The Pennsylvania State University
The Graduate School

DUAL ESTIMATION IN STATE SPACE MODELS WITH VIOLATION TO NORMALITY: A COMPARISON BETWEEN THE EXTENDED KALMAN FILTER AND THE PARTICLE FILTER

A Thesis in
Statistics
by
Meng Chen

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

December 2019

The thesis of Meng Chen was reviewed and approved* by the following:

Murali Haran
Professor of Statistics
Thesis Advisor, Chair of Committee

Le Bao
Associate Professor of Statistics

Ephraim Mont Hanks
Associate Professor of Statistics
Chair of Graduate Studies

*Signatures are on file in the Graduate School.

Abstract

In the field of psychology and developmental science, researchers often study the change of some underlying latent construct over time. It is of interest both to estimate the latent states that an individual is in and to extract patterns that would characterize the change process. Translated into dynamic modeling language, researchers are interested in the dual estimation of states and model parameters. Filtering methods, such as the commonly adopted Kalman filter, can aid in this process. However, when the linear and normality assumptions of the Kalman filter is challenged, the estimates may no longer be reliable. This thesis set out to investigate how one algorithm from the Kalman filter family, the extended Kalman filter (EKF), and an alternative, simulation-based approach of particle filter, behave under the ideal condition of normality and when the normality assumption is violated, through a set of simulations. Results from simulations show, for both algorithms, overall satisfactory performance under the ideal normal condition, and frequently biased parameter estimates when the distribution of process noises was skewed. The particle-filter-associated approach slightly outperforms the EKF-associated approach when the optimization problem becomes harder. Caveats regarding the interpretation of results are discussed along with potential future research directions.

Table of Contents

List of Figures **vi**

List of Tables **viii**

Chapter 1
Introduction **1**
1.1 Model and Notation 2
1.2 Filtering Algorithms 3
1.3 Proposed Work 4

Chapter 2
The Kalman Filter Family Approach **5**
2.1 The Kalman Filter 5
2.2 The Extended Kalman Filter (EKF) 7
2.3 Estimation of Model Parameters 8

Chapter 3
The Sequential Monte Carlo Approach **10**
3.1 The Particle Filter 10
3.2 Iterated filtering (IF) for Dual Estimation 12

Chapter 4
Comparing Algorithms **15**
4.1 Literature Review 15
4.2 A Simulation-based Comparison 16

Chapter 5
Discussion **26**
5.1 Summary of Conclusions 26
5.2 Caveats and Future Work 27

Bibliography **32**

List of Figures

3.1 Flow Chart of Iterated Filtering at Each Iteration m 13

4.1 An example of generated skewed process noises compared with normal process noises in trajectories of latent states (left) and manifest observations (right): a collection of positively skewed process noises resulted in both the latent states and observations to exhibit somewhat upwards trajectories. 18

4.2 Results comparison (one-dimensional optimization) under the first simulation condition: the two algorithms behave similarly in terms of the distributions of estimated parameters, and the means of the estimates are close to the true values marked by the horizontal lines. 19

4.3 Results comparison (two-dimensional optimization) under the first simulation condition: the two algorithms behave similarly in the resulted distributions of estimated parameters, and the means of the estimates are close to the true values marked by the horizontal lines. 20

4.4 Results comparison (three-dimensional optimization) under the first simulation condition: the two algorithms behave similarly for the two variance parameters σ_ζ^2 and σ_ϵ^2, but for β, IF resulted in a more condensed collection of parameter estimates, and the mean was slightly closer to the true value (horizontal line). 21

4.5 Results comparison (one-dimensional optimization) under the second simulation condition: results from the two algorithms are similarly off from the true parameter values (horizontal lines), with IF on average having estimates slightly closer to true values. 22

4.6 Results comparison (two-dimensional optimization) under the second simulation condition: results from the two algorithms are similarly off from the true parameter values (horizontal lines), with IF on average having estimates slightly closer to true values. 24

4.7 Results comparison (three-dimensional optimization) under the second simulation condition: results from the two algorithms are similarly off from the true parameter values. The EKF may exhibit a wider spread because the results were based on the 37 successful MC trials. 25

5.1 Trajectories of β Estimate with the Same Dataset and Different Replications 29
5.2 Trajectories of β Estimate under Different Combinations of Random Walk Parameters for One Replication 30

List of Tables

4.1 Performance comparison in condition 1 with 1-D optimization. MCSD: Monte Carlo standard deviation. Biases and RMSEs from the two approaches are comparable in value. SEs from both approaches are underestimated. 19

4.2 Performance comparison in condition 1 with 2-D optimization. Biases from the two approaches are comparable in value. IF results in smaller RMSEs in two out of the three parameter combinations. Similar to the results of 1-D optimization, SEs are underestimated. 21

4.3 Performance comparison in condition 1 with 3-D optimization. Biases from the two approaches are comparable in value. IF outperforms the EKF in terms of biases and RMSEs, but underestimates the SE more than the EKF. 22

4.4 Comparison of mean-square-error (MSE) for latent states estimation in condition 1. IF has smaller MSE for 3-D optimization, and is comparable to the EKF in 1-D and 2-D. 22

4.5 Performance comparison in condition 2 with 1-D optimization. IF yields consistently smaller relative bias and RMSE, and underestimates the SE less. 23

4.6 Performance comparison in condition 2 with 2-D optimization. IF yields consistently smaller relative bias and RMSE. Both algorithms underestimate SEs. 23

4.7 Performance comparison in condition 2 with 3-D optimization. The two algorithms are comparable in their relative bias and RMSE. Both algorithms underestimate SEs. 23

4.8 Comparison of mean-square-error (MSE) for latent states estimation in condition 2. The two algorithms are comparable. 25

5.1 Average Computation Time (in seconds) 28

Chapter 1

Introduction

In the study of human development, where researchers are interested in how individuals develop (most often psychologically) through time, the term "dynamic system" is not a foreign concept. For example, to Thelen and Smith (2007), dynamic system theory links the micro-level real-time behaviors to more macro-level life-time growth (or dynamic patterns), and to van Geert (2018), it allows researchers to utilize a variety of dynamic modeling tools as they capture changes and evolution through their sets of rules, which in term helps researchers understand the universal patterns or trends exhibits in free-willed individuals. Some examples of phenomena modeled by dynamic systems include regulatory process (with system unit being one individual) (e.g. Oravecz et al., 2016; Bringmann et al., 2018; Kuppens et al., 2010), interplay within networks of psychopathological symptoms (Borsboom and Cramer, 2013; Bringmann et al., 2013; Fried et al., 2017) and coordination amongst family members (with system unit being one couple dyad, one parent-child dyad or one family)(e.g. Schermerhorn et al., 2007; Chow et al., 2017). Sometimes, the foci of investigation are observed behaviors (as in the case of psychopathological symptoms), but often, research questions involve changes in underlying psychological process or more abstract concepts that are not directly observable and must be inferred from the behavioral they manifest. When both the unobserved process and evolution patterns (and potentially, predictions based on unobserved evolution trajectories) are of interest, in the state

space model formulation of the problem, this requires estimation of both model parameters governing the system and individual state where the system is at given a particular time. Such estimation is often called "dual-estimation"(Wan et al., 2000). It usually includes a filtering process that iteratively estimates the latent states coupled with an optimization process that finds the most plausible set of parameters in the model formulation, and is often difficult when the system does not evolve in a linear fashion or when the states exhibit high non-normality, both making the likelihood hard to evaluate.

1.1 Model and Notation

The dynamic model of focus in this thesis is of the general form of

$$\boldsymbol{\eta}_t = f(\boldsymbol{\eta}_{t-1}, \boldsymbol{\beta}) + \boldsymbol{\zeta}_t, \tag{1.1}$$

where $\boldsymbol{\eta}_t$ is a $p-$dimensional vector representing latent states for the dynamic system at time t, $f : \mathbb{R}^p \to \mathbb{R}^p$ is the state transition function from $t-1$ to t, and $\boldsymbol{\beta}$ is a $k-$dimensional vector of parameters in f. Since the dynamic in Equation 1.1 is represented by a first-order difference equation, a model of this particular formulation is also commonly referred to as a "state space model". The process noise in the system at time t is represented by $\boldsymbol{\zeta}_t$, which is a $p-$dimensional vector representing process noise. It usually is used to account for random disturbances to the system states or environmental influences that are not accounted for in f. To link the latent system states to the observed manifestations, a measurement model is also included:

$$\boldsymbol{y}_t = \boldsymbol{\Lambda}\boldsymbol{\eta}_t + \boldsymbol{\epsilon}_t \tag{1.2}$$

$$\boldsymbol{\epsilon}_t \sim \boldsymbol{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}). \tag{1.3}$$

Here $\boldsymbol{y}_t$ is a $q-$dimensional vector of observed measurements originating from the latent states $\boldsymbol{\eta}_t$, $\boldsymbol{\Lambda}$ is a $q \times p$ matrix of the measurement loading that links the latent states $\boldsymbol{\eta}_t$ to the observed $\boldsymbol{y}_t$. For this thesis, I assumed linear measurement functions, thus represented by matrix multiplications.$\boldsymbol{\epsilon}_t$ is a $q-$dimensional vector

that represents measurement error, assuming measurements are sometimes not exactly accurate. From Equation 1.1 to 1.3, the entire model involves a parameter vector

$$\boldsymbol{\theta} = \{\boldsymbol{\beta}, \boldsymbol{\Sigma}_{\boldsymbol{\zeta}}, \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}, (\boldsymbol{\Lambda})\},$$

and $\boldsymbol{\Lambda}$ may or may not be included in the parameter vector based on assumed accuracy of measurements.

1.2 Filtering Algorithms

The earliest filtering method and thus the most adopted in Psychology is the Kalman filter (Kalman, 1960). The Kalman filter originated from the field of signal processing. It estimates latent states of a linear discrete-time dynamic model with process noise, and has gained traction in the recent decades in other scientific fields such as economics, physiology and neuroscience (e.g. Harvey, 1990; Haykin, 2004; Tarvainen et al., 2009). The initial idea behind this algorithm was to update latent state estimates in real time as data are being collected, but its dependency on normality assumptions enables the derivation of likelihood function via prediction errors, thus generating the object for optimization of model parameters along the filtering process. However, when the system is nonlinear or in general when normality is violated, mean-covariance-structure that the Kalman filter depends on can no longer provide accurate information of the state distribution or the likelihood function. Researchers have since worked out a few extensions that opens up the possibility to apply Kalman filter on nonlinear systems, the most common one being the extended Kalman filter (EKF, in e.g. Daum, 2005; Montella, 2014). In theory, the EKF accommodates the nonnormality in state distribution that has its source in transitional nonlinearity. For estimating model parameters, the Kalman filter can be used to calculate the likelihood associated with a particular parameter set and thus researchers can choose to use any optimization method that they see fit.

