component can be very different. This means that the bandwidth could be different for different parameters. However that can be overcome by using a different bandwidth for each parameter. A bigger problem is that the support and shape of the posterior marginal density can change with time, as has been discussed while explaining the interpolation methods in Chapter 4. That means a constant monitoring regarding the choice of bandwidth is necessary. This question the very essence of a sequential algorithm, quite in line with the curve fitting methods discussed in Chapter 4.

Another very big problem that will be evident in later examples, is the shape of the posterior marginal density at a given time point. In the grid updating scheme, calculations

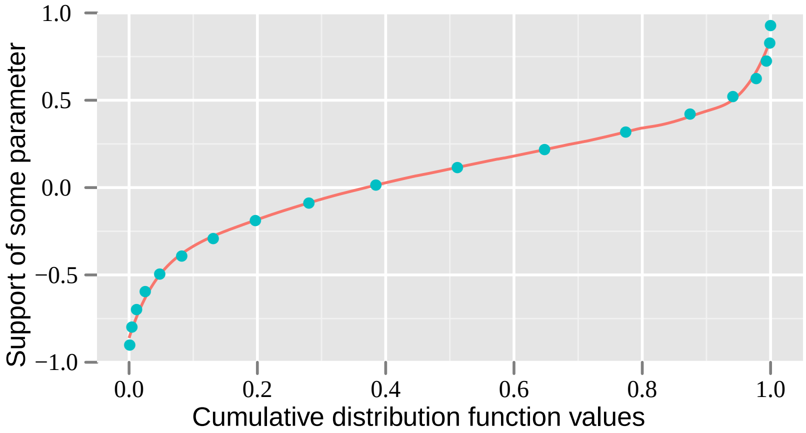


Figure 6.2: Fit of Nadaraya-Watson estimator over the grid values plotted with respect to the CDF values.The CDF values being on the x-axis, inverse probability distribution is applied to get the 2.5% and 97.5% percentiles.

(say of the posterior) related to the new grid points are approximate. They are not as accurate as the existing grid points which made use of the sequential update scheme. So while the posterior of some of the points have gone through sequential updating over a period of time, others have been added quite recently and hence are going through the "learning" process. This may cause the posterior density to be multi-modal over time. It is extremely difficult to fit a curve over a set of points representing a multi-modal density (Scott 1992). Again one could say that all of these problems can be tackled by suitably choosing the bandwidth. But that would involve too much computation. The algorithm would have to be stopped to check for a good fit, and then any corrections, if necessary are to be made. This makes the fitting procedure not practical for a sequential algorithm.

The problem related to the difference in the domain of different parameters can be solved by using the *cumulative distribution function* (CDF). In our algorithm, each of the parameters are treated as discrete random variables and their individual CDFs are calculated. The discretized CDF points are then fitted by a smooth curve. This method has an advantage over the previous one. Because cumulative frequency lies between 0 and 1, we can use the inverse distribution function to get the necessary percentiles, thus avoiding the problem of fitting a curve on varying support for different parameters. Now the CDF values are used as the x-values and the parameter values defining the grid are used as y-values. Since for any parameter, the domain of the x-values is the same, it is expected that a smooth fit with a particular bandwidth will work for all the parameters. Figure 6.2 shows the plot of a smooth fit over the grid values with respect to the cumulative distribution function of a certain parameter.

But even for this technique, updating the grid sometimes creates an issue. The grid update causes new points to the added to the existing grid, while also dropping some at times. So the number of grid points for a certain parameter varies over time. While sometimes the grid structure can be quite sparse (resulting in the addition of more points internal to the support), it can also be quite dense (usually when the update adds points external to the existing support). The bandwidth again forms a problem in the sense that it needs to be varied according to the sparseness of the grid points.

Thus the basic problem with any curve fitting method can be highlighted here. A dynamic grid like ours in a sequential inference method requires a dynamic bandwidth selection method for non-parametric curve fitting to work. Bandwidth selection methods do exist which are fast but are crude and can sometimes give unwanted results. The "ideal" bandwidth is the one which minimizes the distance between the true and the fitted curve (Silverman 1986). Selecting an appropriate bandwidth at each time point is a non-trivial problem. To compute the "ideal" data-driven bandwidth, a lot of computation would be required, which in turn will make the algorithm slow. A good reference for existing bandwidth selection techniques used in kernel density estimation is Turlach (1993).

As has been done in Chapter 4, we use a method which does not require constant monitoring. Instead of smooth curves, we use linear interpolation in the inverse probability function to get the percentiles. Thus the first task is of course to identify the four grid points $(\theta_{L_1}, \theta_{L_2})$ and $(\theta_{H_1}, \theta_{H_2})$ already defined earlier in this section. Linear interpolation is used to determine the two values θ_L and θ_H using the available values mentioned above. This procedure avoids having to choose a bandwidth and hence will not require any monitoring. Thus in the computation of probability bounds we have opted for a compromise between fast computation and accuracy.

On several instances in this section, we have used trace plots showing the values of the approximate mode and the probability bounds on the marginal posterior density. The trace plot of the estimated mode over time is consistently shown in red, whereas that of the probability bounds are shown in blue. The true value of the parameter which has been used in the model to generate data has been shown by a red line in the same plot.

6.2.2 Mahalanobis Distance

It has already been mentioned earlier that the method of computing a MSE matrix is not done in this thesis for lack of clarity. Instead we have used the *Mahalanobis distance* as a measure of accuracy to judge how close (or far) estimates of the parameters are from the true values. The Mahalanobis distance for some vector θ (true parameter values in our case) is defined as:

$$\mathbb{D}(\hat{\theta}) = \sqrt{(\hat{\theta} - \theta)^T S_\theta^{-1} (\hat{\theta} - \theta)}, \tag{6.2.2}$$

where $\hat{\theta}$ is some sample set (basically any vector from parameter space) and S_θ is the covariance matrix of θ. In our case, $\hat{\theta}$ is some estimate of the parameter vector and S_θ is the posterior covariance matrix. This measure is a distance measure which is used to ascertain the "dissimilarity" between two vectors (replace θ by θ_1 and $\hat{\theta}$ by θ_2 in Equation (6.2.2)) or an estimate from a true value (this is what has been shown in the formulae). The Mahalanobis distance is computed at every time point. This helps us to track the estimation of the posterior density of the parameters with respect to the true values.

For multivariate normally distributed data, the Mahalanobis distance value is approximately distributed as a χ_p^2, where p is the dimension of the datum (Filzmoser 2004). It is true that there is more chance for the parameters to have a non-Gaussian distribution, and hence using a χ^2 value to judge the distance metric is not correct in our case. However, it can be used as a reference value.

6.2.3 Coverage Proportion

Bad performance in inference can be either due to a wrong model or a faulty method. By faulty method we mean a fault in the computation or implementation. But in our case, since we are dealing with simulated data, bad performance can only be due to the method. The effect of performance in our case can be studied using coverage proportion. Data are simulated several times and the algorithm is applied to each of these data sets. To monitor the statistical properties of the sequential algorithm, the proportion of times the true value of the parameter lies within the probability bounds are noted for each time point. So if at time t, for a certain replication m_i, $\theta_L^{(m_i)}$ and $\theta_H^{(m_i)}$ are the lower and upper probability limits for parameter θ, then the coverage proportion for that time point t is given by:

$$CP = \frac{\sum\limits_{m_i=1}^{M} \mathbb{I}(\theta_L^{(m_i)} \leq \theta_{true} \leq \theta_H^{(m_i)})}{M},$$

where $\mathbb{I}$ is the *indicator function* such that,

$$\mathbb{I}(x) = \begin{cases} 1 & \text{if } x \text{ is true,} \\ 0 & \text{otherwise.} \end{cases},$$

for some set $\mathbb{H}$. M is the total number of replications done.

6.2.4 Computation Time

As a last but very important measure, we take note of the computation time for our method, and compare it to other methods. Computation time for each of the models is taken into account both for SMC and INLA. While there is a direct comparison between SINLA and SMC on the basis of speed, it is of course obvious that INLA will be slow as the data grow in size.

6.3 Application of different algorithms in different models

This is a very short section providing details of what methodology is applied in which example along with justifications. As has been discussed at the beginning of this chapter, we will be applying Sequential INLA on different data-sets simulated from three different models in this section. While the basic assumptions and methodology remains the same for all the models, certain things are different. For a start, the computation of the filtering density and prediction density depends on the type of model. While the Kalman filter is used for a linear Gaussian model, for models with nonlinearity or non-Gaussian errors, unscented Kalman filter is applied. Also, the correction factor discussed in Chapter 5 has been applied in the nonlinear model.

A small table is provided here to show which method has been used in which model. Note that the crude method has been used only on the linear model since it produced unsatisfactory results, as will be seen later.

		Method			
		Crude	KF	UKF	Correction
Model	Linear	√	√	×	×
	Nonlinear	×	×	√	√
	Non-Gaussian	×	×	√	×

6.4 Linear Model with Additive Gaussian Errors

An example of the problem under consideration can be described by the following: a single radio antenna is transmitting at a fixed frequency and the signal is being received simultaneously at several spatially distinct nodes. The power loss in the signal during its transmission from transmitter to receiver is assumed to be the same for all receivers (this is not essential and can be removed if a reasonably accurate path loss model is available). As such, differing measurements at the various receivers are all assumed to be noisy observations of the same latent process. A pictorial illustration is given in Figure 6.3.

For real life data, one can use the classical time series model identification techniques (ACF and PACF plots) (Brockwell & Davis 2002) to create the model for the latent process. Several different transformation techniques can be applied on the data to suit the assumptions defined in our thesis. Also transformations can sometimes be necessary on the model itself (log of a multiplicative model). These issues have already been discussed in Chapter 2. Here though, we simulate data based on the above pictorial set-up. We assume the data to be noisy observations of a latent signal which follows a first order auto-regressive model (AR(1)). The latent process thus has the form:

$$X_t = \phi X_{t-1} + w_t, \tag{6.4.1}$$

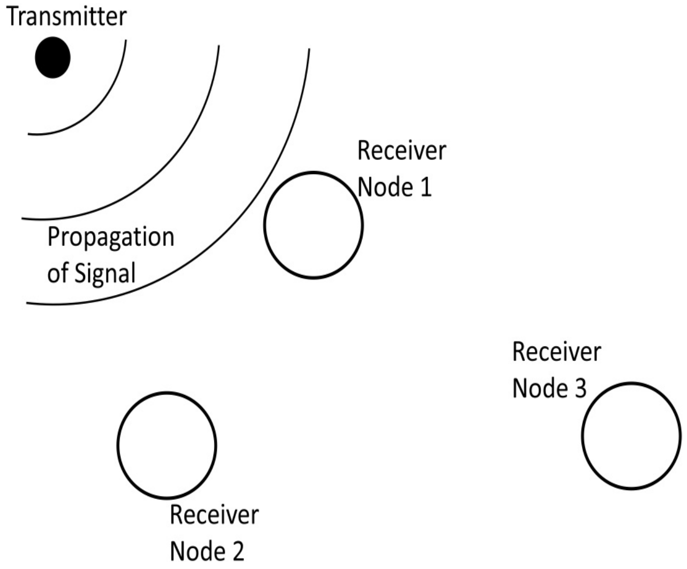


Figure 6.3: This is a pictorial illustration of the radio antenna and the transmitters under consideration.

where $w_t \sim N(0, \sigma_x^2)$ and $Cov(w_s, X_t) = 0$ for $s < t$.

At time t the observations $\mathbf{Y}_t$ are assumed to be from three separate nodes which are spatially correlated, and this is incorporated within the model by claiming that the observation vector $\mathbf{Y}_t \sim \mathcal{MVN}\,(\underline{\mathbf{1}}X_t, \Sigma)$. Thus the observation equation can be written in the following linear form:

$$\begin{aligned}\mathbf{Y}_t &= \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} X_t + \mathbf{w}_t, \\ &= \underline{\mathbf{1}}X_t + \mathbf{w}_t. \end{aligned} \tag{6.4.2}$$

The covariance matrix Σ is given a very general and flexible form and the entries are of the following type:

$$\begin{aligned}\Sigma_{ii} &= \sigma_y^2, \\ \Sigma_{ij} &= \sigma_y^2 \exp^{-rd(i,j)},\end{aligned}$$

where $r > 0$ and $d(i,j)$ is a measure of distance between nodes i and j. This is the well known Gaussian spatial process (Matérn 1986). We assume r to be known. Thus we have to estimate the set of static parameters $\theta = \{\phi, \sigma_y^2, \sigma_x^2\}$, apart from the latent process X_t.

Data were generated by fixing the values at $\phi = 0.35, \sigma_y^2 = 0.004, \sigma_x^2 = 0.035$, the value of r at 2/3 and the distance values were set at $d(1,2) = 1$, $d(1,3) = 3$ and $d(2,3) = \sqrt{10}$. The covariance matrix is thus

$$\Sigma = \sigma_y^2 \begin{pmatrix} 1 & e^{(-2/3)} & e^{(-2)} \\ e^{(-2/3)} & 1 & e^{(-2\sqrt{10}/3)} \\ e^{(-2)} & e^{(-2\sqrt{10}/3)} & 1 \end{pmatrix}.$$

The set of hyperparameters are redefined as $\Theta = \{\phi, \rho_y, \rho_x\}$ instead of $\{\phi, \sigma_y, \sigma_x\}$, where now $\rho_y = 250$ and $\rho_x = 816.32$.

6.4.1 Method 1: Crude Method of Chapter 4

A crude solution for sequentially updating the posterior of $\theta \mid \mathbf{Y}_{1:t}$ was proposed in Chapter 4. The sequential multiplier $g_t(\theta)$, which updates the posterior density at time $t-1$ defined as $\mathbb{P}(\theta \mid \mathbf{Y}_{1:t-1}) \times g_t(\theta) = \mathbb{P}(\theta \mid \mathbf{Y}_{1:t})$ when at time t a new observation $\mathbf{Y}_t$ is recorded, is specified as

$$g_t(\theta) = \left.\frac{\mathbb{P}(X_t \mid X_{t-1}, \theta)\mathbb{P}(\mathbf{Y}_t \mid X_t, \theta)}{\mathbb{P}_G(X_t \mid \mathbf{Y}_t, \theta)}\right|_{X_t = x_t^*}. \qquad (6.4.3)$$

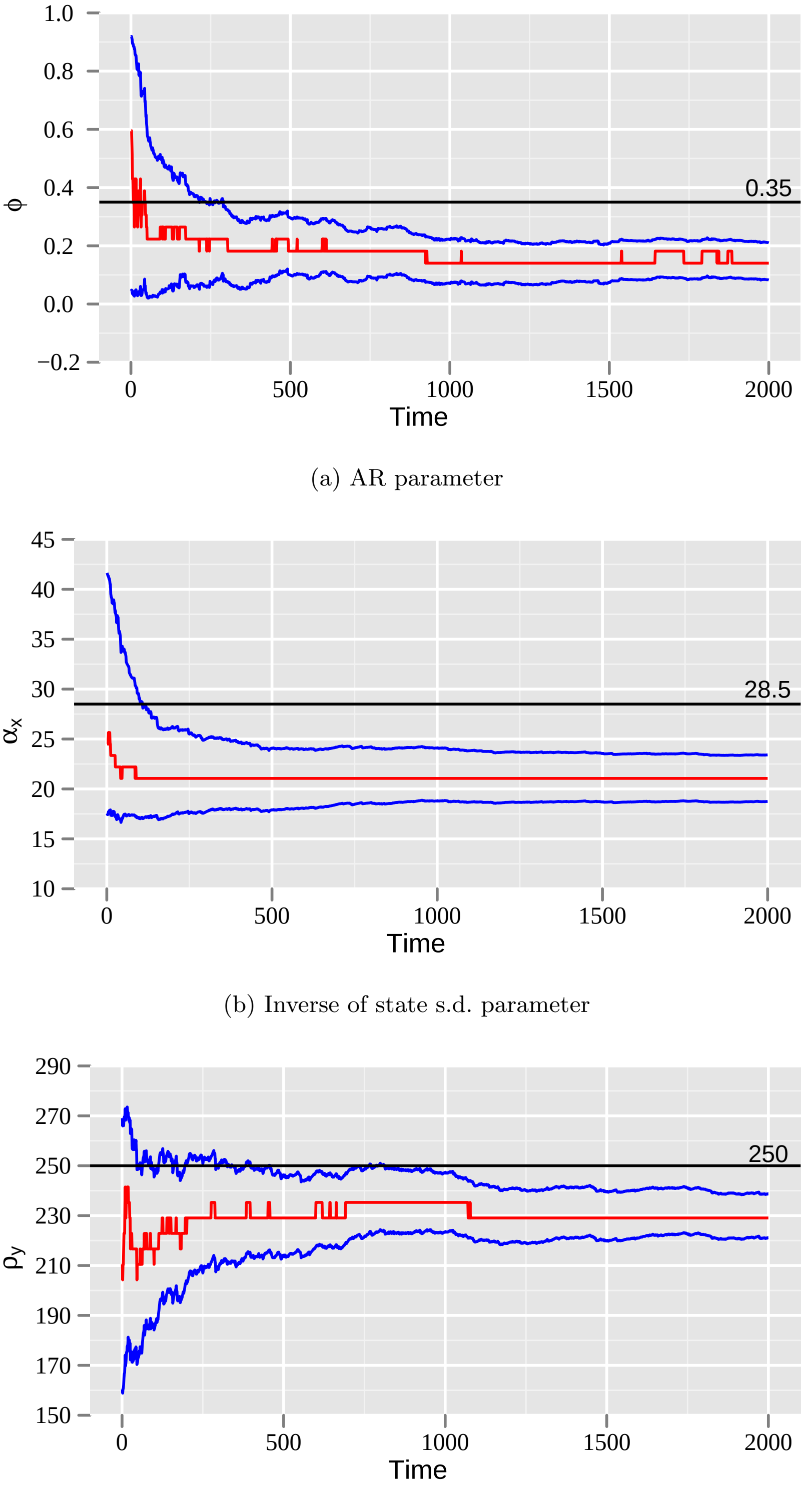
6.4.1.1 Computation of $\mathbb{P}(X_t \mid \mathbf{Y}_t, \theta)$

The denominator term in Equation 6.4.3 computes the Gaussian approximation to the density $\mathbb{P}(X_t \mid \mathbf{Y}_t, \theta)$, which in this example is Gaussian. The terms in the numerator and denominator are known exactly. A Gaussian approximation needs to be computed for $\mathbb{P}(X_t \mid \mathbf{Y}_t, \theta)$, for each grid value of θ. There is no need to store any statistic for this method, since the mean of $\mathbb{P}(X_t \mid \mathbf{Y}_t, \theta)$ is computed at every time point.

6.4.1.2 Application of the Crude Method

This method is applied to data simulated from the linear Gaussian model. One point needs to be mentioned here before the results are shown. Instead of estimating the precision parameter ρ_x $(= 1/\sigma^2)$, we estimate $1/\sigma$ (say α_x), the inverse of state standard deviation. The performance of this method is shown in Figure 6.4. A trace plot of the approximate mode (red line) and approximate 95% probability intervals (blue lines) is done. It is clear that the algorithm based on crude method is not being able to identify the true values at all. Note, however, from the plot that INLA identified the true parameters quite well. The true parameter value lies in the support of the initial grid. Thus it is evident from the figure that even though our algorithm started off well, the performance detoriated over time as the crude sequential algorithm was initiated.

The other diagnostic criteria were not applied at all for judging the performance of this method, since we anticipated them to fail in any case. Measures like coverage or Mahalanobis distance would have very bad values for this method, which is quite obvious if one follows the performance of this method in Figure 6.4. This method was applied to a few other data sets, simulated from the same model, and in all of them it was not able to identify the true value of the parameters. It was quite clear that in practice this method established the theoretical drawbacks associated with it. Since it was mentioned in Section 4.3.1 of Chapter 4 that the assumptions made for this method are indeed quite



(a) AR parameter

(b) Inverse of state s.d. parameter

(c) Observation precision Parameter

Figure 6.4: Plot of the mode and 95% probability interval for all the three parameters of the linear model, over time. The actual value of the parameters do not fall within bounds for any of the cases, symbolizing poor performance for our algorithm when used with the crude method.

crude and hence unpractical, it was decided not to try this method with a large number of simulated data sets or pursue the use of it any further.

6.4.2 Method 2: Sequential Update of $\mathbb{P}(\theta \mid \mathbf{Y}_{1:t})$

The sequential Bayesian update to the posterior of the parameters of the model is applied here to compute the posterior distribution at time t. This update as given by Equation (4.3.4) in Chapter 4 can be written as:

$$\mathbb{P}(\theta \mid \mathbf{Y}_{1:t}) \propto \mathbb{P}\left(\theta \mid \mathbf{Y}_{1:t-1}\right) \frac{\mathbb{P}\left(\mathbf{Y}_t \mid \mathbf{X}_t, \theta\right) \mathbb{P}\left(\mathbf{X}_t \mid \mathbf{Y}_{1:t-1}, \theta\right)}{\mathbb{P}\left(\mathbf{X}_t \mid \mathbf{Y}_{1:t}, \theta\right)}, \tag{6.4.4}$$

where it is assumed that the posterior at $t-1$, i.e. $\mathbb{P}\left(\theta \mid \mathbf{Y}_{1:t-1}\right)$ is known. We study the performance of the algorithm in the next few sub-sections.

6.4.2.1 Computation of $\mathbb{P}(X_t \mid \mathbf{Y}_{1:t-1}, \theta)$ and $\mathbb{P}(X_t \mid \mathbf{Y}_{1:t}, \theta)$ for Sequential Update

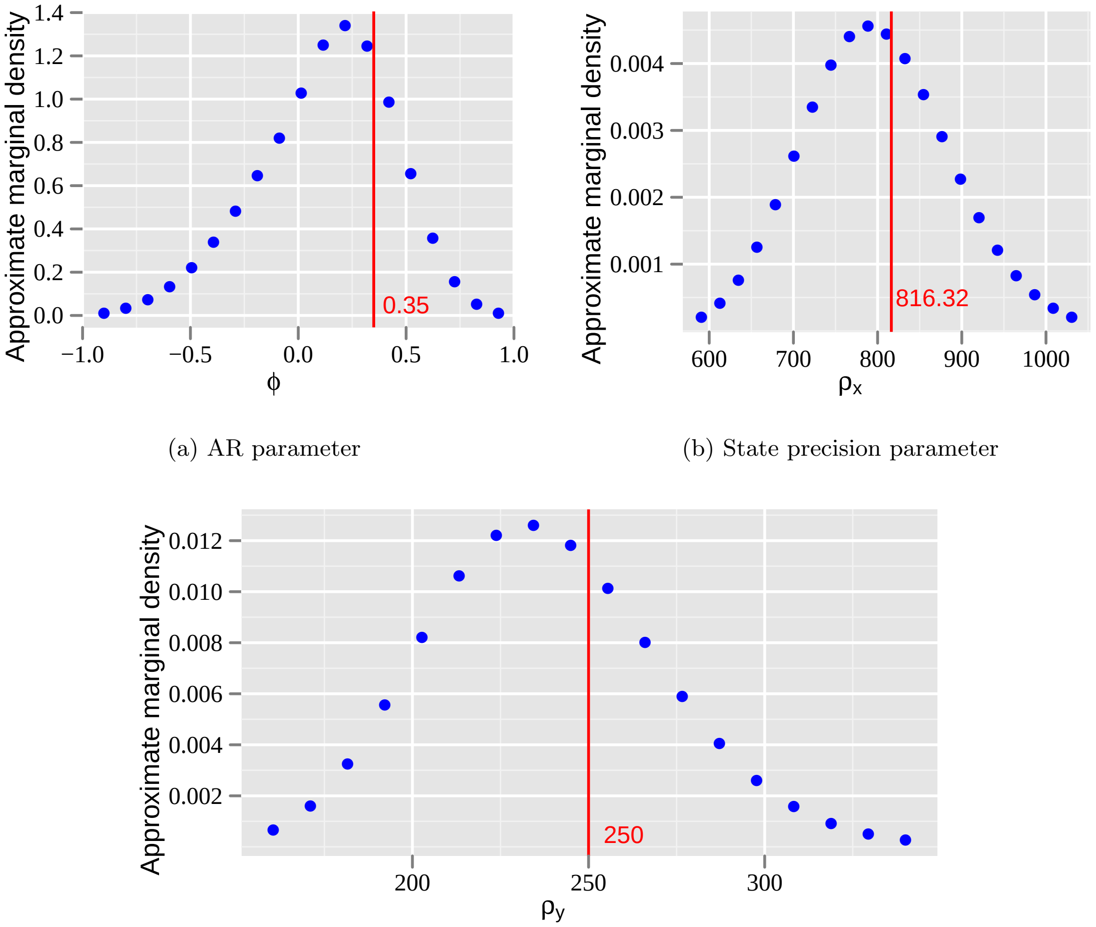
The model for this problem is linear with additive Gaussian errors. Hence computation of the two distributions viz., $\mathbb{P}(X_t \mid \mathbf{Y}_{1:t-1}, \theta)$ and $\mathbb{P}(X_t \mid \mathbf{Y}_{1:t}, \theta)$, which are Gaussian, is exact. The Kalman filter provides the mean and covariance for both these two terms, given the mean and covariance of $\mathbb{P}(X_{t-1} \mid \mathbf{Y}_{1:t-1}, \theta)$. INLA provides the starting mean and variance for $\mathbb{P}(X_{t-1} \mid \mathbf{Y}_{1:t-1}, \theta)$, along with the grid for $\mathbb{P}(\theta \mid \mathbf{Y}_{1:n})$ in this problem. Thus the sequential update for this model is written as:

$$\tilde{\mathbb{P}}\left(\theta \mid \mathbf{Y}_{1:t}\right) = \tilde{\mathbb{P}}\left(\theta \mid \mathbf{Y}_{1:t-1}\right) \left(\left. \frac{\mathbb{P}_{KF}\left(X_t \mid \mathbf{Y}_{1:t-1}, \theta\right) \mathbb{P}\left(\mathbf{Y}_t \mid X_t, \theta\right)}{\mathbb{P}_{KF}\left(X_t \mid \mathbf{Y}_{1:t}, \theta\right)} \right|_{X_t^*(\theta)} \right). \tag{6.4.5}$$

Here X_t^* is the Kalman mean of $\mathbb{P}\left(X_t \mid \mathbf{Y}_{1:t}, \theta\right)$, which is exact. There is no error associated with this method, since Kalman filter gives exact estimations for the unknown state process.

