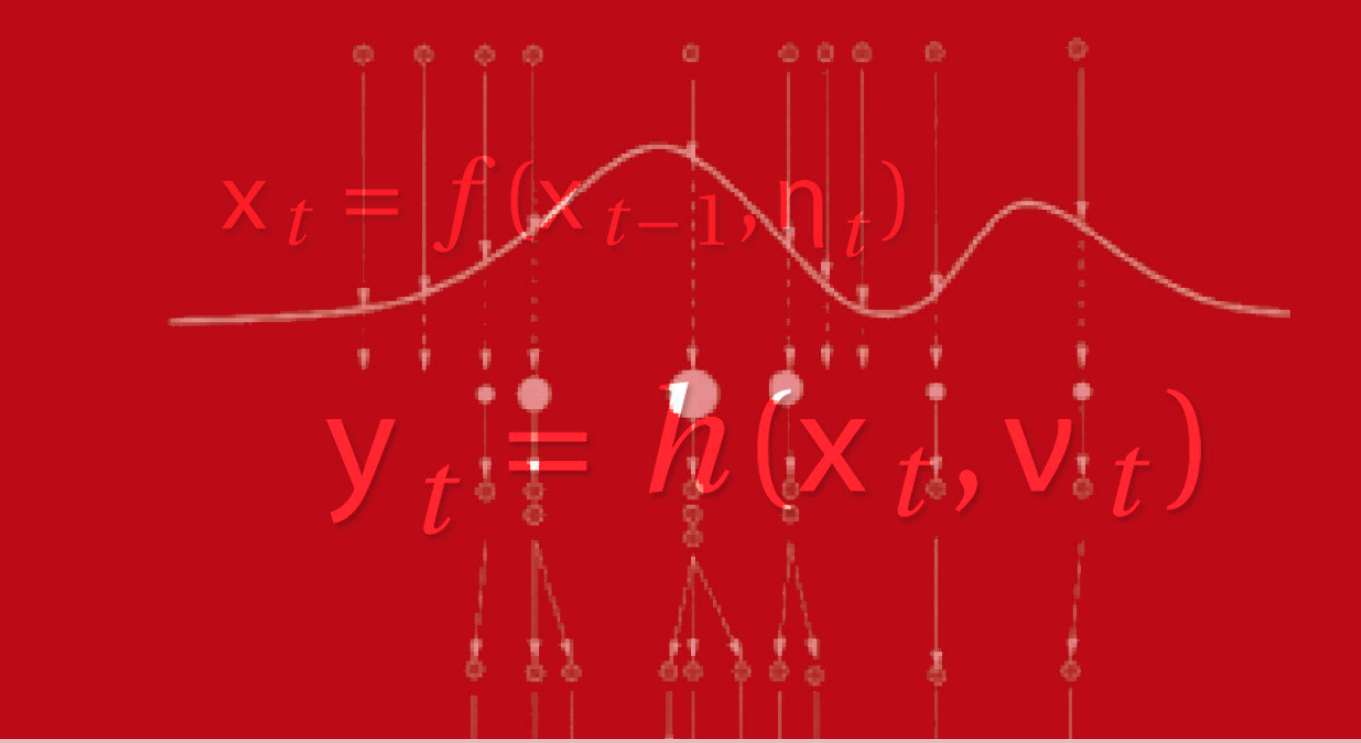


Particle Filtering Estimation for Linear and Nonlinear State-Space Models

Lesly María Acosta Argueta

PhD Thesis
Department of Statistics and Operations Research

UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Barcelona, 2013

Particle Filtering Estimation for Linear and Nonlinear State-Space Models

Lesly María Acosta Argueta
PhD Thesis directed by
Dr. M. Pilar Muñoz Gràcia

Department of Statistics and Operations Research
Technical University of Catalonia
BARCELONA**TECH**

Thesis presented in partial fulfillment of the requirements for the
Degree of Doctor by the Technical University of Catalonia
Technical and Computer Applications of Statistics, Operations Research and Optimization Program

2012–2013

About the Cover

The cover illustration subsumes the ideas presented under the title of the thesis "Particle Filtering Estimation for Linear and Nonlinear State-Space Models", as it represents the well-known transition-and-measurement equations that conform the possibly nonlinear dynamic state-space model, where estimation is carried out using the particle filtering methodology. The background illustration that is presented on the cover illustrates how a particle filter based on sampling importance resampling works. Notice that the plot shown is chosen from Figure 2.3 in Chapter 2, which is itself taken from Doucet, de Freitas, and Gordon (2001).

The cover of this thesis was illustrated by the graphic designer Isabel Flemisch.

Dedication

Para mi madre Bertha.
Para tí Klaus.

Y a la memoria de
mi padre Cesar † (Dec/2004).

Lámpara es a mis pies tu palabra y lumbrera a mi camino.
Salmos 119:105.

ACKNOWLEDGEMENTS

Creo firmemente en que gracias a la contribución de muchas *voluntades* hoy estoy culminando la meta de mi tesis doctoral y por ello quiero agradecer, en primer lloc, a la Dra. M. Pilar Muñoz, la directora d'aquesta tesi, per tot l'esforç que aquesta tesi també li ha suposat, sobretot gràcies per la seva paciència davant d'un treball que va requerir en el nostre argot de sèries temporals, una anàlisi d'intervenció no gaire estàndard. Me llena de satisfacción exclamar: ¡Meta lograda!

Faltan escasas dos horas para culminar el proceso oficial de depósito de mi tesis doctoral, pero quiero expresar mi profundo agradecimiento a todos aquellos, que de forma directa o indirecta, han contribuido a la consecución de una meta profesional y de un ansiado sueño.

Thank you very much/ Ganz herzlichen Dank/ Moltes gràcies/ Muchas gracias:

A la UPC y especialmente a todo el departamento de EIO, donde culmino mis estudios de doctorado, por darme la oportunidad de ejercer mi vocación: el ser partícipe en el dinámico proceso de Enseñanza-Aprendizaje.

A mis colegas del ETSEIB por hacerme sentir una más del grupo. Especial gratitud a Xavier Tort y a Pere Grima. También gracias a todos los colegas amigos que en algún momento u otro me han expresado su apoyo más allá del ámbito profesional.

Gracias a Toni y a Carme por su paciencia, amabilidad y ayuda eficiente en todo momento. Al Dr. Martí-Recober y a tí, Celia: no se me olvidará nunca la calidez con que me acogisteis a mi llegada al Departamento.

A Xavi Puig, por la lectura y sugerencias valiosas respecto a los primeros capítulos de esta tesis.

A Lluis Marco, colega y amigo, por su apoyo amable y oportuno.

To the external referees whose comments have been very helpful to improve the quality of this PhD thesis.

A mis compañeras de doctorado y también amigas latinoamericanas Nancy, Jeaneth y Alba. Atesoro los momentos compartidos y departidos.

En general, a todos los amigos y compañeros que de una manera u otra nos han ayudado a lo largo de estos años, sobre todo en esos tiempos difíciles cuando la tesis estaba en un muy relegado último plano. Gracias a vosotros, en parte, he podido retomar la senda doctoral.

A María Eugenia, Danelia, Laura, Reyna, Alicia, Ayla y familia, Helen, Eli, Mari, Daysi, Pilar, Dani, Victor y Eliseo: Gracias por mantener los lazos y la cercanía entrañable.

To my “adoptive” parents Jack Bristol and Lillian Mayberry for their invaluable contribution to my higher education; from the bottom of my heart, receive a special thanks.

Ganz herzlichen Dank auch an meine liebe Familie in Deutschland für all die Zuneigung sowie die Unterstützung in all diesen Jahren. Ich bin euch sehr sehr dankbar.

Dear Janice: Thanks a lot for your thoughtful kindness in proofreading the writing of this thesis. Any last minute changes that may lead to mistakes are my full responsibility.

A mi madre Bertha por su sencillez pero extremada sabiduría para conducirse por la vida; su amor y apoyo mantuvieron a flote la motivación para obtener este logro profesional y personal. A mis hermanos Julio, Manuel, Lourdes y Vanessa; ustedes y mis 10 sobrin@s son mi mayor tesoro hondureño. Esta tesis es también un brindis a ustedes y a la memoria de mi padre† y de mi hermano pequeño Luis Gustavo† (Dec/2005).

Y a tí Klaus, agradecerte tu inconmensurable apoyo y por caminar junto a mí en todo momento. Es fällt mir schwer, meine Dankbarkeit in Worte zu fassen. Innigen Dank!

ABSTRACT

The sequential estimation of the states (filtering) and the corresponding simultaneous estimation of the states and fixed parameters of a dynamic state-space model, being linear or not, is an important problem in many fields of research, such as in the area of finance.

The main objective of this research is to estimate sequentially and efficiently –from a Bayesian perspective via the particle filtering methodology– the states and/or the fixed parameters of a non-standard dynamic state-space model: one that is possibly nonlinear, non-stationary or non-Gaussian.

The present thesis consists of seven chapters and is structured into two parts. Chapter 1 introduces basic concepts, the motivation, the purpose, and the outline of the thesis.

Chapters 2-4, the first part of the thesis, focus on the estimation of the states. Chapter 2 provides a comprehensive review of the most classic algorithms (non-simulation based: KF, EKF, and UKF; and simulation based: SIS, SIR, ASIR, EPF, and UPF[1]) used for filtering solely the states of a dynamic state-space model. All these filters scattered in the literature are not only described in detail, but also placed in a unified notation for the sake of consistency, readability and comparability.

Chapters 3 and 4 confirm the efficiency of the well-established particle filtering methodology, via extensive Monte Carlo (MC) studies, when estimating only the latent states for a dynamic state-space model, being linear or not. Also, complementary MC studies are conducted to analyze some relevant issues within the adopted approach, such as the degeneracy problem, the resampling strategy, or the possible impact on estimation of the number of particles used and the time series length.

Chapter 3 specifically illustrates the performance of the particle filtering methodology in a linear and Gaussian context, using the exact Kalman filter as a benchmark. The performance of the four studied particle filter variants (SIR, SIRopt, ASIR, KPF, the latter being a special case of the EPF algorithm) is assessed using two apparently simple, but important time series processes: the so-called Local Level Model (LLM) and the AR(1) plus noise model, which are non-stationary and stationary, respectively. An exhaustive study on the effect of the signal-to-noise ratio (SNR) over the quality of the estimation is additionally performed. Complementary MC studies are conducted to assess the degree of degeneracy and the possible effect of increasing the number of particles and the time series length.

Chapter 4 assesses and illustrates the performance of the particle filtering methodology in a non-linear context. Specifically, a synthetic nonlinear, non Gaussian and non-stationary state space model

[1] See thet list of acronyms on page xxix

taken from literature is used to illustrate the performance of the four competing particle filters under study (SIR, ASIR, EPF, UPF) in contraposition to two well-known non-simulation based filters (EKF, UKF). In this chapter, the residual and stratified resampling schemes are compared and the effect of increasing the number of particles is addressed.

In the second part (Chapters 5 and 6), extensive MC studies are carried out, but the main goal is the simultaneous estimation of states and fixed model parameters for chosen non-standard dynamic models. This area of research is still very active and it is within this area where this thesis contributes the most.

Chapter 5 provides a partial survey of particle filter variants used to conduct the simultaneous estimation of states and fixed parameters. Such filters are an extension of those previously adopted for estimating solely the states. Additionally, a MC study is carried out to estimate the state (level) and the two fixed variance parameters of the non-stationary local level model; we use four particle filter variants (LW, SIRJ, SIRoptJ, KPFJ), six typical settings of the SNR and two settings for the discount factor needed in the jittering step. In this chapter, the SIRJ particle filter variant is proposed as an alternative to the well-established filter of Liu West (LW PF). The combined use of a Kalman-based proposal distribution and a jittering step is proposed and explored, which gives rise to the particle filter variant called: the Kalman Particle Filter plus Jittering (KPFJ).

Chapter 6 focuses on estimating the states and three fixed parameters of the non-standard basic stochastic volatility model known as stochastic autoregressive volatility model of order one: SARV(1). After an introduction and detailed description of the stylized features of financial time series, the estimation ability of two competing particle filter variants (SIRJ vs LW (Liu and West)) is shown empirically using simulated data. The chapter ends with an application to real data sets from the financial area: the Spanish IBEX 35 returns index and the Europe Brent Spot prices (in dollars).

The contribution in chapters 5 and 6 is to propose new variants of particle filters, such as the KPFJ, the SIRJ, and the SIRoptJ (a special case of the SIRJ that uses an optimal proposal distribution) that have developed along this work. The thesis also suggests that the so-called EPFJ (Extended Particle Filter with Jittering) and the UPFJ (Unscented Particle Filter with Jittering) algorithms could be reasonable choices when dealing with highly nonlinear models. In this part, also relevant issues within the particle filtering methodology are discussed, such as the potential impact on estimation of the discount factor parameter, the time series length, and the number of particles used.

Throughout this work, pseudo-codes are written for all filters studied and are implemented in R-Language. The reported findings are obtained as the result of extensive MC studies, considering a variety of case-scenarios described in the thesis. The intrinsic characteristics of the model at hand guided -according to suitability– the choice of filters in each specific situation. The comparison of filters is based on the RMSE, the elapsed CPU-time and the degree of degeneracy.

Finally, Chapter 7 includes the discussion, contributions, and future lines of research. Some complementary theoretical and practical aspects are presented in the appendixes.

RESUM

L'estimació seqüencial dels estats (filtratge) i la corresponent estimació simultània dels estats i els paràmetres fixos d'un model dinàmic formulat en forma d'espai d'estat –sigui lineal o no– constitueix un problema de rellevada importància en molts camps, com ser a l'àrea de finances.

L'objectiu principal d'aquesta tesi és el d'estimar seqüencialment i de manera eficient –des d'un punt de vista bayesià i usant la metodologia de filtratge de partícules– els estats i/o els paràmetres fixos d'un model d'espai d'estat dinàmic no estàndard: possiblement no lineal, no gaussià o no estacionari.

El present treball consisteix de 7 capítols i s'organitza en dues parts. El Capítol 1 hi introdueix conceptes bàsics, la motivació, el propòsit i l'estructura de la tesi.

La primera part d'aquesta tesi (capítols 2 a 4) se centra únicament en l'estimació dels estats. El Capítol 2 presenta una revisió exhaustiva dels algorismes més clàssics no basats en simulacions (KF, EKF, UKF[2]) i els basats en simulacions (SIS, SIR, ASIR, EPF, UPF). Per a aquests filtres, tots esmentats en la literatura, a més de descriure'ls detalladament, s'ha unificat la notació amb l'objectiu que aquesta sigui consistent i comparable entre els diferents algorismes implementats al llarg d'aquest treball.

Els capítols 3 i 4 se centren en la realització d'estudis Monte Carlo (MC) extensos que confirmen l'eficiència de la metodologia de filtratge de partícules per estimar els estats latents d'un procés dinàmic formulat en forma d'espai d'estat, sigui lineal o no. Alguns estudis MC complementaris es duen a terme per avaluar diferents aspectes de la metodologia de filtratge de partícules, com ser el problema de la degeneració, l'elecció de l'estratègia de remostreig, el nombre de partícules usades o la grandària de la sèrie temporal.

Específicament, el Capítol 3 il·lustra el comportament de la metodologia de filtratge de partícules en un context lineal i gaussià en comparació de l'òptim i exacte filtre de Kalman. La capacitat de filtratge de les quatre variants de filtre de partícules estudiades (SIR, SIRopt, ASIR, KPF; l'últim sent un cas especial de l'algorisme EPF) es va avaluar sobre la base de dos processos de sèries temporals aparentment simples però importants: els anomenats *Local Level Model* (LLM) i el *AR (1) plus noise*, que són no estacionari i estacionari, respectivament. Aquest capítol estudia en profunditat temes rellevants dins de l'enfocament adoptat, com l'impacte en l'estimació de la relació entre el senyal i el soroll (SNR: *signal-to-noise-ratio*, en aquesta tesi), de la longitud de la sèrie temporal i del nombre de partícules.

El Capítol 4 avalua i il·lustra el comportament de la metodologia de filtratge de partícules en un context no lineal. En concret, s'utilitza un model d'espai d'estat no lineal, no gaussià i no estacionari

[2] Veure llista d'acrònims a la pàgina xxix

pres de la literatura per il·lustrar el comportament de quatre filtres de partícules (SIR, ASIR, EPF, UPF) en contraposició a dos filtres no basats en simulació ben coneguts (EKF, UKF). Aquí es comparen els esquemes de remostreig residual i estratificat i s'avalua l'efecte d'augmentar el nombre de partícules.

A la segona part (capítols 5 i 6), es duen a terme també estudis MC extensos, però ara l'objectiu principal és l'estimació simultània dels estats i paràmetres fixos de certs models seleccionats. Aquesta àrea de recerca segueix sent molt activa i és on aquesta tesi hi contribueix més.

El Capítol 5 proveeix una revisió parcial dels mètodes per dur a terme l'estimació simultània dels estats i paràmetres fixos a través de la metodologia de filtratge de partícules. Aquests filtres són una extensió d'aquells adoptats anteriorment només per estimar els estats. Aquí es realitza un estudi MC per estimar l'estat (nivell) i els dos paràmetres de variància del model LLM no estacionari; s'utilitzen quatre variants (LW, SIRJ, SIRoptJ, KPFJ) de filtre de partícules, sis escenaris típics del SNR i dos escenaris per a l'anomenat factor de descompte necessari en el pas de diversificació. En aquest capítol, es proposa la variant de filtre de partícules SIRJ (*Sample Importance Resampling with Jittering*) com a alternativa al filtre de referència de Liu i West (LW PF). També es proposa i explora l'ús combinat d'una distribució d'importància basada en el filtre de Kalman i un pas de diversificació (*jittering*) que dóna lloc a la variant del filtre de partícules anomenada *Kalman Particle Filtering with Jittering* (KPFJ).

El Capítol 6 se centra en l'estimació dels estats i dels paràmetres fixos del model bàsic no estàndard de volatilitat estocàstica denominat *Stochastic autoregressive model of order one*: SARV (1). Després d'una introducció i descripció detallada de les característiques pròpies de sèries temporals financeres, es demostra mitjançant estudis MC la capacitat d'estimació de dues variants de filtre de partícules (SIRJ vs. LW (Liu i West)) utilitzant dades simulades. El capítol acaba amb una aplicació a dos conjunts de dades reals dins de l'àrea financera: l'índex de rendiments espanyol IBEX 35 i els preus al comptat (en dòlars) del Brent europeu.

La contribució en els capítols 5 i 6 consisteix en proposar noves variants de filtres de partícules, com poden ser el KPFJ, el SIRJ i el SIRoptJ (un cas especial de l'algorisme SIRJ utilitzant una distribució d'importància òptima) que s'han desenvolupat al llarg d'aquest treball. També se suggereix que els anomenats filtres de partícules EPFJ (*Extended Particle Filter with Jittering*) i UPFJ (*Unscented Particle Filter with Jittering*) podrien ser opcions raonables quan es tracta de models altament no lineals; el KPFJ sent un cas especial de l'algorisme EPFJ. En aquesta part, també es tracten aspectes rellevants dins de la metodologia de filtratge de partícules, com ser l'impacte potencial en l'estimació de la longitud de la sèrie temporal, el paràmetre de factor de descompte i el nombre de partícules.

Al llarg d'aquest treball s'han escrit (i implementat en el llenguatge R) els pseudo-codis per a tots els filtres estudiats. Els resultats presentats s'obtenen mitjançant simulacions Monte Carlo (MC) extenses, tenint en compte variats escenaris descrits en la tesi. Les característiques intrínseques del model baix estudi van guiar l'elecció dels filtres a comparar en cada situació específica. A més, la comparació dels filtres es basa en el RMSE (*Root Mean Square Error*), el temps de CPU i el grau de degeneració.

Finalment, el Capítol 7 presenta la discussió, les contribucions i les línies futures de recerca. Alguns aspectes teòrics i pràctics complementaris es presenten en els apèndixs.

RESUMEN

La estimación secuencial de los estados (filtrado) y la correspondiente estimación simultánea de los estados y los parámetros fijos de un modelo dinámico formulado en forma de espacio de estado –sea lineal o no– constituye un problema de relevada importancia en muchos campos, como ser en el área de finanzas.

El objetivo principal de esta tesis es el de estimar secuencialmente y de manera eficiente –desde un punto de vista bayesiano y usando la metodología de filtrado de partículas– los estados y/o los parámetros fijos de un modelo de espacio de estado dinámico no estándar: posiblemente no lineal, no gaussiano o no estacionario.

El presente trabajo consta de 7 capítulos y se organiza en dos partes. El Capítulo 1 introduce conceptos básicos, la motivación, el propósito y la estructura de la tesis.

La primera parte de esta tesis (capítulos 2 a 4) se centra únicamente en la estimación de los estados. El Capítulo 2 presenta una revisión exhaustiva de los algoritmos más clásicos no basados en simulaciones (KF, EKF, UKF[3]) y los basados en simulaciones (SIS, SIR, ASIR, EPF, UPF). Para todos estos filtros, mencionados en la literatura, además de describirlos en detalle, se ha unificado la notación con el objetivo de que ésta sea consistente y comparable entre los diferentes algoritmos implementados a lo largo de este trabajo.

Los capítulos 3 y 4 se centran en la realización de estudios Monte Carlo (MC) extensos que confirman la eficiencia de la metodología de filtrado de partículas para estimar los estados latentes de un proceso dinámico formulado en forma de espacio de estado, sea lineal o no. Algunos estudios MC complementarios se llevan a cabo para evaluar varios aspectos de la metodología de filtrado de partículas, como ser el problema de la degeneración, la elección de la estrategia de remuestreo, el número de partículas usadas o el tamaño de la serie temporal.

Específicamente, el Capítulo 3 ilustra el comportamiento de la metodología de filtrado de partículas en un contexto lineal y gaussiano en comparación con el óptimo y exacto filtro de Kalman. La capacidad de filtrado de las cuatro variantes de filtro de partículas estudiadas (SIR, SIRopt, ASIR, KPF; el último siendo un caso especial del algoritmo EPF) se evaluó en base a dos procesos de series temporales aparentemente simples pero importantes: los denominados *Local Level Model* (LLM) y el *AR (1) plus noise*, que son no estacionario y estacionario, respectivamente. Este capítulo estudia en profundidad temas relevantes dentro del enfoque adoptado, como el impacto en la estimación de la relación

[3] Ver lista de acrónimos en la página xxix

entre la señal y el ruido (SNR: *signal-to-noise-ratio*, en esta tesis), de la longitud de la serie temporal y del número de partículas.

El Capítulo 4 evalúa e ilustra el comportamiento de la metodología de filtrado de partículas en un contexto no lineal. En concreto, se utiliza un modelo de espacio de estado no lineal, no gaussiano y no estacionario tomado de la literatura para ilustrar el comportamiento de cuatro filtros de partículas (SIR, ASIR, EPF, UPF) en contraposición a dos filtros no basados en simulación bien conocidos (EKF, UKF). Aquí se comparan los esquemas de remuestreo residual y estratificado y se evalúa el efecto de aumentar el número de partículas.

En la segunda parte (capítulos 5 y 6), se llevan a cabo también estudios MC extensos, pero ahora el objetivo principal es la estimación simultánea de los estados y parámetros fijos de ciertos modelos seleccionados. Esta área de investigación sigue siendo muy activa y es donde esta tesis contribuye más.

El Capítulo 5 provee una revisión parcial de los métodos para llevar a cabo la estimación simultánea de los estados y parámetros fijos a través de la metodología de filtrado de partículas. Dichos filtros son una extensión de aquellos adoptados anteriormente sólo para estimar los estados. Aquí se realiza un estudio MC para estimar el estado (nivel) y los dos parámetros de varianza del modelo LLM no estacionario; se utilizan cuatro variantes (LW, SIRJ, SIRoptJ, KPFJ) de filtro de partículas, seis escenarios típicos del SNR y dos escenarios para el llamado factor de descuento necesario en el paso de diversificación. En este capítulo, se propone la variante de filtro de partículas SIRJ (Sample Importance resampling with Jittering) como alternativa al filtro de referencia de Liu y West (LW PF). También se propone y explora el uso combinado de una distribución de importancia basada en el filtro de Kalman y un paso de diversificación (jittering) que da lugar a la variante del filtro de partículas denominada *Kalman Particle Filtering with Jittering* (KPFJ).

El Capítulo 6 se centra en la estimación de los estados y de los parámetros fijos del modelo básico no estándar de volatilidad estocástica denominado Stochastic autoregressive model of order one: SARV (1). Después de una introducción y descripción detallada de las características propias de series temporales financieras, se demuestra mediante estudios MC la capacidad de estimación de dos variantes de filtro de partículas (SIRJ vs. LW (Liu y West)) utilizando datos simulados. El capítulo termina con una aplicación a dos conjuntos de datos reales dentro del área financiera: el índice de rendimientos español IBEX 35 y los precios al contado (en dólares) del Brent europeo.

La contribución en los capítulos 5 y 6 consiste en proponer nuevas variantes de filtros de partículas, como pueden ser el KPFJ, el SIRJ y el SIRoptJ (Caso especial del algoritmo SIRJ utilizando una distribución de importancia óptima) que se han desarrollado a lo largo de este trabajo. También se sugiere que los llamados filtros de partículas EPFJ (*Extended Particle Filter with Jittering*) y UPFJ (*Unscented Particle Filter with Jittering*) podrían ser opciones razonables cuando se trata de modelos altamente no lineales; el KPFJ siendo un caso especial del algoritmo EPFJ. En esta parte, también se tratan aspectos relevantes dentro de la metodología de filtrado de partículas, como ser el impacto potencial en la estimación de la longitud de la serie temporal, el parámetro de factor de descuento y el número de

partículas.

A lo largo de este trabajo se han escrito (e implementado en el lenguaje R) los pseudo-códigos para todos los filtros estudiados. Los resultados presentados se obtienen mediante simulaciones Monte Carlo (MC) extensas, teniendo en cuenta variados escenarios descritos en la tesis. Las características intrínsecas del modelo bajo estudio guiaron la elección de los filtros a comparar en cada situación específica. Además, la comparación de los filtros se basa en el RMSE (*Root Mean Square Error*), el tiempo de CPU y el grado de degeneración.

Finalmente, el Capítulo 7 presenta la discusión, las contribuciones y las líneas futuras de investigación. Algunos aspectos teóricos y prácticos complementarios se presentan en los apéndices.

CONTENTS

List of Tables

LIST OF FIGURES

List of Algorithms

Acronyms

ASIR	Auxiliary Sampling Importance Resampling
EKF	Extended Kalman Filter
EPF	Extended Particle Filter
EPFJ	Extended Particle Filter with Jittering
IS	Importance Sampling
KF	Kalman Filter
KPF	Kalman Particle Filter
KPFJ	Kalman Particle Filter with Jittering
LW	Liu and West Particle Filter
NIF	Numerical Integration Filter
PDF	Probability Density Function
PF	Particle Filter
SIS	Sequential Importance Sampling
SISR	Sampling Importance Resampling (Bootstrap filter)
SIR	Sampling Importance Resampling
SIRJ	Sampling Importance Resampling with Jittering
SMC	Sequential Monte Carlo
SNR	Signal-to-noise-ratio
SUT	Scaled Unscented Transformation
UKF	Unscented Kalman Filter

UPF Unscented Particle Filter

UPFJ Unscented Particle Filter with Jittering

UT Unscented Transformation

CHAPTER 1

INTRODUCTION

1.1 Motivation and Purpose

Sequential state–estimation (filtering) and the simultaneous estimation of states and parameters of nonlinear dynamic models is an important issue in many fields, such us in target tracking, modeling, finance applications and so on. Within the finance literature, the most important class of nonlinear models is the family of conditional heteroscedastic models including, among others, the stochastic volatility model. Stochastic volatility models are quite popular in finance applications, especially in describing series with sudden changes in the magnitude of variation of the observed data (Taylor 1986).

The need to develop optimal estimation algorithms for non-standard[1] time series models, formulated as state-space models, has been the motivating starting point of our research. For instance, in the last two decades there has been a greater interest to develop and apply adequate methods to estimate the underlying volatility of stochastic volatility models and other involved model parameters. Thus, estimation issues regarding volatility series, such as exchange rates or return prices, are a must for finance analysts and practitioners (Taylor 1994).

Most common statistical packages are not suitable to handle non-standard models, at least not directly. For instance, it is known that the Box-Jenkins methodology provides solutions for the well-known autoregressive integrated and moving average (ARIMA) time series; which assume Gaussian error terms. Additionally, it is assumed that stationarity must be fulfilled before modeling the series. For a review on time series analysis and its modeling via the Box-Jenkins methodology, see Box, Jenkins, and Reinsel (1994), Brockwell and Davis (1996), and Shumway and Stoffer (2006).

[1] By non-standard we mean series that are specified by dynamic models that exhibit either a nonlinear, non Gaussian, or a non-stationary behavior.

According to our knowledge, the use of state-space models for the analysis of time series began with the papers of Kalman (1960) and Akaike (1974), in the area of control engineering and the analysis of ARMA processes, respectively. Durbin and Koopman (2001) also consider time series analysis based on a state-space representation, a formulation that results to be very flexible. Indeed, this formulation allows a broader range of models to be cast in state-space form. For instance, all ARIMA time series may be represented in state-space form. Specifically, it can be shown that an ARIMA(1,0,1) model is equivalent to the state-space formulation of an AR(1) plus measurement noise model. This type of relationship can be seen as an advantage when trying to estimate, for example, the parameters for (possibly) non-Gaussian ARIMA models. Moreover, under this framework, non-standard time series can be "more naturally" modeled for a later estimation of the underlying states and involved parameters. In this thesis we adopt the state-space framework.

Before getting any further, we consider it appropriate to introduce some commonly used notation. Let $\boldsymbol{y}_{1:s} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_s\}$ denote the sequentially observed data up to time s. Likewise, the set of unobserved signals or latent state vectors to be estimated is denoted by $\boldsymbol{x}_{1:s} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_s\}$. Additionally, let $P\left(\boldsymbol{x}_t|\boldsymbol{y}_{1:s}\right)$ be the conditional density of the state vector $\boldsymbol{x}_t$ given the observations $\boldsymbol{y}_{1:s}$ and define $\boldsymbol{y}_{1:T}$ as the complete observed time series. Thus, depending on the values of t and s in $P\left(\boldsymbol{x}_t|\boldsymbol{y}_{1:s}\right)$ we have the following stages:

- **Prediction** is given if $s < t$; we speak of one step prediction when $s = t - 1$.
- **Filtering** is given, in the case that $s = t$, $t \leqslant T$.
- **Smoothing** is given, when $s > T$, where $t < T$.

As previously mentioned, this work focuses mainly on non-standard state-space models. It is well known that in the specific situation of filtering Gaussian linear models, the recursive Kalman Filter provides an optimal and closed form solution, which can be explicitly derived (Kalman 1960; Kalman and Bucy 1961). In practice, however, one mostly deals with data that is specified by non-standard dynamic state-space models.

To deal with dynamic filtering non-standard models, several methods have been developed. The most-used approaches are based on Taylor series expansions (Anderson and Moore 1979). Other methods are derived based on the underlying density functions, see e.g. Kitagawa (1987), Tanizaki and Mariano (1998), and Acosta, Martí-Recober, and Muñoz (2003). More traditional approaches based on maximum likelihood or grid-based methods are also an alternative (Shumway 1988; Shumway and Stoffer 2006; Muñoz 1988).

Simulation-based approaches, such as the Markov chain Monte Carlo (MCMC) type filters and the sequential Monte Carlo (SMC) type filters, can also be used to deal with filtering these non-standard state-space models. Some authors consider the estimation of general state-space models via MCMC-based filters; see for instance Carlin, Polson, and Stoffer (1992), Frühwirth-Schnatter (1994), Shephard and Pitt (1995), and Chib, Nardari, and Shephard (2002). Further, Lopes and Tsay (2011) state that

the MCMC-based filters can be prohibitively costly when dealing with the sequential estimation of states and model parameters. More recently, a particle filter variant named the Particle Markov chain Monte Carlo (PMCMC) approach has been introduced by Andrieu, Doucet, and Holenstein (2010). These authors combine two approaches, powerful by themselves: the SMC and the MCMC methods, whereby the former is used to construct an efficient (high dimensional) proposal distribution which is used by the latter.

In this thesis, we restrict ourselves to study, in deep, some of the classic sequential Monte Carlo methods, which are known as particle filters. The possibly nonlinear state-space models that are dealt with consist of a univariate latent state vector $\boldsymbol{x}_t$ and of a (possibly unknown) multivariate vector Θ (order up to three) of fixed model parameters.

The simulation-based approach known as particle filtering is a sequential Monte Carlo methodology suitable to deal with possibly non-Gaussian or nonlinear problems. A particle filter (PF) is a very flexible SMC method that departs from a state-space formulation and allows an implementation of a recursive Bayesian filter through Monte Carlo simulations (Kitagawa 1998; Doucet et al. 2001; Arulampalam et al. 2002; Doucet and Johansen 2011). Under this methodology, once a process is cast in state-space form, one is faced with the so-called optimal filtering problem, which consists of estimating the conditional probability density function $P(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$, of the latent variable/vector $\boldsymbol{x}_t$ given the observations $\boldsymbol{y}_{1:t}$. Hence, once this probability density function (PDF) is exactly or approximately known, the optimal filtering problem is said to be solved and thus any characteristic of the states can be easily obtained; for example the mean, root mean square error, credible intervals, median, kurtosis, and so on.

Our main goal is to be able to sequentially estimate together the states and fixed parameters (apart of only the states) of "non-standard" dynamic models and we aim to perform such estimation from a Bayesian point of view. To achieve our goal, we adopt the very suitable methodology called particle filtering. Have in mind that the used statistical and computational criteria to assess the performance of the competing filters are the mean CPU time and root mean square error (RMSE), unless stated otherwise.

Additionally, since we are aware of the inherent degeneracy drawback that can potentially affect particle filters, we construct a measure to somehow quantify the degree of degeneracy present in the particle filters studied. We believe that it is not only important to find the so-called effective sample size (ESS) that tells the percentage of 'surviving' particles at a specific point of time, but also to quantify how distinct the 'surviving' particles are. We provide a measure, denoted as %uNp, determining the mean (in percentage) number of unique particles obtained at a specific point of time; usually we provide this measure at the end of the time trajectory. This measure will account for both types of degeneracy that one may encounter: the collapse of particles to few ones (or even to a single particle), and the collapse to few and non unique particles; the latter being a more acute problem in case of the simultaneous estimation of states and fixed model parameters. We consider that this idea of quantifying the degree of degeneracy is justified per se, but it also goes in line with ideas stated recently in Andrieu, Doucet,

and Holenstein (2010).

This work is structured into two parts. Part I focuses on filtering only the states of chosen linear and nonlinear dynamic models, whereas Part II deals with estimating simultaneously the states and fixed parameters of some selected non-standard dynamic models. Throughout this work, pseudo-codes are written (and implemented in R-Language) for all filters studied. The comparison of filters is based on the RMSE, the elapsed CPU-time and the degree of degeneracy. The reported findings are obtained as the result of extensive MC studies, considering a variety of case-scenarios described in the thesis. The intrinsic characteristics of the model at hand guided –according to suitability– the choice of filters in each specific situation. Following, we provide an outline of the next chapters of this thesis.

1.2 Outline of the Thesis

Chapters 2–4, the first part of the thesis, deal with states estimation only. Chapter 2 reviews some traditional and some sequential Monte Carlo Bayesian approaches for filtering (mainly) non-standard dynamic models. Sections 2.1 and 2.2 present general concepts such as the state-space formulation for dynamic models and the respective general prediction and filtering probability density function expressions. A fairly detailed description of the most useful traditional approaches to solve the nonlinear and possible non-Gaussian filtering problem is given in Section 2.3. Therein, we begin by describing the analytical and linear Kalman filter and then the nonlinear extended Kalman filter, which is based on Taylor series expansions. Next, the main features of the so-called unscented Kalman filter, which is also nonlinear, are provided. Finally, in Section 2.4, a rather complete survey of the classic sequential Monte Carlo methods, named Particle filters is provided. Several aspects of the particle filtering methodology are thoroughly explained therein. Throughout the chapter, all filters studied are not only fully described, but also corresponding pseudo-codes are provided.

Chapter 3 illustrates the performance of the particle filtering methodology in a linear and Gaussian context. We are aware that in this case the Kalman filter provides not only a closed form analytical solution, but the best possible solution; its optimality based on minimizing the root mean square error. We take advantage of this fact to assess the ability of the studied particle filters for filtering two apparently simple but important (as they are commonly used in theoretical and applied work) time series processes: the so-called local level model and the first order autoregressive AR(1) plus noise model. Thus, the optimal Kalman filter is taken here as a gold standard benchmark for the other filters entertained. In this chapter, we make an exhaustive study of the impact of the signal-to-noise-ratio value on the quality of the estimation in order to provide some useful guidelines for practitioners interested in these type of models. Additionally, for the competing particle filters in question, key issues within the adopted approach are addressed, such as the influence on estimation of the increase of the number of particles, the length of the time series length, or the degree of degeneracy.

In Chapter 4, we illustrate the filtering performance of particle filters in a nonlinear context. Specifically, a synthetic nonlinear, non-stationary and non Gaussian dynamic state-space model taken from the literature is chosen to illustrate the performance of the four particle filter variants under study

in contraposition to two traditional non-simulation based approaches: the nonlinear EKF and UKF filters. Additionally, for the competing particle filters in question, key aspects like assessing the effect of the increase of the number of particles or the choice of a resampling strategy are addressed.

In the second part (Chapters 5 and 6), extensive MC studies are also carried out, but the main goal is the simultaneous estimation of states and fixed model parameters for some chosen non-standard dynamic models. This area of research is still very active and it is in this area where this thesis contributes the most. Chapter 5 provides a partial survey of methods for conducting the simultaneous estimation of states and fixed parameters via particle filtering. Such filters, which are described in detail, arrive as an extension of those previously adopted for estimating solely the states. Specific aspects, such as how to avoid the collapse of the particles, are also fully described. In this chapter, one can find fully documented pseudo-codes for all described algorithms, including those proposed by us. As new particle filter variants we propose the KPFJ (Kalman particle filter with jittering), the SIRJ (sample importance resampling particle filter with jittering), and the SIRoptJ (a special case of the SIRJ that uses and optimal proposal distribution). We also suggest that the so-called EPFJ (extended particle filter with jittering) and the UPFJ (unscented particle filter with jittering) algorithms could be reasonable choices when dealing with highly nonlinear models; these filters combine a Kalman based proposal PDF with a jittering step. Additionally, apart from a partial study of the impact of the signal-to-noise-ratio on the quality of the estimations, relevant issues within the particle filtering methodology are also addresed, such as the potential impact of the chosen discount factor parameter, the number of particles used in the estimation procedure, or the time series length.

Chapter 6 focuses on estimating the states and parameters of the basic nonlinear stochastic volatility model known as stochastic autoregressive volatility model of order one: SARV (1). Therein, the stylized features of a financial time series are described, the two most common stochastic volatility models are briefly introduced, the corresponding state formulation of the SARV(1) model is specified, and also two Monte Carlo studies are conducted: one for estimating only the states (volatility of volatility) and another for estimating together the states and the parameters involved in the model. This chapter ends up with an application to two data sets containing volatile data. The aim therein is to illustrate the estimation ability of the two competing particle filter variants (SIRJ vs LW (Liu and West)) using real data sets from the financial area: the Spanish IBEX 35 returns index and the Europe Brent Spot prices (in dollars).

Finally, Chapter 7 presents the discussion and future lines of research. Some complementary theoretical and practical aspects are contained in the appendixes. We remark that the contents of Appendix B can be found in `http://www-eio.upc.edu/~lacosta/AppendixB.pdf` [last visited: September 2013]. This document contains complementary graphical displays corresponding to the Monte Carlo experiments carried out along the thesis using two linear models: the non-stationary local level model and the stationary autoregressive plus noise model of order one (AR(1) plus noise model). In the sequel, all figures with name starting with B are contained in the mentioned website.

Part I

Filtering

CHAPTER 2

STATE-SPACE FORMULATION FOR DYNAMIC MODELS AND STATE ESTIMATION WITH KNOWN PARAMETERS

Sequential–state–estimation or filtering is an interesting problem that researchers are often faced with. In tracking problems, one could be interested in the kinematic characteristics of a target, for instance, in tracking the position and velocity of a ship; see for example Pitt and Shephard (1999). Alternatively, in finance problems, an important point of interest is the estimation of the underlying volatility of return time series; see, among others, Márquez (2002), Acosta, Martí-Recober, and Muñoz (2004), and Muñoz, Márquez, and Acosta (2007). Inflation time series is also a topic of interest as can be seen, for example, in Stock and Watson (2007), Pellegrini (2009) and Rodriguez (2010).

In practice, nonlinear dynamic problems are the rule rather than the exception. Thus, to deal with dynamic filtering non-standard models, several methods have been developed. The most heuristic and appealing approaches, are based on Taylor series expansions; see for example Anderson and Moore (1979). Other methods are derived based on the underlying density functions, (Kitagawa 1987; Muñoz, Egozcue, and Martí-Recober 1988; Tanizaki 2001). Moreover, in the last two decades, there is a renewed interest on sequential Monte Carlo (SMC) methods to perform Bayesian filtering, see among others Kitagawa (1996, 1998), Doucet (1998), Doucet, de Freitas, and Gordon (2001), Andrieu, Doucet, and Holenstein (2010) and Lopes and Tsay (2011). The sequential Monte Carlo methods known as particle filters, naturally suited for online estimation, are the focus of this work.

This chapter is organized as follows. Section 2.1 specifies the state-space formulation for a general dynamic model. Then in Section 2.2, the general prediction and filtering expressions are provided. Section 2.3 describes three traditional, non simulation based, approaches used for state estimation in dynamic models. Likewise, Section 2.4 describes four simulation based algorithms used for state estimation in dynamic models. In this chapter, indeed in chapters 2 to 4, all existing model parameters are assumed to be fixed and known.

2.1 State Space Formulation

Let $\boldsymbol{y}_t$ be the data sequentially observed at discrete time points $t = 1, \ldots, T$. The parametric state-space formulation for a general dynamic model can be described by the following two equations (see, for example, Shumway 1988; Tanizaki 1996):

$$\boldsymbol{x}_t = f(\boldsymbol{x}_{t-1}, \boldsymbol{\eta}_t), \qquad \textit{(Transition equation)} \tag{2.1}$$

$$\boldsymbol{y}_t = h(\boldsymbol{x}_t, \boldsymbol{\nu}_t), \qquad \textit{(Measurement equation)} \tag{2.2}$$

where $\boldsymbol{x}_t \in R^{n_x}$ is the unobserved state vector, $\boldsymbol{\eta}_t \in R^{n_\eta}$ is the process noise and $\boldsymbol{\nu}_t \in R^{n_\nu}$ is the measurement noise. The functional forms of $f \in R^{n_x} \times R^{n_\eta} \to R^{n_x}$ and $h \in R^{n_x} \times R^{n_\nu} \to R^{n_y}$ are assumed to be known, but not necessarily linear. Both, $\boldsymbol{\eta}_t \sim p_{\boldsymbol{\eta}}$ and $\boldsymbol{\nu}_t \sim p_{\boldsymbol{\nu}}$ are generally white noise processes, not necessarily Gaussian; $p_{\boldsymbol{\eta}}$ and $p_{\boldsymbol{\nu}}$ being the probability density function (PDF) of the state and measurement noise, respectively. To complete the state-space model specification, it is assumed that the PDF of the initial-state vector $\boldsymbol{x}_0$, $p(\boldsymbol{x}_0|\boldsymbol{y}_0) \equiv p_{\boldsymbol{x}_0}$, is available; where $\boldsymbol{y}_0$ is the set of no measurements. Note that the functional forms f and h, as well as the probability density functions $p_{\boldsymbol{\eta}}$ and p_ν, may depend on parameters, say $\boldsymbol{\Theta}$.

Sometimes, it is convenient to formulate the general state-space model based on conditional distributions (Kitagawa and Sato 2001). In such case, the transition and measurement equations are specified by:

$$\boldsymbol{x}_t|\boldsymbol{x}_{t-1} \sim p(\cdot|\boldsymbol{x}_{t-1}), \tag{2.3a}$$

$$\boldsymbol{y}_t|\boldsymbol{x}_t \sim p(\cdot|\boldsymbol{x}_t). \tag{2.3b}$$

Under the state-space formulation framework, two basic assumptions are made. First, the states $\boldsymbol{x}_t$ have a Markovian nature of order one. Second, the observations $\boldsymbol{y}_t$ are conditionally independent given the states $\boldsymbol{x}_t$. For an illustration of the state-space model specification, see Figure 2.1. Note that the state-space formulation cover a broad class of models with many practical applications; see for example West and Harrison (1989), Kim, Shephard, and Chib (1998) and Muñoz, Márquez, Martí-Recober, Villazón, and Acosta (2004).

In this work, we aim to obtain the marginal posterior probability density function, $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$, and not the joint posterior PDF $p(\boldsymbol{x}_{1:t}|\boldsymbol{y}_{1:t})$. Thus, we do not need to keep track of the complete state vector trajectory and hence less storage capacity is needed.

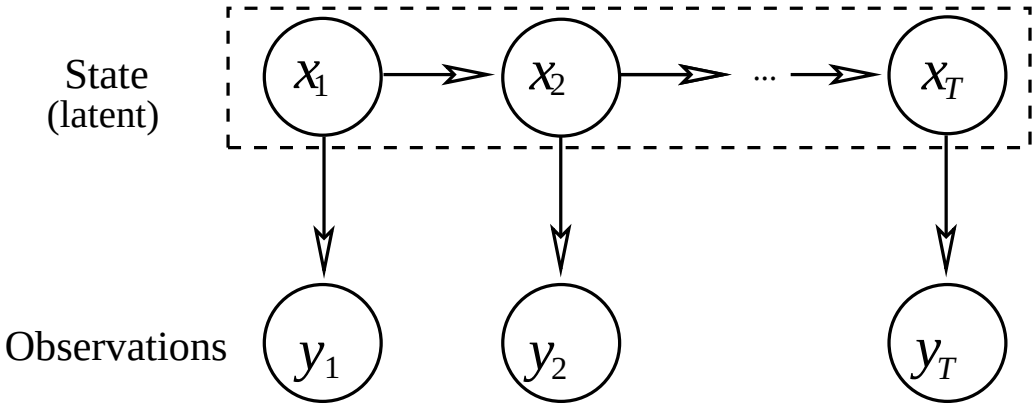


Figure 2.1: State-space model graphic illustration: At any time index the latent state $\boldsymbol{x}_t$ can only be accessed through the noisy observation $\boldsymbol{y}_t$ that is available. Two properties hold: the states $\boldsymbol{x}_t$ have a Markovian nature of order one and the observations $\boldsymbol{y}_t$ are conditionally independent of the states $\boldsymbol{x}_t$.

2.2 General Prediction and Filtering Expressions

Herein, the general prediction and filtering expressions are derived for the completely specified parametric state-space model in equations (2.1) and (2.2). These well-known expressions are obtained as a combined result of the basic assumptions of the state-space formulation and the use of the Bayes Rule, see e.g. Jazwinski (1970), Shumway (1988), and Tanizaki (1996).

Predictive PDF

At a time $t-1$, assume that the prior PDF $p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1})$ is available. Additionally, assume that the fixed parameter vector $\boldsymbol{\Theta}$ is known. Then, using the Markovian property of the states, the general expression for the one step-ahead prediction (time update) is given by:

$$\begin{aligned} p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1}) &= \int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{y}_{1:t-1})\, p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1})\, d\boldsymbol{x}_{t-1} \\ &= \int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})\, p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1})\, d\boldsymbol{x}_{t-1} \end{aligned} \tag{2.4}$$

where $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ is the state evolution density specified in equation (2.3a).

Filtering PDF

Using Bayes Rule, once a new observation $\boldsymbol{y}_t$ arrives, an update of the state vector can be obtained. This update is given by the filtering PDF $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$[1]. The general expression for this filtering (measurement update) PDF is derived as follows:

[1] Notice that the posterior PDF at time t becomes the prior for next time-index

$$
\begin{aligned}
p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) &= \frac{p(\boldsymbol{y}_{1:t}|\boldsymbol{x}_t)p(\boldsymbol{x}_t)}{p(\boldsymbol{y}_{1:t})} \\
&= \frac{p(\boldsymbol{y}_t, \boldsymbol{y}_{1:t-1}|\boldsymbol{x}_t)p(\boldsymbol{x}_t)}{p(\boldsymbol{y}_t, \boldsymbol{y}_{1:t-1})} \\
&= \frac{p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{x}_t)p(\boldsymbol{y}_{1:t-1}|\boldsymbol{x}_t)p(\boldsymbol{x}_t)}{p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1})p(\boldsymbol{y}_{1:t-1})} \\
&= \frac{p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})\cancel{p(\boldsymbol{y}_{1:t-1})p(\boldsymbol{x}_t)}}{p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1})\cancel{p(\boldsymbol{y}_{1:t-1})p(\boldsymbol{x}_t)}} \\
&= \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})}{\int p(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})\, d\boldsymbol{x}_t} \\
&\propto p(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1}).
\end{aligned} \tag{2.5}
$$

The denominator $p\left(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}\right) = \int p(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})\, d\boldsymbol{x}_t$ is the normalizing constant, which, except in few special cases, cannot be computed analytically. Additionally, $p(\boldsymbol{y}_t|\boldsymbol{x}_t)$ is the measurement evolution density (likelihood of $\boldsymbol{y}_t$) specified in equation (2.3b) and $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})$ is the predictive expression presented in equation (2.4).

Recursive Filtering PDF expression

Plugging (2.4) into (2.5), a recursive expression for the filtering PDF is obtained:

$$
p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) \propto p(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1}) = \underbrace{p(\boldsymbol{y}_t|\boldsymbol{x}_t)}_{\text{Likelihood dty.}} \int \underbrace{p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}_{\text{Transition dty.}} \underbrace{p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1})}_{\text{Filtering dty.}} d\boldsymbol{x}_{t-1} \tag{2.6}
$$

This recursive filtering PDF is defined in terms of the likelihood density (equation (2.2)), the state transition density (equation (2.1)), and the filtering density at the previous time step.

Since the state vector $\boldsymbol{x}_t$ embodies all the relevant information about the general dynamic state-space model specified in equations (2.1) and (2.2), once $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$ is known, the optimal filtering problem is said to be solved. Within the Bayesian framework, this is known as the Bayesian filtering problem. Solving the optimal filtering problem consists thus on recursively estimating the posterior PDF of the state $\boldsymbol{x}_t$ given the noisy observations $\boldsymbol{y}_{1:t}$. Then, any characteristic of the state, such as the mean, median, credible intervals, kurtosis, and so on, can be easily obtained.

Later on in this work, the criterion used to assess the statistical performance of competing filters is based on the mean square error (MSE); the lowest the MSE, the more eficient the filter is. In other words, the mean of the posterior PDF $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}$ given by $\boldsymbol{x}_{t|t} = \mathrm{E}(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) = \int p(\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$ is used as a Bayesian estimator, which is known to be optimal in terms of mean square error (MSE). In the context of linear and Gaussian dynamic models, the aforementioned PDFs are all Gaussian, which implies that the optimal Bayesian estimator is exact. For general dynamic models, however, a close form solution cannot (easily) be obtained as it may involve the evaluation of multiple integrals; see expressions in equations (2.4)–(2.6). This issue highlights the need for approximative approaches.

Practical applications of the optimal filtering problem include target tracking (Gordon, Salmond, and Smith 1993) and estimation of stochastic volatility (Chib, Nardari, and Shephard 2002; Muñoz, Márquez, and Acosta 2007). In the remainder of this chapter, for the sake of brevity, we refer to optimal filtering by just the word "filtering".

When coped with the filtering problem, one usually applies a specific type of methodological approach according to the class of models and data available. For instance, Bayesian methods are naturally suited for online filtering that sequentially updates the knowledge of the states as a new observation becomes available on time. Offline filtering, however, estimates the states given a batch of data. The data itself can be dynamic or static, depending on whether temporal dependence is present. Also, the data could be observed/studied at discrete or continuous time points. Our emphasis in this work is on discrete, dynamic, online, Bayesian and on non-standard – possibly non-stationary non-Gaussian nonlinear – state-space models.

In the rest of this chapter, we describe some methods – scattered in the literature – used to solve the nonlinear filtering problem. For completion, the linear and exact Kalman filter (KF) is included. Thus, next section gives an overview of some analytical and approximative approaches for filtering.

2.3 Dynamic State Estimation: Traditional Filtering Methodology

Herein, some implementation algorithms of the predictive and filtering equations in (2.4)–(2.6) are described. First, three well-known traditional approaches are presented; the Kalman filter, the extended Kalman filter, and the unscented Kalman filter. Then, the particle filtering methodology, which is based on simulations, is fully treated. Further, we write pseudo-codes for all filters described in the present chapter. In later chapters, these algorithms will be further studied and implemented in R language.

2.3.1 The Kalman Filter

Linear and Gaussian models (known also as Kalman models) have been extensively investigated by the engineering and control communities for decades. Traditionally, the emphasis has been on signal extraction; that is, on filtering problems. The well–known Kalman filter (KF) gives a benchmark recursive solution for the filtering problem in case of Gaussian and linear models with known model parameters (Kalman 1960; Kalman and Bucy 1961). This analytical filter provides an exact computation of the equations (2.4)–(2.6) provided that the functional forms f and h are linear, and that the probability density functions $\boldsymbol{\eta}_t$, $\boldsymbol{\nu}_t$ and x_0 are all Gaussian. Thus, in case of Kalman models, all the probability density functions ($p(\boldsymbol{x}_t|\boldsymbol{y}_{1:s})$, $s, t \in \{1,\ldots,T\}$) in equations (2.4)–(2.6) are Gaussian and, as a result, there exists a finite number of parameters that characterize these densities. This implies that, at every time step, the filtering PDF $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$ is Gaussian and hence parameterized by a mean and a covariance.

Intuitively, assuming that certain conditions hold, the KF solves the filtering problem by recursively estimating the mean and covariance which characterize the posterior Gaussian PDF $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$ of the

target state vector $\boldsymbol{x}_t$. Thus, in case of linear and Gaussian models, (2.1) and (2.2) can be rewritten as:

$$\boldsymbol{x}_t = F_t \boldsymbol{x}_{t-1} + \boldsymbol{\eta}_t, \qquad \textit{(Transition equation)} \tag{2.7}$$

$$\boldsymbol{y}_t = H_t \boldsymbol{x}_t + \boldsymbol{\nu}_t, \qquad \textit{(Measurement equation)} \tag{2.8}$$

where F_t and H_t are known matrices defining the linear functions. The known covariances of the state and measurement noise densities $\boldsymbol{\eta}_t$ and $\boldsymbol{\nu}_t$ are Q_t and R_t respectively. Notice that the system and measurement matrices F_t and H_t as well as noise parameters Q_t and R_t could be allowed to vary. In this work, unless stated otherwise, both $\boldsymbol{\eta}_t$ and $\boldsymbol{\nu}_t$ are assumed to have zero means.

Based on the expressions for predictive and filtering probability density functions in (2.4)-(2.5), the first and second moments $\boldsymbol{x}_{t|t}$ and $\boldsymbol{\Sigma}_{\boldsymbol{x}_{t|t}}$ of a Kalman model state-vector $\boldsymbol{x}_t$ are easily obtained by the following recursive relationship:

$$p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1}) = \mathcal{N}(\cdot; \boldsymbol{x}_{t-1|t-1}, \boldsymbol{\Sigma}_{t-1|t-1}) \tag{2.9}$$

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1}) = \mathcal{N}(\cdot; \boldsymbol{x}_{t|t-1}, \boldsymbol{\Sigma}_{t|t-1}) \tag{2.10}$$

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) = \mathcal{N}(\cdot; \boldsymbol{x}_{t|t}, \boldsymbol{\Sigma}_{\boldsymbol{x}_{t|t}}) \tag{2.11}$$

where

$$\boldsymbol{x}_{t|t-1} = F_t \boldsymbol{x}_{t-1|t-1} \tag{2.12}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{x}_{t|t-1}} = F_t \boldsymbol{\Sigma}_{\boldsymbol{x}_{t-1|t-1}} F_t' + Q_t \tag{2.13}$$

$$\boldsymbol{y}_{t|t-1} = H_t \boldsymbol{x}_{t|t-1} \tag{2.14}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{y}_{t|t-1}} = H_t \boldsymbol{\Sigma}_{\boldsymbol{x}_{t|t-1}} H_t' + R_t \tag{2.15}$$

$$K_t = \boldsymbol{\Sigma}_{\boldsymbol{x}_{t|t-1}} H_t' \boldsymbol{\Sigma}_{\boldsymbol{y}_{t|t-1}}^{-1} \tag{2.16}$$

$$\boldsymbol{x}_{t|t} = \boldsymbol{x}_{t|t-1} + K_t(\boldsymbol{y}_t - \boldsymbol{y}_{t|t-1}) \tag{2.17}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{x}_{t|t}} = \boldsymbol{\Sigma}_{\boldsymbol{x}_{t|t-1}} - K_t H_t \boldsymbol{\Sigma}_{\boldsymbol{x}_{t|t-1}} \tag{2.18}$$

and where $\mathcal{N}$ denotes a Gaussian density. Herein, $\boldsymbol{x}_{r|s}$ and $\boldsymbol{\Sigma}_{\boldsymbol{x}_{r|s}}$ stand for the posterior mean and covariance matrix of the state vector $\boldsymbol{x}$ at time r given the observations up to time s, $\boldsymbol{y}_{1:s}$. That is, for any random state $\boldsymbol{x}_t$, $\boldsymbol{x}_{r|s} = \mathrm{E}(\boldsymbol{x}_r|\boldsymbol{y}_{1:s})$ and $\boldsymbol{\Sigma}_{\boldsymbol{x}_{r|s}} = \mathrm{Cov}(\boldsymbol{x}_r|\boldsymbol{y}_{1:s})$, $\forall r$ and $\forall s$. Also, H' and H^{-1} stand for the transpose and inverse matrix, respectively.

Expressions (2.12) – (2.18) define the famous Kalman filter, which is a recursive updating procedure that makes a preliminary estimate of the state $\boldsymbol{x}_{t|t-1}$ and then revises that estimate by incorporating a correction step $\boldsymbol{x}_{t|t} = \boldsymbol{x}_{t|t-1} + K_t(\boldsymbol{y}_t - \boldsymbol{y}_{t|t-1})$, where the so-called Kalman Gain K_t plays a key role in revising the preliminary estimate of the state, see e.g. Wei (1994). Under the KF, the equations (2.12) and (2.13) are called prediction (time update) equations. Likewise, equations (2.14) and (2.15) are the

prediction estimate and conditional covariance of the observed data y_t. The weight given to the information provided by the new observation is determined by the magnitud of the Kalman Gain K_t computed in equation (2.16); $\boldsymbol{y}_t - \boldsymbol{y}_{t|t-1}$ in equation (2.17) is called the innovation term. Finally, equations (2.17) and (2.18) are called filtering (measurement update) equations.

The following pseudo-code (Algorithm 1) summarizes the KF algorithm.

Algorithm 1 Kalman Filter

Initialization $t = 0$
Set initial conditions: $\boldsymbol{x}_{0|0}$ and $\boldsymbol{\Sigma}_{\boldsymbol{x}_{0|0}}$.
for $t = 1$ to T **do**
Step 1 Prediction step (time update)
Compute the predictive expectation $\boldsymbol{x}_{t|t-1}$ and covariance $\boldsymbol{\Sigma}_{\boldsymbol{x}_{t|t-1}}$ using equations (2.12) and (2.13), respectively
Step 2 Kalman Gain step (including computation of the prediction estimate and conditional covariance of the observed data y_t.)
Compute the prediction estimate $\boldsymbol{y}_{t|t-1}$ and covariance $\boldsymbol{\Sigma}_{\boldsymbol{y}_{t|t-1}}$ using equations (2.14) and (2.15), respectively.
Compute the Kalman Gain K_t with equation (2.16).
Step 3 Filtering step (based on new observation $\boldsymbol{y}_t$)
Compute the filtering expectation $\boldsymbol{x}_{t|t}$ and covariance $\boldsymbol{\Sigma}_{\boldsymbol{x}_{t|t}}$ using equations (2.17) and (2.18), respectively
end for

Notice that from a computational point of view, the most expensive step is the inversion of the matrix in equation (2.16). However, since its computation does not depend on the observations, it could be computed offline.

As Jazwinski (1970) points out, in case of Kalman models, it is a relatively simple matter to compute the conditional densities in (2.4) and (2.5). By contrast, in the nonlinear case, the situation is vastly more difficult, because in general, there does not exist a finite number of parameters which characterize these densities. Moreover, as Meinhold and Singpurwalla (1989) state, Kalman filters based on normality assumption are known to be non–robust, which implies that the posterior density may become unrealistic when there is a large difference between the prior density and the observed data.

In few cases, an explicit expression for the filtering problem can be derived in closed form; the linear and Gaussian dynamic model being one of the exceptions (Tanizaki 1991). Since most real-world problems are specified by nonlinear and (possibly) non-Gaussian state-space dynamic models, that usually preclude an exact solution of the filtering problem, approximative approaches are needed. For instance, when f and h in (2.1) and (2.2) are nonlinear, Algorithm 1 is no longer operational, because it involves the expectations of nonlinear functions. Therefore, the nonlinear measurement and transition equations would need to be approximated in order to evaluate the expectations involved in equations (2.9) – (2.11).

A great deal of literature is devoted to the theory and development of algorithms to tackle the nonlinear filtering problem. The most heuristic and easiest approximation is based on the use of Taylor

series expansions in order to linearize the nonlinear state-space model described by the system and/or measurement equations (2.1) and (2.2). The extended Kalman filter is one of such methods.

Although the main focus of our work is the so-called particle filtering methodology (fully described later on), next we first describe two non-simulation based nonlinear filters that have been used upon improving existing particle filters; say the aforementioned EKF and the so-called unscented Kalman filter.

2.3.2 The Extended Kalman Filter

Under the extended Kalman filter (EKF), the main idea is first to linearize the nonlinear functions f and h in (2.1) and (2.2) to later apply the Kalman filter given in equations (2.12) – (2.18). That is, the two nonlinear functions $f(\boldsymbol{x}_{t-1}, \boldsymbol{\eta}_t)$ and $h(\boldsymbol{x}_t, \boldsymbol{\nu}_t)$ are approximated by the first-order Taylor series expansion around $(\boldsymbol{x}_{t-1}, \boldsymbol{\eta}_t) = (\boldsymbol{x}_{t-1|t-1}, \mathbf{0})$ and $(\boldsymbol{x}_t, \boldsymbol{\nu}_t) = (\boldsymbol{x}_{t|t-1}, \mathbf{0})$, respectively. This is done so that the expectations in (2.12) – (2.18) can be evaluated explicitly; see for example Wishner, Tabaczynski, and Athans (1969), Anderson and Moore (1979), and Tanizaki and Mariano (1996).

Hence, in order to approximate the expectations of the dynamic nonlinear state-space model defined by (2.1)–(2.2), the corresponding measurement and transition equations are approximated by the following state-space model (SSM):

$$\begin{aligned} \boldsymbol{x}_t &= f(\boldsymbol{x}_{t-1}, \boldsymbol{\eta}_t) \\ &\approx f_{t|t-1} + J_{\boldsymbol{x}_{t-1}}(\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-1|t-1}) + J_{\boldsymbol{\eta}_t}\boldsymbol{\eta}_t, \end{aligned} \tag{2.19}$$

where

$$\begin{aligned} f_{t|t-1} &= f(\boldsymbol{x}_{t-1|t-1}, 0), \\ J_{\boldsymbol{x}_{t-1}} &= \frac{\partial f(\boldsymbol{x}_{t-1}, \boldsymbol{\eta}_t)}{\partial \boldsymbol{x}'_{t-1}}|_{(\boldsymbol{x}_{t-1}, \boldsymbol{\eta}_t)=(\boldsymbol{x}_{t-1|t-1}, 0)}, \\ J_{\boldsymbol{\eta}_t} &= \frac{\partial f(\boldsymbol{x}_{t-1}, \boldsymbol{\eta}_t)}{\partial \boldsymbol{\eta}'_t}|_{(\boldsymbol{x}_{t-1}, \boldsymbol{\eta}_t)=(\boldsymbol{x}_{t-1|t-1}, 0)}, \end{aligned}$$

and

$$\begin{aligned} \boldsymbol{y}_t &= h(\boldsymbol{x}_t, \boldsymbol{\nu}_t) \\ &\approx h_{t|t-1} + J_{\boldsymbol{x}_t}(\boldsymbol{x}_t - \boldsymbol{x}_{t|t-1}) + J_{\boldsymbol{\nu}_t}\boldsymbol{\nu}_t \end{aligned} \tag{2.20}$$

where

$$\begin{aligned} h_{t|t-1} &= h(\boldsymbol{x}_{t|t-1}, 0), \\ J_{\boldsymbol{x}_t} &= \frac{\partial h(\boldsymbol{x}_t, \boldsymbol{\nu}_t)}{\partial \boldsymbol{x}'_t}|_{(\boldsymbol{x}_t, \boldsymbol{\nu}_t)=(\boldsymbol{x}_{t|t-1}, 0)}, \\ J_{\boldsymbol{\nu}_t} &= \frac{\partial h(\boldsymbol{x}_t, \boldsymbol{\nu}_t)}{\partial \boldsymbol{\nu}'_t}|_{(\boldsymbol{x}_t, \boldsymbol{\nu}_t)=(\boldsymbol{x}_{t|t-1}, 0)}. \end{aligned}$$

Once the linearized system and measurement equations (2.19) and (2.20) are obtained, one is able to apply the KF given in Algorithm 1. That means that expressions (2.9) – (2.18) are now equivalent to the following alternative expressions (2.21) – (2.30):

$$p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1}) \approx \mathcal{N}(\cdot; \boldsymbol{x}_{t-1|t-1}, \boldsymbol{\Sigma}_{t-1|t-1}) \tag{2.21}$$

$$p(\boldsymbol{x}_{t}|\boldsymbol{y}_{1:t-1}) \approx \mathcal{N}(\cdot; \boldsymbol{x}_{t|t-1}, \boldsymbol{\Sigma}_{t|t-1}) \tag{2.22}$$

$$p(\boldsymbol{x}_{t}|\boldsymbol{y}_{1:t}) \approx \mathcal{N}(\cdot; \boldsymbol{x}_{t|t}, \boldsymbol{\Sigma}_{\boldsymbol{x}_{t|t}}) \tag{2.23}$$

where

$$\boldsymbol{x}_{t|t-1} = f_{t|t-1} \tag{2.24}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{x}_{t|t-1}} = J_{\boldsymbol{x}_{t-1}} \boldsymbol{\Sigma}_{\boldsymbol{x}_{t-1|t-1}} J'_{\boldsymbol{x}_{t-1}} + J_{\boldsymbol{\eta}_t} Q_t J'_{\boldsymbol{\eta}_t} \tag{2.25}$$

$$\boldsymbol{y}_{t|t-1} = h_{t|t-1} \tag{2.26}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{y}_{t|t-1}} = J_{\boldsymbol{x}_t} \boldsymbol{\Sigma}_{\boldsymbol{x}_{t|t-1}} J'_{\boldsymbol{x}_t} + J_{\boldsymbol{v}_t} R_t J'_{\boldsymbol{v}_t} \tag{2.27}$$

$$K_t = \boldsymbol{\Sigma}_{\boldsymbol{x}_{t|t-1}} J'_{\boldsymbol{x}_t} \boldsymbol{\Sigma}^{-1}_{\boldsymbol{y}_{t|t-1}} \tag{2.28}$$

$$\boldsymbol{x}_{t|t} = \boldsymbol{x}_{t|t-1} + K_t(\boldsymbol{y}_t - \boldsymbol{y}_{t|t-1}) \tag{2.29}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{x}_{t|t}} = \boldsymbol{\Sigma}_{\boldsymbol{x}_{t|t-1}} - K_t J_{\boldsymbol{x}_t} \boldsymbol{\Sigma}_{\boldsymbol{x}_{t|t-1}} \tag{2.30}$$

Expressions (2.24) – (2.30) define the extended Kalman filter, with a pseudo-code given in Algorithm 2.

Algorithm 2 Extended Kalman Filter

Initialization $t = 0$
Set initial conditions: $\boldsymbol{x}_{0|0}$ and $\boldsymbol{\Sigma}_{\boldsymbol{x}_{0|0}}$.
for $t = 1$ to T **do**
Step 1 Prediction step (time update)
Compute $f_{t|t-1}$, $J_{\boldsymbol{x}_{t-1}}$, and $J_{\boldsymbol{\eta}_t}$ with equation (2.19).
Compute the predictive expectation $\boldsymbol{x}_{t|t-1}$ and covariance $\boldsymbol{\Sigma}_{\boldsymbol{x}_{t|t-1}}$ using (2.24) and (2.25), respectively.
Step 2 Kalman Gain step
Compute $h_{t|t-1}$, $J_{\boldsymbol{x}_t}$, and $J_{\boldsymbol{v}_t}$ using equation (2.20).
Compute the prediction estimate $\boldsymbol{y}_{t|t-1}$ and covariance $\boldsymbol{\Sigma}_{\boldsymbol{y}_{t|t-1}}$ using equations (2.26) and (2.27), respectively.
Compute the Kalman Gain K_t with equation (2.28).
Step 3 Filtering step (measurement update)
Compute the filtering expectation $\boldsymbol{x}_{t|t}$ and covariance $\boldsymbol{\Sigma}_{\boldsymbol{x}_{t|t}}$ using (2.29) and (2.30), respectively.
end for

Recall that Q_t stands here for the state-noise covariance matrix and R_t for the measurement-noise covariance matrix. Both the system noise, as well as the measurement noise, are assumed to be zero

mean Gaussian. Moreover, $J_{x_{t-1}}$ and J_{η_t} are the Jacobian for the system, and J_{x_t} and J_{v_t} are the Jacobian for the measurement. Notice that the EKF reduces to the plain KF algorithm when h and f are both linear.

A drawback of this algorithm is that it can lead to poor representations of the nonlinear functions and target probability distributions (Anderson and Moore 1979). The approximation errors introduced when computing the posterior mean and covariance estimates obtained via the EKF could be large. As a result, this filter may have poor estimation performance and could even diverge (Van der Merwe 2004).Additionally, from a computational point of view, also some efficiency is lost because no computation can be made offline.

Since approximating the expectations of a nonlinear function by Taylor series expansion may give a biased estimate and often seems to underestimate the covariance of the states, other methods are required. Tanizaki (1996) studies different nonlinear algorithms based on Taylor series expansions (including the EKF). Also, various density based filters are developed, including the well-known numerical integration filter (NIF) and the so-called rejection sampling filter (RSF) introduced by Kitagawa (1987) and Tanizaki and Mariano (1998), respectively. Another filter, introduced to overcome the approximation drawbacks of the EKF and studied in this work, is the so-called unscented Kalman filter (UKF) (Wan and Van der Merwe 2001).

Next we provide a detailed description of the unscented Kalman filter as well as of the so-called Unscented transformation UT) and the scaled Unscented transformation (SUT) methods, which are the building blocks of the UKF.

2.3.3 The Unscented Kalman Filter

To fully describe the unscented Kalman filter (UKF) some fundamental ideas need to be presented: the concepts of unscented transformation and scaled unscented transformation. Following, we describe these concepts.

The Unscented Transformation

The unscented transformation (UT) is a method used to compute the statistics of a random variable which undergoes a nonlinear transformation. It is founded on the principle that "it is easier to approximate a probability distribution than it is to approximate an arbitrary nonlinear function" (Julier and Uhlmann 1997).

The UT works as follows: Let $\boldsymbol{x}$ be an n_x-dimensional random variable with mean $\bar{\boldsymbol{x}}$ and covariance $\Sigma_{\boldsymbol{x}}$. Suppose that the aim is to calculate the statistics of a nonlinear transformation or nonlinear function, say $\boldsymbol{y} = f(\boldsymbol{x})$. That is, the aim is to propagate the random vector $\boldsymbol{x}$ through the nonlinear function f.

According to the UT, one proceeds to deterministically choose and calculate a set of $2n_x+1$ weighted sigma (sample) points $\{(\boldsymbol{\chi}_i', \boldsymbol{\omega}_i'), i = 0,\ldots,2n_x\}$.

The deterministically chosen sigma points and respective weights are given by:

$$\begin{aligned}
\boldsymbol{\chi}'_0 &= \bar{\boldsymbol{x}}, & \boldsymbol{\omega}'_0 &= \kappa/(n_x+\kappa),\ i=0\\
\boldsymbol{\chi}'_i &= \bar{\boldsymbol{x}} + \left(\sqrt{(n_x+\kappa)\Sigma_{\boldsymbol{x}}}\right)_i, & \boldsymbol{\omega}'_i &= \frac{1}{2(n_x+\kappa)},\quad i=1,\ldots,n_x\\
\boldsymbol{\chi}'_i &= \bar{\boldsymbol{x}} - \left(\sqrt{(n_x+\kappa)\Sigma_{\boldsymbol{x}}}\right)_i, & \boldsymbol{\omega}'_i &= \frac{1}{2(n_x+\kappa)},\quad i=n_x+1,\ldots,2n_x.
\end{aligned} \tag{2.31}$$

where

- κ is a scaling parameter (its choice is critical to guarantee a positive semi-definite covariance matrix),
- $\left(\sqrt{(n_x+\kappa)\Sigma_{\boldsymbol{x}}}\right)_i$ is the ith row[2] or column of the matrix square root[3] of $(n_x+\kappa)\Sigma_{\boldsymbol{x}}$,
- $\boldsymbol{\omega}'_i$ are the corresponding computed sigma points weights, such that $\sum_{i=0}^{2n_x}\boldsymbol{\omega}'_i = 1$.