Simulation-based methods came to the fore with technological development that resulted in increased computing power. A particularly useful simulation-based

method in this problem is the particle filter (Ristic et al., 2004). This is a flexible method that eliminated the reliance on the normal distribution, as the filtering is not feature-dependent, meaning that it carries not the mean-covariance-structure but a collection of simulated states based on the distributional assumption. Optimization of model parameters following the particle filter can be done in a nested iterative simulation fashion ("iterated filtering", in Ionides et al., 2015), or the particle filter can be used as solely a way of likelihood computation for parameter optimization (see Kantas et al., 2015, for some examples).

1.3 Proposed Work

For this thesis, the goal is to compare the dual-estimation methods from two families through simulated examples for a evaluation of their performance, to determine the appropriate algorithm to use under studied circumstances that balances the computational time and robustness to normality violations. I also include a linear system with normal state distributions for a baseline comparison. The goals of the dual-estimation are to estimate parameters $\boldsymbol{\theta}$ through maximizing the likelihood and to estimate the time string of latent states $\boldsymbol{\eta_{1..T}}$ through filtering given the parameter estimates. The methods included in the investigation are: EKF coupled with a quasi-newton method for parameter optimization, and iterative filtering (IF). The remaining chapters of this thesis are organized as follows: Chapter 2 introduces the details of how the standard Kalman filter is utilized for the estimation of latent states $\boldsymbol{\eta}$, as well as how the EKF extends from the standard Kalman filter, and a rough layout of the optimization algorithm that is used to accompany EKF in achieving parameter estimation; Chapter 3 introduces the idea of sequential Monte Carlo approach via the particle filter, and iterated filtering (IF), which builds on the particle filter for parameter estimation; Chapter 4 includes a brief literature review of previous effort of comparing these two kinds of algorithms, and new simulation studies conducted under the baseline condition and the non-normal distribution condition; Chapter 5 is the discussion section that overviews the conclusions of this project, examines its limitations and proposes some future directions.

Chapter 2

The Kalman Filter Family Approach

This chapter focuses on one of the two algorithms this project aims to investigate: the extended Kalman filter (EKF). In order to describe the EKF, it is first necessary to outline the standard Kalman filter, which, since its introduction in 1960, has slowly gained traction in the modeling of human dynamics, such as behaviors, core affect, and brain connectivity (Chow et al., 2011; Molenaar et al., 2016, 1992).

2.1 The Kalman Filter

Consider a version of dynamic model (Equation 1.1) where the state transition function f is linear,

$$\begin{aligned} \boldsymbol{\eta}_t &= f(\boldsymbol{\eta}_{t-1}, \boldsymbol{\beta}) + \boldsymbol{\zeta}_t \\ &= \boldsymbol{B}\boldsymbol{\eta}_{t-1} + \boldsymbol{\zeta}_t, \end{aligned}$$

and thus can be represented by matrix multiplication where $\boldsymbol{B}$ is a matrix containing the model parameters in the vector $\boldsymbol{\beta}$. $\boldsymbol{\zeta}_t$ is assumed to follow a multivariate normal distribution, $\boldsymbol{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{\zeta}})$.

The Kalman filter is a recursive procedure for computing the *optimal* estimator (in that it minimizes the mean square errors and achieves the Cramér-Rao bound) of latent states $\boldsymbol{\eta}$ at time t given the information available at time t (Kalman,

1960; Harvey, 2001). It relies on properties of multivariate normal distributions to specify the conditional distributions of states at time t, such that these conditional distributions can be adequately specified by their means and covariance matrices. Under the normality assumptions, the final state estimate $\hat{\boldsymbol{\eta}}_{t|t}$, which is the conditional mean of a latent state $\boldsymbol{\eta}_t$, is the minimum mean square estimator (MMSE) of $\boldsymbol{\eta}_t$ (Harvey, 1990). At each iteration, to estimate $\boldsymbol{\eta}_t$, the Kalman filter consists of two steps:

Step 1: Prediction

When we have data up to time incident $t-1$, we can predict the next time t using results from the previous iteration of the Kalman filter, which are derived from observations $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{t-1}$:

$$\hat{\boldsymbol{\eta}}_{t|t-1} \triangleq E(\boldsymbol{\eta}_t|\boldsymbol{y}_0, ..., \boldsymbol{y}_{t-1}) = \boldsymbol{B}\hat{\boldsymbol{\eta}}_{t-1|t-1} \tag{2.1}$$

$$\boldsymbol{P}_{t|t-1} \triangleq cov(\boldsymbol{\eta}_t|\boldsymbol{y}_0, ..., \boldsymbol{y}_{t-1}) = \boldsymbol{B}\boldsymbol{P}_{t-1|t-1}\boldsymbol{B}^T + \boldsymbol{\Sigma}_{\boldsymbol{\zeta}} \tag{2.2}$$

Step 2: Update

Once data of the next time incident become available, we can update the prediction of $\boldsymbol{\eta}_t$ from the prediction step:

$$\boldsymbol{v}_t \triangleq \boldsymbol{y}_t - E(\boldsymbol{y}_t|\hat{\boldsymbol{\eta}}_{t|t-1}) = \boldsymbol{y}_t - \boldsymbol{\Lambda}\hat{\boldsymbol{\eta}}_{t|t-1} \tag{2.3}$$

$$\boldsymbol{V}_t \triangleq cov(\boldsymbol{v}_t) = \boldsymbol{\Lambda}\boldsymbol{P}_{t|t-1}\boldsymbol{\Lambda}^T + \boldsymbol{\Sigma}_\epsilon \tag{2.4}$$

$$\boldsymbol{K}_t = \boldsymbol{P}_{t|t-1}\boldsymbol{\Lambda}^T\boldsymbol{V}_t^{-1}.$$

In the above equations, $\boldsymbol{v}_t$ is referred to as prediction error at time t, and $\boldsymbol{V}_t$ is the corresponding variance. $\boldsymbol{K}_t$, called the Kalman gain, can be seen as a relative weight between the new observation, which is affected by the measurement error covariance $\boldsymbol{\Sigma}_\epsilon$, and the current state estimate, whose accuracy is captured by $\boldsymbol{P}_{t|t-1}$). The more noisy observations are, the lower the Kalman gain value is. Thus less weight is given to the new observation when updating the state prediction)

$$\hat{\boldsymbol{\eta}}_{t|t} = \hat{\boldsymbol{\eta}}_{t|t-1} + \boldsymbol{K}_t\boldsymbol{v}_t$$

$$\boldsymbol{P}_{t|t} = \boldsymbol{P}_{t|t-1} - \boldsymbol{K}_t\boldsymbol{\Lambda}\boldsymbol{P}_{t|t-1}$$

To summarize, the Kalman filter takes an observed time series $\boldsymbol{y}_{1,\ldots,T}$, and produces the estimated underlying latent states $\boldsymbol{\eta}_{1,\ldots,T}$. The normal distribution assumption is key for both steps of the filtering process, and the prediction step also requires that the transition function f is linear. When f is nonlinear (i.e. can no longer be represented by matrix multiplication), the EKF can be a helpful and relatively easy means of solving the state estimation problem.

2.2 The Extended Kalman Filter (EKF)

The challenge a nonlinear transition function poses to the Kalman filter lies in the fact that one can no longer propagate the mean and covariance structure forward as in Equations 2.1 - 2.2. The EKF circumvents this problem by linearizing the nonlinear function locally using Taylor series expansion. In this thesis, I focus on the first-order EKF, for which:

$$f(\boldsymbol{\eta}_{t-1}) \approx f(\hat{\boldsymbol{\eta}}_{t-1|t-1}) + f^{'}(\hat{\boldsymbol{\eta}}_{t-1|t-1})(\boldsymbol{\eta}_{t-1} - \hat{\boldsymbol{\eta}}_{t-1|t-1})$$

Let $\boldsymbol{J_f}(\hat{\boldsymbol{\eta}}_{t-1|t-1})$ represent the Jacobian matrix of the transition function f evaluated at $\hat{\boldsymbol{\eta}}_{t-1|t-1}$, the updated estimate of $\boldsymbol{\eta}_{t-1}$, and $\boldsymbol{e}_{t-1} \equiv \boldsymbol{\eta}_{t-1} - \hat{\boldsymbol{\eta}}_{t-1|t-1}$, then:

$$\begin{aligned}
E(\boldsymbol{\eta}_t|\boldsymbol{y}_0,\ldots,\boldsymbol{y}_{t-1}) &\approx f(\hat{\boldsymbol{\eta}}_{t-1|t-1}) + \boldsymbol{J_f}(\hat{\boldsymbol{\eta}}_{t|t-1})E(\boldsymbol{e}_{t-1}|\boldsymbol{y}_0,\ldots,\boldsymbol{y}_{t-1}) = f(\hat{\boldsymbol{\eta}}_{t-1|t-1}).\\
\text{Let } \tilde{\boldsymbol{e}}_t &= \boldsymbol{\eta}_t - f(\hat{\boldsymbol{\eta}}_{t-1|t-1})\\
&= f(\boldsymbol{\eta}_{t-1}) + \boldsymbol{\zeta}_t - f(\hat{\boldsymbol{\eta}}_{t-1|t-1}),\\
&\approx \boldsymbol{J_f}(\hat{\boldsymbol{\eta}}_{t-1|t-1})\boldsymbol{e}_{t-1} + \boldsymbol{\zeta}_t\\
cov(\boldsymbol{\eta}_t|\boldsymbol{y}_0,\ldots,\boldsymbol{y}_{t-1}) &= E(\tilde{\boldsymbol{e}}_t\tilde{\boldsymbol{e}}_t^T)\\
&= \boldsymbol{J_f}(\hat{\boldsymbol{\eta}}_{t-1|t-1})E(\boldsymbol{e}_{t-1}\boldsymbol{e}_{t-1}^T)\boldsymbol{J_f}^T(\hat{\boldsymbol{\eta}}_{t-1|t-1}) + E(\boldsymbol{\zeta}_t\boldsymbol{\zeta}_t^T)\\
&= \boldsymbol{J_f}(\hat{\boldsymbol{\eta}}_{t-1|t-1})\boldsymbol{P}_{t-1|t-1}\boldsymbol{J_f}^T(\hat{\boldsymbol{\eta}}_{t-1|t-1}) + \boldsymbol{\Sigma}_\zeta.
\end{aligned}$$

The procedure of EKF is mostly identical to the standard Kalman filter, except that Step 1 (Equations 2.1 - 2.2) is replaced by:

$$\begin{aligned}
\hat{\boldsymbol{\eta}}_{t|t-1} &= f(\hat{\boldsymbol{\eta}}_{t-1|t-1})\\
\boldsymbol{P}_{t|t-1} &= \boldsymbol{J_f}(\hat{\boldsymbol{\eta}}_{t-1|t-1})\boldsymbol{P}_{t-1|t-1}\boldsymbol{J_f}^T(\hat{\boldsymbol{\eta}}_{t-1|t-1}) + \boldsymbol{\Sigma}_\zeta,
\end{aligned}$$

and Step 2 in the standard Kalman filter still applies since I am assuming a linear measurement model (the same deduction works for a nonlinear measurement function). However, more investigation on this algorithm later suggests that it is only reliable on transition functions that can be well approximated on the time scale of observation intervals (Julier and Uhlmann, 1997; Schiff, 2012). Additionally, it also requires calculations of Jacobians at each iteration, which may become computationally burdensome with higher state dimensions and more complex functions.