INLA is implemented on the first 20 observations to compute $\mathbb{P}(\theta \mid \mathbf{y}_{1:20})$. The AR parameter ϕ, has a truncated Gaussian prior, while the precisions parameters ρ_x and ρ_y have Gamma priors associated with them. Figure 6.5 shows the starting grid for each of the three parameters along with the true values of $\Theta = (\phi, \rho_x, \rho_y)$. It is evident from the plot that INLA does a good job in inferring the true values of the parameter, based on only very few observations. But it can also propose a starting grid far from the true value of the parameter. Figure 6.6 shows the posterior marginal density of the observation precision parameter ρ_y for some particular data set, again simulated from the same model. Here the true value of the parameter lies at the very edge, and has been found to be well outside the 95% probability limits.

The performance of the grid update scheme is demonstrated in Figure 6.7. In Figure



(a) AR parameter

(b) State precision parameter

(c) Observation precision parameter

Figure 6.5: The posterior marginal distribution for each of the parameters are plotted along with the true values given by the red line. Note that INLA is successful in identifying the true value of the parameter for some specific data set from the linear model.

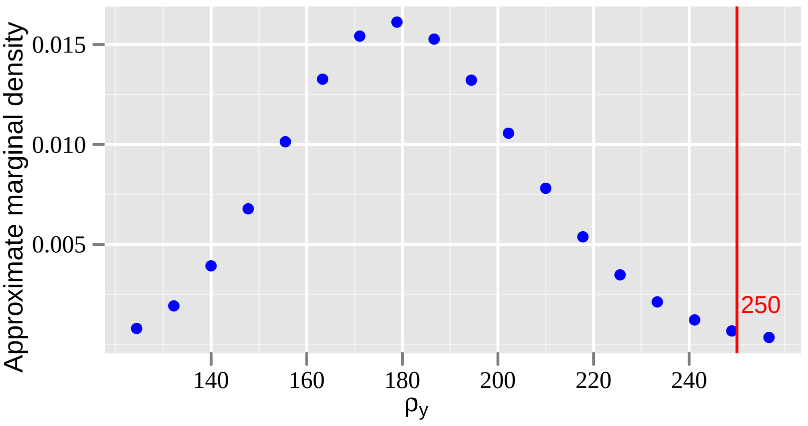
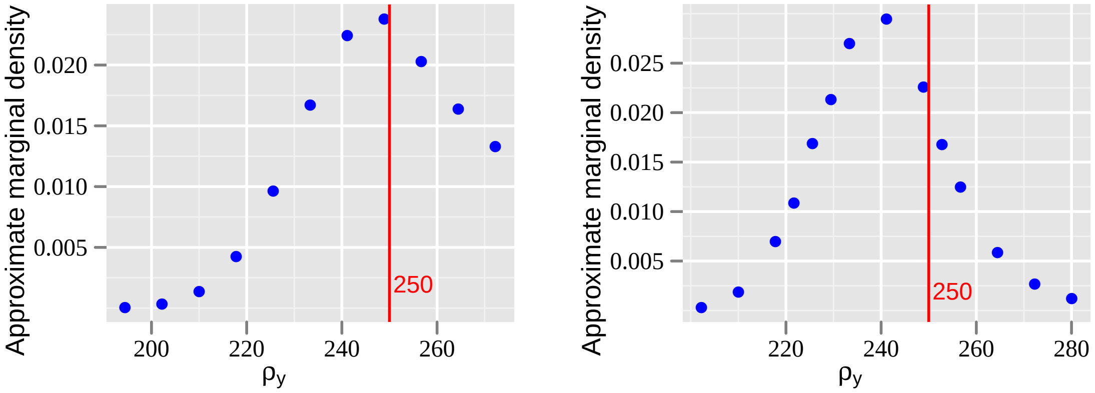


Figure 6.6: Posterior marginal density defined over the grid for the observation precision parameter ρ_y as computed by INLA. The true value lies outside the probability limits, signifying a poor performance for INLA.



(a) At time 150

(b) At time 250

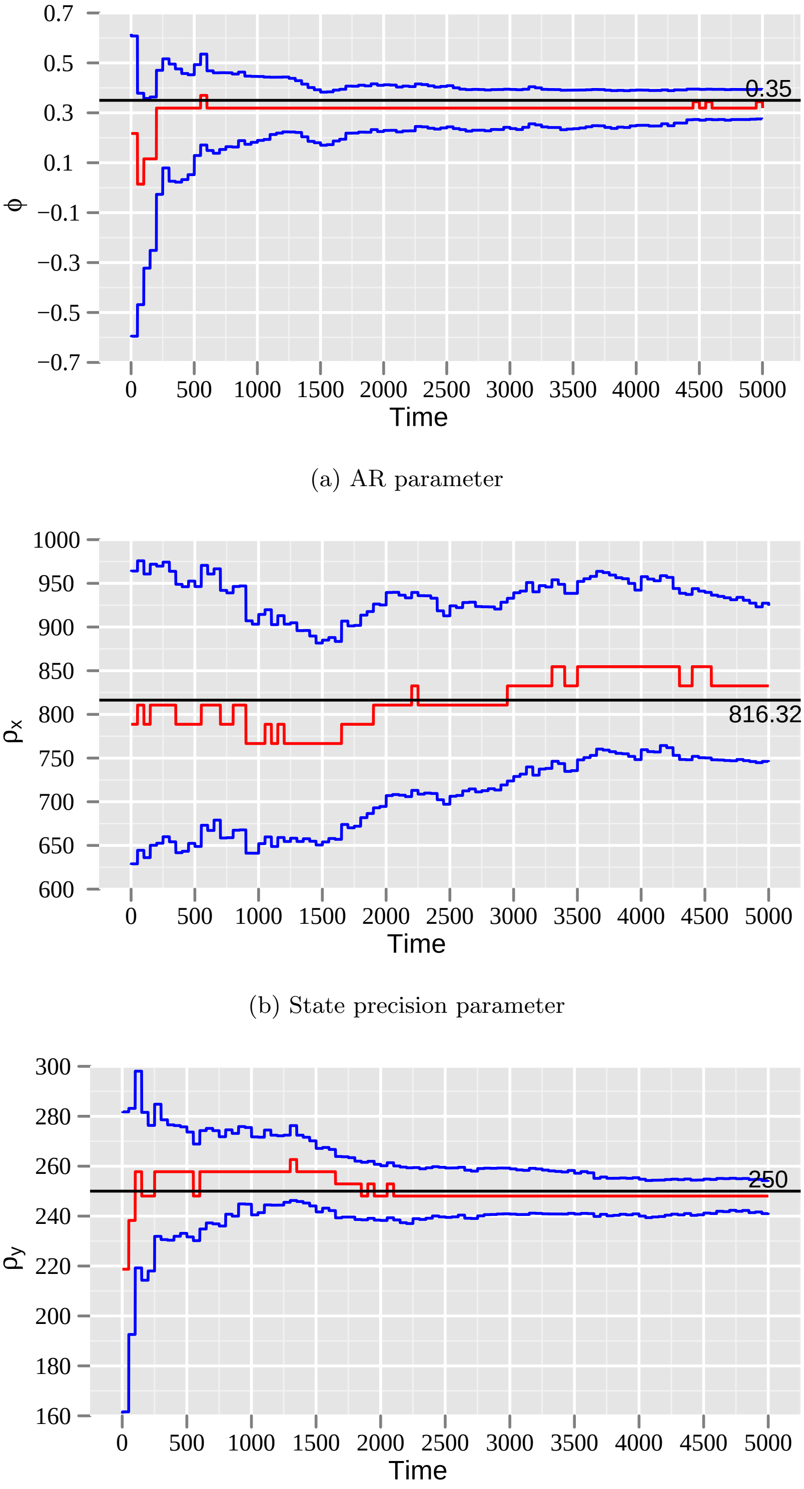
Figure 6.7: Posterior marginal density adding more grid points to the right tail and dropping points at the left tail. Thus from the starting grid provided by the INLA, the grid has moved considerably to accommodate for the true value.

6.7a one would expect the grid update scheme to add more grid points towards the right tail of the support as more data are used to sequentially update the posterior. Figure 6.7b shows how with time the support of the observation precision parameter changes to accommodate more grid points and the posterior slowly shifts in the right direction by adding new points, while dropping grid points at the left tail. This suggests that the grid updating algorithm is doing what we desired and is continuously updating the support of the grid points to match with the true support of the posterior. We could have provided plots of the posterior computed at time points further down the line. However, it was felt to be not of great necessity since in most of the situations the approximate posterior converged to a fixed (almost) support. Hence the grid update wasn't required even though the check was always on.

6.4.2.2 Results after implementing the algorithm

Figure 6.8 shows the performance of the algorithm in estimating the three unknown parameters for a specific data set among the group of simulated data-sets. Given this data set, it is evident that our algorithm performed well in inferring the true values of the parameters. The posterior mode for all three parameters nearly converges to the real values. Also one should note the shrinkage of the probability intervals over time, signifying a shrinking of the support of the posterior density. This is an expected phenomenon in Bayesian methodology, where if we start with a prior with a wide support (suggesting prior ignorance), the posterior has a tendency to shrink. It should be mentioned that this is not a general phenomenon though.

However, in certain other data sets simulated from the same model, the algorithm was unable to identify the parameter values. In Figure 6.9 the actual value of the state



(a) AR parameter

(b) State precision parameter

(c) Observation precision parameter

Figure 6.8: Plot of the mode and 95% probability interval for all the three parameters over time for some particular data set simulated from the linear model. The performance of the algorithm is quite good for this particular case.

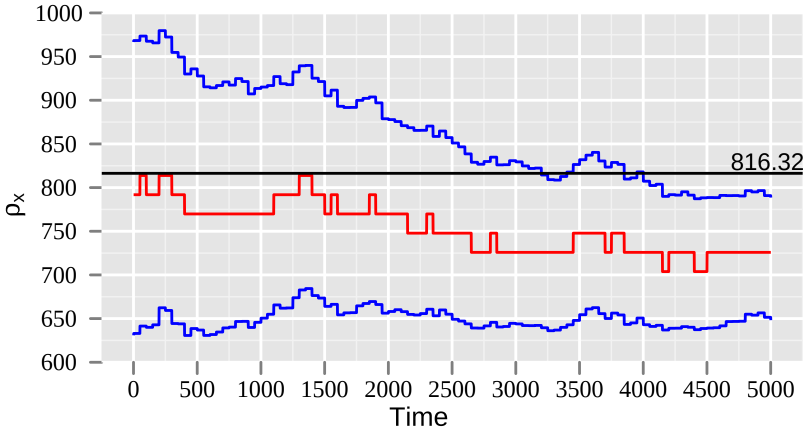


Figure 6.9: The actual value of the state precision parameter, ρ_x of the linear model lies outside the probability bounds after 5000 updates to the posterior distribution.

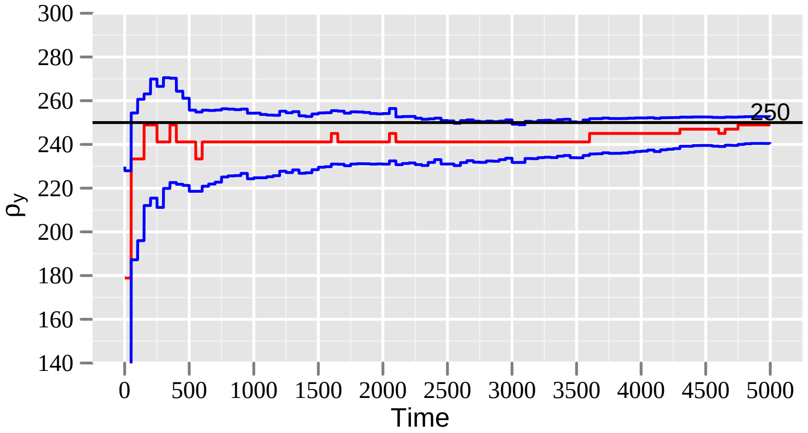
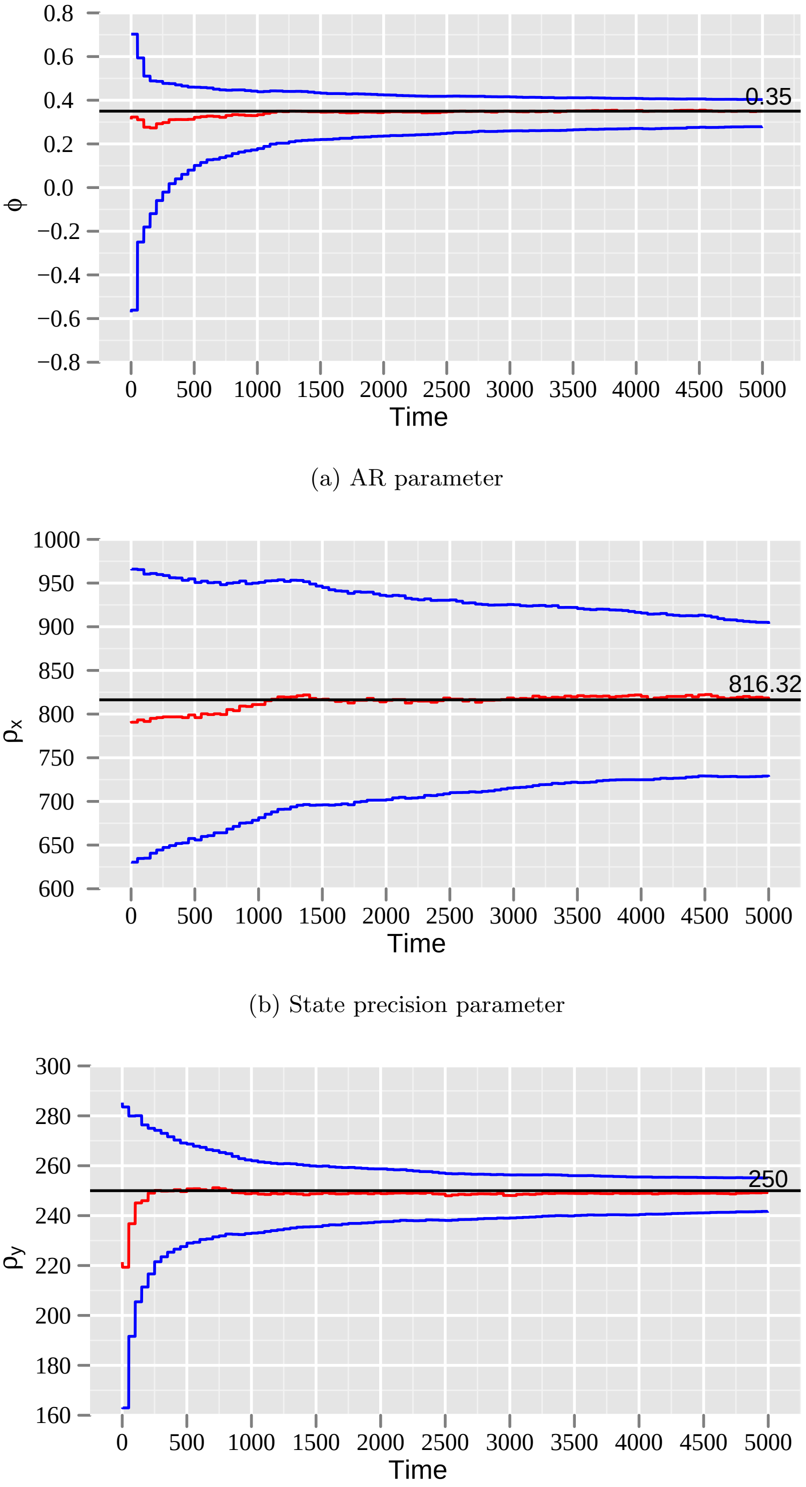


Figure 6.10: In this example the starting grid does not cover the actual value of the observation precision parameter, ρ_y of the linear model. However, the posterior density adds points externally over time to shift and finally the actual value falls within the probability bounds.

precision parameter lies outside the bounds and the sequence does not converge to it even after 5000 time points. Another interesting example for some specific data set, again simulated from the same model, is how the grid updating procedure add points externally and slowly the grid shifts for the posterior density to converge to the actual parameter value. Figure 6.10 illustrates this.

50 different data sets where simulated using the linear Gaussian model and the sequential algorithm was applied on them. The mode of each of the parameters was recorded over time and stored for each data set. Similarly the approximate 95% probability limits of the parameters for each of the data sets were also recorded and stored over time. Figure 6.11 shows the results for the average of the mode and probability limits. One can see that the actual parameter value has been estimated correctly on average and it stays



(a) AR parameter

(b) State precision parameter

(c) Observation precision parameter

Figure 6.11: Plot of the averaged mode and 95% probability interval for all the three parameters over time. The average is done over all the data sets simulated from the linear Gaussian model. Hence on an average the performance of our approach has been excellent for this particular model.

well within the bounds. The algorithm takes its time to "learn" from the observations and the posterior modes converge to their respective true values by time point 1500. For this particular model, the algorithm correctly estimated the parameters and hence the performance can be termed as a success. One more interesting thing that is obvious from Figure 6.11c, is that on average, almost always the posterior mode as provided by INLA is quite far away from the actual value. 20 observations are not really enough to produce good inference, and that causes this inaccurate starting grid. The algorithm does well to update its grid structure and converge to the true value.

6.4.3 Particle filter

SMC methods are also used to estimate the three unknown parameters for this model. The performance of our methodology can then be compared to the existing SMC methods. It has already been discussed in Chapter 2, that the application of an ordinary particle filter on static parameters, results in gradual degeneracy of the particles. In our example for the linear model, we set an informative prior for the parameters and performed SMC method on the states and the parameters. The problem of degeneracy with static parameters is confirmed by this example where we applied SMC on data sets simulated from the linear Gaussian model. For particle filters, re-parameterisation has been applied for the parameters in Equations (6.4.1) and (6.4.2) and the vector of parameters is now defined as $\theta = \{\kappa, \text{log-}\sigma_x^2, \text{log-}\sigma_y^2\}$. The re-parameterisation is shown below:

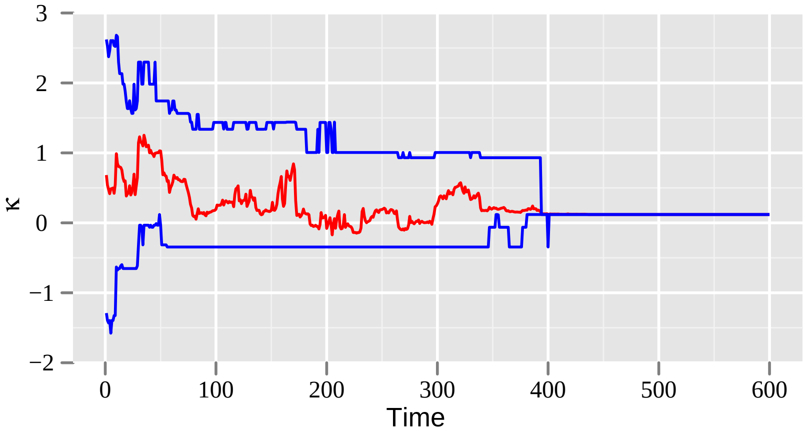
$$\begin{aligned} \kappa &= \text{logit}(\frac{\phi+1}{2}), \\ \text{log-}\sigma_x^2 &= \log(\sigma_x^2) \qquad (6.4.6) \\ \text{log-}\sigma_y^2 &= \log(\sigma_y^2). \end{aligned}$$

Figure 6.12 shows a plot of the estimated Monte Carlo mean along with Monte Carlo 95% credible intervals. Note that after a certain time-point, the credible intervals and the mean merge in to a single line. The degeneracy phenomenon is evident from this toy example. Figure 6.13 plots the number of unique particles for all the three parameters combined at each time point. It is very easy to notice the gradual decrease in the number of unique particles and then finally all the particles degenerate into a single particle.

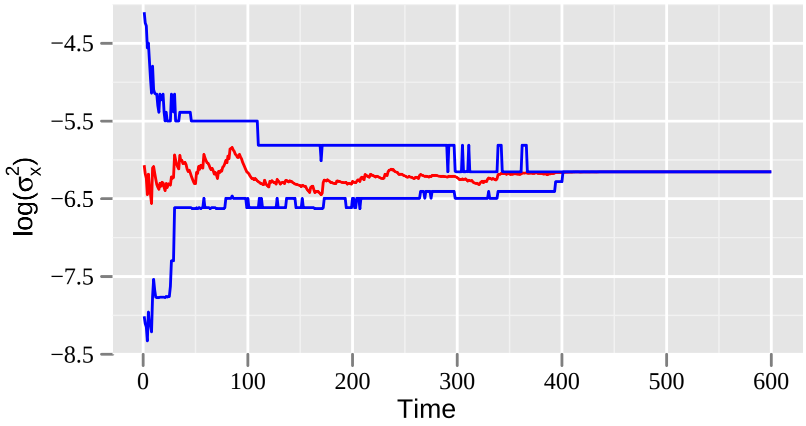
Artificial evolution of the parameters is one way by which this degeneracy phenomenon can be avoided. Thus the parameters in our model, $\Theta = (\kappa, \text{log-}\sigma_x^2, \text{log-}\sigma_y^2)$, are put into an artificial evolution as given by the following equation:

$$\Theta_t = \Theta_{t-1} + \epsilon_t \quad \text{where } \epsilon \sim \mathcal{N}(\mathbf{0}, \Delta). \qquad (6.4.7)$$

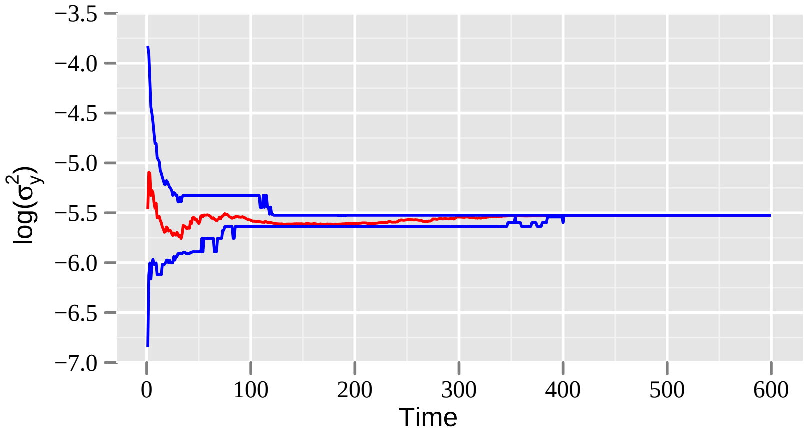
The variance matrix Δ is set by the user. Also, the prior at time t_0 needs to be very



(a) Kappa



(b) Log state variance parameter



(c) Log observation variance parameter

Figure 6.12: Plot of the mean and 95% probability interval of all the three parameters over time using particle filter for the linear model. Note that the intervals and the mean merge with each other at around time point 400.