Once the sigma points $\boldsymbol{\chi}'_i$ and respective weights $\boldsymbol{\omega}'_i$ are computed, each sigma point is propagated in time through the true nonlinear function, say $\boldsymbol{y}' = f(\boldsymbol{\chi}'_i)$, $i = 0,\ldots,2n_x$. Then, an estimate of the nonlinear transformed mean and covariance, based on statistics of the transformed sigma points, is obtained. Specifically, the mean and covariance for $\boldsymbol{y}$ are approximated by a weighted average and a weighted outer product of the transformed sigma points, $\boldsymbol{y}'$. That is,

$$\begin{aligned}
\bar{\boldsymbol{y}} &\approx \sum_{i=0}^{2n_x}\boldsymbol{\omega}'_i\boldsymbol{y}'_i\\
\Sigma_{\boldsymbol{y}} &\approx \sum_{i=0}^{2n_x}\boldsymbol{\omega}'_i(\boldsymbol{y}'_i-\bar{\boldsymbol{y}})(\boldsymbol{y}'_i-\bar{\boldsymbol{y}})^T.
\end{aligned}$$

Likewise, the cross-covariance is approximated by a weighted cross-covariance $\Sigma_{\boldsymbol{xy}}$ computed by

$$\Sigma_{\boldsymbol{xy}} \approx \sum_{i=0}^{2n_x}\boldsymbol{\omega}'_i(\boldsymbol{\chi}'_i-\bar{\boldsymbol{x}})(\boldsymbol{y}'_i-\bar{\boldsymbol{y}})^T.$$

The right panel of Figure 2.2 illustrates the functioning of the (just described) UT (building block of the UKF) method. Additionally, this figure illustrates that the UT generally outperforms the EKF when approximating the first two moments of a Gaussian random variable that undergoes an arbitrary nonlinear transformation. The poor performance in the posterior mean (clearly biased) and covariance (highly inaccurate) estimates obtained via the EKF (middle) as opposed to the optimal filter (left) and the UKF (right) is clearly portrayed.

Recall that the EKF has two well-known major drawbacks. First, if the local linearity assumption does not hold, the performed linearization can lead to unstable filters. Second, implementation difficulties can occur in applications with nontrivial Jacobian matrices derivation. The UT, on the other

[2] Use the rows, if the matrix square root $\mathbf{A}$ of $\mathbf{P}$ is of the form $\mathbf{P} = \mathbf{A}'\mathbf{A}$. In case $\mathbf{P} = \mathbf{A}\mathbf{A}'$, the sigma points are formed from the colums of A

[3] For its computation, a numerically efficient method should be used, such as Cholesky decomposition or singular-value decomposition; see Wood (2004) and R Development Core Team (2013), respectively.

hand, does not have to deal with such approximation problems. Therefore, the UT is used to construct the scaled unscented transformation, which itself is used as a building block for the so-called unscented Kalman filter (explained later on) aiming to overcome the approximation drawbacks of the extended Kalman filter.

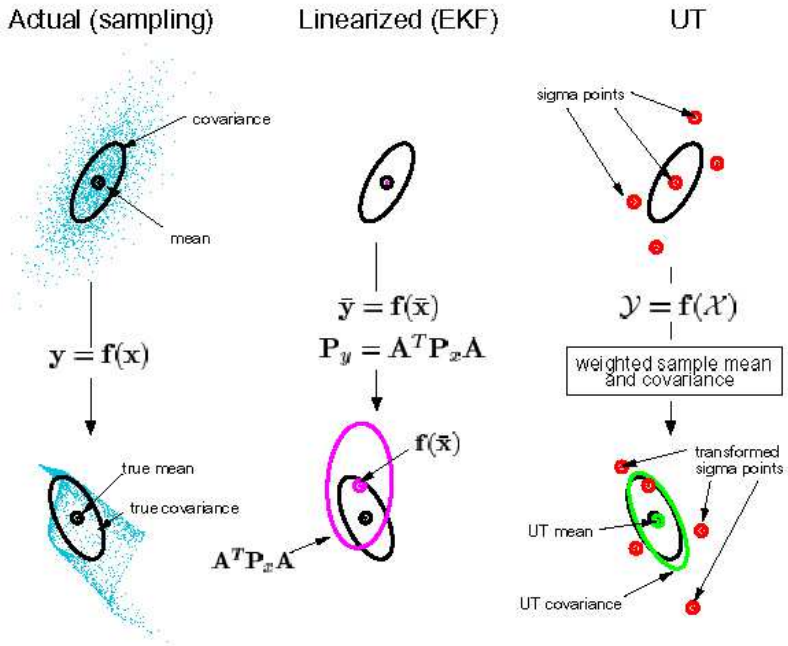


Figure 2.2: Illustration of the functioning of the UT (building block of the UKF) method. Additionally, this figure illustrates that the UT generally outperforms the EKF when approximating the first two moments of a Gaussian random variable that undergoes an arbitrary nonlinear transformation. The poor performance in the posterior mean (clearly biased) and covariance (highly inaccurate) estimates obtained via the EKF (middle) as opposed to the optimal filter (left) and the UKF (right) is clearly portrayed. Figure reproduced from Julier and Uhlmann (2004)

The unscented transformation provides a good approximation of the mean and covariance of $\boldsymbol{y}' = f(\boldsymbol{x})$, nonetheless scaling problems might occur. In case of severe nonlinearities, things become worse. For instance, the selection strategy just described has the property that the dispersion of the sigma points increases as the dimension of the random variable increases. To avoid difficulties an appropriate choice of the scaling parameter κ is then crucial to guarantee a positive definite covariance matrix. All this can be summarized as follows: the use of UT under severe nonlinearities may cause scaling problems. Thus, as a solution the scaled unscented transformation (SUT) is proposed to handle such possible scaling problems.

The Scaled Unscented Transformation

The main feature of the SUT is that the sigma points can be scaled towards or away from the mean of the prior distribution by a proper choice of the scaling parameter κ; see Wan and Van der Merwe (2001)

and Julier (2002). Under this method, the original set of sigma points $\{\boldsymbol{\chi}'_i\}$ are scaled-transformed and thus replaced by

$$\boldsymbol{\chi}_i = \boldsymbol{\chi}'_0 + \alpha(\boldsymbol{\chi}'_i - \boldsymbol{\chi}'_0), \quad i = 0,\ldots,2n_x, \tag{2.32}$$

where α is a positive scaling parameter which determines the spread of the sigma points around $\boldsymbol{\chi}'_0 = \bar{\boldsymbol{x}}$. The advantage of using the later formulation is that it allows the controlling of the scaling of the sigma points without causing the resulting covariance to possibly become non-positive semidefinite.

A form of obtaining the transformed sigma points $\boldsymbol{\chi}_i$ in (2.32) is simply to apply the previously described plain UT on a set of pre-scaled sigma points. Following this approach, the new set of properly scaled sigma points and weights, $\{(\boldsymbol{\chi}_i, \boldsymbol{\omega}_i), i = 0,\ldots,2n_x\}$ is obtained by

$$\begin{aligned}
\boldsymbol{\chi}_0 &= \bar{\boldsymbol{x}}, & \boldsymbol{\omega}_0^{(m)} &= \lambda/(n_x+\lambda), & i &= 0\\
\boldsymbol{\chi}_i &= \bar{\boldsymbol{x}} + \left(\sqrt{(n_x+\lambda)\Sigma_{\boldsymbol{x}}}\right)_i, & \boldsymbol{\omega}_i^{(m)} &= \frac{1}{2(n_x+\lambda)}, & i &= 1,\ldots,n_x\\
\boldsymbol{\chi}_i &= \bar{\boldsymbol{x}} - \left(\sqrt{(n_x+\lambda)\Sigma_{\boldsymbol{x}}}\right)_i, & \boldsymbol{\omega}_i^{(m)} &= \frac{1}{2(n_x+\lambda)}, & i &= n_x+1,\ldots,2n_x\\
\boldsymbol{\omega}_0^{(c)} &= \lambda/(n_x+\lambda) + (1-\alpha^2+\beta), & \boldsymbol{\omega}_i^{(c)} &= \boldsymbol{\omega}_i^{(m)}, & i &= 1,\ldots,2n_x
\end{aligned} \tag{2.33}$$

where

- α is a positive scaling parameter which is usually set to a small positive value (e.g., 0.001). In case of severe nonlinearities, $0 < \alpha \leq 1$ is chosen.
- κ is a secondary scaling parameter which is usually set to 0. Though its choice is not critical, to guarantee a positive definite covariance matrix, $\kappa \geq 0$ must be chosen.
- β is a non-negative parameter introduced to incorporate prior knowledge of the distribution of $\boldsymbol{x}$. That is, $\beta \geq 0$ must be chosen and $\beta = 2$ is optimal for Gaussian distributions.
- $\lambda = \alpha^2(n_x+\kappa) - n_x$ is the new scaling parameter defined in terms of α, κ and n_x.
- $\left(\sqrt{(n_x+\lambda)\Sigma_{\boldsymbol{x}}}\right)_i$ is the ith row or column of the matrix square root of $(n_x+\lambda)\Sigma_{\boldsymbol{x}}$,
- $\boldsymbol{\omega}_i$ are the corresponding computed sigma points weights, such that $\sum_{i=0}^{2n_x}\boldsymbol{\omega}_i = 1$.

Once the scaled sigma points and corresponding weights are computed, each sigma point is propagated through the nonlinear transformation, say $\boldsymbol{y} = f(\boldsymbol{\chi}_i)$, $i = 0,\ldots,2n_x$. The computation of the mean $\bar{\boldsymbol{y}}$, variance $\Sigma_{\boldsymbol{y}}$ and cross-covariance $\Sigma_{\boldsymbol{xy}}$, based on the former properly scaled sigma points, is given by

$$\begin{aligned}
\bar{\boldsymbol{y}} &\approx \sum_{i=0}^{2n_x}\boldsymbol{\omega}_i^{(m)}\boldsymbol{y}_i\\
\Sigma_{\boldsymbol{y}} &\approx \sum \boldsymbol{\omega}_i^{(c)}\left(\boldsymbol{y}_i - \bar{\boldsymbol{y}}\right)\left(\boldsymbol{y}_i - \bar{\boldsymbol{y}}\right)^T\\
\Sigma_{\boldsymbol{xy}} &\approx \sum_{i=0}^{2n_x}\boldsymbol{\omega}_i(\boldsymbol{\chi}_i - \bar{\boldsymbol{x}})(\boldsymbol{y}_i - \bar{\boldsymbol{y}})^T.
\end{aligned}$$

The previous estimates, obtained through the SUT, are as accurate as the ones obtained through the plain UT, but the scale unscented transformation allows one to better tackle the scale–errors introduced by high non-linearities. The SUT is thus the main building block for the design of the unscented Kalman filter.

The Unscented Kalman Filter Implementation

The unscented Kalman filter (UKF) is a straightforward application of the SUT to the recursive Kalman filtering estimation problem (Wan and Van der Merwe 2000; Wan and Van der Merwe 2001; Van der Merwe 2004). That is, the UKF provides a Gaussian approximation to the posterior state distribution, where the first two moments are updated and obtained by using the SUT.

The general UKF approach works as follows:

First, under the UKF, the state random variable is redefined as the concatenation of the the original state and the noise variables; say $\boldsymbol{x}_t^{\boldsymbol{a}} = (\boldsymbol{x}', \boldsymbol{\eta}', \boldsymbol{\nu}')'$. Likewise, the augmented state covariance matrix Σ^a is redefined in terms of the individual covariances of the state, process and measurement variables; that is,

$$\Sigma^a = \begin{pmatrix} \Sigma & 0 & 0 \\ 0 & Q & 0 \\ 0 & 0 & R \end{pmatrix}. \tag{2.34}$$

The augmented state random variable $\boldsymbol{x}_t^{\boldsymbol{a}}$ has the effective dimension $n_a = n_x + n_\eta + n_\nu$; being n_x, n_η and n_ν the original state-, the process noise- and observation noise dimension, respectively.

A natural consequence of augmenting the original state random variable with the noise variables is that not only the uncertainty present in the states, but also the uncertainty in the noise disturbances, are taken into account during the sigma (sample) points propagation. However in case of the EKF, the nonlinear functions are first approximated (linearized) by a Taylor series expansion of order one around the current expected means and the uncertainty is thus not taken into account in the linearization.

Second, the SUT sigma point selection scheme, specified by equation (2.33), is used to compute the new set of scaled sigma points $\boldsymbol{\chi}_t^a$ for the augmented state random variable $\boldsymbol{x}_t^{\boldsymbol{a}}$. Recall that λ is the scale parameter computed itself as a function of the other parameters α (which determines the spread of sigma points), κ (which guarantees a positive covariance matrix) and β (which incorporates prior knowledge of distribution). The values of the parameters (α, κ, β) involved in the UKF can be tuned by the researcher.

Finally, based on the SUT statistics, corresponding expressions for prediction, Kalman Gain, innovation, and filtering (equivalent to the KF equations (2.12)–(2.18)) are obtained. The following pseudocode (Algorithm 3) subsumes the UKF algorithm.

Algorithm 3 Unscented Kalman Filter

Initialization $t = 0$

Initial conditions:

$\bar{\boldsymbol{x}}_0 = \mathrm{E}(\boldsymbol{x_0})$

$\Sigma_0 = E\left((\boldsymbol{x_0} - \bar{\boldsymbol{x}}_0)(\boldsymbol{x_0} - \bar{\boldsymbol{x}}_0)'\right)$

$\bar{\boldsymbol{x}}_0^a = \mathrm{E}(\boldsymbol{x_0^a}) = (\bar{\boldsymbol{x}}_0', \boldsymbol{0}, \boldsymbol{0})'$

$\Sigma_0^a = E\left((\boldsymbol{x_0^a} - \bar{\boldsymbol{x}}_0^a)(\boldsymbol{x_0^a} - \bar{\boldsymbol{x}}_0^a)'\right)$ using (2.34)

Set parameter values: α, β and κ.

Compute the dimension of augmented state: $n_a = n_x + n_\eta + n_\nu$

for $t = 1$ to T **do**

Step 1 Computation of weighted sigma points

Compute λ using $\lambda = \alpha^2(n_x + \kappa) - n_x$

Define the augmented state and corresponding weights as:

$\boldsymbol{x_{t-1}^a} = (\boldsymbol{x}', \boldsymbol{\eta}', \boldsymbol{\nu}')'$ and $\boldsymbol{\chi_{t-1}^a} = ((\boldsymbol{\chi}^x)', (\boldsymbol{\chi}^\eta)', (\boldsymbol{\chi}^\nu)')'$.

Obtain the sigma points with

$$\boldsymbol{\chi}_{t-1}^a = \left[\bar{\boldsymbol{x}}_{t-1}^a, \bar{\boldsymbol{x}}_{t-1}^a \pm \left(\sqrt{(n_a + \lambda)\Sigma_{t-1}^a}\right)\right]$$

and the corresponding weights using (2.33)

Step 2 Time update step (propagation of sigma points into the future)

$$\boldsymbol{\chi}_{t|t-1}^x = f(\boldsymbol{\chi}_{t-1}^x, \boldsymbol{\chi}_{t-1}^\eta)$$

$$\bar{\boldsymbol{x}}_{t|t-1} = \sum_{i=0}^{2n_a} \boldsymbol{\omega}_i^{(m)} \boldsymbol{\chi}_{i,t|t-1}^x$$

$$\Sigma_{t|t-1} = \sum_{i=0}^{2n_a} \boldsymbol{\omega}_i^{(c)} (\boldsymbol{\chi}_{i,t|t-1}^x - \bar{\boldsymbol{x}}_{t|t-1})(\boldsymbol{\chi}_{i,t|t-1}^x - \bar{\boldsymbol{x}}_{t|t-1})'$$

$$\tilde{\boldsymbol{y}}_{t|t-1} = h(\boldsymbol{\chi}_{t|t-1}^x, \boldsymbol{\chi}_{t-1}^\nu)$$

$$\bar{\boldsymbol{y}}_{t|t-1} = \sum_{i=0}^{2n_a} \boldsymbol{\omega}_i^{(m)} \tilde{\boldsymbol{y}}_{i,t|t-1}$$

Step 3 Measurement update equations step (incorporate new observation)

$$\Sigma_{\tilde{\boldsymbol{y}}_t, \tilde{\boldsymbol{y}}_t} = \sum_{i=0}^{2n_a} \boldsymbol{\omega}_i^{(c)} (\tilde{\boldsymbol{y}}_{i,t|t-1} - \bar{\boldsymbol{y}}_{t|t-1})(\tilde{\boldsymbol{y}}_{i,t|t-1} - \bar{\boldsymbol{y}}_{t|t-1})'$$

$$\Sigma_{\boldsymbol{x}_t, \tilde{\boldsymbol{y}}_t} = \sum_{i=0}^{2n_a} \boldsymbol{\omega}_i^{(c)} (\boldsymbol{\chi}_{i,t|t-1} - \bar{\boldsymbol{x}}_{t|t-1})(\tilde{\boldsymbol{y}}_{i,t|t-1} - \bar{\boldsymbol{y}}_{t|t-1})'$$

$$K_t = \Sigma_{\boldsymbol{x}_t, \tilde{\boldsymbol{y}}_t} \Sigma_{\tilde{\boldsymbol{y}}_t, \tilde{\boldsymbol{y}}_t}^{-1}$$

$$\Sigma_t = \Sigma_{t|t-1} - K_t \Sigma_{\tilde{\boldsymbol{y}}_t, \tilde{\boldsymbol{y}}_t} K_t'$$

$$\bar{\boldsymbol{x}}_t = \bar{\boldsymbol{x}}_{t|t-1} + K_t(\boldsymbol{y}_t - \bar{\boldsymbol{y}}_{t|t-1})$$

end for

A summarized comparison of two Nonlinear Filters: The EKF vs. the UKF

Comparing the EKF and UKF nonlinear filters, it is concluded that:

- Both filters, the EKF (with pseudo-code in Algorithm 2) and UKF (with pseudo-code in Algorithm 3), provide a Gaussian approximation to the posterior state distribution, but differ in the way the first two moments of such Gaussian posterior are estimated.

- Under the EKF, the nonlinear functions contained in the state-space model are first approximated using a Taylor series expansion of order one, and then the two moments (mean and covariance) of the Gaussian PDF of the states are propagated (in time) via the Kalman filter. That is, a straightforward application of the Kalman filter (Algorithm 1) to the linearized nonlinear functions involved in the state-space model gives rise to the nonlinear EKF.

- In contrast to the EKF, the UKF directly approximates the Gaussian posterior distribution of the states $p(\boldsymbol{x}|\boldsymbol{y}_{1:t})$ using the specified nonlinear model, being the UKF a straightforward application of the SUT to the recursive Kalman filtering estimation problem (Wan and Van der Merwe 2000; Wan and Van der Merwe 2001). In other words, the UKF uses a minimal set of deterministically chosen sigma points that are propagated (in time) through the true nonlinearity and are then used to produce the Gaussian approximation to $p(\boldsymbol{x}|\boldsymbol{y}_{1:t})$. These sigma points, which are propagated (in time) through the true nonlinearity are said to completely capture the true mean and covariance of the filtering Gaussian random variable; see again illustration in Figure 2.2.

- Under the UKF, the approximated first two moments of the filtering PDF of the states are generally more accurate than the ones yielded by the EKF (Van der Merwe 2004); specially in case of high nonlinearities. Intuitively, if the nonlinear functions involved in the stat-space model are not well approximated, the quality of the estimated mean and covariance of the filtering Gaussian PDF will be negatively affected, since large errors may have been introduced.

- In contrast to the EKF, in case of the UKF no Jacobians need to be computed, which means that the UKF is derivative free. As aforementioned, the EKF has two well-known major drawbacks. First, if the local linearity assumption does not hold, the performed linearization can lead to unstable filters. Second, implementation difficulties can occur in applications with nontrivial Jacobian matrices derivation. Thus, the Unscented Kalman filter aims to overcome the approximation drawbacks of the extended Kalman filter.

- The computational cost of the UKF is said to be of about the same order as that of the EKF (Van der Merwe, Doucet, de Freitas, and Wan 2001); we will consider this point further in Chapter 4 dealing with MC studies with nonlinear state-space model. A glance to the pseudo-codes for the EKF (Algorithm 2) and the UKF (Algorithm 3) shows that the implementation of the UKF involves more complex instructions.

- Based on simulations, Wan and Van der Merwe (2000) show that compared to the EKF, the UKF provides more accurate state estimates and much better estimates for the covariance of the states. These authors also found that the UKF has the capability of generating heavier tailed distributions than the EKF, and that the computational efficiency of the UKF and the EKF is of the same order. In conclusion, they reported a superior filtering performance of the UKF over the EKF. This feature naturally leads to the later use of the UKF in order to improve existing particle filters as done by Van der Merwe, Doucet, de Freitas, and Wan (2001) and Van der Merwe (2004).

According to an analysis of reviewed literature, the UKF proves its superior performance in case of complex nonlinearities (where the EKF is more prone to fail). This is confirmed by our MC experiments in Chapter 4 (partially based on a smaller number of experiments presented in (Van der Merwe, Doucet, de Freitas, and Wan 2001) and Van der Merwe (2004)), dealing with a dynamic state-space model which is nonlinear, non-stationary and non-Gaussian. Though we also consider that the UKF improves greatly upon the EKF, a major drawback remains: it does not apply to general non-Gaussian distributions.

As mentioned earlier, the fully described nonlinear EKF and UKF filtering methods rely on approximating the filtering distribution of the state variable by a Gaussian PDF. This may naturally lead to estimation difficulties when dealing with general non-Gaussian models, specially the EKF which has a lower order estimation accuracy compared to the UKF. A very different nonlinear filtering approach, based on simulations, is the particle filtering methodology, which in contrast to previously described filters does not impose a Gaussian posterior distribution of the variable (states) of interest.

Next section makes a thorough description of the main concepts within the particle filtering methodology. The limitations of this approach are also highlighted, together with proposed solutions presented throughout the time, which have given rise to several particle filter variants. This work focuses on the most classic particle filter variants, based on sequential importance sampling, considered by us the building blocks of more recent approaches within the particle filtering methodology.

2.4 Dynamic State Estimation: Sequential Monte Carlo Filtering Methodology

The last two decades have brought about great developments in the theory and application of the sequential Monte Carlo (SMC) methodology known as particle filtering. This evolution can be noticed by the number of publications that have appeared in the field after Gordon's 1993 paper. For filtering the states, see for instance the work of the authors Gordon, Salmond, and Smith (1993), Kitagawa (1996), Fearnhead (1998), Pitt and Shephard (1999),Van der Merwe, Doucet, de Freitas, and Wan (2000), Wan and Van der Merwe (2001),Arulampalam, Maskell, Gordon, and Clapp (2002), Van der Merwe (2004), Muñoz, Márquez, and Acosta (2007), and Doucet and Johansen (2011).

We state that our first incursion into the sequential Monte Carlo methodology began with Kitagawa's 1996 paper which deals solely with optimal filtering and smoothing, assuming known model parameters. To our knowledge, Kitagawa is the first author that uses the word 'particles' as a synonym of the word samples, though the name 'particle filter' seems to be due to Fearnhead (1998).

A particle filter (PF) is a very flexible SMC approach that allows an implementation of a recursive Bayesian filter through Monte Carlo simulations (Doucet, Godsill, and Andrieu 2000; Arulampalam, Maskell, Gordon, and Clapp 2002; Doucet and Johansen 2011). It can be applied to a wide range of dynamic state-space models; being linear or not, Gaussian or not, stationary or not, discrete, continuous or hybrid. Though flexible, the particle filtering methodology has its own drawbacks, such as sample degeneracy and sample impoverishment. The existence of such drawbacks, the different attempts to overcome them and thus improve upon existing particle filters, is what has given raise to the different PF variants.

Particle filters have the main feature of approximating a target posterior probability distribution function by a set of weighted 'particles'. Hence, under the particle filtering methodology, the target posterior PDF of the state $\boldsymbol{x}_t$, in our case $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$, is approximated by samples (particles) that are recursively generated from the prediction and filtering distributions as new information becomes available. Once the filtering PDF $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$ is approximated, any characteristic of the state $\boldsymbol{x}_t$ can be estimated from the set of weighted particles.

By definition, all particle filters variants are sequential in nature and thus need a sampling scheme to generate the particles sequentially over time. To achieve that, this work relies on the so-called sequential importance sampling (SIS) principle. We are aware, however, that other approaches such as a rejection sampling filter can be used to generate samples in a sequential fashion (Hürzeler and Künsch 1998).

Let us first explain briefly the principle of importance sampling (IS).

Importance sampling

In practice it is seldom possible to sample directly from a true PDF, say $p(z)$, which often is only known up to a proportionality constant. In those cases, the IS principle allows us to choose an alternative PDF, say $q(z)$, from which it is easy to sample. This alternative PDF is called proposal or importance distribution. Therefore, instead of sampling directly from $p(z)$, we rather sample indirectly from it through $q(z)$. This proposal distribution must be as close as possible to the true one. Particularly, the support of $q(z)$ must include that of $p(z)$ (Geweke 1989).

As stated before, a particle filter is a simulation based filter that aims to recursively approximate the filtering PDF $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$ in (2.5). Suppose that at *fixed time t*, the filtering PDF of the random variable $\boldsymbol{x}_t|\boldsymbol{y}_{1:t}$ is approximated by a sufficiently large set of M particles[4] $\boldsymbol{x}_t^{(1)},\ldots,\boldsymbol{x}_t^{(M)}$ with discrete probability mass of $\boldsymbol{\omega}_t^{(1)},\ldots,\boldsymbol{\omega}_t^{(M)}$. Suppose further that these particles could not be drawn directly from $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$

[4]Throughout this work, both M or N_p are used to denote the number of particles.

but indirectly from the normalized importance PDF $q(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$, then the true filtering PDF $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$ at a *fixed* time t would be approximated by

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) \approx \sum_{j=1}^{M} \widetilde{\boldsymbol{\omega}}_t^{(j)} \delta(\boldsymbol{x_t} - \boldsymbol{x}_t^{(j)}) \tag{2.35}$$

where

$$\widetilde{\boldsymbol{\omega}}_t^{(j)} = \frac{\boldsymbol{\omega}_t^{(j)}}{\sum_{j=1}^{M} \boldsymbol{\omega}_t^{(j)}}, \quad \text{and} \quad \boldsymbol{\omega}_t^{(j)} \approx \frac{p(\boldsymbol{x}_t^j|\boldsymbol{y}_{1:t})}{q(\boldsymbol{x}_t^j|\boldsymbol{y}_{1:t})}, \tag{2.36}$$

and the values of $\boldsymbol{\omega}_t^{(j)}$ are the importance weights with corresponding normalized weights given by $\widetilde{\boldsymbol{\omega}}_t^{(j)}$. The approximation in (2.35) converges to the true filtering PDF $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$ when the sample size $M \to \infty$, being $\delta(.)$ the Dirac function (Doucet 1998; Doucet, Godsill, and Andrieu 2000).

However, since one needs to be able to estimate $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$ recursively in time, it becomes necessary to define a proposal distribution that works well in a sequential framework. This gives raise to the Bayesian sequential importance sampling (SIS) filter. Following, we present a detailed description of the SIS filter.

2.4.1 The Bayesian Sequential Importance Sampling Filter

In a sequential framework, the IS principle must be modified so that at any time k, the estimate of $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})$ can be propagated in time without modifying subsequently the past simulated trajectories. In other words, the importance PDF at time $k+1$ must admit as a marginal distribution the importance PDF at previous time k (Doucet 1998). This is possible if one chooses proposal distributions that factorize, such as

$$q(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) = q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{y}_t) q(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1}) \tag{2.37}$$

which provides samples $\boldsymbol{x}_t^{(j)} \sim q(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$ by augmenting each of the existing samples $\boldsymbol{x}_{t-1}^{(j)} \sim q(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1})$ with the new state $\boldsymbol{x}_t^{(j)} \sim q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{y}_t)$. Substituting the recursive filtering PDF expression (2.6) and (2.37) in (2.36), the importance weight equation can be rewritten as

$$\boldsymbol{\omega}_t^{(j)} \propto \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t^{(j)}) p(\boldsymbol{x}_t^{(j)}|\boldsymbol{x}_{t-1}^{(j)}) p(\boldsymbol{x}_{t-1}^{(j)}|\boldsymbol{y}_{1:t-1})}{q(\boldsymbol{x}_t^{(j)}|\boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{y}_t) q(\boldsymbol{x}_{t-1}^{(j)}|\boldsymbol{y}_{1:t-1})} = \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t^{(j)}) p(\boldsymbol{x}_t^{(j)}|\boldsymbol{x}_{t-1}^{(j)})}{q(\boldsymbol{x}_t^{(j)}|\boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{y}_t)} \boldsymbol{\omega}_{t-1}^{(j)} \tag{2.38}$$

and therefore the true posterior PDF $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$ can be sequentially approximated by the empirical PDF

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) \approx \sum_{j=1}^{M} \widetilde{\boldsymbol{\omega}}_t^{(j)} \delta(\boldsymbol{x}_t - \boldsymbol{x}_t^{(j)}), \tag{2.39}$$

where in this case the normalized weights $\widetilde{\boldsymbol{\omega}}_t^{(j)}$ are obtained from the weights defined in equation (2.38). These weights provide a measure of how likely the corresponding particle is, that is, large weights are assigned to particles with a large likelihood, and low weights to the ones with small likelihood values.

Algorithm 4 SIS Filter

Initialization $t = 0$
  for $j = 1$ to M **do**
    Sample $\boldsymbol{x}_0^{(j)} \sim p(\boldsymbol{x}_0)$ (Random sample taken from $p(\boldsymbol{x}_0)$ with $\boldsymbol{\omega}_0^{(j)} = \frac{1}{M}$)
  end for
  for $t = 1$ to N **do**
Step 1 Importance sampling step
    for $j = 1$ to M **do**
      Prediction Sample particles from proposal PDF $\boldsymbol{x}_t^{(j)} \sim q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{y}_t)$
      Filtering Assign to each particle $\boldsymbol{x}_t^{(j)}$ the importance weight $\boldsymbol{\omega}_t^{(j)}$ according to (2.38)
    end for
    for $j = 1$ to M **do**
      Normalize the importance weights: $\tilde{\boldsymbol{\omega}}_t^{(j)} = \frac{\boldsymbol{\omega}_t^{(j)}}{\sum_{i=1}^{M} \boldsymbol{\omega}_t^{(i)}}$
    end for
  end for

Following, the corresponding pseudo-code (Algorithm 4) for the general form of a SIS filter is given.

An important issue-but not an easy task-is the design of an optimal proposal distribution. Next, this topic is briefly commented.

Proposal distribution design

Doucet (1998) mentions how to choose a good importance PDF and shows that the optimal importance distribution is one of the form $q(\boldsymbol{x}_t^{(j)}|\boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{y}_t)$, which includes the information provided by the last observation. In practice, however, the most usual (and appealing) strategy consists of sampling from the probabilistic model of the states evolution, meaning to use as proposal PDF the so-called transition prior:

$$q(\boldsymbol{x}_t^{(j)}|\boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{y}_t) = p(\boldsymbol{x}_t^{(j)}|\boldsymbol{x}_{t-1}^{(j)}) \tag{2.40}$$

The transition prior is used as a proposal PDF by various authors; among others by Gordon, Salmond, and Smith (1993), Kitagawa (1996), Hürzeler and Künsch (1998), Arulampalam, Maskell, Gordon, and Clapp (2002) and Muñoz, Márquez, and Acosta (2007). If the transition prior (2.40) is used as proposal distribution, the general expression for the importance weights in (2.38) becomes

$$\boldsymbol{\omega}_t^{(j)} \propto \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t^{(j)})p(\boldsymbol{x}_t^{(j)}|\boldsymbol{x}_{t-1}^{(j)})}{q(\boldsymbol{x}_t^{(j)}|\boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{y}_t)}\boldsymbol{\omega}_{t-1}^{(j)} = \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t^{(j)})p(\boldsymbol{x}_t^{(j)}|\boldsymbol{x}_{t-1}^{(j)})}{p(\boldsymbol{x}_t^{(j)}|\boldsymbol{x}_{t-1}^{(j)})}\boldsymbol{\omega}_{t-1}^{(j)} = p(\boldsymbol{y}_t|\boldsymbol{x}_t^{(j)})\boldsymbol{\omega}_{t-1}^{(j)} \tag{2.41}$$

and are proportional to the product of the likelihood of the new observation $\boldsymbol{y_t}$ given that particle and the previous weight. Hence, under the Bayesian sequential importance sampling filter using the transition prior as a proposal PDF, the importance weights are propagated forward and updated as a new observation arrives using equation (2.41). As a result, the prediction and filtering steps in Algorithm 4 will be modified and given by:

Prediction Sample particles from transition prior $\boldsymbol{x}_t^{(j)} \sim q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{y}_t) = p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^{(j)})$ by
- generating $\boldsymbol{\eta}_t^{(j)}$ according to the state–noise density (2.1)
- setting $\boldsymbol{x}_t^{(j)} = f(\boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{\eta}_t^{(j)})$

Filtering Assign to each particle $\boldsymbol{x}_t^{(j)}$ the importance weight $\boldsymbol{\omega}_t^{(j)}$ according to (2.41)

The previously described filter relies on sequential importance sampling. Sequential MC methods based on importance sampling exist since the fifties and late sixties as mentioned in Van der Merwe et al. (2000), but all these earlier Monte Carlo implementations based on SIS degenerate. In other words, the SIS filter is known to fail after a few iterations because only the weights appended to the particles are updated and as a result, many particles will have an almost zero contribution (negligible weights) to the final estimate. Indeed, Doucet, Godsill, and Andrieu (2000) state that for importance functions of the form (2.37), the variance of the importance weights increases (stochastically) over time. Doucet (1998) gives formal proof of the divergence of the Bayesian sequential importance sampling filter. To help to overcome the degeneracy drawback of the SIS algorithm, a resampling step is introduced and since then several versions of the sequential MC filters known as particle filters have emerge.

Generally speaking, different particle filter variants arise to deal with some inherent drawbacks encountered under the particle filtering methodology. For instance, some PF variants aim to overcome the inherent PF degeneracy problem or to overcome the collapse of the particles. Clearly, the main motivation of all authors, also ours, is to contribute to the improvement of the performance of existing particle filters so they can efficiently tackle a broader variety of models.

In the following section, as done with all previously studied filters, we present some particle filter variants scattered in the literature; we unify notation for the sake of consistency and readability. For a detailed monographic review on particle filters, see Doucet et al. (2001) and the references therein. Also, Arulampalam et al. (2002), and Doucet and Johansen (2011) provide a tutorial on some PF variants.

2.4.2 Sequential Importance Sampling with Resampling

As aforementioned, all sequential MC implementations based solely on sequential importance sampling (SIS) degenerate, and therefore are prone to diverge as t increases. The first known effective attempt aiming to surmount these drawbacks consists of adding a resampling step to the SIS filter with pseudo-code in Algorithm 4. This modification gives rise to a filter generally known as sequential importance sampling with resampling (SISR). Some issues regarding the resampling step are presented below.

The Resampling Step

At a *fixed* time t, the resampling step consists of generating samples, say M, by sampling with replacement from $\boldsymbol{x}_t^{(j)}$ particles, according to sampling probabilities given by the corresponding importance weights, $\boldsymbol{\omega}_t^{(j)}$. That is, aiming to approximate $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$, both particles and weights are updated by mapping a weighted measure to an unweighted measure, as illustrated in expression (2.42).

$$\text{At a } \textit{fixed} \text{ time } t\text{: } \{x_t^{(j)}, \boldsymbol{\omega}_t^{(j)}\} \longrightarrow \{\boldsymbol{x}_t^{j^*}, 1/M\},\ j = 1 \ldots M,\ j^* = 1 \ldots M. \tag{2.42}$$

For the sake of simplicity, we assume that the number of resampled particles, say R, equates the original number of particles M. However, when needed, one might resample a greater amount of particles, say $\{\boldsymbol{x}_t^{j^*}\}$, $j^* = 1 \ldots R$ with $R >> M$.

The addition of the resampling step to the SIS filter gives rise to the first operational algorithm within the particle filtering methodology, named the sequential importance sampling with resampling (SISR) PF variant. Thus, with the aim to prevent the degeneracy drawback, a resampling step is added, and in this way the resulting filter concentrates on particles with large weights and eliminates particles with negligible weights. Under the SISR PF variant, both, the particles and the weights are updated, and not only the weights as happens with the non-operational SIS filter. The pseudo-code for the generic form of a SISR particle filter variant is given in Algorithm 5. Figure 2.3 illustrates the SISR filter behavior for $M = 10$ particles, when resampling takes place at every time step.

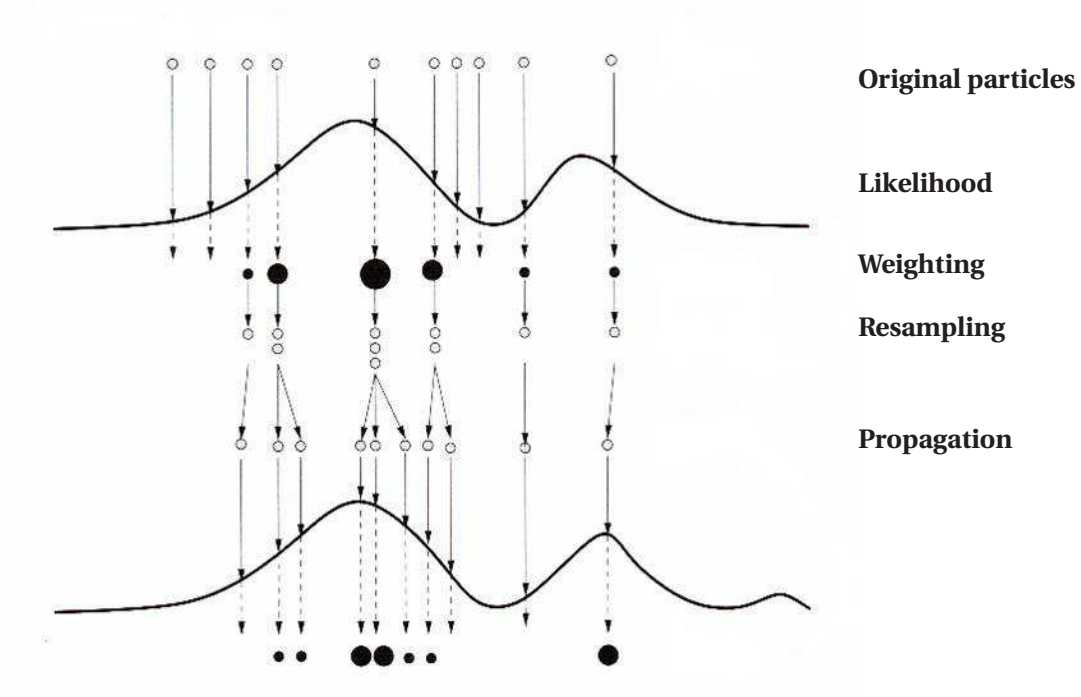


Figure 2.3: Illustration of SIS filter with resampling. Figure from Doucet et al. (2001)

Algorithm 5 SISR PF

Initialization $t = 0$
  for $j = 1$ to M **do**
    Sample $\boldsymbol{x}_0^{(j)} \sim p(\boldsymbol{x}_0)$
  end for
  for $t = 1$ to N **do**
Step 1 Importance sampling step
    for $j = 1$ to M **do**
      Prediction Sample $\boldsymbol{x}_t^{(j)} \sim q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{y}_t)$
      Filtering Assign to each particle $\boldsymbol{x}_t^{(j)}$ the importance weight $\boldsymbol{\omega}_t^{(j)}$ according to (2.38)
    end for
    for $j = 1$ to M **do**
      Normalize the importance weights: $\tilde{\boldsymbol{\omega}}_t^{(j)} = \frac{\boldsymbol{\omega}_t^{(j)}}{\sum_{i=1}^{M} \boldsymbol{\omega}_t^{(i)}}$
    end for
Step 2 Resampling step
    Resample with replacement the particles $\boldsymbol{x}_t^{(1)}, \ldots, \boldsymbol{x}_t^{(M)}$ according to importance weights $\{\widetilde{\boldsymbol{\omega}}_t^{(1)}, \ldots, \widetilde{\boldsymbol{\omega}}_t^{(M)}\}$
  end for

Notice that if resampling is applied at every time step, as portrayed in Figure 2.3, at time $t-1$ the weights $\boldsymbol{\omega}_{t-1}^{(j)}$ are all equal to $1/M$. Therefore, the importance weights in (2.38) would become

$$\boldsymbol{\omega}_t^{(j)} \propto \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t^{(j)})\, p(\boldsymbol{x}_t^{(j)}|\boldsymbol{x}_{t-1}^{(j)})}{q(\boldsymbol{x}_t^{(j)}|\boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{y}_t)}. \tag{2.43}$$

If in addition, the transition prior is used as a proposal, the importance weights (2.43)are further simplified as

$$\boldsymbol{\omega}_t^{(j)} \propto \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t^{(j)})\, p(\boldsymbol{x}_t^{(j)}|\boldsymbol{x}_{t-1}^{(j)})}{p(\boldsymbol{x}_t^{(j)}|\boldsymbol{x}_{t-1}^{(j)})} = p(\boldsymbol{y}_t|\boldsymbol{x}_t^{(j)}). \tag{2.44}$$

An algorithm that resamples at every time step and uses the transition prior as a proposal is known as sampling importance resampling (SIR) PF; being clearly a variant of the SISR particle filter. For instance, the papers of Gordon et al. (1993) and Kitagawa (1996) are examples of this PF variant. In fact, the SIR PF variant has been independently proposed and used by various authors and in different fields. For instance, in computer vision area it is called condensation algorithm and in probabilistic networks it is known as survival of the fittest; see MacCormick and Blake (1999) and Kanazawa, Koller, and Russel (1995), respectively. In this work, we introduce the sampling importance resampling (SIR) PF variant as described by Kitagawa (1996), since our incursion to the simulation based algorithms started with that publication.

2.4.3 The Sampling Importance Resampling Particle Filter

The authors Gordon et al. (1993) and Kitagawa (1996) proposed the addition of a resampling step to the SIS filter that uses the transition prior as a proposal PDF. The inclusion of this resampling step gives raise to the first effective operational particle filter, which is itself based on the weighted bootstrap method presented in Smith and Gelfand (1992). Gordon et al. (1993) named this filter 'bootstrap filter'. The SIR filter aims to prevent the inherent degeneracy drawback of particle filters.

The pseudo-code for the SIR particle filter variant, as described in Kitagawa's 1996 article, is given in Algorithm 6.

Algorithm 6 SIR PF

Initialization $t = 0$
  for $j = 1$ to M **do**
    Sample $\boldsymbol{x}_0^{(j)} \sim p(\boldsymbol{x}_0)$
  end for
  for $t = 1$ to N **do**
Step 1 Importance sampling step
    for $j = 1$ to M **do**
      Prediction Sample $\boldsymbol{x}_t^{(j)} \sim q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{y}_t) = p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^{(j)})$ by
        • generating $\boldsymbol{\eta}_t^{(j)}$ according to the state-noise density (2.1)
        • setting $\boldsymbol{x}_t^{(j)} = f(\boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{\eta}_t^{(j)})$
      Filtering: Assign to each particle $\boldsymbol{x}_t^{(j)}$ the weight $\boldsymbol{\omega}_t^{(j)}$ according to (2.44)
    end for
    for $j = 1$ to M **do**
      Normalize the importance weights: $\tilde{\boldsymbol{\omega}}_t^{(j)} = \frac{\boldsymbol{\omega}_t^{(j)}}{\sum_{i=1}^{M} \boldsymbol{\omega}_t^{(i)}}$
    end for
Step 2 Resampling step
    Resample with replacement the particles $\boldsymbol{x}_t^{(1)}, \ldots, \boldsymbol{x}_t^{(M)}$ according to importance weights $\{\tilde{\boldsymbol{\omega}}_t^{(1)}, \ldots, \tilde{\boldsymbol{\omega}}_t^{(M)}\}$
  end for

Although the resampling step is crucial to tackle the degeneracy problem, it may lead to other problems. For instance, the resampling step is considered a bottleneck for the opportunity to parallelize particle filters. Another potential problem is the so-called sample impoverishment or sample attrition. Intuitively, frequent resampling can cause loss of diversity in the samples and thus the variance of the current estimates can increase, leading to a reduced accuracy. Hence resampling must be applied with caution because, on the one hand, it helps on concentrating particles into domains of higher posterior probability, but on the other hand, it may lead to a degradation of the support of the particles (sample attrition), see Fearnhead (1998)[5].

[5] When smoothing is a target, any smoothed–estimate based on impoverished particles' paths can certainly lead to degeneracy

In his PhD thesis, Fearnhead (1998) provides guidelines, under importance resampling, questioning whether resampling is necessary at every time step. For instance, this author states that resampling can be adopted every k time steps, where the value of k must be decided beforehand by the researcher.

Another alternative is to use a measure of the accuracy of particle filters called effective sample size (ESS). This measure estimates how many particles from the target posterior PDF are necessary in order to get an accurate estimate of the (functions of) states. Thus, a particle filter estimator based on $\widehat{\text{ESS}} = M_{eff}$ particles is considered to be appropriate. To get an estimate of the ESS, a formula based on the particles weights is used (Liu and Chen 1998).

$$\widehat{\text{ESS}} = M_{eff} = \Big(\sum_{j=1}^{M} (\widetilde{\boldsymbol{\omega}}_t^{(j)})^2 \Big)^{-1}. \tag{2.45}$$

Intuitively, the estimated ESS takes possible values spanning from 1 to M. Moreover, extreme results correspond to two cases: if all particles have equal weights $1/M$ then $\widehat{\text{ESS}} = M_{eff} = M$, whereas if all but one of the particles have negligible weights $\widehat{\text{ESS}} = M_{eff} = 1$.

The decision as to whether to resample or not is taken by comparing the estimated $\widehat{\text{ESS}} = M_{eff}$ with a threshold value M_{thr} chosen arbitrarily by the researcher. The rule specifies that resampling is needed when $\widehat{\text{ESS}} = M_{eff}$ is below the chosen threshold value M_{thr}. The present work adopts to resample at every time instant, unless stated otherwise.

As can be inferred, the implementation of the resampling step is an important issue when using the particle filtering methodology. The implementation of the resampling step can be performed in various ways. Some authors use a multinomial resampling strategy; see for instance Ripley (1987) and Gordon et al. (1993). In this work, we implement the stratified resampling strategy suggested in Kitagawa (1996). Additionally, the widely-used residual resampling strategy is considered for comparison reasons (Higuchi 1997; Bergman 1999).

Resampling Implementation Strategies

Again, suppose that at *fixed* time t, the filtering random variable $\boldsymbol{x}_t|\boldsymbol{y}_{1:t}$ is approximated by a sufficiently large set of particles $\boldsymbol{x}_t^{(1)},\ldots,\boldsymbol{x}_t^{(M)}$ with discrete probability mass of $\boldsymbol{\omega}_t^{(1)},\ldots,\boldsymbol{\omega}_t^{(M)}$. As aforementioned, the task of the resampling step consists in approximating $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$ by mapping a weighted measure to an un-weighted measure as illustrated in expression (2.42) and Figure 2.3. This mapping task is achieved by obtaining an empirical distribution function that mimics the target distribution of the particles. Kitagawa (1996) points out that random resampling is thus not essential.

This author describes a basic algorithm for resampling-implementation, and states that several modifications are possible depending on the way sorting and random number generation is accomplished. Following we present an algorithm for the basic random resampling strategy.

As aforementioned, the addition of the resampling step introduces extra variability and for that reason the use of a variance reduction resampling scheme, such as the stratified or residual resam-

Algorithm 7 Basic Random Resampling

Step 1 Initial random measure step

Assume we have arranged particles $\boldsymbol{x}_t^{(1)},\ldots,\boldsymbol{x}_t^{(M)}$ with corresponding weights $\boldsymbol{\omega}_t^{(1)},\ldots,\boldsymbol{\omega}_t^{(M)}$

Step 2

for $j = 1$ to M **do**

(2.1) Generate a uniform random number $u_t^{(j)} \sim U[0,1]$

(2.2) Find position j^t such that $c^{(j^t-1)} = \frac{\sum_{k=1}^{j^t-1}\boldsymbol{\omega}_t^{(j)}}{C} < u_t^{(j)} \leq \frac{\sum_{k=1}^{j^t}\boldsymbol{\omega}_t^{(j)}}{C} = c^{(j^t)}$ with $C = \sum_{k=1}^{M}\boldsymbol{\omega}_t^{(k)}$

(2.3) Define the j^{th} filtering particle as $\boldsymbol{x}_t^{(j^t)}$.

end for

pling strategies, is very favorable. The *stratified* resampling results after modifying the previous basic random resampling algorithm, as follows.

Stratified Resampling

Carpenter, Clifford, and Fearnhead (1999) justify the use of stratification via sampling ideas. Stratified resampling consists in dividing the interval $[0,1)$ in M subintervals with identical width and a unique realization is drawn from a group of particles with total weight $1/M$.

Two special cases (systematic or deterministic) of stratified resampling arise by modifying the uniform random number generation Step (2.1) in Algorithm 7. Thus, the systematic and deterministic resampling strategies are gotten as follows

$$(2.1-S) \quad u_t^{(j)} \sim U[\frac{j-1}{M}, \frac{j}{M}), \quad j = 1\ldots M \tag{2.46}$$

$$(2.1-D) \quad u_t^{(j)} = \frac{j-u_\alpha}{M} \text{ for fixed } u_\alpha \in [0,1). \tag{2.47}$$

Kitagawa (1996) performed a comparative study among the resampling schemes mentioned, and concludes that the stratified resampling scheme is an efficient method, specifically the deterministic version of it. By efficient, one means that the method is easy to implement and that reduces the Monte Carlo variation. Specifically, he concludes that the deterministic $(i-D)$ and the systematic $(i.S)$ methods outperformed the random one and that within $(i-D)$ and $(i-S)$, the former is better.

An illustration of the resampling step using the deterministic stratified resampling scheme is provided in Figure 2.4, which is inspired in Bolic (2004). Therein, $M = 5$ particles are used and $C^{(i)}$ denotes the cumulative sum for the i_{th} particle. Thus, once the resampling step has been applied, the first particle is sampled twice, whereas the second, third and and fifth particle only once. Note that in this case, the fourth particle is not chosen at all.

The stratified algorithm is implemented by us in the R programming language, following descriptions in Kitagawa (1996).

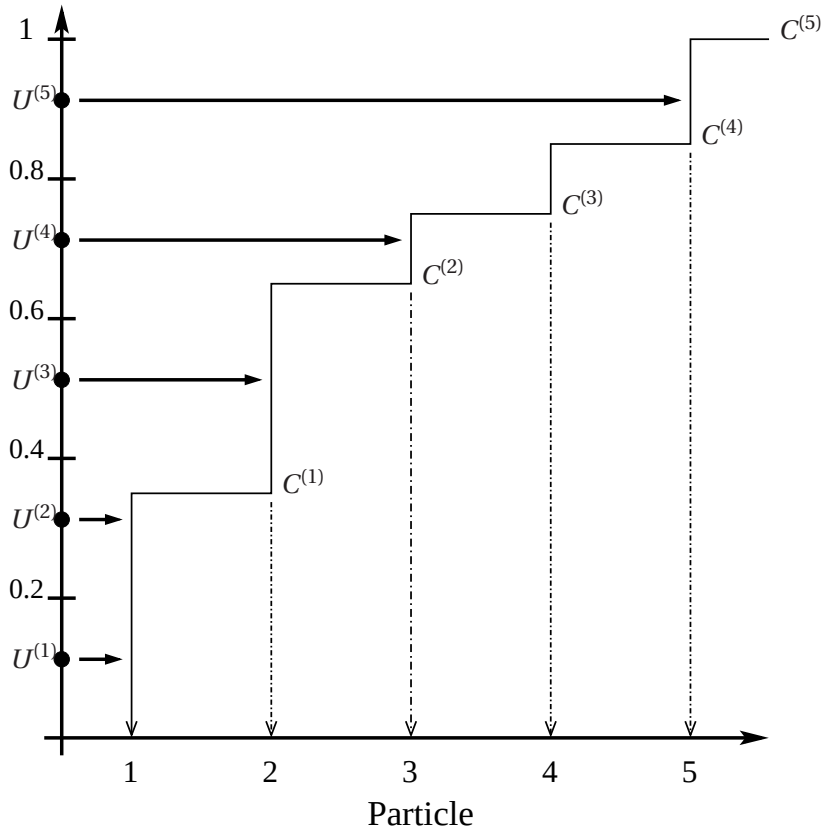


Figure 2.4: An illustration of stratified deterministic resampling; inspired in Bolic (2004)

Residual Resampling

This widely-used resampling strategy is based on set restriction rather than on sampling methods (Higuchi 1997; Bergman 1999; Liu and Chen 1998).

Under this method, a large part of the number of copies M_j of particle j is obtained by taking the integer part of $M \cdot \boldsymbol{\omega}_j$; i.e. without using random numbers. Since it is not guaranteed that the total number of resampled particles $\sum_{j=1}^{M} M_j$ is M, a residual number of particles M_r must be randomly selected with replacement according to normalized residual weights $\boldsymbol{\omega}_{res}$. These ideas are summarized in the following algorithm:

Step 1 Initial random measure step

Assume we have arranged particles $\boldsymbol{x}_t^{(1)}, \ldots, \boldsymbol{x}_t^{(M)}$ with normalized weights $\boldsymbol{\omega}_t^{(1)}, \ldots, \boldsymbol{\omega}_t^{(M)}$

Step 2 Obtain Mj copies of particle $\boldsymbol{x}_j$ using: $M_j = \llcorner M\boldsymbol{\omega}_j \lrcorner$, where $\llcorner z \lrcorner$ denotes the integer part of z.

Step 3 Compute the residual number of particles to sample by $M_{res} = M - \sum_{j=1}^{M} M_j$

Step 4 Resample M_{res} particles from $\{\boldsymbol{x}_j\}$ using weights $\boldsymbol{\omega}_{res} \propto M\boldsymbol{\omega}_j - M_j$ using another resampling scheme (we use the stratified strategy).

Step 5 Copy the resampled trajectories.

This residual algorithm is written by us in the R programming language following existing MATLAB instructions (reproduced in Appendix A). The original MATLAB instructions of Doucet and de Freitas for implementing the residual resampling algorithm of Liu and Chen (1998) can also be found in `http://vismod.media.mit.edu/pub/yuanqi/mcep/residualR.m` [last visited: September 2013].

Since different PF variants arise by incorporating new features aiming to improve upon existing particle filters, following we study other three PF variants which are distinguished by the use of different proposal distributions. All of them are optimal in the sense that they all use the information provided by the last observation. These particle filter variants are the auxiliary particle filter (ASIR, the extended particle filter (EPF) and the unscented particle filter (UPF).

The following PF variant, auxiliary particle filter (ASIR), aims to improve upon the SIR particle filter variant by using a "better" proposal PDF, that is by using a proposal PDF which includes the information provided by the last observation.

2.4.4 The Auxiliary Particle Filter

Pitt and Shephard (1999) extend the SIR particle filter that had recently been suggested independently by various authors. In order to improve upon the simulation design of the SIR particle filter variant, these authors propose the auxiliary sampling importance resampling (ASIR) PF variant. A distinguishing feature of the ASIR particle filter is that it appends an auxiliary variable (k) to the state vector $\boldsymbol{x}_t$ and thus samples from a higher dimensional state vector, $(\boldsymbol{x}_t, k)$. The auxiliary variable k is introduced to aid the task of simulation. Thus, instead of generating samples from the filtering PDF $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$ as done by Algorithm 6, the auxiliary sampling importance resampling PF variant generates samples from the joint PDF $p(\boldsymbol{x}_t, k|\boldsymbol{y}_{1:t})$. It can be verified by the Bayes rule that

$$p(\boldsymbol{x}_t, k|\boldsymbol{y}_{1:t}) \propto p(\boldsymbol{y}_t|\boldsymbol{x}_t)\, p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^k)\boldsymbol{\omega}_{t-1}^k,\ k = 1,\ldots,M. \tag{2.48}$$

Pitt and Shephard aim to design a proposal distribution $q(\boldsymbol{x}_t, k|\boldsymbol{y}_{1:t})$ that allows sampling (indirectly) from the joint PDF $p(\boldsymbol{x}_t, k|\boldsymbol{y}_{1:t})$. The ASIR method is said to be adaptable and extremely flexible, because one has complete control over the design of the (optimal) proposal PDF, say q, which can depend on $\boldsymbol{y}_t$ and $\boldsymbol{x}_{t-1}^k$. Further, these authors suggest a generic procedure for the choice of a proposal PDF q. This procedure consists in approximating $p(\boldsymbol{x}_t, k|\boldsymbol{y}_{1:t})$ by sampling from an importance PDF of the form

$$q(\boldsymbol{x}_t, k|\boldsymbol{y}_{1:t}) \propto p(\boldsymbol{y}_t|\mu_t^k)\, p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^k)\, \boldsymbol{\omega}_{t-1}^k,\ k = 1,\ldots,M, \tag{2.49}$$

where μ_t^k denotes a characteristic associated to the density of $\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^k$. This characteristic could be the mean $\mathrm{E}(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^k)$, the mode, a draw, or some other likely value of the state $\boldsymbol{x}_t$ given $\boldsymbol{x}_{t-1}^k$. The form of the approximating density is designed so that

$$q(k|\boldsymbol{y}_{1:t}) \propto \boldsymbol{\omega}_{t-1}^k \int p(\boldsymbol{y}_t|\mu_t^k)\, p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^k d\boldsymbol{x}_t) = \boldsymbol{\omega}_{t-1}^k p(\boldsymbol{y}_t|\mu_t^k). \tag{2.50}$$

Thus, if $q(\boldsymbol{x}_t|k, \boldsymbol{y}_{1:t})$ is defined as $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^k)$, the importance PDF (2.49) can be rewitten as the product

$$q(\boldsymbol{x}_t, k|\boldsymbol{y}_{1:t}) \propto q(k|\boldsymbol{y}_{1:t})\, q(\boldsymbol{x}_t|k, \boldsymbol{y}_{1:t}). \tag{2.51}$$

In conclusion, in order to sample from $p(\boldsymbol{x}_t, k|\boldsymbol{y}_{1:t})$, we sample from the alternative importance PDF $q(\boldsymbol{x}_t, k|\boldsymbol{y}_{1:t})$. This is done by first simulating the index k with probability $\boldsymbol{\lambda}_k$ proportional to $q(k|\boldsymbol{y}_{1:t}) = \boldsymbol{\omega}_{t-1}^k p(\boldsymbol{y}_t|\mu_t^k)$ in (2.50) and then discarding the index k sample $\boldsymbol{x}_t^{(j)}$ from the marginalized PDF $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$. That is, once the index k is sampled, we sample $\boldsymbol{x}_t^{(j)}$ just in the same way as done by SIR. The weights $\boldsymbol{\lambda}_k$ are called the first-stage weights. The so-called second stage weights, say $\boldsymbol{\omega}_t^{(j)}$, are given as the ratio of equations (2.48) and (2.49); that is

$$\boldsymbol{\omega}_t^{(j)} \propto \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t^{k^{(j)}})}{p(\boldsymbol{y}_t|\mu_t^{k^{(j)}})} \tag{2.52}$$

The following pseudo-code (Algorithm 8) summarizes the ASIR PF algorithm.

Algorithm 8 ASIR PF

Initialization $t = 0$
  for $k = 1$ to M **do**
    Sample $\boldsymbol{x}_0^{(k)} \sim p(\boldsymbol{x}_0)$
  end for
  for $t = 1$ to N **do**
Step 1 Auxiliary variable resampling step
    for $k = 1$ to M **do**
      Select and calculate $\mu_t^{(k)}$ associated to the conditional PDF of $(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^{(k)})$
      Calculate the first stage weights $\boldsymbol{\lambda}_t^{(k)} = q(k|\boldsymbol{y}_{1:t})$ in (2.50)
    end for
    for $k = 1$ to M **do**
      Normalize the first stage weights $\tilde{\boldsymbol{\lambda}}_t^{(k)} = \frac{\boldsymbol{\lambda}_t^{(k)}}{\sum_{i=1}^M \boldsymbol{\lambda}_t^{(k)}}$
    end for
    Sample with replacement the index $k_{j=1}^M$ according to the computed first stage weights.
Step 2 Importance sampling step
    for $j = 1$ to M **do**
      Sample $\boldsymbol{x}_t^j \sim q(\boldsymbol{x}_t|k^{(j)}, \boldsymbol{y}_{1:t}) = p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^{k^{(j)}})$ as in the SIR filter, that is
      Calculate the second stage weights $\boldsymbol{\omega}_t^{(j)}$ using (2.52)
    end for
    for $k = 1$ to M **do**
      Normalize the second stage weights $\tilde{\boldsymbol{\omega}}_t^{(k)} = \frac{\boldsymbol{\omega}_t^{(k)}}{\sum_{k=1}^M \boldsymbol{\omega}_t^{(k)}}$
    end for
Step 3 Second Resampling step
    Resample with replacement the particles $\boldsymbol{x}_t^{(1)}, \ldots, \boldsymbol{x}_t^{(M)}$ according to importance weights $\{\widetilde{\boldsymbol{\omega}}_t^{(1)}, \ldots, \widetilde{\boldsymbol{\omega}}_t^{(M)}\}$
  end for

In contrast to the SIR approach, under the ASIR approach, one simulates from particles that have high predictive likelihood. That is, making proposals with high conditional likelihood causes that particles with very low likelihoods will not be resampled at the second stage of the filter. Also, it is expected that the weights (2.52) under the ASIR are much less variable than the ones under the SIR PF variant.

Following, the so-called extended particle filter (EPF) variant is introduced. This PF variant aims to improve upon the SIR particle filter variant by using an "ideal" proposal PDF; that is done by using a Gaussian proposal PDF which includes the information provided by the last observation.

2.4.5 The Extended Particle Filter

The extended particle filter (EPF) is a hybrid method used for nonlinear online estimation. This particle filter variant combines a non-simulation based approach (the EKF) with the sequential Monte Carlo particle filter approach. The key feature of this filter is that a Gaussian EKF approximation is used as the proposal distribution for a particle filter (de Freitas, Niranjan, Gee, and Doucet 2000; Van der Merwe, Doucet, de Freitas, and Wan 2000). Note that since the proposal PDF which is used includes the information provided by the last observation, this Gaussian PDF is considered optimal.

Under the EPF, one samples the particles $\boldsymbol{x}_t^{(j)}$ from the Gaussian importance PDF obtained via the EKF. That is, $\boldsymbol{x}_t^{(j)} \sim q(\boldsymbol{x}_t^{(j)}|\boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{y}_{1:t}) \triangleq \mathcal{N}(\bar{\boldsymbol{x}}_t^{(j)^{EKF}}, \boldsymbol{\Sigma}_t^{(j)^{EKF}})$ and thus the corresponding expression for the importance weights in equation (2.38) is given by

$$\boldsymbol{\omega}_t^{(j)} \propto \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t^{(j)^{EKF}})\, p(\boldsymbol{x}_t^{(j)^{EKF}}|\boldsymbol{x}_{t-1}^{(j)})}{\mathcal{N}(\boldsymbol{x}_t^{(j)}; \bar{\boldsymbol{x}}_t^{(j)^{EKF}}, \boldsymbol{\Sigma}_t^{(j)^{EKF}})} \boldsymbol{\omega}_{t-1}^{(j)}, \tag{2.53}$$

and by

$$\boldsymbol{\omega}_t^{(j)} \propto \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t^{(j)^{EKF}})\, p(\boldsymbol{x}_t^{(j)^{EKF}}|\boldsymbol{x}_{t-1}^{(j)})}{\mathcal{N}(\boldsymbol{x}_t^{(j)}; \bar{\boldsymbol{x}}_t^{(j)^{EKF}}, \boldsymbol{\Sigma}_t^{(j)^{EKF}})}, \tag{2.54}$$

if resampling takes place at every time step (all $\boldsymbol{\omega}_{t-1}^{(j)} = 1/M$). The following pseudo-code (Algorithm 9) on page 39 summarizes the general EPF algorithm.

Algorithm 9 Extended PF (EPF)

Initialization $t = 0$

for $j = 1$ to M **do**

Sample $x_0^{(j)} \sim p(x_0)$ and fixed known parameters.

end for

for $t = 1$ to N **do**

Step 1 Importance sampling step

for $j = 1$ to M **do**

Step 2 Prediction step

Compute $J_{\boldsymbol{x}_{t-1}}$, and $J_{\boldsymbol{\eta}_t}$ as in equation (2.19).

Compute the predictive expectation $\bar{\boldsymbol{x}}_{t|t-1}^{(j)}$ and covariance $\boldsymbol{\Sigma}_{\boldsymbol{x}_{t|t-1}}^{(j)}$ using

$\bar{\boldsymbol{x}}_{t|t-1}^{(j)} = f(\boldsymbol{x}_{t-1}^{(j)}, 0)$ and

$\boldsymbol{\Sigma}_{\boldsymbol{x}_{t|t-1}}^{(j)} = J_{\boldsymbol{x}_{t-1}}^{(j)} \boldsymbol{\Sigma}_{\boldsymbol{x}_{t-1|t-1}}^{(j)} J_{\boldsymbol{x}_{t-1}}^{\prime(j)} + J_{\boldsymbol{\eta}_t}^{(j)} Q_t J_{\boldsymbol{\eta}_t}^{\prime(j)}$, respectively.

Step 3 Kalman Gain step

Compute $J_{\boldsymbol{x}_t}$, and $J_{\boldsymbol{v}_t}$ as in equation (2.20).

Compute the prediction estimate $\boldsymbol{y}_{t|t-1}^{(j)}$ and covariance $\boldsymbol{\Sigma}_{\boldsymbol{y}_{t|t-1}}^{(j)}$ using equations (2.26) and (2.27), respectively.

$\boldsymbol{y}_{t|t-1}^{(j)} = h_{t|t-1}^{(j)} = h(\bar{\boldsymbol{x}}_{t|t-1}^{(j)}, 0)$

$\boldsymbol{\Sigma}_{\boldsymbol{y}_{t|t-1}} = J_{\boldsymbol{x}_t}^{(j)} \boldsymbol{\Sigma}_{\boldsymbol{x}_{t|t-1}}^{(j)} J_{\boldsymbol{x}_t}^{\prime(j)} + J_{\boldsymbol{v}_t}^{(j)} R_t J_{\boldsymbol{v}_t}^{\prime(j)}$

Compute the Kalman Gain K_t with equation (2.28).

$K_t = \boldsymbol{\Sigma}_{\boldsymbol{x}_{t|t-1}}^{(j)} J_{\boldsymbol{x}_t}^{\prime(j)} \boldsymbol{\Sigma}_{\boldsymbol{y}_{t|t-1}}^{-1(j)}$

Step 4 Filtering step

Compute the filtering expectation $\bar{\boldsymbol{x}}_{t|t}$ and covariance $\boldsymbol{\Sigma}_{\boldsymbol{x}_{t|t}}$ using (2.29) and (2.30), respectively.

$\bar{\boldsymbol{x}}_t^{(j)^{EKF}} = \bar{\boldsymbol{x}}_{t|t-1}^{(j)} + K_t(\boldsymbol{y}_t - \boldsymbol{y}_{t|t-1}^{(j)})$

$\boldsymbol{\Sigma}_t^{(j)^{EKF}} = \boldsymbol{\Sigma}_{\boldsymbol{x}_{t|t-1}}^{(j)} - K_t J_{\boldsymbol{x}_t}^{(j)} \boldsymbol{\Sigma}_{\boldsymbol{x}_{t|t-1}}^{(j)}$

Sample $\boldsymbol{x}_t^{(j)} \sim q(\boldsymbol{x}_t^{(j)} | \boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{y}_{1:t}) \stackrel{\circ}{=} N(\bar{\boldsymbol{x}}_t^{(j)^{EKF}}, \boldsymbol{\Sigma}_t^{(j)^{EKF}})$

end for

for $j = 1$ to M **do**

Evaluate the importance weights up to a normalizing constant.

$\boldsymbol{\omega}_t^{(j)} \propto \frac{p(\boldsymbol{y}_t | \boldsymbol{x}_t^{(j)^{EKF}})\, p(\boldsymbol{x}_t^{(j)^{EKF}} | \boldsymbol{x}_{t-1}^{(j)})}{q(\boldsymbol{x}_t^{(j)} | \boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{y}_t)}$ using equation (2.53) or equation (2.54), as appropriate.

end for

for $j = 1$ to M **do**

Normalize the importance weights $\tilde{\boldsymbol{\omega}}_t^{(j)} = \frac{\boldsymbol{\omega}_t^{(j)}}{\sum_{i=1}^{M} \boldsymbol{\omega}_t^{(i)}}$.

end for

Step 5 Resample the discrete PDF to obtain a sample of size M.

Multiply/Supress particles $\boldsymbol{x}_t^{(j)^{\text{EKF}}}$ according to high/low importance weights, $\tilde{\boldsymbol{\omega}}_t^{(j)}$

end for

2.4.6 The Unscented Particle Filter

The unscented particle filter (UPF) is a novel method developed to tackle the nonlinear, non-Gaussian, on-line estimation problem; see Van der Merwe, Doucet, de Freitas, and Wan (2001) and Van der Merwe (2004). Similarly to the EPF, the UPF is a hybrid method which combines a non-simulation based approach (the UKF) with the sequential particle filter approach. In contrast to the EPF, the UPF uses the UKF Gaussian approximation as an optimal proposal distribution.

Thus, in this case, the particles $\boldsymbol{x}_t^{(j)}$ are sampled from the Gaussian importance PDF obtained via the UKF. That is, $\boldsymbol{x}_t^{(j)} \sim q(\boldsymbol{x}_t^{(j)}|\boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{y}_{1:t}) \triangleq N(\bar{\boldsymbol{x}}_t^{(j)^{UKF}}, \boldsymbol{\Sigma}_t^{(j)^{UKF}})$ and thus the corresponding expression for the importance weights in equation (2.38) is given by

$$\boldsymbol{\omega}_t^{(j)} \propto \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t^{(j)^{UKF}})p(\boldsymbol{x}_t^{(j)^{UKF}}|\boldsymbol{x}_{t-1}^{(j)})}{N(\boldsymbol{x}_t^{(j)}; \bar{\boldsymbol{x}}_t^{(j)^{UKF}}, \boldsymbol{\Sigma}_t^{(j)^{UKF}})}\boldsymbol{\omega}_{t-1}^{(j)}, \tag{2.55}$$

and by

$$\boldsymbol{\omega}_t^{(j)} \propto \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t^{(j)^{UKF}})p(\boldsymbol{x}_t^{(j)^{UKF}}|\boldsymbol{x}_{t-1}^{(j)})}{N(\boldsymbol{x}_t^{(j)}; \bar{\boldsymbol{x}}_t^{(j)^{UKF}}, \boldsymbol{\Sigma}_t^{(j)^{UKF}})}, \tag{2.56}$$

if resampling takes place at every time step (all $\boldsymbol{\omega}_{t-1}^{(j)} = 1/M$).

The pseudo-code in Algorithm 10 (see page 41) summarizes the general UPF algorithm.

The authors of the UPF, Van der Merwe et al. (2001), state that a particle filter with a proposal distribution obtained using the UKF is able to outperform not only the EKF and the UKF but also any other standard PF variants. As stated in Subsection 2.3.3, the UKF is able to provide better mean and covariance estimates than its counterpart EKF. Moreover, the UKF is able to generate heavier tailed distributions than the EKF, but it can struggle when dealing with general non-Gaussian distributions. A careful glance to the UPF's pseudo-code indicate that its implementation results relatively more complex.

Particularly, we believe that a reason for the mentioned superior performance (Chapter 4 will corroborate this) of the UPF is that it is a hybrid approach obtained by combining the non-simulation based UKF algorithm and the particle filtering methodology. The accuracy attained by the UKF together with the lack of distributional assumptions of the particle filters make of the UPF a worthy alternative to deal with more complex models which are highly nonlinear or largely departing from normality.

Algorithm 10 Unscented PF (UPF)

Initialization $t = 0$

Set parameter values: α, β and κ.

Compute the dimension of the augmented state: $n_a = n_x + n_\eta + n_v$

for $j = 1$ to M **do**

Sample $x_0^{(j)} \sim p(x_0)$ and set:

$\bar{\boldsymbol{x}}_0^{(j)} = \mathrm{E}(\boldsymbol{x_0}^{(j)})$

$\Sigma_0^{(j)} = E\left((\boldsymbol{x_0}^{(j)} - \bar{\boldsymbol{x}}_0^{(j)})(\boldsymbol{x_0}^{(j)} - \bar{\boldsymbol{x}}_0^{(j)})'\right)$

$\bar{\boldsymbol{x}}_0^{(j)a} = \mathrm{E}(\boldsymbol{x}_0^{(j)a}) = ((\bar{\boldsymbol{x}}_0^{(j)})', \mathbf{0}, \mathbf{0})'$

$\Sigma_0^{(j)a} = E\left((\boldsymbol{x}_0^{(j)a} - \bar{\boldsymbol{x}}_0^{(j)a})(\boldsymbol{x}_0^{(j)a} - \bar{\boldsymbol{x}}_0^{(j)a})'\right)$ and fixed known parameters.

end for

for $t = 1$ to N **do**

Step 1 Importance sampling step

for $j = 1$ to M **do**

Step 2 Update the particles with the UKF

Compute the sigma points

$$\boldsymbol{\chi}_{t-1}^{(j)a} = [\bar{\boldsymbol{x}}_{t-1}^{(j)a}, \bar{\boldsymbol{x}}_{t-1}^{(j)a} \pm \left(\sqrt{(n_a + \lambda)\Sigma_{t-1}^{(j)a}}\right)]$$

Time update (propagate particles into the future)

$$\boldsymbol{\chi}_{t|t-1}^{(j)x} = f(\boldsymbol{\chi}_{t-1}^{(j)x}, \boldsymbol{\chi}_{t-1}^{(j)\eta}) \qquad \bar{\boldsymbol{x}}_{t|t-1}^{(j)} = \sum_{i=0}^{2n_a} \boldsymbol{\omega}_i^{(m)} \boldsymbol{\chi}_{i,t|t-1}^{(j)x}$$

$$\Sigma_{t|t-1}^{(j)} = \sum_{i=0}^{2n_a} \boldsymbol{\omega}_i^{(c)} (\boldsymbol{\chi}_{i,t|t-1}^{(j)x} - \bar{\boldsymbol{x}}_{t|t-1}^{(j)})(\boldsymbol{\chi}_{i,t|t-1}^{(j)x} - \bar{\boldsymbol{x}}_{t|t-1}^{(j)})'$$

$$\boldsymbol{y}_{t|t-1}^{(j)} = h(\boldsymbol{\chi}_{t|t-1}^{(j)x}, \boldsymbol{\chi}_{t-1}^{(j)v}) \qquad \bar{\boldsymbol{y}}_{t|t-1}^{(j)} = \sum_{i=0}^{2n_a} \boldsymbol{\omega}_i^{(m)} \boldsymbol{y}_{i,t|t-1}^{(j)}$$

Measurement update (incorporate new observation)

$$\Sigma_{\tilde{\boldsymbol{y}}_t, \tilde{\boldsymbol{y}}_t} = \sum_{i=0}^{2n_a} \boldsymbol{\omega}_i^{(c)} (\boldsymbol{y}_{i,t|t-1}^{(j)} - \bar{\boldsymbol{y}}_{t|t-1}^{(j)})(\boldsymbol{y}_{i,t|t-1}^{(j)} - \bar{\boldsymbol{y}}_{t|t-1}^{(j)})'$$

$$\Sigma_{\boldsymbol{x}_t, \tilde{\boldsymbol{y}}_t} = \sum_{i=0}^{2n_a} \boldsymbol{\omega}_i^{(c)} (\boldsymbol{\chi}_{i,t|t-1}^{(j)x} - \bar{\boldsymbol{x}}_{t|t-1}^{(j)})(\boldsymbol{y}_{i,t|t-1}^{(j)} - \bar{\boldsymbol{y}}_{t|t-1}^{(j)})'$$

$$K_t = \Sigma_{\boldsymbol{x}_t, \tilde{\boldsymbol{y}}_t} \Sigma_{\tilde{\boldsymbol{y}}_t, \tilde{\boldsymbol{y}}_t}^{-1}$$

$$\bar{\boldsymbol{x}}_t^{(j)UKF} = \bar{\boldsymbol{x}}_{t|t-1}^{(j)} + K_t(\boldsymbol{y}_t - \bar{\boldsymbol{y}}_{t|t-1}^{(j)}) \quad \Sigma_t^{(j)UKF} = \Sigma_{t|t-1}^{(j)} - K_t \Sigma_{\tilde{\boldsymbol{y}}_t, \tilde{\boldsymbol{y}}_t} K_t'$$

Sample $\boldsymbol{x}_t^{(j)} \sim q(\boldsymbol{x}_t^{(j)} | \boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{y}_{1:t}) \stackrel{\circ}{=} N(\bar{\boldsymbol{x}}_t^{(j)UKF}, \boldsymbol{\Sigma}_t^{(j)UKF})$

end for

for $j = 1$ to M **do**

Evaluate the importance weights up to a normalizing constant.