2.3 Estimation of Model Parameters

The Kalman filter and the EKF can be readily embedded into any optimization method. With a given set of parameter value $\boldsymbol{\theta}$, the likelihood of $\boldsymbol{\theta}$ given observed data can be calculated from by-products of the Kalman filter (or the EKF). Both $\boldsymbol{v}_t$ (Equation 2.3) and $\boldsymbol{V}_t$ (Equation 2.4) are used in the calculation of likelihood through what is referred to as "prediction error decomposition form" of the log-likelihood:

$$\log\ell(\boldsymbol{\theta}) = -\frac{1}{2}\sum_{j=1}^{\boldsymbol{T}}\Big[\log(2\pi) + \log|\boldsymbol{V}_{t_j}| + \boldsymbol{v}_{t_j}^T\boldsymbol{V}_{t_j}^{-1}\boldsymbol{v}_{t_j}\Big], \tag{2.5}$$

Parameter estimation can thus be done using any preferred optimization method to search of the set of parameters maximizing Equation 2.5. For this project, I adopted a quasi-newton scheme method, the Broyden-Fletcher-Goldfard-Shanno (BFGS) method, which searches for the minimum of the negative log-likelihood through backtracking line search utilizing the Jacobian and approximated Hessian. The pseudocode for the BFGS method can be written as follows:

1. Initialize: Select reasonable $\boldsymbol{\theta}_0$ and approximate the corresponding Hessian H_0, or set $H_0 = I$.

2. Repeat the following steps until convergence criteria is met (need to select convergence criteria):

   (a) Calculate $\boldsymbol{\theta}_{direction} = -H_{\boldsymbol{\theta}_n}^{-1}G_{\boldsymbol{\theta}_n}$.
   $G_{\boldsymbol{\theta}_n}$ is the Jacobian of target function (in our case the negative log

likelihood function) with respect to $\boldsymbol{\theta}$ evaluated at $\boldsymbol{\theta}_n$, and can be (are) approximated using numerical derivatives.

(b) Solve for step size α via backtracking line search[1]:
$\alpha = \text{argmin } -2\log\ell(\boldsymbol{\theta}_n + \alpha\boldsymbol{\theta}_{direction})$.

(c) Update parameters: $\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n + \boldsymbol{s}$; $\boldsymbol{s} = \alpha\boldsymbol{\theta}_{direction}$

(d) Calculate differences in the Jacobians: $\boldsymbol{d} = G_{\boldsymbol{\theta}_{n+1}} - G_{\boldsymbol{\theta}_n}$

(e) Update inverse of Hessian: $H^{-1}_{\boldsymbol{\theta}_{n+1}} = (I - \frac{\boldsymbol{sd^T}}{\boldsymbol{d^T s}})H^{-1}_{\boldsymbol{\theta}_n}(I - \frac{\boldsymbol{ds^T}}{\boldsymbol{d^T s}}) + \frac{\boldsymbol{ss^T}}{\boldsymbol{d^T s}}$
Alternatively, the above can be computed more efficiently by: $H^{-1}_{\boldsymbol{\theta}_{n+1}} = H^{-1}_{\boldsymbol{\theta}_n} + \frac{(\boldsymbol{s^T d} + \boldsymbol{d^T} H^{-1}_{\boldsymbol{\theta}_n}\boldsymbol{d})(\boldsymbol{ss^T})}{\boldsymbol{s^T d}^2} - \frac{H^{-1}_{\boldsymbol{\theta}_n}\boldsymbol{ds^T} + \boldsymbol{sd^T}H^{-1}_{\boldsymbol{\theta}_n}}{\boldsymbol{s^T d}}$.

For dual estimation of model parameters and latent states, the final latent state estimates are done by running the Kalman filter (or EKF) through all the observations with the final parameter estimates at convergence.

In summary, the EKF utilizes local linearization to handle nonlinear transition functions, and should perform relatively well for weak nonlinearity, but still relies on assumptions that noises are normally distributed. Therefore it would be interesting to examine its robustness when the normality assumptions are violated.

[1]Backtracking line search usually involves:

i. Initialize: Select maximum step size α_0, and control parameters $\tau \in (0,1)$ and $c \in (0,1)$.

ii. Repeat $\alpha_n = \tau\alpha_{n-1}$ until the Armijo-Goldsterin condition is fullfilled:

$$f(\boldsymbol{\theta} + \alpha\boldsymbol{\theta}_{direction}) < f(\boldsymbol{\theta}) + c\alpha G(\boldsymbol{\theta})^T\boldsymbol{\theta}_{direction}$$

Chapter 3

The Sequential Monte Carlo Approach

This chapter describes the second type of algorithms this project aims to investigate: the particle filter, which is a common sequential Monte Carlo method, for latent state estimation as well as the related iterated filtering algorithm for dual-estimation.

3.1 The Particle Filter

The particle filter method is a sequential Monte Carlo method. The key idea of particle filtering is to approximate the distribution of $\boldsymbol{\eta}_t|\boldsymbol{y}_t$ with a sample, or a "particle swarm" ($\boldsymbol{\eta}_t^{*()}$, $^{()}$ will be used to indicate a swarm in the notations to follow), generated according to the dynamic model (Doucet et al., 2001; Ristic et al., 2004). Each particle in the swarm is generated and evolved independently of the other particles, thus each particle can be seen as a Monte Carlo replication, and moments of interests (such as mean and variance of $\boldsymbol{\eta}_t|\boldsymbol{y}_t$, mirroring the Kalman Filter scheme) can be approximated with the moments of the sample distribution. The particle filter provides an unbiased estimate of the likelihood (and thus a biased but consistent estimate of the log likelihood), factoring it in such a way (King and Ionides, 2019):

$$\ell(\boldsymbol{\theta}) = f_{\boldsymbol{y}_{1:T}}(\boldsymbol{y}_{1:T}; \boldsymbol{\theta})$$

$$= \prod_{t=1}^{T} f_{\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}}(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1};\boldsymbol{\theta})$$
$$= \prod_{t=1}^{T} \int f_{\boldsymbol{y}_t|\boldsymbol{\eta}_t}(\boldsymbol{y}_t|\boldsymbol{\eta}_t;\boldsymbol{\theta}) f_{\boldsymbol{\eta}_t|\boldsymbol{y}_{1:t-1}}(\boldsymbol{\eta}_t|\boldsymbol{y}_{1:t-1};\boldsymbol{\theta}) d\boldsymbol{\eta}_t$$

Two distributions are important to the particle filter: the prediction distribution and the filtering distribution. The prediction distribution follows the Markov property of the first-order difference equation model in our problem:

$$f_{\boldsymbol{\eta}_t|\boldsymbol{y}_{1:t-1}}(\boldsymbol{\eta}_t|\boldsymbol{y}_{1:t-1};\boldsymbol{\theta}) = \int f_{\boldsymbol{\eta}_t|\boldsymbol{\eta}_{t-1}}(\boldsymbol{\eta}_t|\boldsymbol{\eta}_{t-1};\boldsymbol{\theta}) f_{\boldsymbol{\eta}_{t-1}|\boldsymbol{y}_{1:t-1}}(\boldsymbol{\eta}_{t-1}|\boldsymbol{y}_{1:t-1};\boldsymbol{\theta}) d\boldsymbol{\eta}_{t-1}. \quad (3.1)$$

The filtering distribution can be derived with Bayes' theorem:

$$f\boldsymbol{\eta}_t|\boldsymbol{y}_{1:t}(\boldsymbol{\eta}_t|\boldsymbol{y}_{1:t};\boldsymbol{\theta}) \quad (3.2)$$
$$= f_{\boldsymbol{\eta}_t|\boldsymbol{y}_t,\boldsymbol{y}_{1:t-1}}(\boldsymbol{\eta}_t|\boldsymbol{y}_t,\boldsymbol{y}_{1:t-1};\boldsymbol{\theta}) \quad (3.3)$$
$$= \frac{f_{\boldsymbol{y}_t|\boldsymbol{\eta}_t}(\boldsymbol{y}_t|\boldsymbol{\eta}_t;\boldsymbol{\theta}) f_{\boldsymbol{\eta}_t|\boldsymbol{y}_{1:t-1}}(\boldsymbol{\eta}_t|\boldsymbol{y}_{1:t-1};\boldsymbol{\theta})}{\int f_{\boldsymbol{y}_t|\boldsymbol{\eta}_t}(\boldsymbol{y}_t|\boldsymbol{x}_t;\boldsymbol{\theta}) f_{\boldsymbol{\eta}_t|\boldsymbol{y}_{1:t-1}}(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1};\boldsymbol{\theta}) d\boldsymbol{x}_t} \quad (3.4)$$

As the name "sequential Monte Carlo" suggests, the particle filter approximates the integrals in the above two distributions sequentially through each t from 1 to T. Assume that, with $\boldsymbol{\theta}$ known, we have a state particle swarm of size J, that is, it consists of J independent particles. At each iteration, to estimate $\boldsymbol{\eta}_t$, the particle filter also consists of two steps:

Step 1: prediction (with the particle swarm carried from the last iteration and derived from information contained in observations $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{t-1}\}$

First, generate a new set of particle swarm according to:

$$\boldsymbol{\eta}_t^{()} \sim f_{\boldsymbol{\eta}_t|\boldsymbol{\eta}_{t-1}}(\boldsymbol{\eta}_t|\boldsymbol{\eta}_{t-1}^{*()};\boldsymbol{\theta}). \quad (3.5)$$

$\boldsymbol{\eta}_t^{()}$ approximates a draw from the prediction distribution in Equation 3.1.