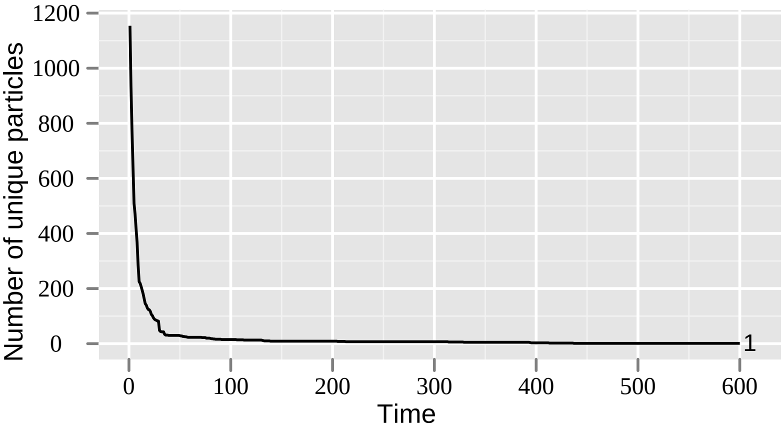


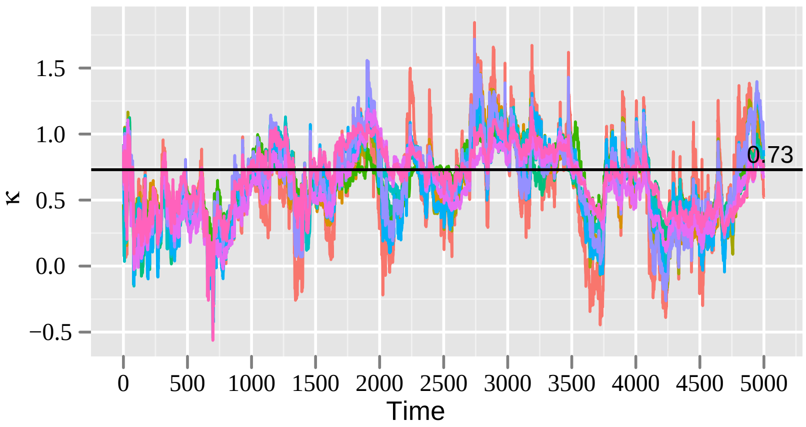
Figure 6.13: The gradual decrease of unique particles of the parameters over time. This happens since there is no sampling-resampling procedure associated with this filter.

good. With a good "informative" prior, a regular SMC was implemented on the linear model along with an artificial evolution of parameters. The SMC method was replicated 10 times, and the mean at each time point plotted in Figure 6.14. The plots of the mean for each of the replications seems to suggest that the filter performed well. The estimated means for each of the parameters seem to be converging towards the true values. This is better understood from Figure 6.15 which is a plot of the average of the means and that of the credible intervals respectively. The true values clearly lie within the bounds suggesting a very good detection.

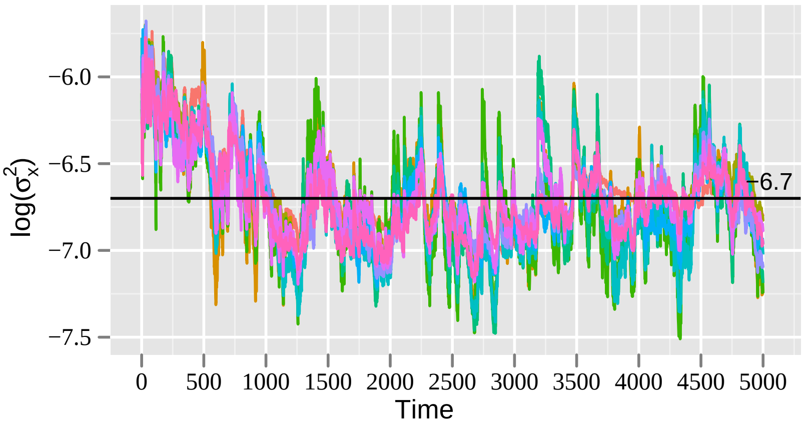
6.4.4 Comparison of the two methods

A summarization of the two methods is done here through the performance measures and methods of accuracy already explained before to compare the two methodologies. Two methods of accuracy are used to judge the performance of our sequential algorithm and SMC method, namely: Mahalanobis distance and coverage probability (proportion).

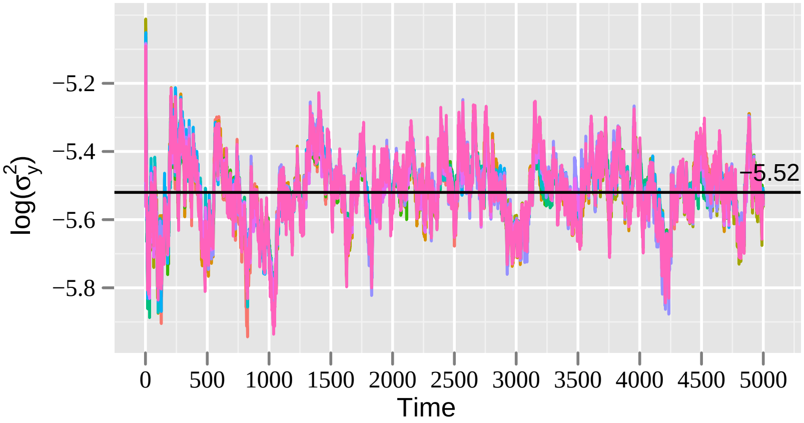
The Mahalanobis distance between the estimates and the true values of the parameters are calculated. Ideally, the smaller the value of the distance, the better the performance of the algorithm is. The Mahalanobis distance is computed over time, to compare the accuracy of estimation between the two algorithms. A box plot of the Mahalanobis distance calculated at regular intervals for all the data sets is shown in Figure 6.16. The plot suggests that both the sequential algorithms have worked well. The Mahalanobis distance values are less than the bound of $\chi^2_3 = 7.82$ (3 dimensional parameter space) for a 5% significance level. The performance of the particle filter, as anticipated, is better than our algorithm which is clearly evident from the box plots. The average of the Mahalanobis distance values for particle filters are consistently lower than that of our method. Between the two methods, the dispersion is also less for particle filters.



(a) Kappa

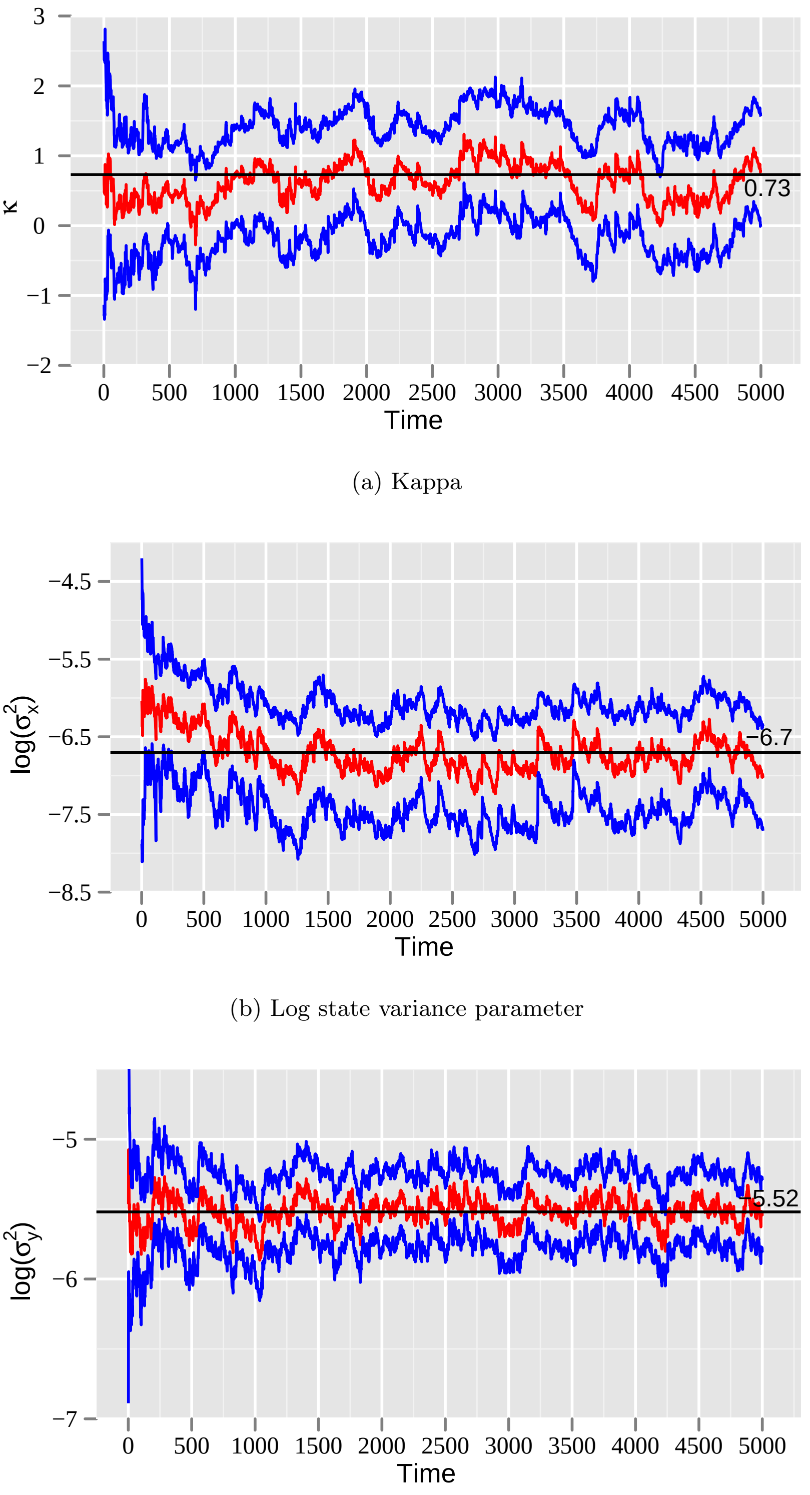


(b) Log state variance parameter



(c) Log observation variance parameter

Figure 6.14: Plot of the mean of the three parameters calculated over time using particle filter, with each trace denoting a certain data set simulated from the linear model.



(a) Kappa

(b) Log state variance parameter

(c) Log observation variance parameter

Figure 6.15: Plot of the average of the mean of the three parameters over time, along with the average of the 2.5% and 97.5% quantile of the particles, calculated from samples generated by particle filter on the linear Gaussian model.

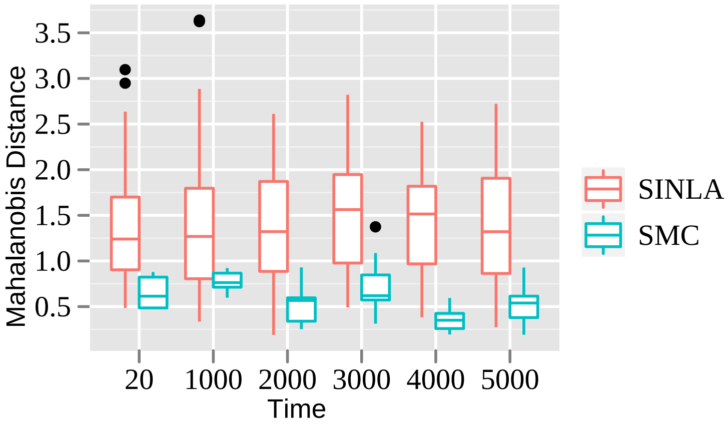


Figure 6.16: Box-plot of Mahalanobis distance computed at regular intervals of time for the two methods implemented for the linear model. The starting box plot on the extreme left of the figure is the Mahalanobis distance computed on the grid provided by INLA, after 25 time points.

The coverage for the three parameters are calculated to get the proportion of times the true value of the parameter falls within the 95% probability limits, for the two algorithms. Figure 6.17 shows the coverage proportion for each of the parameters at regular intervals of time. Coverage proportion has been calculated starting from INLA, which was applied to the first 20 observations, and then computed at regular intervals. The parameterizations are different for the two methods. The parameters for our sequential algorithm are $[\phi, \rho_x, \rho_y]$, while for particle filter they are $[\kappa, \log(\sigma_x^2), \log(\sigma_y^2)]$. The coverage probability should equate the confidence level, which in our case is set at 0.95, if all the assumptions regarding computation of the intervals are met. In our plots, we find the coverage probability for the particle filter to be equal to 1 for all the time points in the plot, thus making the coverage highly conservative. For our algorithm, the ρ_y parameters approximately maintains a coverage probability of 0.95. The ϕ parameter also has the desired value for two of the time points plotted in the graph. So overall, one can say that the coverage probability of the parameters suggest that the confidence intervals have been quite conservative for both the methods.

In addition to all this, a comparison between particle filter, INLA and our method was done based on computation speed and method of accuracy. The average computation time (secs) is computed for each method and is plotted against the average Mahalanobis distance. Figure 6.18 shows the comparison for the two time points 500 and 1000. The computation time for INLA is very high as compared to the two on-line methods, as expected. The particle filter performs as well as INLA for 1000 time points, but not so well for 500 time points. The sequential INLA algorithm returns an average value which remains more or less the same with time. If one can recall from Figure 6.11, on an average, our algorithm identified the true parameter value quite early and stayed on an even path

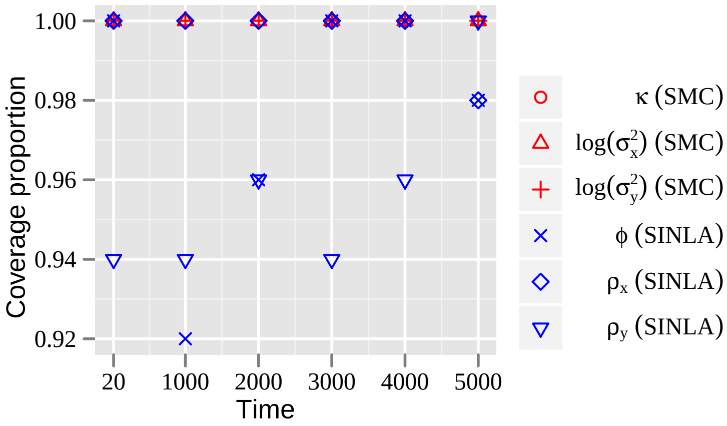


Figure 6.17: Coverage proportion for each of the parameter values plotted over time for the linear Gaussian model. We start with plotting the coverage proportion for INLA based on the starting 20 observations.

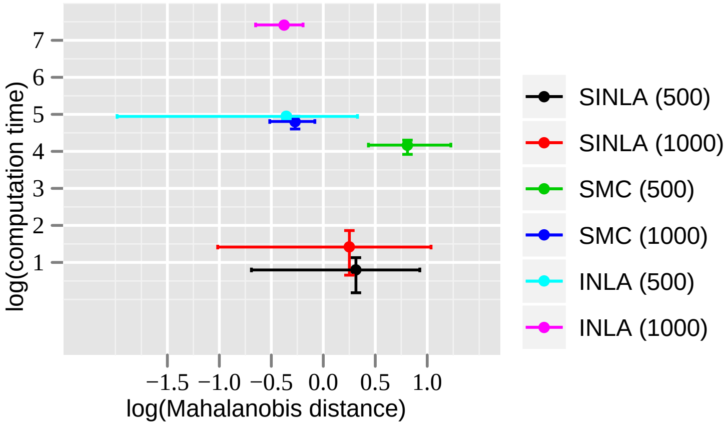


Figure 6.18: Error bar of log-computation time (secs) of each algorithm plotted against Mahalanobis distance computed for time 500 and 1000 for the linear Gaussian model. The limit of the bars are the 25^{th} and 75^{th} percentile respectively.

over time.

6.5 Nonlinear Model

A nonlinear model with additive Gaussian errors is the next example from which data are simulated. The nonlinearity lies in the observation equation, with a time dependent sinusoidal mean component added to the state equation.

$$y_t = \theta x_t^2 + v_t, \tag{6.5.1}$$

$$x_{t+1} = 4 + sin(\omega \pi t) + \phi x_t + w_t. \tag{6.5.2}$$

Here $v_t \sim \mathcal{N}(0, \sigma_y^2)$, $w_t \sim \mathcal{N}(0, \sigma_x^2)$ and ω is assumed to be known. The hyperparameters are given by the vector $\Psi \equiv \left(\phi, \theta, \sigma_x^2, \sigma_y^2\right)$.

Data were generated using the values of the parameters set at $\phi = 0.7, \theta = 2, \sigma_x^2 = 0.0001$ and $\sigma_y^2 = 0.35$, respectively and the value of ω was set at 1.718. A re-parameterisation of the variance parameters was done to facilitate computer arithmetic operations. Thus now we have $\Psi \equiv (\phi, \theta, \rho_y, \rho_x)$ and their actual values are now given as $(\phi = 0.7, \theta = 2, \rho_x = 2.87, \rho_y = 10000)$.

6.5.1 Method Used: Sequential Update

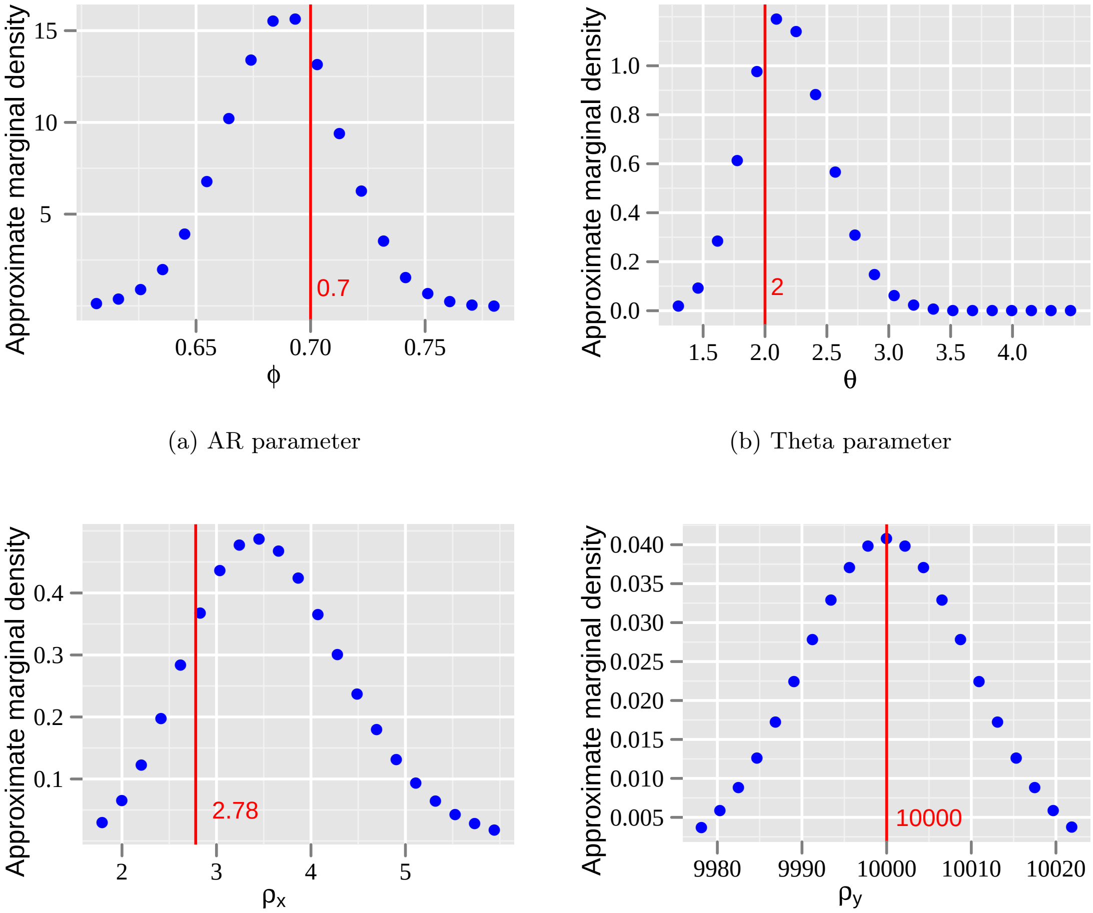
For this model, we did not apply the *Crude Method* at all, since its shortcomings were shown for the linear model. The sequential update as defined in Equation 6.4.4 is being used to update the posterior at each grid point.

6.5.1.1 Computation of $\mathbb{P}(X_t \,|\, \mathbf{Y}_{1:t-1}, \theta)$ and $\mathbb{P}(X_t \,|\, \mathbf{Y}_{1:t}, \theta)$ for Sequential Update

Computation of the two distributions viz., $\mathbb{P}(X_t \,|\, \mathbf{Y}_{1:t-1}, \theta)$ and $\mathbb{P}(X_t \,|\, \mathbf{Y}_{1:t}, \theta)$ is more complex since they are not Gaussian. Any of the extensions of Kalman filter can be used here to address the inference of the state process. In our algorithm, we have used the UKF. Like the Kalman filter, UKF also provides the mean and covariance for both these two terms, given the mean and covariance of $\mathbb{P}(X_{t-1} \,|\, \mathbf{Y}_{1:t-1}, \theta)$. As before, INLA provides the starting mean and variance along with the grid for $\mathbb{P}(\theta \,|\, \mathbf{Y}_{1:t})$. Thus the sequential update for this model is written as:

$$\tilde{\mathbb{P}}\left(\theta \,|\, \mathbf{Y}_{1:t}\right) = \tilde{\mathbb{P}}\left(\theta \,|\, \mathbf{Y}_{1:t-1}\right) \left(\left. \frac{\mathbb{P}_{UKF}\left(X_t \,|\, \mathbf{Y}_{1:t-1}, \theta\right) \mathbb{P}\left(\mathbf{Y}_t \,|\, X_t, \theta\right)}{\mathbb{P}_{UKF}\left(X_t \,|\, \mathbf{Y}_{1:t}, \theta\right)} \right|_{X_t^*(\theta)} \right). \tag{6.5.3}$$

Here X_t^* is the mean of $\mathbb{P}\left(X_t \,|\, \mathbf{Y}_{1:t}, \theta\right)$. One should note that there is a degree of error associated with this method. UKF provides approximation to the terms $\mathbb{P}(X_t \,|\, \mathbf{Y}_{1:t-1}, \theta)$



(a) AR parameter

(b) Theta parameter

(c) State precision parameter

(d) Observation precision parameter

Figure 6.19: The posterior marginal distribution for each of the parameters are plotted along with the true values of the parameters for the nonlinear model, given by the red line. INLA has successfully identified the true parameter values in this specific data set simulated from the model.

and $\mathbb{P}(X_t \mid \mathbf{Y}_{1:t}, \theta)$. Hence it contributes to some error through the multiplicative term:

$$\mathbb{P}_{UKF}\left(X_t \mid \mathbf{Y}_{1:t-1}, \theta\right) / \mathbb{P}_{UKF}\left(X_t \mid \mathbf{Y}_{1:t}, \theta\right).$$

Also the correction factor as discussed in Chapter 5 has been implemented in this model. Hence it is not possible to quantify the error for this algorithm.