$\boldsymbol{\omega}_t^{(j)} \propto \frac{p(\boldsymbol{y}_t | \boldsymbol{x}_t^{(j)UKF})\, p(\boldsymbol{x}_t^{(j)UKF} | \boldsymbol{x}_{t-1}^{(j)})}{q(\boldsymbol{x}_t^{(j)} | \boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{y}_t)}$ using equation (2.53) or equation (2.54), as appropriate.

end for

for $j = 1$ to M **do**

Normalize the importance weights $\tilde{\boldsymbol{\omega}}_t^{(j)} = \frac{\boldsymbol{\omega}_t^{(j)}}{\sum_{i=1}^{M} \boldsymbol{\omega}_t^{(i)}}$.

end for

Step 3 Resample the discrete PDF to obtain a sample of size M.

Multiply/Supress particles $\boldsymbol{x}_t^{(j)\mathrm{UKF}}$ according to high/low importance weights, $\tilde{\boldsymbol{\omega}}_t^{(j)}$

end for

2.4.7 Final Remarks

In summary, the present chapter dealt with the theory of optimal filtering, that is, optimal latent-state estimation, specially in a non-standard framework. Herein, a survey of different nonlinear filters is provided, including the gold standard filter for linear Gaussian models: the Kalman filter. The level of detail in the description of all the algorithms is chosen having mostly a practitioner researcher in mind, but aiming to be meticulous enough for all users. Indeed we make a comprehensive coverage of all the filters that are later used within several Monte Carlo experiments carried out in this work.

As mentioned in Section 2.4, the particle filtering methodology is more flexible than the previously described EKF and UKF filters. Particularly, the particle filtering methodology does not impose linearity nor distributional restrictions, which makes it more suitable for the estimation of non-standard dynamic state-space models.

We end this chapter by providing Tables 2.1–2.3. Table 2.1 summarizes the main features of three non simulation-based filters (KF, EKF and UKF), which have been used to improve upon existing particle filters. In Table 2.2, we provide a summary of the main features related to the historical evolution of the particle filter variants that have been fully described in this chapter: the SISR, SIR, ASIR, EPF and UPF. Later on, when implementing these filters, some of them will be modified as needed. For instance, a SIR particle filter variant that uses a fully adapted proposal distribution will be called by us: optimal sampling importance resampling (SIRopt). Similarly, the EPF nonlinear filter (that uses the EKF) will be modified to deal with a linear and Gaussian dynamic model; that is, by using the KF in place of the EKF. The resulting filter will be called Kalman particle filter (KPF); being clearly a special case of the EPF particle filter variant. All these filters will

Table 2.3 (also presented in Acosta and Muñoz (2007)) displays a summary of the form taken by the importance weights under the different particle filter variants studied. The form of these weights vary according to the adopted proposal or importance PDF $q(\boldsymbol{x}_t^{(j)}|\boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{y}_t)$. Be reminded that the generic form of the importance weights is $\omega_t^{(j)} \propto \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t^{(j)})p(\boldsymbol{x}_t^{(j)}|\boldsymbol{x}_{t-1}^{(j)})}{q(\boldsymbol{x}_t^{(j)}|\boldsymbol{x}_{t-1}^{(j)},\boldsymbol{y}_t)}\omega_{t-1}^{(j)}$ as given in equation (2.38).

The following two chapters (Chapter 3 and 4)are devoted to filtering dynamic models in a linear Gaussian context and in a nonlinear non-Gaussian framework, respectively. On one hand, Chapter 3 illustrates the performance of a subset (or special cases) of the described algorithms – the KF, SIR, SIRopt[6], ASIR, and KPF[7] – in a linear Gaussian context. On the other hand, Chapter 4 illustrates the performance of all described algorithms used in a nonlinear non-Gaussian context; say the EKF, UKF, SIR, ASIR, EPF and UPF filters. In subsequent chapters, some of these algorithms are further modified to deal with the simultaneous estimation of states and model parameters.

[6] the fully adapted sampling importance resampling (SIRopt) uses the optimal proposal distribution of the form $q(\boldsymbol{x}_t^{(j)}|\boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{y}_t) = p(\boldsymbol{x}_t^{(j)}|\boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{y}_t)$ as mentioned by Doucet (1998)

[7] being in the linear Gaussian context, we call this filter KPF instead of EPF because it uses the normal distribution provided by the KF as the optimal proposal distribution. The EPF uses, instead, the normal distribution obtained by the EKF

Table 2.1: Historical evolution of the studied non-simulation based filters that tackle solely the estimation of the states.

Filter	Authors (year)	Stylized Features[a]	Other Remarks
KF	Kalman (1960); Kalman and Bucy (1961)	Provides an optimal and closed form solution in a linear and Gaussian context	Linear filter
EKF	Wishner, Tabaczynski, and Athans (1969) Anderson and Moore (1979)	Approximates the the two nonlinear functions involved in the state-space model. Uses a first order Taylor series expansion to linearize such nonlinear functions.	Nonlinear filter
UKF	Wan and Van der Merwe (2000, 2001) Julier (2002); Julier and Uhlmann (1997, 2004)	Approximates the filtering distribution of the states through the specified nonlinear model. Uses a minimal set of deterministically chosen (weighted) sigma points that are propagated (in time) through the true nonlinearity, and are then used to produce the Gaussian approximation to the filtering PDF $p(\boldsymbol{x}\|\boldsymbol{y}_{1:t})$.	Nonlinear Filter

[a] All these filters approximate the posterior distribution of the states by a Gaussian distribution

Table 2.2: Historical evolution of the studied simulation based filters that tackle solely the estimation of the states.

Particle Filter	Authors (year)	Resampling Scheme	When to resample?	Stylized Features	Other Remarks
SISR	Version of: Gordon et al. (1993)	Multinomial	At every time step	Uses the transition prior as a proposal PDF.	First operational particle filter; also known as Bootstrap filter.
	Version of: Kitagawa (1996)	Multinomial, Stratified	At every time step	Uses the transition prior as a proposal PDF Named in this work: **SIR PF**[a]	The author concludes that the stratified scheme is more efficient (easy to implement and reduces the Monte Carlo variation).
ASIR	Pitt and Shephard (1999)	Residual	At every time step	Appends an auxiliary index-variable k to the state vector $\boldsymbol{x}_t$ and thus samples from a higher dimensional state vector $(\boldsymbol{x}_t, k)$. Uses a function of the past states μ_t^k, being $E(x_t\|x_{t-1})$ the most commonly used, to pre-select more likely state-particles.	The authors mention that the second resampling step could be optional
EPF	de Freitas et al. (2000)	Residual	At every time step	Uses the normal distribution obtained via the EKF as a proposal PDF.	Named by us KPF when the KF is used as a proposal PDF.
UPF	Van der Merwe et al. (2001)	Residual	At every time step	Uses the normal distribution obtained via the UKF as a proposal PDF.	

[a] The SIRopt PF variant arrives when instead of the transition prior, a fully adapted proposal (as mentioned by Doucet (1998)) is used

Table 2.3: Form of the importance weights varying according to the adopted proposal or importance PDF $\boldsymbol{q(x_t^{(j)}|x_{t-1}^{(j)}, y_t)}$ under each filter studied. The generic form of the importance weights given by $\boldsymbol{\omega}_t^{(j)} \propto \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t^{(j)})p(\boldsymbol{x}_t^{(j)}|\boldsymbol{x}_{t-1}^{(j)})}{q(\boldsymbol{x}_t^{(j)}|\boldsymbol{x}_{t-1}^{(j)},\boldsymbol{y}_t)}\boldsymbol{\omega}_{t-1}^{(j)}$ in equation (2.38)

Filter	**Imp. PDF** $\boldsymbol{q(x_t^{(j)}|x_{t-1}^{(j)}, y_t)}$	**Imp. weights form** $\boldsymbol{\omega}_t^{(j)} \propto$	**Some Remarks**
SISR PF	$p(\boldsymbol{x}_t^{(j)}\|\boldsymbol{x}_{t-1}^{(j)})$	$p(\boldsymbol{y}_t\|\boldsymbol{x}_t^{(j)})\boldsymbol{\omega}_{t-1}^{(j)}$ See equation (2.41)	These weights are reduced to the expression in equation (2.44) when resampling takes place at every time step.
EPF	$\mathcal{N}(\bar{\boldsymbol{x}}_t^{(j)^{EKF}}, \boldsymbol{\Sigma}_t^{(j)^{EKF}})$	$\frac{p(\boldsymbol{y}_t\|\boldsymbol{x}_t^{(j)^{EKF}})p(\boldsymbol{x}_t^{(j)^{EKF}}\|\boldsymbol{x}_{t-1}^{(j)})}{\mathcal{N}(\bar{\boldsymbol{x}}_t^{(j)^{EKF}}, \boldsymbol{\Sigma}_t^{(j)^{EKF}})}\boldsymbol{\omega}_{t-1}^{(j)}$ See equation (2.53)	These weights are reduced to the expression in equation (2.54) when resampling takes place at every time step.
UPF	$\mathcal{N}(\bar{\boldsymbol{x}}_t^{(j)^{UKF}}, \boldsymbol{\Sigma}_t^{(j)^{UKF}})$	$\frac{p(\boldsymbol{y}_t\|\boldsymbol{x}_t^{(j)^{UKF}})p(\boldsymbol{x}_t^{(j)^{UKF}}\|\boldsymbol{x}_{t-1}^{(j)})}{\mathcal{N}(\bar{\boldsymbol{x}}_t^{(j)^{UKF}}, \boldsymbol{\Sigma}_t^{(j)^{UKF}})}\boldsymbol{\omega}_{t-1}^{(j)}$ See equation (2.55)	These weights are reduced to the expression in equation (2.56) when resampling takes place at every time step.
ASIR PF	$q(\boldsymbol{x}_t, k\|\boldsymbol{y}_{1:t}) \propto$ $p(\boldsymbol{y}_t\|\mu_t^k)\, p(\boldsymbol{x}_t\|\boldsymbol{x}_{t-1}^k)\, \boldsymbol{\omega}_{t-1}^k$ See equation (2.49)	$\frac{p(\boldsymbol{y}_t\|\boldsymbol{x}_t^{k^{(j)}})}{p(\boldsymbol{y}_t\|\mu_t^{k^{(j)}})}$; $\lambda_t^k \propto p(\boldsymbol{y}_t\|\mu_t^k)\, w_{t-1}^k$ See equation (2.52)	The auxiliary variable k is introduced to aid the task of simulation. When ASIR is implemented, first the index k is sampled according to the so-called first stage weights λ_t^k. Once the index k is sampled, the state-particle x_t^j are sampled as done by the SIR PF, but according to the so-called second-stage weights $\boldsymbol{\omega}_t^{(j)}$ obtained as the ratio of equations (2.48) and (2.49).

CHAPTER 3

BENCHMARK SIMULATION STUDY: FILTERING IN A LINEAR FRAMEWORK

This chapter aims to illustrate the performance of the particle filtering methodology in a linear and Gaussian context. For that reason, we conduct two exhaustive Monte Carlo studies confronting some existing particle filter variants (or special cases of those) already described in Chapter 2 –named the sampling importance resampling (SIR), the adapted sampling importance resampling (special case of SISR PF, called by us SIRopt), the auxiliary sampling importance resampling (ASIR), and the special form of the extended particle filter (called by us KPF)– in contraposition to the analytical and well-known Kalman filter. Notice that all particle filters studied are variants of the generic SISR particle filter, but are mainly distinguished by the use of a different proposal PDF or the adoption of a distinct resampling scheme; see Table 2.2.

To achieve our goal, two apparently simple but important dynamic linear processes are chosen as benchmark models: 1) the so-called local level model, also known as random walk plus noise model, and 2) the contaminated AR(1) model, also known as AR(1) plus noise model. Consequently, we carry out two distinguishing simulation experiments corresponding to the two chosen models: the non-stationary local level model and the stationary AR(1) plus noise model.

This chapter is organized as follows. In Section 3.1, a generic state-space formulation of the linear models under study is specified. Also, therein some issues regarding our interest in the two chosen models are stated. Section 3.2 considers the general procedure used in the design of the simulation studies. This general procedure undertakes three steps: Data and State Generation, Filtering Estimation and Filtering Performance Criteria Computation. For completion a sketch illustrating the comparison criteria used in both the non-simulation based filters and the different PF variants, is given. This section also introduces the notation and formulae needed to define the criteria used for comparing all

the filters; additionally, some implementation issues are pointed out. This section concludes with a summary of the general simulation settings for the Monte Carlo (MC) experiments.

Section 3.3 presents the simulation study I dealing with the linear and Gaussian but non-stationary dynamic model named local level model. Therein, the state-space formulation of the model is specified and the relationship between the traditional ARIMA(0,1,1) process and the structural local level model is explicited. Also, for $N_p = 200$ particles, the corresponding simulation results, remarks and conclusions for simulation study I are reported, including a measure of degeneracy defined as the number of unique particles at the end time-period $t = T$. Additional Monte Carlo studies are conducted to determine the impact of increasing the number of particles and/or the time series length over the statistical performance of the chosen particle filters. using representative signal-to-noise-ratio settings.

Likewise, Section 3.4 deals with the simulation study II; in this case, the linear and Gaussian dynamic model under study is the stationary AR(1) plus noise model. Therein, the state-space formulation of the model is specified and some specific simulation settings are defined. Also, for $N_p = 200$ particles, the corresponding simulation results, remarks and conclusions for simulation study II are reported, including a measure of degeneracy. Additional Monte Carlo studies are also conducted to determine the impact of increasing the number of particles and/or the time series length over the statistical performance of the chosen particle filters.

To end up the chapter, in Section 3.5, final remarks and conclusions are stated.

We remark that in this chapter, all existing model parameters are assumed to be fixed and known and the stratified resampling scheme is adopted. We show all our findings in an empirical fashion using Monte Carlo experiments and put special effort in assessing the impact of the so-called signal-noise-ratio[1] (SNR) over the statistical performance of the studied filters.

3.1 Linear Models Under Study

As aforementioned, to illustrate how the chosen filters perform in a linear context, we use as a benchmark two linear dynamic models commonly studied by several authors and under different approaches: the non-stationary local level model and the stationary AR(1) plus noise model. See among others West and Harrison (1989), Harvey (1996), Kitagawa (1996), Doucet, de Freitas, and Gordon (2001), Tanizaki (2001), Durbin and Koopman (2001), Stock and Watson (2007), Pellegrini (2009) and Rodriguez (2010).

Following, a generic state-space formulation for the two dynamic linear models under study is provided. Specifically, the state transition equation for these two models can be expressed as

$$x_t = \phi_1 x_{t-1} + \eta_t \tag{3.1}$$

[1] concept coming from the engineering terminology indicating a measure of the relative variation of the state evolution equation to the observation equation; see West and Harrison (1989), p. 47.

where $\phi = 1$ indicates that we are dealing with the local level model. On the other hand, $|\phi| < 1$ specifies a stationary AR(1) plus noise process. The corresponding measurement equation is specified by

$$y_t = x_t + \nu_t. \tag{3.2}$$

Notice that the assumed uncorrelated transition and measurement disturbances η_t and ν_t could be non-Gaussian. However, unless stated otherwise, it is assumed that the state noise η_t follows a Gaussian distribution with mean zero and variance $\sigma^2_{\eta_t}$ and that the measurement noise ν_t also follows a Gaussian distribution with mean zero and variance $\sigma^2_{\nu_t}$. Consequently, the sequences defined by the disturbances η_t and ν_t are not only uncorrelated but also mutually independent.

The above equations (3.1) and (3.2) clearly specify a linear, Gaussian and (non-) stationary dynamic model state-space formulation. As known, if $|\phi| < 1$, the state-space model just defined is a stationary one; $\phi = 1$ implies non-stationarity. A graphical representation of three exemplar runs of the generated univariate data y_t and corresponding state values x_t with $SNR = 1$ is displayed in Figure 3.1, where the local level model (with $\phi = 1$) plots are shown in panel a) and the ones for the AR(1) plus noise model in panels b) and c) for $\phi = 0.3$ and $\phi = 0.8$, respectively. Panel a) illustrates the seemingly non-stationary character of the local level model; the generated state values x_t do not vary about a fixed level. On the contrary, panels b) and c) portray the stationary behavior typical of the specified contaminated AR(1) models; the generated state values x_t appear to vary about a fixed level.

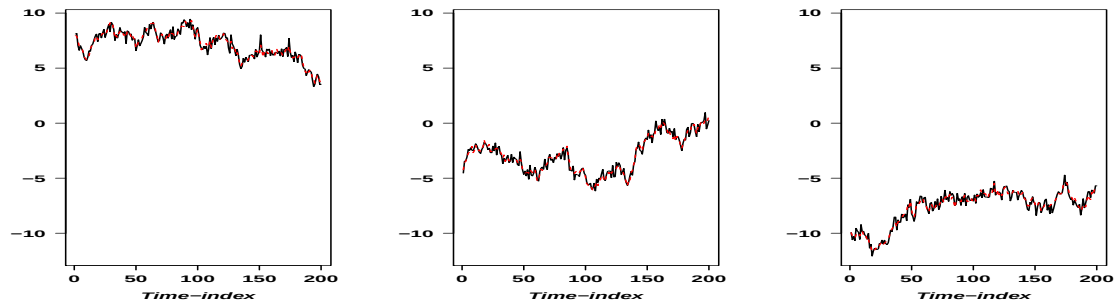
Following, we state the reasons for our interest in these two models:

1. The local level model is the simplest form of the so-called structural time series models which are a keystone within the time series analysis. Indeed, literature shows that they are important for the further definition and analysis of more complex models (Harvey 1996).

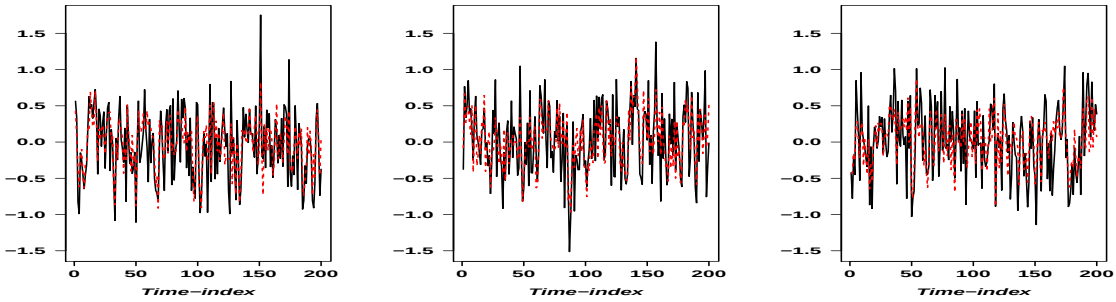
2. The studied linear models are related to the traditional and well-known Box and Jenkins' ARIMA time series processes (Box and Jenkins 1976). It can be shown that the reduced form of the local level model is represented by the ARIMA(0,1,1) model; see for instance Durbin and Koopman (2001). Likewise, the reduced form of the AR(1) plus noise model can be represented as an ARIMA(1,0,1) process; see among others Wei (1994), Gómez and Maravall (1994) and Shumway and Stoffer (2000).

3. The two chosen models are not only commonly found in the literature as illustrative examples but also have multiple applications. For instance, see West and Harrison (1989), Doucet, de Freitas, and Gordon (2001), Stock and Watson (2007), Pellegrini (2009) and Rodriguez (2010) for academic and practical applications. Particularly, Stock and Watson (2007) propose the combination of the local level model and the so-called stochastic volatility model to reflect the main features of the U.S. inflation time series.

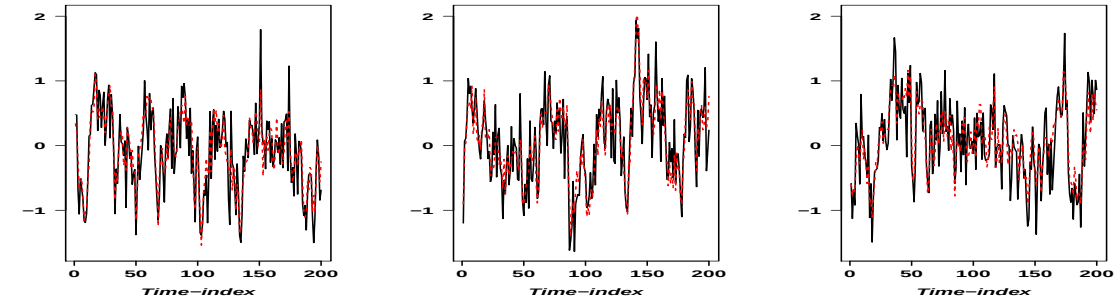
4. Most important, it is well known that when dealing with a linear and Gaussian dynamic model where all parameters are known, a closed form solution exists and it is given by the non-simulation based KF (pseudo-code in Algorithm 1), which full filter description is given in Section 2.3. The



(a) Local level model ($\phi = 1$)



(b) AR(1) plus noise model with $\phi = 0.3$



(c) AR(1) plus noise model with $\phi = 0.8$

Figure 3.1: Three exemplar runs of the generated data $\boldsymbol{y}_t$ (black/continuous) and simulated states $\boldsymbol{x}_t$ (red/dashed) for each of the three models specified by $\phi = 1$, $\phi = 0.3$, and $\phi = 0.8$, respectively. Results shown for $SNR = 1$ with $\sigma_\eta^2 = \sigma_\nu^2 = 0.1$.

fact that an analytical filtering solution exists helps us to get acquainted with the implementation issues of the simulation based SIR, SIRopt, ASIR and KPF algorithms and to assess their behavior in contrast to the analytical KF approach. Thus, this is an ideal context not only to illustrate the performance of the particle filtering methodology in contraposition with the gold standard KF but also, at the same time, to carry out a thorough MC study on the impact of the signal-to-noise-ratio over the quality of the estimations.

In the literature, one finds some references to the use of the signal-to-noise-ratio concept in relation to applications and/or estimation issues. For instance, according to West and Harrison (1989), the major number of applications of the local level model with constant variances (called by them: constant model) are in short term forecasting and control with, typically, signal-to-noise-ratios spanning from 0.001 to 0.2. Also, Pellegrini (2009) considers some MC studies involving the estimation of the parameters of the local level model with signal-to-noise-ratio values $q \in \{1,2\}$. This author also considers an empirical application where the local level model is fitted to the daily Pound/Euro exchange rate time series ($T = 1626$ and estimated value of the signal-to-noise-ratio $\hat{q} = 2.338$)[2].

To our knowledge, no one has performed an exhaustive MC study of the influence of the signal-to-noise-ratio on the quality of the estimations of different competing filters that usually appear in the literature in separate studies. We carry out such a study and show (empirically) through simulations the impact of the signal-to-noise-ratio over the statistical performance of the five chosen competing filters. The chosen settings for the signal-to-noise-ratio are the 13 values in the set $q \in \{0.0001, 0.001, 0.01, 0.05, 0.1, 0.2, 0.3, 0.5, 1, 2, 3, 5, 10, 100\}$, which clearly cover the values usually found in the consulted literature as well as extremely low/high values. Indeed, we consider this as one of the main contributions of this chapter.

Next, we provide the general procedure to be followed for any benchmark Monte Carlo Study which is tackled.

3.2 Simulation Design

The general procedure for our benchmark Monte Carlo studies undertakes the following three steps:

- **STEP I**: Data and state generation
- **STEP II**: Filtering estimation
- **STEP III**: Filtering performance criteria computation

Following, we provide a detailed description of the aforementioned general simulation steps. Notice that within every simulation step, we further specify the needed instructions to carry out the MC experiments with the two chosen models.

3.2.1 STEP I: Data and State Generation

Generate $S = 100$ realizations of the chosen dynamic model. For the models studied in this chapter, this is carried out as follows:

[2] They use quasi-maximum likelihood (QML) estimation to obtain this value.

(Ia) Specify the value for the parameter ϕ. Remind that $\phi = 1$ indicates that we are dealing with the local level model and if $|\phi| < 1$ with the stationary AR(1) plus noise model. In this work, we will handle values of $\phi \in \{0.3, 0.8, 1\}$

(Ib) Specify the measurement noise variance $\boldsymbol{\sigma}^2_{\nu_t} = 0.1$, the time series data length $T = 200$ and the value of the transition noise variance $\boldsymbol{\sigma}^2_{\eta_t}$. Notice that $\boldsymbol{\sigma}^2_{\eta_t}$ takes a value from the second column of Table 3.1 that contains 13 different settings.

(Ic) Generate the random numbers $\boldsymbol{\eta}_t$ and $\boldsymbol{\nu}_t$. In this case, both noises are generated from a univariate normal distribution $\boldsymbol{\eta}_t \sim \mathcal{N}(0, \boldsymbol{\sigma}^2_{\eta_t})$ and $\nu_t \sim \mathcal{N}(0, \boldsymbol{\sigma}^2_{\nu_t})$, respectively.

(Id) Simulate the state-values $\boldsymbol{x}_t$ and data $\boldsymbol{y}_t$, $t = 1, \ldots, T$, from the transition equation (3.1) and the measurement equation (3.2), respectively.

(Ie) Repeat (Ia)–(Id) $S = 100$ times.

As aforementioned, we consider 13 simulation settings varying accordingly to the 13 specified values of the state noise variance parameter σ^2_η; the true model measurement noise variance parameter is fixed to $\sigma^2_\nu = 0.1$. For the sake of simplicity, in the sequel we refer to these different scenaries in terms of the so-called signal-to-noise-ratio which is defined as the ratio $q = \frac{\sigma^2_\eta}{\sigma^2_\nu}$; see columns two and three in Table 3.1. Notice that different signal-to-noise-ratio settings are chosen and ordered moving from a non informative observations case ($\sigma^2_\nu >> \sigma^2_\eta$) up to a very informative observations case ($\sigma^2_\nu << \sigma^2_\eta$). In the sequel, we refer to signal-to-noise-ratio values by SNR or q, indistinctively. The entries on the last two columns of Table 3.1, related to the local level model, will be commented on later on.

We remark that in this chapter all model parameters are assumed to be fixed and known. Therefore, we will focus on filtering the states via the chosen filters.

3.2.2 STEP II: Filtering Estimation

Herein, we provide the notation and general formulae needed in order to define the criteria for comparing the filters. Then, we list the further steps needed in order to perform the filtering estimation.

Notation and Formulae

Denote by T, N_p, and S the total number of observations (time series length), the number of particles, and the number of generated data sets, respectively.

For the replication set i, $i = 1, \ldots, S$, let $\{\boldsymbol{x}_{t,[i]}, t = 1, \ldots, T\}$ be the set formed by all the 'true' state variable $\boldsymbol{x}_{t,[i]}$ at time t. Likewise, denote by $\{\hat{\boldsymbol{x}}^f_{t,[i]}, t = 1, \ldots, T\}$ the ith set containing the filtering estimates $\hat{\boldsymbol{x}}^f_{t,[i]}$ obtained through the specified filter, say f. Recall that the estimate $\hat{\boldsymbol{x}}^f_{t,[i]}$ is just the filtered mean computed directly in case of the non-simulation based filter, in this case the KF. However, when dealing with any of the particle filter variants, this estimate is computed as the arithmetic average of

Table 3.1: Settings for the simulation studies with $\sigma_v^2 = 0.1$

				LLM	
Case	$\boldsymbol{\sigma_\eta^2}$	**SNR (q)**	**Type**	$\boldsymbol{\theta}^a$	$\mathbf{A}^b$
1	0.00001	0.0001	Non informative	−0.99	0.01
2	0.0001	0.001	↓	−0.97	0.03
3	0.001	0.01	↓	−0.90	0.10
4	0.005	0.05	↓	−0.80	0.20
5	0.01	0.1	↓	−0.73	0.27
6	0.02	0.2	⋮	−0.64	0.36
7	0.03	0.3	⋮	−0.58	0.42
8	0.05	0.5	⋮	−0.50	0.50
9	0.1	1	⋮	−0.38	0.62
10	0.2	2	↓	−0.27	0.73
11	0.5	5	↓	−0.15	0.85
12	1	10	↓	−0.08	0.92
13	10	100	Very informative	−0.01	0.99

[a] Corresponding MA parameter value of the reduced form ARIMA(0,1,1) process

[b] Limiting value of the adaptive coefficient defining the rate of adaptation to new data: $A = q(\sqrt{(1+4/q)}-1)/2$, West and Harrison (1989), p. 53.

the N_p posterior particles, $\widetilde{\boldsymbol{x}}_t^{f(j)}$, $j = 1,\ldots,N_p$, generated by the specific particle filter f. That is, for a replication set i, a chosen particle filter variant f and time index t, the filtering estimate is given by

$$\hat{\boldsymbol{x}}_{t,[i]}^f = \frac{1}{N_p}\sum_{j=1}^{N_p}\widetilde{\boldsymbol{x}}_t^{f(j)}. \tag{3.3}$$

Then, for each filter f, we compute the root mean square error (RMSE) over time corresponding to replication set i as:

$$RMSE_{[i]}^f = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(\boldsymbol{x}_{t,[i]} - \hat{\boldsymbol{x}}_{t,[i]}^f\right)^2}. \tag{3.4}$$

Additionally, a measure of the computational performance of the filters is defined. That is, for each filter f and replication set i, the total elapsed time for estimating the states of a time series of length T, is computed as:

$$CPU_{[i]}^f = \sum_{t=1}^{T} CPU_{t,[i]}^f. \tag{3.5}$$

Filtering Steps

For each filter f, obtain both the statistical and computational measure of performance of the studied filter f; these are based on the previously defined root mean square error (RMSE) and the CPU time criterion, respectively. Specifically, assuming known model parameters and given the simulated data $\boldsymbol{y}_{1:T} = y_1, \ldots, y_T$ generated in step I), for replication set i, $i = 1, \ldots, S$, proceed to

(IIa) Compute the filtering estimates $\hat{\boldsymbol{x}}^f_{t,[i]}$, $t = 1, \ldots, T$, using the filter in question, say f. Recall that $f \in$ {KF, SIR, SIRopt, ASIR, KPF}.

(IIb) Compute $RMSE^f_{[i]}$: the RMSE over time index $t = 1, \ldots, T$ with equation (3.4)

(IIc) Compute $CPU^f_{[i]}$: the total elapsed time for a total of T observations with equation (3.5)

(IId) Repeat steps (IIa)–(IIc) $S = 100$ times.

Following, we explicitly provide the criteria for comparing the filters as well as the specific steps to compute the statistical and computational measures of filtering performance.

3.2.3 STEP III: Filtering Performance Criteria Computation

The statistical performance of the filters is explicitly defined in terms of the mean and the variance over replication sets of the root mean square errors computed in equation (3.4). In other words, for each filter *f*, both Mean$(RMSE)_f$ and Var$(RMSE)_f$ are computed as the arithmetic average and variance of $RMSE^f_{[i]}$, $i = 1, \ldots, S$. Specifically,

$$\text{Mean}(RMSE)_f = \frac{1}{S}\sum_{i=1}^{S} RMSE^f_{[i]}, \tag{3.6}$$

$$\text{Var}(RMSE)_f = \frac{1}{S-1}\sum_{i=1}^{S}\left(RMSE^f_{[i]} - \text{Mean}(RMSE)_f\right)^2. \tag{3.7}$$

Notice that two sketches are provided for a better illustration of the simulation design and performance criteria used. The first sketch illustrates the criteria for comparing the non-simulation based filters; see Figure 3.2. Additionally, Appendix A, located at the end of this work, includes a sketch of the comparison criteria for the simulation based filters; see Figure A.1.

The computational performance of the filters is defined in terms of the mean over replication sets of the total elapsed time for estimating the states of a time series of length T already computed with equation (3.5).

$$\text{Mean}(CPU)_f = \frac{1}{S}\sum_{i=1}^{S} CPU^f_{[i]}. \tag{3.8}$$

Therefore, the followings specific steps must be undertaken:

(IIIa) In step (IIb), we end up with $S = 100$ estimates of the RMSE: $RMSE^f_{[i]}$; see the first column of the second table in Figure 3.2 and Figure A.1. Based on these, obtain the mean and the variance of the root mean square (RMSE) computed over time and over replication sets using equations (3.6) and (3.7), respectively.

(IIIb) In step (IIc), we end up with $S = 100$ CPU elapsed-time estimates: $CPU^f_{[i]}$; see the second column of the second table in Figure 3.2 and Figure A.1. Based on these, obtain the mean CPU elapsed-time computed over replication sets using (3.8).

		Time index					
Set(i)	**Filter (f)**	**1**	**2**	.....	**T**		**Comparison criteria**
1	**EKF/UKF**	$\hat{\boldsymbol{x}}^f_{1,[1]}$	$\hat{\boldsymbol{x}}^f_{2,[1]}$	.....	$\hat{\boldsymbol{x}}^f_{T,[1]}$	$\longrightarrow$	$\text{RMSE}^f_{[1]}$ $\quad$ $\text{CPU}^f_{[1]}$
2	**EKF/UKF**	$\hat{\boldsymbol{x}}^f_{1,[2]}$	$\hat{\boldsymbol{x}}^f_{2,[2]}$	.....	$\hat{\boldsymbol{x}}^f_{T,[2]}$	$\longrightarrow$	$\text{RMSE}^f_{[2]}$ $\quad$ $\text{CPU}^f_{[2]}$
$\vdots$				$\vdots$			$\vdots$ $\quad$ $\vdots$
$\vdots$				$\vdots$			$\vdots$ $\quad$ $\vdots$
S	**EKF/UKF**	$\hat{\boldsymbol{x}}^f_{1,[S]}$	$\hat{\boldsymbol{x}}^f_{2,[S]}$	.....	$\hat{\boldsymbol{x}}^f_{T,[S]}$	$\longrightarrow$	$\text{RMSE}^f_{[S]}$ $\quad$ $\text{CPU}^f_{[S]}$
							$\Downarrow$ $\quad$ $\Downarrow$
							Mean(RMSE)_f $\quad$ Mean(CPU)_f
							Var(RMSE)_f

Figure 3.2: Sketch I: Comparison criteria of non-simulation based filters EKF and UKF

We remark that we choose the statistical performance criteria presented by the authors of the unscented particle filter (Van der Merwe et al. 2001): the RMSE. Consequently, the general skeleton of our main program is based on and inspired by these authors and on the technical report of Van der Merwe et al. (2000) that develop a program-Demo which is the base for the results presented in Van der Merwe et al. (2001). We stress, however, that the full implementation of every specific filter under study is performed by the author of this PhD thesis and is one of the contributions of this work. Apart from the statistical performance measures defined by the aforementioned authors and also used by us, we constructed the computational performance measure defined above and specified in equations (3.5) and (3.8).

Following, we list the simulation settings needed to define all possible scenaries that are common to the two carried out MC studies; further simulation settings are specified when needed.

3.2.4 Summary of General Simulation Settings

Herein, we present a summary of the simulation settings of the two Monte Carlo experiments which have been conducted. As aforementioned, we consider two distinguishing simulations studies: Sim-

ulation I referring to the non-stationary local level model and Simulation II when dealing with the stationary AR(1) plus noise model.

- **Filters**: KF, SIRopt, SIR, ASIR and KPF
- **Comparison criteria**: RMSE and CPU time
- **Initial state variance used in estimation procedure**: $P_0 = \sigma_\eta^2$ and $P_0 = 100 \cdot \sigma_\eta^2$ (In Simulation II, we consider only the case with $P_0 = 100 \cdot \sigma_\eta^2$).
- **Measurement noise variance**: Fixed to $\sigma_\nu^2 = 0.1$
- **State noise variance** σ_η^2: 13 settings defined in the second column of Table 3.1; see the corresponding signal-to-noise-ratio values in the third column. In this work, we carry out an exhaustive study of the impact of the signal-to-noise-ratio on the quality of the filtering estimates.
- **Resampling scheme**: Stratified resampling
- **Number of replications**: $S = 100$
- **Number of particles**: $N_p = 200$
- **Time series length**: $T = 200$

Notice that though we first focus our attention on the performance of the different particle filters solely for 200 particles, later we also present results which reflect how the increase on the number of particles influence the quality of the estimation for the simulation based filters.

3.3 Simulation Study I: The Non-stationay Local Level Model

Under linearity and Gaussianity assumptions, this Monte Carlo study basically compares the performance of some particle filter variants with the performance of the gold standard filter: the Kalman Filter. The benchmark model in this case is the non-stationary process known as the local level model.

3.3.1 State Space Representation

The equations (3.1) and (3.2) with $\phi = 1$ specify the state-space formulation of the linear, Gaussian and non-stationary dynamic model known as local level model. In that case, the parametric state-space formulation for this dynamic model can be described and subsumed by the following two equations (see, for instance, Shephard and Harvey (1990) and Durbin and Koopman (2001)):

$$\begin{aligned} x_t &= x_{t-1} + \eta_t \\ y_t &= x_t + \nu_t \end{aligned} \tag{3.9}$$

To complete the state-space formulation, we must assume a distribution for the initial state variable x_0. In this particular case, $x_0 \sim N(\mu_{x_0}, \Sigma_{x_0})$, the noise disturbances η_t and ν_t are assumed to follow a Gaussian distribution with fixed and known variance parameters $\sigma^2_{\eta_t}$ and $\sigma^2_{\nu_t}$; thus only the states x_t are estimated. When the scale noise parameters are fixed and known, the local level model is called constant model by West and Harrison (1989). Some other authors called it random walk plus noise model; see for instance Harvey (1996).

As aforementioned, the reduced form of the local level model is an ARIMA process. Following, the explicit relationship between a traditional ARIMA process and the structural local level model is presented.

3.3.2 Reduced Form of the Local Level Model: an ARIMA(0,1,1) Model

The local level model can be represented as an invertible ARIMA(0,1,1) model under suitable values for the moving average parameter θ.

Define $\Delta Z_t = Z_t - Z_{t-1}$ as the first difference for any stochastic variable Z_t. Then, departing from the state-space formulation of the local level model in (3.9), an ARIMA(0,1,1) type process can be obtained as $\Delta y_t = \eta_t + \Delta \nu_t = a_t + \theta a_{t-1}$ where the white noise, a_t, follows a Gaussian distribution, $a_t \sim \mathcal{N}(0, \sigma^2)$, and θ, the moving average parameter, is a function of the signal-to-noise-ratio, say $\theta = \left((q^2 + 4q)^{0.5} - 2 - q\right)/2$; see (Harvey 1996).

In the reduced form of the local level model, the moving average parameter θ must lie in the interval $-1 < \theta < 0$. We verify this, empirically, by computing the moving average parameter value corresponding to the 13 settings on page 53 defined by the 13 chosen signal-to-noise-ratio values; focus on the third and fifth columns of Table 3.1. As listed therein, for this model we consider a variety of signal-to-noise-ratio settings including extremely low and high values, and we observe that as the signal-to-noise-ratio q increases, the moving average parameter θ decreases. In column six of Table 3.1, we also report the measure A which defines the rate of adaptation to new incoming data; see page 53 of West and Harrison (1989). Notice that in this case, $A = 1 + \theta$. In this work, we aim to determine if some patterns are observed on the quality of the filtering estimates as a function of the chosen signal-to-noise-ratio values.

Following, the simulation results, remarks and conclusions regarding Simulation study I are presented.

3.3.3 Results, Remarks and Conclusions for Simulation Study I

Experimental Results

In Table 3.2, we provide the numeric results which summarize and assess the performance of the different filters under study in handling the local level model. Firstly, this table is organized in three different blocks where the first two correspond to the two estimation setting values used for the initial state vari-

ance, $P_0 = \sigma_\eta^2$ and $P_0 = 100 \cdot \sigma_\eta^2$. Each one of these blocks is composed itself of two columns containing the two measures: Mean(RMSE) and Var(RMSE).

Secondly, being aware that by definition all particle filters at some point in time suffer the degeneracy problem, in the third block (last column) of Table 3.2 we report the mean and standard deviation (mean (SD)) of the variable uNp, meaning unique number of particles at the end time-period $t = T$. This is done with the aim to somehow quantify the potential degeneracy problem of all particle filter variants under study. Finally, the average time in seconds for handling a data set containing $T = 200$ observations is also computed but reported afterwards in the text.

All the reported simulation results are later commented on, and, for the sake of a better understanding of them, different types of plots are constructed.

Table 3.2: Summary of simulation study I under 13 different settings: $\phi = 1$, $\sigma_\nu^2 = 0.1$, $T = 200$, and $N_p = 200$

		$P_0 = \sigma_\eta^2$		$P_0 = 100 \cdot \sigma_\eta^2$		
		RMSE		RMSE		uNp[2]
Setting	Filter	Mean	Var	Mean	Var	Mean(SD)
Case 1 $\sigma_\eta^2 = 1e-05$, SNR $= 1e-04$						
	KF	0.148	0.015	0.109	0.008	—
	SIRopt	0.156	0.021	0.129	0.017	194.02 (4.88)
	SIR	0.156	0.021	0.129	0.017	194.03 (4.88)
	ASIR	0.158	0.021	0.134	0.018	199.69 (1.28)
	KPF	0.179	0.024	0.174	0.022	35.78 (1.88)
Case 2 $\sigma_\eta^2 = 1e-04$, SNR $= 1e-03$						
	KF	0.101	0.004	0.073	9e-04	—
	SIRopt	0.103	0.005	0.078	0.002	189.04 (8.74)
	SIR	0.103	0.005	0.078	0.002	188.89 (8.85)
	ASIR	0.106	0.005	0.081	0.003	198.23 (2.88)
	KPF	0.116	0.006	0.095	0.004	59.73 (6.60)
Case 3 $\sigma_\eta^2 = 0.001$, SNR $= 0.01$						
	KF	0.109	6e-04	0.102	2e-04	—
	SIRopt	0.109	6e-04	0.102	3e-04	180.43 (14.07)
	SIR	0.109	7e-04	0.102	3e-04	179.65 (14.5)
	ASIR	0.111	8e-04	0.103	3e-04	193.89 (4.7)
	KPF	0.112	8e-04	0.105	3e-04	94.10 (8.62)

[2] uNp: average number (standard deviation) of unique particles at time $t = T$.

Table 3.2: Summary of simulation study I under 13 different settings: $\phi = 1$, $\sigma_v^2 = 0.1$, $T = 200$, and $N_p = 200$ (continued)

		$P_0 = \sigma_\eta^2$		$P_0 = 100 \cdot \sigma_\eta^2$		
		RMSE		RMSE		uNp
Setting	Filter	Mean	Var	Mean	Var	Mean (SD)
Case 4 $\sigma_\eta^2 = 0.005$, SNR = 0.05						
	KF	0.144	3e-04	0.142	2e-04	—
	SIRopt	0.144	3e-04	0.144	2e-04	172.64 (17.79)
	SIR	0.145	3e-04	0.144	2e-04	169.02 (19.57)
	ASIR	0.145	3e-04	0.144	3e-04	185.63 (8.67)
	KPF	0.145	3e-04	0.145	2e-04	121.17 (13.98)
Case 5 $\sigma_\eta^2 = 0.01$, SNR = 0.1						
	KF	0.165	2e-04	0.164	2e-04	—
	SIRopt	0.166	2e-04	0.166	2e-04	169.22 (18.86)
	SIR	0.166	3e-04	0.166	2e-04	162.43 (21.76)
	ASIR	0.166	3e-04	0.165	2e-04	179.90 (10.65)
	KPF	0.166	3e-04	0.166	2e-04	132.92 (17.58)
Case 6 $\sigma_\eta^2 = 0.02$, SNR = 0.2						
	KF	0.188	2e-04	0.188	2e-04	—
	SIRopt	0.189	2e-04	0.189	2e-04	165.97 (19.88)
	SIR	0.190	2e-04	0.190	2e-04	154.33 (24.43)
	ASIR	0.189	2e-04	0.189	2e-04	171.82 (13.42)
	KPF	0.190	2e-04	0.190	2e-04	142.20 (19.40)
Case 7 $\sigma_\eta^2 = 0.03$, SNR = 0.3						
	KF	0.203	2e-04	0.203	2e-04	—
	SIRopt	0.204	2e-04	0.204	2e-04	164.99 (20.61)
	SIR	0.204	2e-04	0.204	2e-04	148.92 (26.02)
	ASIR	0.204	2e-04	0.204	2e-04	165.51 (14.83)
	KPF	0.204	2e-04	0.204	2e-04	147.37 (20.25)
Case 8 $\sigma_\eta^2 = 0.05$, SNR = 0.5						
	KF	0.222	2e-04	0.222	2e-04	—
	SIRopt	0.223	2e-04	0.223	2e-04	164.81 (20.37)
	SIR	0.223	2e-04	0.223	2e-04	140.95 (27.71)
	ASIR	0.223	2e-04	0.223	2e-04	156.12 (16.99)
	KPF	0.223	2e-04	0.223	2e-04	153.32 (20.68)

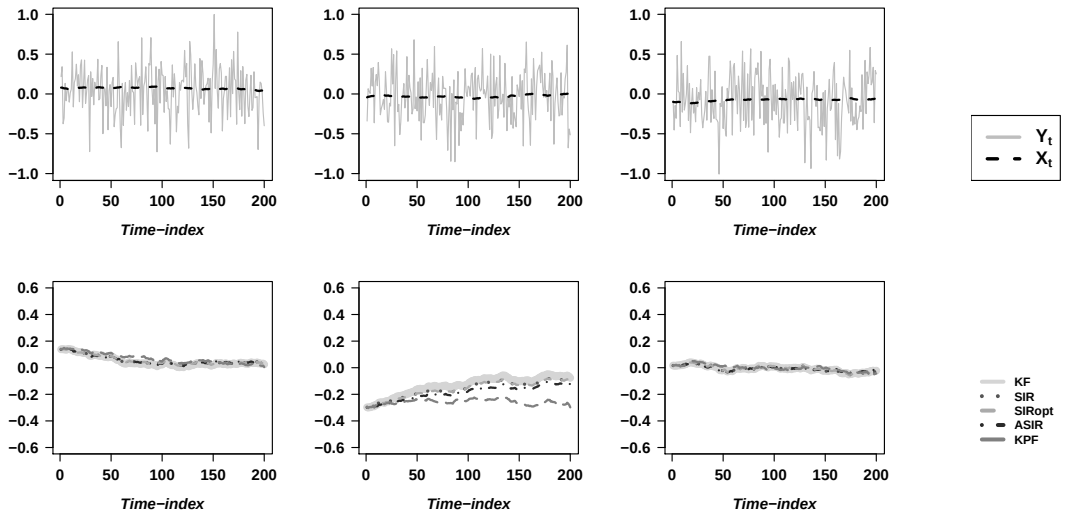
Table 3.2: Summary of simulation study I under 13 different settings: $\phi = 1$, $\sigma_v^2 = 0.1$, $T = 200$, and $N_p = 200$ (continued)

		$P_0 = \sigma_\eta^2$		$P_0 = 100 \cdot \sigma_\eta^2$		
		RMSE		RMSE		uNp
Setting	Filter	Mean	Var	Mean	Var	Mean (SD)
Case 9 $\sigma_\eta^2 = 0.1$, SNR = 1						
	KF	0.246	2e-04	0.246	2e-04	—
	SIRopt	0.247	2e-04	0.247	2e-04	166.11 (19.85)
	SIR	0.248	2e-04	0.248	2e-04	127.81 (29.28)
	ASIR	0.247	2e-04	0.248	2e-04	139.86 (20.38)
	KPF	0.247	2e-04	0.247	2e-04	160.82 (20.83)
Case 10 $\sigma_\eta^2 = 0.2$, SNR = 2						
	KF	0.268	2e-04	0.268	2e-04	—
	SIRopt	0.269	2e-04	0.269	2e-04	170.23 (18.74)
	SIR	0.270	2e-04	0.270	2e-04	112.15 (28.97)
	ASIR	0.270	2e-04	0.271	2e-04	118.18 (23.55)
	KPF	0.269	2e-04	0.269	2e-04	168.09 (19.66)
Case 11 $\sigma_\eta^2 = 0.5$, SNR = 5						
	KF	0.289	2e-04	0.289	2e-04	—
	SIRopt	0.290	2e-04	0.290	2e-04	176.78 (15.39)
	SIR	0.292	2e-04	0.292	2e-04	89.44 (25.79)
	ASIR	0.297	2e-04	0.298	2e-04	87.92 (23.34)
	KPF	0.290	2e-04	0.290	2e-04	175.54 (15.73)
Case 12 $\sigma_\eta^2 = 1$, SNR = 10						
	KF	0.299	2e-04	0.300	2e-04	—
	SIRopt	0.300	2e-04	0.302	2e-04	181.93 (11.58)
	SIR	0.304	2e-04	0.305	2e-04	72.13 (21.86)
	ASIR	0.315	3e-04	0.315	2e-04	66.56 (20.61)
	KPF	0.300	2e-04	0.300	2e-04	181.77 (12.67)
Case 13 $\sigma_\eta^2 = 10$, SNR = 100						
	KF	0.312	2e-04	0.312	2e-04	—
	SIRopt	0.313	2e-04	0.314	2e-04	193.11 (2.85)
	SIR	0.351	0.002	0.351	0.002	30.13 (10.01)
	ASIR	0.373	0.002	0.369	0.001	25.66 (8.53)
	KPF	0.313	2e-04	0.313	2e-04	193.31 (2.97)

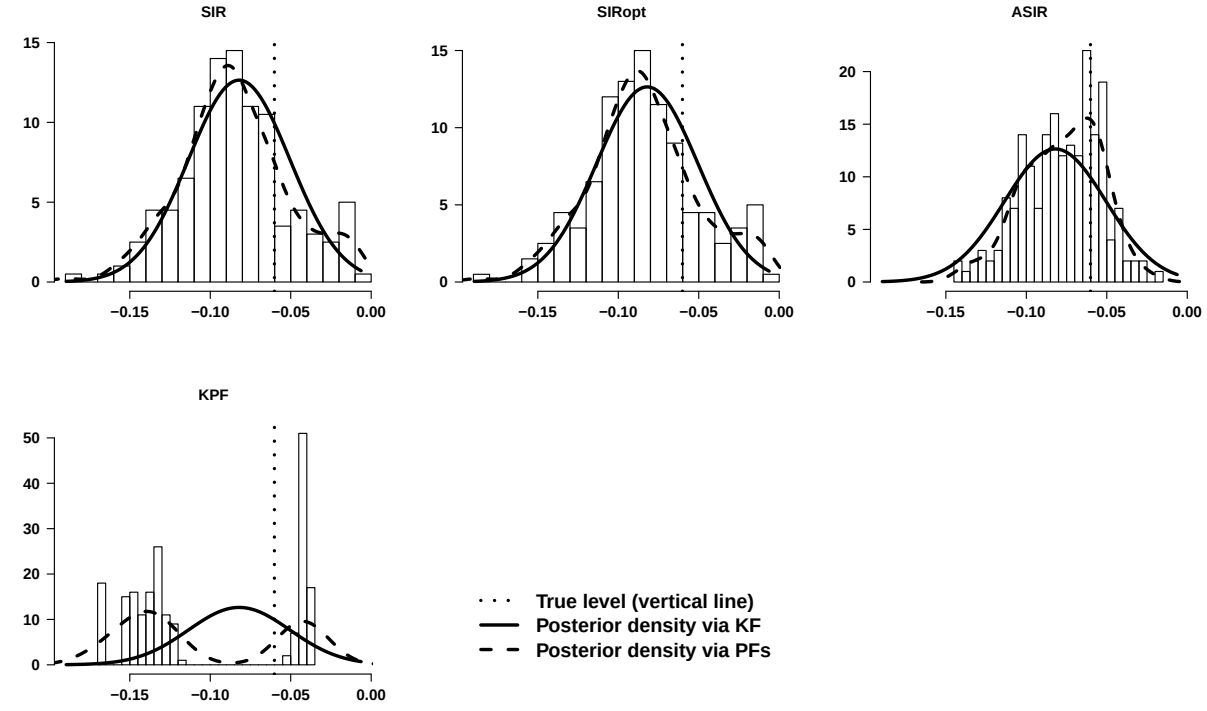
As aforementioned, we produce some figures, distinguished by the chosen signal-to-noise ratio value denoted by q, to better illustrate the performance of the competing filters in estimating the states for the local level model. Before proceeding to comment on the simulation results and figures, we remark that in this chapter, on pages 62–64, we only show figures corresponding to three out of the 13 possible cases defined in Table 3.1; specifically, we present Figures 3.3–3.5 displaying results for cases 1, 9 and 13 which correspond to signal-to-noise ratio values q equal to 0.0001, 1 and 100, respectively. The idea is to illustrate the performance of the competing filters at two extreme values of the signal-to-noise-ratio and at a moderate value $q = 1$. For completion, the rest of the figures are included in Appendix B, which can be found on the following website: `http://www-eio.upc.edu/~lacosta/AppendixB.pdf` [last visited: September 2013].

In each figure, for each signal-to-noise ratio setting, we show a graphical illustration of the generating process (observations and states) as well as of the filtering performance. In other words, for three particular sets of data, in the first row of the upper panel of these figures we plot together the evolution of the generated observations y_t and states $x_{t|t}$, $t = 1,\dots,T$. In the second row of this panel, the evolution of the difference between the estimated state values and corresponding true state values $(\hat{x}_{t|t} - x_t, t = 1,\dots,T)$ obtained via the studied filters, is displayed. Figures 3.3–3.5 (pages 62–64), with the same y-scale, confirm the general results reported in Table 3.2 which suggest that an increase in the signal-to-noise-ratio value q is related to less precise estimation results; we observe that the evolution of the difference between estimated and true-state values show more variability as q increases.

For the last exemplar run and last time-index T, to further illustrate the behavior of the different filters under study, in the bottom panel of each of the aforementioned graphs, we present the marginal posterior densities (with corresponding histograms) estimated via the four PF variants under study in contraposition with the exact Gaussian posterior density yielded by the gold standard KF. What is shown in these plots is also in concordance with the results presented in Table 3.2. For example, Table 3.2 and Figure 3.3 corresponding to Case 1 ($q = 0.0001$) indicate that the KPF particle filter variant not only shows the worst performance in terms of RMSE but also the lowest average number of unique particles at last time-index $t = T$. This worst performance is perfectly reflected in the histograms where we observe that the estimated posterior density obtained via the KPF is the one that departs the most from the density estimated by the optimal KF.

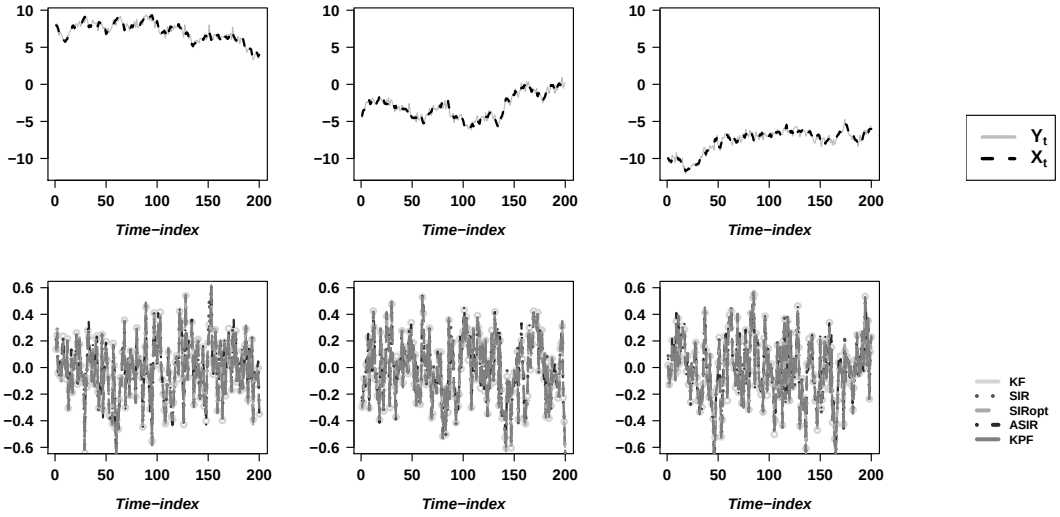


(a) First row: Generated states and observations. Second row: Difference between estimated and true-state values $\hat{x}_{t|t} - x_t$, $t = 1, \ldots, T$.

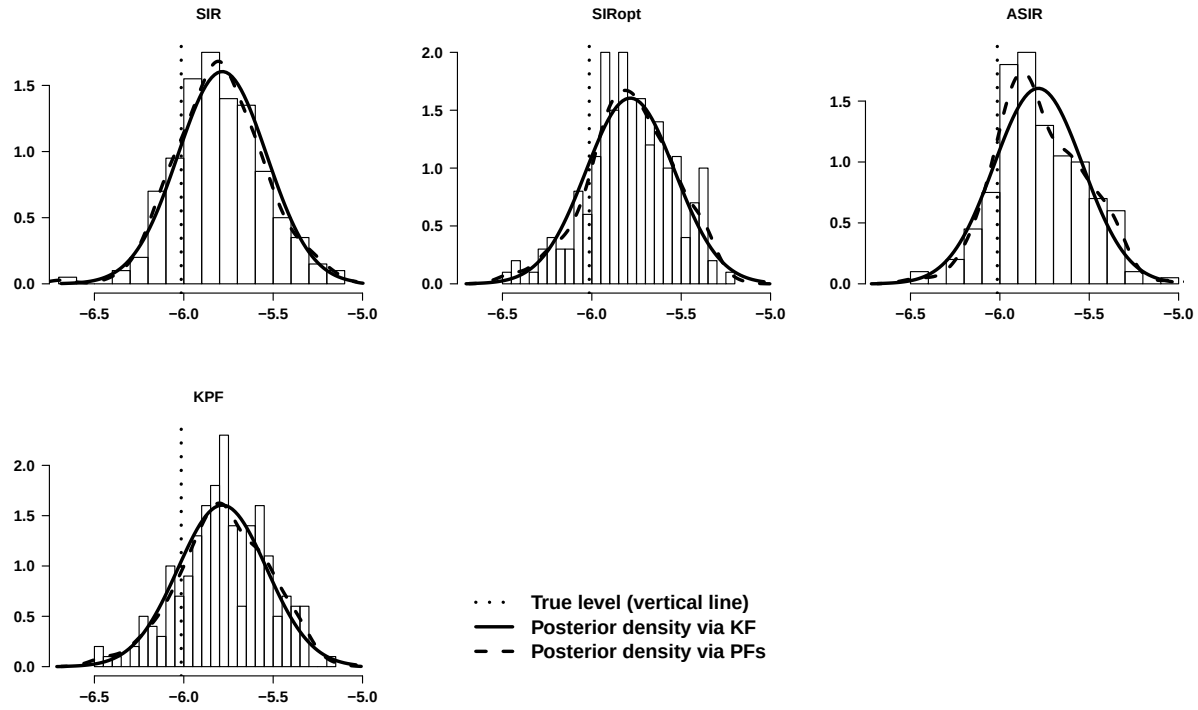


(b) Histogram (together with the estimated posterior density; black/dashed) of state values $\hat{x}_{T|T}$ for last data set. The exact posterior density obtained via the KF (black/continuous) is overlaid to each histogram.

Figure 3.3: Local level model: Case 1 with SNR $q = 0.0001$ ($\sigma_\eta^2 = 1e-5$ and $\sigma_\nu^2 = 0.1$)

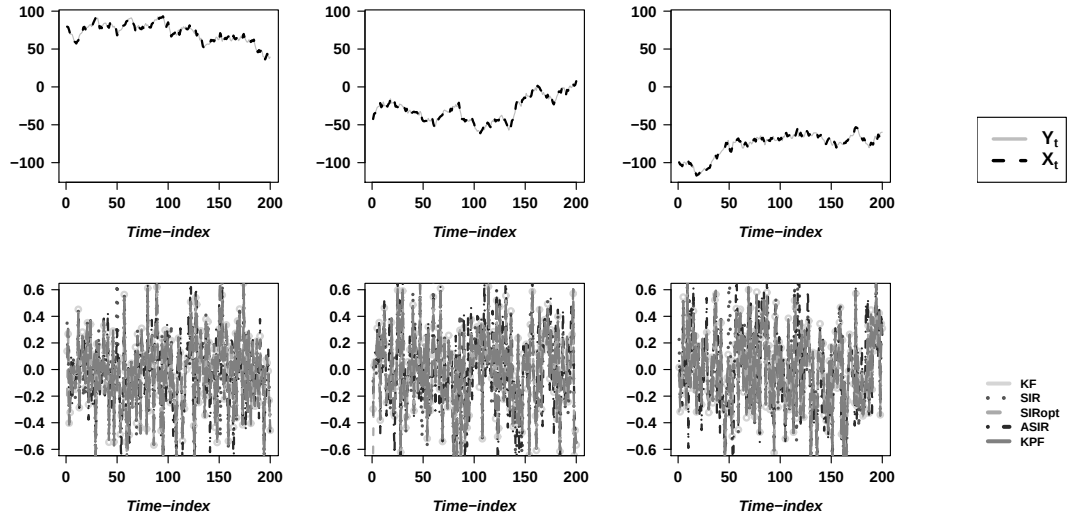


(a) First row: Generated states and observations. Second row: Difference between estimated and true-state values $\hat{x}_{t|t} - x_t, t = 1, \ldots, T$.

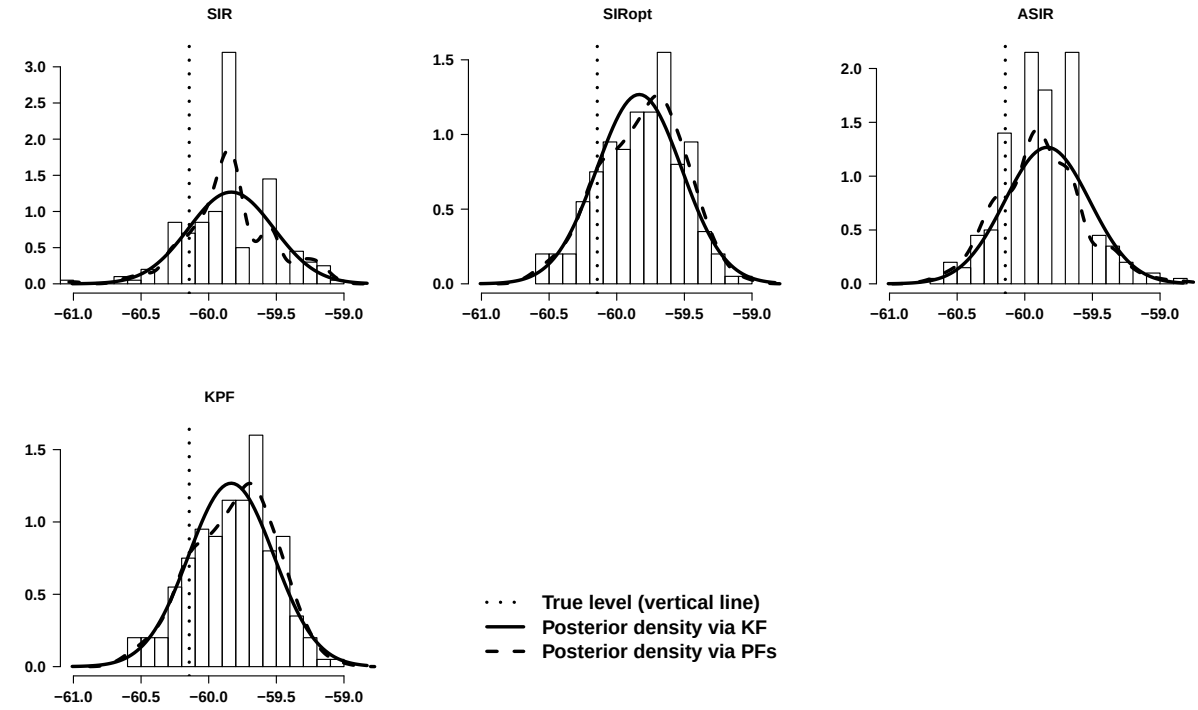


(b) Histogram (together with the estimated posterior density; black/dashed) of state values $\hat{x}_{T|T}$ for last data set. The exact posterior density obtained via the KF (black/continuous) is overlaid to each histogram.

Figure 3.4: Local level model: Case 9 with SNR $q = 1$ ($\sigma_\eta^2 = 0.1$ and $\sigma_\nu^2 = 0.1$)



(a) First row: Generated states and observations. Second row: Difference between estimated and true-state values $\hat{x}_{t|t} - x_t, t = 1, \ldots, T$.



(b) Histogram (together with the estimated posterior density; black/dashed) of state values $\hat{x}_{T|T}$ for last data set. The exact posterior density obtained via the KF (black/continuous) is overlaid to each histogram.

Figure 3.5: Local level model: Case 13 with SNR $q = 100$ ($\sigma_\eta^2 = 10$ and $\sigma_\nu^2 = 0.1$)

Remarks and Conclusions for $N_p = 200$ Particles

Based on simulation results reported in Table 3.2, considering only $N_p = 200$ particles, we make the following remarks and conclusions regarding the performance of the different filters under study when handling the local level model:

First, we refer to the effect of the initial-state-variance, used in the estimation procedure, on the statistical performance of the filters. To achieve that, we focus on the comparison of the respective mean-RMSE[3] estimates obtained at the two chosen initial state variances, $P_0 = \sigma_\eta^2$ and $P_0 = 100 \cdot \sigma_\eta^2$, and find that:

- For signal-to-noise-ratio values $q \leq 0.01$ (Cases 1–3)[4] all the filters under study are affected by the increase of the initial state variance used in the estimation procedure; that is, all filters accuse a reduction of the mean-RMSE when using $P_0 = 100 \cdot \sigma_\eta^2$ as the initial state variance instead of the true state noise variance value $P_0 = \sigma_\eta^2$. In all other cases with $0.01 < q < 100$ (Cases 4–13) there is practically no such effect.

- Therefore, based on the above results, when dealing with the local level model, in the future we choose to use a bigger value of the true initial state variance: in this case $P_0 = 100 \cdot \sigma_\eta^2$ is used. This somehow goes in line with the theory encountered in the literature which suggest the use of a diffuse prior when nothing is known about that variance (Shephard and Harvey 1990).

Second, we assess the impact of the signal-to-noise-ratio values q on the statistical performance of the competing filters. To achieve that, we focus on the comparison of the mean-RMSE attained at these different signal-to-noise-ratio settings and for the sake of simplicity and clarity we create Figure 3.6. This figure depicts on the upper panel the mean-RMSE as a function of the 13 chosen signal-to-noise-ratio values, and in the bottom panel the relative statistical performance of the competing filters in relation to the SIR particle filter variant measured by the ratio $\frac{RMSE(f)}{RMSE(SIR)}$, where $f \in$ {KF, SIRopt, ASIR, KPF} denotes a competing filter. Notice that although the values displayed in these plots correspond to the numerical results presented in second block of Table 3.2, that is when $P_0 = 100 \cdot \sigma_\eta^2$, the conclusions obtained when $P_0 = \sigma_\eta^2$ are in concordance with the former diffuse case. Thus, irrespective of the initial state variance (P_0) used in the estimation procedure, we observe that:

- Starting with Case 2 ($q = 0.001$), for each filter the mean-RMSE increases as the signal-to-noise-ratio value q increases. However, an apparently ill-behavior is attained at very low signal-to-noise-ratio values as we observe that, contrary to the expected, when the signal-to-noise-ratio value q increases from 0.0001 to 0.001, the mean-RMSE decreases; see upper panel of Figure 3.6. We will further comment on this point at the end of the section.

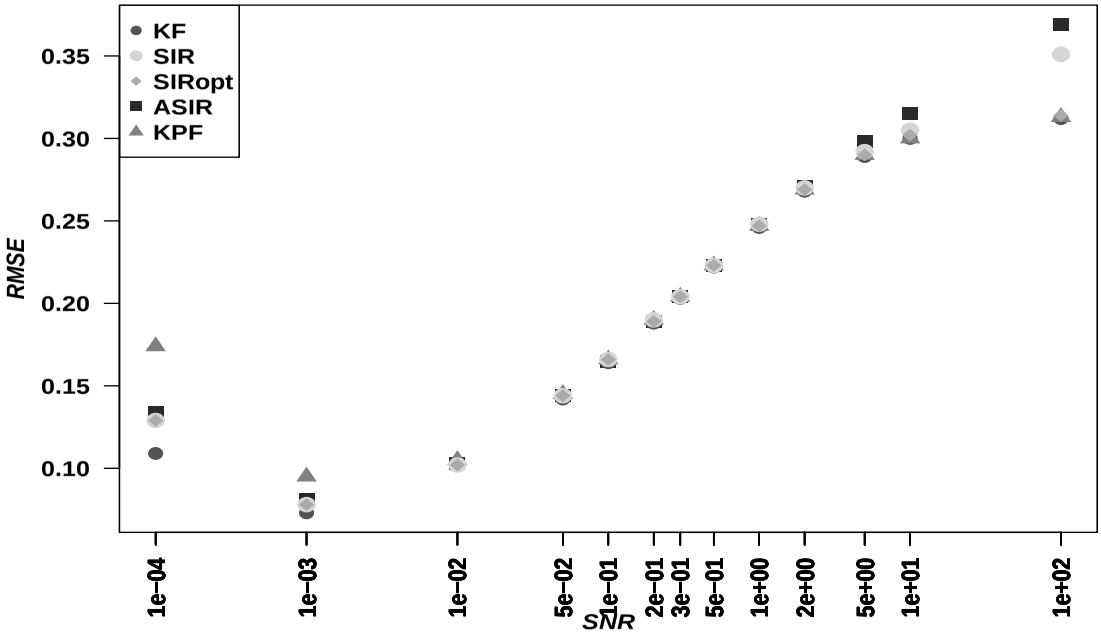
[3] Mean-RMSE and Mean(RMSE) used interchangeably

[4] In this section, whenever we refer to a case, it should be understood as a case included in Table 3.2

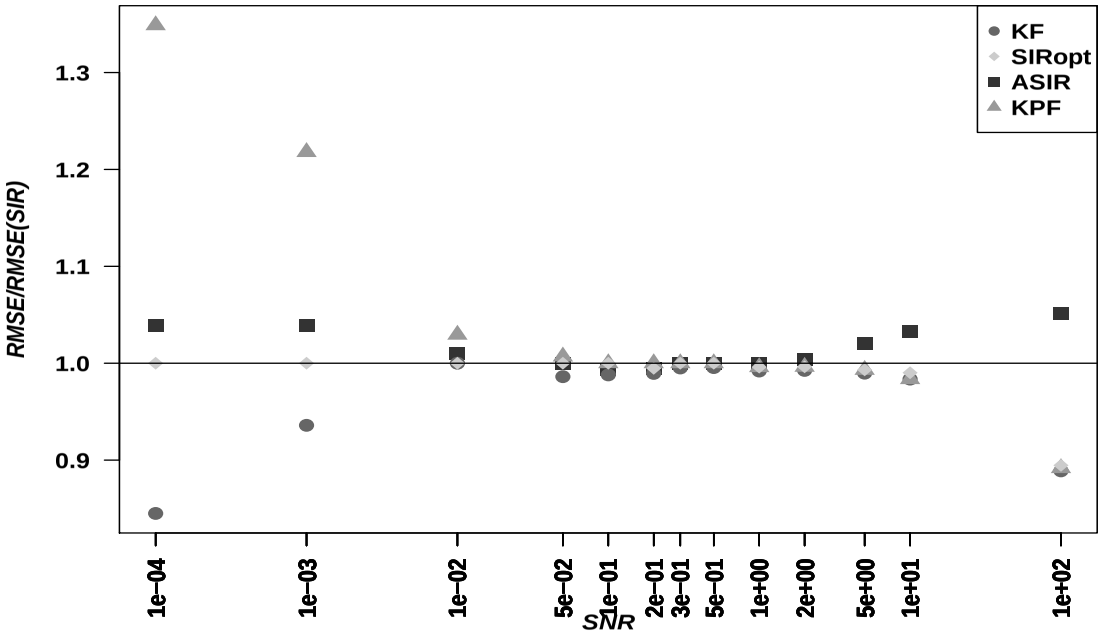
- Since the benchmark dynamic model is non-stationary but linear and Gaussian with known model parameters, this implies that a closed form analytical solution exists, which is optimal in terms of the RMSE. Further, being aware that this optimal solution is given by the KF, we decide to consider the KF as the gold standard filter. In other words, in what follows we will not only compare the different simulation based particle filters among themselves but also take as a reference the filtering performance attained by the gold standard KF.

- For the local level model at hand, we found that, as expected, among the competing filters the analytical KF provides the optimal filtering solution; that is, our simulation study confirms that the gold standard KF always yields the minimum RMSE; see Figure 3.6 where the dark circle (representing the KF) is always below or coincides with the other symbols (representing the four particle filter variants). We want to stress, however, that for the local level model the KF is not always able to adequately filter the state; that is, even the KF may provide unsatisfactory estimations of the mean level of the series as it is shown in Figure 3.3(a) corresponding to Case 1 with signal-to-noise-ratio value $q = 0.0001$. Something similar, though in a less degree, can be said for Case 2 with $q = 0.001$ (see Figure B.2 shown on website `http://www-eio.upc.edu/~lacosta/AppendixB.pdf` [last visited: September 2013].)