Step 2: update (or more accurately, resample)

When a new occasion of observation becomes available, we can evaluate the like-

lihood of the prediction swarm:

$$\boldsymbol{w}_t^{(j)} = f_{\boldsymbol{y}_t|\boldsymbol{\eta}_t}(\boldsymbol{y}_t|\boldsymbol{\eta}_t^{(j)};\boldsymbol{\theta});\ j = 1,\ldots,J,$$

and use $\boldsymbol{w}_t^{()}$ as resampling weights. The likelihood associated with this particular set of parameters $\boldsymbol{\theta}$ can be approximated by the average of this Monte Carlo sample:

$$\ell(\hat{\boldsymbol{\theta}}) = \frac{1}{J}\sum_{j}^{J} \boldsymbol{w}_t^{(j)}$$

Since linear measurement functions and normally distributed measurement errors are assumed in this thesis, $\boldsymbol{w}_t^{(j)} = \boldsymbol{\psi}_{\boldsymbol{\Lambda}\boldsymbol{\eta}_t^{(j)},\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}}(\boldsymbol{y}_t)$,where $\boldsymbol{\psi}_{\boldsymbol{\mu},\boldsymbol{\Sigma}}$ is the probability density function of a multivariate normal distribution with mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\Sigma}$. Then $\boldsymbol{\eta}_t^{()}$ are resampled according to $\boldsymbol{w}_t^{()}$ and the resampled particles, $\boldsymbol{\eta}_t^{*()}$, are carried into the next iteration. The above two steps are then repeated until the final time of observation, T, is reached.

3.2 Iterated filtering (IF) for Dual Estimation

IF[1], as the name suggests, is an iterative algorithm that involves, in each iteration, a complete run of particle filter carried out through every timepoint within the time-series for the parameter particles of this iteration. At the end of each particle filter (when the particle filter reaches the end of the time-series), the parameter particles are recycled to be perturbed (with a perturbation density $h(.,\sigma_{h_m})$ of choice, but often a random walk is adopted with σ_{h_m} being the standard deviation of the walk in iteration m) and generate new starting parameters for the next iteration of particle filter. The idea of iteration-specific $\sigma_{h_m}$is to shrink the perturbation intensity through iterations so the algorithm will explore regions with higher likelihood in a gradually local scale. A flow chart of IF is included in Figure 3.1. King and Ionides (2019) claimed that this procedure converges towards the parameter space maximizing the likelihood. This was proved in Theorem 2 in Ionides et al. (2015) and subsequently demonstrated in a simulated toy example, where the la-

[1]The IF algorithm I adopted was the IF2 algorithm in Ionides et al. (2015). Variations prior to this version exist but this one was shown to outperform.

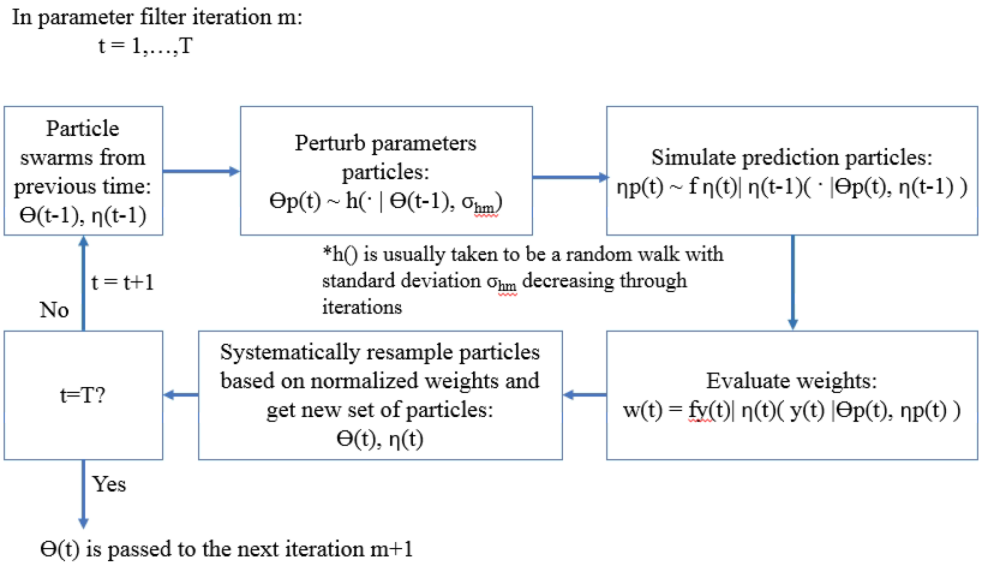

Figure 3.1. Flow Chart of Iterated Filtering at Each Iteration m

tent process was "constant and nonrandom" ($\boldsymbol{\eta}_t = (exp(\theta_1), \theta_2 exp(\theta_1))$, θ_1 and θ_2 being unknown parameters) and the measurement model followed a multivariate Normal distribution.

The attractive property of the particle filter (and subsequently the iterated filtering) is that it only requires, from the model of interest, that realizations of it can be generated at arbitrary parameter values ("plug-and-play property", in Bretó et al., 2009), which is true for most models. However, there is often a tradeoff between higher estimation accuracy gained with more particles and additional computational time that sometimes is not trivial. As will be described in the literature review in section 4.1, the term "particle filter" is a general method that include a lot of variations mainly stemmed from different proposal densities one can choose to generate the particles from. In the case of this project, I adopt the widely used proposal density, the transition density $f_{\boldsymbol{\eta}_t|\boldsymbol{\eta}_{t-1}}$, so the resampling weight would be equal to the conditional likelihood (Van Der Merwe et al., 2001). However, it is worth pointing out that performance of particle filters can largely depend on the proposal density, especially when the model gets complex, resulting in the a difficult-to-probe likelihood surface (for example, as discussed in Freitas et al. (2000), if high likelihood occurs at the tail of the proposal density). Though it is straight-forward to carry out the central piece of particle filter, the various

details make it hard to generalize results when compared to other algorithms.

Chapter 4

Comparing Algorithms

This chapter starts out by reviewing the existing research that has compared the two families of algorithms, and follows with my own implementation of the two algorithms, EKF paired with BFGS and IF, for a set of simulated conditions: one with perfectly normally distributed noises in the generated data, and one with positively skewed noises. Results for parameter estimation, in terms of biases, root-mean-square errors (RMSEs) and standard errors (SEs), as well as latent state estimation based on the parameter estimates, in terms of mean-square-error (MSE), were compared.

4.1 Literature Review

In simulation studies involving various adaptations of the particle filter as well as the EKF, the particle filtering approach was mostly reported to outperform the Kalman filter family approach (e.g. Alkhatib et al., 2008; Arulampalam et al., 2007). For example, Van Der Merwe et al. (2001) compared the EKF and the particle filter through a simulation example using a transformation function with an additive cyclic curve and an additive process noise following a gamma distribution, coupled with a nonstationary observation (measurement) model, and the various particle filter approaches on average showed less MSE in the estimation of latent states (in terms of the mean and variance) compared to EKF. However, in Van Der Merwe et al. (2001)'s simulation study, the best performance in terms of MSE was from a particle filtering approach using the unscented transformation

to calculated the proposal distribution in each step to draw particles from. As demonstrated by the above mentioned studies and noted by the authors, the crucial component of particle filter is the proposal density from which the particles are drawn. Therefore the performance of particle filter is dependent on a carefully selected "good" proposal. A generic particle filter may perform worse than EKF. In addition, as Daum (2005) pointed out, one obvious shortcoming of the particle filter, being a Monte Carlo method, is the limitation by real time computer speed, which then limits its performance for higher dimensional problems if the problems are less "nice".

The various possible adaptations of the filters mentioned make it hard to compare them universally. A certain level of problem-specific fine tuning is required in the particle filter. In addition, comparisons when the problem involves dual-estimation are relatively rare. Hence, a simulation study was conducted to extend the comparison between two filtering approaches (EKF and the particle filter) with the aim of dual-estimation.

4.2 A Simulation-based Comparison

Given the rare cases of algorithm comparison in the problems of dual-estimation, and the increasing demand in the study of human development in solving such problems, this thesis aims to further expand the existing literature, using state space models that have been utilized in modeling human behaviors and challenges from nonnormality in these models that data collection in studies of human behaviors often present. Two conditions were simulated with a linear state space model: one with normally distributed state spaces as a baseline comparison, and another with a skewed process noise error distribution that would violate the normality assumptions underlying EKF.

Data were simulation from a simplified one dimensional realization of Equations 1.1 - 1.3:

$$\eta_t = \beta\eta_{t-1} + \zeta_t$$
$$y_t = \eta_t + \epsilon_t$$
$$\epsilon_t \sim N(0, \sigma_\epsilon^2).$$

In the first condition, ζ_t assumes:

$$\zeta_t \sim N(0, \sigma_\zeta^2)$$

with the parameter vector $\boldsymbol{\theta}$=$\{\beta = 0.5, \sigma_\zeta^2 = 0.25, \sigma_\epsilon^2 = 1\}$. In the second condition, ζ follows a skewed normal distribution:

$$f(\zeta_t) = \frac{2}{\omega}\phi(\frac{\zeta_t - \xi}{\omega})\Phi(\alpha\frac{\zeta_t - \xi}{\omega}), \tag{4.1}$$

where $\phi(.)$ and $\Phi(.)$ are the probability density function and cumulative distribution function respectively for a standard normal distribution. The mean and variance of the generated skewed distribution were controlled to be the same as in the first condition (mean = 0, variance = 0.25), and α was selected in a way that the generated skewness was close to the expected skewness in empirical research studies[1]. The two control criteria above yielded the setup of $\{\alpha = 2, \omega = 0.7138, \xi = 0.5094\}$ in Equation 4.1. Comparisons of the generated process noises under two conditions can be found in Figure 4.1.