INLA was used iteratively for the first 50 data values after which the sequential algorithm started. The AR parameter ϕ had a truncated Gaussian prior, θ had a Gaussian prior, while the precisions parameters ρ_x and ρ_y had Gamma priors associated with them. Figure 6.19 shows the starting grid for each of the three parameters along with the true values of $(\phi, \theta, \rho_x, \rho_y)$. INLA again does a very good job at inferring the posterior density of the parameters. For most of the data sets simulated from this nonlinear model, we find a good detection of the actual parameter values by the starting grid. Only in a few cases

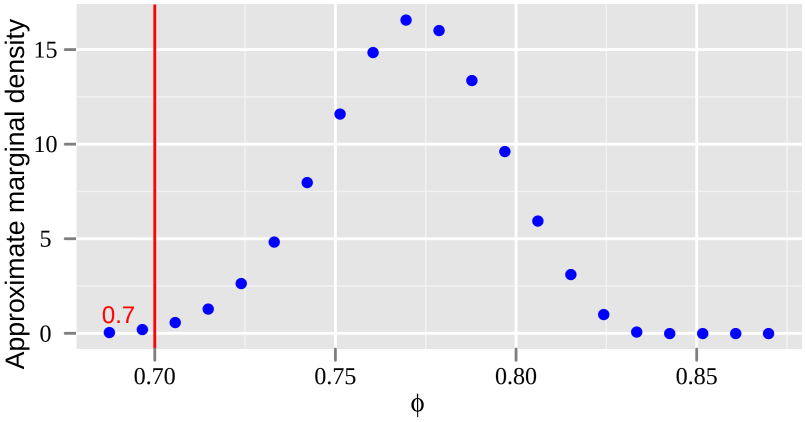
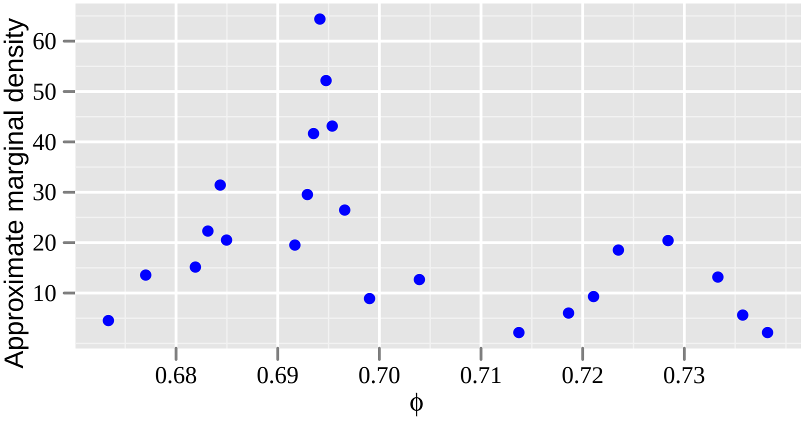


Figure 6.20: This is an example, where the starting grid provided by INLA is not able to identify the true value of the parameter ϕ, for a specific data set simulated from the nonlinear model.

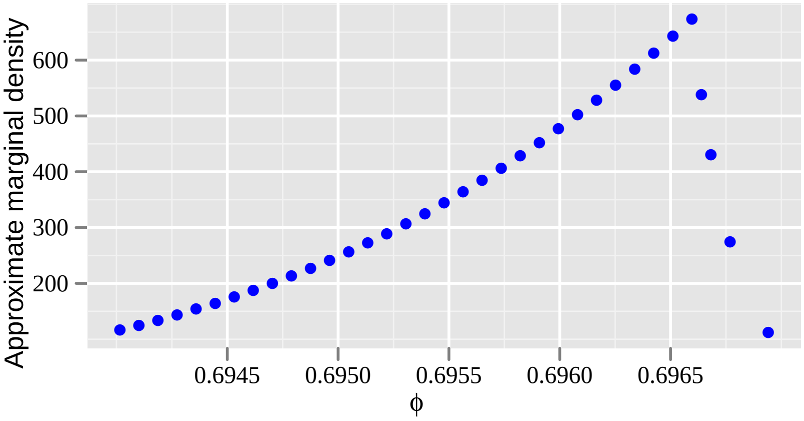
do we see the posterior density of some parameter not identifying the actual parameter value correctly. Figure 6.20 shows one such case, where the true value of the parameter lies at the tail of the posterior marginal distribution.

As has been mentioned earlier in Chapter 4, the shape of the posterior density changes shape and support as incoming new observations affect the updating process. This is particularly evident in this model. More often than not, the posterior distribution for this problem becomes multi-modal. Unlike the linear Gaussian model, the rate at which grid points are added and dropped for this model is very high and that affects the shape of the posterior a great deal. Also unlike the linear model, the filtering density is not exact. It is approximate since it assumes normality of the distributions while propagating the actual nonlinearity of the model. For each new grid point, hence it is obvious, that UKF takes longer time to converge to the actual value for that particular grid point. Figure 6.21a shows the multimodal shape of the posterior density for one of the parameters ϕ. The addition of internal or external points diminishes considerably with time. Also the shape of the posterior becomes unimodal, as more and more observations train it. Figure 6.21b shows this behavior after around 2000 observations have been used to update the posterior density.

The constant update of the grid does not produce a smooth looking posterior, even after a good number of time points. At the beginning of the sequential process, points are added and dropped very frequently. The updates of the posterior for the new points never really match up to the posterior of the actual starting grid. The inability of the posterior of the new grid points to quickly update themselves approximately close to the real density gives the multimodal shape of the approximate sequential posterior distribution. Methods which are extensions of the Kalman filter are sub-optimal filters. The approximating errors get multiplied at each iteration, hence adding to the overall error of our sequential method.

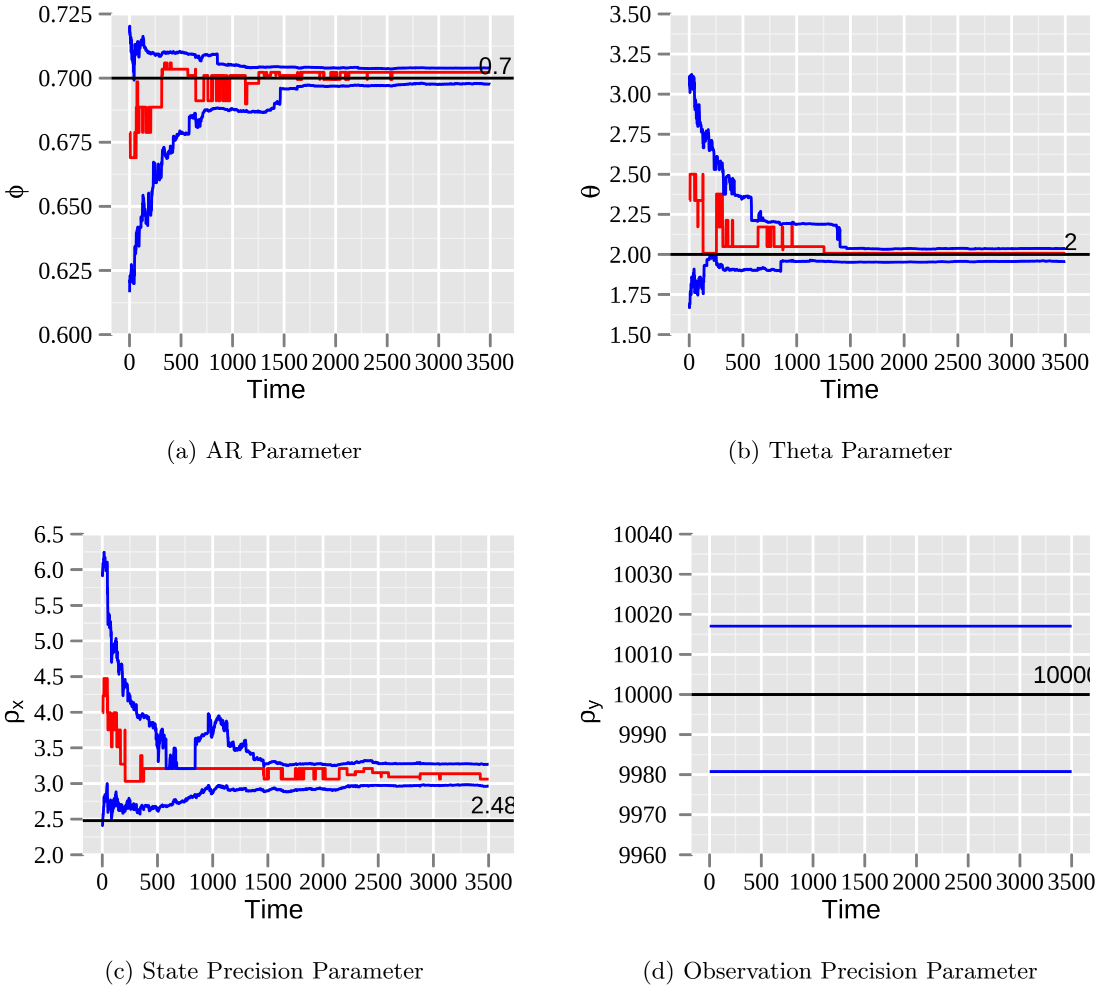


(a) Posterior after approximately 500 time points



(b) Posterior after approximately 2000 time points

Figure 6.21: Posterior distribution of one of the parameters at different time points of the algorithm. Note the change in support of the posterior and also the incredible change in shape of the density.



(a) AR Parameter

(b) Theta Parameter

(c) State Precision Parameter

(d) Observation Precision Parameter

Figure 6.22: Plot over time, of the mode and 95% probability interval for all four parameters from the nonlinear model. The performance of SINLA has been good overall, with failure to identify the state precision parameter completely.

6.5.1.2 Performance of the Sequential Algorithm

The performance of the algorithm for this model is not as good as for the linear model. Figure 6.22 shows the performance of the algorithm in estimation of the unknown parameters for a particular data set. For this particular data set, the three parameters ϕ, θ and ρ_y are estimated quite well and are well within the intervals. The state precision parameter ρ_x is not estimated well by INLA as is evident from the plot. With time, the mode of the posterior density of ρ_x converges at around a value of 3. For certain data sets, the posterior distribution of the state precision parameter does converge to the actual value. Figure 6.23 shows us a situation where the algorithm successfully picks out the true parameter value.

As in the previous example, 50 data sets are simulated using the nonlinear model. Figure 6.24 shows the results for the average of the mode and 95% probability intervals of each of the parameters. The performance of the filter for this model is again quite good, as

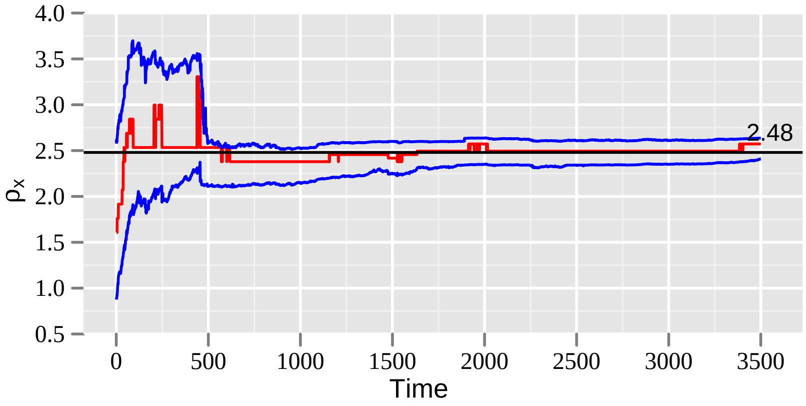
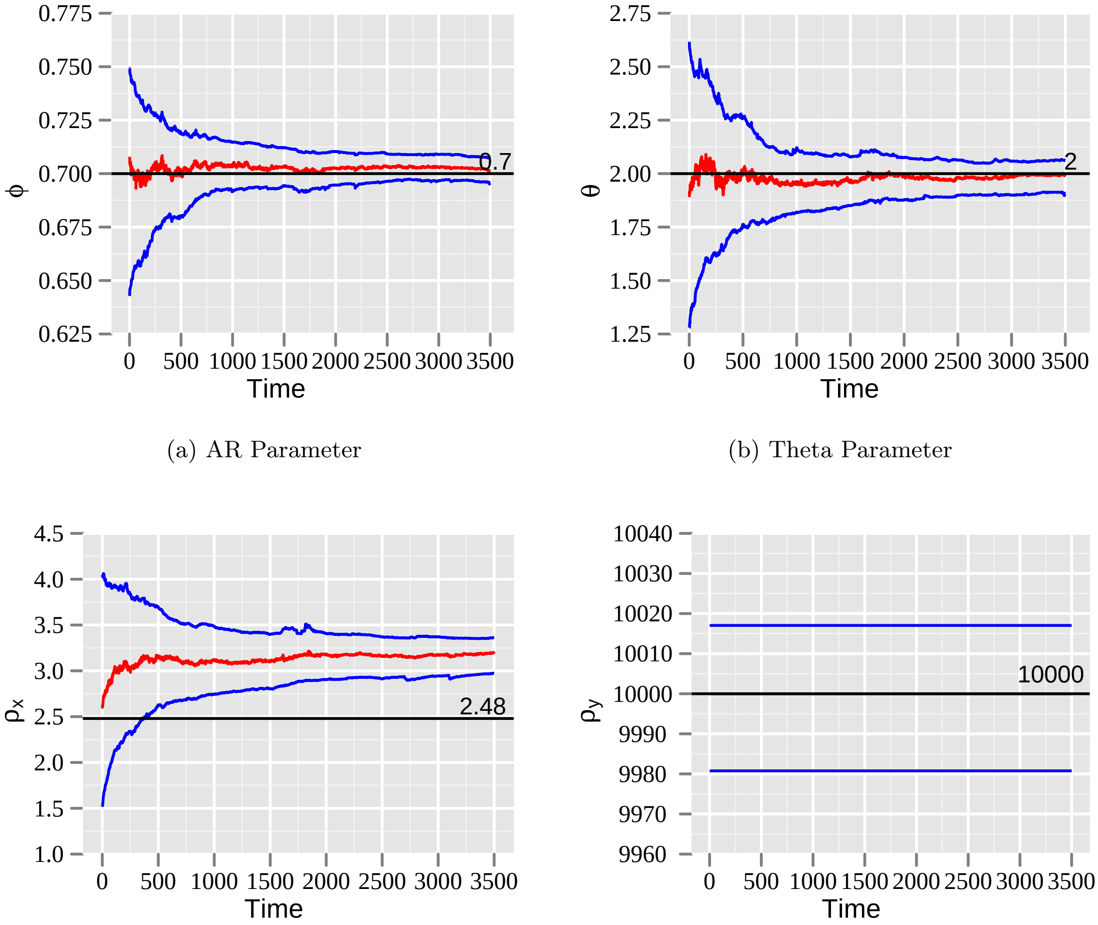


Figure 6.23: An example of the sequential algorithm identifying the true value of the state precision parameter, ρ_x, from the nonlinear model.



(a) AR Parameter

(b) Theta Parameter

(c) State Precision Parameter

(d) Observation Precision Parameter

Figure 6.24: Trace plot of the averaged mode and 95% probability interval for all the four parameters from the nonlinear model. The average is done over all the data sets simulated from this particular model.

seen in Figure 6.24. The actual values of the parameters ϕ, θ and ρ_y lie absolutely within the bounds. The state precision parameter ρ_x however is not well estimated. There is an over-estimation for this parameter. The posterior for all the parameters shrink with time as the algorithm takes its time for the observations to affect the posterior density $\mathbb{P}(\theta \mid \mathbf{Y}_{1:t})$.

6.5.2 Particle filter

A simple particle filter is used for sampling from the posterior distributions of the parameters, conditional on the data at time t. As before, 10 replications of particle filter runs are being done for the model. A re-parameterisation is done for this model too as in Equation (6.4.6), with artificial evolution allowed for the new parameters. Figure 6.25 shows the plot of the means of each of the parameters, at time t for all the replications. It is obvious from the plot that a few of the runs have produced particles which are far away from the real value. This bad estimation varies from parameter to parameter. The worst performance seems to be with the log-σ_y^2 parameter. Even when we compute the probability bounds, the performance of the particle filter does not look promising for this model. Figure 6.26 shows the grand mean of all the replications along with the grand mean of the probability bounds. Other than θ, all the other parameters are ill-estimated, the worst with log-σ_y^2.

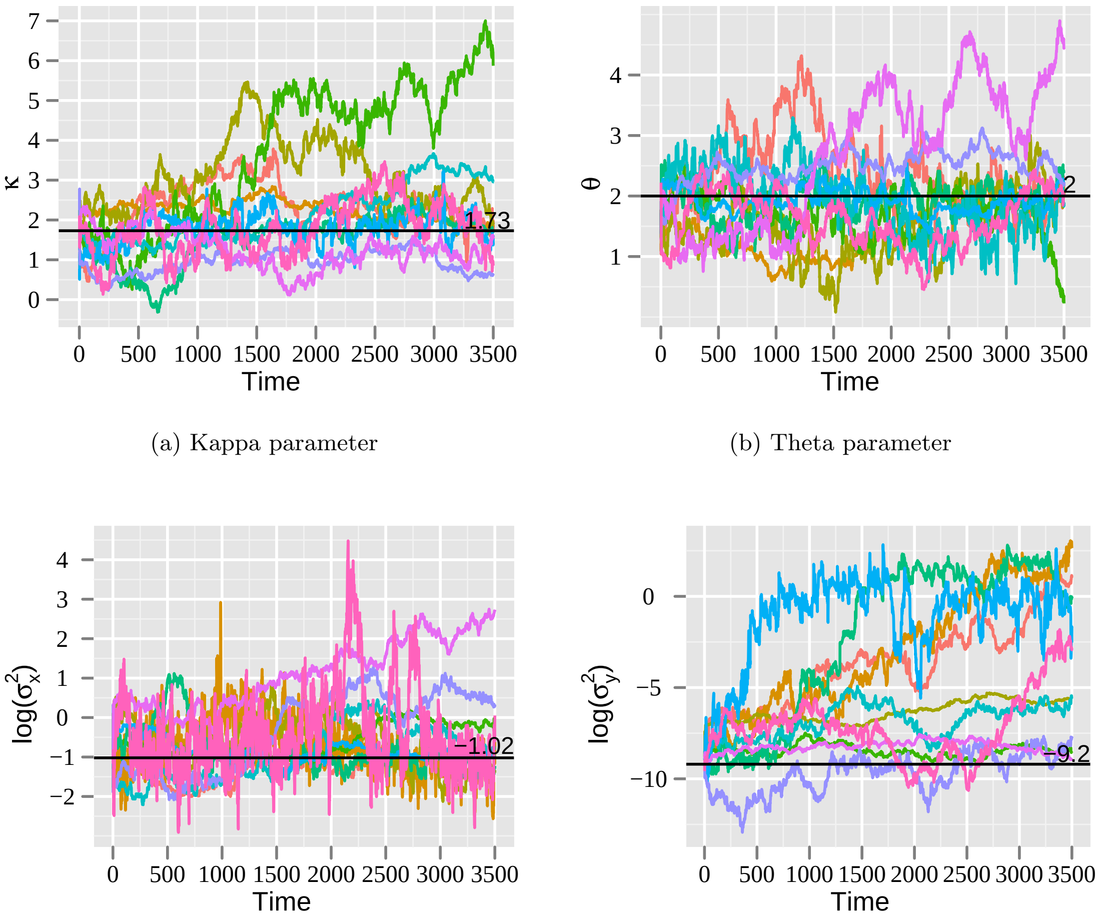
6.5.3 Comparison of the two methods

As have been done in Section 6.4.4 for the linear Gaussian model, the performance of the two methodologies will be compared here through Mahalanobis distance and coverage probability. The performance of the two methods can be seen in Figure 6.27.

From the figure, it is quite obvious that the performance of our algorithm is much better than vanilla SMC. This of course makes sense, since the performance of SMC for this model, as we have seen in the previous plots is not good. For a 4 dimensional parameter space, the χ^2 value is 9.49, under a 5% level of significance. The Mahalanobis distance, for our sequential method is on an average around 6 to 7. That shows a very good performance of our algorithm for the nonlinear model.

It is obvious from the previous plots that the coverage proportions for the SMC algorithm would be small while for sequential INLA they will be higher. Figure 6.28 shows a plot of the coverage proportions. The coverage is much below 0.95 for nearly all the parameters, indicating that the intervals are permissive. While the coverage proportion in particle filters stay below 0.5 for most of the cases, those of our algorithm generally stay above 0.5.

Like in the previous example, we present a plot of computation time versus a measure



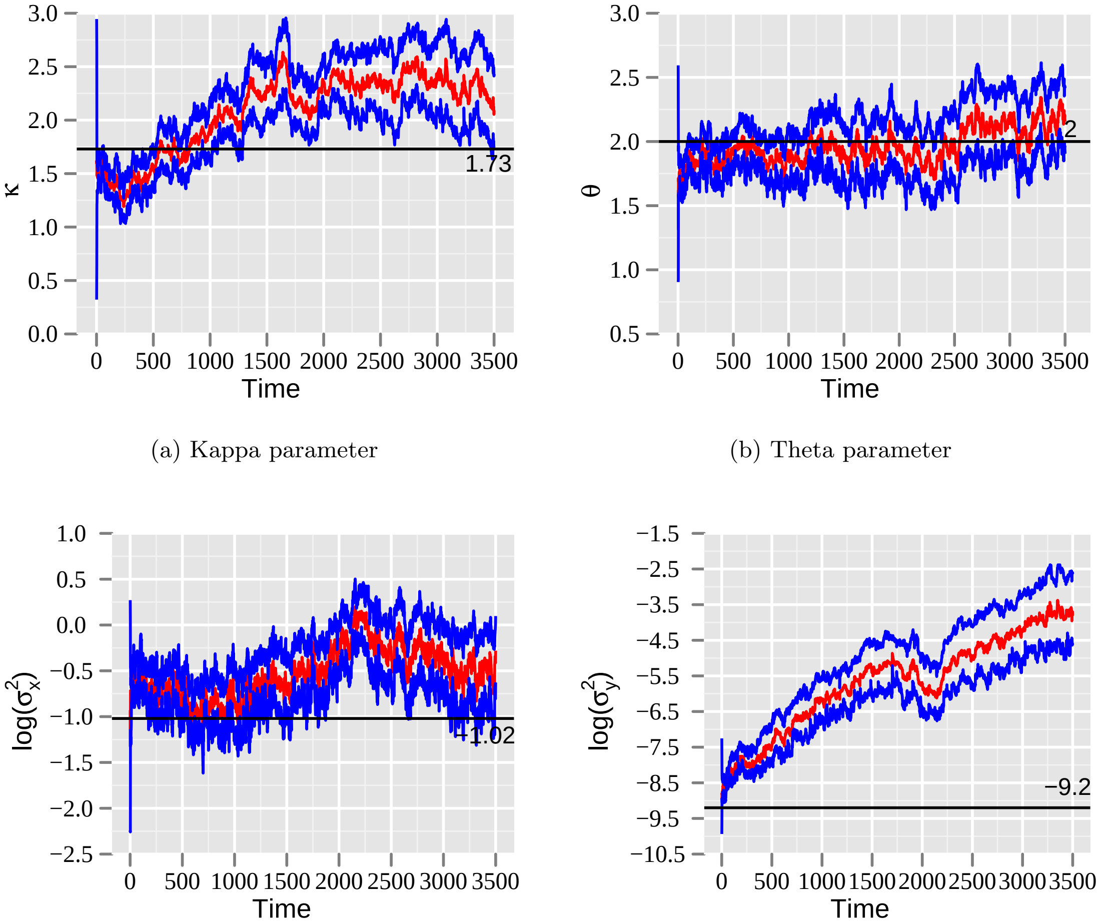
(a) Kappa parameter

(b) Theta parameter

(c) Log state variance parameter

(d) Log observation variance parameter

Figure 6.25: Plot showing mean of the particles generated using SMC, at each time point for all the data sets simulated from the nonlinear model. Some of the runs have produced divergence, where the trace plot has moved away and continue to do so after several iterations.



(a) Kappa parameter

(b) Theta parameter

(c) Log state variance parameter

(d) Log observation variance parameter

Figure 6.26: Plot showing grand mean of the particles for all the parameters of the non-linear model along with 95% probability bounds. The particle filter fails to identify the actual value of nearly all the parameters.

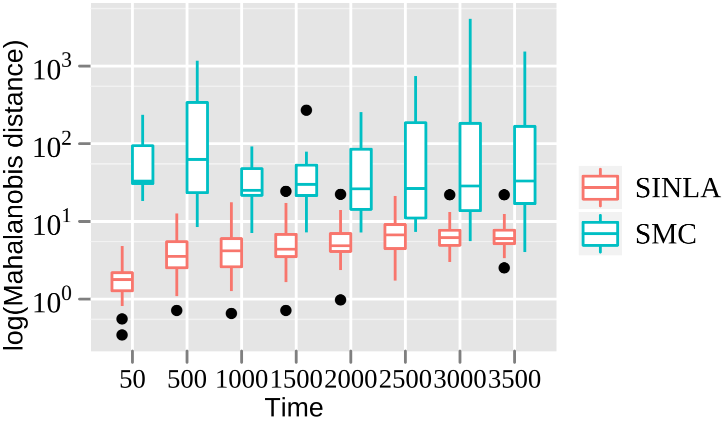


Figure 6.27: Box plot of log of Mahalanobis distance for the two methods applied on the data sets simulated from the nonlinear model. There are many extreme values using SMC, for which the distance is calculated on a log scale.