  After confirming that the KF yields the best possible filtering estimates of the level for the model at hand, our experimental results also indicate that –for a rather small number of particles $N_p = 200$– the particle filtering methodology is able to perform (nearly) as good as the gold standard KF at most signal-to-noise ratio values. Following, we discuss this point in detail for each signal-to-noise-ratio setting.



(a) For all filters: Mean(RMSE) vs signal-to-noise-ratio values



(b) Ratio of mean(RMSE)/mean(RMSE(SIR)) vs signal-to-noise-ratio values

Figure 3.6: Local level model: Impact of the signal-to-noise ratio value on the statistical performance of the filters indicated by the mean(RMSE); $T = 200$ and $N_p = 200$

  - For cases 1 and 2, corresponding to small signal-to-noise ratio values $q \in \{0.0001, 0.001\}$, the KPF variant shows the worst statistical performance and, as expected, the analytical KF yields the best one. The other three PF variants, SIR, SIRopt and ASIR, behave similarly among themselves with RMSE values greater than the one obtained via the KF; see upper panel of Figure 3.6 (and although not strictly necessary, for completion you may refer to Figure 3.3 and Figure B.2 corresponding to the signal-to-noise-ratio values $q = 0.0001$ and $q = 0.001$, respectively). We will further discuss this case at the end of this section.
  - For case 3 with signal-to-noise ratio value $q = 0.01$, all particle filters, except the KPF that shows the worse behaviour, have similar statistical performance to the gold standard KF; see Figure 3.6 (although not strictly necessary, for completion you may refer to Figure B.3).
  - For cases 4 and 5 with signal-to-noise ratio values $q \in \{0.05, 0.1\}$, all PF variants behave similarly among themselves with RMSE values that are slightly greater (practically the same) than the one obtained via the KF; see Figure 3.6 (although not strictly necessary, for completion you may refer to Figures B.4 and B.5).
  - For cases 6–10 with signal-to-noise ratio values $q \in \{0.2, 0.3, 0.5, 1, 2\}$, all PF variants achieve similar performance to the KF; see Figure 3.6 and Figure 3.4 (although not strictly necessary, for completion you may refer to Figures B.6–B.10).
  - For cases 11 and 12 with signal-to-noise ratio values $q \in \{5, 10\}$ the SIRopt and the KPF have similar statistical performance to the gold standard KF. The SIR and the ASIR show worse behaviour, specially the second, as they do not reach the KF's RMSE value; see Figure 3.6 (although not strictly necessary, for completion you may refer to Figures B.11–B.12).
  - Finally, for case 13 with high signal-to-noise ratio value $q = 100$ both the SIR and the ASIR, specially the second one, show worse performance. In this particular case, the best found particle filter variant is the KPF followed by the SIRopt with RMSE values similar to the one taken by the gold standard KF; see Figures 3.5 and 3.6, but mainly the second. We will further discuss this case at the end of this section.

- Therefore, based on above findings and results displayed in Table 3.2 and Figure 3.6, we conclude that the particle filtering methodology (with $N_p = 200$ particles) is able to reach a similar/equal statistical performance to the gold standard KF at moderate signal-to-noise-ratio values; discarding extremely low/high values. We observe that when the signal-to-noise-ratio takes an extremely high value $q = 100$, the mean-RMSE yields higher values for the SIR and ASIR PF variant; this explains partly the poor quality of the estimations obtained via these two filters in comparison with the rest of the filters. Additionally, as previously mentioned, for extremely small signal-to-noise-ratio values, say $q \in \{0.0001, 0.001\}$, it is (empirically) shown that all four particle filter variants display greater mean-RMSE values than the KF, being the KPF the worst. However, in these two cases all filters, including the gold standard KF, fail to provide an adequate solution for filtering the states; focus the attention on Figure 3.3(a).

Third, in order to compare the relative statistical performance of the different filters in relation to the SIR particle filter variant at different signal-to-noise ratio settings, we focus on the bottom panel of Figure 3.6 which represents the measure $\frac{RMSE(f)}{RMSE(SIR)}$ where $f \in$ {KF, SIRopt, ASIR, KPF} denotes a competing filter; values above one indicate worse performance in relation to the SIR particle filter variant. From it, we confirm all the results commented above about the impact of the signal-to-noise ratio on the statistical performance of the competing filters. Thus, the measure of the relative statistical performance of the filters in relation to the SIR PF variant allows us to conclude that:

- As expected, the non-simulation based KF shows best statistical performance, but all PF variants are able to practically equate KF's mean-RMSE, as described below.
- For cases 1 and 2 with low signal-to-noise ratio values $q \in \{0.0001, 0.001\}$, the KF has a better performance than the reference SIR particle filter variant, which itself has an equal statistical performance as the SIRopt and a better performance to the ASIR and the KPF; being the KPF the worst.
- For case 3, the competing PF variants SIRopt and the ASIR behave in a similar manner as the reference SIR PF variant, which equates KF's mean-RMSE. The KPF, however, shows a slight worse performance.
- For cases 4–10 with signal-to-noise ratio values $q \in \{0.05, 0.1, 0.2, 0.3, 0.5, 1, 2\}$ all competing filters –including the KF– show practically equal performance as the reference SIR PF variant.
- For cases 11-13 with higher signal-to-noise-ratio values $q \in \{5, 10, 100\}$ the competing PF variants SIRopt and the KPF equate KF's mean-RMSE outperforming the reference SIR PF variant, which itself outperforms the ASIR PF variant. Thus, in these three cases both the SIR and the ASIR have the worse performance; specially the latter at $q = 100$.

Fourth, we focus on exploring the impact of the SNR on the degeneracy problem which we know inherently affects particle filters. That is, analyzing the reported mean (SD) of the unique number of particles uNp at last time-index $t = T$, we aim to answer the question: Does the SNR play a role in the degeneracy problem? We find that the degeneracy behavior varies from filter to filter as a function of the signal-to-noise-ratio. These findings are summarized below:

- For the KPF particle filter variant, the mean(uNp) increases from about 36 to 193 as the signal-to-noise-ratio q increases from $q = 0.0001$ to $q = 100$; focus on the last column of Table 3.2. Of course, the higher the number of unique particles, the better.
- For the SIR and ASIR particle filter variants, contrary to what happens with the KPF, the mean(uNp) decreases as the signal-to-noise-ratio q increases from $q = 0.0001$ to $q = 100$. Specifically, in our simulation study, the mean(uNp) spans from about 194 to 30 for the SIR PF and from about 200 to 26 for the ASIR PF variant.

- For the SIRopt particle filter variant, a rather distinct pattern in the behavior of the unique number of particles is observed. In this situation, the mean(uNp) first decreases from about 194 to 165 as the signal-to-noise-ratio q increases from $q = 0.0001$ to $q = 0.5$. Then, the opposite happens, since we observe that the mean(uNp) increases from about 165 to 193 as the signal-to-noise-ratio q increases from $q = 0.5$ to $q = 100$. That is, a decreasing pattern is observed on mean(uNp) for signal-to-noise-ratio values q less than 1 and an increasing pattern for q greater than 1. We believe that an even more exhaustive study could be performed in the future to confirm all the aforementioned suggested results. For example: will this behavior be confirmed if one uses a higher number of particles or a greater time series length? Recall that the previous MC experiments only consider $T = 200$ observations and $N_p = 200$ particles.

- Therefore, what the attained results confirm is that in general, the SIRopt less suffers the degeneracy problem, that the KPF suffers it more at low signal-to-noise-ratio values (focus on first 3 cases) and that both the SIR and the ASIR particle filter variants are more affected by it at high signal-to-noise-ratio values (focus on last three cases; specially in the last one with $q = 100$).

- As a by-product, our results indicate that worse statistical performance is generally attained when the filters show more degeneracy. Thus, later on we will explore the effect of increasing the number of particles to help to prevent (or at least postpone) the degeneracy problem.

Fifth, focusing on the performance of the different filters in terms of the computational time, we conclude that, as expected, the KF is the less expensive algorithm with mean CPU time values around 0.09[0.01] (average [SD] time in seconds in handling a data set containing $T = 200$ observations), followed by the SIR (0.16 [0.02]), the SIRopt (0.19 [0.02]), the KPF (0.23 [0.02]) and the ASIR (0.30 [0.03]) filter. Thus, for $N_p = 200$ particles, clearly the KPF and the ASIR show worse computational performance. These CPU discrepancies among filters are greater when increasing the number of particles (N_p) and the time series length (T).

Next, we perform a small complementary study to further investigate the effect of the increase of the number of particles on the RMSE; we will also explore increasing the time series length from $T = 200$ to $T = 1000$.

3.3.4 Complementary Study: Increasing the Number of Particles and/or the Time Series Length

Herein, we study the impact of increasing the number of particles N_p, from a lowest value $N_p = 200$ to a highest value $N_p = 5000$, on the statistical performance of the filter. That is, for the local level model with time-series-length $T = 200$, we analyze the impact of increasing N_p on the $RMSE$ yielded by the different simulation based filters under study: the SIRopt, SIR, ASIR and KPF particle filter variants.

Exploring the Increase of the Number of Particles

As stated above, we aim to study the impact of increasing the number of particles on the mean-RMSE. To achieve that, we construct Figure 3.7 which shows the effect of increasing the number of particles, $N_p \in \{200, 500, 1000, 2000, 5000\}$, on the performance of the four particle filter variants under study. Notice that we consider only 8 (8 out of 13) representative[5] cases which, as known, are defined by the signal-to-noise ratio settings q in Table 3.2. Thus, based on obtained results plotted in Figure 3.7 the following conclusions arise:

In general, the statistical performance of all four particle filters are positively affected by the increase of the number of particles. That is, for each particle filter the increment of N_p leads to a decrease of the mean-RMSE, though in many cases that effect is practically unnoticeable. We observe that as N_p increases, the mean-RMSE approaches Kalman Filter's mean-RMSE benchmark value. However, it becomes also clear from this plot that at extreme low and high signal-to-noise-ratio values this impact is not that marked since some particle filters are still not able to reach KF's statistical performance. Following, we provide a detailed summary of our findings:

- For case 1 with small signal-to-noise-ratio value $q = 0.0001$, though all particle filter variants are affected by the increase of the number of particles, the estimated mean-RMSE are always (slightly) greater than the one yielded by the KF; as depicted in Figure 3.7. In this case, the KPF variant shows worse performance, followed by the ASIR. Indeed, the ASIR is only able to reach the statistical performance of the competing particle filter variants (SIR and SIRopt) with $N_p = 5000$. The KPF, however, with (and up to) 5000 particles (we even tried using $N_p = 20000$), is still not able to reach the statistical performance of the other particle filter variants nor the one of the KF algorithm.

- For case 2 with signal-to-noise-ratio value $q = 0.001$, all particle filter variants, except the KPF, show similar statistical performance as the gold standard KF starting with $N_p = 500$ particles; the KPF filter only achieves this when $N_p = 5000$ is used.

[5] A similar pattern as the one one observed in the included Case 6 with $q = 0.2$ is also obtained in the excluded cases with SNR values $q \in \{0.05, 0.1, 0.3, 0.5, 2\}$. Notice that in such cases, the impact of the number of particles over the statistical performance (mean-RMSE) is rather unnoticeable.

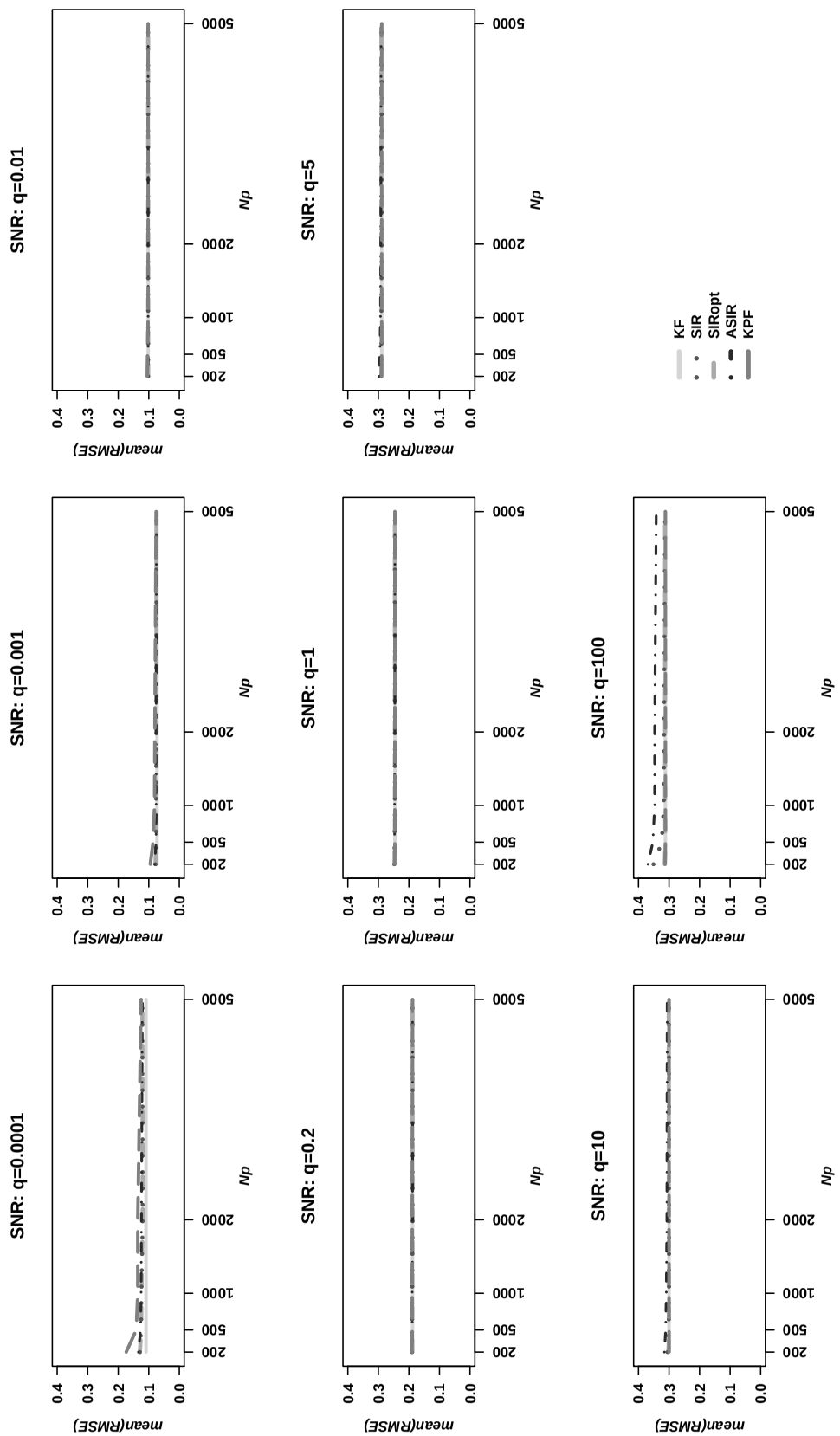


Figure 3.7: Local level model: Effect of the number of particles over the mean-RMSE; fixed $T = 200$

- For cases 3-12 with signal-to-noise-ratio values q ranging from 0.01 to 10, the increase of the number of particles has practically no effect for all studied particle filter variants. We remark that in these cases the mean-RMSE values obtained via the different particle filter variants practically coincide with the one obtained via the gold standard KF already with $N_p = 500$.

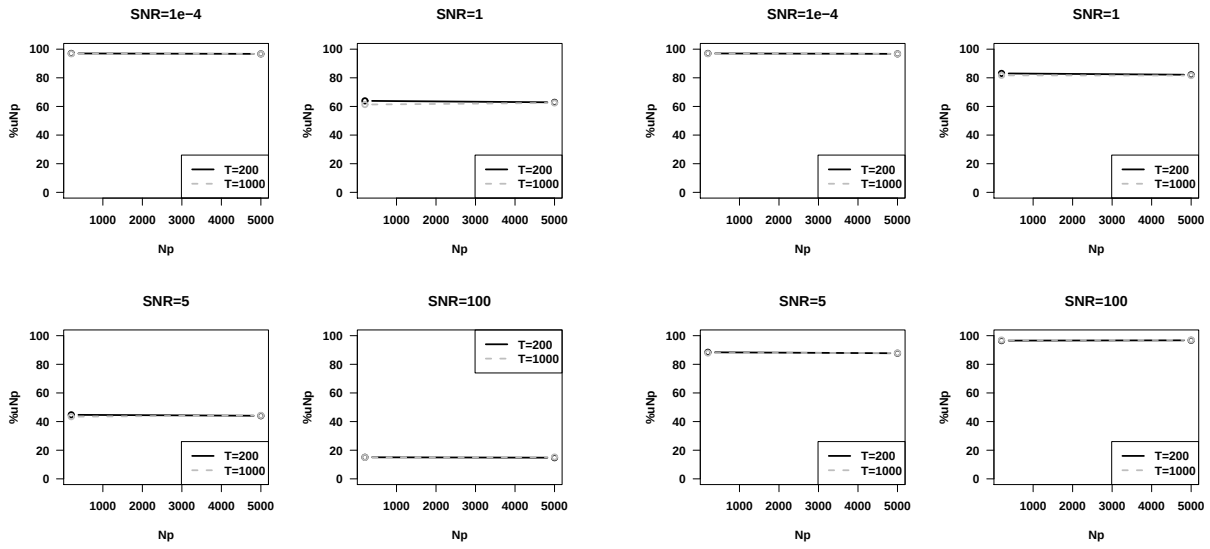
- For case 13 with high signal-to-noise-ratio value $q = 100$, the increase of the number of particles affects markedly only the SIR and ASIR particle filters but in a distinct way; up to $N_p = 5000$ particles, both filters accuse a decrease on their mean-RMSE but the ASIR never equates KF's performance. Specifically, for the ASIR, increasing the number of particles from $N_p = 200$ to $N_p = 500$ produces a decrease of the mean-RMSE but a further increase of the number of particles (effect shown up to 5000 particles, but we even try $N_p = 20000$) has practically no effect and never reach a similar statistical performance to any other PF variant nor to the KF; also a higher number of particles would obviously produce an increase in both, memory and CPU-time requirements. The SIR PF variant, however, achieves similar statistical performance to the KF starting with $N_p = 500$ and equates it with $N_p = 5000$ particles. In contraposition, the other two filters, the SIRopt and the KPF, show a similar performance to the gold standard KF with a low number of particles $N_p = 200$ and equates it with just $N_p = 500$ particles. Thus, according to our simulation results, in this case the ASIR PF variant shows worse statistical performance.

After finding the minimum number of particles needed to achieve a similar/equal statistical performance to the gold standard KF when dealing with the local level model, another question remains: Is the estimated posterior marginal density obtained with this minimum found number of particles a reliable posterior? We think this is closely related to the degree of degeneracy observed in the particle filters. Next, we will explore the effect of increasing the numbers of particles on degeneracy. Also, we will assess the degree of degeneracy when using a higher time series length.

Exploring the Increase of both the Number of Particles and the Time Series Length

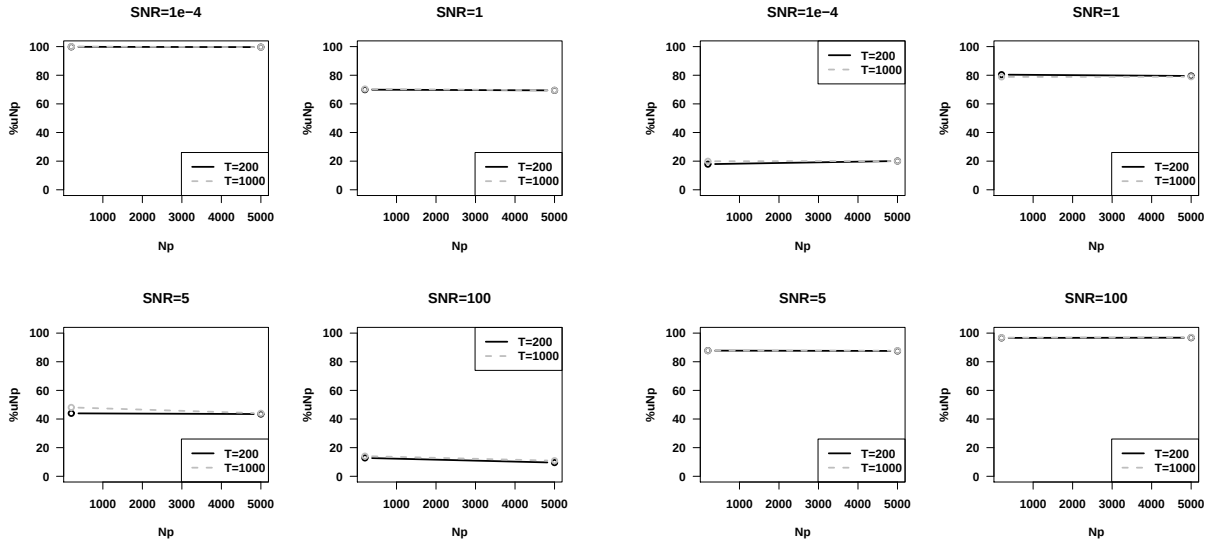
In a previous simulation study, with time series length $T = 200$ and $N_p = 200$ particles, we found that in general, the SIRopt less suffers the degeneracy problem, that the KPF suffers it more at low signal-to-noise-ratio values (focus on first 3 cases in Table 3.2) and that both the SIR and the ASIR particle filter variants are more affected by it at high signal-to-noise-ratio values (focus on last three cases; specially in the last one with $q = 100$). Those simulation results suggested that the competing particle filter variants show worse statistical performance when more degeneracy is present.

Following, we explore further the effect of increasing the time series length as well as the number of particles on degeneracy. Our aim is twofold: to verify if the general findings obtained using $T = 200$ and $N_p = 200$ are confirmed and also to prevent (or at least postpone) the degeneracy problem with the hope of consequently improving the statistical performance of the filters. To achieve these goals,



(a) Percentage number of unique particles under SIR

(b) Percentage number of unique particles under SIRopt



(c) Percentage number of unique particles under ASIR

(d) Percentage number of unique particles under KPF

Figure 3.8: Local level model: Percentage of unique number of particles at time index $t = T$ (T=200, black/continuous; $T = 1000$, grey/dashed) in relation to the original number of particles $N_p \in \{200, 5000\}$ obtained by the four competing particle filter variants (Top left: SIR, Top right: SIRopt, Bottom left: ASIR and Bottom right: KPF) at selected signal-to noise-ratio settings $q \in \{1e-4, 1, 5, 100\}$.

we construct Figure 3.8, which basically represents in the y-coordinate the percentage of unique number of particles (%uNp) and in the x-coordinate the chosen settings for the original number of particles (N_p). For simplicity, we show results for only four signal-to-noise-ratio settings $q \in \{1\text{e-}4, 1, 5, 100\}$, two time series length settings $T \in \{200, 1000\}$ and two number of particles settings $N_p \in \{200, 5000\}$. Thus, Figure 3.8 includes a total of 16 individual plots organized in four sub-figures; each sub-figure corresponds to a type of filter (four). Each sub-figure itself contains four different plots representing for each one of the four chosen signal-to-noise-ratio settings the $2 \cdot 2 = 4$ values of the mean-percentage number of unique particles that are obtained at the combination of the two time series length settings and two number of particles settings.

The sub-figures in Figure 3.8 allow us to confirm the previously stated results for $T = 200$ and $N_p = 200$. Further, we find out that this behavioral pattern holds regardless of the number of particles and time series length. In other words, the SIRopt continues suffering in less degree the degeneracy problem (top right sub-figure), the KPF suffers it more at low signal-to-noise-ratio values (focus on bottom right sub-figure and case: $q = 1e-4$) and both the SIR and the ASIR particle filter variants are more affected by it at high signal-to-noise-ratio values (focus on top left and bottom left sub-figures and case: $q = 100$). Since we want to assess the effect increasing the number of particles and the time series length using four signal-to-noise-ratio settings per filter, a detailed description of the degeneracy related performance of the four competing particle filter variants is given below:

- First, both the SIR and ASIR show a rather similar pattern on their performance as a function of the signal-to-noise-ratio setting: the number of unique particles decreases as the signal-to-noise-ratio increases; the former filter has a worst case scenario at the high signal-to-noise-ratio $q = 100$ where approximately a constant percentage of unique particles (about 15%) is obtained for any value of T and N_p; see top left sub-figure in Figure 3.8. Something similar can be said for the ASIR, but in this case the percentage of unique particles declines from about 14% to 10% when increasing N_p from 200 to 5000 particles. Both filters show best performance at the lowest signal-to-noise-ratio $q = 1e-4$, where the percentage of unique particles is about 97% irrespective of the values of T and N_p.

- On the contrary, the KPF has worst degeneracy related performance at low signal-to-noise-ratio $q = 1e-4$ with about 18% of unique particles at $N_p = 200$ that increases slightly to about 20% as N_p increases to 5000 particles; practically, no differences are observed for the two values of the time-series length T. This filter best-case scenario occurs at high signal-to-noise-ratio $q = 100$ with approximately constant percentage of number of particles (about 97%) obtained for any T and N_p.

- Finally, the SIRopt, that shows in general less degree of degeneracy, exhibits a distinct behavioral pattern for signal-to-noise-ratio values less than 1 and for signal-to-noise-ratio values greater than one. Indeed, this filter best case scenario occurs at extremely low $q = 1e-4$ and high $q = 100$ signal-to-noise-ratio values where approximately a constant percentage of number of particles

(about 97%) is obtained for any T and N_p; its worst case scenario occurs around $q = 1$ where practically a constant percentage of number of particles (about 82%) is obtained for any T and N_p.

- We find these results very encouraging as they suggest that, for the local level model at hand and the particle filters studied, the time series length barely affects the percentage of unique number of particles obtained at last time-index $t = T$, and that the used number of particles N_p also has only a slight effect on degeneracy given a signal-to-noise-ratio value; see bottom left subfigure showing that the ASIR displays a small decrease on the percentage of unique particles at high SNR when N_p increases from $N_p = 200$ to $N_p = 5000$. In general, however, these percentages remain practically stable irrespective of the value of T and N_p. Naturally, as the number of particles used in the estimation procedure increases, the absolute number of unique number of particles also increases.

- As a by-product, for the local level at hand, these results suggest that if we have prior information about the relative variation present in our data, we could somehow decide a minimum number of particles so as to avoid degeneracy. The pattern found on the behavior of the percentage number of unique number of particles is a very nice result since we are aware of the importance in dealing with the degeneracy problem within the particle filtering methodology.

  To further illustrate the role played by the time series length T and the number of particles N_p used in the estimation procedure on the mean-RMSE and the computational cost (CPU-time in seconds) of competing filters, we present Table 3.3 summarizing the simulation results corresponding to a sub-study (Case 5 with signal-to-noise-ratio $q = 0.1$). This table reports measures of the RMSE, the CPU-time, and the percentage of unique particles at last time-index $t = T$, where the used number of particles are $N_p \in \{200, 500, 1000, 5000\}$. The estimation results yielded by the exact KF algorithm using $T \in \{200, 500, 1000, 2000\}$ are taken as a reference and they are given by mean-RMSE = $\{0.165, 0.165, 0.165, 0.164\}$ and mean CPU-time = {0.09, 0.23, 0.49, 0.88}, respectively. Although this table provides the results for the particular case with $q = 0.1$, it perfectly illustrates the general pattern observed on the results yielded at other SNR settings. That is, given a signal-to-noise-ratio value, it seems that for the local level model at hand and particle filters studied, the mean-percentage of unique particles remains stable irrespective of the time series length T and number of particles N_p used in the estimation procedure.

- As known, all particle filters suffer the degeneracy problem and our results do not contradict that fact. Our contribution herein is on describing in detail what happens in particular situations – characterized by the chosen dynamic model and varied simulation settings– in order to provide some guidelines for the practitioner interested in using the studied filters. Thus, for the local level model, putting together the statistical performance in terms of RMSE and a measure of degeneracy given by the percentage of unique number of particles %uNp, we recommend as a rule of thumb to use $N_p = 5000$ particles irrespective of the particle filter variant and the signal-to-

noise-ratio; discarding from this generalization extremely low/high signal-to-noise-ratio values. Notice that if we focus on the degeneracy related performance, the choice of $N_p = 5000$ particles would yield about 500 (that is, 10%) unique particles in the worst of the worse case scenarios, what we think is a reasonable enough amount of particles to produce a reliable marginal posterior representation of the states. When only interested in a particular particle filter variant, the reader can refer to previously presented specific remarks indicating that even a smaller number of particles could be appropriate for obtaining a reliable posterior.

Table 3.3: Summary of MC Sub-study (Case 5 with $q = 0.1$, representative of most cases): Illustrating the role of the number of particles and/or the time series length on the mean-RMSE and the computational cost (CPU-time in seconds) of competing filters. For particle filters, the degree of degeneracy is also reported where the used number of particles are $N_p \in \{200, 500, 1000, 5000\}$. For reference, use the results yielded by the exact KF algorithm at $T \in \{200, 500, 1000, 2000\}$, which are given by: mean-RMSE = {0.165, 0.165, 0.165, 0.164} and mean-CPU-time(SD) = {0.091(0.011), 0.227(0.014), 0.489(0.020), 0.884(0.028)}.

			$N_p = 200$		$N_p = 500$		$N_p = 1000$		$N_p = 5000$	
Filter	**T**	**Criterion**	**Mean**	**Var**	**Mean**	**Var**	**Mean**	**Var**	**Mean**	**Var**
SIRopt	**200**	**RMSE**	0.166	2e-4	0.165	2e-4	0.165	2e-4	0.164	2e-4
		%UNp	84.61	9.43	84.02	9.78	84.04	9.7	84.14	9.75
		CPU	0.186	0.022	0.255	0.024	0.412	0.036	2.081	0.05
	500	**RMSE**	0.165	1e-4	0.165	1e-4	0.165	1e-4	0.165	1e-4
		%UNp	83.03	10.1	82.76	10.05	82.93	9.91	83.2	9.82
		CPU	0.424	0.014	0.634	0.032	1.028	0.076	4.53	0.111
	1000	**RMSE**	0.165	1e-4	0.165	1e-4	0.165	1e-4	0.165	1e-4
		%UNp	83.44	10.23	83.88	9.91	83.91	9.92	83.44	10.18
		CPU	0.954	0.041	1.261	0.054	1.994	0.093	8.938	0.083
	2000	**RMSE**	0.164	1e-6	0.164	1e-6	0.164	1e-6	0.164	1e-6
		%UNp	84.39	9.64	83.37	10.21	83.46	10.06	83.64	9.93
		CPU	1.716	0.036	2.567	0.12	3.599	0.093	14.873	0.182
SIR	**200**	**RMSE**	0.166	2e-4	0.165	2e-4	0.165	2e-4	0.164	2e-4
		%UNp	81.22	10.88	80.66	11.18	80.76	11.11	80.71	11.1
		CPU	0.164	0.025	0.221	0.021	0.362	0.038	1.732	0.058
	500	**RMSE**	0.165	1e-4	0.165	1e-4	0.165	1e-4	0.165	1e-4
		%UNp	79.91	11.35	79.18	11.49	79.41	11.28	79.48	11.26
		CPU	0.389	0.019	0.558	0.029	0.871	0.056	3.668	0.043

Table 3.3: Summary of MC Sub-study (Case 5 with $q = 0.1$, representative of most cases): Illustrating the role of the number of particles and/or the time series length on the mean-RMSE and the computational cost (CPU-time in seconds) of competing filters. For particle filters, the degree of degeneracy is also reported where the used number of particles are $N_p \in \{200, 500, 1000, 5000\}$ (continued).

			$N_p = 200$		$N_p = 500$		$N_p = 1000$		$N_p = 5000$	
			Mean	**Var**	**Mean**	**Var**	**Mean**	**Var**	**Mean**	**Var**
	1000	**RMSE**	0.166	1e-4	0.165	1e-4	0.165	1e-4	0.165	1e-4
		%UNp	79.61	11.73	79.04	11.73	80.05	11.57	80.03	11.54
		CPU	0.873	0.036	1.103	0.034	1.674	0.036	7.291	0.098
	2000	**RMSE**	0.164	1e-6	0.164	1e-6	0.164	1e-6	0.164	1e-6
		%UNp	80.86	11.35	79.81	11.38	79.73	11.49	79.88	11.47
		CPU	1.565	0.038	2.201	0.055	3.064	0.035	11.996	0.148
ASIR	**200**	**RMSE**	0.165	2e-4	0.165	2e-4	0.165	2e-4	0.164	2e-4
		%UNp	89.95	5.32	89.78	5.25	89.97	5.3	89.9	5.35
		CPU	0.296	0.03	0.384	0.028	0.664	0.039	3.999	0.156
	500	**RMSE**	0.166	1e-4	0.165	1e-4	0.165	1e-4	0.165	1e-4
		%UNp	88.64	5.98	90.06	5.41	89.31	5.59	89.55	5.39
		CPU	0.644	0.043	0.991	0.036	1.513	0.045	8.204	0.241
	1000	**RMSE**	0.165	1e-4	0.165	1e-4	0.165	1e-4	0.165	1e-4
		%UNp	90.08	5.75	89.33	5.88	89.7	5.62	89.71	5.53
		CPU	1.407	0.042	1.951	0.56	3.048	0.402	15.348	0.102
	2000	**RMSE**	0.164	1e-6	0.164	1e-6	0.164	1e-6	0.164	1e-6
		%UNp	88.28	6.28	88.84	5.97	89.9	5.37	89.61	5.55
		CPU	2.539	0.039	3.722	0.058	5.438	0.052	22.88	0.167
KPF	**200**	**RMSE**	0.166	2e-4	0.165	2e-4	0.165	2e-4	0.164	2e-4
		%UNp	66.46	8.79	65.25	8.07	65.14	7.93	65.38	8.01
		CPU	0.232	0.02	0.325	0.134	0.52	0.035	3.184	1.06
	500	**RMSE**	0.166	1e-4	0.165	1e-4	0.165	1e-4	0.165	1e-4
		%UNp	64.56	8.21	66.02	9.09	64.87	8.39	65.01	8.17
		CPU	0.524	0.027	0.785	0.057	1.231	0.061	5.617	0.087
	1000	**RMSE**	0.166	1e-4	0.165	1e-4	0.165	1e-4	0.165	1e-4
		%UNp	64.52	8.19	65.19	8.51	65.05	8.11	65.32	8.59
		CPU	1.135	0.029	1.522	0.081	2.369	0.096	11.217	0.101
	2000	**RMSE**	0.165	1e-6	0.164	1e-6	0.164	1e-6	0.164	1e-6
		%UNp	63.78	9.25	65.54	9.13	65.28	8.67	65.13	8.55
		CPU	2.045	0.031	2.979	0.101	4.314	0.103	17.491	0.07

The simulation results portrayed in Table 3.3 are useful to further illustrate the impact of increasing the number of particles and/or the time Series over the mean-RMSE and CPU-times. The impact of increasing the number of particles on the RMSE is already described and illustrated in Figure 3.7, but are also being confirmed by the results in the above-mentioned table: considering alone the impact of increasing N_p generally does not report relevant advantages over the yielded RMSE values, except at extreme SNR settings. Likewise, the increase of the time series length is generally related to a decrease on the mean-RMSE, but this decrease is rather unnoticeable for the local level model at hand and all filters (including the KF) studied. Additionally, to illustrate how the CPU-times are affected by an increase of the number of particles and the time series length we construct Figure 3.9, confirming that overall the least expensive algorithm is the KF followed by the SIR, the SIRopt, the KPF and the ASIR; naturally the obtained CPU-time discrepancies among filters are greater when increasing the number of particles (N_p) and/or the time series length (T).

To conclude this section, we provide some comments about the peculiar estimation behavior observed at the two extreme signal-to-noise-ratio scenarios $q = 0.0001$ and $q = 100$:

As repeatedly commented on in the section, the two cases (case 1 with $q = 0.0001$ and case 13 with $q = 100$) always present a peculiar behavior, specially for certain filters like the ASIR and the KPF. Following, we further discuss these two cases. That is, we aim to explain the apparently ill-behavior attained at extremely low ($q = 0.0001$) and extremely high ($q = 100$) signal-to-noise-ratio values. To achieve this, we remit ourselves to the theoretical result $m_t = Ay_t + (1 - A)m_{t-1}$ given in West and Harrison (1989), which basically states that the estimated state value at current time, t, depends on the weighted average of the current observation and the previous estimated state. Recall that the weight attached to the observation, denoted by A, is the value of the adaptive coefficient that defines the rate of adaptation to new data. In the sixth column of Table 3.1, we report the computed adaptive coefficients corresponding to the 13 signal-to-noise-ratio simulation settings used for the local level model.

Referring to Case 1, with very small signal-to-noise-ratio value $q = 0.0001$, all previously presented simulation results repeatedly suggest an apparently ill-behavior: for instance, it was found that all filters –including the gold standard KF– might fail to adequately filter the level x_t. In this particular case, the adaptive coefficient takes the value $A = 0.01$ meaning that the observations have practically null contribution to update the value of the level x_t and that we are practically dealing with white noise; indeed, in this case, the best level estimate m_t will be given by the previous estimated level m_{t-1}. On the contrary, for Case 13 with very high signal-to-noise-ratio value $q = 100$, the corresponding adaptive coefficient takes the value $A = 0.99$, meaning that the observations are very informative when updating the level x_t and that we are practically dealing with observations evolving as a random walk; in this case the best level estimate m_t will be given by the current observation y_t and as stated by West and Harrison (1989), the model is of little use for prediction.

According to West and Harrison (1989), the major number of applications of the local level model with constant variances (called by them: constant model) are in short term forecasting and control with typical signal-to-noise-ratios spanning from 0.001 to 0.2. Also, applications recently found in the

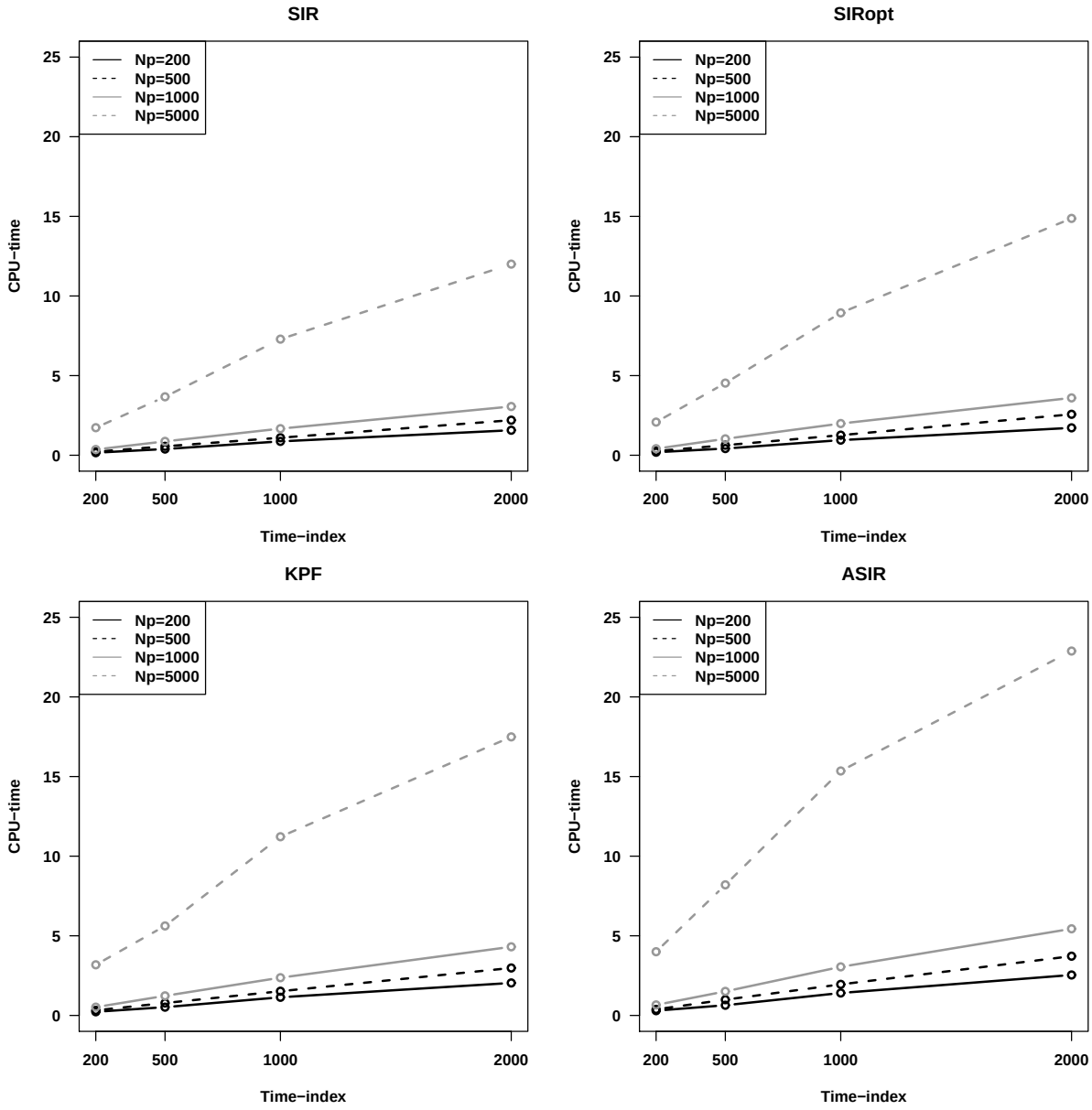


Figure 3.9: Local level model (Case 5 with $q = 0.1$ but representative of all cases): Behavior of the estimated mean-CPU-elapsed time (in seconds) for the SIR, SIRopt, KPF and the ASIR PF variants. Assessment of both, the impact of the time series length (x-axis) and the number of particles. The CPU-times yielded by the exact KF algorithm using various time series lengths $T \in \{200, 500, 1000, 2000\}$ are given by CPU-time = {0.09, 0.23, 0.49, 0.88}.

literature deal with signal-to-noise-ratio values around 0.2, 0.3, 1 and 2; see for instance Stock and Watson (2007) and Pellegrini (2009). In such cases, according to our Monte Carlo studies, any of the particle filter variants with only $N_p = 500$ particles could be used, except for the KPF with $SNR = 0.001$, as they all reach a similar statistical performance to the gold standard KF. In case of the KPF

with $SNR = 0.001$, about $N_p = 5000$ particles would be required to achieve such performance. Thus, the previously given rule of thumb (use $N_p = 5000$ particles) clearly holds in these typically found scenarios irrespective of the particle filter used. For cases corresponding to other signal-to-noise-ratio values, refer to the comments given in previous remarks.

The next section, deals with a second simulation study regarding the estimation of the states of the stationary AR(1) plus noise model. Notice that the implementation of this is straightforward once we have developed the R language program for the local level model, since we must only change the value of the autoregressive parameter ϕ. Since our main interest is on studying non-standard state-space dynamic models, only the non-stationary dynamic local level model will be revisited in the second part of this PhD thesis when estimating simultaneously the states and parameters.

3.4 Simulation Study II: The Stationary AR(1) plus noise Model

This Monte Carlo study proceeds as simulation study I, but in this case the stationary linear and Gaussian dynamic model called AR(1) plus noise is used as a benchmark. In other words, a comparison of the performance of the studied particle filter variants with the gold standard filter KF is carried out. Following, the state-space formulation for the so-called contaminated AR(1) model is given.

3.4.1 State Space Representation

Recall that the equations (3.1) and (3.2) with $|\phi| < 1$, specify the state-space formulation of the linear, Gaussian and stationary dynamic model known as AR(1) plus noise model. In that case, the parametric state-space formulation for this dynamic model can be described and subsumed by the following two equations; see, for instance, Gómez and Maravall (1994) and Shumway and Stoffer (2000):

$$\begin{aligned} x_t &= \phi x_{t-1} + \eta_t \\ y_t &= x_t + \nu_t \end{aligned} \tag{3.10}$$

To complete the state-space formulation, we must assume a distribution for the initial state variable x_0. In this particular case, $x_0 \sim N(\mu_{x_0}, \Sigma_{x_0})$, ϕ is the autoregressive parameter, and the noise disturbances η_t and ν_t are assumed to follow a Gaussian distribution with fixed and known variance parameters $\sigma^2_{\eta_t}$ and $\sigma^2_{\nu_t}$; thus only the states x_t are estimated.

Similarly to the local level, the reduced form of the AR(1) plus noise model is an ARIMA process. Following, we explicitly describe the relationship between a traditional ARIMA process and the so-called contaminated AR(1) model.

3.4.2 Reduced Form of the Local level model: an ARIMA(1,0,1) Model

The reduced form of the contaminated AR(1) model is an ARIMA(1,0,1) process; this means that the AR(1) plus noise model can be represented as a stationary and invertible ARIMA(1,0,1) model. Fol-

lowing, the explicit relationship between a traditional ARIMA process and the so-called contaminated AR(1) model is described.

Departing from the state-space model of the contaminated AR(1) model specified in equation (3.10), by substituting back the state equation into the measurement equation, simple algebra and then using the definition of the measurement equation (3.10) we obtain

$$\begin{aligned} y_t &= x_t + \nu_t \\ &= (\phi x_{t-1} + \eta_{t-1}) + \nu_t \\ &= (\phi x_{t-1} + (\theta + \phi)\nu_{t-1}) + \nu_t, \text{ with } \eta_{t-1} = (\theta + \phi)\nu_{t-1} \\ &= (\phi(x_{t-1} + \nu_{t-1}) + \theta\nu_{t-1}) + \nu_t \\ &= (\phi y_{t-1} + \theta\nu_{t-1}) + \nu_t \end{aligned}$$

which clearly corresponds to the expression of the ARIMA(1,0,1) process with $\eta_{t-1} = (\theta + \phi)\nu_{t-1}$ under suitable values for the moving average parameter θ.

3.4.3 Results, Remarks and Conclusions for Simulation Study II

Specific Simulation settings

In this MC study we consider the same general settings used in simulation I which are provided on page 55 of Section 3.2.4, but as already stated there, this second simulation experiment assumes a diffuse prior for the initial state x_0; specifically the state initial variance is set to 100 times its true value, say $P_0 = 100 \cdot \sigma_\eta^2$. Since the AR(1) plus noise model also includes the autoregressive parameter ϕ, two extra simulation scenarios are defined depending on the two chosen settings for that parameter; in this case, we consider the models with autoregressive parameters $\phi = 0.3$ and $\phi = 0.8$, respectively.

Experimental Results

Herein, in Table 3.4 we provide the numeric results which summarize and assess the performance of the five different filters under study when handling the AR(1) plus noise model. This table is organized in two different blocks that correspond to the two estimation settings used for the autoregressive parameter, say $\phi = 0.3$ and $\phi = 0.8$. Each one of these blocks is composed itself by two sub-blocks where the first contains the measures: Mean(RMSE) and Var(RMSE); the second contains measures of the variable uNp: Mean(SD). Recall that uNp means 'unique number of particles at the end time period $t = T$' and it was created to somehow quantify the potential degeneracy problem of particle filters. Notice that we also report results for the non-simulation based KF, which is optimal and used as a benchmark filter.

All the obtained simulation results are reported in Table 3.4 displayed below:

Table 3.4: Summary of simulation study II under 13 different settings: $\sigma_v^2 = 0.1$, $T = 200$, and $N_p = 200$.

		$\phi = 0.3$			$\phi = 0.8$		
		RMSE		uNp	RMSE		uNp
Setting	Filter	Mean	Var	Mean(SD)	Mean	Var	Mean(SD)
Case 1 $\sigma_\eta^2 = 0.00001$, SNR = 0.0001							
	KF	0.003	$< 1e-06$	—	0.005	$< 1e-06$	—
	SIRopt	0.003	$< 1e-06$	159.85 (1.96)	0.005	$< 1e-06$	178.28 (1.89)
	SIR	0.003	$< 1e-06$	199.25 (0.61)	0.005	$< 1e-06$	198.87 (0.90)
	ASIR	0.003	$< 1e-06$	199.44 (0.67)	0.005	$< 1e-06$	199.38 (1.13)
	KPF	0.003	$< 1e-06$	195.99 (0.10)	0.005	$< 1e-06$	152.69 (2.72)
Case 2 $\sigma_\eta^2 = 0.0001$, SNR = 0.001							
	KF	0.010	$< 1e-06$	—	0.017	$< 1e-06$	—
	SIRopt	0.010	$< 1e-06$	160.45 (2.23)	0.017	$< 1e-06$	178.48 (2.00)
	SIR	0.010	$< 1e-06$	197.91 (1.66)	0.017	$< 1e-06$	196.79 (2.52)
	ASIR	0.010	$< 1e-06$	198.07 (1.82)	0.017	$< 1e-06$	198.08 (2.22)
	KPF	0.010	$< 1e-06$	195.85 (0.39)	0.017	$< 1e-06$	152.65 (2.45)
Case 3 $\sigma_\eta^2 = 0.001$, SNR = 0.01							
	KF	0.033	$< 1e-06$	—	0.052	$< 1e-06$	—
	SIRopt	0.033	$< 1e-06$	160.47 (1.90)	0.052	$< 1e-06$	177.03 (3.54)
	SIR	0.033	$< 1e-06$	193.66 (4.95)	0.052	$< 1e-06$	190.10 (7.52)
	ASIR	0.033	$< 1e-06$	193.78 (5.25)	0.052	$< 1e-06$	193.79 (5.08)
	KPF	0.033	$< 1e-06$	195.33 (1.36)	0.052	$< 1e-06$	153.96 (3.05)
Case 4 $\sigma_\eta^2 = 0.005$, SNR = 0.05							
	KF	0.072	$< 1e-06$	—	0.104	1e-04	—
	SIRopt	0.072	$< 1e-06$	160.20 (3.24)	0.104	1e-04	172.7 (9.01)
	SIR	0.072	$< 1e-06$	185.87 (10.42)	0.104	1e-04	179.26 (14.19)
	ASIR	0.072	$< 1e-06$	186.32 (10.31)	0.104	1e-04	185.68 (10.02)
	KPF	0.072	$< 1e-06$	194.11 (3.58)	0.104	1e-04	157.74 (8.59)
Case 5 $\sigma_\eta^2 = 0.01$, SNR = 0.1							
	KF	0.099	$< 1e-06$	—	0.133	1e-04	—
	SIRopt	0.099	$< 1e-06$	159.93 (4.42)	0.133	1e-04	170.8 (11.82)
	SIR	0.099	$< 1e-06$	180.18 (14.04)	0.134	1e-04	172.11 (17.77)
	ASIR	0.099	$< 1e-06$	180.56 (13.84)	0.133	1e-04	180.26 (12.02)
	KPF	0.099	$< 1e-06$	193.40 (3.97)	0.134	1e-04	159.29 (11.44)

Table 3.4: Summary of simulation study II under 13 different settings: $\sigma_\nu^2 = 0.1$, $T = 200$, and $N_p = 200$ (continued).

		$\phi = 0.3$			$\phi = 0.8$		
		RMSE		uNp	RMSE		uNp
Setting	Filter	Mean	Var	Mean(SD)	Mean	Var	Mean(SD)
Case 6 $\sigma_\eta^2 = 0.02$, SNR = 0.2							
	KF	0.133	1e-04	—	0.165	1e-04	—
	SIRopt	0.133	$< 1e-06$	160.78 (6.59)	0.165	1e-04	168.06 (14.03)
	SIR	0.133	1e-04	172.01 (18.25)	0.166	1e-04	163.24 (21.61)
	ASIR	0.133	1e-04	172.65 (17.90)	0.165	2e-04	172.01 (15.63)
	KPF	0.133	1e-04	193.33 (4.54)	0.166	1e-04	162.34 (14.42)
Case 7 $\sigma_\eta^2 = 0.03$, SNR = 0.3							
	KF	0.155	1e-04	—	0.184	2e-04	—
	SIRopt	0.156	1e-04	161.55 (7.92)	0.185	2e-04	167.05 (14.63)
	SIR	0.156	1e-04	165.83 (20.78)	0.185	2e-04	157.01 (23.97)
	ASIR	0.156	1e-04	166.61 (20.16)	0.185	2e-04	166.09 (17.11)
	KPF	0.156	1e-04	191.64 (5.17)	0.185	2e-04	164.49 (15.79)
Case 8 $\sigma_\eta^2 = 0.05$, SNR = 0.5							
	KF	0.185	1e-04	—	0.208	2e-04	—
	SIRopt	0.186	1e-04	163.03 (10.07)	0.209	2e-04	166.95 (14.89)
	SIR	0.186	1e-04	156.35 (23.96)	0.209	2e-04	147.84 (26.69)
	ASIR	0.186	1e-04	157.15 (22.77)	0.209	2e-04	155.93 (20.32)
	KPF	0.186	1e-04	191.52 (6.33)	0.209	2e-04	167.73 (16.61)
Case 9 $\sigma_\eta^2 = 0.1$, SNR = 1							
	KF	0.224	1e-04	—	0.238	2e-04	—
	SIRopt	0.225	1e-04	166.61 (12.86)	0.239	2e-04	168.11 (14.79)
	SIR	0.225	1e-04	140.19 (27.17)	0.240	2e-04	132.94 (28.84)
	ASIR	0.225	1e-04	140.93 (25.76)	0.240	2e-04	140.18 (22.53)
	KPF	0.225	1e-04	191.05 (7.31)	0.239	2e-04	171.34 (16.81)
Case 10 $\sigma_\eta^2 = 0.2$, SNR = 2							
	KF	0.257	2e-04	—	0.264	2e-04	—
	SIRopt	0.258	2e-04	172.05 (13.97)	0.265	2e-04	171.65 (13.33)
	SIR	0.258	2e-04	120.92 (28.08)	0.266	2e-04	115.67 (28.59)
	ASIR	0.258	2e-04	121.72 (26.12)	0.266	2e-04	119.65 (24.14)
	KPF	0.258	2e-04	190.57 (6.91)	0.265	2e-04	175.64 (15.64)

Table 3.4: Summary of simulation study II under 13 different settings: $\sigma_v^2 = 0.1$, $T = 200$, and $N_p = 200$ (continued).

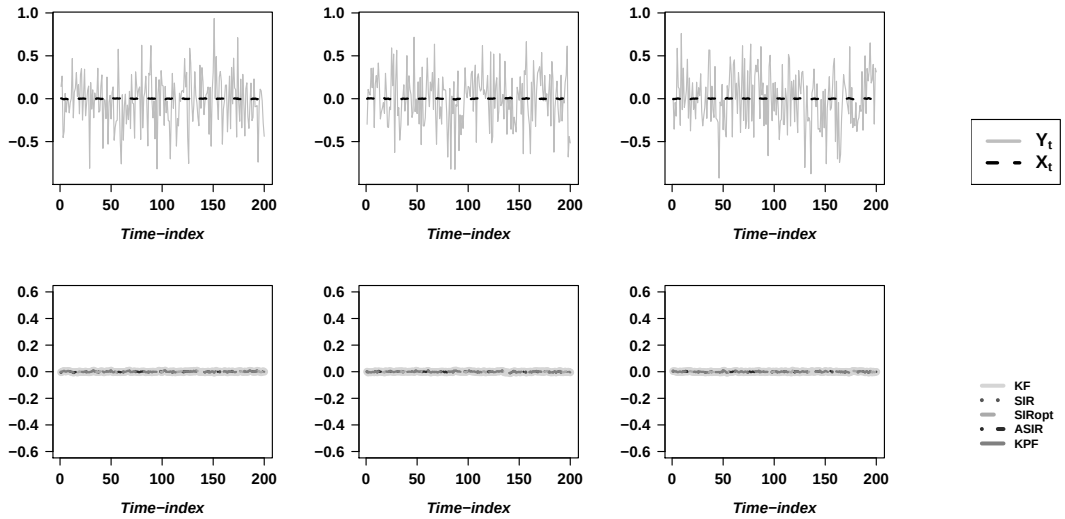
		$\phi = 0.3$			$\phi = 0.8$		
		RMSE		**uNp**	**RMSE**		**uNp**
Setting	**Filter**	**Mean**	**Var**	**Mean(SD)**	**Mean**	**Var**	**Mean(SD)**
Case 11 $\sigma_\eta^2 = 0.5$, SNR = 5							
	KF	0.286	2e-04	—	0.288	2e-04	—
	SIRopt	0.287	2e-04	179.51 (12.55)	0.289	2e-04	178.29 (11.4)
	SIR	0.289	2e-04	94.05 (25.32)	0.291	2e-04	91.14 (25.38)
	ASIR	0.29	2e-04	93.26 (23.87)	0.296	2e-04	89.76 (22.55)
	KPF	0.287	2e-04	190.99 (4.88)	0.289	2e-04	181.81 (12.92)
Case 12 $\sigma_\eta^2 = 1$, SNR = 10							
	KF	0.298	2e-04	—	0.299	2e-04	—
	SIRopt	0.300	2e-04	184.43 (10.05)	0.300	2e-04	183.40 (8.91)
	SIR	0.303	2e-04	74.90 (21.61)	0.304	2e-04	73.32 (21.70)
	ASIR	0.308	2e-04	72.11 (20.88)	0.312	2e-04	67.56 (19.69)
	KPF	0.300	2e-04	193.87 (2.70)	0.300	2e-04	184.16 (9.55)
Case 13 $\sigma_\eta^2 = 10$, SNR = 100							
	KF	0.312	2e-04	—	0.312	2e-04	—
	SIRopt	0.313	2e-04	194.55 (3.67)	0.313	2e-04	194.26 (3.47)
	SIR	0.349	0.002	30.96 (10.58)	0.350	0.002	30.36 (10.24)
	ASIR	0.372	0.001	24.74 (10.13)	0.373	0.002	25.42 (9.19)
	KPF	0.313	2e-04	194.37 (2.85)	0.313	2e-04	194.04 (2.69)

Similar to the analysis for the local level model, the simulation results obtained for the AR(1) plus noise model are fully commented on. In this stationary context, we also illustrate the filtering performance of the competing filters by displaying a graphical illustration for chosen exemplar runs and three signal-to-noise-ratio values; the same settings considered for the non-stationary local level model, say $q \in \{0,0001, 1, 100\}$. The resulting plots when the autoregressive parameter takes the value $\phi = 0.3$ are displayed in Figures 3.10–3.12 on pages 87–89; similar plots for $\phi = 0.8$ are shown in Figures 3.13–3.15.

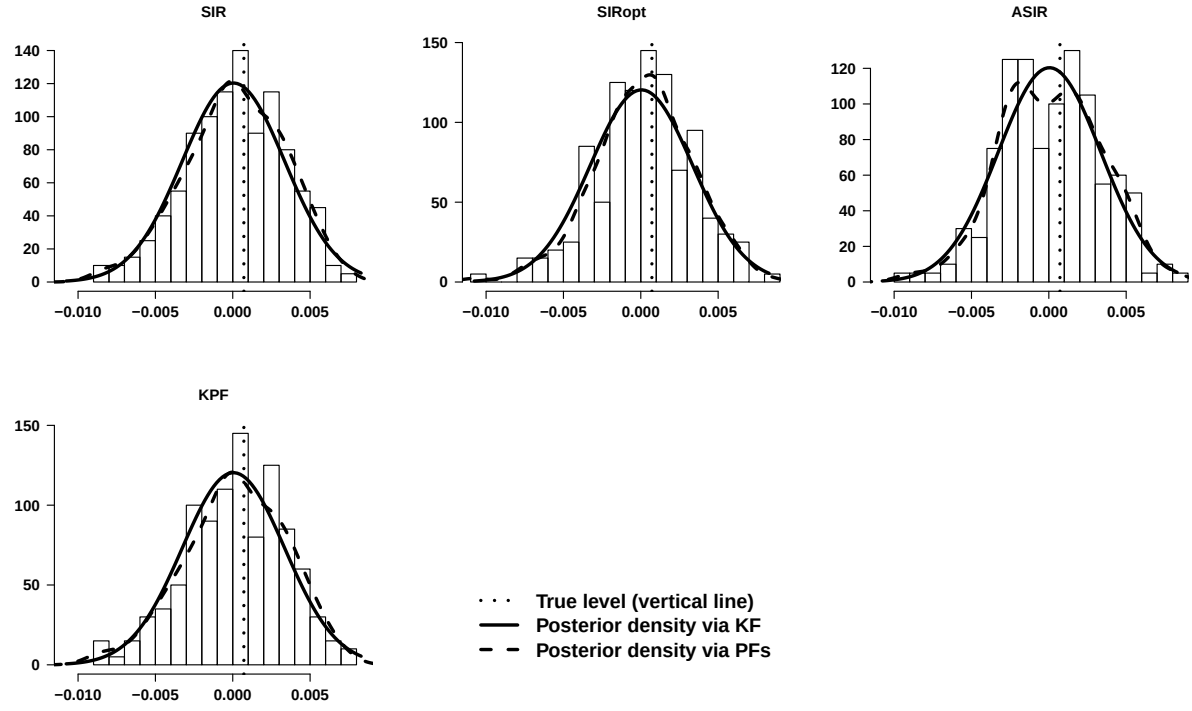
These plots confirm the general results reported in Table 3.4 which suggest that an increase in the signal-to-noise-ratio value q is related to less precise estimation results. Indeed, we observe that the evolution of the difference between estimated and true-state values shows more variability as q increases; the same conclusion was obtained for the local level model.

On panel b) of the aforementioned figures, for each particle filter variant under study, the histogram (together with the estimated posterior density) of the state values $x_{T|T}$ is depicted. Particularly,

for $N_p = 200$ particles and a high signal to noise-ratio value $q = 100$, the histograms shown in Figures 3.12 and 3.15 suggest that both the SIR and the ASIR filters, specially the second, continue showing worse statistical behavior at high signal to noise-ratio values.

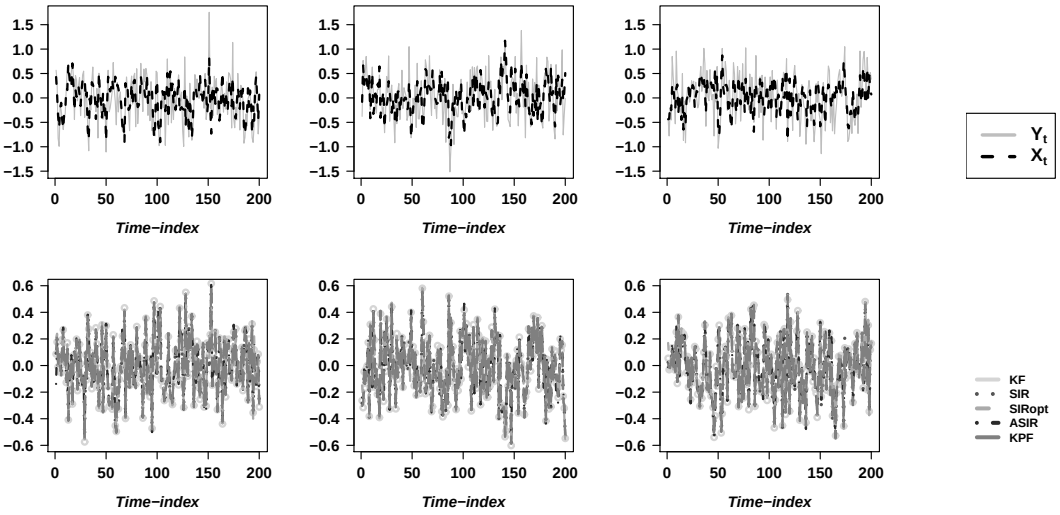


(a) First row: Generated states and observations. Second row: Difference between estimated and true-state values $\hat{x}_{t|t} - x_t$, $t = 1, \ldots, T$.

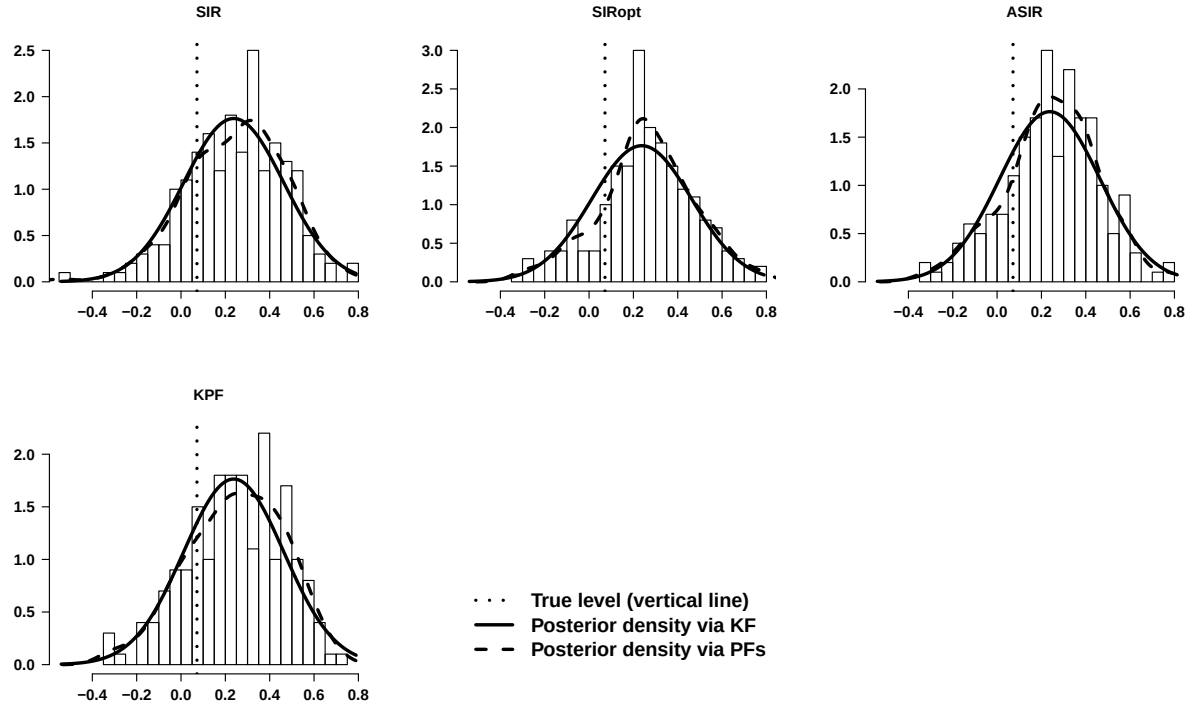


(b) Histogram (together with the estimated posterior density; black/dashed) of state values $\hat{x}_{T|T}$ for last data set. The exact posterior density obtained via the KF (black/continuous) is overlaid to each histogram.

Figure 3.10: AR(1) plus noise model ($\phi = 0.3$): Case 1 with SNR $q = 1e-4$ ($\sigma_\eta^2 = 1e-5$ and $\sigma_\nu^2 = 0.1$).

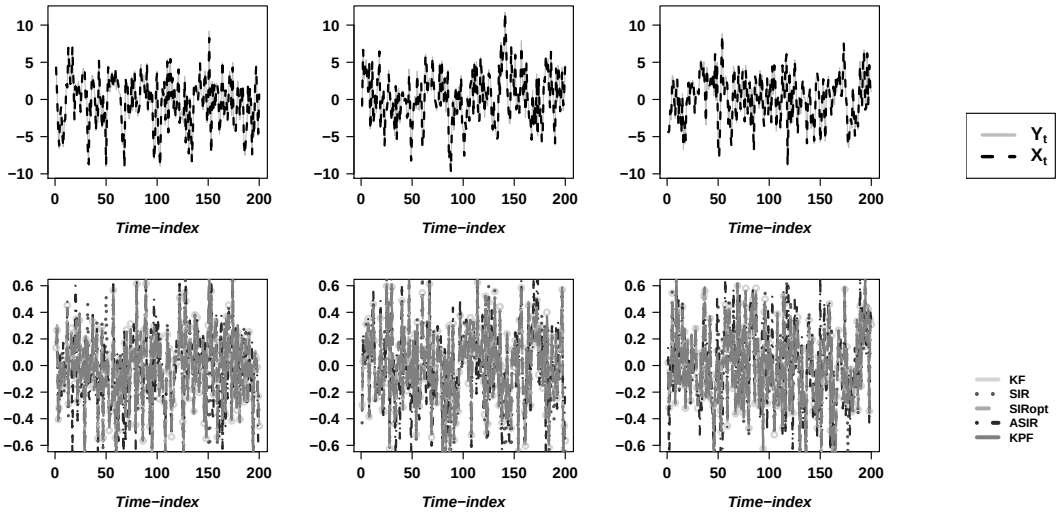


(a) First row: Generated states and observations. Second row: Difference between estimated and true-state values $\hat{x}_{t|t} - x_t, t = 1, \ldots, T$.

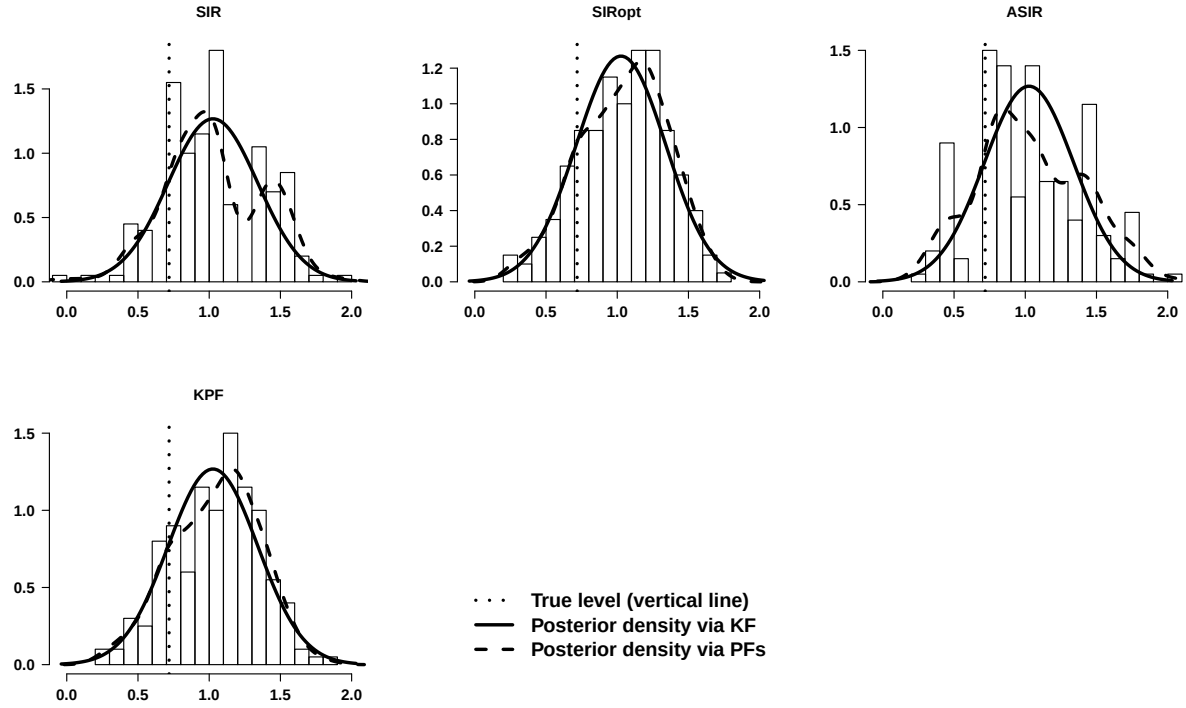


(b) Histogram (together with the estimated posterior density; black/dashed) of state values $\hat{x}_{T|T}$ for last data set. The exact posterior density obtained via the KF (black/continuous) is overlaid to each histogram.

Figure 3.11: AR(1) plus noise model ($\phi = 0.3$): Case 9 with SNR $q = 1$ ($\sigma_\eta^2 = 0.1$ and $\sigma_v^2 = 0.1$).

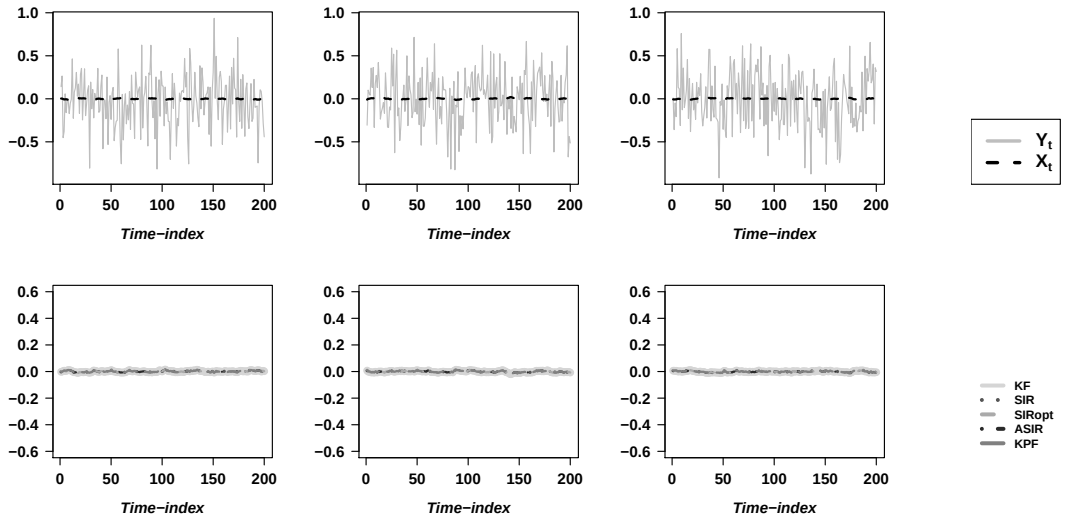


(a) First row: Generated states and observations. Second row: Difference between estimated and true-state values $\hat{x}_{t|t} - x_t$, $t = 1, \ldots, T$.