In both conditions, three parameters needed to be estimated. Due to the increase in difficulty involved in multi-dimensional optimization, the parameter estimations were compared in one-dimensional, two-dimensional, and three-dimensional optimization. In one-dimensional estimation, only one parameter was estimated in each Monte Carlo replication, and the other two were held fixed at the true data generating value, whereas in two-dimension estimation, parameters were grouped in three groups with two parameters each, and the remaining one was fixed at the true value. For the EKF, once the parameter optimization was done, the final parameter estimates were used to run the EKF one more time to achieve the latent state estimates. On the other hand, in the IF procedure, I adopted a setting of 100 parameter iterations ($M = 100$) though five parallel chains, and at the last iteration, the estimate(s) from the chain that achieved the highest log likelihood (approximated by a separate particle filtering procedure using estimate(s) at the last iteration) was selected as the final estimate(s). The latent state estimates from

[1]Cain et al. (2017) reviewed 1567 univariate measures from 194 studies appeared in two prestigious journals in Psychology and Education and found the sample size-weighted mean skewness is 0.47. In this simulation, α was selected to be 2, which would yield an expected skewness to 0.454

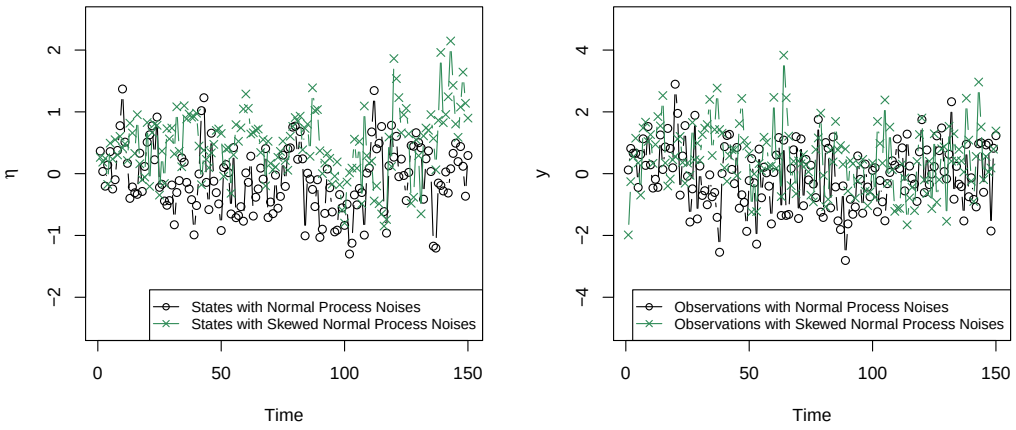

Figure 4.1. An example of generated skewed process noises compared with normal process noises in trajectories of latent states (left) and manifest observations (right): a collection of positively skewed process noises resulted in both the latent states and observations to exhibit somewhat upwards trajectories.

the corresponding chain were then treated as the final latent state estimates from IF associated with that particular trial. The perturbation standard deviation σ_{h_m} was set to start as 0.02 (recommended by King and Ionides (2019)) and follow a two-step geometric shrinkage schedule such that it shrinks to 0.01 at $m = 50$ and 0.002 at $m = 100$.

A comparison of estimated parameters for the first condition can be found in Figures 4.2 -4.4 and Tables 4.1 - 4.3. With normal process noises, all parameters are recovered quite well on average for both algorithms, except when the optimization became three-dimensional, where EKF yielded larger biases and RMSEs compared to IF (Table 4.3). Consequently, the MSE for latent state estimation in the three-dimensional optimization condition was higher for EKF, whereas in one-dimensional and two-dimensional conditions the MSEs were comparable (Table 4.4).

In the second simulation condition with positively skewed process noises, performances were less satisfactory for both algorithms. To begin with, the optimization scheme associated with the EKF exhibited a certain possibility of failure, where the parameter probing stopped at the boundary values (e.g. with the σ_ζ^2 probing trajectory eventually going to 0). A total of 24% MC trials failed in the two-

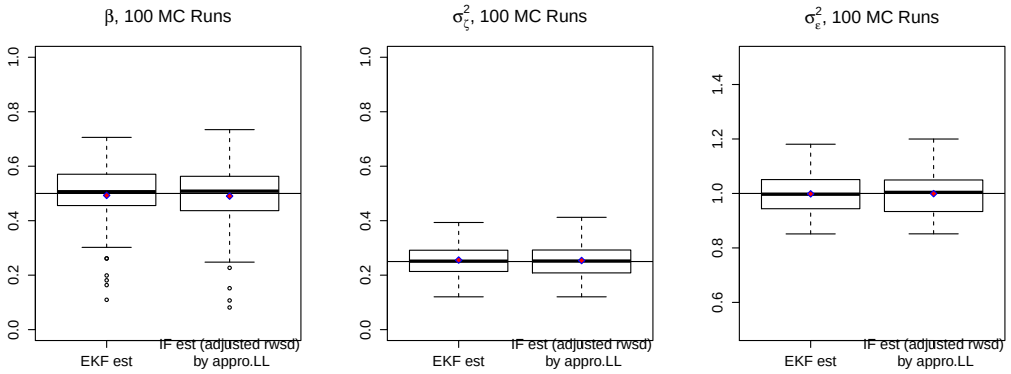

Figure 4.2. Results comparison (one-dimensional optimization) under the first simulation condition: the two algorithms behave similarly in terms of the distributions of estimated parameters, and the means of the estimates are close to the true values marked by the horizontal lines.

	Bias		RMSE			MCSD		SE	
	EKF	IF	EKF	IF		EKF	IF	EKF	IF
β	-0.007	-0.010	0.112	0.117	β	0.112	0.117	0.071	0.049
σ^2_ζ	0.005	0.005	0.064	0.066	σ^2_ζ	0.064	0.066	0.002	0.039
σ^2_ϵ	-0.002	-0.001	0.076	0.078	σ^2_ϵ	0.076	0.078	0.003	0.044

Table 4.1. Performance comparison in condition 1 with 1-D optimization. MCSD: Monte Carlo standard deviation. Biases and RMSEs from the two approaches are comparable in value. SEs from both approaches are underestimated.

dimensional optimization of (β,σ^2_ζ), and 63% trials failed in the three-dimensional optimization for the EKF. With the trials that did come to their final parameter estimates, the estimates were a lot more biased across the board compared to the normal process noise condition. When comparing the successful trials in the EKF with IF for which there was no failure, the EKF consistently yielded higher relative biases and RMSEs in parameter estimation (Tables 4.5 - 4.7 and Figures 4.5 - 4.7).

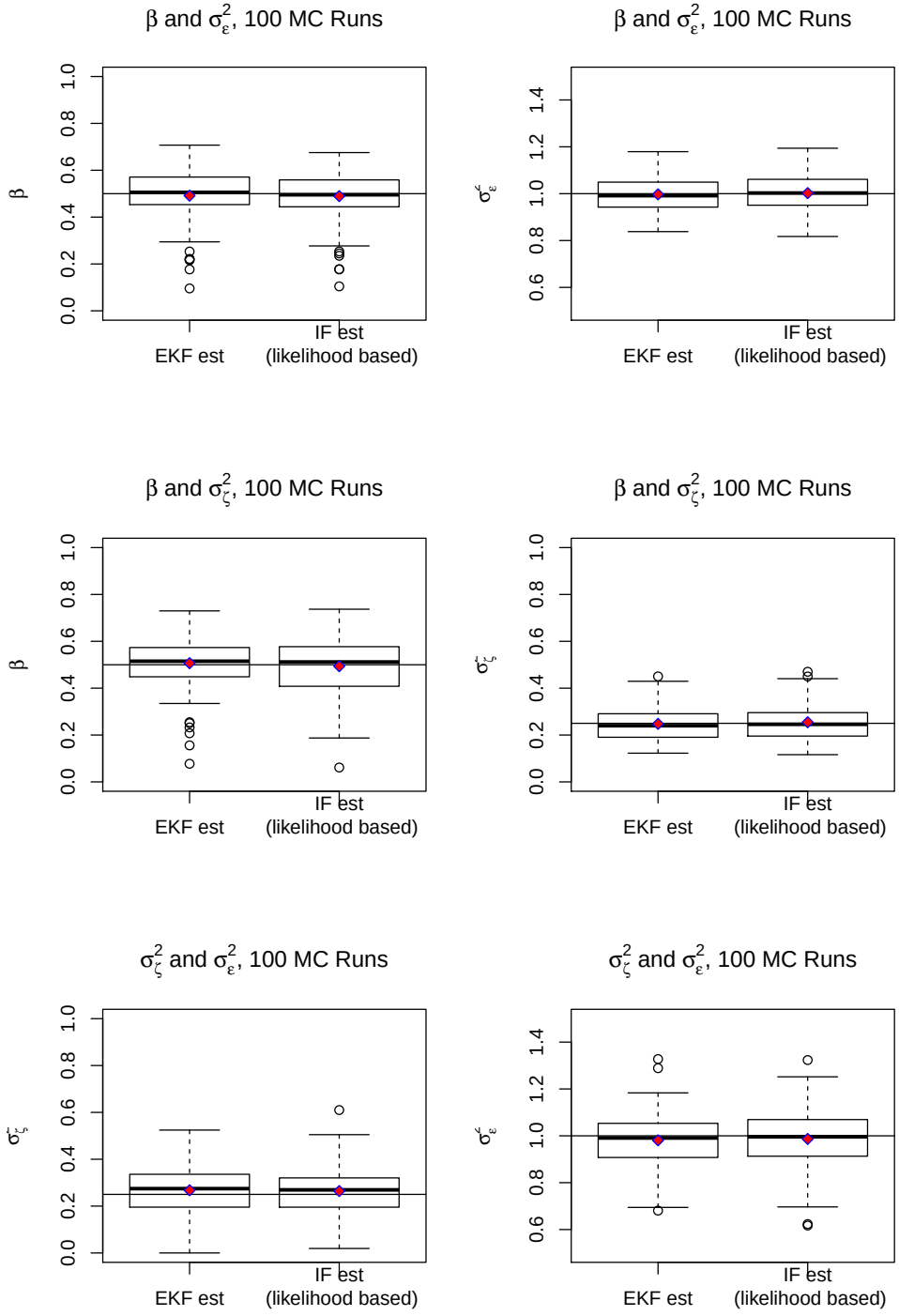

Figure 4.3. Results comparison (two-dimensional optimization) under the first simulation condition: the two algorithms behave similarly in the resulted distributions of estimated parameters, and the means of the estimates are close to the true values marked by the horizontal lines.