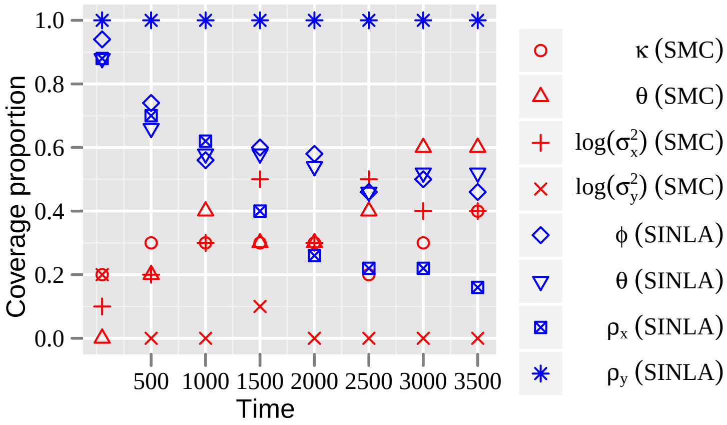


Figure 6.28: Coverage proportion for each of the parameter values of the nonlinear model plotted over time. We start with plotting the coverage proportion for INLA based on the first 50 observations. The performance of both the methods is unsatisfactory as shown in this plot.

of accuracy of estimation, in our case Mahalanobis distance for all the three methods, viz. SINLA, INLA and SMC. Figure 6.29 shows the picture. While the computation time of INLA is quite high but it is extremely accurate (the value of Mahalanobis distance is less than 1), that of the SMC is very bad, even though it is quite fast. However, since we have computed the average of the distance metric, it should be mentioned here that there was a huge outlier. That has definitely affected the value of the accuracy estimate. Our sequential algorithm is quite fast and its accuracy, though not as good as INLA, is much better than SMC.

6.6 Non-Gaussian Model

The final example is that of a model with observations following a Poisson distribution which is dependent on an AR(1) latent process. The model is given by the following equations:

$$y_t \sim \mathcal{P}(\exp^{(6+x_t)}), \tag{6.6.1}$$

$$x_t = \phi x_{t-1} + w_t, \quad w_t \sim \mathcal{N}(0, \sigma_x^2). \tag{6.6.2}$$

The Kalman filter or any extension of it requires the observation equation to be of the form:

$$y_t = f(x_t, \epsilon),$$

where ϵ is the error. Hence Equation 6.6.1 needs to be “suitably” approximated by a different model. Several models were tried for this representation. For most of them, the algorithm was able to identify the AR parameter ϕ but not the state precision parameter

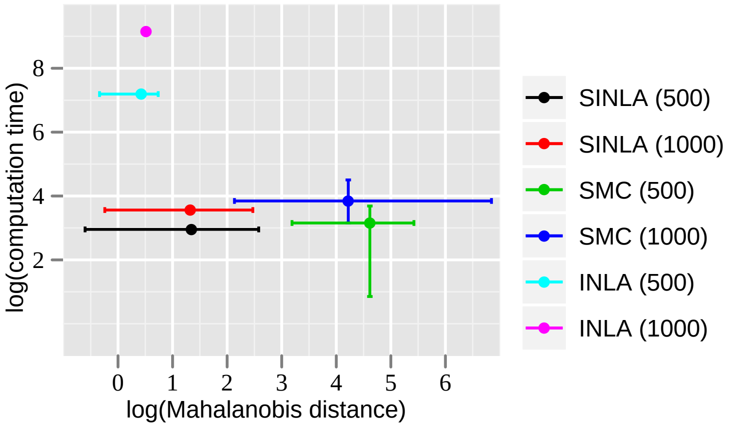


Figure 6.29: Error bar of log-computation time for each algorithm applied on the data sets simulated from the nonlinear model, plotted against Mahalanobis distance computed for time 500 and 1000. The limit of the bars are the 25^{th} and 75^{th} percentile respectively.

at all. Equation 6.6.3 shows some of the representations we tried for the state equation.

$$\begin{aligned} y_t &= \exp^{(6+x_t)} + \eta_t, \quad \eta_t \sim \mathcal{N}(0, \exp^{(6+x_t)}), \\ y_t &= \exp^{(6+x_t)} + \eta_t, \quad \eta_t \sim \mathcal{N}(0, \sigma_y^2), \\ y_t &= \exp^{(6+x_t)} + \eta_t, \quad \eta_t \sim \text{Skewed-}\mathcal{N}(\cdot, \cdot, \cdot). \end{aligned} \tag{6.6.3}$$

Finally it was decided to represent the observation model by the following Equation 6.6.4:

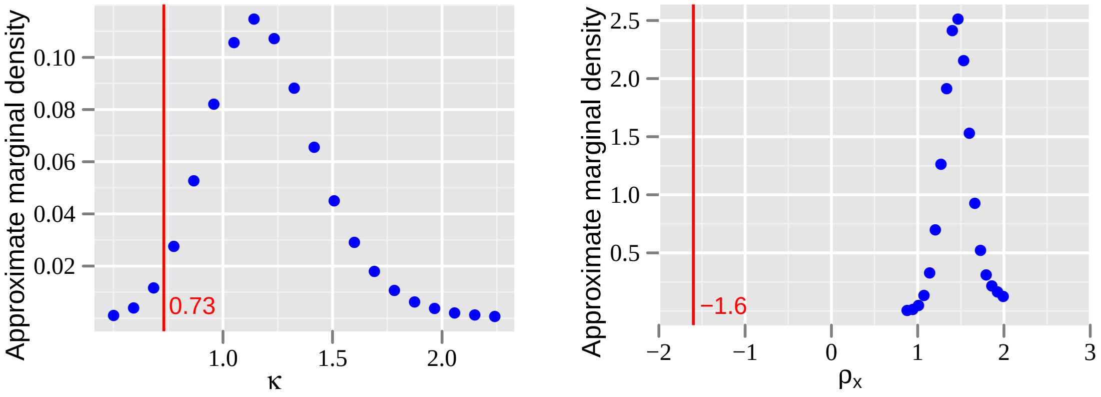
$$y_t = e^{(6+x_t)} \eta_t, \quad \eta_t \sim \log\text{-}\mathcal{N}(0, \sigma_y^2). \tag{6.6.4}$$

The performance of our sequential method was not very good, even with this model. Again the state precision parameter was badly inferred while the inference for ϕ was good. A transformation for parameters was done. We define $\kappa = \text{logit}\frac{(\phi+1)}{2}$, $\log\text{-}\sigma_y^2 = \log(\sigma_y^2)$ and $\log\text{-}\sigma_x^2 = \log(\sigma_x^2)$.

Data were generated with the value of the parameters set at $\phi = 0.35$ and $\sigma_x^2 = 0.2$. Equivalently from the re-parameterisation of the parameters, we have $\kappa = 0.73$ and $\log\text{-}\sigma_x^2 = -1.6$. We are actually interested in these two parameters and not in $\log\text{-}\sigma_y^2$, which is in some sense a dummy parameter for this case.

6.6.1 Method Used: Sequential Update

The sequential update explained in Equation 6.4.4 is used to update the posterior at each grid point for this particular model. The crude method was not tried.



(a) Kappa parameter

(b) Log state variance parameter

Figure 6.30: The posterior marginal distribution for each of the parameters of the non-Gaussian model are plotted along with the true values given by the red line. INLA performs very poorly in identifying the transformed parameters for this specific simulated data set.

6.6.1.1 Computation of $\mathbb{P}(X_t \,|\, \mathbf{Y}_{1:t-1}, \theta)$ and $\mathbb{P}(X_t \,|\, \mathbf{Y}_{1:t}, \theta)$ for Sequential Update

As in the previous nonlinear model, UKF was used to determine the filtering density and 1-step ahead prior of the state process. As before, the sequential update for this model remains:

$$\tilde{\mathbb{P}}\left(\theta \,|\, \mathbf{Y}_{1:t}\right) = \tilde{\mathbb{P}}\left(\theta \,|\, \mathbf{Y}_{1:t-1}\right) \left(\left. \frac{\mathbb{P}_{UKF}\left(X_t \,|\, \mathbf{Y}_{1:t-1}, \theta\right) \mathbb{P}\left(\mathbf{Y}_t \,|\, X_t, \theta\right)}{\mathbb{P}_{UKF}\left(X_t \,|\, \mathbf{Y}_{1:t}, \theta\right)} \right|_{X_t^*(\theta)} \right), \tag{6.6.5}$$

where X_t^* is the mean of the latent process at time t No correction factor needed to be used for this model.

INLA was implemented iteratively on the first 100 observations for this particular model. Since all three parameters associated with this model viz., κ, log-σ_y^2 and log-σ_x^2 have the real line as their support, a Gaussian prior was given to all the three parameters. The parameters of interest for us are κ and log-σ_x^2. So we start by taking a look at the posterior marginal distribution of them as given in Figure 6.30. INLA, as one can see here, completely fails to identify the actual parameter value of the model even though it uses a considerable number of observations for this model. Although, this figure represents the performance of INLA on a certain data set, the algorithm performs similarly for all the data sets for the parameter log-σ_x^2. The other parameter κ is quite well identified in certain data sets. An example of such a grid as computed by using INLA is shown in Figure 6.31.

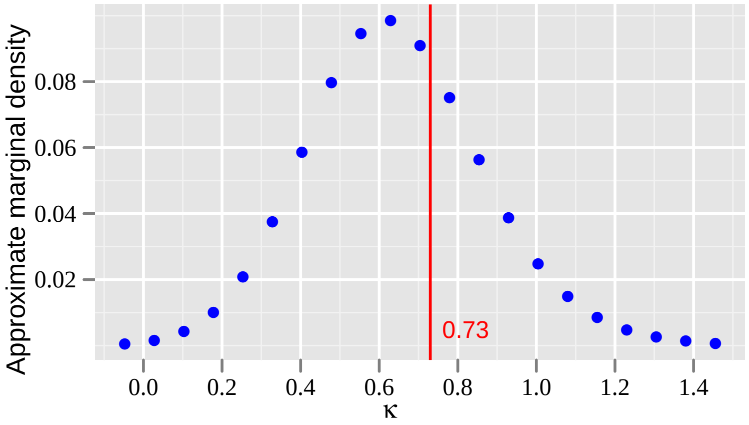


Figure 6.31: Good identification of the actual value of the transformed parameter, κ as seen from the plot of the posterior marginal density, computed using INLA.

6.6.1.2 Performance of the Sequential Algorithm

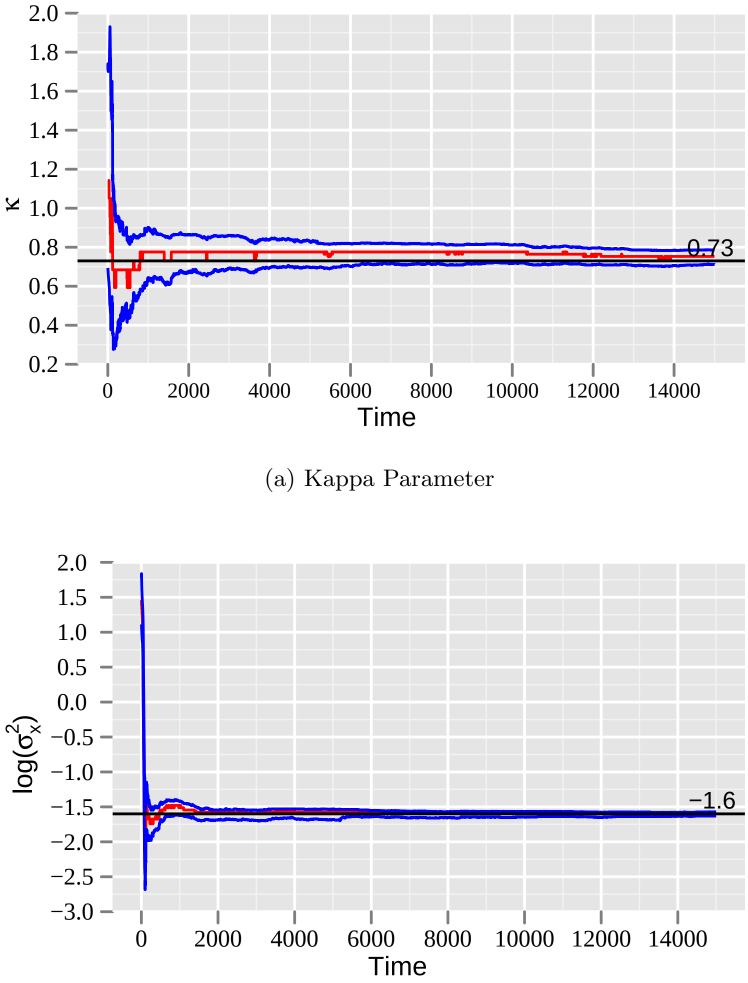
The performance of the algorithm for this model is almost always very good. Figure 6.32 shows the performance of the algorithm in estimation of the unknown parameters for a particular data set. For this particular data set, the two parameters κ and log-σ_x^2 are estimated almost exactly and are well within the intervals.

As has been done for the previous two models, 50 data sets were simulated from the non-Gaussian model. Figure 6.33 shows the results for the average of the mode and 95% probability intervals of each of the parameters. It is quite obvious from the plots that the sequential algorithm identifies the true values very well on average for both the parameters. The starting grid provides a posterior well away from the true value of the parameters. However, the grid update algorithm adds grid points externally and finally converges to a posterior density which does contain the actual values. It is also evident from the plot that the posterior density shrinks by a great deal.

6.6.2 Particle filter

One big advantage of the particle filter is that it allows us to sample directly from the observation and state model, and that there is no need for the models to be represented in any particular form as in Equation 6.6.4. Thus Equation 6.6.1 remains in its original form, and the non-Gaussian example model is defined as:

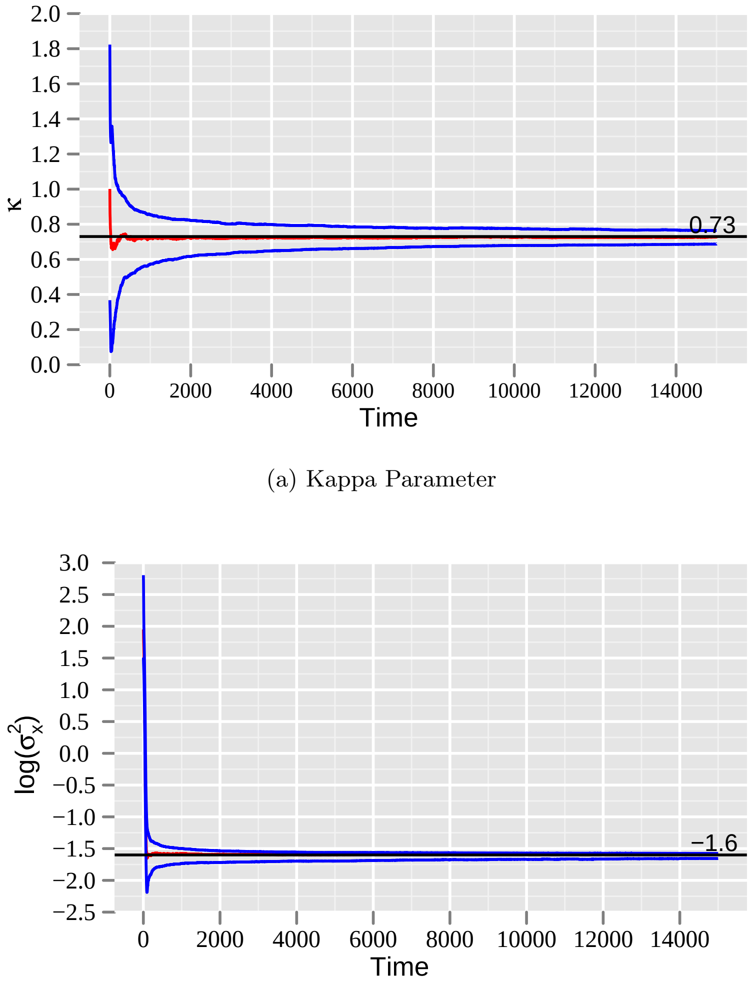
$$y_t \sim \mathcal{P}(\exp^{(6+x_t)}),$$
$$x_t \sim \mathcal{N}(\phi x_{t-1}, \sigma_x^2)$$
$$x_0 \sim \mathcal{N}(0, \frac{\sigma_x^2}{1-\phi^2}).$$



(a) Kappa Parameter

(b) Log State Variance Parameter

Figure 6.32: Plot of mode and 95% probability interval for both of the transformed parameters of the non-Gaussian model over time for a certain simulated data set.



(a) Kappa Parameter

(b) Log State Variance Parameter

Figure 6.33: Plot of the averaged mode and 95% probability interval for both of the transformed parameters from the non-Gaussian model, over time. The average is done over estimates of the statistics calculated over all the simulated data sets.

The two unknown parameters here are $\Theta = (\phi, \sigma_x^2)$. To use ordinary sequential Monte Carlo on this model to estimate the parameters, as always we will have to put artificial evolution into the parameters. And for that, a re-parameterization is necessary. We have already done such parameterizations in the previous two examples, and κ and Log-σ_x^2 are defined the same as before. Thus Θ is redefined as the vector of the two parameters κ and Log-σ_x^2. The observation and state models are further extended by the following:

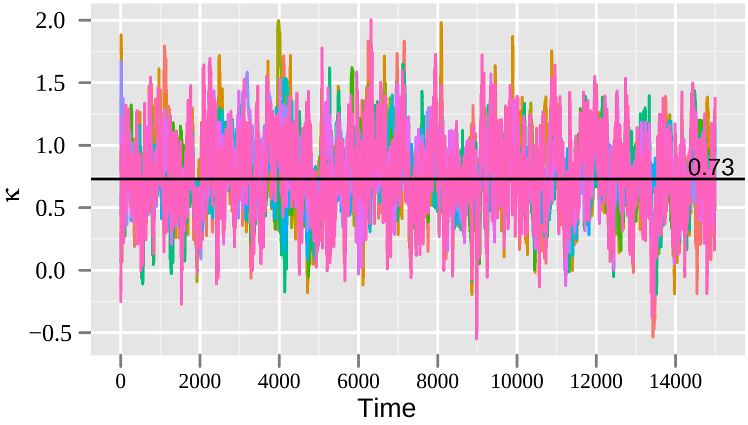
$$\Theta_t \sim \mathcal{N}(\Theta_{t-1}, \Sigma),$$
$$\Theta_0 \sim \mathcal{N}(\mathbf{C}, \Sigma),$$

where Σ is the variance matrix and $\mathbf{C}$ is some constant value, both of which will be provided by the user. In real life applications, Σ is chosen to ensure low variability, especially for the static parameter situation. C is generally a vector of 0's, or whatever suits the application.

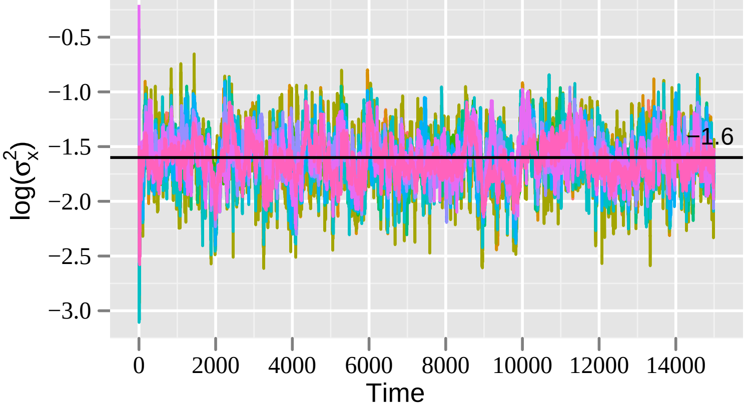
Using the above model, the sequential Monte Carlo algorithm was applied several times on a single data set simulated from the non-Gaussian model example. Figure 6.34 plots the means at each time point for every replication that we have done with the SMC method. As is expected, not all the traces converge to the true value but overall it seems to be doing well. A better idea can be had from plotting the average of the means and the credible intervals. As intervals, we have just plotted the average of 2.5% and 97.5% percentiles. The plot of these two, as can be seen in Figure 6.35, clearly shows that on an average the performance of the filter is quite good in detecting the true parameter values. Although, the mean has not totally converged to the true values, the intervals always cover the true values.

6.6.3 Comparison of the two methods

As for the previous two models, methods of accuracy and performance measures are computed and compared for both the algorithms. The Mahalanobis distance for the two methods are plotted on the same graph. Different parameterizations are done for each of the algorithms, but that should not be a problem in assessing an algorithm's accuracy. Figure 6.36 shows the plot of the distance metric at regular time points. The distance metric for our sequential method is very high for time 100, i.e. for the output provided by INLA based on 100 observations. This is quite evident from the sequence plots shown before, for example in Figure 6.33. There were a couple of outliers with very high values among the means computed for different simulated data sets. It was found that for a specific simulated data set, our method diverged, resulting in extreme values for the estimates. They have been retained in the plot to facilitate a better understanding of the relative performance of these two methods. The particle filter has done much better for this model, as one can clearly see. Under Gaussian approximation for the parameters, the Mahalanobis distance follows a χ_2^2 distribution. For 5% level of significance, the value of

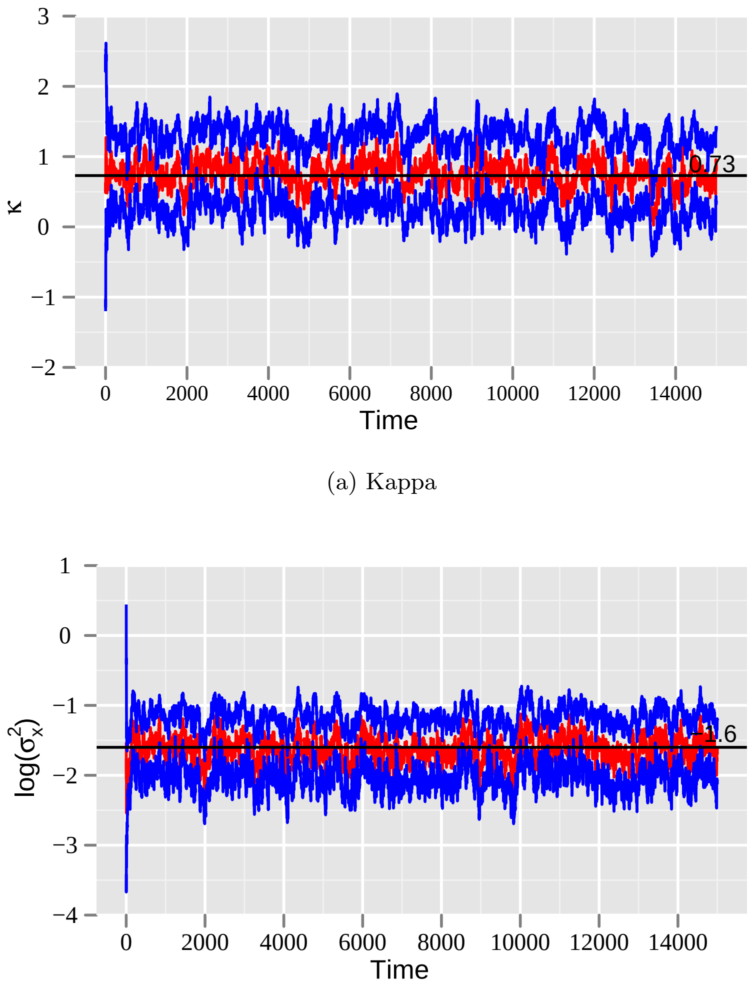


(a) Kappa



(b) Log State Variance Parameter

Figure 6.34: Plot of the mean of the two transformed parameters of the non-Gaussian model, over time, with each trace denoting a certain simulated data set. Particle filter is quite successful in identifying the true values of the parameters.



(a) Kappa

(b) Log State Variance Parameter

Figure 6.35: Trace plot of the average of the mean of the two transformed parameters of the non-Gaussian model. The average has been done over all the data sets simulated from this model.

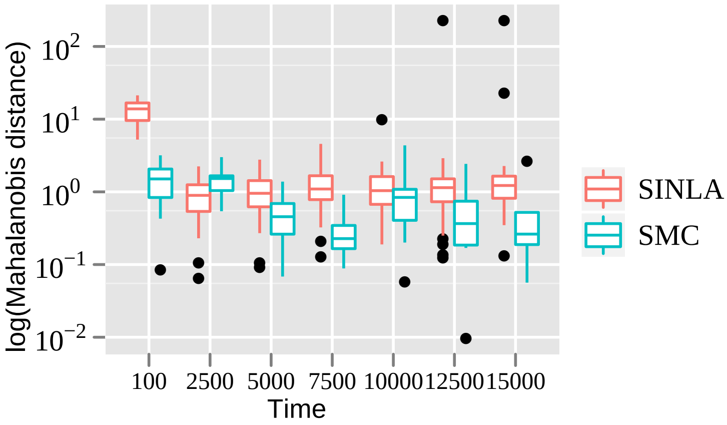


Figure 6.36: Box plot of log of Mahalanobis distance for the two methods applied on the data sets simulated from the non-Gaussian model. The values are plotted on a log-scale because of the presence of outliers for the sequential INLA method. It has been observed that our algorithm diverged for a specific simulated data set, resulting in extreme values of the mode.