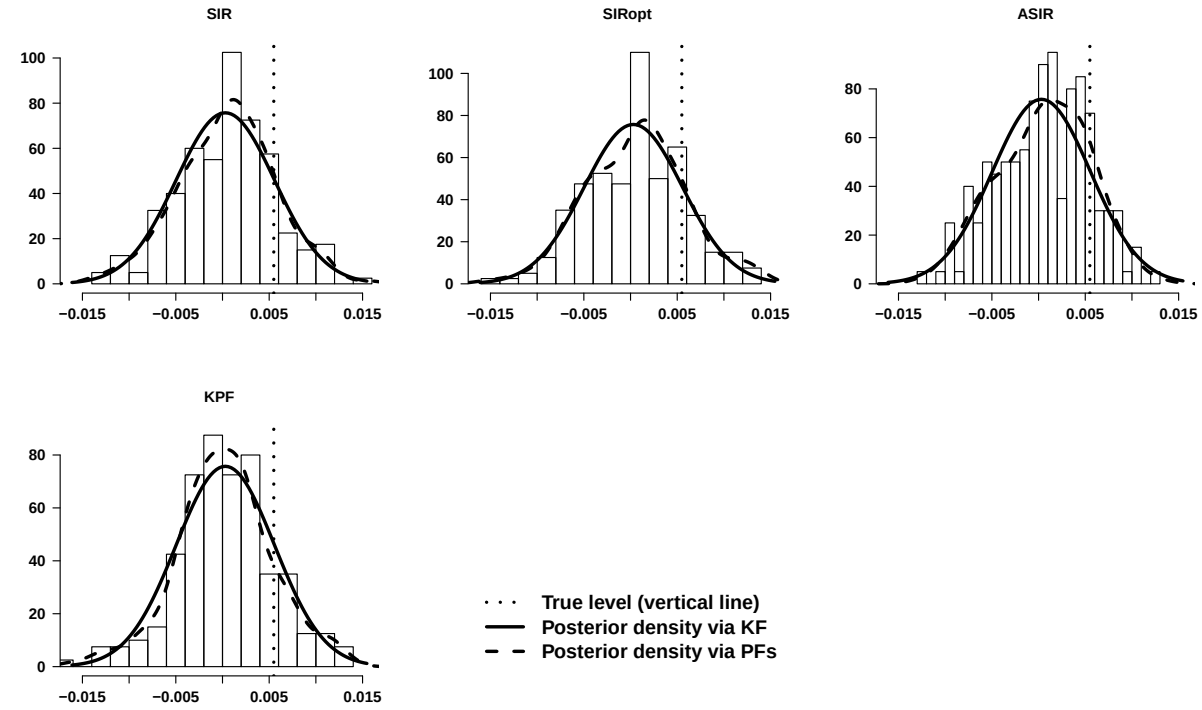


(b) Histogram (together with the estimated posterior density; black/dashed) of state values $\hat{x}_{T|T}$ for last data set. The exact posterior density obtained via the KF (black/continuous) is overlaid to each histogram.

Figure 3.12: AR(1) plus noise model ($\phi = 0.3$): Case 13 with SNR $q = 100$ ($\sigma_\eta^2 = 10$ and $\sigma_\nu^2 = 0.1$).

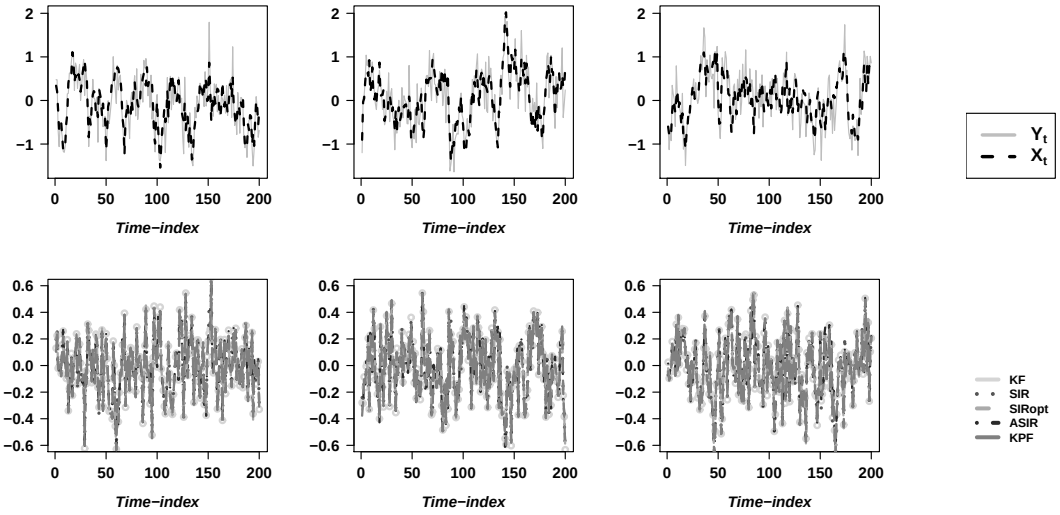


(a) First row: Generated states and observations. Second row: Difference between estimated and true-state values $\hat{x}_{t|t} - x_t, t = 1, \ldots, T$.

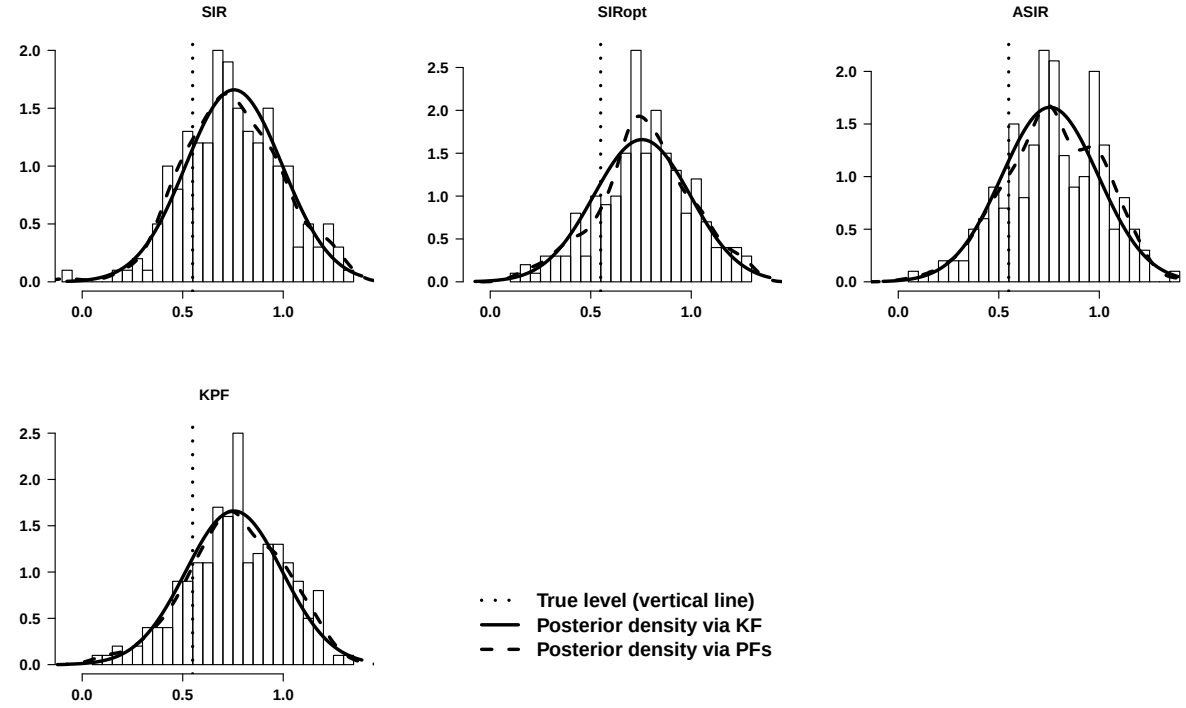


(b) Histogram (together with the estimated posterior density; black/dashed) of state values $\hat{x}_{T|T}$ for last data set. The exact posterior density obtained via the KF (black/continuous) is overlaid to each histogram.

Figure 3.13: AR(1) plus noise model ($\phi = 0.8$): Case 1 with SNR $q = 1e-4$ ($\sigma_\eta^2 = 1e-5$ and $\sigma_\nu^2 = 0.1$).

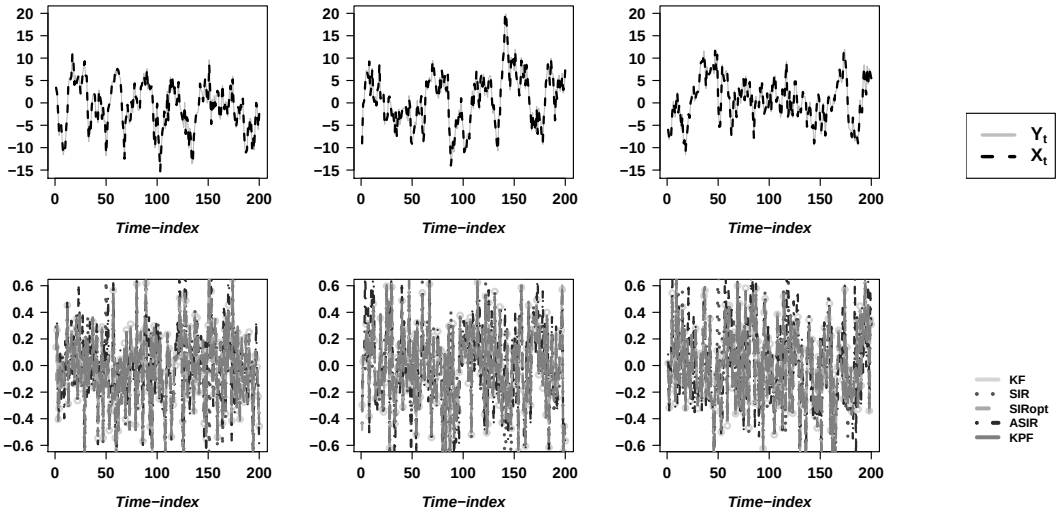


(a) First row: Generated states and observations. Second row: Difference between estimated and true-state values $\hat{x}_{t|t} - x_t, t = 1, \ldots, T$.

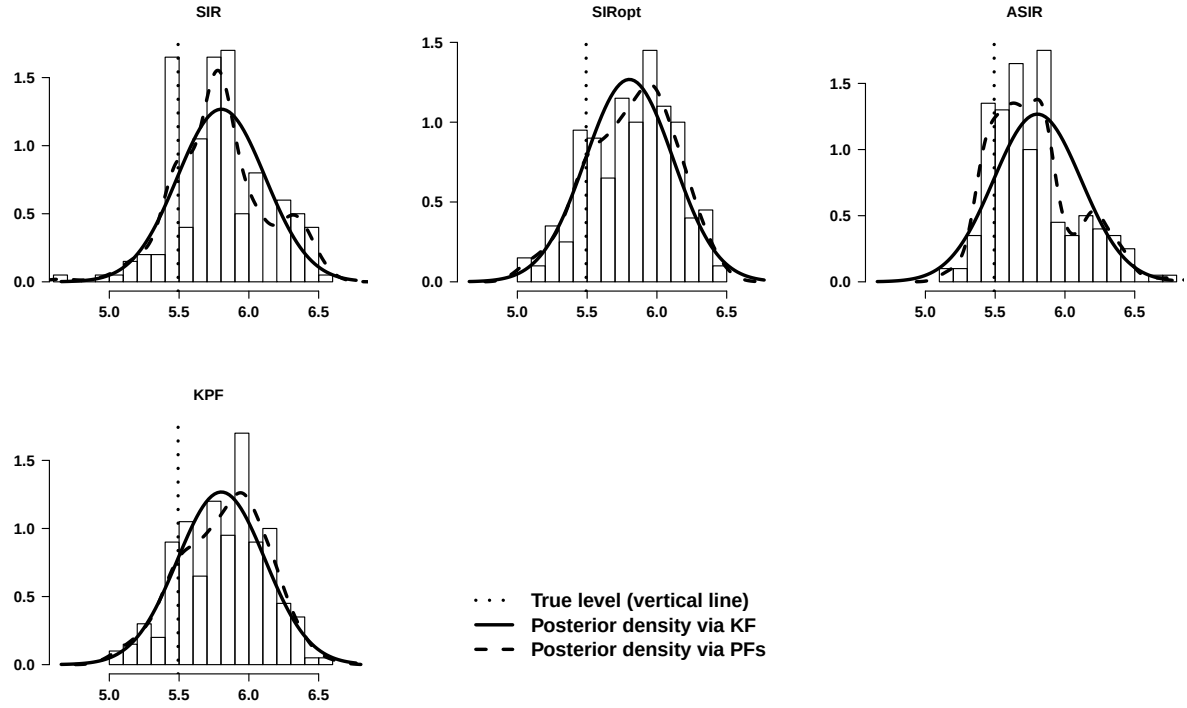


(b) Histogram (together with the estimated posterior density; black/dashed) of state values $\hat{x}_{T|T}$ for last data set. The exact posterior density obtained via the KF (black/continuous) is overlaid to each histogram.

Figure 3.14: AR(1) plus noise model ($\phi = 0.8$): Case 9 with SNR $q = 1$ ($\sigma_\eta^2 = 0.1$ and $\sigma_v^2 = 0.1$).



(a) First row: Generated states and observations. Second row: Difference between estimated and true-state values $\hat{x}_{t|t} - x_t, t = 1, \ldots, T$.



(b) Histogram (together with the estimated posterior density; black/dashed) of state values $\hat{x}_{T|T}$ for last data set. The exact posterior density obtained via the KF (black/continuous) is overlaid to each histogram.

Figure 3.15: AR(1) plus noise model ($\phi = 0.8$): Case 13 with SNR $q = 100$ ($\sigma_\eta^2 = 10$ and $\sigma_\nu^2 = 0.1$).

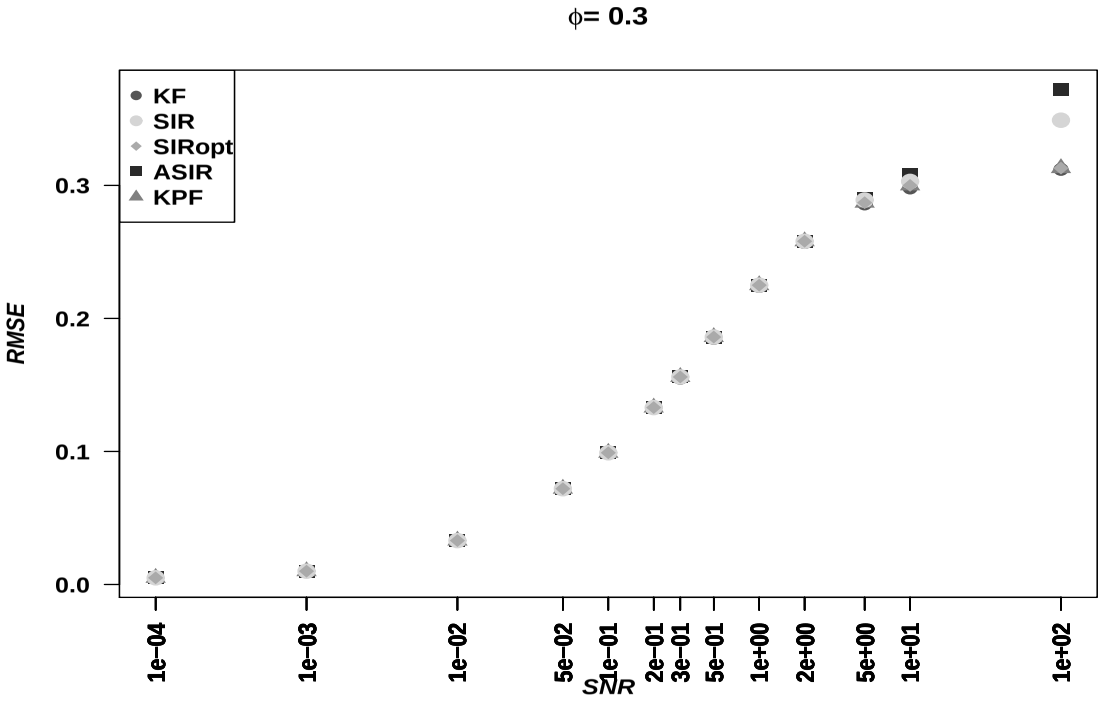
Following, all reported simulation results are discussed in detail, but first we create Figures 3.16 – 3.17. As in the previous section, for each setting of the autoregressive parameter ($\phi = 0.3$ and $\phi = 0.8$), these figures are constructed to give a visual picture of the statistical performance of the competing filters displayed already in Table 3.4.

Remarks and Conclusions for $N_p = 200$ Particles

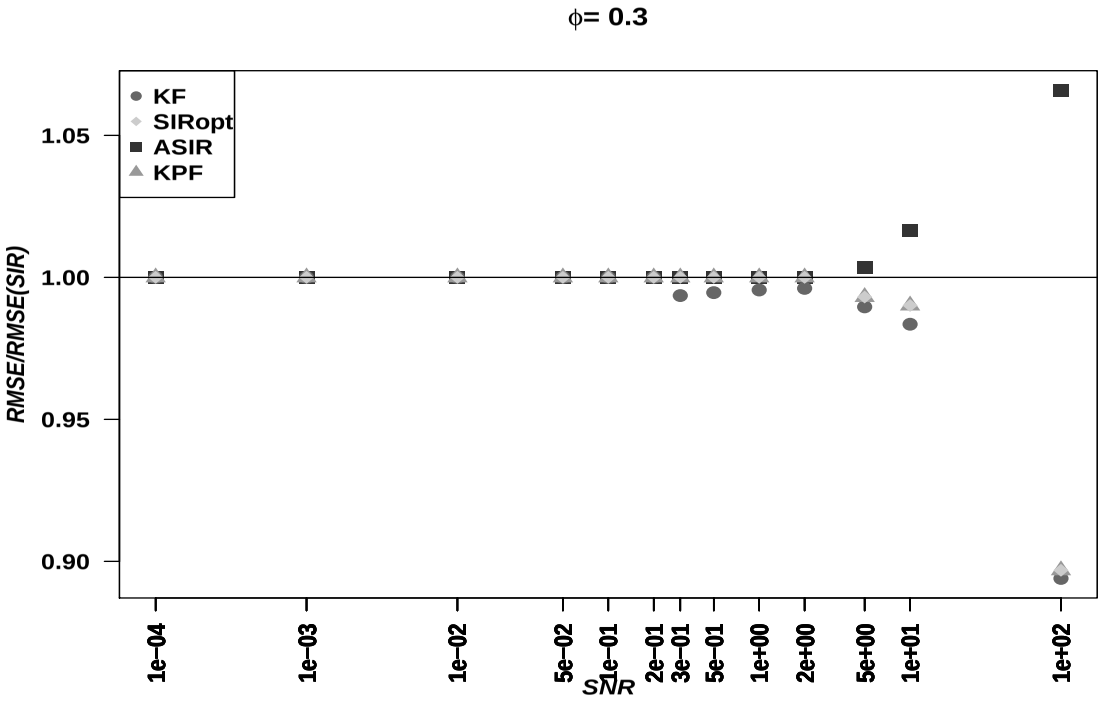
Based on simulation results reported in Table 3.4 and depicted in Figures 3.16–3.17, considering only $N_p = 200$ particles, we make the following remarks and conclusions regarding the performance of the competing filters under study when handling the stationary AR(1) plus noise model:

First, focusing on the statistical performance of the different filters in relation to the 13 signal-to-noise-ratio settings, we conclude that:

- For each filter, the mean-RMSE increases as the signal-to-noise-ratio value q increases irrespective of the value of the autoregressive parameter ϕ; see upper panels of Figures 3.16–3.17.
- As in the previous section, the model at hand is linear and Gaussian but in this case, stationary. In such a scenario, theory dictates that the gold standard KF must display, and it does, the best statistical performance in terms of RMSE. As done with the local level model, in what follows we will not only compare the different simulation based particle filters among themselves, but also take as a reference the filtering performance attained by the gold standard KF.
- For the stationary AR(1) plus noise model at hand, we find that, as expected, among the competing filters the analytical KF yields always the minimum RMSE; see Figures 3.16–3.17 where the dark circle (representing the KF) is always below or coincides with the other symbols (representing the four particle filter variants).
- For signal-to-noise-ratio values $q < 5$ (Cases 1–10) all particle filters under study show a very similar statistical performance to the KF, irrespective of the value ϕ; focus your attention on the upper panels of the Figures 3.16–3.17.
- In the other three cases with $5 \leq q \leq 100$ (Cases 11–13) both the SIR and the ASIR, specially the second, show worse performance in relation to the KF and other two particle filter variants. What the simulation results appear to confirm is that the ASIR particle filter variant continues behaving the worst at high signal-to-noise-ratio values followed by the SIR and that this behavior seems to be more notorious when the autoregressive parameter takes higher values, $\phi = 0.8$ in contraposition to $\phi = 0.3$. Notice that, for these signal-to-noise-ratio settings, the best found particle filters are the SIRopt and the KPF with mean-RMSE values very similar to the one taken by the KF.

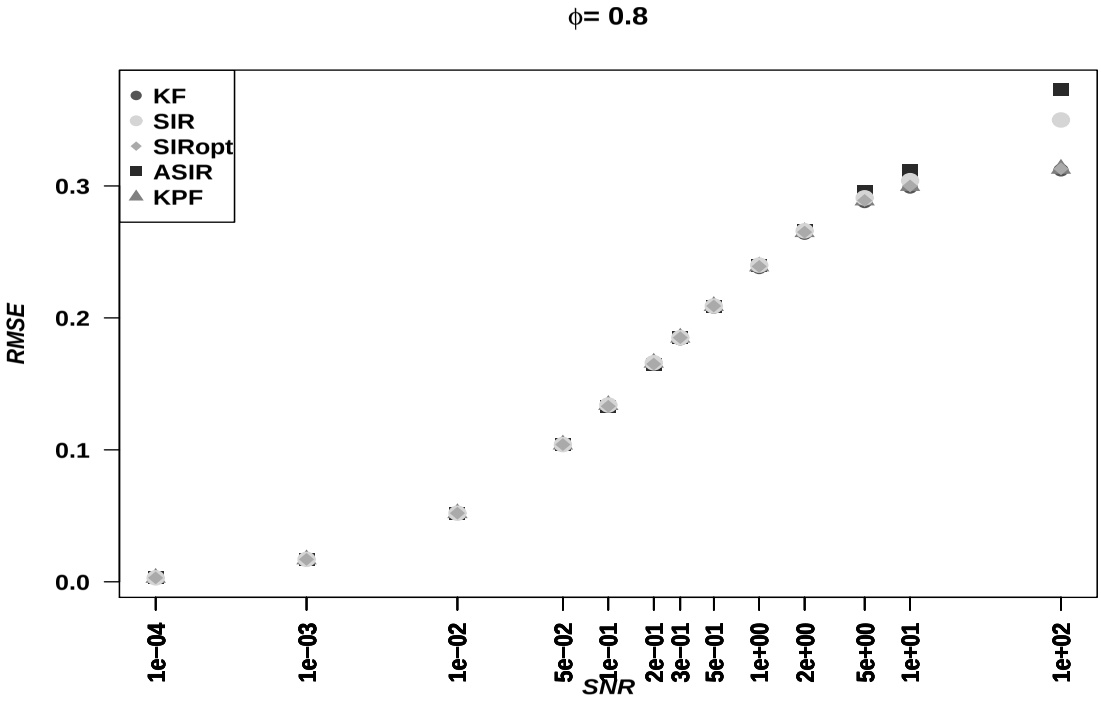


(a) For all filters: Mean(RMSE) vs signal-to-noise-ratio values

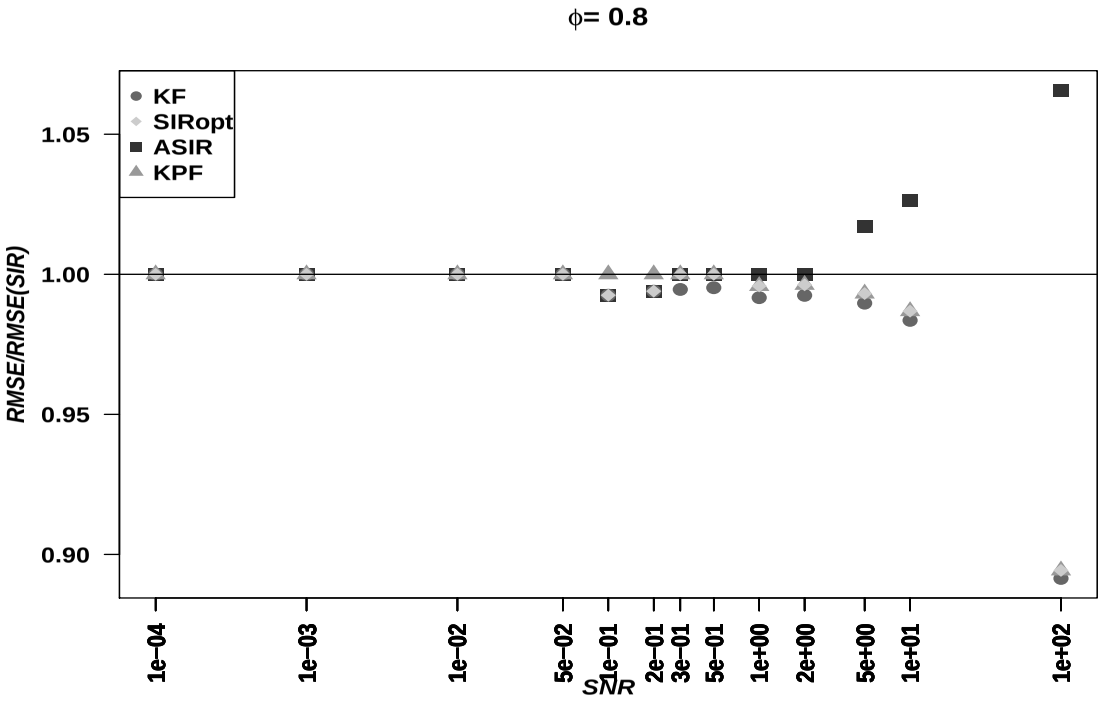


(b) Ratio of mean(RMSE)/mean(RMSE(SIR)) vs signal-to-noise-ratio values

Figure 3.16: AR(1) plusnoise model with $\phi = 0.3$: Impact of the signal-to-noise ratio over the filters mean(RMSE); $N_p = 200$



(a) For all filters: Mean(RMSE) vs signal-to-noise-ratio values



(b) Ratio of mean(RMSE)/mean(RMSE(SIR)) vs SNR values

Figure 3.17: AR(1) plusnoise model with $\phi = 0.8$: Impact of the signal-to-noise ratio over the filters mean(RMSE); $N_p = 200$

Second, in order to compare the relative statistical performance of the different filters in relation to the SIR particle filter variant at different signal-to-noise ratio settings, we focus on the bottom panel of Figures 3.16–3.17. They represent, for each autoregressive parameter setting, the measure $\frac{RMSE(f)}{RMSE(SIR)}$ where $f \in$ {KF, SIRopt, ASIR, KPF} denotes a competing filter, as done in the previous section for the local level model. A visual inspection of the bottom panels of these two figures allows us to confirm all the results commented above about the impact of the signal-to-noise ratio on the statistical performance of the competing filters. Thus, the measure of the relative statistical performance of the filters in relation to the SIR PF variant allows us to conclude that:

- As expected, the non-simulation based KF shows the best statistical performance, but all PF variants are able to practically equate KF's mean-RMSE at most signal-to-noise-ratio values, as described below.

- For $\phi = 0.3$ and lower signal-to-noise ratio values $q \in \{0.0001, 0.001, 0.01, 0.05, 0.1, 0.2\}$ (Case 1 - Case 6) all particle filter variants, including the reference SIR particle filter, equate the statistical performance of the gold standard KF. A similar conclusion can be reached for $\phi = 0.8$, but only for cases 1 - 4. This suggests that the value of ϕ plays a distinctive role in the estimation.

- For $\phi = 0.3$ and middle signal-to-noise ratio values $q \in \{0.3, 0.5, 1, 2\}$ (Case 7 - Case 10) all competing particle filter variants, including the reference SIR particle filter, behave the same among themselves with mean-RMSE values slightly greater than the ones yielded by the KF. For $\phi = 0.8$ and cases 5 - 10, however, mixed mean-RMSE results are obtained, but they are still very close among themselves and to the KF.

- For cases 11-13 with higher signal-to-noise ratio values $q \in \{5, 10, 100\}$, irrespective of the value of the autoregressive parameter ϕ, the competing PF variants SIRopt and the KPF equate KF's mean-RMSE outperforming the reference SIR PF variant, which itself outperforms the ASIR PF variant. Thus, clearly, the SIR and ASIR show in these cases worse performance; specially the second.

- After confirming that the KF yields the best possible filtering estimates of the states for the model at hand, our experimental results also indicate that for a rather small number of particles $N_p = 200$, the particle filtering methodology is able to perform (nearly) as good as the gold standard KF at most (10 out of 13) signal-to-noise ratio values ($q < 5$).

- From the two aforementioned figures, it is thus concluded that all particle filter variants under study practically equate KF's mean-RMSE at lower signal-to-noise-ratio values. Depending on the values of ϕ, mixed results (though very close to KF's) are obtained at middle values of the signal-to-noise-ratio. At higher signal-to-noise-ratio values, the SIR and the ASIR show worst performance, specially the ASIR with $q = 100$. Focus your attention on bottom panels of Figures 3.16–3.17.

Third, we focus on exploring the impact of the SNR on the degeneracy problem of particle filters. As in the previous section, we analyze the reported mean (SD) of the unique number of particles (uNp) at last time-index $t = T$ and find the following degeneracy related patterns:

- For the KPF particle filter variant and $\phi = 0.8$, the mean(uNp) shows the same pattern observed in the non-stationary local level model ($\phi = 1$). That is, the mean(uNp) increases from about 153 to 194 as the signal-to-noise-ratio q increases from $q = 0.0001$ to $q = 100$; focus on last column of Table 3.4. When $\phi = 0.3$, however, there is not a clear pattern, but the values stay between 192-196, which we consider a very satisfactory behavior; focus on third column of first block in Table 3.4. It appears that degeneracy gets worse as the value of ϕ increases: recall that for $\phi = 1$ we have that the mean(uNp) increases from 53 to 193, showing that degeneracy worsens as a function of ϕ specially at lower signal-to-noise-ratios.

- For the SIR and ASIR particle filter variants, similarly to what happens in the local level model, the mean(uNp) decreases as the signal-to-noise-ratio increases irrespective of the autoregressive parameter value ϕ. Specifically, the mean(uNp) spans from about 199 to 30 for the SIR PF and from about 199 to 25 for the ASIR PF.

- For the SIRopt particle filter variant, we observe the same general pattern of the unique number of particles as in the local level model if $\phi = 0.8$. In this situation, the mean(uNp) first decreases from about 178 to 167 as the signal-to-noise-ratio q increases from $q = 0.0001$ to $q = 0.5$. Then, the opposite happens, since we observe that the mean(uNp) increases from about 165 to 194 as the signal-to-noise-ratio q increases from $q = 0.5$ to $q = 100$. For $\phi = 0.3$, however, a general increasing pattern is observed on the mean(uNp) going from a value around 160 to 194. As for the local level model, we believe that an even more exhaustive study could be performed in the future to confirm all the aforementioned suggested results. For example: will this behavior be confirmed if one uses a higher number of particles or a greater time series length? Recall that the previous MC experiments only consider $T = 200$ observations and $N_p = 200$ particles.

- As a by-product, our results indicate that worse statistical performance is attained when the filters show extreme degeneracy; focus on yielded SIR and ASIR mean-RMSE values for case 13.

- Therefore, for the AR(1) plus noise model at hand, what the attained results confirm is that the SIRopt and the KPF suffer the degeneracy problem to a lesser degree compared to the SIR and the ASIR. That is, the SIRopt and the KPF end up with more unique particles than its counterparts SIR and ASIR (which still show satisfying performance, except at very high signal-to-noise-ratio values).

Fourth, focusing on the performance of the different filters in terms of the computational time using a time series length $T = 200$ and $N_p = 200$, we conclude that, as expected, about the same results obtained for the local level model are gotten. That is, the KF is the computationally least expensive algorithm (in terms of the mean-CPU time values in seconds in handling a data set containing $T = 200$

observations), followed by the SIR, the SIRopt, the KPF and the ASIR filter. Thus, as happened for the local level model, for the AR(1) plus noise model at hand also the KPF and the ASIR show worse computational performance. The obtention of these similar CPU-times makes sense, since the only change in the R-implementation is to substitute the value of $\phi = 1$ for $\phi = 0.3$ or $\phi = 0.8$; the reader may refer to Figure 3.9 in previous section.

Next, we perform a small complementary study to further investigate the effect of the increase of the number of particles (from a lowest value $N_p = 200$ to a highest value $N_p = 5000$) on the RMSE and degree of degeneracy of filters studied, also exploring the increase of the time series length from $T = 200$ to $T = 1000$.

3.4.4 Complementary Study: Increasing the Number of Particles and/or the Time Series Length

Herein, we study the impact of increasing the number of particles N_p on the performance of the filters under study. That is, for the AR(1) plus noise model at hand with time-series-length $T = 200$, we first analyze the impact of increasing N_p on the $RMSE$ yielded by the different simulation based filters under study: the SIRopt, SIR, ASIR and KPF particle filter variants. Later, we also present results of the impact of increasing the number of particles and time-series length on the degree of degeneracy.

Exploring the Increase of the Number of Particles

As in the previous section, we aim to study the impact of increasing the number of particles on the mean-RMSE. To achieve that, we construct Figures 3.18 and 3.19 which show the effect of increasing the number of particles, $N_p \in \{200, 500, 1000, 2000, 5000\}$, on the performance of the four particle filter variants under study. We choose the same eight (8 out of 13) representative signal-to-noise ratio settings used in the last section. Thus, based on results reported in Table 3.4 and plotted in Figures 3.18 and 3.19, the following conclusions arise, irrespective of the value of ϕ:

- Except for Case 13 with high signal-to-noise-ratio value $q = 100$, all studied particle filters are practically not affected by the increase of the number of particles, but notice that they already show a very similar statistical performance to the gold standard KF with a rather low number of particles $N_p = 200$.
- In the particular case with high signal-to-noise-ratio $q = 100$, our results show that the ASIR has worse performance. The mean-RMSE yielded by the ASIR approaches the KF's mean-RMSE, but it always stays slightly above. The SIR PF variant also shows unsatisfactory behavior for 200 particles, but achieves a statistical performance close to the KF starting with $N_p = 500$ and $N_p = 1000$ for $\phi = 0.3$ and $\phi = 0.8$, respectively. Generally, it appears that as ϕ increases, larger RMSE values are obtained.

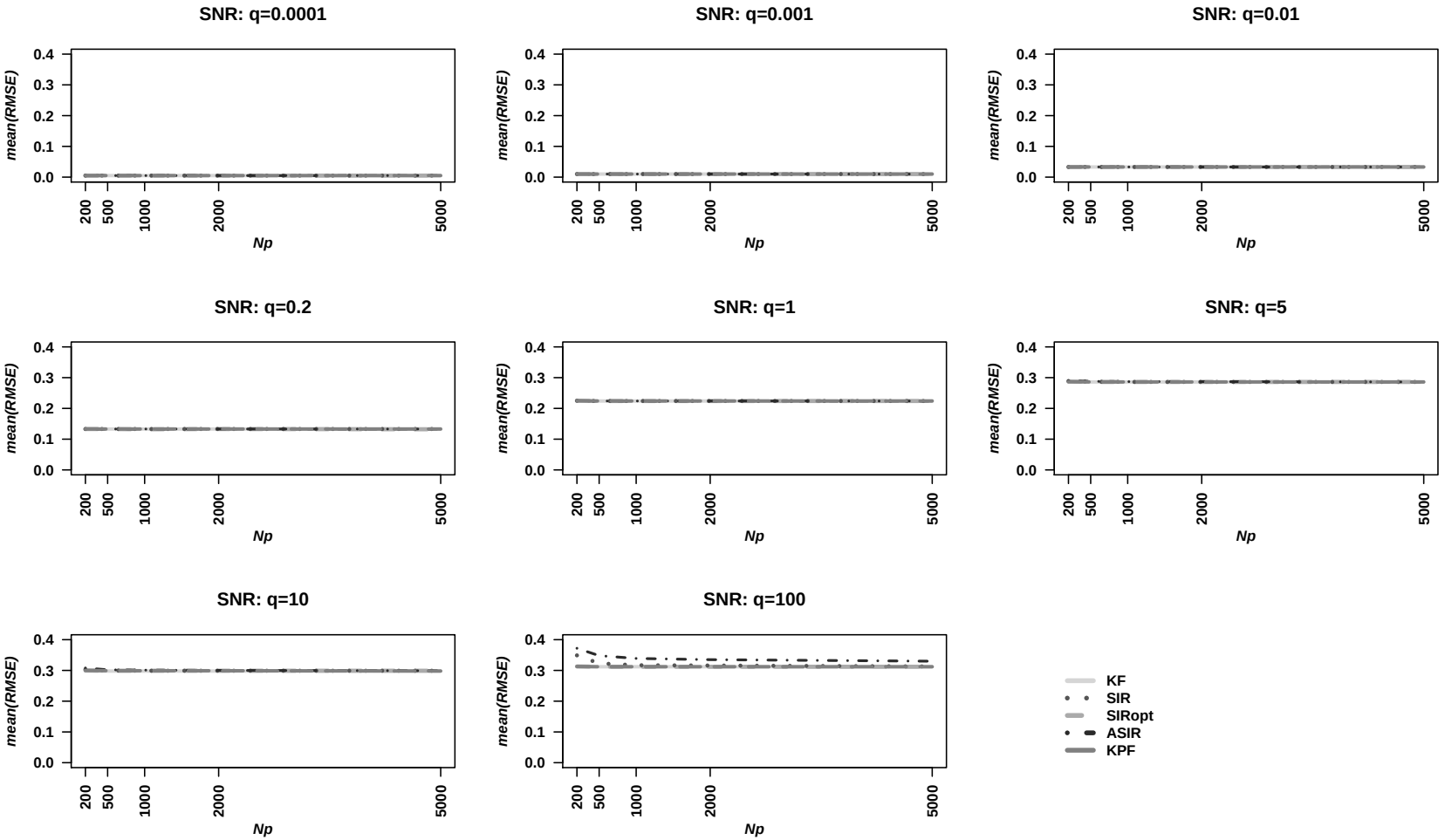


Figure 3.18: AR(1) plus noise model: Effect of the number of particles over the mean-RMSE; fixed $T = 200$ and $\phi = 0.3$.

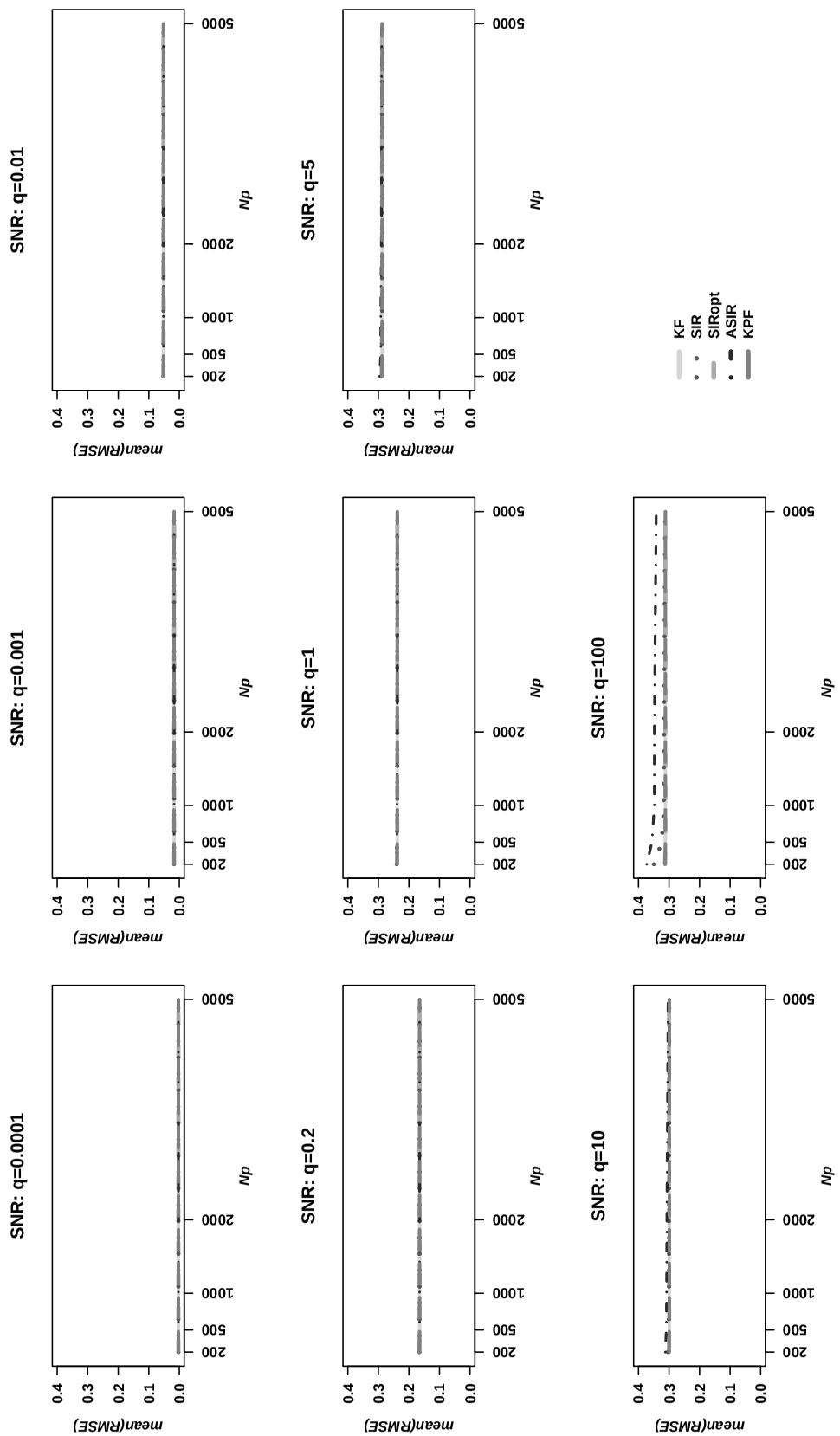


Figure 3.19: AR(1) plus noise model: Effect of the number of particles over the mean-RMSE; fixed $T = 200$ and $\phi = 0.8$.

After finding the minimum number of particles needed to achieve a similar/equal statistical performance to the gold standard KF when dealing with the AR(1) plus noise model, another question remains: Is the estimated posterior marginal density obtained with the minimum found number of particles a reliable posterior? Next, we explore the effect of increasing the numbers of particles on the degree of degeneracy observed, since we think this is closely related to reliability of the estimated posterior marginal densities. Also, we proceed to assess the degree of degeneracy when using a higher time series length.

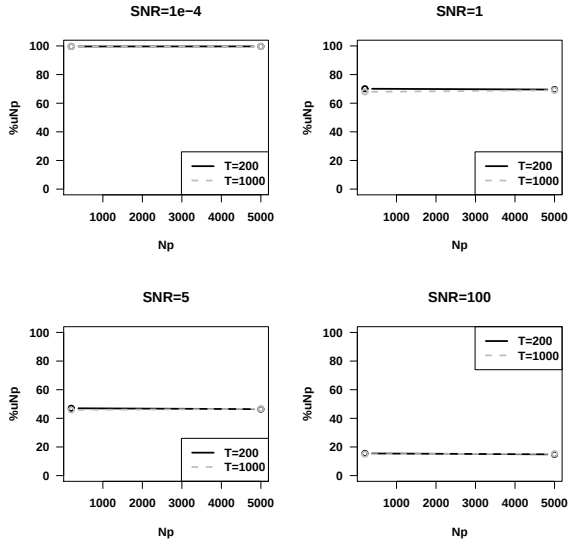
Exploring the Increase of both the Number of Particles and the Time Series Length

In a previous simulation study, with time series length $T = 200$ and $N_p = 200$ particles, we found that the SIRopt and KPF suffer the degeneracy problem, in general, to a lesser degree and that both the SIR and the ASIR particle filter variants are more affected by it at high signal-to-noise-ratio values (in Table 3.4, focus on last three cases, specially in the last one with $q = 100$). Those simulation results suggested that the competing particle filter variants show worse statistical performance when more degeneracy is present.

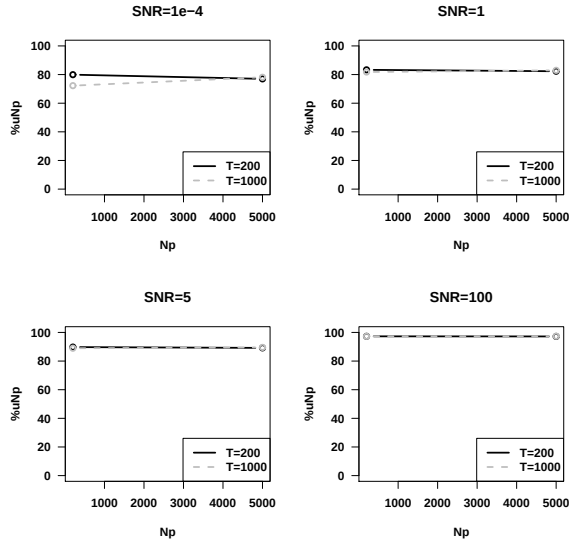
Following, we further explore the effect of increasing the time series length as well as the number of particles on degeneracy. Our aim is twofold: to verify if the general findings obtained using $T = 200$ and $N_p = 200$ are confirmed and also to prevent (or at least postpone) the degeneracy problem with the hope of consequently improving the statistical performance of the particle filters. To achieve these goals, we construct Figures 3.20 – 3.21, which basically represent in the y-coordinate the percentage of the unique number of particles (%uNp) and in the x-coordinate the chosen settings for the original number of particles (N_p). As done for the local level model, we show results for only four signal-to-noise-ratio settings $q \in \{1e\text{-}4, 1, 5, 100\}$, two time series length settings $T \in \{200, 1000\}$ and two number of particles settings $N_p \in \{200, 5000\}$. Thus we have a total of $4 \cdot 2 \cdot 2 = 16$ different settings resulting in 16 plots. As can be seen in the constructed figures, we organize these 16 plots (for each chosen value of the autoregressive parameter, $\phi = 0.3$ and $\phi = 0.8$) in four sub-figures (per type of filter) with each subfigure containing the four plots corresponding to the four signal-to-noise ratio settings used.

The sub-figures in the aforementioned figures allow us to confirm the previously stated results for $T = 200$ and $N_p = 200$. Further, we find out that this behavioral pattern seems to hold regardless of the number of particles and time series length. In other words, the SIRopt (top right sub-figure) and the KPF (bottom right sub-figure) continue suffering the degeneracy problem to a lesser degree. Both the SIR and the ASIR particle filter variants are more affected by it at high signal-to-noise-ratio values (focus on top left and bottom left sub-figures and case: $q = 100$). Since we want to assess the effect increasing the number of particles and the time series length using four signal-to-noise-ratio settings per filter, a detailed description of the degeneracy related performance of the four competing particle filter variants is given below:

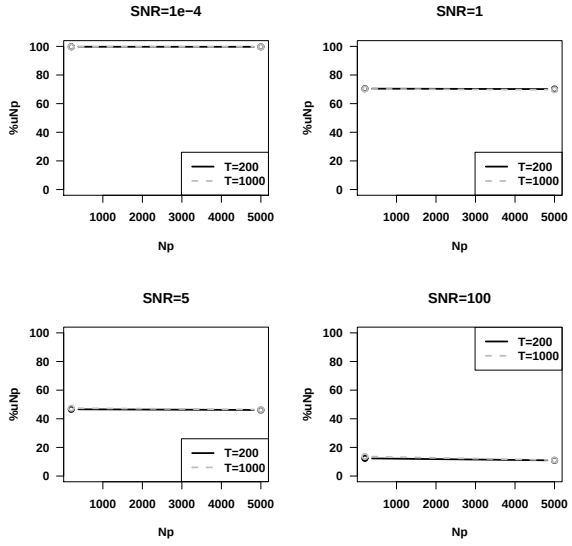
- First, both the SIR and ASIR show a rather general similar pattern on their performance as a function of the signal-to-noise-ratio setting: the number of unique particles decreases as the



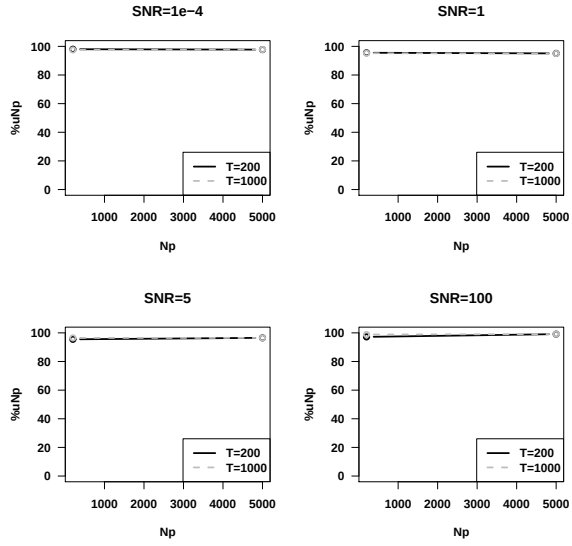
(a) Percentage number of unique particles under SIR



(b) Percentage number of unique particles under SIRopt

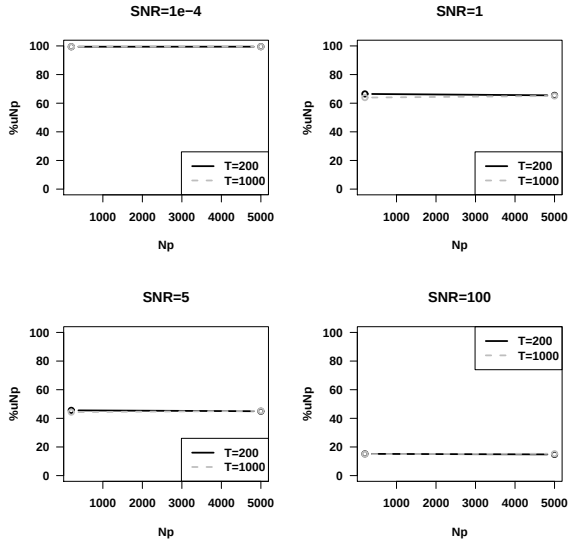


(c) Percentage number of unique particles under ASIR

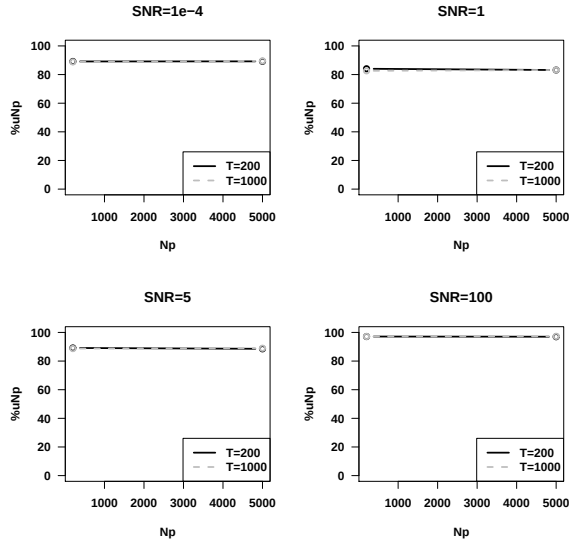


(d) Percentage number of unique particles under KPF

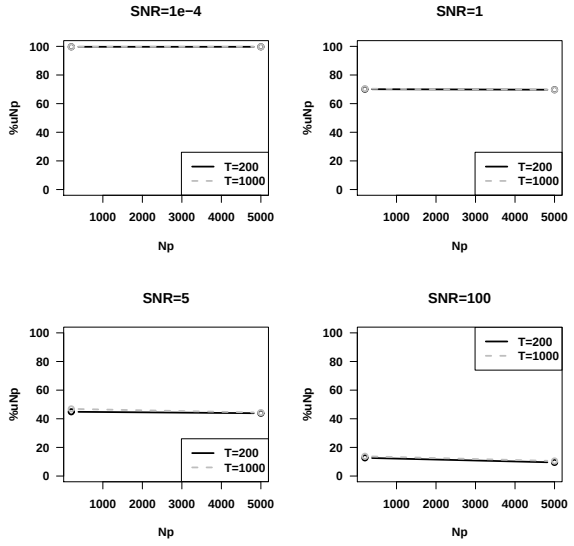
Figure 3.20: AR(1) plus noise model with $\phi = 0.3$: Percentage of unique number of particles at time index $t = T$ (T=200,black/continuous; $T = 1000$, grey/dashed) in relation to the original number of particles $N_p \in \{200, 5000\}$ obtained by the four competing particle filter variants (Top left: SIR, Top right: SIRopt, Bottom left: ASIR and Bottom right: KPF) at selected signal-to noise-ratio settings $q \in \{1e-4, 1, 5, 100\}$.



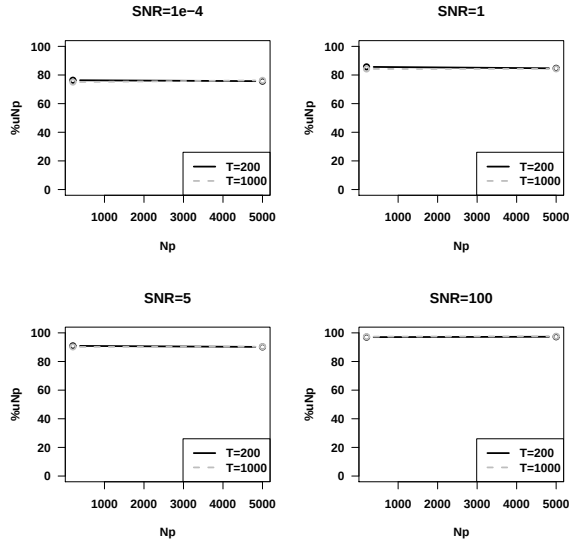
(a) Percentage number of unique particles under SIR



(b) Percentage number of unique particles under SIRopt



(c) Percentage number of unique particles under ASIR



(d) Percentage number of unique particles under KPF

Figure 3.21: AR(1) plus noise model with $\phi = 0.8$: Percentage of unique number of particles at time index $t = T$ (T=200, black/continuous; $T = 1000$, grey/dashed) in relation to the original number of particles $N_p \in \{200, 5000\}$ obtained by the four competing particle filter variants (Top left: SIR, Top right: SIRopt, Bottom left: ASIR and Bottom right: KPF) at selected signal-to noise-ratio settings $q \in \{1e-4, 1, 5, 100\}$.

signal-to-noise-ratio increases, irrespective of the autoregressive parameter value ϕ. However, small discrepancies are observed. For $\phi = 0.8$ the former filter has a worst-case scenario at high signal-to-noise-ratio $q = 100$ where approximately a constant percentage of unique particles (about 15%) is obtained for any value of T and N_p; see top left sub-figure in Figure 3.8. Something similar can be said for the ASIR, but in this case, the percentage of unique particles declines from about 13% to 10% when increasing N_p from 200 to 5000 particles. Both filters show best performance at the lowest signal-to-noise-ratio $q = 1e-4$ where the percentage of unique particles is slightly above 99% irrespective of the values of T and N_p.

- The KPF shows a very satisfactory degeneracy related performance at all signal-to-noise-ratio settings irrespective of the value of the autoregressive parameter ϕ. This filter best case scenario occurs at high signal-to-noise-ratio $q = 100$ with approximately constant percentage of number of particles (about 97%) obtained for any T and N_p.

- Finally, the SIRopt that shows in general a lesser degree of degeneracy, exhibits a distinct behavioral pattern for signal-to-noise-ratio values less than 1 and for signal-to-noise-ratio values greater than one. Indeed, this filter best-case scenario occurs at extremely high $q = 100$ signal-to-noise-ratio values, where approximately a constant percentage of number of particles (about 97%) is obtained for any T and N_p; its worst-case scenario occurs around $q = 1$ where practically a constant percentage of number of particles (about 83%) is obtained for any T and N_p.

- We find these results very encouraging, as they suggest that also for the AR(1) plus noise model the increase of the time-series length (up to $T = 1000$) practically does not affect the obtained percentage of unique number of particles, and that the used number of particles N_p has only a slight effect for the ASIR, a small decrease.

- As a by-product, also for the AR(1) plus noise model at hand, these results suggest that if we had prior information about the relative variation present in our data, we could somehow decide a minimum number of particles so as to avoid degeneracy. The pattern found on the behavior of the percentage number of unique number of particles (it seems stable irrespective of the time series length T and number of particles N_p used in the estimation procedure), is a very nice result since we are aware of the importance in dealing with the degeneracy problem within the particle filtering methodology.

- As known, all particle filters suffer the degeneracy problem and our results do not contradict that fact. Our contribution herein is on describing in detail what happens in particular situations –characterized by the chosen dynamic model and varied simulation settings– in order to provide some guidelines for the practitioner interested in using the filters which have been studied. Thus, for AR(1) plus noise model, putting together the statistical performance in terms of RMSE and a measure of degeneracy given by the percentage of unique number of particles %uNp, we also recommend as a rule of thumb to use $N_p = 5000$ particles, irrespective of the particle filter

variant and the signal-to-noise-ratio, but discarding from this generalization extremely low/high signal-to-noise-ratio values. Notice that if we focus on the degeneracy related performance, the choice of $N_p = 5000$ particles would yield about 500 (that is, 10% unique particles in the worst of the worse-case scenarios, what we think is a reasonable enough amount of particles to produce a reliable marginal posterior representation of the states. If the reader is only interested in a particular particle filter variant, previously presented specific remarks can be referred to indicating that even a smaller number of particles could be used.

3.5 Final Remarks and Conclusions

Following, we provide a summary of the findings obtained through the two Monte Carlo experiments carried out in the context of two linear and Gaussian dynamic state-space models with known parameters: the non-stationary local level model and the stationary AR(1) plus noise models.

- **Best non-simulation based filtering solution**: It is confirmed that when dealing with a linear and Gaussian dynamic state-space model with known model parameters the exact Kalman filter provides the best filtering solution.

- **Best found PF variant in terms of RMSE and degeneracy**: Based on the simulation results, among the particle filter variants studied in this chapter (SIRopt, SIR, ASIR and KPF), the best choice would be the SIRopt as it reaches the Kalman filter's RMSE with only 200 particles at all 13 signal-to-noise-ratios case-scenarios for the stationary type-of-model ($\phi \in \{0.3, 0.8\}$). For the non-stationary type-of-model ($\phi = 1$), in most SNR case-scenarios (12 out of 13) only 500 particles are needed to reach a RMSE value similar or equal to the KF's RMSE. For very low signal-to-noise-ratio value (Case 1 with $q = 1e-4$), increasing the number of particles from 200 to 500 produces a noticeable decrease of the RMSE, but thereafter it decreases slowly so that with 5000 particles it still remains slightly above the KF's RMSE.

- Regarding degeneracy[6], the SIRopt suffers less the degeneracy problem yielding worst and best (percentage mean number of unique particles at last time index $t = T$) $\%\text{uNp}_T$ results of %80 and %97, respectively. Additionally, among all four competing particle filters, the SIRopt has relatively a low computational cost, since overall simulations the SIR is found to be the less expensive particle filter variant followed by the SIRopt, the KPF and the ASIR; being the latter the most costly. Thus, the recommendation in favor of the SIRopt holds irrespective of the number of particles used in the estimation procedure, of the setting of the autoregressive parameter ϕ[7] and also irrespective of the signal-to-noise ratio value (13 settings).

[6] Be reminded that the degree of degeneracy is measured by us as the percentage mean number of unique particles at last time index $\%\text{uNp}_T$; the larger the better.

[7] Be reminded that three values of ϕ are considered, $\phi \in \{0.3, 0.8, 1\}$

- In the ideal linear and Gaussian context, the simulation results have shown that the particle filtering methodology adopted in this thesis is also operational[8]. Indeed, all four particle filter variants studied in this chapter prove to be able to reach the KF's RMSE, although in most cases this can only be attained by using a larger number of particles in the estimation procedure. Naturally, an increase of the number of particles leads to more computational cost and to a increase in memory requirements, but we consider that this higher cost does not represent a major problem for the problems at hand with today's computer resources.

- For the two linear dynamic models at hand, although among the four studied particle filter variants (SIRopt, SIR, ASIR, KPF) the SIRopt PF would be the first choice, most practical problems deviate from the linear and Gaussian ideal context treated in this chapter. In such cases, it can be either not possible or not straightforward to use a fully adapted proposal PDF (as explained in Chapter 2) as required by the SIRopt, being then mandatory to adopt alternative particle filter variants such as the other three algorithms studied in this chapter. Notice that the simulation results indicate that generally, any of these other three studied particle filter variants (SIR, ASIR, KPF) are also able to reach the KF's statistical performance if more particles are used in the estimation procedure. The needed number of particles for these three particle filters to reach the KF's RMSE varies according to the adopted filter, the signal-to-noise-ratio and also according to the value of the autoregressive parameter ϕ.

- In the stationary context ($\phi = 0.3$ and $\phi = 0.8$), at most signal to noise ratio settings (10/13) all particle filters are able to reach a RMSE value similar or equal to the KF's RMSE with only 200 particles, except for the SIR and ASIR at higher signal-to-noise-ratio settings ($q \in \{5, 10, 100\}$) where about 500 and 1000 particles would be required when $\phi = 0.3$ and $\phi = 0.8$, respectively.

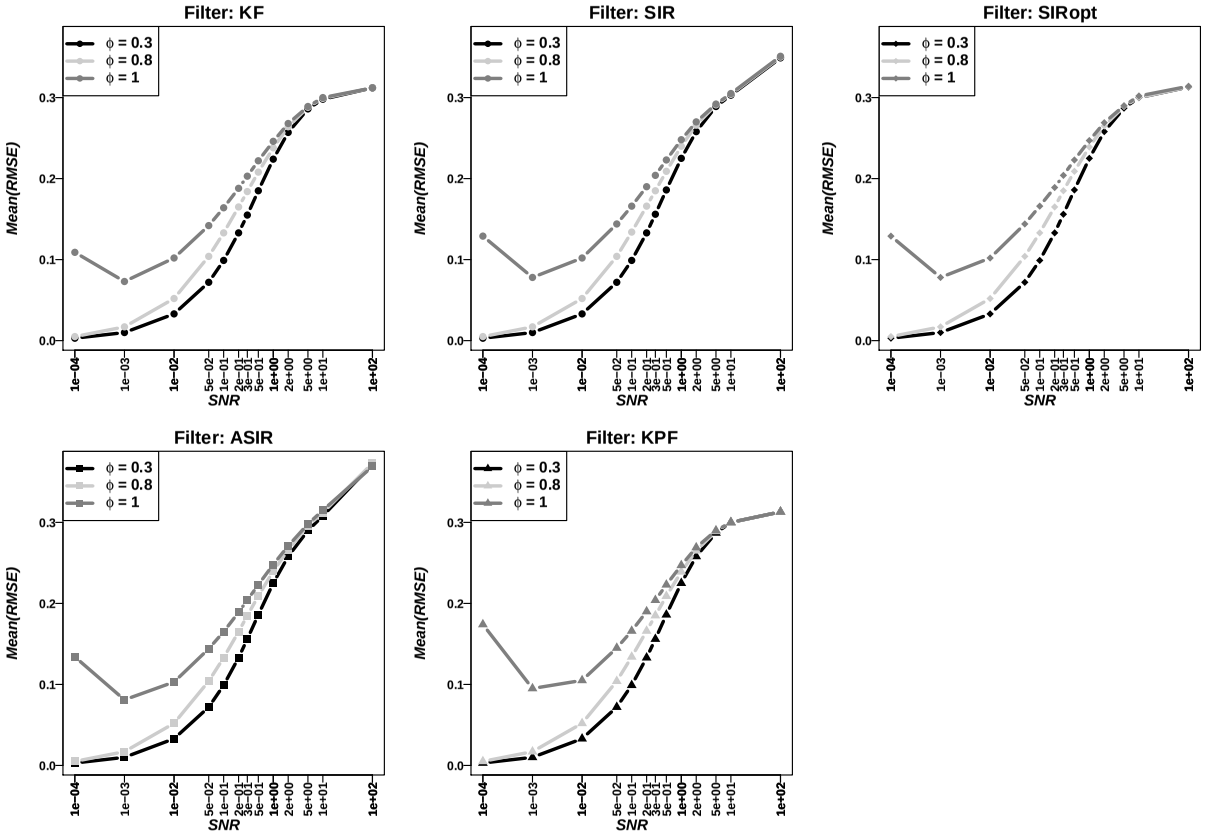
  On the other hand, in the non-stationarity context ($\phi = 1$) results are more varied depending on the filter and signal-to-noise-ratio. However, at most signal-to-noise-ratio settings all three particle filters are able to reach a RMSE value similar or equal to the KF's RMSE with already 500 particles, except for the SIR at very low signal-to-noise-ratio ($q = 1e-4$), the ASIR at very low ($q = 1e-4$) and very high ($q = 100$) signal-to-noise-ratio settings, and for the KPF at lower signal-to-noise-ratio values ($q \in \{1e-4, 1e-3\}$). Indeed, the SIR and the ASIR require about 5000 particles to get RMSE values slightly larger than the benchmark KF's at very low SNR $q = 1e-4$. At very high SNR $q = 100$, the ASIR with 20000 particles is still not able to equate the KF's RMSE, but it remains slightly above KF's RMSE. Similarly, at very low SNR $q = 1e-4$ even with 20000 particles the KPF is not able to equate the KF's RMSE, whereas at $q = 1e-3$ about 5000 particles will be enough.

  The above findings indicate that the value of the autoregressive parameter also has a certain impact on the statistical performance of the competing particle filters. Specifically, as ϕ increases, the attained RMSE values get larger; see upper panel of Figure 3.22.These results also suggest

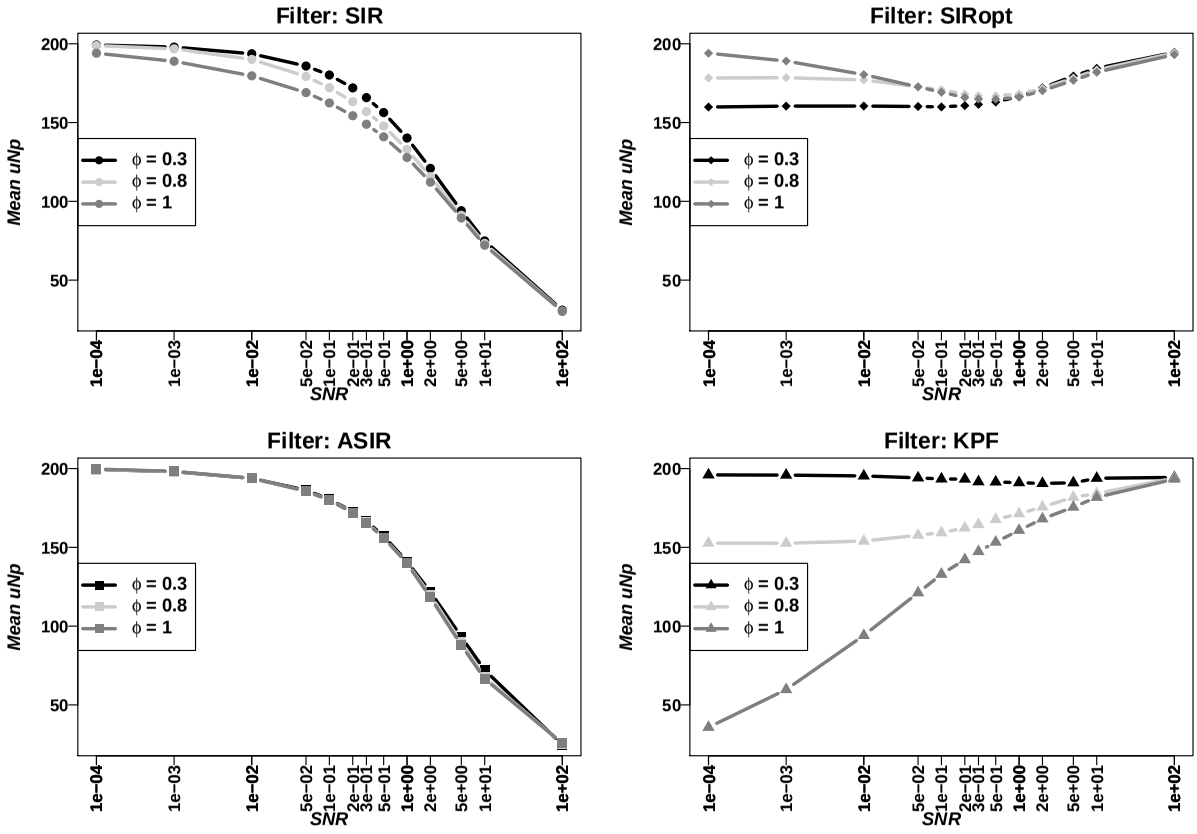
[8] Herein, by operational we mean a filter that is able to reach a similar or equal statistical performance as the exact KF.

that irrespective of the value of the ϕ parameter, generally about 5000 particles are enough to reach the benchmark RMSE value yielded by the Kalman filter.

- Focusing on the impact of the signal-to-noise-ratio on degeneracy, the simulation results indicate that as the signal-to-noise-ratio values get larger the degree of degeneracy worsens (get smaller) for the SIR and the ASIR particle filters, improves for the KPF, and shows relatively stable and very good values for the SIRopt. Indeed, over all 13 signal-to-noise-ratio settings and the three autoregressive parameters settings, the worse/best degree of degeneracy results obtained are %80/%97 for the SIRopt, %18/%98 for the KPF, %15/%99 for the SIR and %12/%99 for the ASIR. These results again suggest that the value of the autoregressive parameter ϕ also plays a certain role on the degree of degeneracy attained over the 13 signal-to-noise-ratio settings, but in a different manner. For the SIR and the KPF particle filter variants when ϕ increases (going from stationarity to non-stationarity), the degree of degeneracy %uNp gets smaller (worse); the ASIR shows an identical but decreasing behavior irrespective of the ϕ setting. For the SIRopt, irrespective of the ϕ setting, the %uNp first tends to decrease a bit until around SNR $q = 0.5$ to later increases again but overall remains at higher values; see bottom panel of Figure 3.22. Thus, in general, the SIRopt suffers less the degeneracy problem, the SIR and the ASIR suffer it less at lower signal-to-noise-ratio values and the KPF suffers it less at higher signal-to-noise-ratio values.



(a) Evolucion of RMSE over 13 SNR's and three values of phi



(b) Evolucion %uNp over 13 SNR's and three values of phi

Figure 3.22: Role of ϕ on RMSE and degeneracy of four studied PFs at the 13 different signal-to noise-ratio settings.

- With respect to the used number of particles and the time series length, the simulation results also indicate that for the two linear models at hand with a fixed time-series length, an increase in the number of particles leads to 1) a reduction of the RMSE, but not in a relevant manner, 2) an increase of the CPU times and to 3) a higher absolute number of unique particles (though the attained percentages remain rather stable as mentioned before). Additionally, when more observations are taken into account, as expected the RMSE tends to decrease, but this decrease is not found relevant. We believe that this is due to the simple structure of the two dynamic linear models at hand.

- An interesting and positive result is observed as a by-product of carried out simulations, since given a fixed value of ϕ and within each combination of particle filter variant and signal-to-noise-ratio setting, the percentage mean of unique particles %uNp seems to remain rather stable; the reader may look back to Figures 3.8, 3.20 and 3.21. And what we consider even more relevant is that this behavior seems to hold irrespective of the time series length and the number of particles used in the estimation procedure; the reader may look back to Table 3.3 illustrating the combined impact of increasing both, the time series length and the number of particles on the degree of degeneracy for Case 5 with SNR $q = 0.1$ (the impact on the RMSE and the CPU-time is also reported therein). Again, our results suggest that worse statistical performance is attained when the filters show more degeneracy, but also that the corresponding RMSE values get smaller (improving filtering performance) when increasing the number of particles and thus it helps to prevent (or at least postpone) the degeneracy problem.

- In practice, we cannot control the signal-to-noise-ratio since it is data dependent, but as a result of the extensive MC experiments carried out, we have been able to characterize the behavior of the different particle filters studied at different low-to-high signal-to-noise-ratios and at three chosen autoregressive parameter values.

  That is, the Monte Carlo results all-together suggest that if we had prior information about the relative variation present in our data, we could somehow decide a minimum number of particles so as to avoid degeneracy. The pattern found on the behavior of the percentage number of unique number of particles (it seems stable irrespective of the time series length T and number of particles N_p used in the estimation procedure), is a very nice result since we are aware of the importance in dealing with the degeneracy problem within the particle filtering methodology. We consider this result as an important contribution that can serve as a guideline for practitioners when choosing the number of particles having in mind to avoid degeneracy problem.

- As known, all particle filters suffer the degeneracy problem and our results do not contradict that fact. Our contribution herein is on describing in detail what happens in particular situations – characterized by the chosen dynamic model and varied simulation settings– in order to provide some guidelines for the practitioner interested in using the filters which have been studied. Thus, analyzing and putting together all the aforementioned simulation results and findings, for the

two studied dynamic linear state-space models –irrespective of the particle filter in question, the signal-to-noise-ratio setting[9] and of the autoregressive ϕ parameter setting– as rule of thumb we recommend the use of $N_p = 5000$ particles not only to get a similar statistical performance to the benchmark KF, but also to avoid the inherent degeneracy drawback. Notice that if we focus on the degeneracy related performance, the choice of $N_p = 5000$ particles would yield about 500 (that is, 10% unique particles in the worst of the worse-case scenarios, what we think is a reasonable enough amount of particles to produce a reliable marginal posterior representation of the states. When only interested in a particular particle filter variant, the reader can refer to previously presented specific remarks indicating that even a smaller number of particles could be appropriate for obtaining a reliable posterior.

- As shown (empirically) for the linear and Gaussian model in question, the choice of a specific particle filter over another depends on the practitioner's expertise (expert knowledge of the structure of the dynamic model at hand and/or the available particle filter variants) or preference to a specific filter combined with available computer memory and CPU time resources.

To close this chapter dealing with two dynamic linear state-space-models, one stationary and the other non-stationary, we state that although the KF provides the best filtering solution, the particle filtering methodology is also capable of reaching such performance at the expense of more computational and memory requirements. We consider that the required CPU cost (time and memory) does not represent a major problem with today's available computer resources. Naturally, the carried out simulation studies regarding the use of the particle filtering methodology in a linear and Gaussian context as the one treated in this chapter responds more to an academic or methodological objective that we consider fulfilled: to characterize the behavior of particle filters in an ideal context where the exact solution exists, additionally to (re-)assess the impact of key factors within the particle filtering methodology, such as: the signal-to-noise-ratio value, the number of particles and the time-series length, and from there to get some deeper knowledge of the inner features (conceptual and implementation) of each competing particle filter variants. This acquired better understanding proves to be useful in future simulation studies dealing with non-standard dynamic state-space models having a more complex structure as the one considered in this chapter.

Indeed, based on these carried out simulation studies and on literature review, we have confirmed that among the simulation-based algorithms, the particle filtering methodology is a good alternative. We must say, however, that this methodology shows its superior performance when dealing with dynamic models with a not so simple structure as the linear models considered in this chapter.