	Bias				RMSE			
	Param1		Param2		Param1		Param2	
	EKF	IF	EKF	IF	EKF	IF	EKF	IF
(β,σ^2_ζ)	0.007	-0.002	-0.004	0.004	1.200	0.731	1.259	0.763
$(\beta,\sigma^2_\epsilon)$	-0.005	-0.006	-0.010	0.000	1.089	0.745	1.116	0.797
$(\sigma^2_\zeta,\sigma^2_\epsilon)$	0.018	-0.019	0.014	-0.013	1.025	1.215	1.057	1.210
	MCSD				SE			
	Param1		Param2		Param1		Param2	
	EKF	IF	EKF	IF	EKF	IF	EKF	IF
(β,σ^2_ζ)	0.120	0.073	0.126	0.077	0.099	0.042	0.004	0.004
$(\beta,\sigma^2_\epsilon)$	0.109	0.075	0.112	0.080	0.071	0.057	0.004	0.004
$(\sigma^2_\zeta,\sigma^2_\epsilon)$	0.101	0.121	0.105	0.121	0.069	0.056	0.004	0.004

Table 4.2. Performance comparison in condition 1 with 2-D optimization. Biases from the two approaches are comparable in value. IF results in smaller RMSEs in two out of the three parameter combinations. Similar to the results of 1-D optimization, SEs are underestimated.

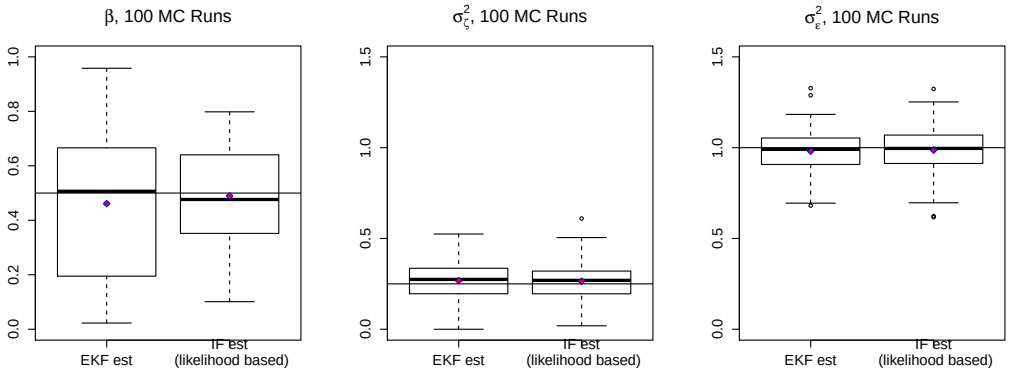

Figure 4.4. Results comparison (three-dimensional optimization) under the first simulation condition: the two algorithms behave similarly for the two variance parameters σ^2_ζ and σ^2_ϵ, but for β, IF resulted in a more condensed collection of parameter estimates, and the mean was slightly closer to the true value (horizontal line).

	Bias		RMSE			MCSD		SE	
	EKF	IF	EKF	IF		EKF	IF	EKF	IF
β	-0.039	-0.010	2.476	1.755	β	0.246	0.176	0.139	0.000
σ^2_ζ	0.272	0.070	5.510	2.185	σ^2_ζ	0.482	0.208	0.148	0.000
σ^2_ϵ	-0.268	-0.071	5.332	2.218	σ^2_ϵ	0.464	0.211	0.056	0.000

Table 4.3. Performance comparison in condition 1 with 3-D optimization. Biases from the two approaches are comparable in value. IF outperforms the EKF in terms of biases and RMSEs, but underestimates the SE more than the EKF.

	EKF	IF
1D: β	0.236	0.236
1D: σ^2_ζ	0.238	0.238
1D: σ^2_ϵ	0.235	0.235
2D: β, σ^2_ζ	0.238	0.239
2D: β, σ^2_ϵ	0.236	0.237
2D: σ^2_ζ, σ^2_ϵ	0.245	0.246
3D	0.450	0.270

Table 4.4. Comparison of mean-square-error (MSE) for latent states estimation in condition 1. IF has smaller MSE for 3-D optimization, and is comparable to the EKF in 1-D and 2-D.

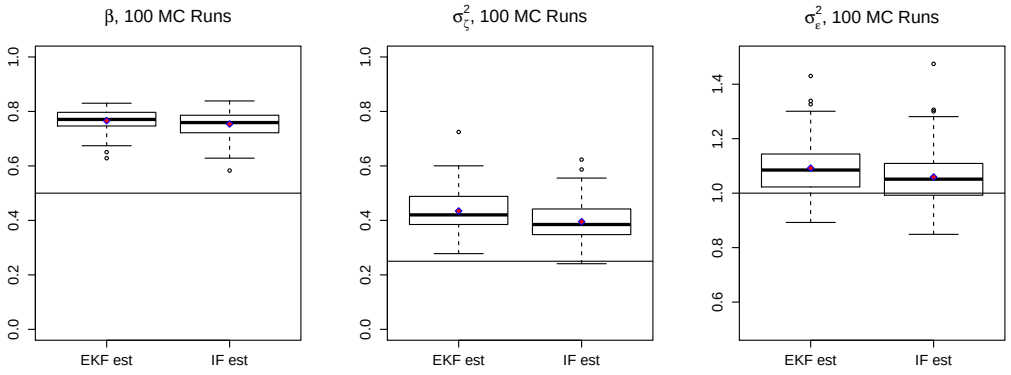

Figure 4.5. Results comparison (one-dimensional optimization) under the second simulation condition: results from the two algorithms are similarly off from the true parameter values (horizontal lines), with IF on average having estimates slightly closer to true values.

	Relative Bias		RMSE			MCSD		SE	
	EKF	IF	EKF	IF		EKF	IF	EKF	IF
β	0.532	0.508	0.269	0.258	β	0.041	0.047	0.030	0.036
σ_ζ^2	0.738	0.582	0.198	0.161	σ_ζ^2	0.073	0.070	0.003	0.041
σ_ϵ^2	0.092	0.059	0.135	0.120	σ_ϵ^2	0.100	0.105	0.004	0.048

Table 4.5. Performance comparison in condition 2 with 1-D optimization. IF yields consistently smaller relative bias and RMSE, and underestimates the SE less.

	Relative Bias				RMSE			
	Param1		Param2		Param1		Param2	
	EKF	IF	EKF	IF	EKF	IF	EKF	IF
(β,σ_ζ^2)	0.741	-0.501	0.748	-0.547	0.387	0.168	0.385	0.164
$(\beta,\sigma_\epsilon^2)$	0.532	-0.006	0.509	-0.012	0.267	0.090	0.259	0.095
$(\sigma_\zeta^2,\sigma_\epsilon^2)$	1.320	-0.242	1.038	-0.210	0.347	0.270	0.279	0.245
	MCSD				SE			
	Param1		Param2		Param1		Param2	
	EKF	IF	EKF	IF	EKF	IF	EKF	IF
(β,σ_ζ^2)	0.111	0.113	0.092	0.091	0.038	0.020	0.032	0.029
$(\beta,\sigma_\epsilon^2)$	0.042	0.090	0.051	0.095	0.030	0.056	0.038	0.042
$(\sigma_\zeta^2,\sigma_\epsilon^2)$	0.108	0.122	0.103	0.127	0.083	0.057	0.039	0.039

Table 4.6. Performance comparison in condition 2 with 2-D optimization. IF yields consistently smaller relative bias and RMSE. Both algorithms underestimate SEs.

	Relative Bias		RMSE			MCSD		SE	
	EKF	IF	EKF	IF		EKF	IF	EKF	IF
β	0.867	0.831	0.440	0.422	β	0.078	0.076	0.032	0.029
σ_ζ^2	-0.748	-0.721	0.209	0.198	σ_ζ^2	0.094	0.083	0.012	0.024
σ_ϵ^2	0.163	0.156	0.204	0.199	σ_ϵ^2	0.124	0.123	0.057	0.047

Table 4.7. Performance comparison in condition 2 with 3-D optimization. The two algorithms are comparable in their relative bias and RMSE. Both algorithms underestimate SEs.