χ^2_2 is 5.99. Even though our parameters obviously do not follow a normal distribution, it is quite clear that the distances are well below the χ^2 value.

Similarly, coverage proportion of the parameters are plotted for both the methods. Figure 6.37 shows the performance of the two algorithms. They are nearly equivalent in their performance, except for time 100 in the case of our sequential method, where INLA provided a bad estimate of the κ parameter. Thus for this model and the two parameters that we estimated, the performance of our method and particle filter is almost the same.

Figure 6.38 shows the plot of computation time versus the averaged Mahalanobis distance for the three methods, as in the previous examples. As far as computation time is concerned, it is still the same here with INLA taking a lot of time as expected, while the on-line methods are very fast. INLA and SMC performs better here with respect to the Mahalanobis distance.

6.7 Conclusion

The performance of our proposed sequential algorithm for inference on static parameters seemed to be satisfactory, given the results presented in the previous sections. The method has been extensively tested on several data sets simulated from three different models. The results when compared to that of the vanilla particle filter proved to be very good. Comparison to INLA has also been done to check the accuracy of its performance, when compared to a static algorithm.

Furthermore, the correction factor as discussed in Chapter 5 has also been implemented in the algorithm for one of the more complex models, namely the nonlinear model. It

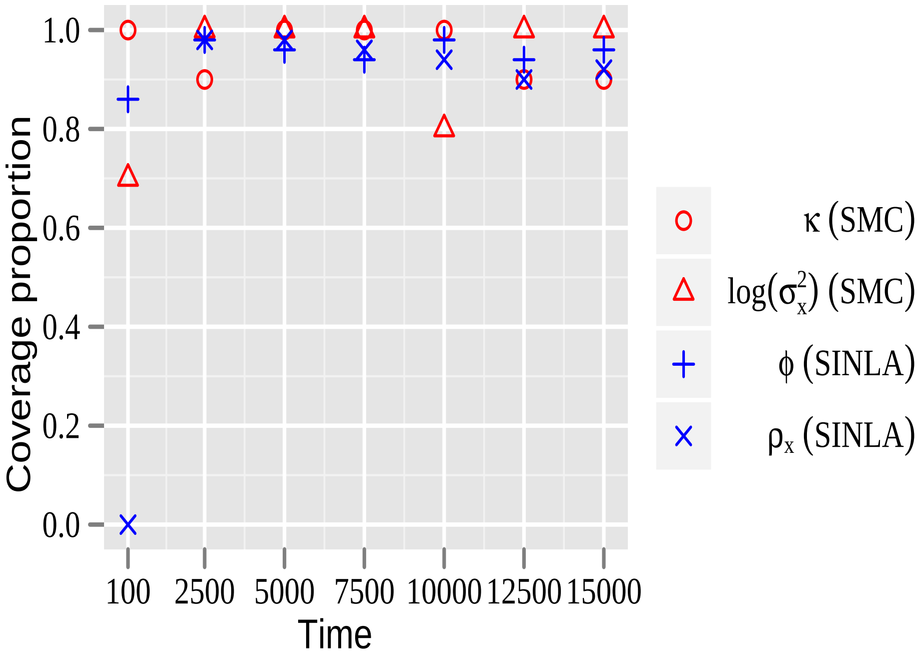


Figure 6.37: Coverage proportion for each of the parameter values plotted over time for the non-Gaussian model. We start with plotting the coverage proportion for INLA based on the first 50 observations.

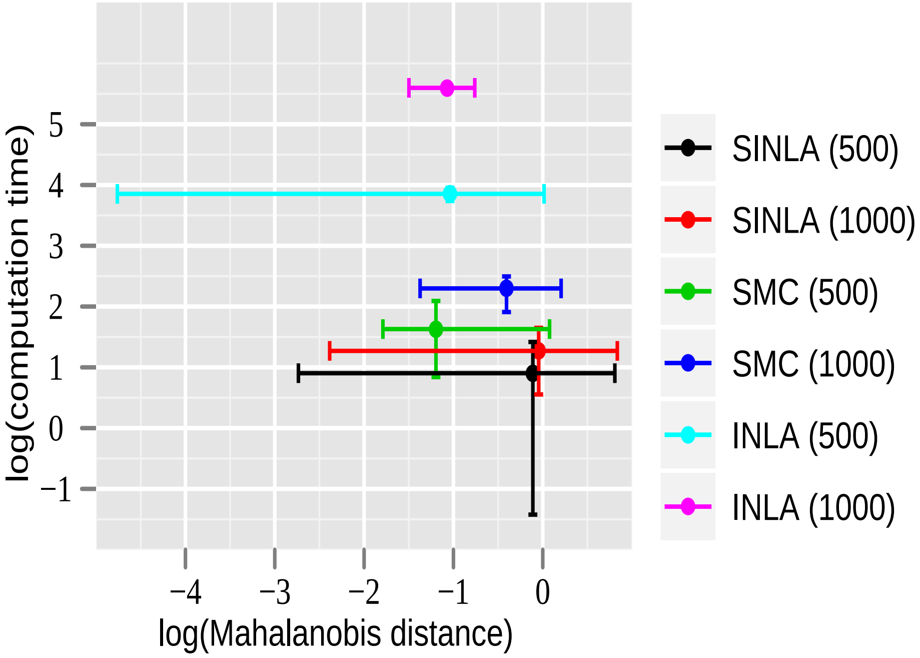


Figure 6.38: Error bar of log-computation time of each algorithm applied on the data sets simulated from the non-Gaussian model, plotted against Mahalanobis distance computed for time 500 and 1000. The limit of the bars are the 25^{th} and 75^{th} percentile respectively.

successfully stopped the occurrence of collapse of the posterior density values to a single point.

The overall performance of our algorithm has been quite satisfying. The fact that this algorithm is very easy to implement and keeps a good balance between accuracy and speed, shows that it can be applied to a wide range of real life application. SINLA has been tested further using these three models, by changing the true parameter values in the model. The results of the different test cases are shown in the Appendix. In the next section, we conclude our thesis and discuss future work which can be possible to improve our methodology.

Chapter 7

Conclusion and Further Work

The motivating application of the research contained in this thesis concerns sequential estimation of static parameters in a dynamic state-space model. There are many challenging features of this problem that involves producing accurate inference, while maintaining speed of the algorithm. In seeking to address these issues, several important research contributions have been made. In the following we summarise these contributions and reflect on several conclusions regarding deterministic approximation to the posterior of static parameters in a dynamic state-space models.

7.1 Conclusion

Dynamic state-space models form a rich class of models that can be applied in a wide range of applications in physics, economics, genetics etc. Real time inference is necessary for the unknown state process and parameters involved with the state and observation process. The area of parameter estimation have not been rigorously looked into, until recently, even though a huge volume of literature exists for estimation of the state process. In this thesis we are particularly interested in inference on "static" parameters. A strong nonlinear relationship with the observations and non-Gaussian distribution (generally) of static parameters make sequential estimation an extremely difficult task to perform.

A new approach for Bayesian inference to dynamic state space models for static parameter estimation in real time has been proposed in the thesis. Our approach is neither based on sampling, nor an extension to the well-known Kalman filter method, and is inspired from the work of Rue et al. (2009). The new method constructs, in real time, the posterior of a vector of static parameters on a discrete grid. It has been named Sequential INLA (SINLA). The grid is allowed to adopt to changes in the support of the posterior and adjust accordingly, as new data affects the support of the posterior over time. In view of the application of SINLA on data sets simulated from different models as example, we conclude that our method has distinct advantages over the existing methods, both in terms of accuracy and computational speed. It is also easy to see that our method can be used along with any existing filtering method like Kalman filter, quadrature filter or particle filters, making it a very useful tool in very complex applications. Furthermore, our approach does not make any distributional assumptions for the parameters and also do not introduce any artificial evolution into the parameters. Application to three different examples starting from the linear Gaussian model to models with nonlinearity and non-Gaussian noise have revealed the strength of this method.

However, the weaknesses of the proposed method are also quite evident. Firstly, the restriction on the number of parameters for which SINLA is applicable, greatly confines the relevance of this method in several application areas, say robotics, where the parameter space may run into thousands (Thrun 2002). If the parameter space is between 10 and 15 our approach can be applied given that the resources available for computation is sufficiently provided. This bottleneck does not exist for Kalman filter based methods since they assume a Gaussian distribution and hence only need to propagate the first two moments. A Rao-Blackwellisation like approach might be needed in such situations, which could help the user use our approach for a restricted number of parameters.

Another problem is the construction of the initial grid. Our approach, for the time being, depends quite heavily on INLA for the construction of the initial grid. INLA, even though quite varied in its applicability, is not the suitable approach for a wide range of models. If the state process is not a GMRF, then the performance of the Gaussian approximation will be very bad. Similarly, for a multi-modal distribution, Gaussian approximation will give inferior results. However, INLA can still be used since it is only necessary for the construction of the grid and the actual performance of our method does not absolutely depend on INLA. It has been mentioned in Chapter 4 that any other approximation can also be used for our method as long as it is computationally fast.

The biggest problem in our approach is the computation of posterior values and any other required statistic for the new points added to the support of the existing grid. Linear interpolation, while simple and quick to perform, does not work well in all situations. It has been seen that in the presence of strong nonlinearity in the model, linear interpolation gives poor results. It is not quite clear how to compute the statistics at the new grid points for a multivariate state process. For example, it would be difficult to compute the covariance matrix of $\mathbf{X}_t \,|\, \mathbf{Y}_{1:t}, \theta^*$ at a new grid point θ^*. The current method of individually computing the values of the covariance matrix using interpolation may prove to be inefficient. Also there is no way to ensure that the computed covariance matrix will be positive semi-definite. These problems need to be worked on as possible future work.

Our approach of course leaves provision for the usage of any method the user finds to be suitable. Nonlinear interpolation or fast nonparametric smoothing techniques can be used. A good smooth fit will always give much better results than linear interpolation.

But as we have mentioned earlier, the simplicity of linear interpolation makes it easy to implement. For high dimensional parameter space, it is not exactly clear how difficult it is to implement nonparametric smoothing or nonlinear interpolation. Linear interpolation is very easy to apply even for very high dimensions.

Significant progress has been made in the area of sequential estimation of static parameters when the data set is collected over a long period of time. Whereas methods like particle filter are destined to fail if applied for a long period of time ultimately falling victim to degeneracy, such problems do not arise for our method. A problem similar to degeneracy has been identified in one of our examples, and has been diagnosed to be due to outlier values in the data set. Though a correction factor has been proposed, the performance of the algorithm is less than desired, indicating further research into the problem. Furthermore, it is not exactly clear if the solution provided in this thesis will work for other situations or not. This area needs a bit of testing on complex models.

The application of our approach on different models has provided considerable information regarding accuracy of this method. For a linear Gaussian model, computation of the posterior of parameters was exact due to the use of Kalman filter in the sequential multiplier. Not that the distribution for the posterior of parameters is not Gaussian, even though the model is linear with Gaussian errors. Even for nonlinear and/or non-Gaussian models, the error committed due to approximations made can be controlled, depending on the algorithm used for the calculation of the filtering and 1-step ahead prediction process. Similarly any divergence issues is also entirely dependent on the filter used for making inference on the state process. This issue has been discussed in the next section.

Evidence for these conclusions are provided by the application of SINLA on three different models of varying complexity. This method has also been implemented further on models with different parameter values. The results for the latter have been provided in the Appendix.

7.2 Future Work

Whilst the research presented in this thesis has contributed substantially in the area of sequential inference of static parameters in state-space models, several outstanding challenges remain. In the following sections, the nature of these challenges are briefly outlined:

7.2.1 Restriction on the number of static parameters

One of the most important drawback for SINLA is that there exists a restriction (in a practical sense) on the number of parameters to be estimated. Of course with increasing computational power, the number can be increased while still maintaining a high speed of the algorithm. Using graphics cards as computation devices can be the next big step forward, which enables efficient GPU programming becoming popular now. That also should enable us to better parallelise the algorithm which will allow more parameters to be estimated in real time. However, increasing computer power can only allow us to include a few more parameters since the number of grid points grow exponentially with dimension. There is a need to look into this problem. For models where the number of *parameters of interest* are within our acceptable limit (set out for SINLA) and the rest of the parameters can be considered to be nuisance parameters, our method can be used. The nuisance parameters can be estimated using the optimization algorithm used in INLA and their estimate can then be used in the model, while SINLA can be used to estimate the *effective number of parameters.*

7.2.2 Convergence of the filter

We have not shown any consistency properties for the filter. There is a need to develop asymptotic properties related to convergence of the filter. There exist really strong convergence properties for MCMC based methods, but very little for approximate Kalman filter extensions.

Our method starts with applying INLA to the first few observations. The accuracy of the posterior of $\theta \mid \mathbf{Y}_{1:t}$ seems to be directly related to the dimension of the latent process as shown by Rue et al. (2009). For a fixed number of $\mathbf{X}$ and parameters, the posterior, up to a normalizing constant, has an error rate of $\mathcal{O}(n^{-1})$, where n is the sample size. This is exactly same as the Laplace approximation of a marginal distribution (Tierney & Kadane 1986). Moreover, it should be noted that in most cases normalizing the approximate density reduces the asymptotic rate, as the dominating terms in numerator and denominator cancel. Tierney & Kadane (1986) show how the normalized approximate marginal density has an error rate of $\mathcal{O}(n^{-2})$, whereas the numerator/denominator has an error rate of $\mathcal{O}(n^{-1})$. One can look up into the convergence properties discussed in *Laplace Gaussian filter*, where the error rate has been proved to be $\mathcal{O}(n^{-\alpha})$, where α is 1 or 2 depending on the order of the Laplace approximation used. If this filter is used, the error rate remains fixed at $\mathcal{O}(n^{-\alpha})$ over time as has already been shown by Koyama et al. (2010).

The new approach provides an identity to compute the posterior density of the parameters. In our thesis we have used UKF as the filtering method for state process in two of the examples. A Cramér-Rao lower bound has been provided by Xiong et al. (2006). So any bound for the posterior constructed by our method will again be dependent on their bound. The most important thing to note is that the normalized posterior density, $\mathbb{P}(\theta \mid \mathbf{Y}_{1:t})$, has a reduced asymptotic error rate than that of the methods used for state filtering, since there is a possible cancellation happening between the denominator and the numerator. However this idea needs to be studied extensively.

7.2.3 Computation of log-posterior and necessary statistics at new grid points

The computation of the posterior log density at new grid points in the parameter space is done using linear interpolation techniques. We have chosen that because of its simplicity and in the process has sacrificed accuracy to some extent. There could be other methods which work better at almost the same speed. Maybe a high dimensional curve fitting method will do just as good. It needs to be looked into quite seriously, since the algorithm can be vastly improved if there is some method which performs well. One can think of resorting to curve fitting methods not over the whole density but only locally. That will provide very accurate results while keeping the computational time within desired limits. A default bandwidth calculation method that provides good approximation will also help this process.

One can also make use of sampling to compute these values locally. Monte Carlo samples can be drawn locally from the existing posterior density, and these samples can be used to compute the posterior density at a particular value of θ. Something similar can also be done for computation of the statistics from the filtering and 1-step ahead density of the state process, by utilising the interpolated value of those statistics at the new grid point as a central value.

7.2.4 Application on real life problem

There is an urgent need for the application of this method on data collected from a real life problem. That would truly test the approach to the full. The authors are presently applying this algorithm on a dynamic classification problem in time series data.

7.3 Final Comments

The new method Sequential INLA provides posterior inference about static parameters in dynamic state-space models. Successful application of this approach has been shown in many examples. Multiple data sets have been simulated from these models and the algorithm has been applied to these data sets. The performance of this method on these models have been extremely encouraging, leading to further research into the topic.

Appendices

Appendix A

Linear models

The linear model used as the first example in Chapter 6 has been re-used here. We have simulated several data sets by varying the value of the parameters for an extensive check for our algorithm. For each set of parameter values, we have generated a few data sets. SINLA has been applied on these data sets to check for the performance of the algorithm. Finally a consolidated trace plot of average values of the approximate mode and 95% probability bounds are presented for each set of parameters.

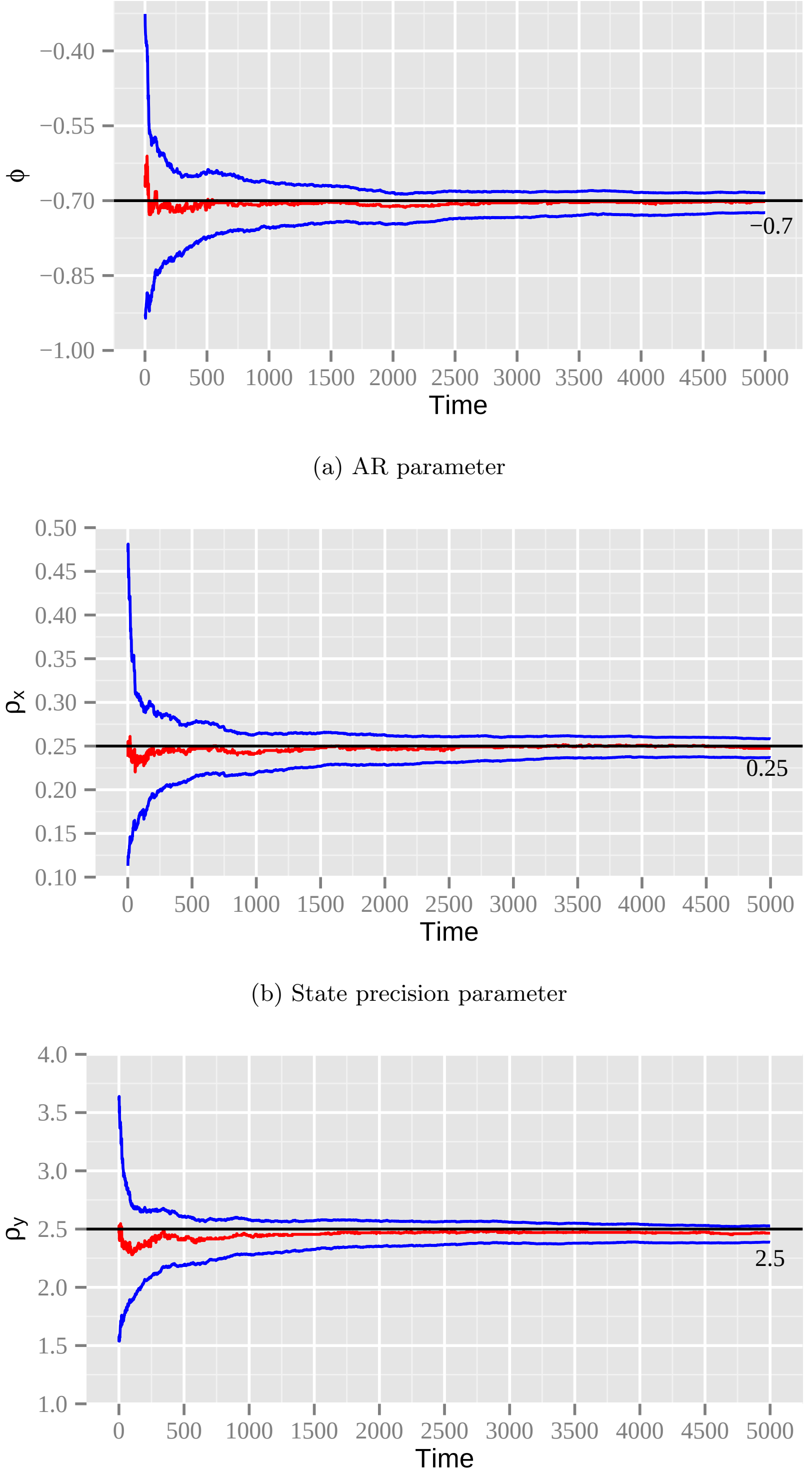
We have generated data sets using values of the parameter which lies at the edge of their support or have values which could prove to be a computational nuisance. The performance for all the cases have been excellent. The results have proved to be very encouraging and goes on to show that our approach can be applied on a wide range of linear models.

A.1 Example 1

Data has been simulated using the linear model with vector of parameters given by $\phi = -0.7$, $\rho_x = 0.25$ and $\rho_y = 2.25$. The average of the approximate mode and probability bounds are plotted over time and the true value of the respective parameter is also provided. Figure A.1 shows the plot with the red trace indicating the mode at time t and blue lines indicating the confidence bounds. It is very obvious that our approach has successfully identified the true parameter values.

A.2 Example 2

In this example, the true parameters of the linear model are given by $\phi = 0.001$, $\rho_x = 1$ and $\rho_y = 10$ respectively. SINLA has determined the true values in this example too, as can be seen in Figure A.2.

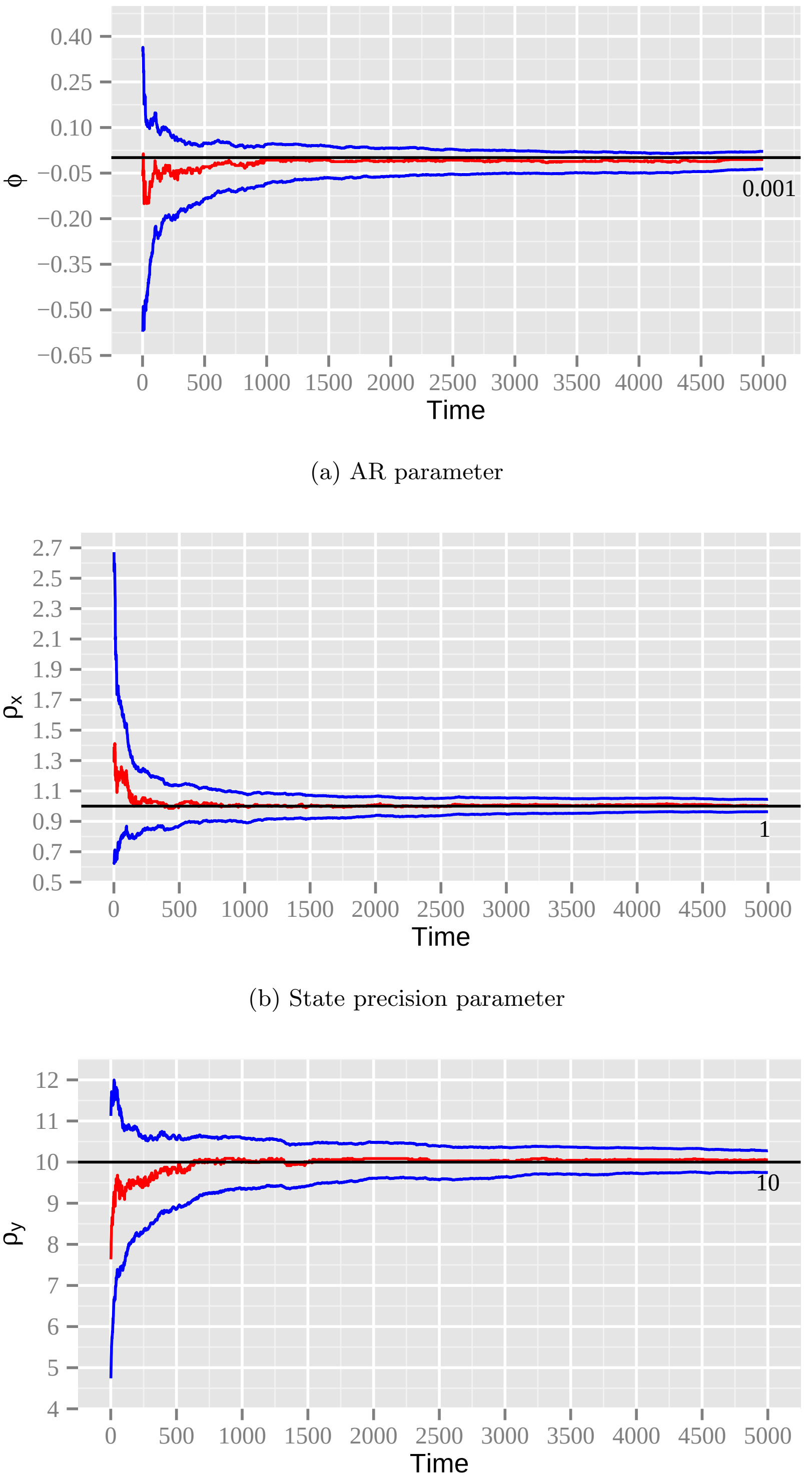


(a) AR parameter

(b) State precision parameter

(c) Observation precision Parameter

Figure A.1: Plot of the mode and 95% probability interval for all the three parameters over time for the linear model in Example A.1. The actual value of the parameters perfectly fall within the bounds indicating successful identification of the parameter by our approach.