Next chapter not only aims to illustrate and to empirically show the superior performance of particle filters over Kalman based approaches when dealing with complex state-space models, but also how some particle filter variants outperform others. We remark that the Monte Carlo study presented

[9] We exclude from this generalization the following two cases for $\phi = 1$: Case 1 with lowest SNR ($q = 1e-4$) and Case 13 with highest SNR ($q = 100$); in such cases the reader may refer back to corresponding results and remarks within this chapter.

in next chapter is based on a short time series length. The reason for it is that the nonlinear model taken as a benchmark is a synthetic one with no further interest than to highlight that particle filters do outperform traditional Kalman based filters in the presence of complex dynamic models as the one at hand in Chapter 4, which was artificially constructed by the authors of the UPF.

CHAPTER 4

BENCHMARK SIMULATION STUDY: FILTERING IN A NONLINEAR FRAMEWORK

This chapter aims to illustrate the filtering performance (state-estimation) ability of six competing algorithms (two non-simulation based and four particle filters) in a nonlinear context. That is, we conduct two Monte Carlo studies confronting some existing particle filter variants already described in Chapter 2 (pseudocodes are therein also presented) named the sampling importance resampling (SIR), the extended particle filter (EPF), the unscented particle filter (UPF) and the adapted auxiliary sampling importance resampling (ASIR). The EPF and UPF are also examples of adapted filters as they both use a Kalman-based proposal distribution that incorporates the latest observation.

Notice that all particle filters considered are variant of the generic SISR particle filter, but are mainly distinguished by the use of different proposal PDF or the adopted resampling scheme; see Table 2.2. Thus, the two entertained benchmark simulation studies focus mainly on assessing the filtering performance of the four mentioned simulation based nonlinear filters. For completion, the non-simulation based filters, EKF and UKF, are also included in the MC experiments. In the second simulation study, two commonly implemented (because of their efficiency) resampling strategies are also used for comparison among filters: the stratified and residual resampling schemes.

To achieve our goals, we design and implement the two mentioned Monte Carlo studies using as a benchmark model a synthetic nonlinear model taken from the literature. Specifically, the chosen model is not only nonlinear and non-Gaussian, but also a non-stationary threshold model, which up to our knowledge was created by the authors of the UPF particle filter variant mainly to show its potential superior performance in contraposition to the SIR and the EPF particle filters; considering also the non-simulation based EKF and UKF filters, see Van der Merwe et al. (2001). This model is presented in Section 4.1.

The first simulation study basically reproduces some of the results presented by Van der Merwe et al. (2001). These authors compare the performance of all the aforementioned nonlinear filters except the ASIR. However, they assess the filtering performance of five nonlinear filters considering solely a statistical measure of performance based on the RMSE. Additionally, they only use a small number of particles $N_p = 200$, and implement only the residual resampling strategy to carry out the selection step of the particle filters.

In a second Monte Carlo study, we make an extension of the first simulation study. This is done by:

1. incorporating an extra particle filter variant we have worked with: the ASIR PF,
2. including the resampling strategy we have always worked with: the stratified one,
3. providing not only a statistical measure of performance of the studied filters, but also a computational measure of performance, and
4. by specifically assessing the effect of incrementing the number of particles.

Hence, the second simulation design includes as a particular case a subset of the Monte Carlo study considered by Van der Merwe et al. (2000, 2001).

This chapter is organized as follows: In Section 4.1, the state-space formulation for the chosen benchmark synthetic nonlinear dynamic model is specified. In addition, some motivating issues regarding the choice of this nonlinear model are presented. Section 4.2 presents the general procedure used in the design of the two simulation studies, which is the same provided in the previous chapter, but adapted to the nonlinear model in question. Then, Section 4.3 considers the specific filter's settings, experimental results, remarks and conclusions for both simulation studies. Finally, Section 4.4 reports some final remarks.

4.1 Synthetic Nonlinear Model Under Study

To illustrate how the filters described in Chapter 2 perform in a nonlinear context, we use as a benchmark nonlinear model the synthetic model used by Van der Merwe et al. (2001). This model has a state transition equation given by

$$x_t = 1 + \sin(\omega\pi(t-1)) + \phi_1 x_{t-1} + \eta_t \tag{4.1}$$

where the state noise η_t follows a Gamma distribution with shape and scale parameters given by $a = 3$ and $b = 1/2$, respectively. That is, $\eta_t \sim \mathcal{G}(3, 1/2)$, and thus its mean and variance are $3/2$ and $3/4$, respectively.

On the other hand, the measurement equation is specified by

$$y_t = \begin{cases} \phi_2 x_t^2 + \nu_t & t \leq T_h, \\ \phi_3 x_t - 2 + \nu_t & t > T_h, \end{cases} \tag{4.2}$$

where the measurement noise v_t follows a Gaussian distribution with mean zero and variance $\sigma^2_{v_t}$. Notice that the measurement equation has a different specification depending on a threshold value defined in terms of the time index t.

The above equations (4.1) and (4.2) clearly specify a nonlinear, non-Gaussian and non-stationary dynamic model state-space formulation. An exemplar graphical representation of the generated univariate data y_t and state values x_t, is displayed in Figure 4.1. The same exemplar run will be considered throughout this chapter.

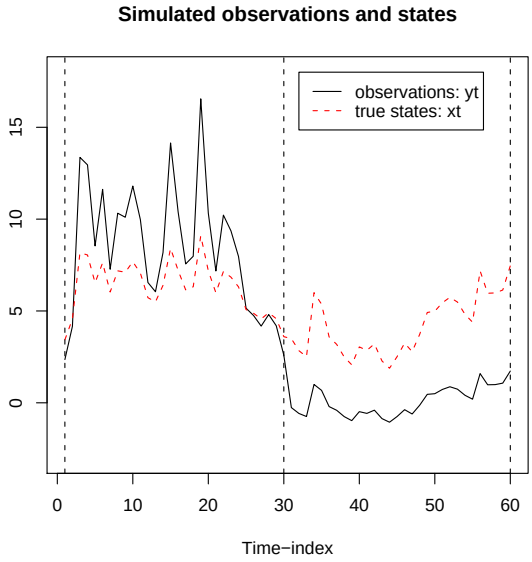


Figure 4.1: An example of the generated data $\boldsymbol{y}_t$ (black/continuous), and simulated states $\boldsymbol{x}_t$ (red/dashed) for the synthetic nonlinear model specified in equations (4.1) and (4.2).

The authors Van der Merwe et al. (2001) choose this particular synthetic nonlinear dynamic model to show the potential superior performance of the UPF algorithm. We choose it mainly to get acquainted with the implementation issues of the UKF, EPF and UPF algorithms and to assess their behavior in contrast to more known approaches, say the EKF, SIR PF, and ASIR PF.

Notice that the four competing particle filter variants (and the two Kalman-based algorithms) usually appear scattered in the literature, and we aim to confront them under exactly the same experimental conditions in order to test first their filtering (only the states) performance, to get some further insight on their functioning and consequently have a better understanding of the inner features of filters studied. This acquired better understanding proves to be useful in future simulation studies dealing not only with states-estimation (plain filtering) of non-standard dynamic state-space models, but also with the simultaneous estimation of states and parameters.

4.2 General Procedure for Simulation Design

The two simulation studies carried out in this chapter follow the same general simulation procedure presented in Section 3.2 of Chapter 3, which undertakes the following three steps:

- **STEP I**: Data and state generation
- **STEP II**: Filtering estimation
- **STEP III**: Filtering performance criteria computation

Following, we provide a detailed description of the aforementioned general simulation steps. Notice that within every simulation step, we further specify the instructions needed to carry out the MC experiments with the nonlinear benchmark model at hand.

STEP I: Data and State Generation

Generate $S = 100$ realizations of the chosen synthetic nonlinear dynamic model. That is,

(Ia) Specify the variance $\boldsymbol{\sigma}^2_{v_t} = 0.00001$, the parameters $a = 3$ and $b = 1/2$ for the Gamma error terms, the time series data length $T = 60$, a threshold value $T_h = T/2 = 30$ and the other known model parameter values $\omega = 0.04$, $\phi_1 = 0.5$, $\phi_2 = 0.2$, and $\phi_3 = 0.5$.

(Ib) Generate the random numbers $\boldsymbol{\eta}_t$ and $\boldsymbol{\nu}_t$. In this case, $\boldsymbol{\eta}_t$ is generated from a univariate gamma distribution $\boldsymbol{\eta}_t \sim \mathcal{G}(a, b)$, and the random numbers $\boldsymbol{\nu}_t$ from a univariate normal distribution $v_t \sim \mathcal{N}(0, \boldsymbol{\sigma}^2_{v_t})$.

(Ic) Simulate the state-values $\boldsymbol{x}_t$ and data $\boldsymbol{y}_t$, $t = 1, \ldots, T$, from the transition equation (4.1) and the measurement equation (4.2), respectively.

(Id) Repeat (Ia)–(Ic) $S = 100$ times.

STEP II: Filtering Estimation

For each nonlinear filter f, obtain both the statistical and computational measure of performance of the studied nonlinear filter f. These are based on the root mean square (RMSE) and on the CPU time. That is, assuming all model parameters are known and given the simulated data $\boldsymbol{y}_{1:T} = y_1, \ldots, y_T$ obtained in step (I), for replication set i, $i = 1, \ldots, S$, proceed to

(IIa) Compute the filtering estimates $\hat{\boldsymbol{x}}^f_{t,[i]}$, $t = 1, \ldots, T$ using the nonlinear filter in question, say f. Recall that $f \in \{\text{EKF, UKF, SIR PF, ASIR PF, EPF, UPF}\}$.

(IIb) Compute $RMSE^f_{[i]}$: the RMSE over time index $t = 1, \ldots, T$ with equation (3.4).

(IIc) Compute $CPU^f_{[i]}$: the total elapsed time for a total of T observations with equation (3.5).

(IId) Repeat steps (IIa)–(IIc) $S = 100$ times.

STEP III: Filtering Performance Criteria Computation

(IIIa) In step (IIb), we end up with $S = 100$ estimates of the RMSE: $RMSE^f_{[i]}$. Based on these, obtain the mean and the variance of the root mean square (RMSE) computed over time and over replication sets using equations (3.6) and (3.7), respectively.

(IIIb) In step (IIc), we end up with $S = 100$ CPU elapsed-time estimates: $CPU^f_{[i]}$. Based on these, obtain the mean CPU elapsed-time computed over replication sets using (3.8).

For completion, the reader may refer to two sketches created in Chapter 3 for a better illustration of the simulation design and performance criteria used. Specifically, the sketch in Figure 3.2 illustrates the criteria for comparing the non-simulation based filters. The corresponding sketch for the different PF variants under study is found in Figure A.1 of Appendix A.

4.3 Simulation Results, Remarks and Conclusions

As aforementioned, we perform two simulation studies. In fact, the first can be considered a subset of the second. Herein, we explicitly provide the simulation settings and results for each of the two simulation studies.

4.3.1 Simulation Study I: Mimic an Existing Study

In a first simulation study, we mimic the UPF authors, who conduct a Monte Carlo experiment in order to assess the potential superior performance of the UPF filter over other PF variants as well as over two non-simulation based filters; see Van der Merwe et al. (2001). Our aim is two-fold: 1) To get acquainted with the implementation issues of the EPF and UPF filters, and 2) to confirm the results of Van der Merwe et al. (2001).

To achieve our purpose, we use exactly the same settings and procedure as the authors Van der Merwe, Doucet, de Freitas, and Wan (2001). These general settings are:

- Competing filters:
  - Non-simulation based: EKF, UKF
  - Simulation based: SIR, EPF and UPF
- Comparison criterion: RMSE
- Resampling scheme: Residual resampling
- Number of replications: $S = 100$
- Number of particles: $N_p = 200$

- Time series length: $T = 60$

For the UKF-based filters, the specific settings $\alpha = 1$, $\beta = 0$, and $\kappa = 2$ are used; see main-programm R-code on page 261, Appendix A. According to literature, these values are optimal for the scalar case, like the one consider here; see Van der Merwe (2004).

The results for simulation study I, are presented in Table 4.1. Therein we report the statistical performance of the different nonlinear filters under study, provided by the mean and the variance of the estimated RMSE computed over time and over replications. Additionally, in square brackets, we report the estimated results published in Van der Merwe et al. (2001).

Table 4.1: Summary of simulation study I with $N_p = 200$

	RMSE[a] **(RES**[c]**)**			
Filter	**Mean**		**Var**	
SIR	0.415	[0.424]	0.056	[0.053]
EPF	0.312	[0.310]	0.016	[0.016]
UPF	0.073	[0.070]	0.007	[0.006]
EKF[d]	0.399	[0.374]	0.017	[0.015]
UKF[d]	0.298	[0.280]	0.012	[0.012]

[a] Root mean square error
[b] Results in square brackets are from Van der Merwe et al. (2001)
[c] Residual resampling
[d] Clearly, the EKF and UKF do not need resampling

To illustrate the filtering performance of the five competing filters, for the same exemplar run used before, a graphical comparison of the simulated state values and its filtering estimates is displayed. In other words, for a particular set of data, the true state values x_t and the estimated states evolution $\hat{x}_{t|t}$, $t = 1,\ldots,T$ are plotted together. Figure 4.2 shows the evolution of the estimated states for the non-simulation based filters EKF and UKF together with the true states values. Likewise, Figure 4.3 displays the estimated states evolution for the simulation-based filters, using in this case residual resampling. Notice that at any time-index $t = 1,\ldots,T$, the EKF and UKF filters yield directly the estimates of the states $\hat{x}_{t|t}$. The particle filters, however, yield an estimation of the whole posterior PDF of the states $P\left(\boldsymbol{x}_t|\boldsymbol{y}_{1:s}\right)$ and based on it, the states posterior mean estimate $\hat{x}_{t|t}$ is computed.

Remarks and Conclusions for Simulation Study I

Based on simulation I results displayed in Table 4.1, considering $T = 60$ and only $N_p = 200$ particles, we make the following remarks and conclusions regarding the performance of the five filters under study when handling the chosen synthetic nonlinear dynamic model:

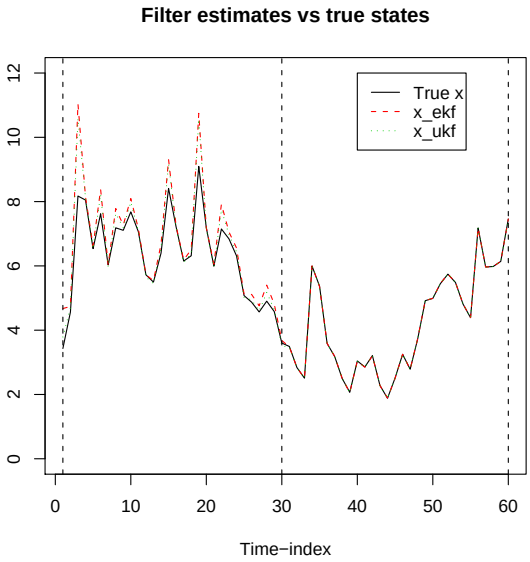


Figure 4.2: Evolution of simulated states x_t and estimated states $\hat{x}_{t|t}$ for synthetic nonlinear model specified in equations (4.1) and (4.2). Results shown for the EKF and UKF non-simulation based filters.

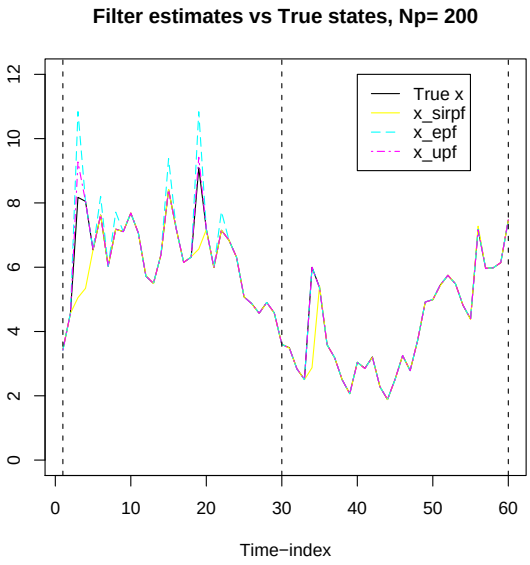


Figure 4.3: Evolution of simulated states x_t and estimated states $\hat{x}_{t|t}$ for the synthetic nonlinear model specified in equations (4.1) and (4.2). Results shown for three simulation based filters (SIR, EPF and UPF) with $N_p = 200$ particles.

- First, focusing on the mean and variances of the RMSE in Table 4.1, we conclude that our estimated results show great concordance with the literature results. Although small differences

are observed in the statistical performance of all filters, we consider this natural since we are not only dealing with generated data sets that are distinct from the ones used by Van der Merwe et al. (2001), but also applying three filters based on simulations. Thus, the observed behavior is characteristic of MC experiments.

- Second, we are able to confirm the findings of the authors Van der Merwe et al. (2001) who conclude that the UPF variant can be able to outperform other nonlinear algorithms. Indeed, for a rather small number of particles $N_p = 200$, the UPF shows a clearly superior statistical performance, in terms of the RMSE, over the other nonlinear algorithms under study.

- Thus, in a complex context like the one defined by the nonlinear model at hand, the results confirm that the non-simulation based filters (EKF and UKF) are not really able to provide an optimal filtering solution. The particle filtering methodology, however, is able to provide superior statistical performance as indicated by the mean-RMSE of the UPF variant. In this case the SIR PF variant shows worse performance (mean-RMSE slightly above the EKF's RMSE), followed by the EPF, with mean-RMSE values slightly above the UKF's RMSE.

- Therefore, based on the above results with $N_p = 200$ and using a residual resampling strategy, we confirm that the UPF is able to outperform all four competing filters: the EKF, the UKF, the SIR and the EPF. Recall that the model at hand is very complex, being nonlinear, non-stationary and non-Gaussian, where the measurement equation has a different specification depending on a threshold value defined in terms of the time-index t.

Next, a second simulation study is carried out, which basically is an extension of the Monte Carlo study presented above.

4.3.2 Simulation Study II : Extension of First Simulation Study

This simulation study extends the former Monte Carlo experiment by incorporating the following new elements: 1) a very popular particle filter variant: the ASIR; 2) another resampling scheme: stratified resampling; 3) another measure of performance of the filters: CPU time; and 4) the effect of increasing the number of particles on the performance of the different particle filters which are chosen.

Summary of Simulation Settings

In this case, the simulation settings of the first Monte Carlo study are also extended (new ones in boldface) and summarized by:

- Competing filters:
  - Non-simulation based: EKF, UKF
  - Simulation based: SIR, EPF, UPF, and **ASIR**

- Comparison criteria: RMSE and **CPU time**
- **Effect of the resampling strategy**: Residual vs **Stratified**
- **Explicit effect of the increase of the number of particles**
- Number of replications: $REP = 100$
- Number of particles: $N_p = 200$, **1000, 2000, 5000** and **10000**
- Time series length: $T = 60$

With this second simulation study, we aim to asses the filtering performance of the four particle filter variants listed above, using the nonlinear model at hand as a benchmark. Also, for completion, we include the two aforementioned analytical filters: the EKF and the UKF algorithms.

To achieve our purpose, we conduct a Monte Carlo study following the same general simulation design as in simulation study I, but incorporating the new elements that are mentioned and highlighted above.

In the sequel, we present the simulation results, remarks and conclusions regarding the extension of the first MC study. This we consider to be the main contribution of this chapter.

Experimental Results

In Table 4.2, the simulation results corresponding to the same number of particles used in simulation I, say $N_p = 200$ particles, are reported. Later, however, we present results which reflect the effect of increasing the number of particles on the quality of the estimations for the simulation based filters which we studied. This table is organized in three different blocks, where the first two correspond to the RMSE values obtained using the residual and stratified resampling schemes, respectively. Each one of these blocks is composed of two columns containing the two measures: Mean(RMSE) and Var(RMSE). Notice that, for completion, the non-simulation based EKF and UKF filters are included in the simulation study; results shown at the bottom of the Table 4.2. The third block contains the mean CPU elapsed-time (in seconds) computed over replications using formula (3.8) for both resampling schemes.

Recall that the newly introduced measure of performance of the filters is based on the mean CPU elapsed-time computed over replications, and it represents the time a filter takes for estimating the states of a time series of length T.

All the reported simulation results are later commented on, and, for the sake of a better understanding of them, some illustrative plots are constructed.

Remarks and Conclusions for Simulation Study II

Based on the results of simulation II displayed in Table 4.2, considering $T = 60$ and only $N_p = 200$ particles, we make the following remarks and conclusions regarding the performance of the six filters under study when handling the chosen synthetic nonlinear dynamic model:

Table 4.2: Summary of simulation study II with $N_p = 200$

	RMSE[a] (RES[b])		RMSE (STR[c])		Mean CPU Time[d]	
Filter	**Mean**	**Var**	**Mean**	**Var**	**RES**	**STR**
SIR	0.415	0.056	0.407	0.058	0.201	0.117
ASIR	0.423	0.056	0.439	0.059	0.262	0.264
EPF	0.312	0.016	0.315	0.017	0.233	0.148
UPF	0.073	0.007	0.071	0.008	5.784	5.566
EKF[e]	0.399	0.017			0.082	
UKF[e]	0.298	0.012			0.058	

[a] Root mean square error
[b] Residual resampling
[c] Stratified resampling
[d] Mean CPU elapsed-time over replications in seconds
[e] EKF and UKF do not need resampling

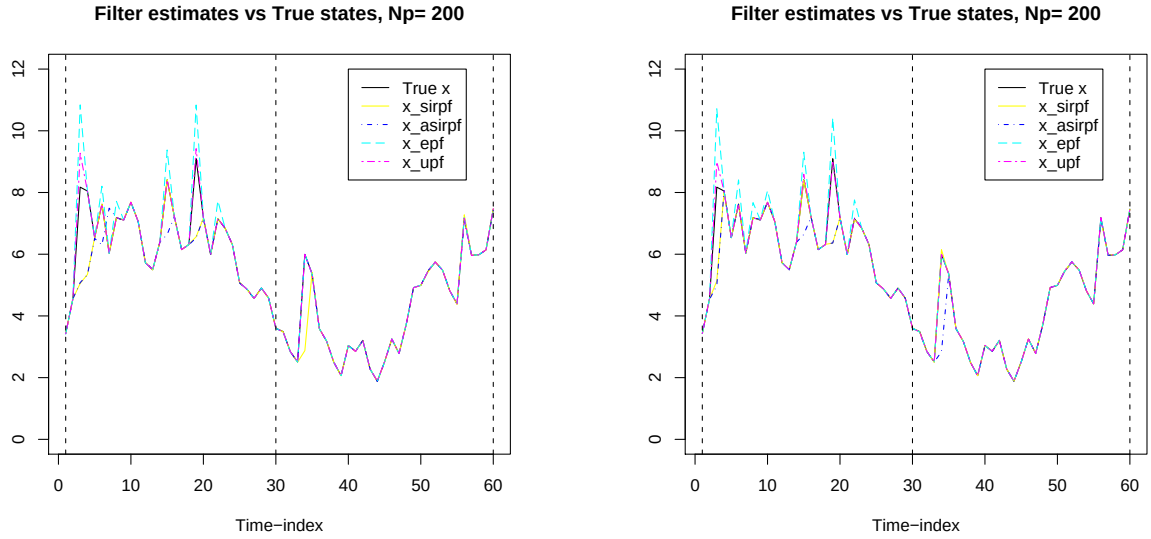
First, we refer to the inclusion of the ASIR particle filter variant into the second simulation study, and find that it does not alter the last conclusion pointed out in simulation study I. That is, using a rather small number of particles $N_p = 200$, the UPF particle filter variant still has the best statistical performance. Indeed, the UPF shows a very small mean-RMSE (around 0.07) in contraposition to the other three competing particle filter variants and to the two analytical filters; focus on Table 4.2. As seen, the EPF is the second-best particle filter variant, followed by the SIR and the ASIR. Thus in this case, the latter two particle filter variants show worse statistical performance. Additionally, as seen already in Simulation I, between the two analytical filters under study, the UKF displays best performance.

Notice that, the mean-RMSE yielded by the EPF particle filter variant and the analytical UKF filter are very close. Likewise, the mean-RMSE of the SIR and the ASIR particle filter variants are close to the mean-RMSE value yielded by the EKF.

Second, we assess the effect of the inclusion of the stratified resampling scheme, and find that it does not change the general conclusions stated in the previous remark: that the UPF outperforms the other three competing particle filter variants (EPF, SIR and ASIR). Focusing our attention on the statistical performance within each particle filter variant, we find that the mean-RMSE values obtained through the two chosen resampling schemes practically coincide; focus on the mean-RMSE per rows in Table 4.2.

To illustrate the filtering performance of the four simulation-based filters in question, a graphical comparison of the true state-values and its filtering estimates is provided. Specifically, Figure 4.4(a) shows the evolution of the estimated states under residual resampling. Likewise, Figure 4.4(b) displays the evolution of estimated states using the stratified resampling scheme. Although these plots are

based on a single exemplar run, they perfectly illustrate the conclusions stated above.



(a) Simulated vs estimated states; Residual Case.

(b) Simulated vs estimated states; stratified Case.

Figure 4.4: An example of a synthetic nonlinear non-Gaussian and non-stationary dynamic model specified in equations (4.1) and (4.2); fixed known parameters.

Third, focusing on the performance of the four competing particle filter variants in terms of the computational time, we conclude that among the simulation-based filters, the SIR is computationally the least-expensive algorithm with mean CPU time values around 0.20 [0.11] (average under residual [stratified] resampling) seconds, followed by the EPF (0.23 [0.15]), the ASIR (0.26 [0.26]) and the UPF (5.78 [5.57]) filter. Thus, for $N_p = 200$ particles, clearly the UPF and the ASIR show worse computational performance; focus on the last two columns of Table 4.2. Clearly, the non-simulation based filters UKF and EKF are the most computationally efficient algorithms with about 0.06 and 0.08 mean CPU times values, respectively. Therefore, the above simulation results suggest that for the nonlinear model at hand, the stratified resampling scheme is, in general, computationally more efficient than its counterpart residual resampling. Notice, however, that the discrepancy in the computational efficiency between resampling schemes is present in a less degree in both: the ASIR PF and the UPF particle filter variants. By contrast, as can be seen on the first four columns of Table 4.2, the statistical efficiency is practically not affected by the resampling strategy; see also the illustration on Figure 4.4.

As mentioned in Section 2.4, our first incursion into the sequential Monte Carlo methodology began with Kitagawa's 1996 paper. In that paper, a comparative study of different resampling strategies (residual resampling not included) is carried out, and the systematic stratified scheme was shown to have a superior performance. Based on Kitagawa's comparison of resampling strategies and also on the conclusions arising from our simulation study II, in the remainder of this work, we restrict ourselves to solely using the stratified resampling scheme, just as is done in the previous chapter. We remark that

we have always worked with the stratified resampling algorithm, and that we have explicitly aimed and hopefully fulfilled its efficient implementation in R language.

Next, considering only the stratified resampling scheme, we explore the impact of increasing the number of particles on both the mean-RMSE and the mean-CPU time of the competing particle filters.

Effect of Increasing the Number of Particles

Herein, for the complex nonlinear model at hand with time-series-length $T = 60$, we assess the impact of increasing the number of particles on the statistical (mean-RMSE) and computational (mean-CPU time) performance of the four competing particle filter variants: the SIR, ASIR, EPF and UPF.

The simulation results for a number of particles in the set $N_p \in \{200, 1000, 2000, 5000, 10000\}$ are reported in Table 4.3. For the sake of clarity, we also construct Figure 4.5 (on page 126) which allows to see at a glance how increasing the number of particles affect the statistical and computational performance of the four particle filter variants under consideration.

Based on results reported in Table 4.3 and plotted in Figure 4.5 the following conclusions arise:

- For the UPF variant, we find that a greater number of particles only slightly effects its statistical performance. In this case, the estimated mean[variance] of the RMSE decreases from about 0.07[0.01] to 0.05[0.01] as the number of particles N_p increase from 200 to 2000; a higher increase of the number of particles has practically no further effect. Notice, however, that for a rather small number of particles, one obtains reasonably good statistical performance, though this filter has the most expensive computational cost.

- The statistical performance of the EPF particle filter variant is practically not influenced by increasing the number of particles. Actually, for this filter, the estimated mean[variance] of the RMSE decreases from about 0.32[0.02] to 0.30[0.02] as the number of particles N_p increases from 200 to 1000, but then remain more or less constant, with mean-RMSE values close to but below to the UKF estimated values 0.30[0.01]. Computationally speaking, we find that the EPF is the second-least expensive particle filter.

- The statistical performance of both the SIR and the ASIR particle filter variants are greatly affected by the increase of the number of particles. Additionally, both nonlinear filters show very close mean-RMSE values for the model at hand, with ASIR values slightly higher up to $N_p = 5000$ particles. However, in this case, the SIR PF has a smaller computational cost. Indeed, we find that the SIR is the least-expensive particle filter variant and the ASIR the second most-expensive one, being the UPF the most expensive.

  We remark that both the SIR and the ASIR particle filter variants are able to equate the UPF's mean-RMSE value (0.05[0.01]; $N_p = 2000$) with 5000 particles. Notice also that the lowest mean-RMSE value is yielded by the SIR/ASIR with $N_p = 10000$ particles. This suggests that simpler (mainly, in terms of implementation issues) particle filter variants like the SIR and the ASIR are

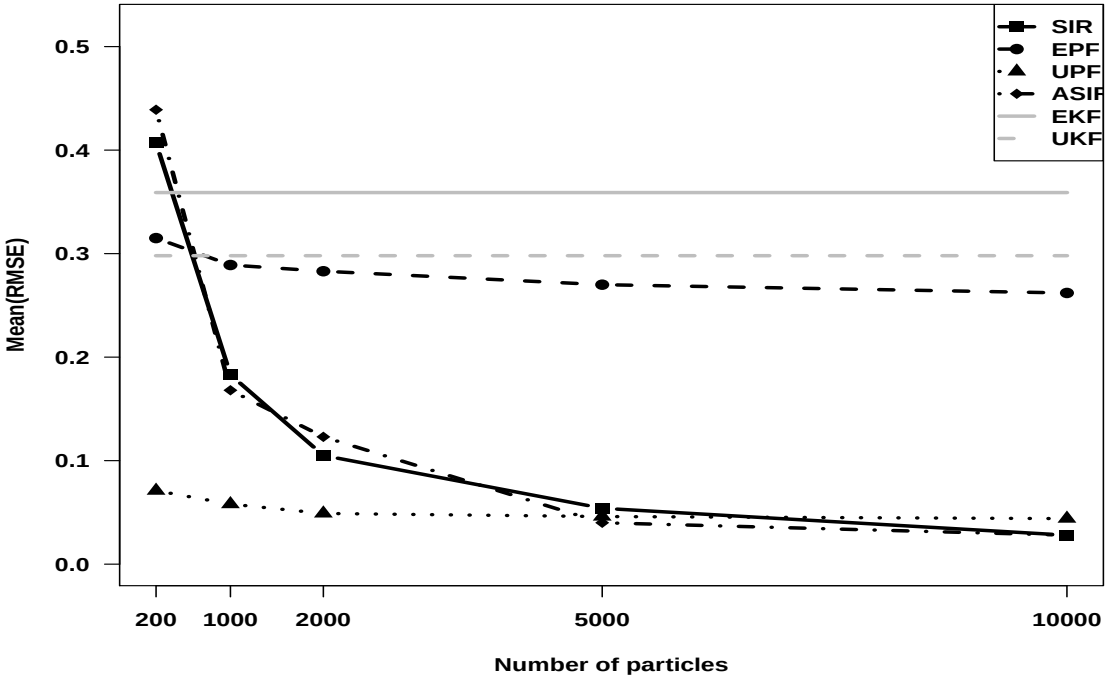
Table 4.3: Summary of Monte Carlo sub-study: Effect of increasing N_p

		RMSE[a] (STR[b])		
Filter	N_p[c]	**Mean**	**Var**	**Mean CPU Time**[d]
SIR PF	200	0.407	0.058	0.117
	1000	0.183	0.062	0.295
	2000	0.105	0.040	0.484
	5000	0.054	0.036	1.174
	10000	0.028	0.014	2.684
EPF	200	0.315	0.017	0.148
	1000	0.289	0.016	0.373
	2000	0.283	0.017	0.631
	5000	0.270	0.016	1.919
	10000	0.262	0.016	3.847
UPF	200	0.071	0.008	5.566
	1000	0.058	0.007	27.539
	2000	0.049	0.006	56.374
	5000	0.046	0.006	137.928
	10000	0.044	0.006	280.954
ASIR PF	200	0.439	0.059	0.264
	1000	0.168	0.059	0.754
	2000	0.123	0.048	1.338
	5000	0.040	0.020	3.049
	10000	0.028	0.014	6.634

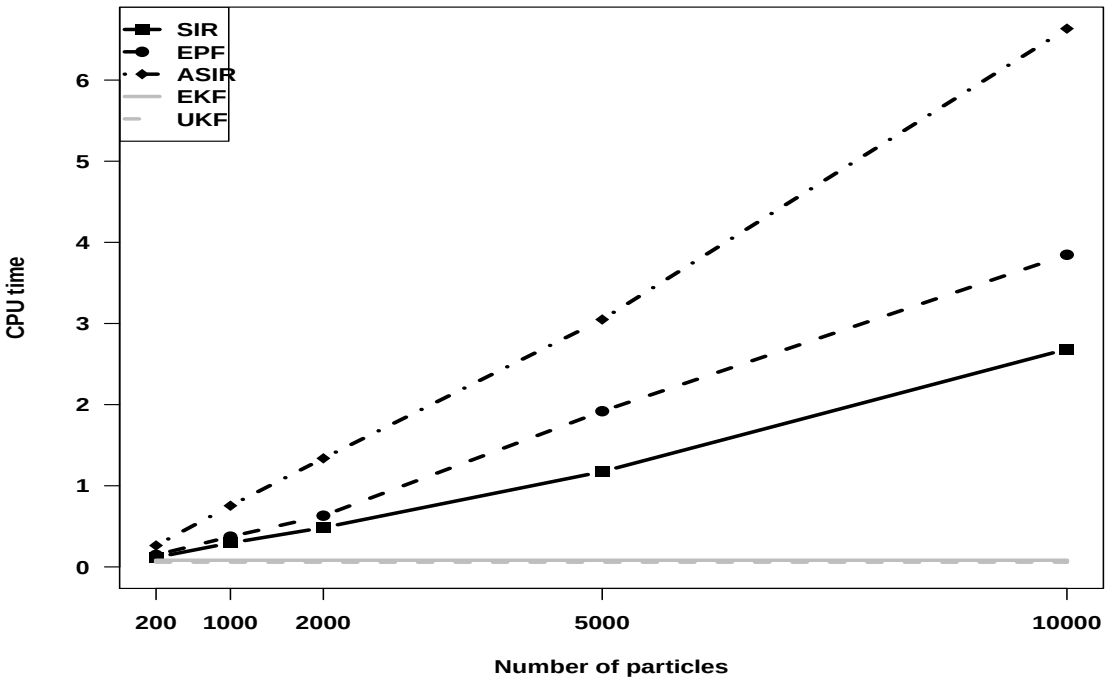
[a] Root Mean Square Error
[b] Stratified Resampling
[c] Number of particles
[d] Mean CPU elapsed-time over replications in seconds

able to reach the performance of more recent and complex filters like the UPF (already efficient enough with a rather small number of particles) at the expense of memory cost, but not necessarily of more computational cost.

- Therefore, computationally, in general, the most expensive nonlinear filter is the UPF, followed by the ASIR, the EPF and the SIR particle filter variant; focus on last column of Table 4.3 and bottom panel of Figure 4.5. From the statistical point of view, we get mixed results as described above.



(a) Mean(RMSE) vs number of particles N_p. For completion, also the mean-RMSE values for the non simulation-based filters, the EKF (grey/continuous) and the UKF (grey/dashed), are plotted.



(b) CPU time (seconds) vs number of particles N_p; the very high UPF CPU time values are not shown, but are presented in Table 4.3.

Figure 4.5: Synthetic Nonlinear Model: Impact of increasing the number of particles on the statistical and computational estimation performance of the four competing particle filters indicated in panel a) by the mean(RMSE) and in panel b) by the CPU time in seconds; $T = 60$ and $N_p = 200$ are used. Results for the UPF are not shown, but are reported in Table 4.3

4.4 Final Remarks and Conclusions

In summary, this chapter considers a Monte Carlo benchmark study of four competing particle filter variants and two Kalman-based filters to assess their performance in a complex nonlinear framework. Recall that the benchmark state-space model – specified in equations (4.1) – (4.2) and taken from the literature – is not only nonlinear, but also non-stationary and non-Gaussian. Additionally, notice that the nonlinearity appears in the measurement equation and is specified via a threshold value defined in terms of the time index t. Therefore, in a complex nonlinear context like this with $T = 60$, our simulation results indicate that:

- According to simulation I with $N_p = 200$ particles:

  For the nonlinear model at hand, the results of our simulation study I allow us to confirm the findings of the authors Van der Merwe et al. (2001), who state that the UPF variant can be able to outperform other nonlinear algorithms; all simulation I results are based on $N_p = 200$ particles and residual resampling. Recall that this first simulation study considers only a statistical comparison criterion, which is known to be based on the RMSE.

  Thus, based on results reported in Table 4.1, we conclude that with $N_p = 200$ and under residual resampling, the UPF variant is found to be the best filter, followed by the EPF (with mean-RMSE close, but below, to the UKF estimated values) and the SIR PF variant (showing a statistical performance close to the one yielded by the EKF).

- According to simulation II with $N_p = 200$ particles:

  With respect to the inclusion of the ASIR particle filter variant into the Monte Carlo experimentation, we find that it shows a very similar statistical performance to the SIR particle filter variant. The ASIR has, though, a higher computational cost; see Table 4.2.

  The inclusion of the stratified resampling scheme does not change the general conclusions stated in simulation study I: that the UPF statistically outperforms all competing particle filter variants (EPF, SIR and ASIR) and the two Kalman-based approaches (EKF and UKF); focus on the first four columns of Table 4.2 and for completion on Figure 4.4.

  Focusing on the statistical performance within each particle filter variant, we find that the mean-RMSE values obtained through the two chosen resampling schemes practically coincide; focus on the mean-RMSE per rows in Table 4.2 and on Figure 4.4. These results suggest that the choice between residual and stratified resampling does not play a distinctive role in the statistical performance of the competing sequential Monte Carlo filters for the model at hand.

  Focusing on the computational performance within each particle filter variant, we find that the mean CPU time values obtained under residual resampling are generally greater than those under stratified resampling; focus on the mean-CPU times in the last two columns of Table 4.2. Thus, these results suggest that the choice between residual and stratified resampling does play

a distinctive role in the computational performance of the competing sequential Monte Carlo filters for the model at hand.

Therefore, the above simulation results suggest that for the nonlinear model at hand, the stratified resampling scheme is, in general, computationally more efficient than its counterpart residual resampling. In case of the ASIR and the UPF particle filter variants, however, the mean-CPU times are very similar. In contrast, all above findings suggest that, in general, the stratified resampling scheme is computationally more efficient. As aforementioned and justified, in the sequel we solely use the stratified resampling scheme.

- According to simulation II considering the impact of increasing the number of particles used in the estimation procedure, $N_p \in \{200, 1000, 2000, 5000, 10000\}$:

  Although for $N_p = 200$ particles the UPF is found to be the best nonlinear filter, we stress that both the SIR and the ASIR particle filter variants are able to reach the UPF statistical performance at the cost of increasing the number of particles (consequently more memory costly), but with a lower computational burden. To illustrate this situation, refer to Table 4.2 and Figure 4.5(b); the UPF CPU times are not shown on the plot because they are too high, but they are reported in the mentioned table. Both the SIR and the ASIR reach similar mean-RMSE values to the UPF particle filter variant with $N_p = 5000$; when $N_p = 10000$, they equate or even slightly improve its statistical performance. We thus consider that $N_p = 5000$ is a good enough number of particles for the SIR and ASIR filters as a trade-off between computational and statistical performance.

  When increasing the number of particles from 200 to 2000, we find that both the UPF and the EPF show a rather slight decrease of the mean-RMSE, afterwards their values remain practically the same. In contrast, both the SIR and the ASIR are highly affected by the increase of the number of particles, though the impact is stronger at lower settings values of N_p. We observe that an increase of the number of particles from $N_p = 200$ to $N_p = 5000$ produces a reduction of the mean-RMSE from 0.41 to 0.05 for the SIR, and from 0.44 to 0.04 for the ASIR, with the ASIR being computationally more costly.

  As depicted on Figure 4.5(b), the effect of increasing the number of particles on the computational performance of the four competing particle filter variants is very noticeable. Clearly, the UPF shows in this case a worse performance, followed by the ASIR, the EPF and the SIR when considering $N_p = 200$ particles or more.

  With the aim to provide fair comparisons among the particle filter variants under scrutiny, we decide to undergo an additional experimental study to assess the computational performance of three out of four competing particle filters for a fixed mean-CPU time; see results shown in Table 4.4. Notice that we do not include the EPF particle filter variant in this sub-study because, as shown before, it hardly gets affected by the increase of the number of particles (in terms of RMSE). Thus, for a fixed mean CPU elapsed-time of about 7 seconds, we find that:

The SIR particle filter variant outperforms the UPF, if $N_p = 25000$ particles are used. Specifically, the SIR with (RMSE[mean,var]=[0.02,0.01],$N_p = 25000$) improves the respective UPF values ([mean,var]=[0.07,0.01], $N_p = 250$). We stress that the SIR already gets a mean-RMSE value of about 0.03 with 10000 particles and a mean CPU-time of about three seconds. Notice also that the SIR yields a mean-RMSE value of about 0.05 with 5000 particles and a mean CPU-time slightly larger than one second.

Likewise, the ASIR particle filter variant statistically outperforms the UPF, if $N_p = 10000$ particles are used. Specifically, the ASIR with (RMSE[mean,var]=[0.03,0.01],$N_p = 10000$) improves the respective UPF values ([mean,var]=[0.07,0.01], $N_p = 250$). Notice also that the ASIR yields a mean-RMSE value of about 0.04 with 5000 particles but with a mean CPU-time of about three seconds; when 10000 particles the ASIR takes about 7 seconds.

Therefore, for a fixed mean CPU time, the SIR proves (empirically) to be an alternative option to the UPF when filtering the states of the nonlinear model in question, at the expense of higher memory requirements. As a second alternative, we have the ASIR; refer to results in Table 4.4. We stress, however, that the UPF already shows a satisfactory computational performance with a very low number of particles.

Table 4.4: Statistical Performance for fixed CPU-time of 7 seconds

Filter	**Mean**	**Var**	N_p
SIR PF	0.02	0.01	25000
ASIR PF	0.03	0.01	10000
UPF	0.07	0.01	250

Based on the Monte Carlo experiments, our results clearly highlight that when dealing with non-standard time series models, the particle filtering methodology is a useful and efficient alternative approach. The superior performance of the particle filtering methodology does not take us by surprise, since it was precisely created to tackle the estimation of nonlinear models, where previous approaches had difficulties.

The simulation results also indicate that an increase in the number of particles improves the precision of the filtering estimates. However, we should keep in mind that more particles increase the computational burden. As shown (empirically) for the nonlinear model in question, the choice of a specific filter over another depends not only on the type of model at hand but also on the practitioner's knowledge of the specific filters available, having in mind the memory and CPU time requirements.

We consider thus, that the particle filters prove their superior performance when dealing with non-standard dynamic models with a more complex structure as covered in the previous chapter. Herein, we have illustrated and (empirically) shown that when departing from an ideal context with relevant non-linearities the particle filtering methodology studied in this work is a valid alternative, being able

to outperform Kalman-based nonlinear algorithms such us the Extended Kalman filter or the Unscented Kalman filter. We remark that the main goal of our research strives in obtaining the simultaneous estimation of states and possibly-fixed unknown model parameters. In such a context, the plain use of the KF is not longer a suitable solution [1]. Therefore, our interest is in studying the behavior of a simulation based approach known as particle filtering methodology having in mind its possible later use for estimating simultaneously the state and parameters. The results of the MC experiments help us to get some useful insight for later simulation studies and for the suitability of the particle filtering methodology for filtering/learning more complex models.

The next Chapter describes how the particle filtering methodology can be used to estimate simultaneously the states and fixed model parameters of a dynamic state-space model. Consequently, in later chapters, some of the filters already presented in Chapters 2–4 are modified (and used when feasible) to deal with the simultaneous estimation of state and fixed model parameters. Additionally, let us remark that since we are particularly interested in non-standard dynamic models, the non-stationary local level model studied in the first part of this thesis will be revisited (specifically, in Chapter 5) later on in the second part of this thesis dealing with the simultaneous estimation of states and involved fixed model parameters.

[1] A possible and commonly used solution is the combination of two non-simulation based methods: the KF and the Maximum Likelihood approach; see for instance (Ruiz 1994)

Part II

Simultaneous Estimation of States and Parameters

CHAPTER 5

SIMULTANEOUS ESTIMATION OF STATES AND PARAMETERS VIA THE PARTICLE FILTERING METHODOLOGY

Generally speaking, there exists a wide range of methods to solve the parameter estimation problem in state-space models. The adoption of a particular type greatly depends on the characteristics of the problem at hand. When confronted with the parameter estimation problem in a nonlinear non-Gaussian framework, several filters based on the underlying density functions have been developed; see for instance, Muñoz, Pagès, and Martí-Recober (1988), Clapp and Godsill (1998), Kitagawa (1998), Kitagawa and Sato (2001), Tanizaki (2001), Doucet et. al (2001), Liu and West (2001), Storvik (2002), Doucet and Tadic (2003), Muñoz, Márquez, and Acosta (2007), Flury and Shephard (2008), Andrieu, Doucet, and Holenstein (2010) and Lopes and Tsay (2011).

According to the literature, particle filters have become a popular approach for online estimation of nonlinear and possibly non-Gaussian state-space models. Indeed, the ideas of Chapter 2 and the results of the Monte Carlo experiments carried out in Chapters 3 and 4 clearly confirm that when dealing with non-standard time series models, the particle filtering methodology is a useful and efficient alternative approach. Remind that by 'non-standard', we mean series that exhibit non-linearities and/or non-Gaussian distributions and/or non-stationarity.

The present chapter takes up the ideas of Chapter 2 – 4 in order to estimate simultaneously the states and fixed parameters for non-standard dynamic models cast in state-space form. More specifically, we also resort to the particle filtering methodology to simultaneously estimate the original state vector $\boldsymbol{x}_t$ and the fixed and unknown model parameters. The sequential Monte Carlo methods – particle filter variants– adopted herein arrive as extensions of some of the nonlinear filters described

in Chapter 2.

Since the appearance of the first operational particle filter in 1993 (Gordon, Salmond, and Smith 1993), many variants of particle filters have been proposed either to estimate solely the states (as covered in previous chapters) or both the states and model parameters. The latter case is of interest in the present chapter. For instance, to estimate simultaneously the states and parameters, some authors like Storvik (2002), Carvalho, Johannes, Lopes, and Polson (2010) and Lopes and Tsay (2011) propose particle filter variants based on sufficient statistics.

In this work, we first focus on two existing classic particle filter variants which appeared in the literature (not based on sufficient statistics) that were proposed as an attempt to tackle the still difficult[1] problem of estimating simultaneously the states and fixed model parameters of a dynamic state-space model. These filters are Kitagawa's self organizing filter (SO; see Kitagawa (1998) and Kitagawa and Sato (2001)) and the approach of Liu and West (Liu and West 2001). Second, inspired by the ideas of these authors, we propose three particle filter variants called sampling importance resampling plus jittering (SIRJ), the extended particle filter plus jittering (EPFJ) and the unscented particle filter plus jittering (UPFJ). A particle filter variant, called by us optimal sampling importance resampling plus jittering (SIRoptJ), is also included in the undergone MC experiments. The SIRoptJ particle filter variant is basically a special case of the SIRJ algorithm, distinguished by the use of a different proposal PDF.

This chapter is organized as follows: Section 5.1 provides some preliminary remarks regarding the need for alternative parameter estimation procedures and about the reasons for the adoption of the particle filtering methodology. In Section 5.2, some general concepts needed in the context of parameter estimation are presented. For instance, the so-called augmented state vector, $\boldsymbol{l}_t$, is defined. Therein also the specific state-space formulation as well as the corresponding theoretic and approximative predictive and filtering expressions are presented. Following, Section 5.3 and Sections 5.5 – 5.7 describe the four particle filter variants chosen for filtering the augmented state vector $\boldsymbol{l}_t$, including our proposed variant called sampling importance resampling with jittering. Therein, pseudocodes for all filters are provided. Section 5.4 presents two alternative approaches to define an artificial evolution of the fixed model parameters.

Section 5.8 revisits the non-stationary local level model to estimate simultaneously the states and fixed model parameters. The augmented state-space formulation for the model at hand is specified where the unknown model parameters are assumed to be random variables and thus have a prior distribution. Therein, a Monte Carlo simulation study is performed using as a criteria the RMSE and CPU time. Notice that the same general procedure for the simulation design used in previous chapters is adapted to the problem at hand. This general procedure undertakes three steps: Data and State Generation, Filtering Estimation and Filtering Performance Criteria Computation. Specifically, only the last two steps need to be slightly modified to be able to compute the RMSE related to the two

[1] The interest on this still-difficult topic within the framework of the particle filtering methodology is evidenced by the active research done in this area aiming to bring some additional improvement over existing particle filter variants; see among others (Niemi 2009), (Carvalho, Johannes, Lopes, and Polson 2010), Lopes and Tsay (2011) and Doucet and Johansen (2011).

unknown parameters of the local level model. Also, this section assesses the impact of the signal-to-noise-ratio on the statistical performance of the competing particle filter variants and ends up with a summary of the general simulation settings for the Monte Carlo (MC) experiments. Additionally, to somehow measure the degree of degeneracy of the competing filters, the average number of unique particles (in %) at last time-index $t = T$ is reported. Experimental results are presented on page 158, Section 5.8. Finally, the potential impact of the so-called discount factor δ is explored.

5.1 Preliminary Remarks about Parameter Estimation

It is well known that when dealing with time series data, it is crucial for the researcher to find a model that adequately represents the problem at hand. In those cases, non-standard time series models are the rule rather than the exception.

In few real-world problems, one is faced with linear and Gaussian dynamic models that allow the use of standard statistical packages and/or analytical algorithms for estimating the model parameters. For instance, under the hypothesis of linearity and Gaussian errors, the recursive estimation of states and model parameters can be carried out based on the combined use of Kalman filtering and maximum likelihood; see for instance Ruiz (1994) and Pollock (2003).

As aforementioned, the data most commonly found is adequately represented by nonlinear models. Additionally, in most cases, the researcher has only the information contained in the data itself and possibly partial knowledge about model parameter values. Therefore, procedures that achieve the estimation of the unknown model parameters are desirable. By doing so, the fitted mathematical model is completely specified and, if required, forecasting could be performed[2].

The well-known seminal paper of Gordon et al. (1993) represented a time break-through for the operational use of the particle filtering methodology. We consider that Kitagawa's article (1996), who to our knowledge worked independently of Gordon et al. (1993), also makes an excellent coverage of the first operational particle filter variant, which we call sampling importance resampling (SIR) (see Algorithm 6 in Chapter 2). Indeed, both papers introduce the resampling step in order to overcome the inherent degeneracy problem of particle filters that use solely a sequential sampling importance sampling (SIS) approach. As already stated in Subsection 2.4.3, the SIR particle filter variant is widely used in different fields to estimate the states of a time series model cast in state-space form, a fact which is also shown by the extensive literature; see, for instance, Doucet et al. (2001) and references therein. An updated tutorial has been published by (Doucet and Johansen 2011), which covers classic as well as more recent particle filter variants.

Various PF variants arrive with the desire and/or the need to improve upon the SIR particle filter variant. For instance, the auxiliary sampling importance resampling (ASIR) PF described in Chapter 2 aims to reduce the variance of the particle filter weights by using a 2-stage sampling procedure. Be

[2] Forecasting is usually a matter of interest for the researcher.

reminded that in 1999 the ASIR particle filter was introduced to estimate only the state, not the parameters of various time series models.

Thus, based on historical particle filtering literature, the material covered in Chapter 2 and our simulation studies of Chapters 3 and 4, it becomes clear that the particle filtering methodology is a flexible and efficient approach to handle the state estimation of non-standard dynamic models. All these results motivate us to extend the use of the particle filtering methodology to tackle the estimation of unknown fixed model parameters in a possibly nonlinear and/or non-stationary context.

Various authors have used and adapted the particle filtering approach to perform recursive parameter estimation. In other words, particle filters originally designed only to estimate the states are modified to also perform the estimation of model parameters. For instance, Kitagawa (1998) makes an extension of the SIR particle filter presented in Kitagawa (1996) by first defining the augmented state vector and then estimating simultaneously the original state variables and the model parameters; his filter is called 'self organizing filter'. Additionally, Liu and West (2001) suggest to apply Kernel smoothing and shrinkage ideas, originally used by West (1993) in another context, in order to tackle the impoverishment problem that we know is inherent to particle filters.

In Acosta et al. (2003), we adapt the SIR particle filter variant presented in Kitagawa (1998) to estimate the parameters of an AR(1) time series model, under the restriction of no measurement noise in the corresponding state-space formulation. It is after this work that we first propose a modified version of the SIR particle filter variant, called by us SIRJ, being actually an extension of Kitagawa's 1998 self organizing filter (SOF); see Acosta et al. (2004) and Muñoz et al. (2004). In Muñoz, Márquez, and Acosta (2007), we apply the SIRJ approach to estimate the states and parameters of threshold volatility models.

In this thesis, we further explore (when feasible) how the jittering step affects the performance of the EPF and UPF variants; these filters are called by us extended PF with jittering (EPFJ) and unscented PF with jittering (UPFJ), respectively. Notice that when the KF is used instead of the EKF, the EPFJ variant is then called Kalman PF with jittering (KPFJ). We consider the proposed SIRJ, SIRJopt, EPFJ (or KPFJ) and UPFJ PF variants as the main methodological contributions of this chapter; especially the first three. Specifically, we revisit the non-stationary local level model to perform the simultaneous estimation of states and parameters via four PF variants whose use is considered appropriate in this context. The performance of the filters studied is assessed through MC simulations considering also the impact of the signal-to-noise ratio and two settings of the discount factor; details are further presented in Section 5.8.

5.2 General Concepts

In order to estimate simultaneously the state and parameters, it has become a common practice to augment the original state vector $\boldsymbol{x}_t$ by appending the parameter vector $\boldsymbol{\Theta}$, see e.g. Muñoz (1988) and

Kitagawa (1998). In such cases, the so-called augmented state vector is defined as:

$$\boldsymbol{l}_t = \begin{bmatrix} \boldsymbol{x}_t \\ \boldsymbol{\Theta} \end{bmatrix}.$$

5.2.1 Augmented State-Space Model Formulation

Herein, the general state-space formulation provided by equations (2.1) and (2.2) is adapted for the augmented state-space vector $\boldsymbol{l}_t = (\boldsymbol{x}_t, \boldsymbol{\Theta})'$. That is, the parametric state-space formulation for a dynamic model dealing with an augmented state vector can be described by the following two equations (Muñoz 1988; Kitagawa 1998):

$$\boldsymbol{l}_t = \tilde{f}(\boldsymbol{l}_{t-1}, \boldsymbol{\eta}_t), \qquad \textit{(Transition equation)} \tag{5.1}$$

$$\boldsymbol{y}_t = \tilde{h}(\boldsymbol{l}_t, \boldsymbol{\nu}_t), \qquad \textit{(Measurement equation)} \tag{5.2}$$

where

$$\tilde{f}(\boldsymbol{l}_t, \boldsymbol{\eta}_t) = \begin{bmatrix} f(\boldsymbol{x}_t, \boldsymbol{\eta}_t) \\ \boldsymbol{\Theta} \end{bmatrix}, \tag{5.3}$$

$$\tilde{h}(\boldsymbol{l}_t, \boldsymbol{\nu}_t) = h(\boldsymbol{x}_t, \boldsymbol{\nu}_t), \tag{5.4}$$

and $\boldsymbol{\Theta}$ is a vector containing the unknown model parameters which in some cases are specified, but in many others are unknown. To complete the formulation, it is assumed that a prior distribution is placed on the initial augmented state vector, say $\boldsymbol{l}_0$. Notice that when the parameters are incorporated into the state vector, we have a nonlinear filtering problem. In the sequel, unless stated otherwise, all filters use the so-called augmented state vector.

Following, general prediction and filtering expressions for the augmented state vector $\boldsymbol{l}_t$ are presented.

5.2.2 Prediction and Filtering Expressions

The corresponding prediction, filtering and recursive-filtering expressions for the augmented state vector are obtained in a similar fashion to equations (2.4), (2.5) and (2.6) in Section 2.2. In other words, the general prediction and filtering expressions for $\boldsymbol{l}_t = (\boldsymbol{x}_t, \boldsymbol{\Theta})'$ are derived as a combined result of the basic assumptions of the state-space formulation and the use of the Bayes Rule.

Predictive PDF

At time $t-1$, assume that the prior PDF $p(\boldsymbol{x}_{t-1}, \boldsymbol{\Theta}_{t-1} | \boldsymbol{y}_{1:t-1})$ at time $t-1$ is available, where $\boldsymbol{x}_t$ is a latent state vector and $\boldsymbol{\Theta}$ is a fixed *unknown* parameter vector. Then, the general predictive expression, the one step-ahead prediction, is given by:

$$p(\boldsymbol{l}_t | \boldsymbol{y}_{1:t-1}) = p(\boldsymbol{x}_t, \boldsymbol{\Theta}_t | \boldsymbol{y}_{1:t-1}) = \int p(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}, \boldsymbol{\Theta}_t)\, p(\boldsymbol{x}_{t-1}, \boldsymbol{\Theta}_t | \boldsymbol{y}_{1:t-1})\, d\boldsymbol{x}_{t-1} \tag{5.5}$$

where $p(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}, \boldsymbol{\Theta}_t)$ is the state evolution density obtained using equation (5.1).

Filtering PDF

Using the Bayes Rule, the knowledge of the augmented state vector can be updated once a new observation $\boldsymbol{y}_t$ becomes available. That is, the filtering PDF is derived as follows:

$$\begin{aligned}
p(\boldsymbol{l}_t|\boldsymbol{y}_{1:t}) = p(\boldsymbol{x}_t,\boldsymbol{\Theta}_t|\boldsymbol{y}_{1:t}) &= \frac{p(\boldsymbol{y}_{1:t}|\boldsymbol{x}_t,\boldsymbol{\Theta}_t)p(\boldsymbol{x}_t,\boldsymbol{\Theta}_t)}{p(\boldsymbol{y}_{1:t})} \\
&= \frac{p(\boldsymbol{y}_t,\boldsymbol{y}_{1:t-1}|\boldsymbol{x}_t,\boldsymbol{\Theta}_t)p(\boldsymbol{x}_t,\boldsymbol{\Theta}_t)}{p(\boldsymbol{y}_t,\boldsymbol{y}_{1:t-1})} \\
&= \frac{p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1},\boldsymbol{x}_t,\boldsymbol{\Theta}_t)p(\boldsymbol{y}_{1:t-1}|\boldsymbol{x}_t,\boldsymbol{\Theta}_t)p(\boldsymbol{x}_t,\boldsymbol{\Theta}_t)}{p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1})p(\boldsymbol{y}_{1:t-1})} \\
&= \frac{p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1},\boldsymbol{x}_t,\boldsymbol{\Theta}_t)p(\boldsymbol{x}_t,\boldsymbol{\Theta}_t|\boldsymbol{y}_{1:t-1})p(\boldsymbol{y}_{1:t-1})p(\boldsymbol{x}_t,\boldsymbol{\Theta}_t)}{p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1})p(\boldsymbol{y}_{1:t-1})p(\boldsymbol{x}_t,\boldsymbol{\Theta}_t)} \\
&= \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t,\boldsymbol{\Theta}_t)p(\boldsymbol{x}_t,\boldsymbol{\Theta}_t|\boldsymbol{y}_{1:t-1})}{p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1})} \\
&\propto p(\boldsymbol{y}_t|\boldsymbol{x}_t,\boldsymbol{\Theta}_t)p(\boldsymbol{x}_t,\boldsymbol{\Theta}_t|\boldsymbol{y}_{1:t-1}). \qquad (5.6)
\end{aligned}$$

where $p(\boldsymbol{y}_t|\boldsymbol{x}_t,\boldsymbol{\Theta}_t)$ is the likelihood of $\boldsymbol{y}_t$ obtained from the measurement evolution density specified in equation (5.2) and $p(\boldsymbol{x}_t,\boldsymbol{\Theta}_t|\boldsymbol{y}_{1:t-1})$ stands for the predictive expression in (5.5). Likewise, the term in the denominator $p\left(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}\right) = \int p(\boldsymbol{y}_t|\boldsymbol{x}_t,\boldsymbol{\Theta}_t)p(\boldsymbol{x}_t,\boldsymbol{\Theta}_t|\boldsymbol{y}_{1:t-1})\,d\boldsymbol{x}_t$ is the normalizing constant which usually is not easy to compute. Additionally, developing (5.6) one step further, one obtains an alternative filtering PDF expression

$$p(\boldsymbol{l}_t|\boldsymbol{y}_{1:t}) = p(\boldsymbol{x}_t,\boldsymbol{\Theta}_t|\boldsymbol{y}_{1:t}) \propto p(\boldsymbol{y}_t|\boldsymbol{x}_t,\boldsymbol{\Theta}_t)p(\boldsymbol{x}_t|\boldsymbol{\Theta}_t,\boldsymbol{y}_{1:t-1})p(\boldsymbol{\Theta}_t|\boldsymbol{y}_{1:t-1}) \qquad (5.7)$$

which explicitly indicates the contribution of the prior PDF of the model parameters vector $p(\boldsymbol{\Theta}_t|\boldsymbol{y}_{1:t-1})$ to update the knowledge about the augmented state vector $\boldsymbol{l}_t$.

It can be verified that in case $\boldsymbol{\Theta}$ is known, equation (5.7) is reduced to the expression in the numerator of (2.5) since the entry $p(\boldsymbol{\Theta}_t|\boldsymbol{y}_{1:t-1})$ degenerates and the known model parameters can be dropped from the conditioning statements.

It is well known that the solution to the optimal filtering problem, stated and fully described in Chapter 2, is to obtain the posterior PDF $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$. In a similar manner, all the relevant information about the general model described in equations (5.1) and (5.2) is embodied in the augmented-state vector $\boldsymbol{l}_t = (\boldsymbol{x}_t,\boldsymbol{\Theta}_t)$. This implies that any specific characteristic of one of the marginals, the state or parameter variables, can be easily obtained once $p(\boldsymbol{l}_t|\boldsymbol{y}_{1:t}) = p(\boldsymbol{x}_t,\boldsymbol{\Theta}_t|\boldsymbol{y}_{1:t})$ is available in an exact or approximative manner. For simplicity, in the remainder of this chapter we shall refer to the state estimation or simultaneous estimation of state and model parameters by just the word filtering. In subsequent chapters, it will be clear from the context to which case one refers to.

In the following section, we present a description of the chosen PF variants adopted for parameter estimation. Therein, a review of two main historical approaches used to adapt the particle filtering methodology for the simultaneous estimation of the original state and the fixed model parameters is also provided.

Parameter Estimation via Particle Filtering

For the simultaneous estimation of states and model parameters via particle filtering, one assumes that at *fixed* time t, the filtering PDF $p(\boldsymbol{x}_t, \boldsymbol{\Theta}_t|\boldsymbol{y}_{1:t})$ is approximated by a sufficiently large set of 'particles' $\{(\boldsymbol{x}_t^{(1)}, \boldsymbol{\Theta}_t^{(1)}), \ldots, (\boldsymbol{x}_t^{(M)}, \boldsymbol{\Theta}_t^{(M)})\}$ with discrete probability masses of $\tilde{\boldsymbol{\omega}}_t^{(1)}, \ldots, \tilde{\boldsymbol{\omega}}_t^{(M)}$. Let us recall that the particles $\{(\boldsymbol{x}_t^{(j)}, \boldsymbol{\Theta}_t^{(j)})\}_{j=1}^M$ are obtained from an alternative proposal PDF, $q(\boldsymbol{x}_t, \boldsymbol{\Theta}_t|\boldsymbol{y}_{1:t})$, which is easier to sample from, but very similar to the target filtering PDF $p(\boldsymbol{x}_t, \boldsymbol{\Theta}_t)|\boldsymbol{y}_{1:t}$. Moreover, note that if the model parameters are assumed fixed, the t suffix on $\boldsymbol{\Theta}_t^j$ particles only indicates that they are from the time t posterior, not that $\boldsymbol{\Theta}_t^j$ is time-varying.

Hence, under the particle filtering methodology, the theoretic predictive and filtering expressions given by equations (5.5) and (5.6) are approximated by corresponding 'empirical densities'. That is, at time $t-1$, assume that a sample $\{(\boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{\Theta}_{t-1}^{(j)})\}_{j=1}^M$ from the prior $p(\boldsymbol{l}_{t-1}|\boldsymbol{y}_{1:t-1})$ is available. Additionally, assume a fixed and unknown parameter vector $\boldsymbol{\Theta}$. Then, the predictive PDF approximation to $p(\boldsymbol{l}_t|\boldsymbol{y}_{1:t-1})$ in equation (5.5) is given by the following expression:

Predictive PDF approximation:

$$p(\boldsymbol{l}_t|\boldsymbol{y}_{1:t-1}) = p(x_t, \boldsymbol{\Theta}|\boldsymbol{y}_{1:t-1}) \approx \sum_{j=1}^{M} p(x_t|x_{t-1}^{(j)}, \boldsymbol{\Theta}^{(j)})\, \tilde{w}_{t-1}^{(j)}, \tag{5.8}$$

where $\tilde{w}_{t-1}^{(j)}$ are the normalized importance weights at previous time $t-1$. Combining both, the likelihood $p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\Theta}_t)$ and the previous predictive PDF approximation, an expression for the approximation to the filtering PDF $p(\boldsymbol{l}_t|\boldsymbol{y}_{1:t})$ in equation (5.6) is obtained as

Filtering PDF approximation:

$$p(\boldsymbol{l}_t|\boldsymbol{y}_{1:t}) = p(x_t, \boldsymbol{\Theta}|\boldsymbol{y}_{1:t}) \approx p(y_t|x_t, \boldsymbol{\Theta}) \sum_{j=1}^{M} p(x_t|x_{t-1}^{(j)}, \boldsymbol{\Theta}^{(j)})\, \tilde{w}_{t-1}^{(j)} \tag{5.9}$$

Hence, the filtering PDF of the augmented state vector $\boldsymbol{l}_t$ is obtained by and is approximated (via particle filtering) by means of an empirical distribution

$$p(\boldsymbol{l}_t|\boldsymbol{y}_{1:t}) = p(\boldsymbol{x}_t, \boldsymbol{\Theta}|\boldsymbol{y}_{1:t}) \approx \sum_{j=1}^{M} \tilde{w}_t^{(j)} \delta(\boldsymbol{l}_t - \boldsymbol{l}_t^{(j)}),$$

where $\sum_{j=1}^M \tilde{w}_t^{(j)} = 1$, $\tilde{w}_t^{(j)} = \frac{w_t^{(j)}}{\sum_{j=1}^M w_t^{(j)}}$ and

$$w_t^{(j)} \propto \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t^{(j)}, \boldsymbol{\Theta}^{(j)})\, p(\boldsymbol{x}_t^{(j)}|\boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{\Theta}^{(j)})}{q(\boldsymbol{x}_t^{(j)}|\boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{\Theta}^{(j)}, \boldsymbol{y}_t)} \cdot \tilde{w}_{t-1}^{(j)}. \tag{5.10}$$

Recall that according to theory, an optimal proposal PDF takes into account the information provided by the last observation, and that in practice however, it is very common to choose the transition prior

$$q(x_t|x_{t-1}^{(j)}, \boldsymbol{\Theta}_t^{(j)}, y_t) = p(x_t|x_{t-1}^{(j)}, \boldsymbol{\Theta}_t^{(j)}) \tag{5.11}$$

as a proposal. In that case the expression for the importance weights given in (5.10) reduces to the simpler expression

$$w_t^{(j)} \propto p(\boldsymbol{y}_t|\boldsymbol{x}_t^{(j)}, \boldsymbol{\Theta}^{(j)}) \cdot \tilde{w}_{t-1}^{(j)}. \tag{5.12}$$

Further, if resampling is made at every time step, the weights have the simplest form

$$w_t^{(j)} \propto p(\boldsymbol{y}_t|\boldsymbol{x}_t^{(j)}, \boldsymbol{\Theta}^{(j)}). \tag{5.13}$$

As seen, the adoption of the particle filtering methodology to estimate simultaneously the original state and the unknown parameter vector gives rise to different particle filter variants for parameter estimation. Kitagawa (1998) introduces the self organizing filter, denoted in this work by SO particle filter, which uses and extends the basic SIR particle filter algorithm in Chapter 2 in order to filter the augmented state vector $(\boldsymbol{x}_t^{(j)}, \boldsymbol{\Theta}^{(j)})$. Following, a brief and concise description of the SO particle filter variant as well as the corresponding pseudocode is presented. Therein, the estimation of a simple but illustrative dynamic model is provided.

5.3 The Self Organizing Particle Filter

Both Gordon et al (1993) and Kitagawa (1996) use their respective SIR particle filter variant to estimate solely time–varying states; they assume all fixed model parameters to be known. Recall that the SIR particle filter variant is fully described in Chapter 2 and subject to Monte Carlo experiments in Chapter 4. Kitagawa (1998) departs from the augmented state vector and uses his SIR particle filter variant to perform the simultaneous estimation of state and unknown fixed model parameters.

Thus, the main feature of the SO particle filter variant is to apply the SIR particle filter approach to the augmented state vector by using the transition prior given by equation (5.11) as a proposal PDF and by resampling at every time step. Thus, modifying Kitagawa's (1996) SIR particle filter to include the parameters gives rise to the pseudocode for the self-organizing particle filter variant presented in Algorithm 11 (on page 141). Notice that Kitagawa uses stratified resampling.

We used the Algorithm 11 in an attempt to estimate the parameters of an AR(1) plus noise process. Following, we present illustrative results concerning the estimation of the autoregressive parameter ϕ.

An illustrative example: AR(1) plus Noise Model

A time series of length $T = 1000$ is generated according to the AR(1) plus noise process, with state-space formulation

$$\begin{aligned} x_t &= \phi x_{t-1} + \sigma_\eta \eta_t \\ y_t &= x_t + \sigma_\nu \nu_t \end{aligned} \tag{5.14}$$

where both η_t and ν_t follow a standard normal distribution and σ_η and σ_ν are scale parameters. In this particular case, the autoregressive parameter ϕ is assumed to be fixed and unknown. On the other

Algorithm 11 Self Organizing Particle Filter (SO PF)

Initialization $t = 0$

for $j = 1$ to M **do**

Sample $x_0^{(j)} \sim p(x_0)$,

Sample $\boldsymbol{\Theta}_0^{(j)} \sim p(\boldsymbol{\Theta}_0)$

end for

for $t = 1$ to N **do**

Step 1 Importance sampling step

for $j = 1$ to M **do**

Prediction: Sample $x_t^{(j)} \sim q(x_t|x_{t-1}^{(j)}, \boldsymbol{\Theta}_{t-1}^{(j)}, y_t) = p(x_t^{(j)}|x_{t-1}^{(j)}, \boldsymbol{\Theta}_{t-1}^{(j)})$.

In this case we need to

– generate a random number $\eta_t^{(j)}$ according to the noise density associated to the state in equation (5.1),

– calculate $\boldsymbol{l}_t = \tilde{f}(\boldsymbol{l}_{t-1}, \boldsymbol{\eta}_t)$ using (5.1) and (5.3).

Filtering: Assign to each combined particle $\boldsymbol{l}_t^{(j)} = (x_t^{(j)}, \boldsymbol{\Theta}_t^{(j)})$ the weight $w_t^{(j)}$ according to (5.13)

Normalize the importance weights:

$$\tilde{w}_t^{(j)} = \frac{w_t^{(j)}}{\sum_{i=1}^{M} w_t^{(i)}}$$

end for

Step 2 Resampling step (Stratified)

Resample with replacement the particles $\left\{\boldsymbol{l}_t^{(j)}\right\}_{j=1}^{M} = \left\{(x_t^{(1)}, \boldsymbol{\Theta}_t^{(1)}), \ldots, (x_t^{(M)}, \boldsymbol{\Theta}_t^{(M)})\right\}$ according to importance weights $\left\{\tilde{w}_t^{(1)}, \ldots, \tilde{w}_t^{(M)}\right\}$.

end for

hand, the scale parameters are assumed to be known. The true model parameters values are $\phi = 0.8$ and $\sigma_\eta = \sigma_v = 1$.

The corresponding state-space model formulation for the augmented-state is thus defined by the vector

$$\boldsymbol{l}_t = \begin{bmatrix} \boldsymbol{x}_t \\ \Theta_t \end{bmatrix} = \begin{bmatrix} x_t \\ \phi \end{bmatrix},$$

where in this particular case, $\Theta_t = \Theta = \phi$ is the unknown fixed autoregressive parameter to be estimated.

The SO particle filter variant with $N_p = 20000$ particles is applied to the linear and gaussian model specified in (5.14). Figure 5.1 shows the posterior distribution of the autoregressive parameter ϕ of the contaminated AR(1) process. The true value of the autoregressive parameter ϕ is indicated by a thick black vertical line; in this case $\phi = 0.8$

Our experimental results show that when one applies the SO particle filter, the estimated parameter is close to the true one, but the particles do not adequately regenerate. For this particular model, we

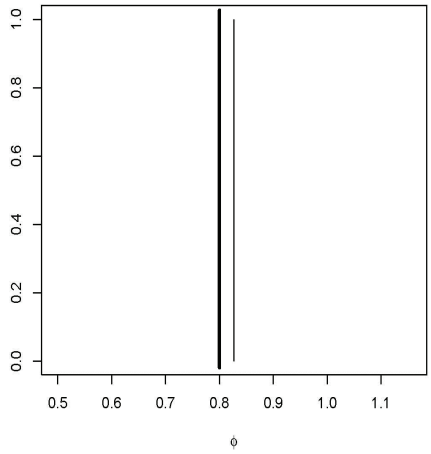


Figure 5.1: SO: Posterior distribution for parameter ϕ of the AR(1) plus noise model specified in equation (5.14). In this case, $T = 1000, N_p = 20000$, $\sigma_\eta = \sigma_\nu = 1$ and $\phi = 0.8$.

even observe that the particles collapse to a unique distinct value despite the large number of particles used, which results in an inadequate posterior distribution.

The SO particle filter variant is revisited in Kitagawa and Sato (2001) where Kitagawa points out that the self-organizing filter originally introduced in Kitagawa (1998) was not really able to cope with the *fixed* parameter estimation problem unless an artificial noise is added to the parameters evolution model.