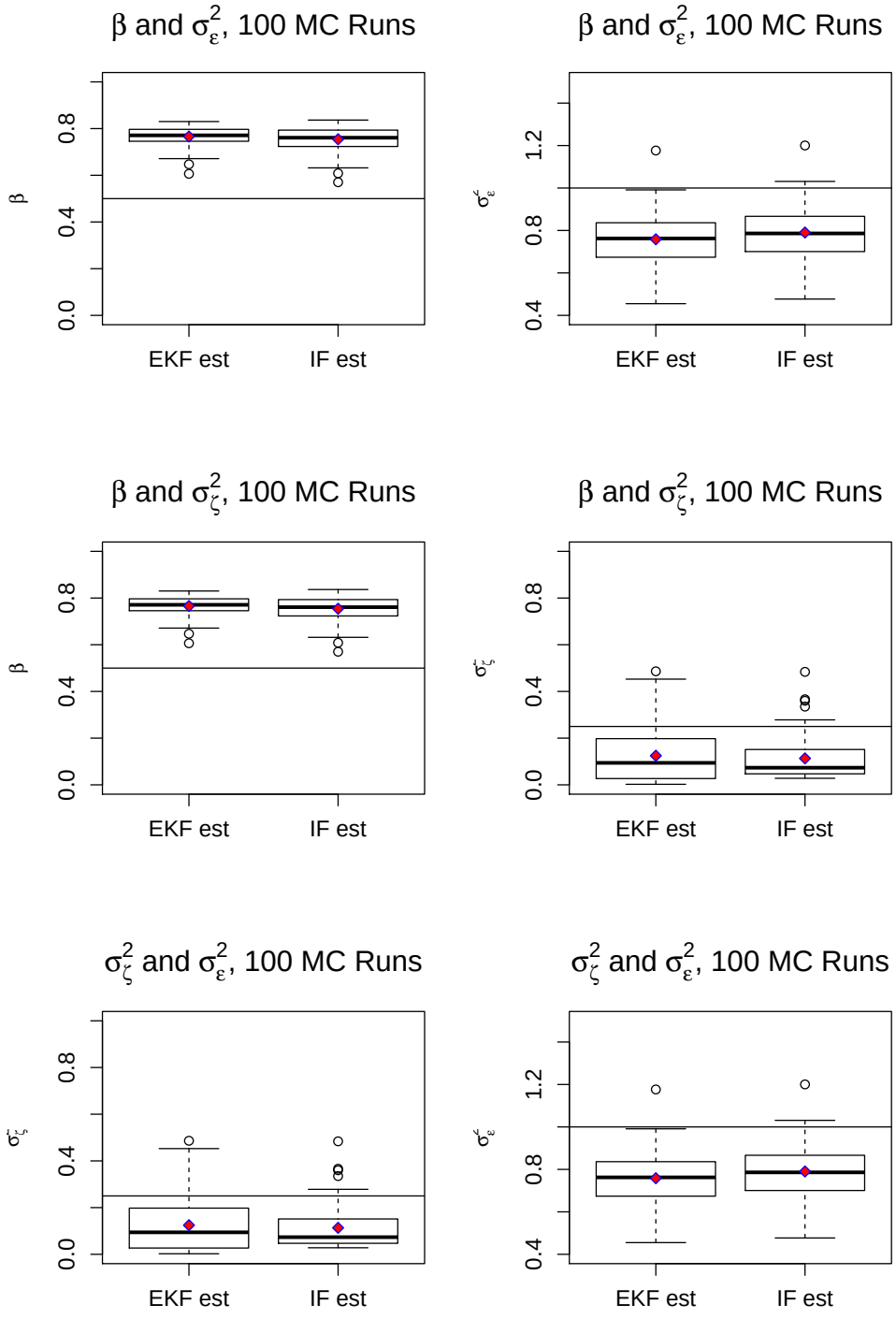

Figure 4.6. Results comparison (two-dimensional optimization) under the second simulation condition: results from the two algorithms are similarly off from the true parameter values (horizontal lines), with IF on average having estimates slightly closer to true values.

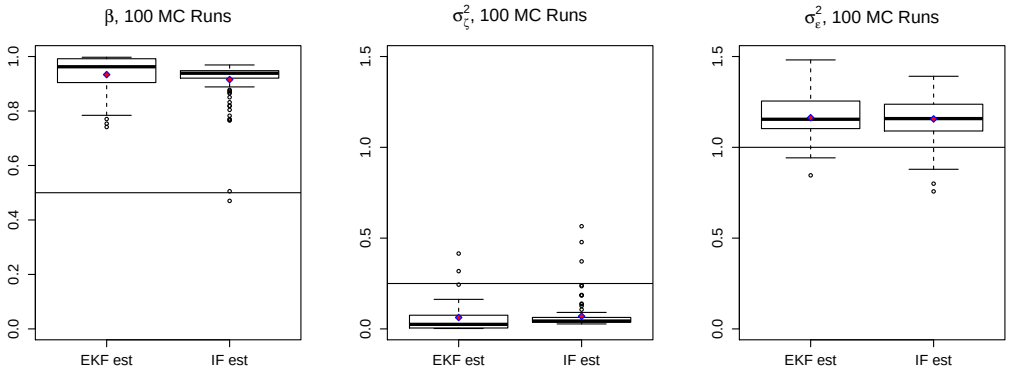

Figure 4.7. Results comparison (three-dimensional optimization) under the second simulation condition: results from the two algorithms are similarly off from the true parameter values. The EKF may exhibit a wider spread because the results were based on the 37 successful MC trials.

	EKF	IF
1D: β	0.293	0.296
1D: σ^2_ζ	0.314	0.319
1D: σ^2_ϵ	0.337	0.333
2D: β, σ^2_ζ	0.299	0.295
2D: β, σ^2_ϵ	0.293	0.297
2D: σ^2_ζ, σ^2_ϵ	0.352	0.348
3D:	0.312	0.302

Table 4.8. Comparison of mean-square-error (MSE) for latent states estimation in condition 2. The two algorithms are comparable.

Chapter 5

Discussion

This chapter gives an overview of the comparison conclusion given the results of the simulations, followed by caveats one should take note of when considering the two algorithms as well as the simulation results, and some options worth exploring in the future.

5.1 Summary of Conclusions

This project set out to investigate two algorithms of different nature, the EKF paired with quasi-Newton scheme optimization and IF, in dual estimation of state space models. Of particular interest was how the two performed under violation of normality and linearity, two key assumptions for the commonly used Kalman filter. The EKF improves from the traditional Kalman filter with a local linearization of the state transition function, thus adapting the Kalman filter to mildly nonlinear models. However, the underlying prediction and update scheme of EKF (e.g. using means and covariance structures) still relies on properties of the normal distribution in order for the estimates to be reasonable. On the other hand, IF expands the pool of transition functions the system can take and process noise distributions, thanks to its Monte Carlo nature. It also is in theory less restrictive in handling distributional assumptions, since all the algorithm requires is that the probability of an observation given the underlying state can be evaluated. With the new set of simulations, I reached the following main conclusions:

1. The EKF paired with BFGS and IF both can recover model parameters under

linear transition and normal additive noises. In this ideal situation, the two algorithms performed almost identically, except that IF may be slightly more accurate in the three-dimensional optimization problem. The two algorithms are also comparable in terms of MSEs in the estimation of latent states $\eta_1 \dots \eta_T$.

2. In the second simulation condition where the process noise exhibit mild positive skewness, both algorithms resulted in biased parameter estimates, especially when the process noise parameter σ_ζ^2 was involved in the optimization. In one-dimensional and two-dimensional optimizations, IF slightly outperforms the EKF in terms of relative bias and RMSE, but the two are comparably biased in three-dimensional optimizations. Again, they are comparable in their latent state estimation.

3. Both algorithms showed lower average standard error estimates compared to the MC standard deviations for almost all conditions.

5.2 Caveats and Future Work

Though the simulation results show that IF outperforms slightly compared to the EKF approach, when one wants to choose between the two algorithms in dual-estimation, he/she needs to also take into consideration the computational resources the two algorithm requires. The computational time one needs to spend on these two algorithms are vastly different. Although this is to be expected given the simulation nature of IF, this may come as a hassle with larger datasets or more difficult problems (e.g. higher dimension that may demands larger particle swarms/more iterations of IF). Under the same computer configuration (2.2 GHz Intel Xeon Processor, 24 CPU/server. 128 GB RAM), in this simulation study, the time spent on IF for optimization and then state estimation with the final optimized parameters is almost 300 times greater than that of EKF (e.g. for one-dimensional optimization: 6.19 seconds on average for EKF, and 37.83 minutes for IF; for a detailed table for computational time please see Table 5.1)

The computation time required by IF is determined by the size of particle swarms, total iterations of parameter sampling, number of replications to check

Algorithm	One-dimensional	Two-dimensional	Three-dimensional
EKF	6.186	17.601	55.842
IF	2277.589	2872.327	3759.899

Table 5.1. Average Computation Time (in seconds)

MC convergence, and parallelization scheme. An aspect of IF that required additional computational time was the dependency on MC approximations. King and Ionides (2019) in their example used multiple independent chains (or, replications) of the same IF procedure with the same dataset, and at the end of each replication used the particle filter (again multiple independent replications of them) to acquire an approximated likelihood associated with final parameter estimate. In the particular execution of simulation studies in this project, I parallelized the independent replications for IF but did not parallelize the particle filtering step used for likelihood calculation at the end. The time in Table 5.1 is a reflection of such decision. Alternatively, one can also parallelize the particle filtering step to save some computational time. Figure 5.1 shows sample trajectories of five independent IF replications for the same dataset.

Often the decision for parameter perturbation schedule to be used in IF is not so trivial, however, during simulation I found that the effect of perturbation schedule on landing the final parameter estimates is minimal if one can afford to run the parameter sampling for enough iterations. In my simulation studies, I experimented with different starting random walk standard deviations and shrinkage speed. Figure 5.2 shows sample parameter trajectories with the same replication under different setups. With enough iterations they all travel to the same region in the parameter space. The setup ended up being used in the simulations was a result of these experimentation, with the goal of balancing exploration and "travel speed" of the parameter estimate.

This project focused on two algorithms that had potentials to be widely applied in dual-estimation for state space models in behavioral sciences. The possible algorithms for this aim are, of course, not exhausted by the two included here. Another algorithm that may be worth investigating is called the "unscented Kalman filter (UKF)"(Julier and Uhlmann, 1997). This algorithm is somewhat like a hybrid between the Kalman filter and the particle filter, in that it adopts part of

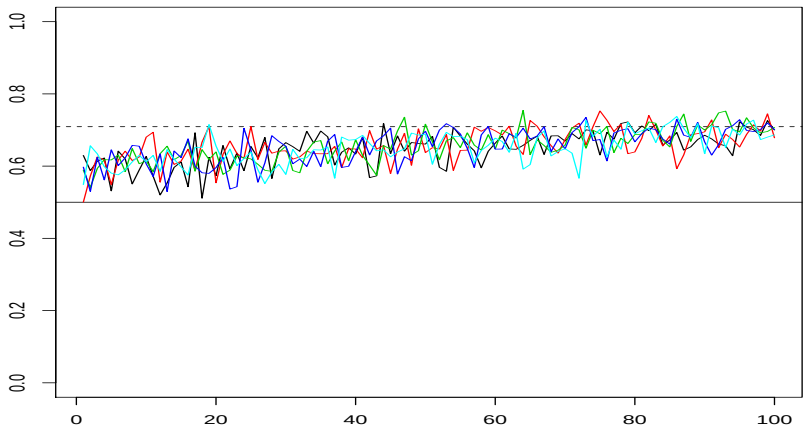

Figure 5.1. Trajectories of β Estimate with the Same Dataset and Different Replications

the general Kalman filter's reliance on the normal distribution properties (in the calculation and progression of means and covariance structures) but also involves particles, so that distributions of states are approximated by the empirical distributions of particle collections. However, instead of independently generating particles according to the target distribution (which would require a large number of particles to achieve a good approximation), Julier and Uhlmann (1997) proposed that by parameterizing according to the normal distribution, by selecting a set of particles (called "sigma points") that would capture the mean and span of the target distribution, the number of particles can be reduced yet accuracy is still preserved. The UKF is shown to generally outperform EKF under various conditions of nonlinearity in state estimation (Kandepu et al., 2008; Van Der Merwe et al., 2001), but may be less desired than the particle filter in some conditions (Arulampalam et al., 2007). Its functionality under dual estimation problems is yet to be determined. Nonetheless, the UKF is still worth investigating given that it may be able to strike a balance between the flexibility brought by particle swarms and the time-efficiency of the general Kalman scheme. In the sequential Monte Carlo realm, as discussed in Chapter 3 and 4, the particle filter can be adapted for more effective sampling. For example, Lee et al. (2019) proposed a fast adaptive particle-based algorithm, which extended Chopin (2002)'s Iterated Batch Importance Sampling Algorithm. Compared to the standard particle filter, this new