(a) AR parameter

(b) State precision parameter

(c) Observation precision Parameter

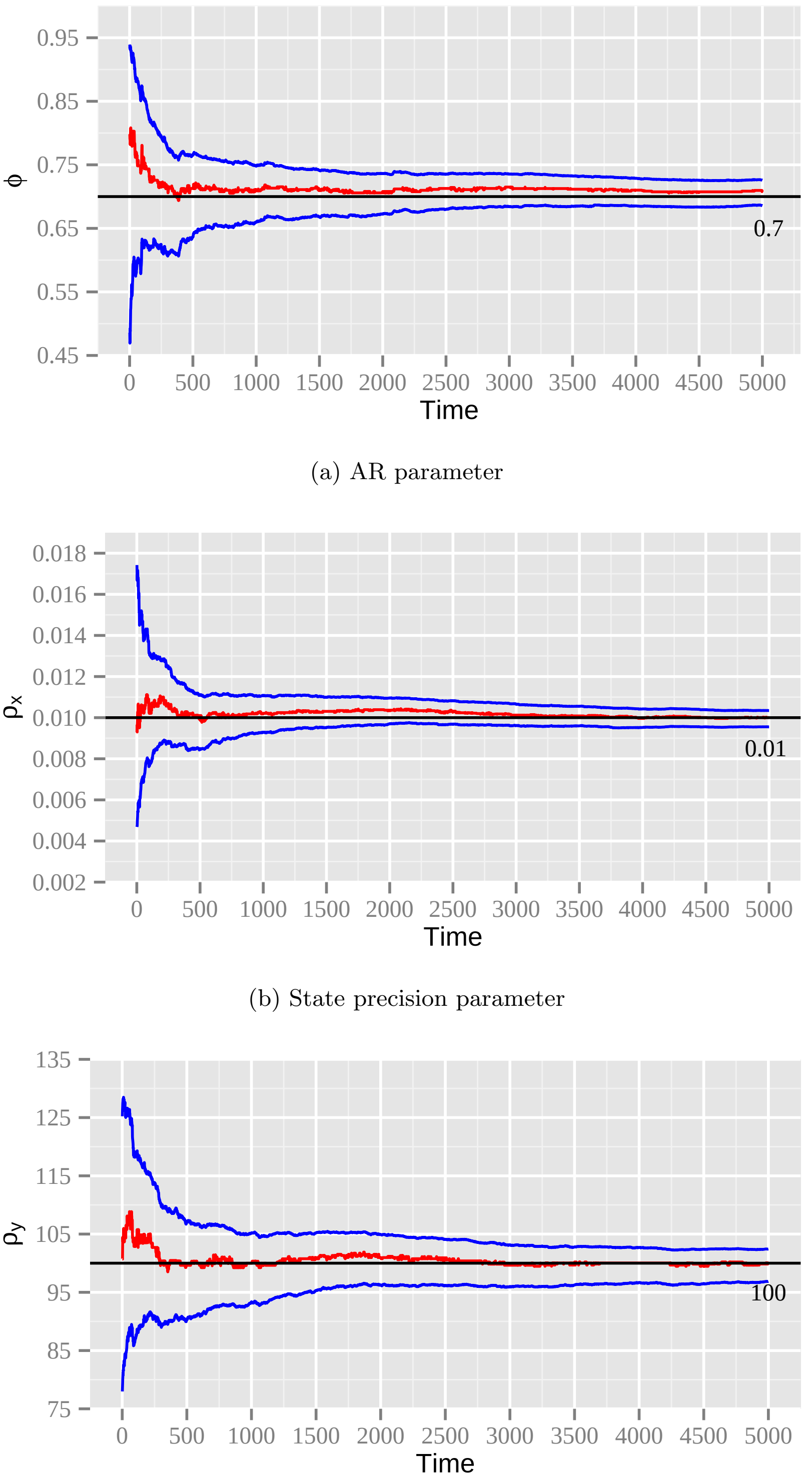
Figure A.2: Plot of the mode and 95% probability interval for all the three parameters over time for the linear model in Example A.2. The actual value of the parameters perfectly fall within the bounds indicating successful identification of the parameter by our approach.

A.3 Example 3

The true parameter values for the 3^{rd} example are $\phi = 0.7$, $\rho_x = 0.01$ and $\rho_y = 100$ respectively. Again, our approach has been successful in providing good estimates for the parameters as is evident in Figure A.3.

A.4 Example 4

For the final example, the true parameter values are given as $\phi = 0.99$, $\rho_x = 1$ and $\rho_y = 10$ respectively. Figure A.4 shows the performance of our sequential method. It is clear that the inference is excellent even though the value of the AR parameter is very close to 1.

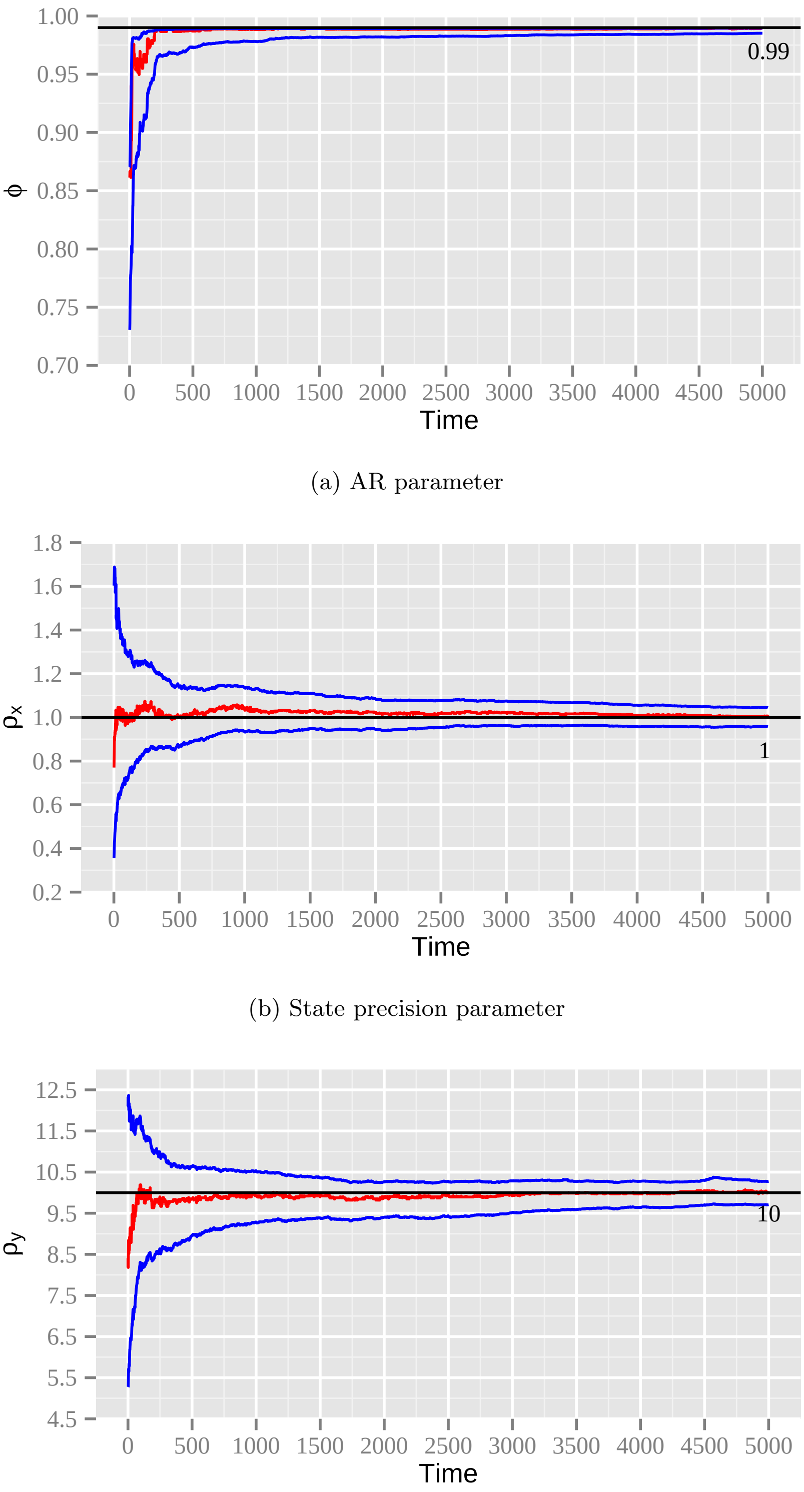


(a) AR parameter

(b) State precision parameter

(c) Observation precision Parameter

Figure A.3: Plot of the mode and 95% probability interval for all the three parameters over time for the linear model in Example A.3. The actual value of the parameters perfectly fall within the bounds indicating successful identification of the parameter by our approach.



(a) AR parameter

(b) State precision parameter

(c) Observation precision Parameter

Figure A.4: Plot of the mode and 95% probability interval for all the three parameters over time for the linear model in Example A.4. The actual value of the parameters perfectly fall within the bounds indicating successful identification of the parameter by our approach.

Appendix B

Nonlinear models

Similar to what has been done in A, data sets have been simulated from multiple nonlinear models of the type presented in Chapter 6. We present here two examples where our algorithm has been applied on data generated from different models. The basic nonlinear model that we have used in this thesis is presented here once again:

$$y_t = \theta x_t^2 + v_t, \tag{B.0.1}$$

$$x_{t+1} = 4 + sin(\omega \pi t) + \phi x_t + w_t, \tag{B.0.2}$$

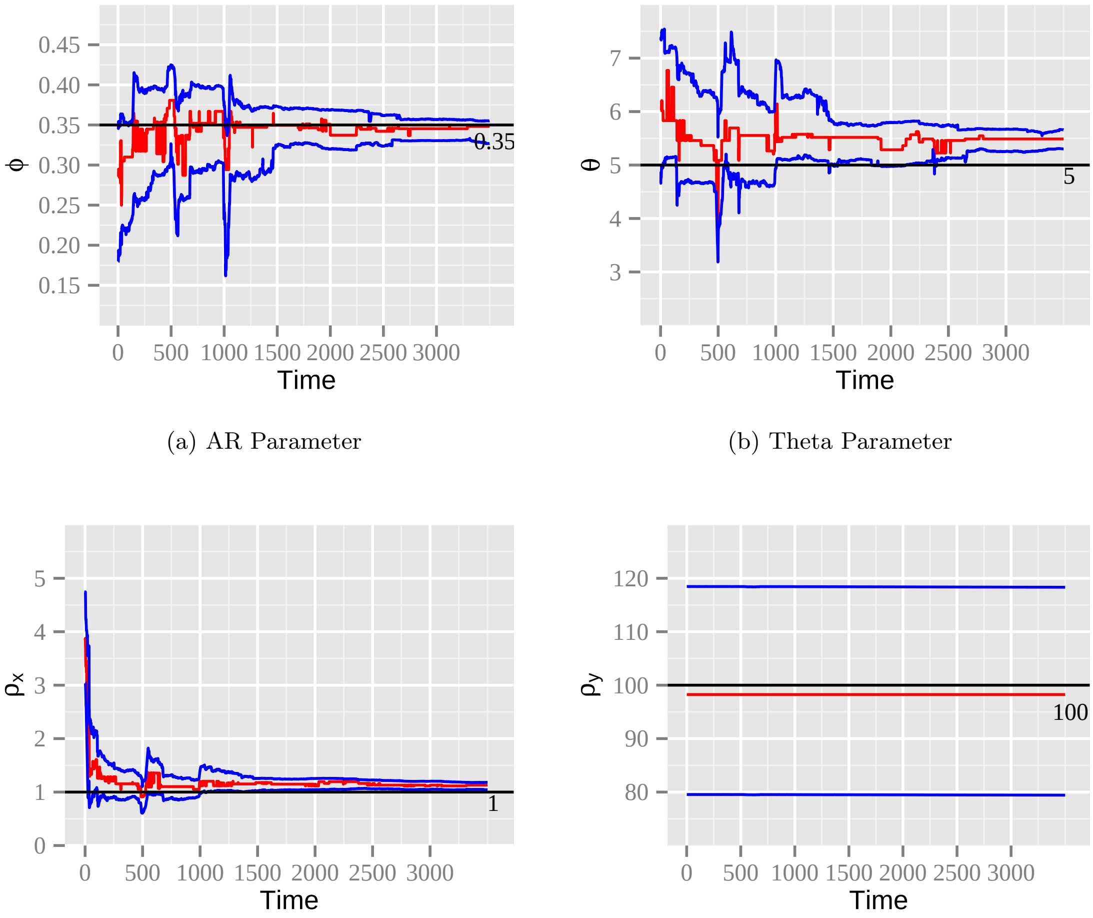
where $v_t \sim \mathcal{N}(0, \sigma_y^2)$, $w_t \sim \mathcal{N}(0, \sigma_x^2)$. ω is assumed to be known.The unknown parameters for this model is given by the vector $\left(\phi, \theta, \sigma_x^2, \sigma_y^2\right)$.

B.1 Example 1

Several data sets are generated from the nonlinear state space model represented by Equations (B.0.1) and (B.0.2), with the parameter values set at $\phi = 0.35$, $\theta = 5$, $\sigma_x^2 = 1$ and $\sigma_y^2 = 0.01$ respectively. The value of ω is fixed at 1.718. The performance of our method is shown in Figure B.1. Except for the θ parameter, the algorithm successfully identified the rest. We should probably have generated more data sets to actually test the method with this model.

B.2 Example 2

Another example has been tried to test our algorithm. Data sets are generated from the nonlinear model with parameter values given by the following, $\phi = -0.7$, $\theta = 1$, $\sigma_x^2 = 0.01$ and $\sigma_y^2 = 0.01$ respectively. Figure B.2 shows the performance of SINLA for this particular model. One can easily see that our method completely fails to identify the true values of the parameters.



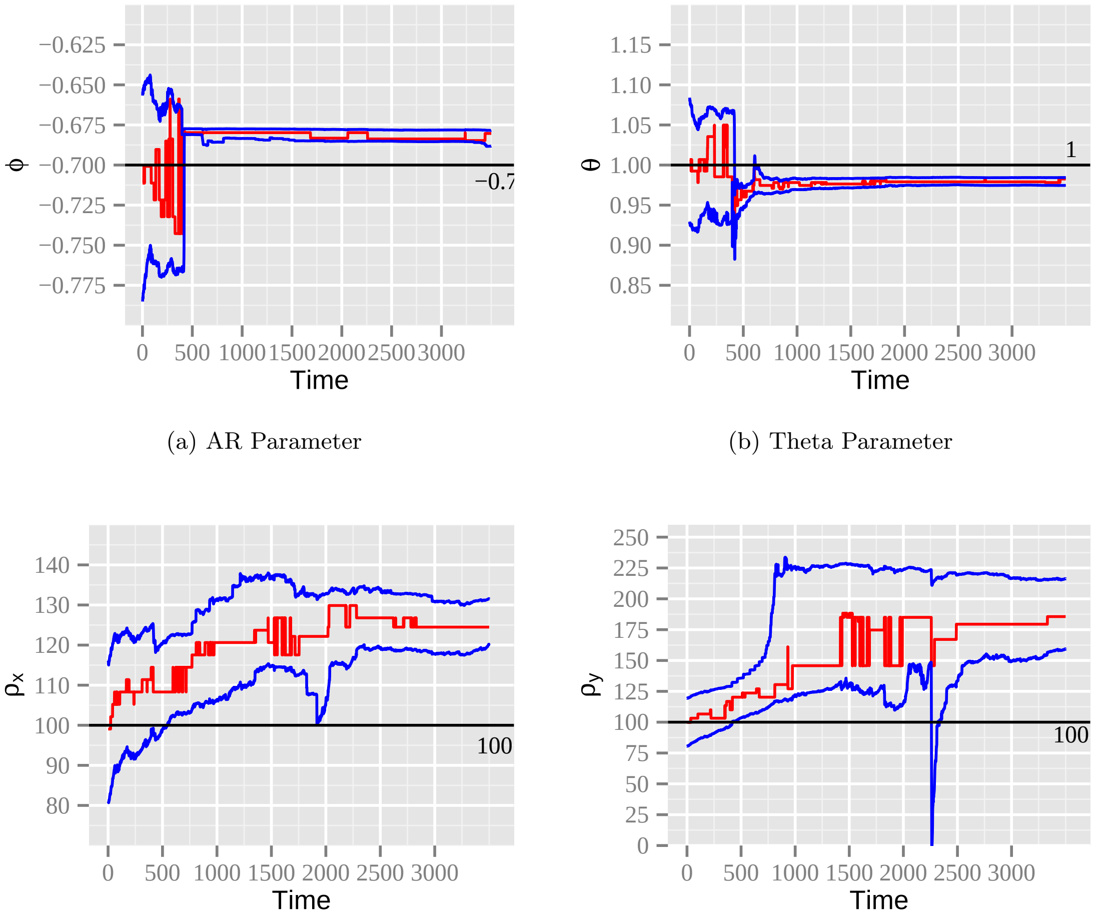
(a) AR Parameter

(b) Theta Parameter

(c) State Precision Parameter

(d) Observation Precision Parameter

Figure B.1: Plot over time, of the mode and 95% probability interval for all four parameters from the nonlinear model. The performance of SINLA has been good overall, with failure to identify the theta parameter completely.



(a) AR Parameter

(b) Theta Parameter

(c) State Precision Parameter

(d) Observation Precision Parameter

Figure B.2: Plot over time, of the mode and 95% probability interval for all four parameters from the nonlinear model. The performance of SINLA has been quite poor, with complete failure to identify most of the parameters.

Appendix C

Non-Gaussian models

Similar to what has been done in A, data sets have been simulated from multiple non-Gaussian models of the form explained in Section 6.6 of Chapter 6. Our approach has been applied on each of these models and consolidated trace plots are presented in view of determining the performance of our algorithm in identifying the true parameter values.

We present the model once again for the sake of clarity:

$$y_t \sim \mathcal{P}(\exp^{(6+x_t)}), \tag{C.0.1}$$

$$x_t = \phi x_{t-1} + w_t, \quad w_t \sim \mathcal{N}(0, \sigma_x^2), \tag{C.0.2}$$

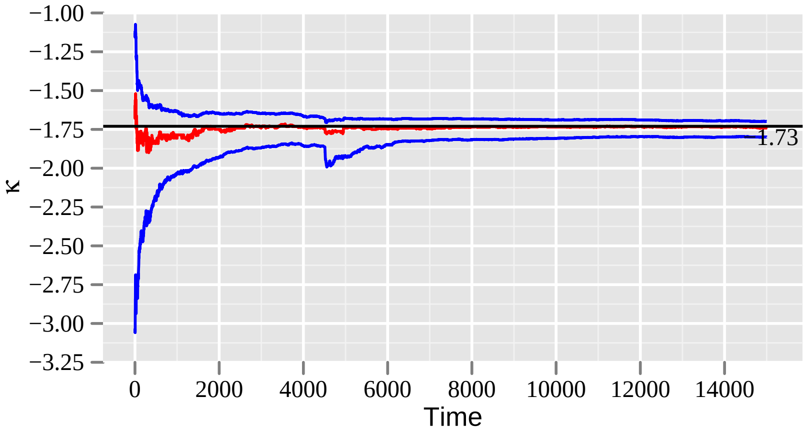
where $\mathcal{P}(\lambda)$ denotes the Poisson distribution with some parameter λ. We had represented the observation model in a state space form given by:

$$y_t = e^{(6+x_t)}\,\eta_t, \quad \eta_t \sim \text{log-}\mathcal{N}(0, \sigma_y^2). \tag{C.0.3}$$

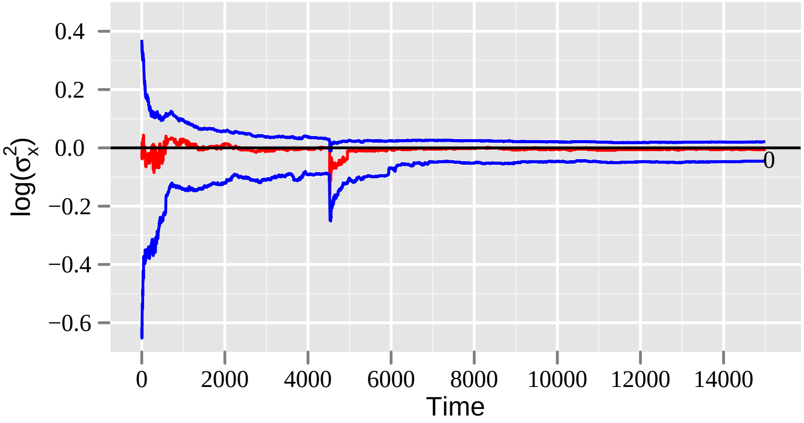
We had mentioned in Chapter 6 that transformed parameter values were used for inference. We have continued to use transformed parameter values given by, κ, log-$\sigma_x^2 = \log(\sigma_x^2)$ and log-$\sigma_y^2 = \log(\sigma_y^2)$, where interest lies only on the former two. Different data sets are generated by varying the values of the unknown parameters ϕ and σ_x^2 given in Equation (C.0.2).

C.1 Example 1

Multiple data sets have been simulated from the model explained above, with the parameter values set at $\phi = -0.7$ and $\sigma_x^2 = 1$. As in all the previous examples, an average of the approximate mode and confidence bounds are computed at each time t, for this particular model and a trace plot of the average is provided in Figure C.1.

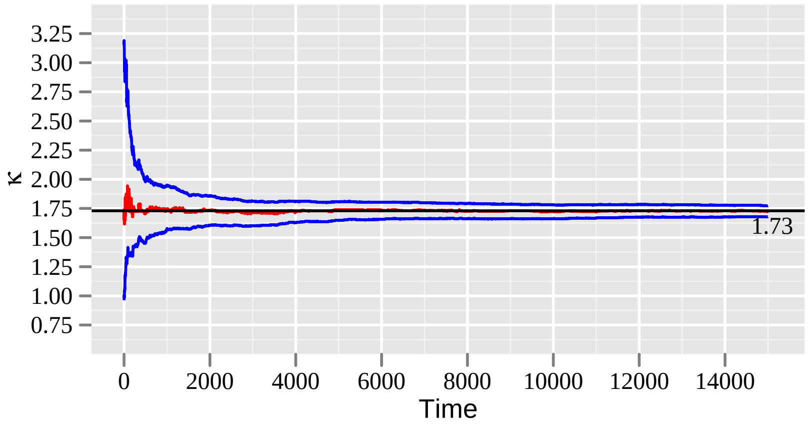


(a) Kappa Parameter

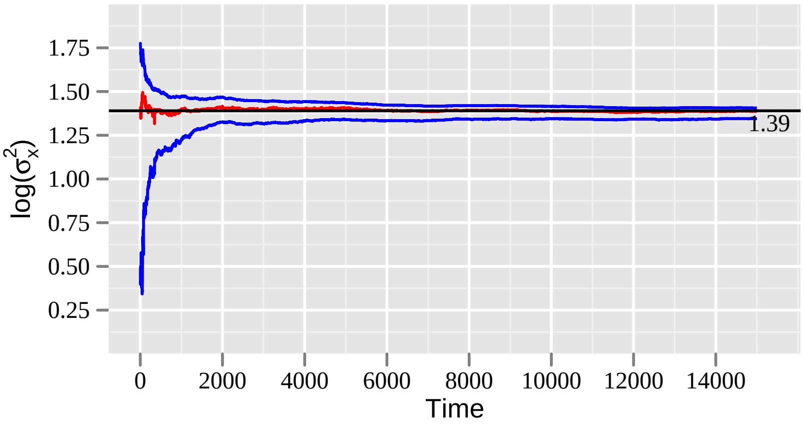


(b) Log State Variance Parameter

Figure C.1: Plot of mode and 95% probability interval for both the transformed parameters of the non-Gaussian model over time for the non-Gaussian model presented in Example Appendix Non-Gaussian models. The performance of our approach is very good, considering the fact that it has successfully identified the true values of the unknown parameters for this model.