We remark that this potential drawback is not exclusive to the SO particle filter; the same problem would occur if one applies the ASIR particle filter variant, described in Chapter 2, to the above model. Thus, in that case, the addition of an artificial noise to the model parameters is also a target.

All this, of course, is in line with the idea presented in Chapter 2, which stated that the introduction of a resampling step is crucial but could lead to a potential drawback called sample impoverishment[3]. Indeed, this problem becomes more acute when dealing with the simultaneous estimation of the original state vector and fixed model parameters, because in this situation the particles associated with the parameters do not adequately regenerate and may end up stacked in a small and possibly wrong subregion of the parameters state-space support.

The next section describes two main historical approaches used to specify the artificial noise addition for model parameters.

[3]The particles collapse to a few or in some cases to a unique distinct value, also known as attrition problem.

5.4 Parameters Artificial Evolution

The problem herein consists of how to define an artificial evolution of the fixed model parameters. Gordon et al. (1993) proposed to add an additional random noise, also-called *roughening penalties*, to sampled state vectors in an attempt to deal with the inherent particle filters degeneracy drawback. This idea of adding a random noise, in the sequel referred as artificial evolution approach, is also applied to specify an artificial evolution model for fixed and unknown parameters.

5.4.1 Parameter Vector Evolution Step via the Artificial Evolution Approach

Kitagawa and Sato (2001) suggest to modify the SO particle filter variant by extrapolating to fixed model parameters the artificial evolution approach. That is, to tackle the sample impoverishment problem, an artificial noise is added to the fixed and unknown parameters aiming to produce a diversification of the model parameters particles. In this manner, the parameter vector artificial evolution, assumed to vary slowly in time, would be described by

$$\begin{aligned} \boldsymbol{\Theta}_t &= \boldsymbol{\Theta}_{t-1} + \boldsymbol{\eta}_t, \\ \boldsymbol{\eta}_t &\sim \mathcal{N}(0, W_t), \end{aligned} \tag{5.15}$$

where $\boldsymbol{\eta}_t$ represents the independent, zero-mean normal increment added to the parameters for some specified variance matrix W_t. It is also assumed that $\boldsymbol{\Theta}_{t-1}$ and $\boldsymbol{\eta}_t$ are conditionally independent given $\boldsymbol{y}_{1:t-1}$. The addition of the specified artificial noise step after the resampling step of Algorithm 11 gives rise to an improved version of the original SO particle filter variant introduced by Kitagawa (1998). As can be seen in Kitagawa et al. (2002), the original SO particle filter works adequately if the model parameters are truly time–varying.

We remark that although the addition of this artificial noise, $\boldsymbol{\eta}_t$, certainly prevents the parameters particles to collapse to a few unique or even to a single particle, it remains not very clear how this can be undertaken without significantly changing the model at hand. Questions arise like how small the variances of the added artificial noise must be or whether they should also be estimated as part of the already augmented state vector. Moreover, the addition of any extra parameters into the estimation problem would necessarily imply extra execution time and memory requirements; one would have to decide then if the extra time is compensated by the quality of the estimation results. In practice, how this is entertained will depend on factors like the model at hand, the researcher's decision and, of course, on its expertise. Certainly, a fact about the artificial evolution method is what, for instance, Liu and West (2001) address:

> "This neat, ad-hoc idea is easily implementable, but suffers the obvious drawback that it "throws away" information about parameters in assuming them to be time-varying when they are, in fact, fixed. The same drawback arises in using the idea in its original form for dynamic states."

Summing up, all previous remarks highlight the need to encounter an alternative approach to tackle the sample impoverishment drawback. However, this alternative approach should be able to avoid the loss of information problem occurred over time when using the aforementioned artificial evolution approach. A possible and already identified solution to the sample impoverishment drawback consists of regenerating fixed model parameters particles by diversifying the *old* ones appealing to modified kernel density estimation methods.

5.4.2 Parameter Vector Evolution Step via the Jittering Approach

Liu and West (2001) review both the kernel and artificial evolution methods in West (1993) and Gordon et al. (1993) respectively, finding inherent structural similarities between them. This implies that through the combined use of artificial evolution ideas with kernel smoothing ideas via shrinkage, the model parameters can be adequately *jittered*, meaning that the parameters particles are diversified but without the loss of information implied by solely using the artificial evolution approach. Thus, as a result of including a *jittering* step, the sample impoverishment problem and the possible prior-to-data conflict which might be present can be tackled. In this case, the *jittering* step is specified by

$$\boldsymbol{\Theta}_t = \boldsymbol{\Theta}_{t-1} + \boldsymbol{\eta}_t \tag{5.16}$$

$$\boldsymbol{\eta}_t \sim \mathcal{N}(0, W_t)$$

$$W_t = V_{t-1}\big(\frac{1}{\delta} - 1\big). \tag{5.17}$$

Notice that the above formulation differs from expression (5.15) in the last entry (5.17) which explicitly specifies how to compute the variance matrix of the added model parameters random disturbance. Herein and in the sequel, V_{t-1} denotes the variance matrix of the marginal PDF $p(\boldsymbol{\Theta}_{t-1}|\boldsymbol{y}_{1:t-1})$ and δ is a discount factor usually taken between 0.95 and 0.995. Additionally, the finite mean of $p(\boldsymbol{\Theta}_{t-1}|\boldsymbol{y}_{1:t-1})$ is denoted by $\overline{\boldsymbol{\Theta}}_{t-1}$.

In the sequel, a concise motivating discussion about using the artificial evolution or the jittering approach in order to add an artificial noise to fixed model parameters particles is presented. This aims to clarify how both approaches are related and how they differ.

5.4.3 Artificial Evolution vs Jittering for Artificial Noise Addition

To begin with, departing from expression (5.15), one obtains the general form for the variance matrix of the prior PDF $p(\boldsymbol{\Theta}_t|\boldsymbol{y}_{1:t-1})$ as

$$\text{Var}(\boldsymbol{\Theta}_t|\boldsymbol{y}_{1:t-1}) = V_{t-1} + W_t + 2\,\text{Cov}(\boldsymbol{\Theta}_{t-1}, \eta_t|\boldsymbol{y}_{1:t-1}). \tag{5.18}$$

Since the artificial evolution approach specified in (5.15) assumes that $\text{Cov}(\boldsymbol{\Theta}_{t-1}, \eta_t|\boldsymbol{y}_{1:t-1}) = 0$, the last expression is simplified to

$$\text{Var}(\boldsymbol{\Theta}_t|\boldsymbol{y}_{1:t-1}) = V_{t-1} + W_t. \tag{5.19}$$

where W_t accounts for the loss of information inherent under the artificial evolution method. Suppose that at fixed time $t-1$, a sufficiently large set of weighted model parameter particles from $p(\boldsymbol{\Theta}_{t-1}|\boldsymbol{y}_{1:t-1})$ is given by $\left\{\boldsymbol{\Theta}_{t-1}^{(j)}, \widetilde{w}_{t-1}^{(j)}\right\}_{j=1}^{M}$. Then, the artificial evolution approach, described in equation (5.15), implicitly assumes that the approximated PDF $p(\boldsymbol{\Theta}_t|\boldsymbol{y}_{1:t-1})$ has a kernel form specified by

$$p(\boldsymbol{\Theta_t}|\boldsymbol{y}_{1:t-1}) \approx \sum_{j=1}^{M} \widetilde{w}_{t-1}^{(j)} \mathcal{N}(\boldsymbol{\Theta_t}; \boldsymbol{\Theta_{t-1}}^{(j)}, W_t), \tag{5.20}$$

which is defined as a mixture of normal distributions weighted by the sample weights $\widetilde{w}_{t-1}^{(j)}$. Notice that this approximation is over–dispersed relative to the target variance V_{t-1}, with over-dispersion quantified by W_t. Notice also that the normal kernels are located around existing sample values, which is a typical situation when dealing with conventional kernel density methods. From (5.15) and (5.20), one can notice that there exists a close tie between the artificial evolution approach and the kernel smoothing methodology. Moreover, since the estimation procedure is performed sequentially in time, the over-dispersion problem persists over time and thus the loss of information builds up. Therefore, we need a method that is able to add an artificial noise to the fixed model parameters but capable to overcome the historical loss of information drawback. This can be achieved by the jittering approach, which consists in modifying the artificial evolution approach so that the following condition holds:

$$\text{Var}(\boldsymbol{\Theta}_t|\boldsymbol{y}_{1:t-1}) = \text{Var}(\boldsymbol{\Theta}_{t-1}|\boldsymbol{y}_{1:t-1}) = V_{t-1}. \tag{5.21}$$

This condition can be fulfilled by introducing correlations between $\boldsymbol{\Theta}_{t-1}$ and the random noise η_t, which obviously implies the existence of a non-zero covariance matrix $\text{Cov}(\boldsymbol{\Theta}_{t-1}, \eta_t|\boldsymbol{y}_{1:t-1})$. From the general variance matrix expression in equation (5.18), taking into account the condition assumed in equation (5.21), the following expression for the covariance matrix is obtained:

$$\text{Cov}(\boldsymbol{\Theta}_{t-1}, \eta_t|\boldsymbol{y}_{1:t-1}) = -W_t/2. \tag{5.22}$$

This means that a structure of negative correlations must be introduced in order to correct the historical loss of information drawback. Further, if it is assumed that $(\boldsymbol{\Theta}_{t-1}, \eta_t|\boldsymbol{y}_{1:t-1})$ has an approximate jointly normal distribution, then the conditional evolution for the model parameters will be normal and specified by

$$p(\boldsymbol{\Theta}_t|\boldsymbol{\Theta}_{t-1}) \sim \mathcal{N}(\boldsymbol{\Theta}_t; A_t\boldsymbol{\Theta}_{t-1} + (I - A_t)\overline{\boldsymbol{\Theta}}_{t-1}, (I - A_t^2)V_{t-1}), \tag{5.23}$$

where $A_t = (I - W_t V_{t-1}^{-1})/2$ is a shrinkage matrix, being I an identity matrix.

The above discussion, departing from West (1993) and Liu and West (2001), motivates the relationship between the artificial evolution approach and kernel smoothing methods via shrinkage. It also implies that the Monte Carlo approximation to $p(\boldsymbol{\Theta}_t|\boldsymbol{y}_{1:t-1})$ has a generalized kernel form with complicated shrinkage patterns as seen in equation (5.23).

Similarly to Liu and West (2001), in this work we restrict our attention to the special case when the specification of the variance W_t is the result of assuming a shrinkage matrix $A_t = aI$ with $a = (3\delta - 1)/2\delta$, and a discount factor δ in $(0,1]$; as stated by these authors the usually taken values for

the discount parameter δ are around 0.95-0.99. That is, W_t has the form given by equation (5.17). Moreover, in that particular case, equation (5.23) is simplified to

$$p(\boldsymbol{\Theta}_t|\boldsymbol{\Theta}_{t-1}) \sim \mathcal{N}(\boldsymbol{\Theta}_t; m_{t-1}, h^2 V_{t-1}) \tag{5.24}$$

$m_{t-1} = a\boldsymbol{\Theta}_{t-1} + (1-a)\overline{\boldsymbol{\Theta}}_{t-1}$ denotes the kernel location parameter. The kernel variance is $h^2 V_{t-1}$, being $h > 0$ a controlling smoothing parameter explicitly defined in terms of a discount factor δ. From the following expressions, the relationship between the smoothing parameter h and the discount factor δ can be obtained. Notice that $h^2 = 1 - a^2$, consequently $a = \sqrt{(}1 - h^2)$ and as previously stated $a = (3\delta - 1)/2\delta$. Therefore, according to the novel shrinkage idea introduced originally by West (1993) and later revisited by Liu and West (2001), the particles $\boldsymbol{\Theta}_{t-1}^{(j)}$ are first pushed towards their sample mean $\overline{\boldsymbol{\Theta}}_{t-1}$ before a small degree of noise is added to them.

The use of the jittering approach in the ASIR particle filter context gives rise to a novel PF variant. Following, a description of this filter as well as the corresponding pseudocode is presented.

5.5 The Liu and West Particle Filter

As known, Pitt and Shephard (1999) introduce the ASIR particle filter variant in order to achieve an optimal state estimation, but they assume all model parameters to be known. Liu and West (2001) extend the ASIR particle filter variant to obtain a combined estimation of time varying states and unknown fixed model parameters. These authors combine old ideas of kernel smoothing via shrinkage for modelling fixed model parameters with newer ideas of auxiliary particle filtering for the dynamic states; we denote this new algorithm by LW particle filter. These authors also claim that the computational cost under the LW particle filter variant is meaningfully reduced from earlier kernel smoothing algorithms used for Bayesian posterior simulation.

Following, the most important features under the LW particle filter are presented:

Assume that, at time $t-1$, $\boldsymbol{\Theta}_{t-1}^{(j)}$ are the model parameters particles from $p(\boldsymbol{\Theta}_{t-1}|\boldsymbol{y}_{1:t-1})$ and that proper kernel location parameters can be defined and computed by

$$m_{t-1}^{(j)} = a\boldsymbol{\Theta}_{t-1}^{(j)} + (1-a)\overline{\boldsymbol{\Theta}}_{t-1}. \tag{5.25}$$

This would imply that the corresponding kernel form approximation to $p(\boldsymbol{\Theta}_t|\boldsymbol{y}_{1:t-1})$ is given by

$$p(\boldsymbol{\Theta_t}|\boldsymbol{y}_{1:t-1}) \approx \sum_{j=1}^{M} \widetilde{w}_{t-1}^{(j)} \mathcal{N}(\boldsymbol{\Theta_t}; m_{t-1}^{(j)}, h^2\, V_{t-1}). \tag{5.26}$$

Substituting (5.26) in the alternative filtering PDF in equation (5.7), a final expression for filtering si-

multaneously the original state vector and the fixed unknown parameters is obtained as follows:

$$p(\boldsymbol{l}_t|\boldsymbol{y}_{1:t}) = p(\boldsymbol{x}_t, \boldsymbol{\Theta}_t|\boldsymbol{y}_{1:t}) \tag{5.27}$$

$$= p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\Theta}_t) p(x_t, \boldsymbol{\Theta}|\boldsymbol{y}_{1:t-1})$$

$$\propto p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\Theta}_t) p(\boldsymbol{x}_t|\boldsymbol{\Theta}_t, \boldsymbol{y}_{1:t-1}) p(\boldsymbol{\Theta}_t|\boldsymbol{y}_{1:t-1}) \tag{5.28}$$

$$\approx p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\Theta}_t) p(\boldsymbol{x}_t|\boldsymbol{\Theta}_t, \boldsymbol{y}_{1:t-1}) \sum_{j=1}^{M} \tilde{w}_{t-1}^{(j)} \mathcal{N}(\boldsymbol{\Theta}_t; m_{t-1}^{(j)}, h^2 V_{t-1}). \tag{5.29}$$

From the above, it becomes clear that the jittering ideas are used in order to model the fixed parameters, and that any filtering method could be used to model the dynamic states; Liu and West (2001) resort to the ASIR particle filter variant. This PF variant is summarized in the pseudocode given in Algorithm 12 (page 148).

As Liu and West (2001) point out, at time t one has a combined sample $\{(x_t^j, \boldsymbol{\Theta}^j),\ j = 1, \ldots, M\}$ representing an importance sampling approximation to the time t posterior $p(\boldsymbol{x}_t, \boldsymbol{\Theta}_t|\boldsymbol{y}_{1:t})$ for both parameter and state. These authors also state that the final resampling step is optional.

The LW particle filter variant with $N_p = 20000$ is applied to the model specified in (5.14). Figure 5.2 shows the posterior distribution of the autoregressive parameter ϕ of the contaminated AR(1) process. The true value of the autoregressive parameter ϕ is indicated by a thick black vertical line; in this case $\phi = 0.8$

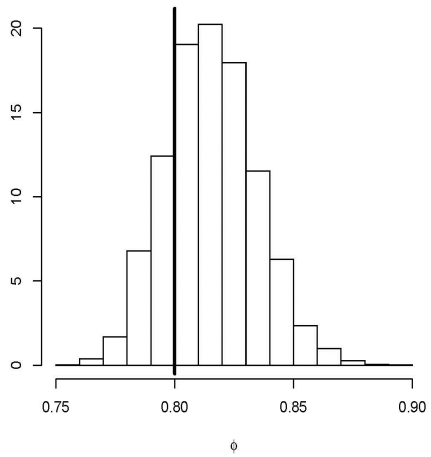


Figure 5.2: LW: Posterior distribution for parameter ϕ of the AR(1) plus noise model specified in equation (5.14). In this case, $T = 1000, N_p = 20000, \sigma_\eta = \sigma_v = 1$ and $\phi = 0.8$.

Notice that the LW particle filter variant has two main features: First, it applies the jittering approach at every time step in order to smooth and regenerate the fixed model parameters particles without the loss of the historical information problem inherent to the aforementioned artificial evolution approach. Second, it uses the ASIR particle filter variant to model the time-varying states.

Algorithm 12 Liu and West Particle Filter (LW PF)

Initialization $t = 0$

for $k = 1$ to M **do**

Sample $\boldsymbol{x}_0^{(k)} \sim p(\boldsymbol{x}_0)$

Sample $\boldsymbol{\Theta}_0^{(k)} \sim p(\boldsymbol{\Theta}_0)$

end for

Choose discount factor δ in $(0,1]$

Compute the tuning parameter $a = (3\delta - 1)/2\delta$ based on the chosen discount factor δ.

Compute the controlling smoothing parameter $h > 0$: $h^2 = 1 - a^2$

for $t = 1$ to N **do**

Step 1 Get prior point estimates of $(\boldsymbol{x}_t, \boldsymbol{\Theta}_t)$, given by $(\mu_t^{(k)}, m_{t-1}^{(k)})$

for $k = 1$ to M **do**

Calculate $\mu_t^{(k)}$ associated to the conditional PDF of $(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}^{(k)}, \Theta_{t-1}^{(k)})$; in this case $\mu_t^{(k)} = \mathrm{E}(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}^{(k)}, \Theta_{t-1}^{(k)})$

Calculate the mean $\bar{\boldsymbol{\Theta}}_{t-1}$ and the variance V_{t-1} of the Monte Carlo approximation to $p(\boldsymbol{\Theta}_t | \boldsymbol{y}_{1:t-1})$

Calculate the kth kernel location $m_{t-1}^{(k)} = a\boldsymbol{\Theta}_{t-1}^{(k)} + (1-a)\bar{\boldsymbol{\Theta}}_{t-1}$ (from equation (5.25))

end for

Step 2 First Resampling step: Auxiliary variable resampling

for $k = 1$ to M **do**

Calculate the first stage weights $\boldsymbol{\lambda}_t^{(k)} = q(k | \boldsymbol{y}_{1:t}) \propto \boldsymbol{\omega}_{t-1}^k p(\boldsymbol{y}_t | \mu_t^k, m_{t-1}^{(k)})$

end for

for $k = 1$ to M **do**

Normalize the first stage weights $\tilde{\boldsymbol{\lambda}}_t^{(k)} = \frac{\boldsymbol{\lambda}_t^{(k)}}{\sum_{i=1}^M \boldsymbol{\lambda}_t^{(k)}}$

end for

Sample with replacement the index $k_{j=1}^M$ according to the computed first stage weights.

Step 3 Jittering step

for $j = 1$ to M **do**

Sample a new parameter vector: $\boldsymbol{\Theta}_t^{k(j)} \sim \mathcal{N}(\bullet | m_{t-1}^{k(j)}, h^2 V_{t-1})$,

end for

Step 4 Importance sampling step

for $j = 1$ to M **do**

Sample $\boldsymbol{x}_t^{k(j)} \sim q(\boldsymbol{x}_t | k^{(j)}, \boldsymbol{y}_{1:t}) = p(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}^{k^{(j)}}, \Theta_t^{k(j)})$ as in the SIR filter; that is particles are sampled from the transition equation.

Calculate the second stage weights $\boldsymbol{\omega}_t^{(j)}$ as: $\boldsymbol{\omega}_t^{(j)} \propto \frac{p(\boldsymbol{y}_t | \boldsymbol{x}_t^{k^{(j)}}, \Theta_t^{k(j)})}{p(\boldsymbol{y}_t | \mu_t^{k^{(j)}}, m_{t-1}^{k(j)})}$

end for

for $k = 1$ to M **do**

Normalize the second stage weights $\tilde{\boldsymbol{\omega}}_t^{(k)} = \frac{\boldsymbol{\omega}_t^{(k)}}{\sum_{k=1}^M \boldsymbol{\omega}_t^{(k)}}$

end for

Up to this point we would have a final posterior approximation $(\boldsymbol{x}_t^{k(j)}, \boldsymbol{\Theta}_t^{k(j)})$ with weights $\tilde{\boldsymbol{\omega}}_t^{(k)}$

Step 5 Second Resampling step (OPTIONAL): when equally weighted sample is required

Resample with replacement the particles $(\boldsymbol{x}_t^{k(j)}, \boldsymbol{\Theta}_t^{k(j)})$, $j = 1, \ldots, M$ with importance weights $\left\{\tilde{\boldsymbol{\omega}}_t^{(1)}, \ldots, \tilde{\boldsymbol{\omega}}_t^{(M)}\right\}$

end for

5.6 The Sampling Importance Resampling plus Jittering Particle Filter Variant

Herein, we propose a particle filter variant that combines the SIR particle filter to model the dynamic states and the jittering approach used by Liu and West (2001) to model the fixed parameters.

5.6.1 Justification/Motivating Remarks

It is already known that Kitagawa (1998) uses the particle filtering approach, specifically the self organizing PF variant, to simultaneously estimate the state-variables and some unknown time-varying model parameters. The main feature of this filter is to append the unknown time-varying model parameters to the original state vector, and then just apply the SIR particle filter variant to such augmented state vector. Notice that in case the parameters evolve dynamically in time, the simultaneous estimation of states and parameters is in practice reduced to a states (augmented one) filtering problem; thus, theoretically, any Chapter 2 filter could be applied.

In a later publication, Kitagawa and Sato (2001), the authors point out that in its original form, the self-organizing PF is not able to cope with the fixed parameter estimation problem, unless an artificial noise is added to them. In that publication these authors suggest to use the described artificial evolution approach as done beforehand by Gordon, Salmond, and Smith (1993) in the context of state estimation.

The usual criticism to this approach is that one is artificially changing the model at hand, when in reality the model parameters are fixed, not time-varying. Additionally one is faced with a new dilemma: how to choose the variances of the added artificial noises or to be able to also estimate them, increasing the size of the augmented state vector. Further, if this is done via the artificial evolution approach, the problem of historical loss of information arises.

As aforementioned, our incursion into the simulation-based methodology called particle filtering began with the papers of Kitagawa (1996) and Kitagawa (1998). We adopted Kitagawa's (1998) SO PF variant in an attempt to estimate the parameters of an autoregressive process, which state-space model representation had no measurement noise. Our simulation results show that the estimated parameters converged to the true parameters values, but the particles degenerate and even collapse to a few distinct ones (Acosta et al. 2003). At that moment, to find a method which is able to regenerate the sample paths became a target. Initially, we tried –rendering mixed results– to diversify the fixed model parameters particles by adding an artificial noise as suggested by Kitagawa and Sato (2001); that is, by fixing the variances to very small values. However, the jittering approach of Liu and West (2001) to define an artificial dynamic model for the fixed model parameters proved to be a better solution to our problem; we then named this particle filter variant SIRJ and it has applied in the framework of stochastic first order autoregressive volatility models and also in threshold volatilty models; see Acosta, Martí-Recober, and Muñoz (2004), Muñoz, Márquez, Martí-Recober, Villazón, and Acosta (2004) and Muñoz et al. (2007), respectively.

5.6.2 Some Details of the Sampling Importance Resampling plus Jittering Particle Filter Variant

In few words, the sampling importance resampling plus jittering (SIRJ) particle filter consists in modifying the SO particle filter by adding the jittering step suggested by Liu and West (2001). That is, this algorithm uses the particle filtering methodology, but also extends upon it by creating an artificial noise for the unknown fixed model parameters via the jittering approach described in Section 5.4. Specifically, by using a modified kernel smoothing method via shrinkage, the fixed parameters particles are jittered so that the problem of the historical loss of information, still present when using the previously suggested artificial evolution approach, is overcome.

Thus, under the SIRJ filter, the particle filtering methodology is used for filtering the dynamic states, whereas the jittering approach based on kernel smoothing via shrinkage is used for modeling and estimating the fixed model parameters, $\boldsymbol{\Theta}$. Specifically, the SO particle filter is combined with the jittering step to ensure that the particles "move" adequately and thus cover the whole parameters space support. Remind that the jittering step avoids the loss of information that is present in previously suggested versions of the artificial evolution approach.

Next, in Algorithm 13 (page 151), we present a pseudocode of the SIRJ particle filter variant.

The SIRJ particle filter variant with $N_p = 20000$ is applied to the model specified in (5.14). Figure 5.3 shows the posterior distribution of the autoregressive parameter ϕ of the contaminated AR(1) process. The true value of the autoregressive parameter ϕ is indicated by a thick black vertical line; in this case $\phi = 0.8$

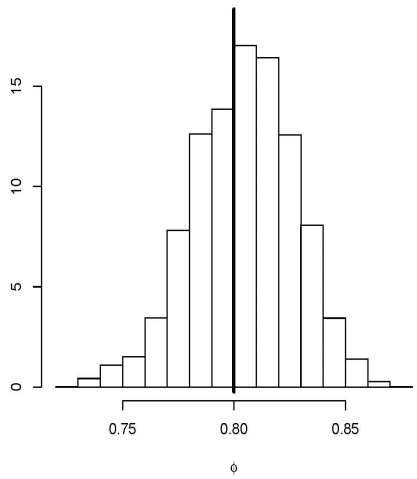


Figure 5.3: SIRJ: Posterior distribution for parameter ϕ of the AR(1) plus noise model specified in equation (5.14). In this case, $T = 1000, N_p = 20000, \sigma_\eta = \sigma_\nu = 1$ and $\phi = 0.8$.

In the following section we also explore how the jittering step performs under the context of the EPF and UPF variants and parameters estimation.

Algorithm 13 Sampling Importance Resampling plus Jittering Particle Filter (SIRJ PF)

Initialization $t = 0$
  for $j = 1$ to M **do**
    Sample $\boldsymbol{x}_0^{(j)} \sim p(\boldsymbol{x}_0)$
    Sample $\boldsymbol{\Theta}_0^{(j)} \sim p(\boldsymbol{\Theta}_0)$
  end for
  for $t = 1$ to N **do**
Step 1 Importance sampling step
    for $j = 1$ to M **do**
      Prediction Sample $\boldsymbol{x}_t^{(j)} \sim q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{y}_t) = p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^{(j)})$ by
        • generating $\boldsymbol{\eta}_t^{(j)}$ according to the state-noise density in equation (5.1)
        • setting $\boldsymbol{x}_t^{(j)} = f(\boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{\eta}_t^{(j)})$
      Filtering: Assign to each particle $\boldsymbol{x}_t^{(j)}$ the weight $\boldsymbol{\omega}_t^{(j)}$ according to expression in (5.12)
      Normalize the importance weights:

$$\tilde{w}_t^{(j)} = \frac{w_t^{(j)}}{\sum_{i=1}^{M} w_t^{(i)}}$$

    end for
Step 2 Resampling step
    Resampling with replacement the particles $\left\{(\boldsymbol{x}_t^{(j)}, \boldsymbol{\Theta}_t^{(j)}),\ j = 1, \ldots, M\right\}$, according to a resampling algorithm (residual or stratified) with importance weights $\tilde{\boldsymbol{\omega}}_t^{(j)}$
Step 3 Jittering step
    Choose discount factor δ in $(0, 1]$
    Compute the tuning parameter $a = (3\delta - 1)/2\delta$ based on the chosen discount factor δ.
    Compute the controlling smoothing parameter $h > 0$: $h^2 = 1 - a^2$
    for $j = 1$ to M **do**
      Compute the mean $\bar{\boldsymbol{\Theta}}_{t-1}$, and the variance V_{t-1} of the Monte Carlo approximation to $p(\boldsymbol{\Theta}|\boldsymbol{y}_{1:t-1})$.
      Sample a new parameter vector: $\boldsymbol{\Theta}_t^{(j)} \sim \mathcal{N}(\bullet|m_t^{(j)}, V_t)$,
      where $m_t^{(j)} = a\boldsymbol{\Theta}_{t-1}^{(j)} + (1-a)\bar{\boldsymbol{\Theta}}_{t-1}$,
      $V_t = h^2 V_{t-1}$
    end for
  end for

5.7 The EPFJ and UPFJ: Exploring the Effect of the Jittering Step in the Extended and Unscented Particle Filter Variants

Herein, the EPF or the UPF particle filter variant is used to model the time-varying states and the jittering approach to model the unknown fixed model parameters. As a result, two PF variants, named by us EPFJ and UPFJ, arrive as an attempt to tackle the sample impoverishment problem when handling the simultaneous estimation of states and parameters. Notice that the EPFJ is also denoted by KPFJ when

the model at hand is linear and the KF is used instead of the EKF. The following pseudocodes outline how the original EPF and UPF filters are modified to tackle the simultaneous estimation of states and fixed model parameters. Thus, the pseudocodes for the EPFJ and UPFJ PF variants are presented in algorithms 14 (page 153) and 15 (page 154), respectively.

In the next section, a Monte Carlo study is carried out to assess the performance of the particle filtering methodology when dealing with the simultaneous estimation of states and parameters. The non-stationary local level process, covered in Chapter 3 and specified in equation (3.9), is used as a benchmark model and Liu and West (2001) particle filter variant as a benchmark filter.

5.8 The Non-Stationary Local Level Model Revisited: Monte Carlo Study for the Simultaneous Estimation of States and Parameters

In Chapter 3, we illustrated, via a Monte Carlo study, the filtering performance of the particle filter variants named: SIR, SIRopt, ASIR and KPFJ, using as a benchmark the non-stationary but linear and Gaussian local level model. Therein, the simulation based filters mentioned were used to estimate the state level and confronted with the analytical Kalman filter, which is optimal for the model at hand.

In contraposition to Chapter 3, which deals solely with state estimation, the present chapter deals with the simultaneous estimation of states and fixed model parameters. In this new scenario, the KF by itself is no longer an alternative and thus we place our effort in studying a couple of existing particle filter variants already used for parameter estimation as well as in adapting (when feasible) some of the particle filter variants described in Chapter 2 and later used in Chapters 3 and 4 for state estimation. In other words, we start by studying the self organizing particle filter variant proposed by the authors: Kitagawa (1998) and Kitagawa and Sato (2001); which is outlined in Algorithm 11. We also study the widely used Liu and West (2001) particle filter variant outlined in Algorithm 12, which basically is constructed by combining the ASIR particle filter variant and kernel smoothing ideas. We provide further details on these filters later on.

As already mentioned, our contribution in this chapter is the definition and implementation in R language of the following three particle filter variants: SIRJ, SIRoptJ and KPFJ, which arise as extensions of existing ones. These are confronted with the widely used LW approach, also implemented by us in R language. In the sequel, all details regarding the undergone Monte Carlo study are presented. We remark that, although in the previous section we propose to explore the combined use of the EPF (or the UPF) with the jittering approach, say to apply the EPFJ and the UPFJ, their use in this simulation study is not needed as the non-stationary and Gaussian local level model at hand has a linear structure. Be reminded, however, that in Chapter 4 we implemented the nonlinear EPF and UPF filters, but in that case we were dealing with estimating the states of a highly complex nonlinear state-space model.

Algorithm 14 Extended Particle Filter plus Jittering (EPFJ)

Initialization $t = 0$

for $j = 1$ to M **do**

Sample $\boldsymbol{x}_0^{(j)} \sim p(\boldsymbol{x}_0)$

Sample $\boldsymbol{\Theta}_0^{(j)} \sim p(\boldsymbol{\Theta}_0)$

end for

for $t = 1$ to N **do**

Step 1 Importance sampling step

for $j = 1$ to M **do**

Step 2 Prediction step

Compute $J_{\boldsymbol{x}_{t-1}}$, and $J_{\boldsymbol{\eta}_t}$ as in equation (2.19).

Compute the predictive expectation $\bar{\boldsymbol{x}}_{t|t-1}^{(j)}$ and covariance $\boldsymbol{\Sigma}_{\boldsymbol{x}_{t|t-1}}^{(j)}$ using

$\bar{\boldsymbol{x}}_{t|t-1}^{(j)} = f(\boldsymbol{x}_{t-1}^{(j)}, 0)$ and

$\boldsymbol{\Sigma}_{\boldsymbol{x}_{t|t-1}}^{(j)} = J_{\boldsymbol{x}_{t-1}}^{(j)} \boldsymbol{\Sigma}_{\boldsymbol{x}_{t-1|t-1}}^{(j)} J_{\boldsymbol{x}_{t-1}}^{\prime(j)} + J_{\boldsymbol{\eta}_t}^{(j)} Q_t J_{\boldsymbol{\eta}_t}^{\prime(j)}$, respectively.

Step 3 Kalman Gain step

Compute $J_{\boldsymbol{x}_t}$, and $J_{\boldsymbol{\nu}_t}$ as in equation (2.20).

Compute the prediction estimate $\boldsymbol{y}_{t|t-1}^{(j)}$ and covariance $\boldsymbol{\Sigma}_{\boldsymbol{y}_{t|t-1}}^{(j)}$ using equations (2.26) and (2.27), respectively.

$\boldsymbol{y}_{t|t-1}^{(j)} = h_{t|t-1}^{(j)} = h(\bar{\boldsymbol{x}}_{t|t-1}^{(j)}, 0)$

$\boldsymbol{\Sigma}_{\boldsymbol{y}_{t|t-1}} = J_{\boldsymbol{x}_t}^{(j)} \boldsymbol{\Sigma}_{\boldsymbol{x}_{t|t-1}}^{(j)} J_{\boldsymbol{x}_t}^{\prime(j)} + J_{\boldsymbol{\nu}_t}^{(j)} R_t J_{\boldsymbol{\nu}_t}^{\prime(j)}$

Compute the Kalman Gain K_t with equation (2.28).

$K_t = \boldsymbol{\Sigma}_{\boldsymbol{x}_{t|t-1}}^{(j)} J_{\boldsymbol{x}_t}^{\prime(j)} \boldsymbol{\Sigma}_{\boldsymbol{y}_{t|t-1}}^{-1(j)}$

Step 4 Filtering step

Compute the filtering expectation $\bar{\boldsymbol{x}}_{t|t}$ and covariance $\boldsymbol{\Sigma}_{\boldsymbol{x}_{t|t}}$ using (2.29) and (2.30), respectively.

$\bar{\boldsymbol{x}}_t^{(j)^{EKF}} = \bar{\boldsymbol{x}}_{t|t-1}^{(j)} + K_t(\boldsymbol{y}_t - \boldsymbol{y}_{t|t-1}^{(j)})$

$\boldsymbol{\Sigma}_t^{(j)^{EKF}} = \boldsymbol{\Sigma}_{\boldsymbol{x}_{t|t-1}}^{(j)} - K_t J_{\boldsymbol{x}_t}^{(j)} \boldsymbol{\Sigma}_{\boldsymbol{x}_{t|t-1}}^{(j)}$

Sample $\boldsymbol{x}_t^{(j)} \sim q(\boldsymbol{x}_t^{(j)} | \boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{y}_{1:t}) \stackrel{\circ}{=} \mathcal{N}(\bar{\boldsymbol{x}}_t^{(j)^{EKF}}, \boldsymbol{\Sigma}_t^{(j)^{EKF}})$

end for

for $j = 1$ to M **do**

Evaluate the importance weights up to a normalizing constant.

$\boldsymbol{\omega}_t^{(j)} \propto \frac{p(\boldsymbol{y}_t | \boldsymbol{x}_t^{(j)^{EKF}})\, p(\boldsymbol{x}_t^{(j)^{EKF}} | \boldsymbol{x}_{t-1}^{(j)})}{q(\boldsymbol{x}_t^{(j)} | \boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{y}_t)}$

end for

for $j = 1$ to M **do**

Normalize the importance weights $\tilde{\boldsymbol{\omega}}_t^{(j)} = \frac{\boldsymbol{\omega}_t^{(j)}}{\sum_{i=1}^{M} \boldsymbol{\omega}_t^{(i)}}$.

end for

Step 5 Resample the discrete PDF to obtain a sample of size M.

Multiply/Supress particles $\boldsymbol{x}_t^{(j)}$ according to high/low importance weights, $\tilde{\boldsymbol{\omega}}_t^{(j)}$

end for

Step 6 Jittering step

Choose discount factor δ in $(0, 1]$

Compute the tuning parameter $a = (3\delta - 1)/2\delta$ based on the chosen discount factor δ.

Compute the controlling smoothing parameter $h > 0$: $h^2 = 1 - a^2$

for $j = 1$ to M **do**

Compute the mean $\bar{\boldsymbol{\Theta}}_{t-1}$, and the variance V_{t-1} of the Monte Carlo approximation to $p(\boldsymbol{\Theta} | \boldsymbol{y}_{1:t-1})$.

Sample a new parameter vector: $\boldsymbol{\Theta}_t^{(j)} \sim \mathcal{N}(\bullet | m_t^{(j)}, V_t)$, where $m_t^{(j)} = a\boldsymbol{\Theta}_{t-1}^{(j)} + (1-a)\bar{\boldsymbol{\Theta}}_{t-1}$,

$V_t = h^2 V_{t-1}$

end for

Algorithm 15 Unscented Particle Filter plus Jittering (UPFJ)

Initialization $t = 0$

for $j = 1$ to M **do**

Sample $x_0^{(j)} \sim p(x_0)$ and set:

$\bar{\boldsymbol{x}}_0^{(j)} = \mathrm{E}(\boldsymbol{x_0}^{(j)}) \qquad \Sigma_0^{(j)} = E\left((\boldsymbol{x_0}^{(j)} - \bar{\boldsymbol{x}}_0^{(j)})(\boldsymbol{x_0}^{(j)} - \bar{\boldsymbol{x}}_0^{(j)})'\right)$

$\bar{\boldsymbol{x}}_0^{(j)a} = \mathrm{E}(\boldsymbol{x}_0^{(j)a}) = ((\bar{\boldsymbol{x}}_0^{(j)})', \mathbf{0}, \mathbf{0})' \quad \Sigma_0^{(j)a} = E\left((\boldsymbol{x}_0^{(j)a} - \bar{\boldsymbol{x}}_0^{(j)a})(\boldsymbol{x}_0^{(j)a} - \bar{\boldsymbol{x}}_0^{(j)a})'\right)$

Sample fixed unknown parameters: $\boldsymbol{\Theta}_0^{(j)} \sim p(\boldsymbol{\Theta}_0)$.

end for

for $t = 1$ to N **do**

Step 1 Importance sampling step

for $j = 1$ to M **do**

Step 2 Update the particles with the UKF

Compute the sigma points

$$\boldsymbol{\chi}_{t-1}^{(j)a} = [\bar{\boldsymbol{x}}_{t-1}^{(j)a}, \bar{\boldsymbol{x}}_{t-1}^{(j)a} \pm \left(\sqrt{(n_a + \lambda)\Sigma_{t-1}^{(j)a}}\right)]$$

Time update (propagate particles into the future)

$$\boldsymbol{\chi}_{t|t-1}^{(j)x} = f(\boldsymbol{\chi}_{t-1}^{(j)x}, \boldsymbol{\chi}_{t-1}^{(j)\eta}) \qquad \bar{\boldsymbol{x}}_{t|t-1}^{(j)} = \sum_{i=0}^{2n_a} \boldsymbol{\omega}_i^{(m)} \boldsymbol{\chi}_{i,t|t-1}^{(j)x}$$

$$\Sigma_{t|t-1}^{(j)} = \sum_{i=0}^{2n_a} \boldsymbol{\omega}_i^{(c)} (\boldsymbol{\chi}_{i,t|t-1}^{(j)x} - \bar{\boldsymbol{x}}_{t|t-1}^{(j)})(\boldsymbol{\chi}_{i,t|t-1}^{(j)x} - \bar{\boldsymbol{x}}_{t|t-1}^{(j)})'$$

$$\boldsymbol{y}_{t|t-1}^{(j)} = h(\boldsymbol{\chi}_{t|t-1}^{(j)x}, \boldsymbol{\chi}_{t-1}^{(j)v}) \qquad \bar{\boldsymbol{y}}_{t|t-1}^{(j)} = \sum_{i=0}^{2n_a} \boldsymbol{\omega}_i^{(m)} \boldsymbol{y}_{i,t|t-1}^{(j)}$$

Measurement update (incorporate new observation)

$$\Sigma_{\tilde{\boldsymbol{y}}_t, \tilde{\boldsymbol{y}}_t} = \sum_{i=0}^{2n_a} \boldsymbol{\omega}_i^{(c)} (\boldsymbol{y}_{i,t|t-1}^{(j)} - \bar{\boldsymbol{y}}_{t|t-1}^{(j)})(\boldsymbol{y}_{i,t|t-1}^{(j)} - \bar{\boldsymbol{y}}_{t|t-1}^{(j)})'$$

$$\Sigma_{\boldsymbol{x}_t, \tilde{\boldsymbol{y}}_t} = \sum_{i=0}^{2n_a} \boldsymbol{\omega}_i^{(c)} (\boldsymbol{\chi}_{i,t|t-1}^{(j)x} - \bar{\boldsymbol{x}}_{t|t-1}^{(j)})(\boldsymbol{y}_{i,t|t-1}^{(j)} - \bar{\boldsymbol{y}}_{t|t-1}^{(j)})' \quad K_t = \Sigma_{\boldsymbol{x}_t, \tilde{\boldsymbol{y}}_t} \Sigma_{\tilde{\boldsymbol{y}}_t, \tilde{\boldsymbol{y}}_t}^{-1}$$

$$\bar{\boldsymbol{x}}_t^{(j)UKF} = \bar{\boldsymbol{x}}_{t|t-1}^{(j)} + K_t(\boldsymbol{y}_t - \bar{\boldsymbol{y}}_{t|t-1}^{(j)}) \quad \Sigma_t^{(j)UKF} = \Sigma_{t|t-1}^{(j)} - K_t \Sigma_{\tilde{\boldsymbol{y}}_t, \tilde{\boldsymbol{y}}_t} K_t'$$

Sample $\boldsymbol{x}_t^{(j)} \sim q(\boldsymbol{x}_t^{(j)} | \boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{y}_{1:t}) \stackrel{\circ}{=} \mathcal{N}(\bar{\boldsymbol{x}}_t^{(j)UKF}, \boldsymbol{\Sigma}_t^{(j)UKF})$

end for

for $j = 1$ to M **do**

Evaluate the importance weights up to a normalizing constant.

$$\boldsymbol{\omega}_t^{(j)} \propto \frac{p(\boldsymbol{y}_t | \boldsymbol{x}_t^{(j)UKF})\, p(\boldsymbol{x}_t^{(j)UKF} | \boldsymbol{x}_{t-1}^{(j)})}{q(\boldsymbol{x}_t^{(j)} | \boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{y}_t)}$$

end for

for $j = 1$ to M **do**

Normalize the importance weights $\tilde{\boldsymbol{\omega}}_t^{(j)} = \frac{\boldsymbol{\omega}_t^{(j)}}{\sum_{i=1}^{M} \boldsymbol{\omega}_t^{(i)}}$.

end for

Step 3 Resample the discrete PDF to obtain a sample of size M.

Multiply/Supress particles $\boldsymbol{x}_t^{(j)}$ according to high/low importance weights, $\tilde{\boldsymbol{\omega}}_t^{(j)}$

end for

Step 4 Jittering step

Choose discount factor δ in $(0, 1]$

Compute the tuning parameter $a = (3\delta - 1)/2\delta$ based on the chosen discount factor δ.

Compute the controlling smoothing parameter $h > 0$: $h^2 = 1 - a^2$

for $j = 1$ to M **do**

Compute the mean $\bar{\boldsymbol{\Theta}}_{t-1}$, and the variance V_{t-1} of the Monte Carlo approximation to $p(\boldsymbol{\Theta} | \boldsymbol{y}_{1:t-1})$.

Sample a new parameter vector: $\boldsymbol{\Theta}_t^{(j)} \sim \mathcal{N}(\bullet | m_t^{(j)}, V_t)$, where $m_t^{(j)} = a\boldsymbol{\Theta}_{t-1}^{(j)} + (1-a)\bar{\boldsymbol{\Theta}}_{t-1}$, $V_t = h^2 V_{t-1}$;

end for

5.8.1 The Augmented State Space Representation

For the non stationary local level model, the state-space formulation for the augmented state-space vector $\boldsymbol{l}_t = (\boldsymbol{x}_t, \boldsymbol{\Theta})' = (x_t, \boldsymbol{\Theta} = (\sigma_\eta, \sigma_\nu))'$; with general formulation presented in equations (5.1) and (5.2), take the specific form:

$$\boldsymbol{l}_t = \begin{bmatrix} \boldsymbol{x}_t \\ \boldsymbol{\Theta} \end{bmatrix} = \tilde{f}(\boldsymbol{l}_{t-1}, \boldsymbol{\eta}_t) = \begin{bmatrix} f(x_{t-1}, \eta_t) \\ \boldsymbol{\Theta} \end{bmatrix} = \begin{bmatrix} x_{t-1} + \eta_t \\ \begin{bmatrix} \sigma_\eta \\ \sigma_\nu \end{bmatrix} \end{bmatrix}, \qquad \textit{(Transition equation)} \qquad (5.30)$$

$$\boldsymbol{y}_t = \tilde{h}(\boldsymbol{l}_t, \boldsymbol{\nu}_t) = h(x_t, \nu_t) = x_t + \nu_t, \qquad \textit{(Measurement equation)} \qquad (5.31)$$

where $\boldsymbol{x}_t$ is the latent local level, the uncorrelated sequences η_t and ν_t follow a Gaussian distribution and $\boldsymbol{\Theta} = (\sigma_\eta, \sigma_\nu))'$ is the parameter vector containing the unknown scale parameters σ_η and σ_ν. Notice that when the parameters are incorporated into the state vector, we have a nonlinear filtering problem. To complete this state-space formulation, a prior distribution on the initial augmented state vector $\boldsymbol{l}_0$, must be assumed. Following, we present the prior distributions used in this particular case.

5.8.2 A Note About the Priors Used

The priors used for the local level model are the ones usually found in the literature, a normal prior for the original state variable x_t (the level) and an inverse gamma prior for the variance parameters σ^2_η and σ^2_ν; see for instance Congdon (2007) and Lopes and Tsay (2011). For our simulations, we choose a normal prior for x_t of the form $x_0 \sim \mathcal{N}(\mu_{x_0}, \Sigma_{x_0})$ with hyper parameters μ_{x_0} and Σ_{x_0}. The assumed priors for the variance parameters are inverse gamma distributions formulated as $\sigma^2_{\eta,0} \sim IG(\frac{n0}{2}, \frac{n0}{2} \cdot S^2_{\eta_{t=0}})$ and $\sigma^2_{\nu,0} \sim IG(\frac{n0}{2}, \frac{n0}{2} \cdot S^2_{\nu_{t=0}})$. The chosen hyper parameters values for these variance priors are $n_0 = 10$, $S^2_{\eta_{t=0}} = \sigma^2_\eta$ and $S^2_{\nu_{t=0}} = \sigma^2_\nu$; the last two equated to the true transition and measurement noise variance values, respectively. Notice that we use diffuse priors non-centered in the true values with prior means given by $\frac{n0}{n0-2}\sigma^2_\eta$ and $\frac{n0}{n0-2}\sigma^2_\nu$, respectively.

Following, we carry out a Monte Carlo study to assess the statistical and computational performance of the four competing particle filter variants when handling the simultaneous estimation of states and parameters of the non-stationary local level model; the LW particle filter variant is taken as a benchmark filter.

5.8.3 General Procedure for the Simulation Design and Summary of Simulation Settings

The simulation study carried out in this chapter uses the same general simulation procedure followed in previous chapters to estimate solely the states, but is adapted to the new situation of estimating simultaneously the states and fixed model parameters. This procedure undertakes the following three steps:

- **STEP I**: Data and state generation

- **STEP II**: Filtering: simultaneous estimation of states and parameters
- **STEP III**: Filtering performance criteria computation

Following, we provide a detailed description of the aforementioned general simulation steps. Notice that within every simulation step, we further specify, when needed, the instructions to carry out the MC experiment.

STEP I: Data and State Generation

Generate $S = 100$ realizations of the chosen non-stationary local level dynamic model. This is done in exactly the same way as explained on page 51 of Chapter 3.

STEP II: Filtering Estimation

For each nonlinear filter f, obtain both the statistical and computational measure of performance of the studied nonlinear filter f. These are based on the root mean square (RMSE) and on the CPU time. That is, assuming all model parameters are unknown and given the simulated data $\boldsymbol{y}_{1:T} = y_1, \ldots, y_T$ obtained in step (I), for replication set i, $i = 1, \ldots, S$, proceed to

(IIa) In this case, the variance parameters are unknown and thus are also assigned a prior distribution.

(IIb) Compute the filtering estimates $\hat{\boldsymbol{x}}^f_{t,[i]}$, $t = 1, \ldots, T$ using the nonlinear filter in question, say f. Recall that $f \in$ {LW, SIRJ, SIRoptJ, KPFJ}. In this case, also obtain the estimates of the two variance parameters of the local level model.

(IIc) Compute $RMSE^f_{[i]}$: the RMSE over time index $t = 1, \ldots, T$ with equation (3.4) of Chapter 3. However, in this case also compute the corresponding RMSE values for the the two unknown fixed model parameters.

(IId) Compute $CPU^f_{[i]}$: the total elapsed time for a total of T observations with equation (3.5).

(IIe) Repeat steps (IIa)–(IIc) $S = 100$ times.

STEP III: Filtering Performance Criteria Computation

(IIIa) In step (IIb), we end up with $S = 100$ estimates of the RMSE: $RMSE^f_{[i]}$. Based on these, obtain the mean and the variance of the root mean square (RMSE) computed over time and over replication sets using equations (3.6) and (3.7), respectively. Likewise, compute the same values corresponding to the the two unknown fixed model parameters.

(IIIb) In step (IIc), we end up with $S = 100$ CPU elapsed-time estimates: $CPU^f_{[i]}$. Based on these, obtain the mean CPU elapsed-time computed over replication sets using (3.8).

For completion, the reader may refer to Figure A.1 in Appendix A, which is a sketch illustrating the comparison criteria for the simulation based filters under study.

Following, we provide a summary list of the simulation settings used for the conducted Monte Carlo experiment:

Simulation Settings

- **Filters**: LW, SIRJ, SIRoptJ, and KPFJ.
- **Measurement noise variance**: Fixed to $\sigma_v^2 = 0.1$.
- **State noise variance** σ_η^2: Six scenarios typically found in real data applications, defined in terms of the SNR $q \in \{0.001, 0.1, 0.5, 1, 2, 5\}$; being $q = \frac{\sigma_\eta^2}{\sigma_v^2}$. Notice that these signal-to-noise values are a subset of the settings presented in Table 3.1, Chapter 3. Also have in mind that both variance parameters are assumed to be fixed and unknown.
- **Resampling scheme**: Stratified resampling.
- **Number of replications**: $S = 100$.
- **Number of particles**: $N_p = 5000$. In Chapter 3 dealing solely with estimating the states of the same local level model, this number of particles is found to provide satisfactory estimation performance for most particle filter variants.
- **Time series length**: $T = 200$. In In Chapter 3 larger values of the time series length T are entertained for this model.
- **Discount parameter**: $\delta \in \{0.95, 0.83\}$. The first discount parameter value lies within the range of values usually taken for this parameter, say between 0.95 and 0.99. The second, lying outside the range of values recommended by (Liu and West 2001), is used to explore its potential distinctive impact on the quality of the estimations.
- **Comparison criteria**: RMSE and CPU time. Additionally, as done in previous chapters, we also report %uNp as a measure of the degree of degeneracy.

Recall that, as defined in Chapter 3 the statistical performance of the filters is defined in terms of the mean and variance of the RMSE, computed with (3.6) and (3.7), respectively. Likewise, the computational performance is measured by the mean elapsed CPU time, computed with (3.8).

In the sequel, we present the simulation results, remarks and conclusions regarding the undergone MC study.

5.8.4 Simulation Results

In Table 5.1, we provide the numeric results which summarize the performance of the different particle filters in handling the simultaneous estimation of states and parameters for the local level model. This table is organized in two different blocks corresponding to the two values used for the discount parameter: $\delta = 0.83$ and $\delta = 0.95$. Each block itself consists of three columns containing the measures: Mean(RMSE), Var(RMSE) and the average number (%) of distinct particles at time-index $t = T$. Have in mind that, except for Case 1, all estimated RMSE values are rounded up to three decimal points; for that reason, many simulation results may appear –at first sight– the same.

Table 5.1: Summary of simulation results: Simultaneous estimation of states and parameters for the local level model; $N_p = 5000$, $T = 200$

			$\delta = 0.83$			$\delta = 0.95$		
			RMSE			RMSE		
Setting	**Filter**	**Θ**	**Mean**	**Var**	**uNp (%uNp)**	**Mean**	**Var**	**uNp (%uNp)**
Case 1: $\sigma_\eta^2 = 0.0001$, SNR = 0.001								
	LW	x_t	0.065	2e-04	4576 (91)	0.065	2e-04	4586 (92)
		σ_η	3e-05	1e-06		3e-05	1e-06	
		σ_ν	0.015	4e-05		0.014	4e-05	
	SIRJ	x_t	0.065	2e-04	4586 (92)	0.065	2e-04	4588 (92)
		σ_η	3e-05	1e-06		3e-05	1e-06	
		σ_ν	0.015	1e-06		0.014	5e-05	
	SIRoptJ	x_t	0.065	2e-04	4590 (92)	0.065	2e-04	4593 (92)
		σ_η	3e-05	1e-06		3e-05	1e-06	
		σ_ν	0.015	5e-05		0.015	5e-05	
	KPFJ	x_t	0.069	3e-04	1559 (31)	0.069	3e-04	1592 (32)
		σ_η	3e-05	1e-06		5e-05	1e-06	
		σ_ν	0.014	8e-05		0.015	2e-04	
Case 2: $\sigma_\eta^2 = 0.01$, SNR = 0.1								
	LW	x_t	0.165	2e-04	3964 (79)	0.165	2e-04	3969 (79)
		σ_η	0.002	1e-06		0.002	1e-06	
		σ_ν	0.015	4e-05		0.015	5e-05	
	SIRJ	x_t	0.165	2e-04	3969 (79)	0.165	2e-04	3971 (79)
		σ_η	0.002	1e-06		0.002	1e-06	
		σ_ν	0.015	5e-05		0.015	5e-05	
	SIRoptJ	x_t	0.165	2e-04	4156 (83)	0.165	2e-04	4158 (83)
		σ_η	0.002	1e-06		0.002	1e-06	
		σ_ν	0.015	5e-05		0.015	5e-05	
	KPFJ	x_t	0.165	2e-04	3255 (65)	0.166	2e-04	3253 (65)
		σ_η	0.002	1e-06		0.002	1e-06	
		σ_ν	0.015	5e-05		0.015	5e-05	

Table 5.1: Summary of simulation results: Simultaneous estimation of states and parameters for the local level model; $N_p = 5000$, $T = 200$ (continued)

			$\delta = 0.83$			$\delta = 0.95$		
			RMSE			RMSE		
Setting	**Filter**	**Θ**	**Mean**	**Var**	**uNp (%uNp)**	**Mean**	**Var**	**uNp (%uNp)**
Case 3: $\sigma_\eta^2 = 0.05$, SNR = 0.5								
	LW	x_t	0.223	2e-04	3432 (69)	0.223	2e-04	3437 (69)
		σ_η	0.010	2e-05		0.010	2e-05	
		σ_ν	0.016	5e-05		0.016	6e-05	
	SIRJ	x_t	0.223	2e-04	3436 (69)	0.223	2e-04	3436 (69)
		σ_η	0.010	2e-05		0.010	2e-05	
		σ_ν	0.016	5e-05		0.016	5e-05	
	SIRoptJ	x_t	0.223	2e-04	4053 (81)	0.223	2e-04	4055 (81)
		σ_η	0.010	2e-05		0.010	2e-05	
		σ_ν	0.016	5e-05		0.016	6e-05	
	KPFJ	x_t	0.223	2e-04	3755 (75)	0.223	2e-04	3756 (75)
		σ_η	0.010	2e-05		0.010	2e-05	
		σ_ν	0.017	6e-05		0.017	6e-05	
Case 4: $\sigma_\eta^2 = 0.1$, SNR = 1								
	LW	x_t	0.247	2e-04	3116 (62)	0.247	2e-04	3118 (62)
		σ_η	0.018	5e-05		0.019	6e-05	
		σ_ν	0.017	6e-05		0.018	7e-05	
	SIRJ	x_t	0.247	3e-04	3115 (62)	0.247	2e-04	3118 (62)
		σ_η	0.019	6e-05		0.019	7e-05	
		σ_ν	0.017	7e-05		0.018	7e-05	
	SIRoptJ	x_t	0.247	2e-04	4086 (82)	0.247	2e-04	4088 (82)
		σ_η	0.018	6e-05		0.018	6e-05	
		σ_ν	0.018	6e-05		0.017	6e-05	
	KPFJ	x_t	0.247	2e-04	3942 (79)	0.247	2e-04	3943 (79)
		σ_η	0.018	6e-05		0.019	6e-05	
		σ_ν	0.018	7e-05		0.018	7e-05	

Table 5.1: Summary of simulation results: Simultaneous estimation of states and parameters for the local level model; $N_p = 5000$, $T = 200$ (continued)

			$\delta = 0.83$			$\delta = 0.95$		
			RMSE			RMSE		
Setting	**Filter**	**Θ**	**Mean**	**Var**	**uNp (%uNp)**	**Mean**	**Var**	**uNp (%uNp)**
Case 5: $\sigma_\eta^2 = 0.2$, SNR = 2								
	LW	x_t	0.269	2e-04	2749 (55)	0.269	2e-04	2757 (55)
		σ_η	0.034	2e-04		0.033	2e-04	
		σ_ν	0.019	9e-05		0.020	1e-04	
	SIRJ	x_t	0.269	2e-04	2753 (55)	0.269	2e-04	2752 (55)
		σ_η	0.034	2e-04		0.035	2e-04	
		σ_ν	0.018	8e-05		0.019	9e-05	
	SIRoptJ	x_t	0.268	2e-04	4175 (84)	0.269	2e-04	4176 (84)
		σ_η	0.033	2e-04		0.034	2e-04	
		σ_ν	0.019	8e-05		0.019	9e-05	
	KPFJ	x_t	0.268	2e-04	4118 (82)	0.269	2e-04	4118 (82)
		σ_η	0.033	2e-04		0.033	2e-04	
		σ_ν	0.019	7e-05		0.019	1e-04	
Case 6: $\sigma_\eta^2 = 0.5$, SNR = 5								
	LW	x_t	0.29	2e-04	2208 (44)	0.29	2e-04	2226 (44)
		σ_η	0.082	0.001		0.081	0.001	
		σ_ν	0.021	1e-04		0.022	2e-04	
	SIRJ	x_t	0.29	2e-04	2213 (44)	0.29	2e-04	2213 (44)
		σ_η	0.081	0.001		0.081	0.001	
		σ_ν	0.022	1e-04		0.022	2e-04	
	SIRoptJ	x_t	0.29	2e-04	4339 (87)	0.29	2e-04	4337 (87)
		σ_η	0.077	0.001		0.077	0.001	
		σ_ν	0.020	1e-04		0.022	1e-04	
	KPFJ	x_t	0.29	2e-04	4327 (87)	0.29	2e-04	4331 (87)
		σ_η	0.076	0.001		0.076	0.001	
		σ_ν	0.020	1e-04		0.021	1e-04	

To aid in the discussion of simulation results, we create a pictorial representation that allows us to have, at a glance, a very good idea of the main statistical findings contained in Table 5.1. Specifically, for the discount parameter $\delta = 0.83$, lying outside the range of values suggested by Liu and West (2001), we construct Figure 5.4 that depicts on the left panel the mean-RMSE[4] attained at the six chosen signal-

[4] Mean-RMSE and Mean(RMSE) used interchangeably

to-noise-ratio settings, and in the right panel the relative statistical performance of the competing filters in relation to the SIRJ particle filter variant. Thus, in this particular plot to represent the relative statistical performance, our proposed particle filter variant named SIRJ (Acosta, Martí-Recober, and Muñoz (2004), Muñoz, Márquez, and Acosta (2007)) is used as a reference algorithm when compared with the competing particle filters $f \in$ {LW, SIRoptJ, KPFJ}. The Figure 5.5 is a similar plot corresponding to the discount factor $\delta = 0.95$, which lies inside the range of values suggested by Liu and West (2001). Have in mind, though, that the LW is our benchmark particle filter variant.

5.8.5 Remarks and Conclusions

Based on simulation results reported in Table 5.1 and depicted on Figures 5.4–5.5, considering $N_p = 5000$ particles, we make the following remarks and conclusions regarding the performance of the competing filters when handling the simultaneous estimation of the states (the level) and parameters (transition and measurement noise variance) of the non-stationary local level model:

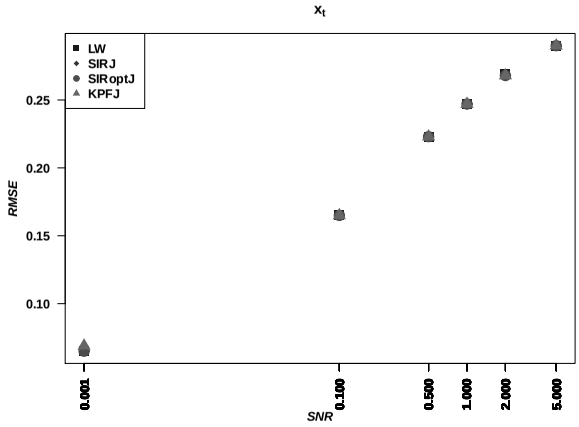
First, we refer to the effect of the discount factor δ, used in the estimation procedure, on the statistical performance of the filters. By choosing the two values of the discount factor δ, we aim to test the potential impact of it on the estimations of the states and the two variance parameters. To achieve that, we focus on the comparison of the respective mean-RMSE estimates obtained at the two chosen discount factors, $\delta = 0.83$ and $\delta = 0.95$. Have in mind, that comparisons among filters is an important target. Taking a look at the two aforementioned figures, we find that they are pretty much alike. Specifically:

- As expected, the statistical efficiency indicated by the mean-RMSE shows an increasing pattern as a function of the signal-to-noise-ratio values; this result holds irrespective of the value that is being estimated and of the discount factor δ; focus on the left panels of Figures 5.4–5.5.

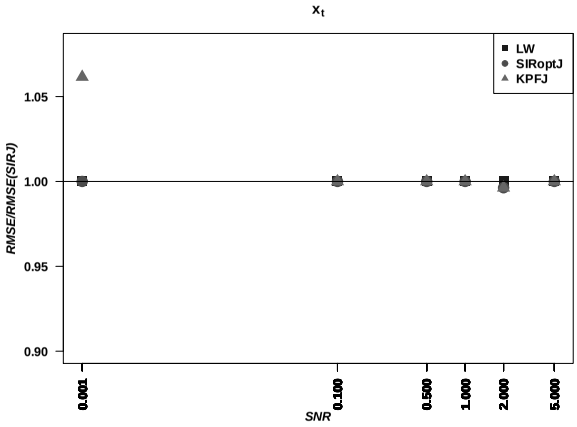
- The RMSE values corresponding to the states are hardly affected by the choice of the discount factor δ; focus on first rows of aforementioned figures. This makes sense, since they are not directly affected by the choice of the discount factor δ.

- As known, the fixed transition and measurement noise variances have been jittered and thus are directly affected by the choice of the discount factor δ. The question is whether this choice has an impact on the estimations of those two variance parameters. We find that some discrepancies are observed in the mean-RMSE of the transition and measurement variance parameters, specially in the latter; focus on last two rows of Figures 5.4–5.5. We consider, however, that the observed differences are too small and that the choice of δ does not seem to have a clear effect on the variance estimations. Notice that differences are observed in the third decimal place; for completion see Table 5.1.

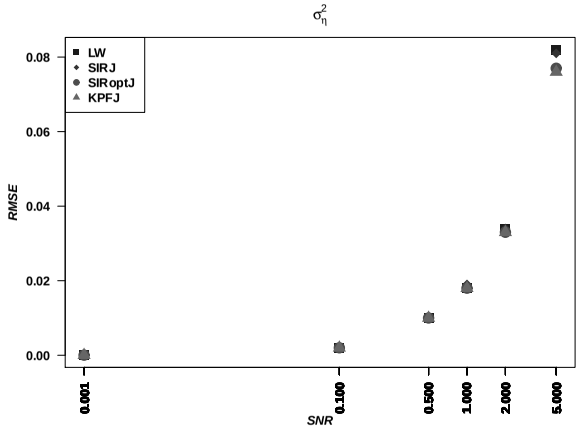
- Therefore, we consider that for the model at hand, any of these two discount values could be used. However, we prefer to use the discount factor $\delta = 0.95$ as we have typically done and also because it belongs to the range of values suggested by Liu and West (2001). We remark, though, that a further Monte Carlo study must be undergone to completely rule out the potential impact of the discount factor choice.



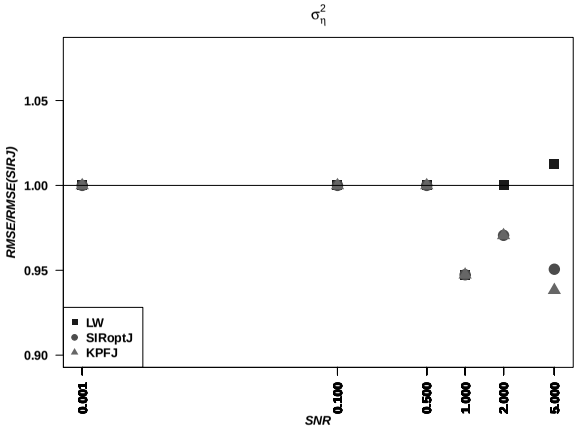
(a) State (x_t): Mean(RMSE) vs signal-to-noise-ratio



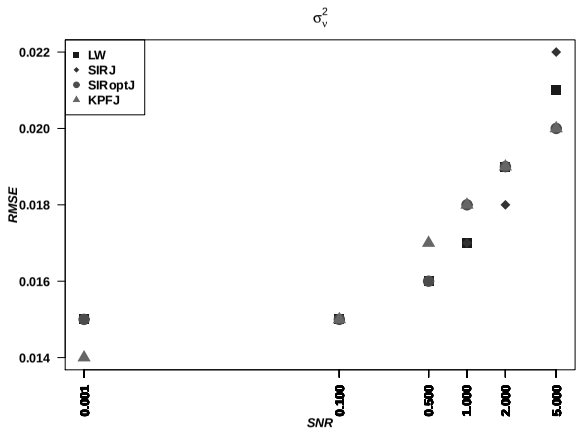
(b) State (x_t): Ratio of mean(RMSE)/mean(RMSE(SIR)) vs signal-to-noise-ratio



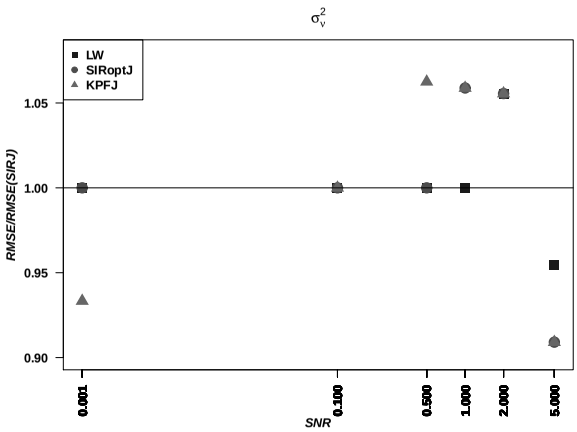
(c) Transition noise variance (σ_η^2): Mean(RMSE) vs signal-to-noise-ratio



(d) Transition noise variance (σ_η^2): Ratio of mean(RMSE)/mean(RMSE(SIR)) vs signal-to-noise-ratio

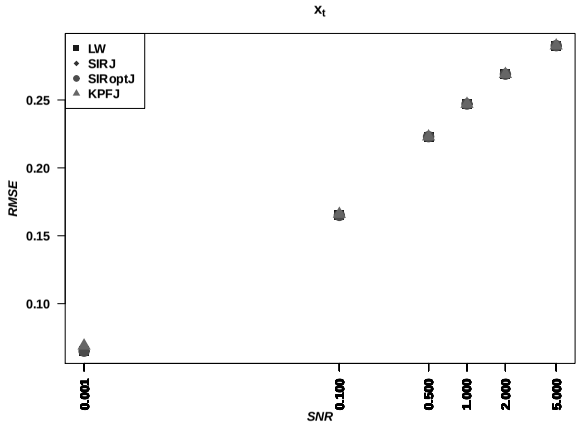


(e) Measurement noise variance (σ_v^2): Mean(RMSE) vs signal-to-noise-ratio

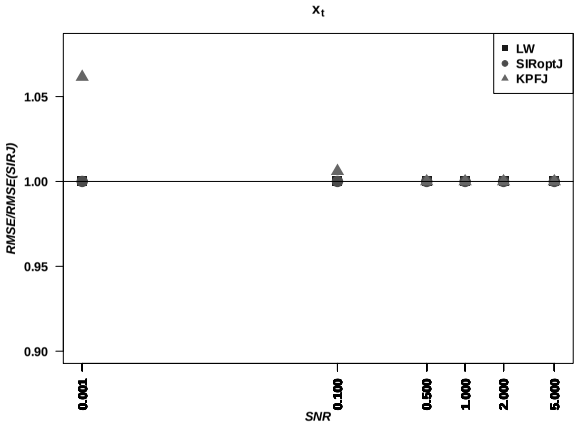


(f) Measurement noise variance (σ_v^2): Ratio of mean(RMSE)/mean(RMSE(SIR)) vs signal-to-noise-ratio

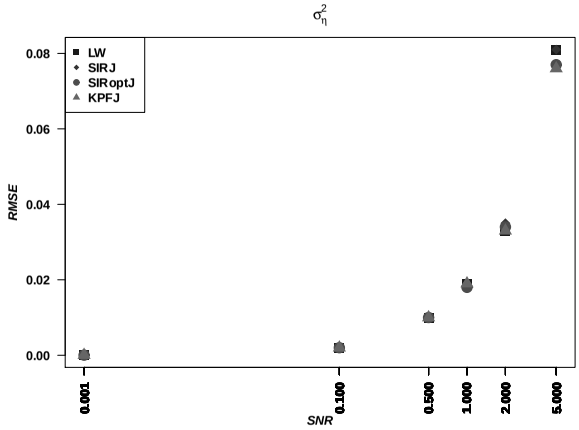
Figure 5.4: Local level model using $\delta = 0.83$: Impact of the signal-to-noise ratio value on the statistical performance of the filters indicated by the mean(RMSE); $T = 200$ and $N_p = 5000$.



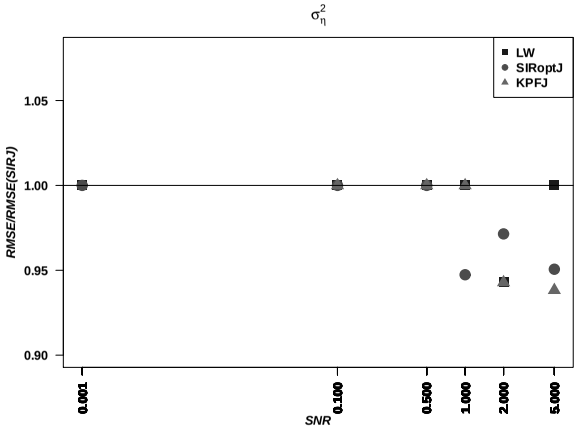
(a) State (x_t): Mean(RMSE) vs signal-to-noise-ratio



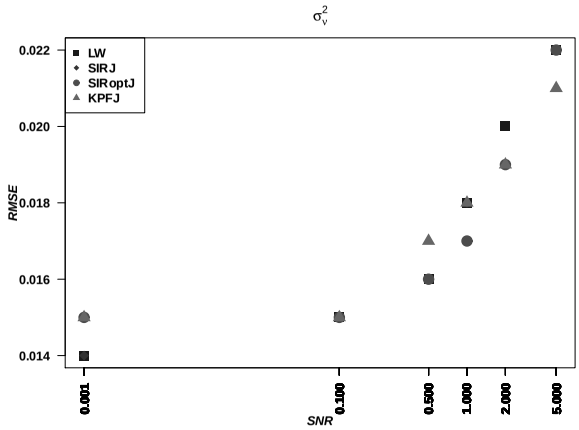
(b) State (x_t): Ratio of mean(RMSE)/mean(RMSE(SIR)) vs signal-to-noise-ratio



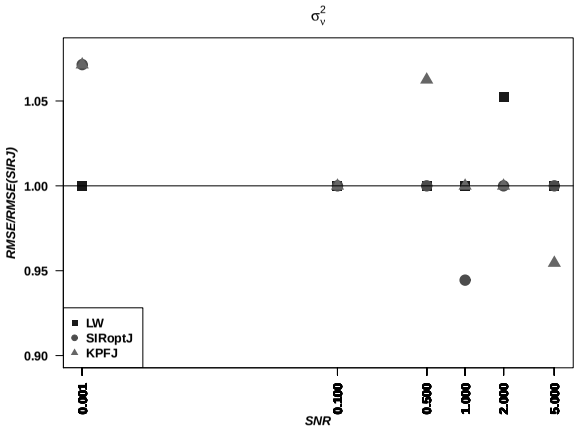
(c) Transition noise variance (σ^2_η): Mean(RMSE) vs signal-to-noise-ratio



(d) Transition noise variance (σ^2_η): Ratio of mean(RMSE)/mean(RMSE(SIR)) vs signal-to-noise-ratio



(e) Measurement noise variance (σ^2_v): Mean(RMSE) vs signal-to-noise-ratio



(f) Measurement noise variance (σ^2_v): Ratio of mean(RMSE)/mean(RMSE(SIR)) vs signal-to-noise-ratio

Figure 5.5: Local level model using $\delta = 0.95$: Impact of the signal-to-noise ratio value on the statistical performance of the filters indicated by the mean(RMSE); $T = 200$ and $N_p = 5000$.

Second, we assess the impact of the signal-to-noise-ratio values q on the statistical performance of the competing filters:

- When estimating the states, all four particle filter variants display an equal statistical performance, except for the KPFJ at low signal-to-noise ratio value $q = 0.001$.