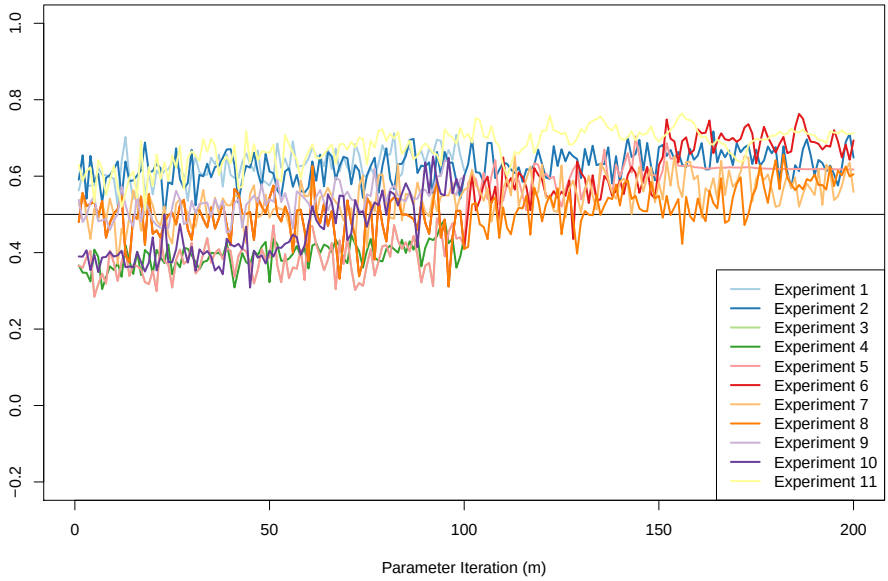

Figure 5.2. Trajectories of β Estimate under Different Combinations of Random Walk Parameters for One Replication

algorithm includes a mutation step after resampling the particles that runs short chains of an Markov Chain MC (MCMC) algorithm with an intermediate posterior distribution generated by likelihood tempering, with the tempering scheme re-evaluated each iteration. This algorithm stops when the entire likelihood has been incorporated, instead of when a pre-determined number of iterations finish running, and has been shown to have shorter computational time compared to the standard particle-based approach with mutation and full MCMC-based approach with a static model. It would be interesting to see how this algorithm can be adapted to a dynamic model.

The comparison of algorithms should also not be restricted to the conditions the current project studied. This project compared two algorithms under one condition of assumption violations to the traditional Kalman filter; it is also of interest to undergo the comparison under other kinds of violations, for example, nonlinear transition functions, which the EKF was first proposed to accommodate.

In conclusion, this project offers to the literature a simulated comparison of the EKF (paired with the BFGS algorithm for optimization) approach and the IF

approach with the aim of estimating both the parameter and the latent states of a state space dynamic model. It appears from the results that the two approaches perform more similarly than differently. Caution should be exercised when generalizing the results towards the overall performance of the algorithms. This project also illustrates the effects of skewed process noises on the estimation of dynamic model parameters under the prevalent dual-estimation approach in the field of psychology and human development, and alerts researchers on the validity of their conclusions drawn from such models when there is reason to believe that the noises are not normally distributed.

Bibliography

Alkhatib, H., Neumann, I., Neuner, H., and Kutterer, H. (2008). Comparison of Sequential Monte Carlo Filtering with Kalman Filtering for Nonlinear State Estimation. *1st International Conference on Machine Control & Guidance*, (1960):1–11.

Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp, T. (2007). A tutorial on particle filters for online nonlinear/nongaussian bayesian tracking. *Bayesian Bounds for Parameter Estimation and Nonlinear Filtering/Tracking*, 50(2):723–737.

Borsboom, D. and Cramer, A. O. (2013). Network Analysis: An Integrative Approach to the Structure of Psychopathology. *Annual Review of Clinical Psychology*, 9(1):91–121.

Bretó, C., He, D., Ionides, E. L., King, A. A., et al. (2009). Time series analysis via mechanistic models. *The Annals of Applied Statistics*, 3(1):319–348.

Bringmann, L. F., Ferrer, E., Hamaker, E. L., Borsboom, D., and Tuerlinckx, F. (2018). Modeling Nonstationary Emotion Dynamics in Dyads using a Time-Varying Vector-Autoregressive Model. *Multivariate Behavioral Research*, 3171:1–22.

Bringmann, L. F., Vissers, N., Wichers, M., Geschwind, N., Kuppens, P., Peeters, F., Borsboom, D., and Tuerlinckx, F. (2013). A network approach to psychopathology: new insights into clinical longitudinal data. *PloS one*, 8(4):e60188.

Cain, M. K., Zhang, Z., and Yuan, K.-H. (2017). Univariate and multivariate skewness and kurtosis for measuring nonnormality: Prevalence, influence and estimation. *Behavior Research Methods*, 49(5):1716–1735.

Chopin, N. (2002). A Sequential Particle Filter Method for Static Models. *Biometrika*, 89(3):539–551.

Chow, S.-M., Ou, L., Cohn, J. F., and Messinger, D. S. (2017). Representing self-organization and nonstationarities in dyadic interaction processes using dynamic

systems modeling techniques. In *Innovative assessment of collaboration*, pages 269–286. Springer.

Chow, S.-M., Zu, J., Shifren, K., and Zhang, G. (2011). Dynamic Factor Analysis Models With Time-Varying Parameters. *Multivariate Behavioral Research*, 46(2):303–339.

Daum, F. (2005). Nonlinear filters: Beyond the kalman filter. *IEEE Aerospace and Electronic Systems Magazine*, 20(8 II):57–68.

Doucet, A., De Freitas, N., and Gordon, N. (2001). An Introduction to Sequential Monte Carlo Method. In *Sequential Monte Carlo methods in practice*, chapter 1, pages 3–14. Springer, New York.

Freitas, J. d., Niranjan, M., Gee, A. H., and Doucet, A. (2000). Sequential monte carlo methods to train neural network models. *Neural computation*, 12(4):955–993.

Fried, E. I., van Borkulo, C. D., Cramer, A. O., Boschloo, L., Schoevers, R. A., and Borsboom, D. (2017). Mental disorders as networks of problems: a review of recent insights. *Social Psychiatry and Psychiatric Epidemiology*, 52(1):1–10.

Harvey, A. C. (1990). *Forecasting, structural time series models and the Kalman filter*. Cambridge university press.

Harvey, A. C. (2001). *Forecasting, structural time series models and the Kalman filter*. Cambridge University Press, Cambridge.

Haykin, S. (2004). *Kalman filtering and neural networks*, volume 47. John Wiley & Sons.

Ionides, E. L., Nguyen, D., Atchadé, Y., Stoev, S., and King, A. A. (2015). Inference for dynamic and latent variable models via iterated, perturbed bayes maps. *Proceedings of the National Academy of Sciences*, 112(3):719–724.

Julier, S. J. and Uhlmann, J. K. (1997). New extension of the Kalman filter to nonlinear systems. page 182.

Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems 1. *Journal of Fluids Engineering*, 82(Series D):35–45.

Kandepu, R., Foss, B., and Imsland, L. (2008). Applying the unscented Kalman filter for nonlinear state estimation. *Journal of Process Control*, 18(7-8):753–768.

Kantas, N., Doucet, A., Singh, S. S., Maciejowski, J., Chopin, N., et al. (2015). On particle methods for parameter estimation in state-space models. *Statistical science*, 30(3):328–351.

King, A. A. and Ionides, E. L. (2019). Likelihood-based inference for pomp models. https://kingaa.github.io/sbied/pfilter/pfilter.htmlsequential-monte-carlo-the-particle-filter. Accessed: 2019-07-24.

Kuppens, P., Allen, N. B., and Sheeber, L. B. (2010). Emotional Inertia and Psychological Maladjustment.

Lee, B. S., Haran, M., Fuller, R., Pollard, D., and Keller, K. (2019). A Fast Particle-Based Approach for Calibrating a 3-D Model of the Antarctic Ice Sheet.

Molenaar, P. C., Beltz, A. M., Gates, K. M., and Wilson, S. J. (2016). State space modeling of time-varying contemporaneous and lagged relations in connectivity maps. *NeuroImage*, 125:791–802.

Molenaar, P. C. M., de Gooijer, J. G., and Schmitz, B. (1992). Dynamic factor analysis of nonstationary multivariate time series. *Psychometrika*, 57:333–349.

Montella, C. (2014). The Kalman Filter and Related Algorithms A Literature Review. (May 2011):1–17.

Oravecz, Z., Tuerlinckx, F., and Vandekerckhove, J. (2016). Bayesian Data Analysis with the Bivariate Hierarchical Ornstein-Uhlenbeck Process Model. *Multivariate Behavioral Research*, 51(1):106–119.

Ristic, B., Arulampalam, S., and Gordon, N. (2004). Beyond the kalman filter. *IEEE Aerospace and Electronic Systems Magazine*, 19(7):37–38.

Schermerhorn, A. C., Cummings, E. M., DeCarlo, C. A., and Davies, P. T. (2007). Children's Influence in the Marital Relationship. *Journal of Family Psychology*, 21(2):259–269.

Schiff, S. J. (2012). Neural control engineering. *Computational Neuroscience ed TJ Sejnowski and TA Poggio (Cambridge, MA: MIT Press)*.

Tarvainen, M. P., Georgiadis, S., Lipponen, J. A., Hakkarainen, M., and Karjalainen, P. A. (2009). Time-varying spectrum estimation of heart rate variability signals with Kalman Smoother algorithm. *Proceedings of the 31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society: Engineering the Future of Biomedicine, EMBC 2009*, pages 1–4.

Thelen, E. and Smith, L. B. (2007). Dynamic Systems Theories. *Handbook of Child Psychology*, pages 258–312.

Van Der Merwe, R., Doucet, A., De Freitas, N., and Wan, E. (2001). The Unscented Particle Filter. *Advances in Neural Information Processing Systems*, 96(6080):584–590.

van Geert, P. L. C. (2018). Development, complexity and dynamical systems. *International Journal of Behavioral Development*, 42.

Wan, E. A., Van Der Merwe, R., and Nelson, A. T. (2000). Dual estimation and the unscented transformation. In *Advances in neural information processing systems*, pages 666–672.