(a) Kappa Parameter



(b) Log State Variance Parameter

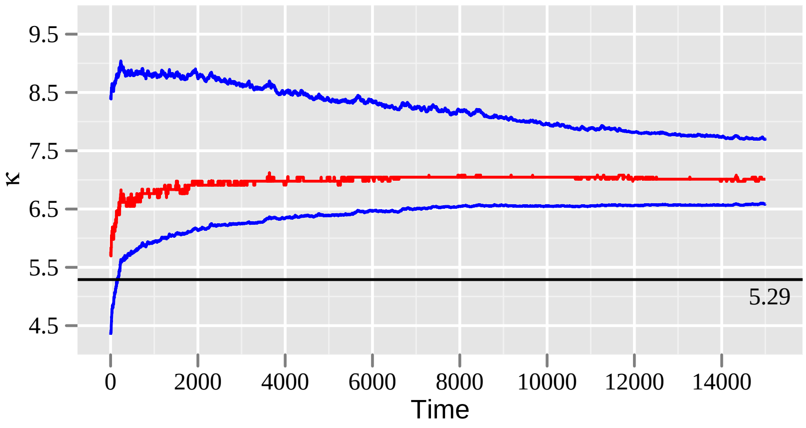
Figure C.2: Plot of mode and 95% probability interval for both the transformed parameters of the non-Gaussian model over time for our non-Gaussian model presented in this section. SINLA has successfully identified the true values of the unknown parameters for this model as is quite evident from the plots.

C.2 Example 2

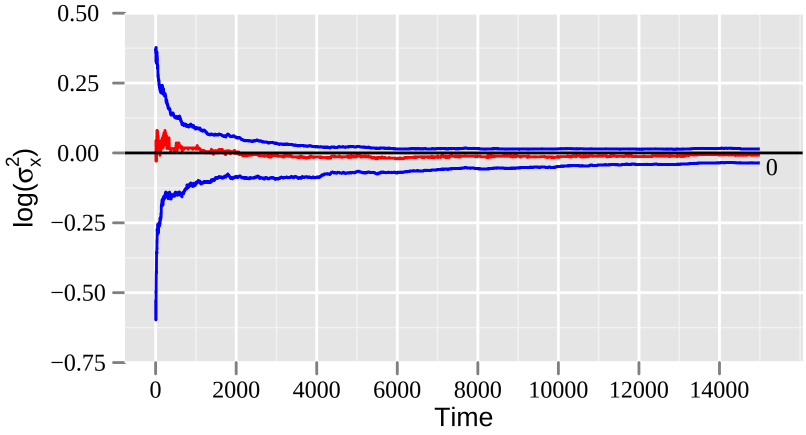
For this example, the parameters of the model have the following values $\phi = 0.7$ and $\sigma_x^2 = 2$. As before data sets are simulated from this particular model and the performance of our algorithm is given by the trace plot in Figure C.2.

C.3 Example 3

We present our final example where the AR parameter lies at the bound and has a value of 0.99. The value of the variance parameter is 1. The results of applying the algorithm is shown through the trace plot given by Figure C.3.



(a) Kappa Parameter



(b) Log State Variance Parameter

Figure C.3: Plot of mode and 95% probability interval for both the transformed parameters of the non-Gaussian model over time for our non-Gaussian model. SINLA has successfully inferred about the true value of the log-variance parameter but failed to identify the κ parameter which is a transformation of the AR parameter.

Bibliography

Alspach, D. & Sorenson, H. W. (1972), 'Nonlinear Bayesian estimation using Gaussian sum approximations', *IEEE Transactions on Automatic Control* **17**(4), 438–448.

Anderson, B. D. O. & Moore, J. B. (1979), *Optimal filtering*, Prentice - Hall.

Anderson, T. W. (2003), *An introduction to multivariate statistical analysis*, Wiley-Interscience.

Andrieu, C., Doucet, A. & Holenstein, R. (2010), 'Particle Markov chain Monte Carlo methods', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **72**(3), 269–342.

Andrieu, C., Doucet, A. & Tadic, V. B. (2005), On-line parameter estimation in general state-space models, *in* IEEE, ed., 'Proceedings of the 44th IEEE Conference on Decision and Control', pp. 332–337.

Andrieu, C., Freitas, N. D. & Doucet, A. (1999), Sequential MCMC for Bayesian model selection, *in* 'Proceedings of the IEEE Signal Processing Workshop on Higher-Order Statistics, 1999.', IEEE, pp. 130–134.

Arasaratnam, I. & Haykin, S. (2009), 'Cubature Kalman filters', *IEEE Transactions on Automatic Control* **54**(6), 1254–1269.

Arasaratnam, I., Haykin, S. & Elliott, R. J. (2007), Discrete-time nonlinear filtering algorithms using Gauss-Hermite quadrature, *in* 'Proceedings of the IEEE', Vol. 95, IEEE, pp. 953–977.

Arulampalam, S., Maskell, S. & Gordon, N. (2002), 'A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking', *IEEE Transactions on Signal Processing* **50**, 174–188.

Barnard, G. A., Jenkins, G. M. & Winsten, C. B. (1962), 'Likelihood inference and time series', *Journal of the Royal Statistical Society. Series A (General)* **125**(3), 321–372.

Bather, J. A. (1965), 'Invariant conditional distributions', *The Annals of Mathematical Statistics* **36**(3), 829–846.

Berger, J. (2006), 'The case for objective Bayesian analysis (with discussion)', *Bayesian Analysis* **1**(3), 385–402.

Bernardo, J. & Smith, A. (1996), *Bayesian theory*, Wiley, New York.

Berzuini, C., Best, N. G., Gilks, W. R. & Larizza, C. (1997), 'Dynamic conditional independence models and Markov chain Monte Carlo methods', *Journal of the American Statistical Association* **92**(440), 1403–1412.

Box, G. E. P., Jenkins, G. M. & Reinsel, G. C. (1970), *Time series analysis: forecasting and control*, John Wiley & Sons.

Box, G. & Tiao, G. (2011), *Bayesian inference in statistical analysis*, Wiley-Interscience.

Brockwell, P. & Davis, R. (2002), *Introduction to time series and forecasting*, Springer.

Bucy, R. S. & Senne, K. D. (1971), 'Digital synthesis of nonlinear filters', *Automatica* **7**, 287–298.

Casella, G. & Berger, R. (2001), *Statistical inference*, Duxbury Press.

Chen, Y. & Singpurwalla, N. D. (1994), 'A non-Gaussian Kalman filter model for tracking software reliability', *Statistica Sinica* **4**, 535–548.

Chib, S. & Greenberg, E. (1994), 'Bayes inference in regression models with arma (p, q) errors', *Journal of Econometrics* **64**(1), 183–206.

Chib, S. & Greenberg, E. (1995), 'Understanding the Metropolis–Hastings algorithms', *The American Statistician* **49**(4), 327–335.

Cowpertwait, P. S. P. & Metcalfe, A. V. (2009), *Introductory Time Series with R*, Springer Publishing Company, Incorporated.

Cseke, B. & Heskes, T. (2010), Improving posterior marginal approximations in latent gaussian models, *in* Y. W. Teh & M. Titterington, eds, 'Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics', Vol. 9, JMLR Workshop and Conference Proceedings, pp. 121 – 128.

de Finetti, B. (1972), *Probability, induction, and statistics*, Wiley, New York.

de Finetti, B. (1974, 1975), *Theory of probability*, Vol. 1, 2, Wiley, New York.

DeGroot, M. H. (2005), *Optimal Statistical Decisions*, Wiley, New York.

Dempster, A. P., Laird, N. M. & Rubin, D. B. (1977), 'Maximum likelihood from incomplete data via the EM algorithm', *Journal of the Royal Statistical Society, Series B (Statistical Methodology)* **39**(1), 1 – 38.

Diggle, P., Moyeed, R. A. & Tawn, J. A. (1998), 'Model-based geostatistics', *Applied Statistics* **47**, 299–350.

Doucet, A., de Freitas, N., Murphy, K. & Russell, S. (2000), Rao-Blackwellised particle filtering for dynamic Bayesian networks, *in* 'Proceedings of the Sixteenth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-00)', Morgan Kaufmann, San Francisco, CA, pp. 176–183.

Doucet, A., Freitas, N. D. & Gordon, N., eds (2001), *Sequential Monte Carlo methods in practice*, Springer.

Doucet, A., Godsill, S. & Andrieu, C. (2000), 'On sequential Monte Carlo sampling methods for Bayesian filtering', *Statistics and Computing* **10**, 197–208.

Doucet, A. & Johansen, A. M. (2009), A tutorial on particle filtering and smoothing: fifteen years later, *in* 'Oxford Handbook of Nonlinear Filtering', Oxford University Press.

Doucet, A. & Tadic, V. B. (2003), 'Parameter estimation in general state-space models using particle methods', *Annals of the Institute of Statistical Mathematics* **55**, 409–422.

Durbin, J. & Koopman, S. J. (2000), 'Time series analysis of non-gaussian observations based on state space models from both classical and bayesian perspectives', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **62**(1), 3–56.

Durbin, J. & Koopman, S. J. (2001), *Time series analysis by state space methods*, Oxford University Press.

Evans, M. & Swartz, T. (1995), 'Methods for approximating integrals in statistics with special emphasis on Bayesian integration problems', *Statistical Science* **10**(3), 254–272.

Evensen, G. (2009), *Data assimilation: the ensemble Kalman filter*, Springer.

Fahrmeir, L. & Künstler, R. (1998), 'Penalized likelihood smoothing in robust state space models', *Metrika* **49**, 172 – 191.

FCC (2003), *Establishment of an interference temperature metric to quantify and manage interference and to expand available unlicensed operation in certain fixed, mobile and satellite frequency bands*, Federal Communications Commission, Tech. Report.

Filzmoser, P. (2004), A multivariate outlier detection method, *in* S. Aivazian, P. Filzmoser & Y. Kharin, eds, 'Proceedings of the Seventh International Conference on Computer Data Analysis and Modeling', Vol. 1, Belarusian State University, Minsk, pp. 18–22.

Fisher, R. (1922), 'On the mathematical foundations of theoritical statistics', *Philosophical Transactions of the Royal Society* **A**(222), 309–368.

Fisher, R. A. (1925), *Statistical methods for research workers*, Oliver and Boyd.

Friedman, J. (1991), 'Multivariate adaptive regression splines.', *Annals of Statistics* **19**, 1 – 67.

Frühwirth-Schnatter, S. (1994), 'Applied state space modelling of non-gaussian time series using integration-based kalman filtering', *Statistics and Computing* **4**, 259–269.

Garthwaite, P. H., Kadane, J. B. & O'Hagan, A. (2005), 'Statistical methods for eliciting probability distributions', *Journal of the American Statistical Association* **100**(470), 680–701.

Gelb, A., Jr., J. F. K., Jr., R. A. N., Price, C. F. & Jr., A. A. S. (1974), *Applied optimal estimation*, MIT Press, Cambridge.

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

Gelman, A., Carlin, J., Stern, H. & Rubin, D. (2004), *Bayesian data analysis*, Chapman & Hall.

Geweke, J. (1989), 'Bayesian inference in econometric models using Monte Carlo integration', *Econometrica* **57**(6), 1317–1339.

Ghosh, J., Delampady, M. & Samanta, T. (2006), *An introduction to Bayesian analysis: theory and methods*, Springer.

Gilks, W. R., Richardson, S. & Spiegelhalter, D. J. (1996*a*), *Markov chain Monte Carlo in practice*, Chapman and Hall.

Gilks, W. R., Richardson, S. & Spiegelhalter, D. J. (1996*b*), *Markov Chain Monte Carlo in practice*, Chapman & Hall, London.

Gordon, N., Salmond, D. J. & Smith, A. F. M. (1993), 'Novel approach to nonlinear/non-Gaussian Bayesian state estimation', *Radar and Signal Processing, IEE Proceedings F* **140**(2).

Grewal, M. S. & Andrews, A. P. (2011), *Kalman Filtering: theory and practice using MATLAB*, Wiley-IEEE Press.

Grimmett, G. R. & Stirzaker, D. R. (2001), *Probability and random processes*, Oxford University Press, USA.

Gupta, N. & Mehra, R. (1974), 'Computational aspects of maximum likelihood estimation and reduction in sensitivity function falculations', *IEEE Transactions on Automatic Control* **19**(6), 774 – 783.

Handschin, J. E. & Mayne, D. Q. (1969), 'Monte carlo techniques to estimate the conditional expectation in multistage nonlinear filtering', *International Journal of Control* **9**(5), 547–559.

Hartigan, J. A. (1969), 'Linear bayesian methods', *Journal of Royal Statistical Society: Series B (Methodological)* **31**(3), 446–454.

Hastie, T. (2011), *gam: Generalized Additive Models*, R package version 1.06.2.

Hastie, T. & Tibshirani, R. (1990), *Generalized additive models*, Chapman & Hall.

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

Haykin, S. (2001), *Kalman filtering and neural networks*, Wiley-Interscience.

Haykin, S. (2005), 'Cognitive radio: Brain-empowered wireless communications', *IEEE Journal on Selected Areas in Communications* **23(2)**, 201–220.

Haykin, S. (2012), *Cognitive Dynamic Systems: Perception-action Cycle, Radar and Radio*, Cambridge University Press.

Ionides, E. L., Bhadra, A., Atchadé, Y. & King, A. (2011), 'Iterated filtering', *Annals of Statistics* **39**(3), 1776–1802.

Ionides, E. L., Bretó, C. & King, A. A. (2006), 'Inference for nonlinear dynamical systems', *Proceedings of the National Academy of Sciences* **103**(49), 18438–18443.

Ito, K. & Xiong, K. (1999), 'Gaussian filters for nonlinear filtering problems', *IEEE Transactions on Automatic Control* **45**, 910–927.

Jaakkola, T. S. & Jordan, M. I. (2000), 'Bayesian parameter estimation via variational methods', *Statistics and Computing* **10**(1), 25–37.

Jeffreys, H. (1946), An invariant form for the prior probability in estimation problems, *in* 'Proceedings of the Royal Society of London', Vol. 186, pp. 453–461.

Jeffreys, H. (1961), *Theory of probability*, Oxford University Press, USA.

Julier, S. J. & Uhlmann, J. (1997), A new extension of the Kalman filter to nonlinear systems, *in* 'Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls, Orlando, Florida', pp. 182–193.

Julier, S. J., Uhlmann, J. K. & Durrant-Whyte, H. F. (1995), A new approach for filtering nonlinear systems, *in* 'Proceedings of the American Control Conference.', Vol. 3, Washington, DC, pp. 1628–1632.

Kadane, J. B. & Wolfson, L. J. (1998), 'Experiences in elicitation', *The Statistician* **47**(1), 3–19.

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

Kalman, R. E. & Bucy, R. S. (1961), 'New results in linear filtering and prediction theory', *Transactions of the ASME. Series D, Journal of Basic Engineering* **83**, 95–107.

Kantas, N., Singh, S. S. & Maciejowski, J. (2009), An overview of sequential Monte Carlo methods for parameter estimation in general state-space models, *in* 'Proceedings IFAC System Identification (SySid) Meeting'.

Kitagawa, G. (1987), 'Non-Gaussian state-space modeling of nonstationary time series', *Journal of the American Statistical Association* **82**(400), 1032–1063.

Kitagawa, G. (1996), 'Monte Carlo filter and smoother for non-Gaussian nonlinear state space models', *Journal of Computational and Graphical Statistics* **5**(1), 1–25.

Kitagawa, G. (1998), 'A self-organizing state-space model', *Journal of the American Statistical Association* **93**(443), 1203–1215.

Koyama, S., Pérez-Bolde, L. C., Shalizi, C. R. & Kass, R. E. (2010), 'Approximate methods for state-space models', *Journal of the American Statistical Association* **105**(489), 170–180.

LaViola, J. J. (2003), A comparison of unscented and extended Kalman filtering for estimating quaternion motion, *in* 'Proceedings of American Control Conference', Vol. 3, IEEE, pp. 2435–2440.

Lefebvre, T., Bruyninckx, H. & Schutter, J. D. (2004), 'Kalman filters for non-linear systems: a comparison of performance', *International Journal of Control* **77**(7), 639–653.

Lindley, D. V. (1980), Approximate bayesian methods, *in* J. M. Bernardo, M. H. Degroot, D. Lindley & A. F. M. Smith, eds, 'Bayesian Statistics', University Press.

Liu, J. S. & Che, R. (1998), 'Sequential Monte Carlo methods for dynamic systems', *Journal of the American Statistical Association* **93**.

Liu, J. & West, M. (2000), Combined parameter and state estimation in simulation-based filtering, *in* D. Freitas & N. J. Gordon, eds, 'Sequential Monte Carlo Methods in Practice. New York', Springer-Verlag, New York.

Lotze, J., Galdo, G. D. & Haardt, M. (2006), Estimation of reflection coefficients for the IlmProp channel modeling environment using path loss models, *in* '51st International Scientific Colloquium (IWK)', Ilmenau, Germany.

Mahalanobis, P. C. (1936), 'On the generalised distance in Statistics', *Proceedings of the National Institute of Sciences of India* **2**(1), 49–55.

Matérn, B. (1986), *Spatial Variation*, Springer-Verlag.

Meinhold, R. J. & Singpurwalla, N. D. (1983), 'Understanding the Kalman filter', *The American Statistician* **37**(2), 123–127.

Merwe, R. V. D. & Wan, E. A. (2001), The square-root unscented kalman filter for state and parameter-estimation, *in* 'in International Conference on Acoustics, Speech, and Signal Processing', pp. 3461–3464.

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

Milborrow, S., derived from mda:mars by Trevor Hastie & Tibshirani., R. (2012), *earth: Multivariate Adaptive Regression Spline Models*, R package version 3.2-3.

Minka, T. P. (2001a), A family of algorithms for approximate Bayesian inference, PhD thesis, Massachusetts Institute of Technology.

Minka, T. P. (2001b), Expectation propagation for approximate bayesian inference, *in* J. S. Breese & D. Koller, eds, 'UAI', Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence,, Morgan Kaufmann, pp. 362–369.

Monahan, J. F. (1983), 'Fully Bayesian analysis of ARMA time series models', *Journal of Econometrics* **21**, 307 – 331.

Monahan, J. F. (2011), *Numerical Methods of Statistics*, Vol. 2, Cambridge University Press.

Nadaraya, E. A. (1989), *Nonparametric estimation of probability densities and regression curves*, Mathematics and its applications (D. Reidel Publishing Company): Soviet series, Kluwer Academic Publishers.

Nieto, F. H. & Guerrero, V. M. (1995), 'Kalman filter for singular and conditional state-space models when the system state and the observational error are correlated', *Statistics & Probability Letters* **22**(4), 303–310.

Nørgaard, M., Poulsen, N. K. & Ravn, O. (2000*a*), Advances in derivative-free state estimation for nonlinear systems, IMM-REP-1998-15, Technical University of Denmark, Denmark.

Nørgaard, M., Poulsen, N. K. & Ravn, O. (2000*b*), 'New developments in state estimation for nonlinear systems', *Automatica* **36**(11), 1627–1638.

Perea, L., How, J., Breger, L. & Elosegui, P. (2007), Nonlinearity in sensor fusion. divergence issues in EKF, modified truncated SOF, and UKF, *in* 'AIAA Guidance Navigation and Control Conference and Exhibit', Hilton Head, South Carolina.

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

Pole, A. & West, M. (1990), ‘Efficient Bayesian learning in nonlinear dynamic models’, *Journal of Forecasting* **9**(2), 119–136.

Poyiadjis, G., Doucet, A. & Singh, S. S. (2005), Maximum likelihood parameter estimation in general state-space models using particle methods, *in* ‘Procedures of the American Statistical Association’.

Poyiadjis, G., Doucet, A. & Singh, S. S. (2011), ‘Particle approximations of the score and observed information matrix in state space models with application to parameter estimation’, *Biometrika* **98**(1), 65–80.

Prado, R. & West, M. (2010), *Time Series: Modeling, Computation, and Inference*, Chapman and Hall/CRC.

Rabiner, L. R. (1989), A tutorial on hidden Markov models and selected applications in speech recognition, *in* ‘Proceedings of the IEEE’, Vol. 77, pp. 257–286.

Robert, C. P. (2001), *The Bayesian Choice: from Decision-Theoretic Motivations to Computational Implementation*, Springer-Verlag, New York.

Roberts, G. O. & Rosenthal, J. S. (2004), ‘General state space Markov chains and MCMC algorithms’, *Probability Surveys* **1**, 20–71.

Rue, H. & Held, L. (2005), *Gaussian Markov random fields: theory and application*, Chapman & Hall.

Rue, H., Martino, S. & Chopin, N. (2009), ‘Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations’, *Journal of the Royal Statistical Society, Series B (Statistical Methodology)* **71**, 319–392.

Scott, D. W. (1992), *Multivariate Density Estimation: Theory, Practice, and Visualization*, John Wiley & Sons.

Shumway, R. H. & Stoffer, D. S. (1982), ‘An approach to time series smoothing and forecasting using the EM algorithm’, *Journal of Time Series Analysis* **3**, 253 – 264.

Silverman, B. W. (1986), *Density Estimation for Statistics and Data Analysis*, Crc Press.

Smídl, V. & Quinn, A. (2008), ‘Variational bayesian filtering’, *IEEE Transactions on Signal Processing* **56**(10-2), 5020–5030.

Sorenson, H. & Stubberud, A. (1968), ‘Nonlinear filtering by approximation of the a posteriori density’, *International Journal of Control* **8**(1), 33–51.

Srinivasan, K. (1970), 'State estimation by orthogonal expansion of probability distributions', *IEEE Transactions on Automatic Control* **15**(1), 3–10.

Steel, M. (2008), Bayesian time series analysis, *in* 'The new Palgrave dictionary of economics', Palgrave MacMillan.

Storvik, G. (2002), 'Particle filters for state-space models with the presence of unknown static parameters', *IEEE Transactions on Signal Processing* **50**(2), 281–289.

Terejanu, G., Singla, P., Singh, T. & Scott, P. D. (2011), 'Adaptive gaussian sum filter for nonlinear bayesian estimation', *IEEE Trans. Automat. Contr.* **56**(9), 2151–2156.

Thrun, S. (2002), Particle filters in robotics, *in* 'Proceedings of the 17th Annual Conference on Uncertainty in AI (UAI)'.

Tierney, L. & Kadane, J. B. (1986), 'Accurate approximations for posterior moments and marginal densities', *Journal of the American Statistical Association* **81**(393), 82–86.

Tierney, L., Kass, R. E. & Kadane, J. B. (1989), 'Fully exponential Laplace approximations to expectations and variances of nonpositive functions', *Journal of the American Statistical Association* **84**(407), 710–716.

Titterington, D. M. (2011), The em algorithm, variational approximations and expectation propagation for mixtures, *in* K. Mengersen, C. Robert & M. Titterington, eds, 'Mixtures: estimation and applications', John Wiley & Sons, pp. 1–29.

Turlach, B. A. (1993), Bandwidth selection in Kernel density estimation: A review, *in* 'CORE and Institut de Statistique'.

Vermaak, J., Lawrence, N. & Perez, P. (2003), Variational inference for visual tracking, *in* 'Proceedings of IEEE conference on computer vision and pattern recognition', Vol. 1, pp. 773–780.

Walrand, J. (2005), 'EE226a - Summary of Lecture 13 and 14 Kalman Filter: Convergence'.
URL: *http://robotics.eecs.berkeley.edu/ wlr/226aF05/L13.pdf*

Wan, E. A. & Merwe, R. V. D. (2000), The unscented Kalman filter for nonlinear estimation, *in* 'Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000', pp. 153–158.

Wan, E. A., van der Merwe, R. & Nelson, A. (2000), Dual estimation and the Unscented transformation, *in* S. A. Solla, T. K. Leen & K. R. Muller, eds, 'Advances in Neural Information Processing Systems (NIPS12)', MIT Press, pp. 666–672.

West, M. & Harrison, J. (1997), *Bayesian forecasting and dynamic models*, Springer series in Statistics, second edn, Springer.

Wood, S. (2006), *Generalized Additive Models: An Introduction with R*, Chapman & Hall/CRC.

Wood, S. N. (2003), 'Thin-plate regression splines', *Journal of the Royal Statistical Society, Series B (Statistical Methodology)* **65**(1), 95–114.

Wood, S. N. (2004), 'Stable and efficient multiple smoothing parameter estimation for generalized additive models', *Journal of the American Statistical Association* **99**(467), 673–686.

Xiong, K., Zhang, H. Y. & Chan, C. W. (2006), 'Performance evaluation of UKF-based nonlinear filtering', *Automatica* **42**(2), 261–270.

Ypma, A. & Heskes, T. (2005), 'Novel approximations for inference in nonlinear dynamical systems using expectation propagation', *Neurocomputing* **69**, 85–99.