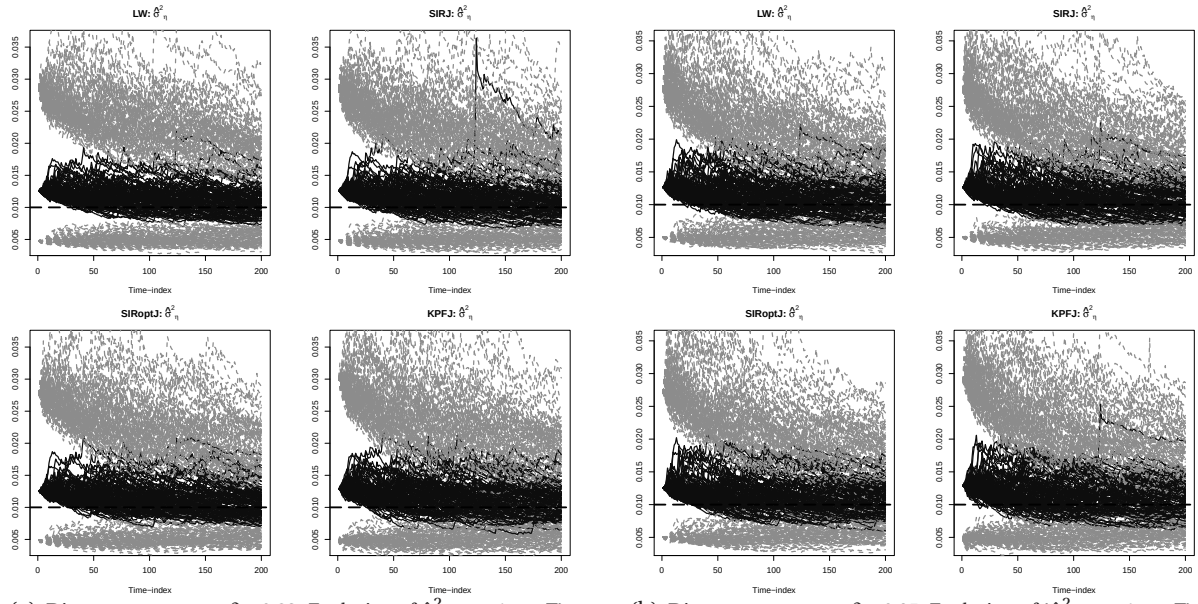
- When estimating the transition noise variance, all four particle filter variants display an equal statistical performance for signal-to-noise ratio values less than one; $q \in \{0.001, 0.1, 0.5\}$. However, for signal-to-noise ratio values greater or equal to one, $q \in \{1, 2, 5\}$, the results are mixed, but the LW and SIRJ yield practically equal mean-RMSE values. The other two filters: the SIRoptJ and the KPFJ are in the latter cases slightly more efficient than the SIRJ/LW particle filters. These results are in concordance with the ones obtained in Chapter 3 that showed that both the SIRopt and the KPF had better statistical performance at higher signal-to-noise-ratio values, and as fully described in the present chapter, these two filters conform the base for the SIRoptJ and the KPFJ particle filter variants, respectively.

- When estimating the measurement noise variance, the results are varied but we confirm that the LW and the SIRJ particle filters equate their statistical performance at most signal-to-noise-ratio values, showing only slight discrepancies at higher values. This is in concordance with the results obtained in Chapter 3 that showed that both the SIR and the ASIR had more difficulties at higher signal-to-noise-ratio values, and as fully described in the present chapter, these two filters conform the base for the SIRJ and the LW particle filter variants, respectively.

- As stated before, the observed differences are too small to indicate a clear preference of one filter over another; differences are observed in the third decimal place.

- To better illustrate that the observed discrepancies among filters are practically unnoticeable, regardless even of the choice of the discount factor δ, we go one step further and create for a signal-to-noise ratio $q = 0.1$ the Figures 5.6 – 5.7 and related Tables 5.2 – 5.3. These figures represent the evolution of the estimated noise variance (black/continuous) for all 100 Monte Carlo replications, $N_p = 5000$ and the four particle filter variants under study. Corresponding 2.5th and 97.5th percentiles are represented by gray/dashed lines and the true noise variance is depicted by a horizontal (black/dashed) line; the first plot corresponds to the transition noise variance parameter and the second to the measurement noise variance parameter. Below each figure, a related table is attached representing the evolution of the corresponding estimated noise variance obtained via the four PF variants under study for all 100 Monte Carlo replications. This evolution is shown for time-indexes $t \in \{50, 100, 150, 200\}$ and the data shown are in the format: Mean (2.5th, 97.5th percentiles) of the posterior mean estimates.

  Based on these figures and related tables, we confirm that for the local level model and the signal-to-noise-ratio value $q = 0.1$, all three competing particle filter variants are valid since they all are able to equate the LW filter statistical performance.



(a) Discount parameter $\delta = 0.83$: Evolution of $\hat{\sigma}^2_{\eta,t}$, $t = 1,\ldots,T\}$.

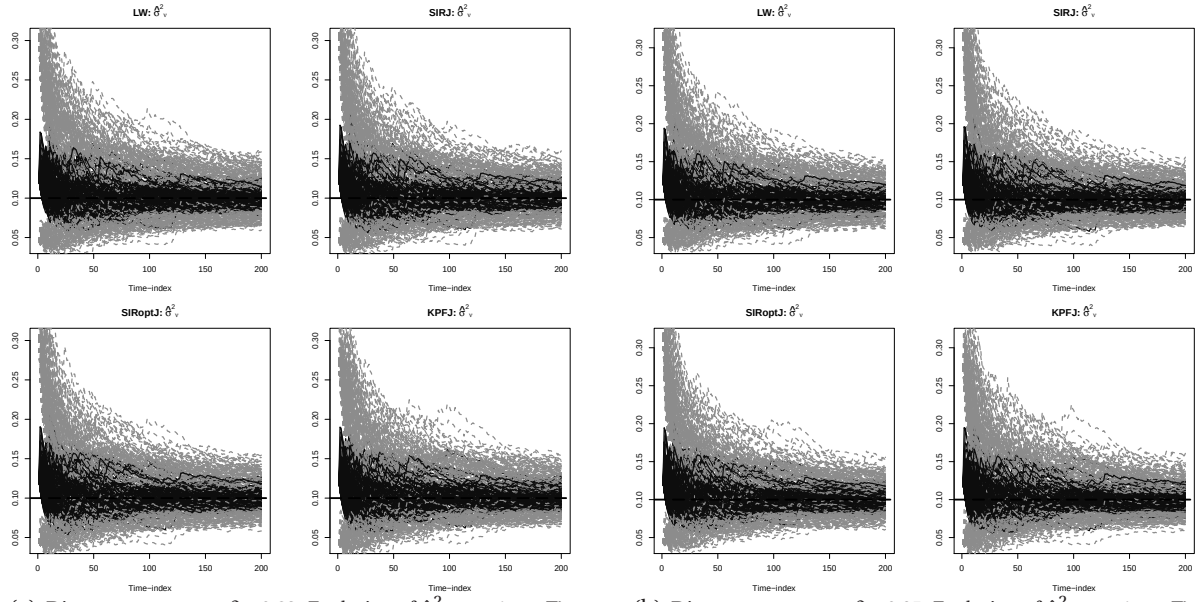
(b) Discount parameter $\delta = 0.95$: Evolution of $\{\hat{\sigma}^2_{\eta,t}$, $t = 1,\ldots,T\}$.

Figure 5.6: Local level model with $SNR = 0.1$: Evolution of estimated transition noise variance $\hat{\sigma}^2_{\eta,t}$ (black/continuous) for all 100 MC replications, $N_p = 5000$ and the four particle filter variants under study. Corresponding 2.5th and 97.5th percentiles are represented by gray/dashed lines and the true state noise variance: $\sigma^2_\eta = 0.01$ is depicted by a horizontal (black/dashed) line. Each left/right panel contains for each chosen discount parameter four sub figures: top-left: LW, top-right: SIRJ, bottom-left: SIRoptJ, bottom-right: KPFJ.

Table 5.2: Evolution of estimated transition noise variance $\hat{\sigma}^2_\eta$ for all 100 MC replications and the four PF variants under study with $SNR = 0.1$ and time series length $T \in \{50, 100, 150, 200\}$. True state noise variance: $\sigma^2_\eta = 0.01$.

δ	Filter	T=50	T=100	T=150	T=200
0.83	**LW**	0.012 (0.010, 0.015)	0.012 (0.009, 0.016)	0.011 (0.008, 0.016)	0.011 (0.008, 0.015)
	SIRJ	0.012 (0.009, 0.015)	0.012 (0.008, 0.016)	0.011 (0.008, 0.017)	0.011 (0.007, 0.016)
	SIRoptJ	0.012 (0.009, 0.016)	0.011 (0.008, 0.016)	0.011 (0.008, 0.016)	0.011 (0.007, 0.016)
	KPFJ	0.012 (0.009, 0.016)	0.012 (0.008, 0.017)	0.011 (0.007, 0.016)	0.011 (0.007, 0.016)
0.95	**LW**	0.012 (0.009, 0.017)	0.011 (0.008, 0.017)	0.011 (0.008, 0.016)	0.011 (0.007, 0.015)
	SIRJ	0.012 (0.009, 0.016)	0.011 (0.008, 0.016)	0.011 (0.007, 0.016)	0.011 (0.008, 0.016)
	SIRoptJ	0.012 (0.009, 0.017)	0.011 (0.008, 0.017)	0.011 (0.007, 0.016)	0.011 (0.007, 0.016)
	KPFJ	0.012 (0.009, 0.018)	0.011 (0.008, 0.017)	0.011 (0.007, 0.016)	0.011 (0.007, 0.015)

Data shown are in the format: Mean (2.5th, 97.5th percentiles) of the posterior mean estimates.



(a) Discount parameter $\delta = 0.83$: Evolution of $\hat{\sigma}^2_{v,t}$, $t = 1,\ldots,T\}$.

(b) Discount parameter $\delta = 0.95$: Evolution of $\hat{\sigma}^2_{v,t}$, $t = 1,\ldots,T\}$.

Figure 5.7: Local level model with $SNR = 0.1$: Evolution of estimated measurement noise variance $\hat{\sigma}^2_{v,t}$ (black/continuous) for all 100 MC replications, $N_p = 5000$ and the four particle filter variants under study. Corresponding 2.5th and 97.5th percentiles are represented by gray/dashed lines and the true state noise variance: $\sigma^2_v = 0.1$ is depicted by a horizontal (black/dashed) line. Each left/right panel contains for each chosen discount parameter four sub figures: top-left: LW, top-right: SIRJ, bottom-left: SIRoptJ, bottom-right: KPFJ.

Table 5.3: Evolution of estimated measurement noise variance $\hat{\sigma}^2_v$ for all 100 MC replications and the four PF variants under study with $SNR = 0.1$ and time series length $T \in \{50, 100, 150, 200\}$. True measurement noise variance: $\sigma^2_v = 0.1$.

δ	Filter	T=50	T=100	T=150	T=200
0.83	**LW**	0.105 (0.076, 0.147)	0.104 (0.081, 0.132)	0.102 (0.085, 0.124)	0.101 (0.084, 0.123)
	SIRJ	0.105 (0.075, 0.149)	0.104 (0.082, 0.131)	0.102 (0.084, 0.125)	0.102 (0.082, 0.122)
	SIRoptJ	0.105 (0.075, 0.152)	0.104 (0.081, 0.131)	0.102 (0.084, 0.124)	0.102 (0.084, 0.121)
	KPFJ	0.105 (0.075, 0.147)	0.105 (0.082, 0.136)	0.103 (0.086, 0.124)	0.102 (0.084, 0.123)
0.95	**LW**	0.105 (0.076, 0.146)	0.104 (0.080, 0.132)	0.102 (0.083, 0.123)	0.101 (0.082, 0.119)
	SIRJ	0.105 (0.075, 0.148)	0.104 (0.082, 0.132)	0.102 (0.085, 0.121)	0.101 (0.084, 0.119)
	SIRoptJ	0.105 (0.076, 0.151)	0.104 (0.082, 0.131)	0.102 (0.084, 0.123)	0.101 (0.083, 0.119)
	KPFJ	0.105 (0.076, 0.152)	0.104 (0.082, 0.132)	0.102 (0.084, 0.123)	0.101 (0.083, 0.121)

Data shown are in the format: Mean (2.5th, 97.5th percentiles) of the posterior mean estimates.

Third, we focus on exploring the impact of the signal-to-noise-ratio on the degeneracy problem which is known to be a major drawback of the particle filters. That is, analyzing the reported mean of the unique number of particles uNp (% uNp) at last time-index $t = T$, we end up with similar general conclusions to the ones arrived in Chapter 3. We consider this natural since we are using the same benchmark model (the non-stationary local level model) and also a subset of the simulation settings considered in that chapter. These findings are summarized below and hold irrespective of the discount parameter δ used:

- For the KPFJ particle filter variant, the percentage mean number of unique particles % uNp increases from about 32 to 87% as the signal-to-noise-ratio q increases from $q = 0.001$ to $q = 5$; focus on last column of Table 5.1. Of course, the higher the number of unique particles, the better; the original number of particles is $N_p = 5000$.

- For the SIRJ and LW particle filter variants, contrary to what happens with the KPF, the %uNp decreases as the signal-to-noise-ratio q increases from $q = 0.001$ to $q = 5$. Specifically, in our simulation study, the %uNp spans from about 91 down to 44% for both filters.

- For the SIRoptJ particle filter variant, a rather distinct pattern in the behavior of the unique number of particles is observed. In this situation, the %uNp first decreases from about 92 to 81% as the signal-to-noise-ratio q increases from $q = 0.001$ to $q = 0.5$. Then, the opposite happens, since we observe that the %uNp increases from about 81 to 87% as the signal-to-noise-ratio q increases from $q = 0.5$ to $q = 5$. That is, a decreasing pattern is observed on %uNp for signal-to-noise-ratio values q less than 1 and an increasing pattern for q greater than 1.

- Therefore, the simulation results confirm the degeneracy related performance already observed in Chapter 3 for the KPF, SIR, ASIR and SIRopt particle filter variants used for state estimation. We consider this a natural behavior since the KPFJ, SIRJ, LW and KPFJ are just extensions of those filters in order to tackle the simultaneous estimation of states and parameters.

- As concluded already in Chapter 3, the SIRoptJ suffers the degeneracy problem, in general, to a lesser degree, the KPFJ suffers it more at low signal-to-noise-ratio values and that both the SIR and the ASIR particle filter variants are more affected by it at high signal-to-noise-ratio values. For instance, with $N_p = 5000$ particles, even in the worst-case scenario for the SIRJ/LW particle filters that occurs at a high signal-to-noise ratio $q = 5$, we end up with about 2226 (44% of 5000) particles, which we consider big enough to produce a reliable marginal posterior distribution.

  To better illustrate the non-degeneracy of the studied particle filters, see the Figures 5.8 (page 170) and 5.9 (page 171) that were created for an exemplar run using a signal-to-noise-ratio $q = 0.1$. The first one depicts the evolution of the estimated state values x_t (black/continuous) and the 95% confidence intervals (grey/dashed) for the local level model specified by the variance parameters $\sigma_\eta^2 = 0.01$ and $\sigma_\nu^2 = 0.1$, respectively; each panel refers to a different particle filter variant.

The second plot (Figure 5.9) shows the histograms (together with the estimated posterior densities; black/ dashed) of the estimated state values and fixed variance parameters at last time-index $T = 200$. In this case, each row refers to a different particle filter variant (LW, SIRJ, SIRoptJ and KPFJ) and each column to a different estimated variable: the states (in the first column), the transition noise variance (in the second column) and the measurement noise variance (in the third column). Based on these two particular illustrations, we state that there is not a noticeable difference among the particle filters in question, as is shown (empirically) in Table 5.1 and related figures. What we confirm is that using $N_p = 5000$, we protected ourselves against the degeneracy problem, but keep in mind that for some filters, even a smaller number of particles could be used; the specific results and conclusions drawn in Chapter 3 could also be used as a guide.

Fourth, focusing on the performance of the four competing particle filter variants in terms of the computational time, we conclude that the SIRJ is the least-expensive algorithm with mean CPU time values around 3.97 seconds (average time in seconds in handling a data set containing $T = 200$ observations using $N_p = 5000$ particles), followed by the LW (4.33), the SIRoptJ (4.72) and the KPFJ with around 5.41 seconds. Of course, these results hold irrespective of the value of the discount parameter δ.

Following, a summarized acccount of obatined results is provided.

5.8.6 Final Remarks and Conclusions

Putting together all above findings we conclude that:

- The choice of the discount factor does not seem to play an important role on the estimation of the non-stationary local level model at hand. In the sequel, when dealing with this model, we use always a discount factor inside the range of values suggested by Liu and West (2001); specifically $\delta = 0.95$ as used herein. We remark, though that the use of $\delta = 0.83$ lying outside the range of values suggested in Liu and West (2001) produces similar estimation results. Therefore, as previously stated, a further Monte Carlo study must be undergone to completely rule out the potential impact of the discount factor choice.

- All three competing particle filter variants proposed have shown to a be a valid alternative to the benchmark LW particle filter variant, since they all are able to reach its statistical performance. Additionally, using a big enough number of particles, all four particle filter variants avoid the degeneracy problem.

- When we face the situation of choosing one filter over another, we recommend to also consider the computational efficiency of the filters. In such case, as stated before, the SIRJ shows the best computational performance, followed by the LW, the SIRoptJ and the KPFJ, respectively. Thus, to achieve a similar statistical performance, the SIRJ proves to be a good alternative to the well-established and widely used LW approach.

- We remark, however, that the approaches used in this work, which are based on diversifying the particles by jittering them, suffer a common criticism that can be summarized as follows: the particles are originally fixed and by jittering them, one is artificially assuming that they vary. We consider valuable, however, the trade-off between assuming an artificial noise for the parameters and the obtained quality of the estimations, since by jittering the originally fixed particles, we are able to avoid the degeneracy of the fixed parameters and at the same time to obtain a satisfactory statistical performance.

As done in Chapter 2, we end this section by providing a summary of the main features related to the historical evolution of the particle filter variants that take part in the undergone Monte Carlo studies; see Table 5.4. Notice that throughout this work we have implemented the EPFJ and UPFJ particle filter variants. For that reason, we include them in the list, regardless of the fact that they are not used in this chapter, because the model at hand, though not stationary, is linear and Gaussian. As described in Chapter 2 and shown (empirically) in Chapter 4, when filtering a nonlinear, non-Gaussian and non-stationary synthetic model, these two filters show their potentiality in case of non-standard models. As seen in case of a linear and Gaussian model, we propose to use the KPFJ particle filter variant as a particular case of the EPFJ particle filter; recall that the KPFJ combines Kalman filtering with particle filtering plus jittering.

We remark that we are aware of the existence of more recent particle filters like the particle Markov chain Monte Carlo approach introduced by Andrieu, Doucet, and Holenstein (2010). These authors combine two approaches, powerful by themselves: the SMC and the MCMC methods, whereby the former is used to construct an efficient proposal distribution which is used by the latter. For an overview of recent particle filter variants used for parameter estimation, the reader may refer to references therein.

The next chapter deals with a nonlinear dynamic model commonly used in the financial community, the so-called stochastic autoregressive volatility model of order one, SARV(1). Therein, we carry out a Monte Carlo study for the simultaneous estimation of the states and fixed model parameters, but also an application to two real data sets showing high volatility is a target: the IBEX 35 returns index and the Brent spot prices series.

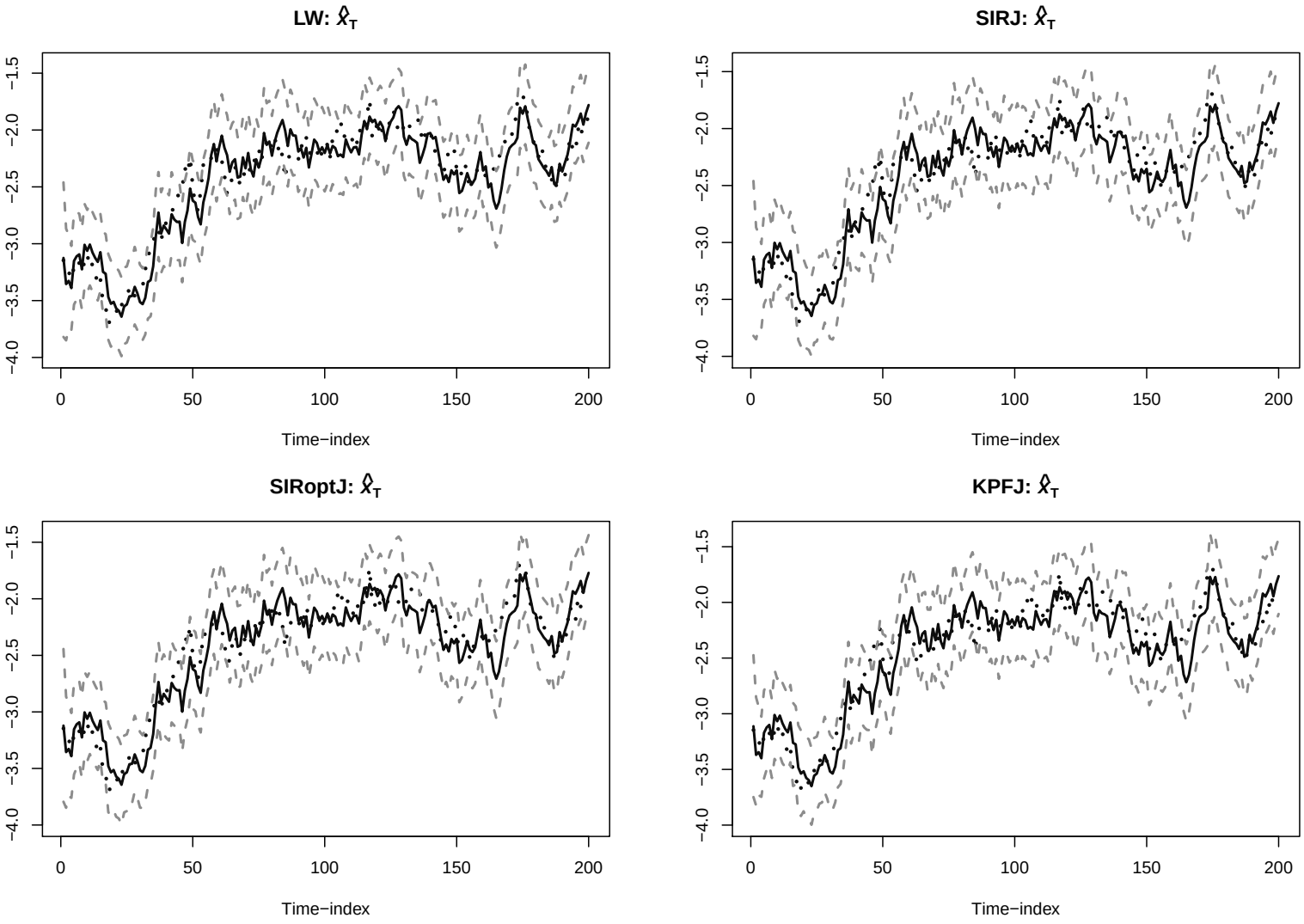


Figure 5.8: Illustration for last exemplar run: Evolution of true (black/dots) and estimated state values x_t (black/continuous) together with 95% CI (grey/dashed) for the LL model specified by σ_η^2 and σ_ν^2, respectively. Notice that each panel refers to a different PF variant. Results shown for $SNR = 0.1$.

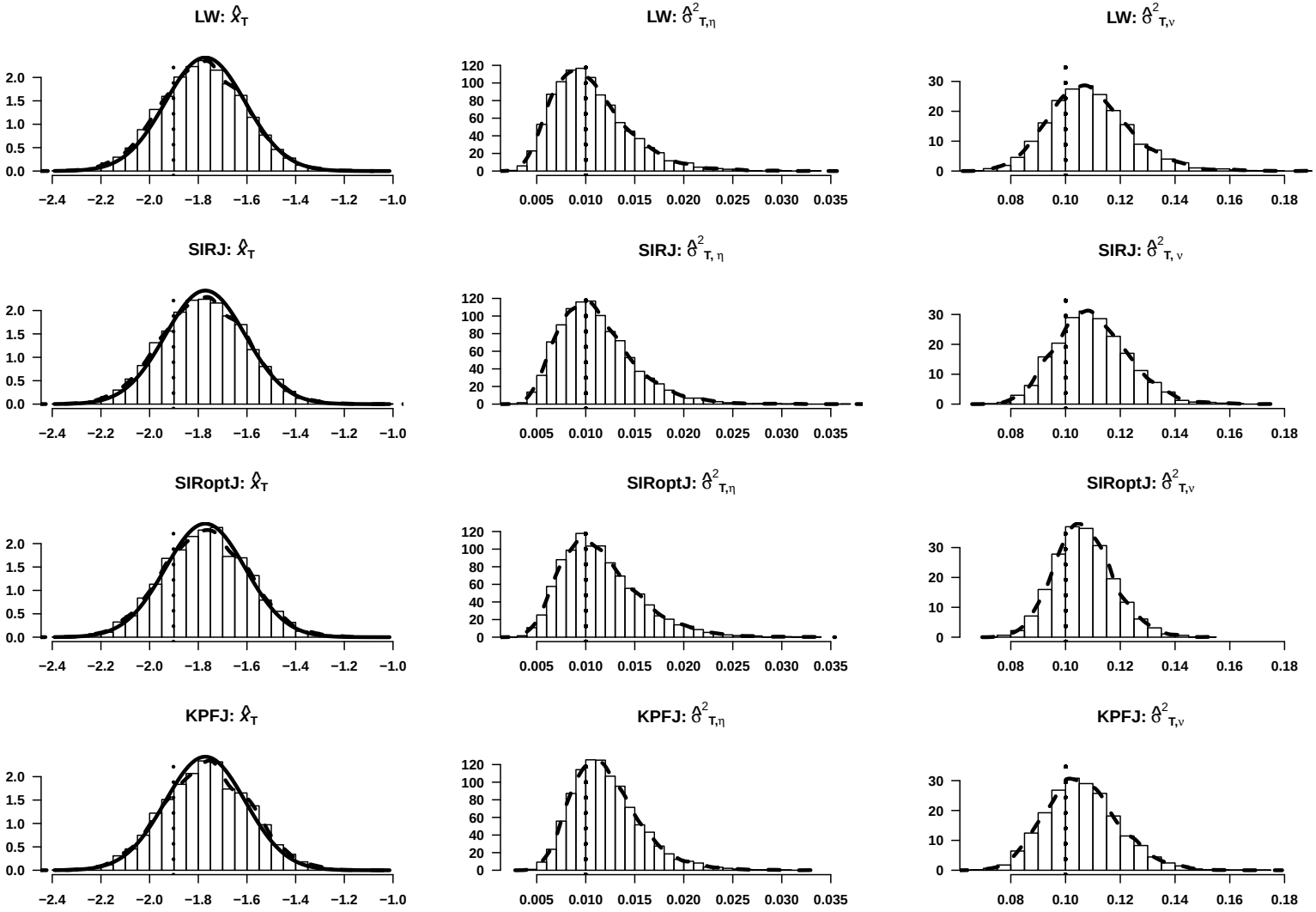


Figure 5.9: Illustration for last exemplar run and last time index $T = 200$: Histograms representing the estimated posterior distributions of: the states (first column), the system noise variance (second column) and the measurement noise variance (third columnd) for the LL model specified by σ^2_η and σ^2_ν, respectively. Notice that each row refers to a different PF variant. Results shown for $SNR = 0.1$.

Table 5.4: Historical evolution of the studied particle filters that tackle the simultaneous estimation of state and parameters. All these filters use an augmented state vector by appending the model parameters. The stratified resampling scheme is adopted, except the LW PF, which uses residual resampling.

Particle filter	Authors (year)	Stylized features
SO	Version of: Kitagawa (1998)	Performs estimation via the SIR PF variant of Kitagawa (1996), which was originally created for states estimation.
	Version of: Kitagawa and Sato (2001)	Incorporates to the SO PF of Kitagawa (1998), the artificial evolution ideas to diversify the parameters of Gordon et al. (1993).
LW	Liu and West (2001)	Combines the ASIR PF and the artificial evolution ideas to diversify the parameters of Liu and West (2001).
SIRJ	Acosta et al. (2004), Muñoz et al. (2007)	Combines the SO PF of Kitagawa (1998) and the artificial evolution ideas to diversify the parameters of Liu and West (2001).
SIRoptJ	Present work	The same as SIRJ, but it uses a fully adapted importance PDF.
EPFJ (KPFJ)	Present work	Combines the EPF(KF) and the artificial evolution ideas to diversify the parameters of Liu and West (2001). The KPFJ is a special case of EPFJ; the former uses an importance PDF based on the KF, whereas the latter is based on the EKF. The EPF remains as a theoretical proposal, whereas the KPFJ is applied in Chapter 5.
UPFJ	Present work	Combines the UPF and the artificial evolution ideas to diversify the parameters of Liu and West (2001). It remains as a theoretical proposal.

CHAPTER 6

ESTIMATION OF A STOCHASTIC VOLATILITY MODEL VIA PARTICLE FILTERING

Within Finance, one is usually interested in modeling volatile data. Particularly, filtering the volatility of financial data is crucial in financial markets (option pricing, risk management and portfolio management) to facilitate the decision–making–process (Bollerslev, Chou, and Kroner 1992).

When confronted with the analysis of financial data, the need to use non-classic time series analysis is certainly important. For instance, return time series are known to embody an added element of uncertainty due to the presence of an underlying and unobserved component called volatility; the same can be said about stock prices and exchange rates. Classic time series analysis would treat the variance for this kind of data to be constant. However, in practice –when dealing with financial data– this assumption, rarely attained, would be improper, leading to the search of alternative time series approaches.

Several stochastic volatility (SV) models have been proposed in mathematical finance and financial econometrics arriving from research made looking at different issues, see Ghysels, Harvey, and Renault (1996). For instance, the popular generalized autoregressive conditional heteroscedastic (GARCH) type models proposed by Bollerslev (1986) use an exact function to –in a deterministic way– describe the evolution of volatility. Taylor's 1986 SV models use a stochastic function to describe such evolution, see also Taylor (1994). Herein, the first type of models are solely introduced to highlight the distinguishing features of the type of models we aim to tackle: the stochastic autoregressive volatility (SARV) type models.

Herein the focus is on modeling the underlying volatility or uncertainty present in economic and financial data by the nonlinear SARV model and on estimating –via particle filtering– both the states and parameters embedded in the specified model, see Liu and West (2001) and Muñoz, Márquez, and

Acosta (2007). Knowledge of the filtered states and estimated model parameters will allow us to fully specify the dynamic nature of the process and when required, would also facilitate volatility forecasting.

This chapter is organized as follows: in Section 6.1, we present a summary of some empirical stylized facts about volatility. These stylized facts or features commonly observed in financial data such as returns time series are illustrated, via two examples, in Section 6.2. Therein, we describe the main features of two data sets containing financial time series. In Section 6.3, the general formulae and details of (G)ARCH models are outlined as introduction to SV models. Following, Sections 6.4 – 6.7 focus on the models we are more interested in: the SARV models. Specifically, Section 6.4 introduces the SARV(1) model state-space representation and alternative parameterizations. In Sections 6.5 and 6.6, we report results corresponding to two Monte Carlo studies: the first simulation undergoes volatility estimation and the second performs the simultaneous estimation of the states (volatility) and the fixed model parameters of the nonlinear SARV(1) model at hand. In both cases, we consider four different commonly found scenarios in the financial literature. As aforementioned, the estimation is achieved via the particle filtering methodology, using (when feasible) the filters already described in Chapter 2 and already applied in Chapter 5. Among those particle filters, two of them are suitable for the nonlinear model at hand: the so-called LW particle filter variant and the SIRJ; the first taken as a benchmark and the second proposed by us. The chapter concludes with an application to two real data sets –IBEX 35 return index and the price (in dollars) of Brent time series– containing highly volatile data, see Section 6.7. The aim of such applications, using the two mentioned return time series, is to evaluate and illustrate the performance of the competing particle filter variants (LW PF and SIRJ with real data sets.

6.1 Stylized Facts of Financial Returns Series

Although volatility is not directly observable, it has some characteristics that are commonly observed in financial data such as returns. These stylized facts about volatility have been well documented in the financial literature, see, for instance, Bollerslev, Engle, and Nelson (1994), Engle and Patton (2001), and Tsay (2002). Following, we summarize some of those frequently observed characteristics in financial returns:

1. Heavy or thick tails
   Since the early sixties, it is well established that asset returns have leptokurtic distributions and this feature should be presented in any volatility model, see, for instance, Mandelbrot (1963) and Fama (1965). Many models aim to capture the leptokurtic behavior by using fat-tailed distributions. Recall that the thickness of the tails of the distribution is measured by the kurtosis coefficient and that a Gaussian distribution has a kurtosis of 3. Typically very extreme non-normality is indicated by kurtosis estimates spanning from 4 to 50.

2. Asymmetric pattern of volatility
   Volatility seems to react differently to a large price increase than to a price drop of the same size. This type of behavior could be evidenced by the presence of an asymmetric effect of positive or negative shocks; refer, for instance, to Andersen et al. (2001).

3. Volatility clustering
   This is one of the first documented features of volatility (Mandelbrot 1963). Financial time series exhibit a time-varying volatility behavior reflected by high and low volatility episodes. In fact, empirical evidence suggests that large returns tend to be followed by large returns and small returns by small returns (Fama 1965). For instance, this could be evidenced by the presence of an asymmetric effect of positive or negative shocks. This asymmetric pattern can be observed by periods with large movements in prices followed by periods during which prices hardly change. Franses and Van Dijk (2000) point out that though the varying nature of volatility has been long time recognized, it is only fairly recently that explicit models reflecting the properties of volatility have been put into practice. At the moment, many models such as ARCH type (Engle 1982) and extensions, as well as SV models are designed to mimic volatility clustering. Further, Ghysels, Harvey, and Renault (1996) mention that volatility clustering and thick tails of asset returns are intimately related and that the latter is a static explanation of the former.

4. Leverage effects
   According to Ghysels, Harvey, and Renault (1996), this term was coined by Black (1976) and it suggests that in some cases stock price movements are negatively correlated with volatility. This feature, observed in financial time series such as stock prices and exchange rates, quantifies the asymmetric effect of positive or negative shocks (or news) on volatility. Indeed, since 1976 it is believed that negative shocks or news affect volatility quite differently than positive shocks of equal size. These authors point out that an increased leverage would imply more uncertainty and thus more volatility and that various empirical studies suggest that leverage alone is too small to explain the empirical asymmetries one observes in stock prices.

5. High persistence
   Volatility is highly persistent. This can be evidenced by a near to unit root behavior of the conditional variance process. In fact, when estimating stochastic volatility models, empirical evidence shows a similar pattern of high persistence; see, for instance, Jacquier, Polson, and Rossi (1994).

Other relevant features observed in squared returns are: the relatively small lower order autocorrelations and the slow decay towards zero of their autocorrelation coefficients. This would indicate a substantial dependence present in the volatility of return time series, even though serial correlation may not be present.

Another common feature present in financial data is the so-called Taylor effect first observed by Taylor (1986) and later on studied in the context of SARV models by Mora-Galan, Perez, and Ruiz (2004).

This empirical property consists in that autocorrelations of absolute returns are commonly larger than the autocorrelations of squared returns.

6.2 Illustrative Examples

To illustrate the aforementioned features, we consider two sets of real data containing daily data. The first data set consists of the Spanish financial index named IBEX 35 and the second data set of the European Brent spot prices (in Dollars per barrel). Have in mind that if P_t denotes the price of an asset at time t, the return (between time t and $t-1$) is defined as the relative variation of the index and is computed as $r_t = \log(P_t) - \log(P_{t-1})$ and then multiplied by 100%. Following, we provide a brief description of the two data sets chosen that also are used further on to validate the filters implemented in this chapter.

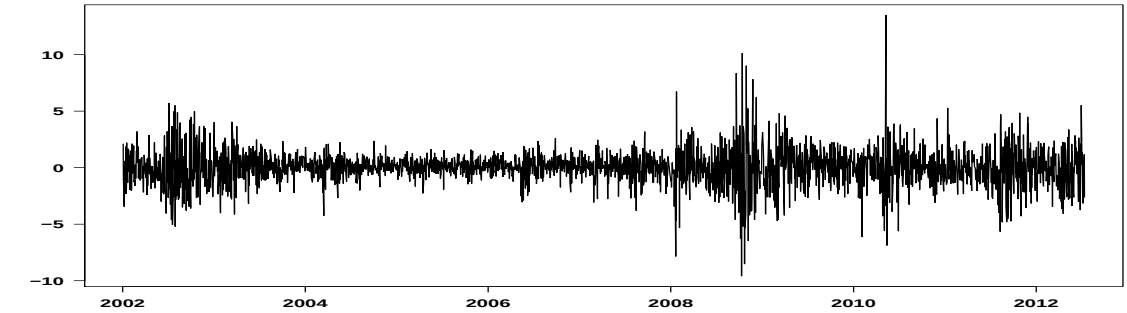
- The IBEX 35 is the official index of the Spanish Stock Exchange market. It is officially established back on 1992[1], though historical values exist since 1989, and it comprises the 35 most liquid Spanish stocks which are reviewed twice annually. We consider the daily IBEX 35 return index (now for illustrative reasons and later on for estimation) and take 2670 observations spanning from January 2, 2002 through July 12, 2012; see Figure 6.1. Notice that closing values of the index and only the days when the market was open are considered.

- The Europe Brent Spot Price is the market spot price (in US Dollars per barrel) of the so-called Brent crude oil[2]. The price of this light crude oil is used as a reference-price for other crude oils. Actually, the Brent crude oil is a blend of other crude oils produced in 15 different oil fields located in the North Sea region. In this work, we consider the returns of this daily Brent crude oil spot price and take 2669 observations spanning from January 2, 2002 through July 10, 2012; see Figure 6.2.

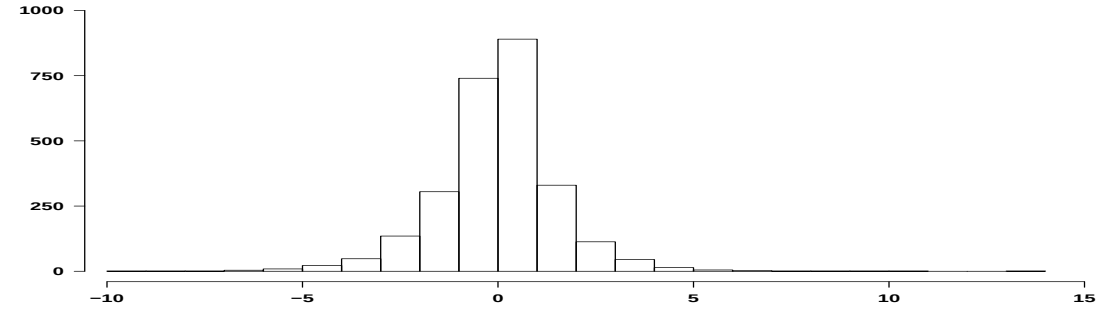
The evolution of the time series of prices and returns is depicted in panels (a) and (b) of Figures 6.1 and 6.2 for the IBEX 35 and Brent data, respectively. Additionally, a statistical description of the IBEX 35 and Brent returns is provided in Table 6.1 (on page 181), together with results of some statistical tests that when found significant at a significance level $\alpha = 0.05$, is indicated by the symbol '*'. Following, based on these plots and the entries of this table, we describe the main features found to be present in the two time series studied.

[1] `http://es.finance.yahoo.com/q/hp?s=^IBEX` [last visited: September 2013]
[2] `http://www.eia.gov` [last visited: September 2013]

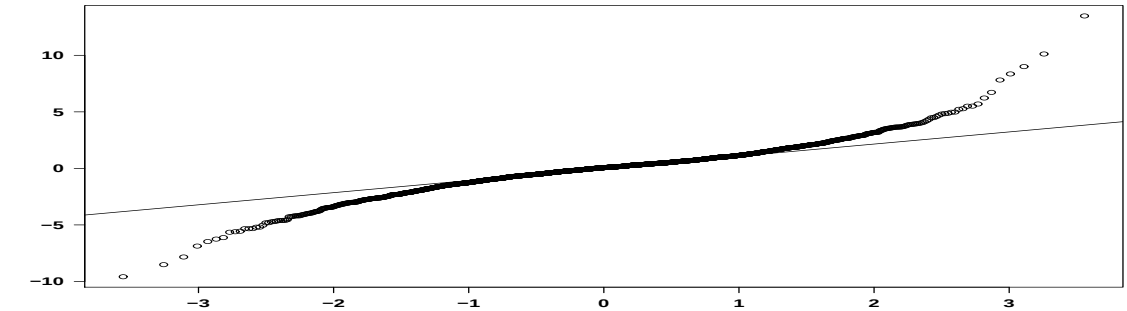
(a) IBEX 35: Original time series.



(b) IBEX 35: Return time series.



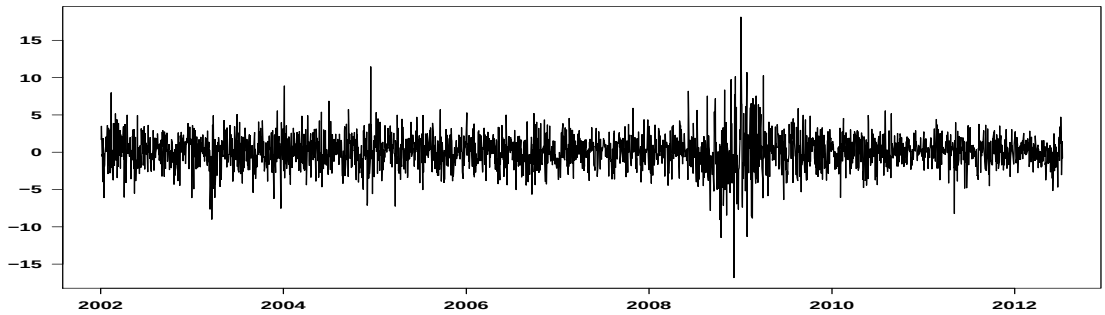
(c) IBEX 35: Histogram of returns.



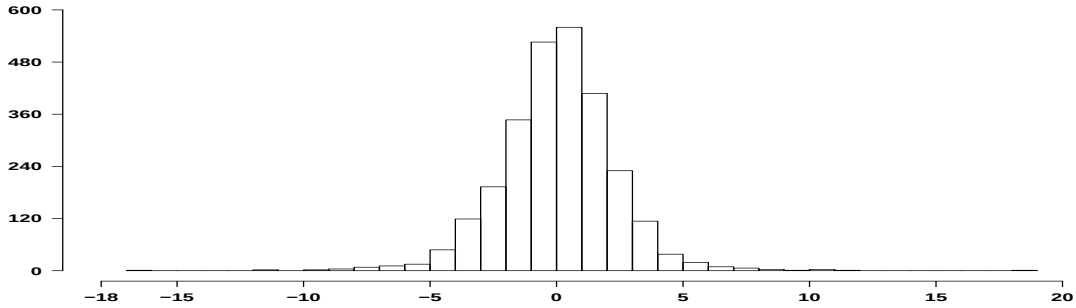
(d) IBEX 35: Normal Q-Q plot of returns.

Figure 6.1: Spanish financial index IBEX 35 (daily): (a–b) Evolution of original time series and return time series, respectively; (c–d) Histogram and Normal Q-Q plot, respectively.

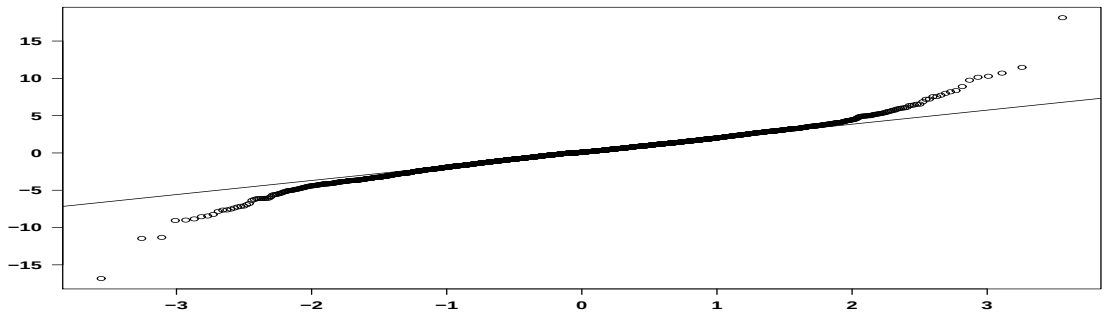
(a) Brent: Spot price time series.



(b) Brent: Return time series.



(c) Brent: Histogram of returns.



(d) Brent: Normal Q-Q plot of returns.

Figure 6.2: Europe Brent (daily, in US Dollars per barrel): (a–b) Evolution of price time series and return time series, respectively; (c–d) Histogram and Normal Q-Q plot, respectively.

For the IBEX 35 and Brent data, focusing on panel (a) of Figures 6.1 and 6.2 apparently a non-stationary (in level) behavior is present in the time evolution of daily prices. For both daily prices at hand, the presence of a unit root can be confirmed by a stationarity test like the Augmented Dickey-Fuller (ADF) test or the Phillips-Perron test, both implemented in the R package `tseries` (Trapletti and Hornik 2012); see Dickey and Fuller (1979) and Perron (1988), respectively. Indeed, the ADF test confirms the presence of a unit root (non-stationarity in level) for both mentioned daily price-series. Similarly, it is also confirmed that the two corresponding return series –though very volatile– do not have a unit root, which means that they show a stationary (in level) behavior; see panel (b) in Figures 6.1 and 6.2.

During the considered period, the computed returns range from a minimum value of -9.586% reached in October 10, 2008 and a maximum value of 13.484% corresponding to May 10, 2010. The mean value for the period reported is -0.009% with a standard deviation of 1.549%. Similarly, the computed Brent returns range from a minimum value of -16.832% reached in December 05, 2008 and a maximum value of 18.130% corresponding to January 02, 2009. The mean value for the period reported is 0.060% with a standard deviation of 2.268%; see Table 6.1.

Observe that both returns series show some of the characteristics of financial time series:

- There is a clear presence of volatility clusters in the two return series; see panel (b) in Figures 6.1 and 6.2. Notice how large returns tend to be followed by large returns and small returns by small returns. As aforementioned, this behavior can be evidenced by an asymmetric effect of positive or negative news. Observe, for instance, that in the period between 2008 to 2010, coinciding with the roots of the European debt crisis, a drastic change in prices (in the negative direction) occurs and is accompanied by a period of high volatility values of returns.

- Both IBEX 35 and Brent return series exhibit high kurtosis values that depart from the Gaussian case (Kurtosis = 3), giving rise to the phenomenon known as *thick tails*. Be reminded that Ghysels, Harvey, and Renault (1996) mention that volatility clustering and thick tails of asset returns are intimately related and that the latter is a static explanation of the former. The leptokurtic behavior is evidenced by large kurtosis values of 9.04 (significant) and 7.474 (significant) for the IBEX 35 and Brent spot return series, respectively. Additionally, the two return time series studied show pronounced peaks, which suggests the presence of observations not proper of Gaussian distributions; see corresponding histograms (panel c) and Normal Q-Q plots (panel d) in Figures 6.1 and 6.2.

- With respect to the asymmetry (focus on the histograms portrayed in panel (c) of Figures 6.1 and 6.2), only the returns of IBEX 35 exhibit a skewness value that clearly deviates from normality (Skewness = 0), whereas the returns of Brent show a very small negative value of skewness. Indeed, for the series of returns of the IBEX 35 and Brent, the Fisher skewness coefficient takes values of 0.151 (significant) and -0.029 (non-significant), respectively.

- The above results suggest that neither the IBEX 35 returns nor the Brent returns are normally distributed, which can be confirmed by the tests results reported in Table 6.1 and Normal Q-Q plots (panel d) in Figures 6.1 and 6.2.

- As typically observed in return time series, the autocorrelation function of the original observations do not show significant values (not shown), but the squared return values do exhibit significant autocorrelations; see first column of Figure 6.3 (on page 182). Observe how lower order autocorrelations are significant but also considerably small (here, below 0.3), and how the correlation coefficients decay slowly towards zero; specially in the IBEX 35. All together, that indicates that autocorrelation exists up to an extended Lag = k in both return series in consideration, which can be confirmed by the Box-Ljung test (Ljung and Box 1978). Indeed, this test allows us to reject (in both examples when applied to original returns, their squared values and their absolute values) the null hypothesis of no autocorrelation present (H_0: autocorrelations up to Lag = k are equal to zero) in studied time series: the computed Box-Ljung statistics are denoted by Q(df); in this case, df = 20 and the critical value at level $\alpha = 0.05$ is 31.4; see Table 6.1.

- According to the Taylor effect property (Taylor (1986) and Mora-Galan et al. (2004)), the autocorrelations of absolute returns are commonly larger than the autocorrelations of squared returns. Notice that the empirical property called Taylor effect is present in both illustrative examples, but specially evident in the IBEX 35; see Figure 6.3. These plots display the autocorrelation values of the squared returns (in first column) and of the absolute value of returns (in second column) for the IBEX 35 and Brent return series. In the case of the Brent return series, this property is not as evident as in the IBEX 35 return series.

Table 6.1: Summary statistics of daily returns of the Spanish IBEX 35 financial index and the Europe Brent spot price

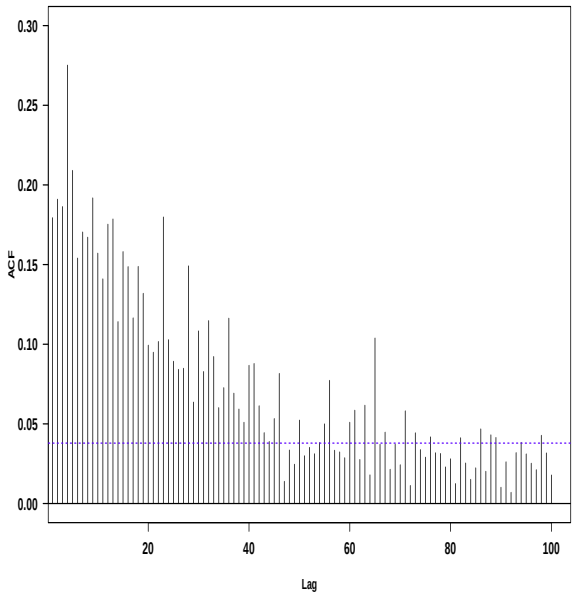
Statistics	**IBEX 35 ($n = 2670$)**	**Brent ($n = 2699$)**
Mean	−0.009	0.06
Stdev	1.549	2.268
Median	0.066	0.094
Minimum	−9.586	−16.832
Maximum	13.484	18.13
Skewness	0.151*	−0.029
Kurtosis	9.04*	7.474*
Autocorrelations r_t		
r(1)[a]	−0.001	−0.008
r(20)[a]	−0.021	−0.014
Q(20)[b]	56.407*	47.053*
Autocorrelations $\|r_t\|$		
r2(1)[c]	0.216*	0.075*
r2(2)[c]	0.279*	0.116*
r2(5)[c]	0.279*	0.121*
r2(10)[c]	0.234*	0.099*
r2(20)[c]	0.2*	0.103*
QA(20)[b]	3161.392*	703.807*
Autocorrelations r_t^2		
r2(1)[c]	0.179*	0.119*
r2(2)[c]	0.191*	0.099*
r2(5)[c]	0.209*	0.083*
r2(10)[c]	0.157*	0.097*
r2(20)[c]	0.099*	0.113*
Q2(20)[b]	1529.925*	922.903*

[a] r(k): Order k autocorrelation of return series r_t

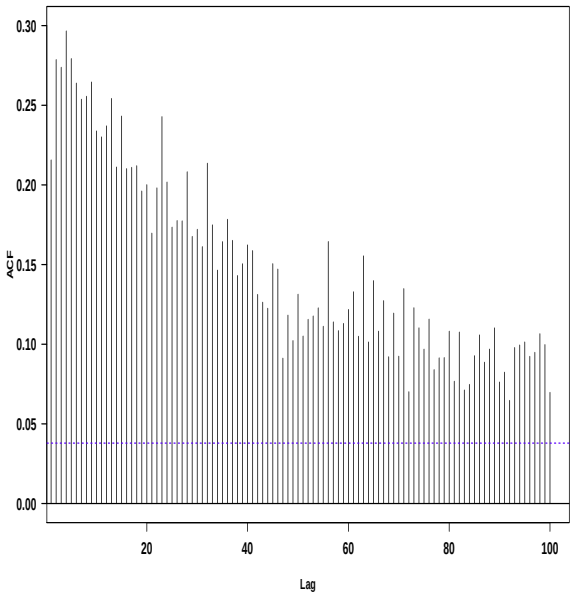
[b] Q(20), QA(20) and Q2(20): Ljung-Box statistics (Lag = 20) to test the autocorrelation of the original, absolute value and squared series: r_t, $|r_t|$ and r_t^2, respectively (critical value = 31.41)

[c] r2(k): Order k autocorrelation of squared observations r_t^2

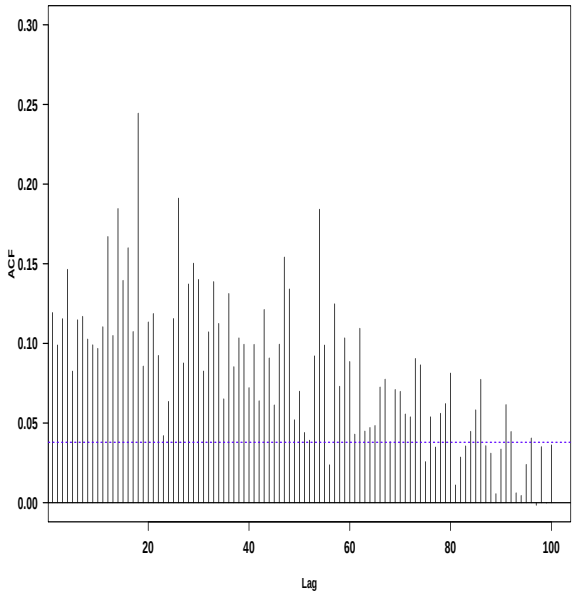
* Significant at 5% level



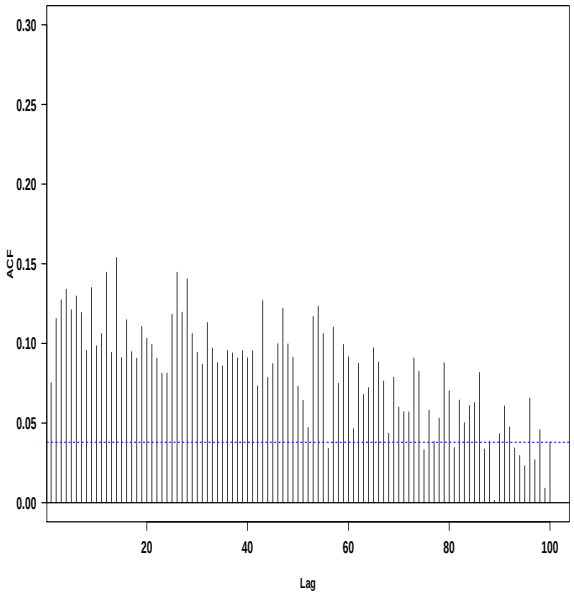
(a) Squared values of IBEX 35 returns.



(b) Absolute values of IBEX 35 returns.



(c) Squared values of Brent returns.



(d) Absolute values of Brent returns.

Figure 6.3: Autocorrelation functions of: (a–b) Spanish financial index: IBEX 35 (in euros); (c–d) Europe Brent spot returns (in US dollars per barrel).

The previous examples illustrate the features commonly present in time series which have an underlying and unknown component called volatility. With such type of data, an aim for the researcher/practitioner is to find a 'good' volatility model: one that is able to capture and reflect all or most of the aforementioned stylized facts of volatile data. Indeed, a variety of models have appeared in the literature with the aim to explain the time-evolution of volatility present in financial data such as return time series. Further, many volatility models emerge as a response to the need of improving upon existing models that are unable to reflect some of the stylized features of volatility. Following, we consider the two most popular approaches for modeling time-varying volatilities: (G)ARCH type models (Bollerslev (1986); extensive review in Bollerslev, Engle, and Nelson (1994)) and SV type models (detailed reviews in Taylor (1994); Ghysels, Harvey, and Renault (1996)).

6.3 Modeling Volatility

In this section we state the main facts about (G)ARCH type models and readily focus on the SV type models that we aim to further study.

6.3.1 (G)ARCH Type Models

In mathematical finance and financial econometrics, the most popular nonlinear models to study the behavior of return time series and its volatility are the nonlinear GARCH type models; see Bollerslev (1986). These models emerge as an extension of the ARCH type models introduced by Engle (1982). For a deeper insight into ARCH type models refer to Bollerslev, Engle, and Nelson (1994), Diebold and Lopez (1995), and Bera and Higgins (1995), among others.

The distinguishing feature of GARCH type models is that they provide a simple parametric function to deterministically describe the evolution of volatility. For instance, a GARCH(1,1) state-space formulation is given by:

$$r_t = \sigma_t \nu_t, \tag{6.1}$$

$$\sigma_t^2 = \kappa + \alpha r_{t-1}^2 + \beta \sigma_{t-1}^2, \tag{6.2}$$

where $\kappa > 0$, $\alpha > 0$ and $\beta \geq 0$ are the model parameters and where the restriction $\alpha + \beta < 1$ guarantees covariance stationarity. The positive restrictions on the parameters guarantee the positiveness of the conditional variance and the observed variable r_t is in general the return of an asset. Additionally, the measurement noise ν_t is a sequence of uncorrelated random variables with zero mean and unit variance. A common assumption for the PDF of ν_t is the standard normal one although not obligated. If normality is assumed for ν_t, it follows from (6.1) and the properties of ν_t that the conditional distribution of r_t given information up to time $t-1$ is also normal with zero mean and variance of σ_t^2. That is, the conditional expectation and variance of the variable r_t are in this case given by:

$$r_{t|t-1} = \mathrm{E}(r_t|r_{1:t-1}) = \mathrm{E}(\nu_t\sigma_t|r_{1:t-1}) = \sigma_t \mathrm{E}(\nu_t|r_{1:t-1}) = 0,$$

$$\Sigma_{r_{t|t-1}} = \mathrm{Cov}(r_t|r_{1:t-1}) = \mathrm{E}(r_t^2|r_{1:t-1}) = \mathrm{E}(\nu_t^2\sigma_t^2|r_{1:t-1}) = \sigma_t^2 \mathrm{E}(\nu_t^2|r_{1:t-1}) = \sigma_t^2.$$

Clearly, the set of two equations (6.1) and (6.2) describes the volatility as a nonlinear function of past returns. Specifically, the variance in (6.2) is a function of squared returns and the variance in the previous time period. This would imply that the (conditional) variance of r_t, given by σ_t^2, must be nonnegative and further that there is no randomness specific to the volatility process.

As previously mentioned, standard GARCH models like the GARCH(1,1) are very popular. One reason for such popularity is that they are rather easily estimated via the Maximum Likelihood approach once one has specified a distribution for the innovations; with the Gaussian distribution being a commonly used distributional assumption. Indeed, the fact that no randomness is assumed in (6.2) simplifies the estimation procedure, since then an analytical closed form of the likelihood function can first be obtained and then directly estimated via Maximum Likelihood.

Another important reason for the popularity of GARCH type models is their capability to reflect most of the main stylized facts of asset returns such as volatility clustering, pronounced excess kurtosis, or fat tails. However, these models are not able to capture other empirically relevant features of volatility such as the leverage effect (Anderson, Nam, and Vahid 1999). One possible explanation is that in these models the conditional variance depends only on the square of the returns being the sign of the returns irrelevant, thereby producing bias in volatility forecasts (Loudon, Watt, and Yadav 2000). Notice that the so-called leverage effect is related to the asymmetric pattern of volatility, evidenced by a market that reacts differently (in terms of volatility) to a large price increase than to a drop price of the same size.

Most nonlinear extensions of the GARCH model are designed to allow asymmetric patterns; see, for instance, Ding, Granger, and Engle (1993), Hamilton and Susmel (1994), Zakoian (1994), and Li and Lam (1995). An alternative to a GARCH model would be, for instance, to use a threshold model with conditional heteroscedasticity (TAR-GARCH model). In this type of models, the GARCH part of the model can describe volatility clustering and excess kurtosis (although not entirely), whereas the TAR variance formulation captures the asymmetric patterns of volatility. Moreover, volatility models that extend upon (G)ARCH aiming to capture mis-reflected features are also covered by Franses and Van Dijk (2000).

A distinct approach arises from the work of Taylor (1986, 1994), who proposed another commonly used discrete time volatility model to study the behavior of the returns and its volatility: the SV models.

6.3.2 SV Type Models

These models, like the GARCH, describe the volatility as a random process and are able to capture most of the stylized facts of asset returns, but they mainly differ due to the fact that SV models assume that there is randomness specific to the volatility process. In other words, the variance of the returns, defined as a latent variable, is a function of its past values plus a noise (Congdon 2007). The stochastic autoregressive (of order p) volatility model, SARV(p), belongs to this class of models.

Our work focuses on the univariate SARV(1) model, and not on its counterpart GARCH(1,1) model. This choice is mostly motivated by literature findings which suggest that the former shows better over-

all performance than the latter. Also, we consider that the SARV(1) model has a more appealing state-space formulation (more realistic and complex) than its alternative, the GARCH(1,1) model, due to the inclusion of randomness in the volatility process. Following, some literature findings regarding the comparison of GARCH and SV type models are presented.

SV models vs GARCH models

- The SV models are more flexible than GARCH to capture excess-kurtosis, which is a stylized fact of financial returns series; see Muñoz et al. (2004) and Carnero, Peña, and Ruiz (2001). The latter authors further point out that the additional noise process in the variance equation of a SV type model makes it much more flexible but, as a result, the likelihood function of the SV model has no closed form, making the direct maximum likelihood estimation difficult; see also Poon and Granger (2003).

- Carnero, Peña, and Ruiz (2004) also point out that both GARCH and SV models generate series with excess kurtosis and autocorrelated squares, but that the latter are better to capture such features. That is, the SV type models better reflect the empirical properties often observed in real financial time series such as returns.

- The estimation issue, though, is harder for SV type models compared to GARCH type models. As mentioned above, this is mainly due to the inclusion of the extra source of randomness in the volatility equation of the SV model. The estimation of a SV model becomes more complicated if a non-Gaussian distribution is assumed for the measure noise. Indeed, contrary to GARCH, one cannot find an analytical form for the one-step ahead forecast density for SV type models, which makes it mandatory to adopt approximative approaches, say numerical or simulation based ones. Notice however that, as previously mentioned, this extra randomness also allows us to have a model whose properties are closer to the properties of real financial time series data.

The preceding ideas make the SV type models a good alternative to reflect stylized facts commonly present in volatile data such as return time series. Let us remark, however, that this chapter does not provide a thorough coverage of possible volatility modeling approaches. We rather focus on assessing the performance of the particle filtering methodology on a commonly used univariate stochastic volatility model named stochastic first order autoregressive volatility model, SARV(1). In other words, the SARV(1) model, which is a discrete time *stochastic* nonlinear volatility model is taken as a benchmark model within the particle filtering 'framework'. For a deeper insight into other SV type models, the reader may refer, for instance, to Ghysels, Harvey, and Renault (1996) and references therein.

Following, we provide the state-space formulation of the SARV(1) model and alternative parameterizations.

6.4 The SARV(1) Model: State-Space Model Formulation

The parametric state-space formulation for general state-space models is specified by the transition and measurement equations (see equations (2.1) and (2.2) of Chapter 2) of the form

$$\begin{aligned} \boldsymbol{x}_t &= f(\boldsymbol{x}_{t-1}, \boldsymbol{\eta}_t), && \textit{(Transition equation)} \\ \boldsymbol{y}_t &= h(\boldsymbol{x}_t, \boldsymbol{\nu}_t), && \textit{(Measurement equation)} \end{aligned}$$

together with the PDF of the initial-state vector $\boldsymbol{x}_0$. This general state-space model can be also formulated in terms of the conditional distributions involved. In such case, the transition and measurement equations are specified by $\boldsymbol{x}_t|\boldsymbol{x}_{t-1} \sim p(\bullet|\boldsymbol{x}_{t-1})$ and $\boldsymbol{y}_t|\boldsymbol{x}_t \sim p(\bullet|\boldsymbol{x}_t)$, respectively.

Place yourself in the following situation: Assume that r_t (like the return of an asset) is the only observed variable at time t, but that our real interest is on the latent volatility of r_t. We adopt the basic nonlinear SARV(1) model for this kind of data and estimate the unobserved volatility and/or model parameters via particle filtering. Thus, the state-space representation of the chosen nonlinear SARV(1) model is given by the following state-transition and measurement equations:

$$x_t = \mu + \phi(x_{t-1} - \mu) + \sigma_\eta \eta_t, \qquad \textit{(Transition equation)} \tag{6.3}$$

$$r_t = \sigma_t \nu_t = \exp(x_t/2)\nu_t, \qquad \textit{(Measurement equation)} \tag{6.4}$$

where r_t is the observed variable and $x_t = \ln(\sigma_t^2)$ is a measure for the unobserved volatility of r_t, meaning that the logarithm of the conditional variance (volatility) follows an autoregressive model of order one. The noise η_t is supposed to be a sequence of uncorrelated standard normal random variables. Likewise, the measurement noise ν_t is supposed to be a sequence of uncorrelated random variables, also Gaussian. Thus, in this case, the parameter vector is $\boldsymbol{\Theta} = (\mu, \phi, \sigma_\eta^2)'$, the unconditional mean level, the degree of persistence and the uncertainty of the volatility process, respectively. Throughout this work, the state vector $\boldsymbol{x}_t$ is univariate.

Based on conditional distributions, the transition and measurement equations of the above state-space model can also be formulated as:

$$\boldsymbol{x}_t|\boldsymbol{x}_{t-1} \sim \mathcal{N}(\bullet; \mu + \phi(\boldsymbol{x}_{t-1} - \mu), \sigma_\eta^2),$$

or

$$p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \frac{1}{\sqrt{2\pi\sigma_\eta^2}} \exp\left\{ -\frac{(x_t - \mu - \phi(x_{t-1} - \mu))^2}{2\sigma_\eta^2} \right\}$$

and

$$\boldsymbol{y}_t|\boldsymbol{x}_t \sim \mathcal{N}(\bullet; 0, \exp(\boldsymbol{x}_t))$$

or

$$p(\boldsymbol{y}_t|\boldsymbol{x}_t) = \frac{1}{\sqrt{2\pi\exp(x_t)}}\exp\left\{-\frac{y_t^2}{2\exp(x_t)}\right\} = \frac{1}{\sqrt{2\pi}}\exp\left\{-\frac{x_t + y_t^2\exp(x_t)}{2}\right\}.$$

The main feature of the SARV(1) state-space formulation is that it describes a discrete-time, nonlinear dynamic system, which evolves as a first-order Markov process. Notice that in this case, the dynamic model defined by the transition equation is linear and Gaussian, but the measurement model is nonlinear. This makes unfeasible an exact inference of filtered posteriors, $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$, which will be discussed later on.

6.4.1 Alternative Parameterizations

An alternative parameterization of the transition equation (6.3) arises by defining $\varrho = \mu(1-\phi)$. In that case, (6.3) can be rewritten as

$$\begin{aligned} x_t &= \mu + \phi(x_{t-1} - \mu) + \sigma_\eta\eta_t \\ &= \mu(1-\phi) + \phi x_{t-1} + \sigma_\eta\eta_t \\ &= \varrho + \phi x_{t-1} + \sigma_\eta\eta_t \end{aligned} \tag{6.5}$$

Likewise, another parameterization of the measurement equation (6.4) can be obtained by taking the logarithm of the squared observations. In that case, equation (6.4) can be expressed as:

$$\begin{aligned} y_t &= \log(r_t^2) = \log(\sigma_t^2) + \log(v_t^2) \\ &= x_t + \epsilon_t. \end{aligned} \tag{6.6}$$

Notice that under the Gaussian distribution assumption for v_t, the new measurement noise variable $\epsilon_t = \log(v_t^2)$ has a $\log\chi_1^2$ distribution with nonzero mean and variance given by $\mathrm{E}(\log(v_t^2)) = -1.27$ and $\mathrm{Var}(\log(v_t^2)) = \pi^2/2$, respectively. Thus, the state-space formulation provided by equations (6.5) and (6.6) is now linear, but non-Gaussian. Specifically, the distribution of $\epsilon_t = \log(v_t^2)$ is given by:

$$p(\epsilon_t) = \frac{1}{\sqrt{2\pi}}\exp\left\{\frac{\epsilon_t - \exp(\epsilon_t)}{2}\right\} \tag{6.7}$$

which highly departs from a normal distribution since it has a high degree of skewness with a long left tail. Notice that ϵ_t can be forced to be zero mean by adding and subtracting the nonzero expected value of $\epsilon_t = \log(v_t^2)$, $\mathrm{E}(\log(v_t^2)) = -1.27$.

In the remainder of the chapter, unless stated otherwise, we work with the nonlinear state-space model representation specified in equations (6.3) and (6.4). Next section tackles the problem of estimating only the states of the nonlinear SARV(1) model using (whenever feasible) all particle filters already described in this work. Therein, all existing model parameters are assumed to be fixed and known and the stratified resampling scheme is adopted. Additionally, all of our findings are shown in an empirical fashion using Monte Carlo experiments, where apart from putting special effort in the

statistical and computational performance of filters, we also assess the impact of the increase of the number of particles and/or the time series length. As a way to somehow measure the degree of degeneracy present in the particle filters under study, we also report the percentage of unique particles at last time index $t = T$. We want to stress that in a paper of Andrieu, Doucet, and Holenstein (2010), in the context of particle Markov chain Monte Carlo methods, the authors suggest that the idea of defining a measure like this to account for particles' degeneracy is correct. Indeed, these authors point out the following:

> "Assessing path degeneracy is certainly essential to evaluate the credibility of the results. A simple proxy to measure degeneracy consists of monitoring the number of distinct particles representing $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:T})$ for various values $k \in \{1,\ldots,T\}$ (preferably low values). If this number is below a reasonable number, say 500, then the particle approximation of $p(\theta,\boldsymbol{x}_{1:T}|\boldsymbol{y}_{1:T})$ is most probably unreliable."

6.5 Simulation Study I: Estimation of the states of the Nonlinear SARV(1) Model

In contraposition to Chapters 3 and 4 dealing with dynamic linear models as a benchmark, the present section deals with a non-standard benchmark model: the SARV(1) model which is nonlinear and non-stationary as specified in equations (6.3) and (6.4).

In this nonlinear context, the traditional Kalman filter does not provide an optimal solution to filter the states as it does in case of linear-Gaussian state-space models. Indeed, this nonlinearity makes unfeasible the computation of exact posteriors in the SARV(1) model. Further, the use of any Kalman based approximation presented in this work (EKF and UKF) to linearize a dynamic model is not suitable. The un-suitability of the mentioned approximative Kalman based approaches relies on the fact that the resulting Kalman Gain is null, meaning that the states are never really updated and the information contained in new observations is discarded. This finding (empirical results not shown here) is in line with Zoeter, Ypma, and Heskes (2004) who state that "in stochastic stock volatility models the traditional unscented Kalman filter is ill suited and it can be proven that the traditional filter effectively never updates prior beliefs".

The un-suitability of the Kalman based approaches presented in this work implies, unfortunately, a natural consequence: A direct implementation of Kalman based filters is not always feasible and/or adequate. That is, with the nonlinear model at hand, none of the Kalman based particle filters studied (in the form presented here) are of real use. Thus, in the present section, we only assess, via a Monte Carlo study, the filtering performance of three of the studied classic particle filter variants: SIS, SIR and ASIR. Notice that the SIS algorithm is known to be non-operational but is included in the benchmark study merely for illustrative reasons; the ASIR PF variant is taken as a benchmark filter.

Before continuing, we want to remark that though equations (6.5) and (6.6) specify a linear model, it is clearly non-Gaussian. For this log-linearized non-Gaussian SARV(1) model, the traditional Kalman

filter does not yield an optimal solution (results not shown). As stated in Chapter 2, Kalman filters based on the normality assumption are known to be non–robust, which implies that the posterior density may become unrealistic (Meinhold and Singpurwalla 1989).

Following, we carry out a Monte Carlo study to assess the statistical and computational performance of three competing particle filter variants (SIS, SIR and ASIR) when estimating solely the states of the nonlinear Gaussian SARV(1) model specified in equations (6.3) and (6.4).

6.5.1 Simulation Study I: Design and Simulation Settings

In this first simulation study, our aim is to obtain the marginal posterior probability density function of the states, $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$ via particle filtering. We remark that under the assumption of normality of the measurement disturbance v_t in the nonlinear state-space model specified in equations (6.3) and (6.4), the adoption of the particle filtering approach would lead to the same filtering estimates as the ones obtained with the alternative linear (log-linearized) state-space model specified in equations (6.5) and (6.6) that uses the truly non-Gaussian distribution of the corresponding measurement disturbance ϵ_t defined in equation (6.7)[3].

As aforementioned, in this work we use the SARV(1) nonlinear state-space formulation given by equations (6.3) and (6.4) with latent states $x_t = \log(\sigma_t^2)$ and where both random disturbances v_t and η_t follow a normal distribution. Specifically, it is assumed that $v_t \sim \mathcal{N}(0,1)$ and $\eta_t \sim \mathcal{N}(0,1)$. Thus, herein the aim is to estimate only the states (volatility) assuming that the fixed parameter vector $\boldsymbol{\Theta} = (\mu, \phi, \sigma_\eta^2)'$ is known.

Simulation Design

Herein, we adopt the same simulation procedure for filtering the states as described in Section 3.2 of Chapter 3 (also used in Chapter 4), but adapted to the nonlinear model at hand. Be reminded that this procedure involves three general steps: STEP I: Data and state generation; STEP II: Filtering the states; and STEP III: Filtering performance criteria computation. As known, the statistical performance of the filters is based on the root mean square error (RMSE) criterion and their computational performance on the elapsed CPU time, respectively. As specified in Chapter 3 the statistical performance of the filters is explicitly defined in terms of the mean and variance of the RMSE, computed with equations (3.6) and (3.7), respectively. Likewise, the computational performance is measured by the mean elapsed CPU time, computed with equation (3.8). Additionally, the degree of degeneracy is assessed by providing the average (%) number of unique particles at last time-index $t = T$, %uNp.

Next, we provide the specific filter settings, experimental results, remarks, and conclusions for this first simulation study.

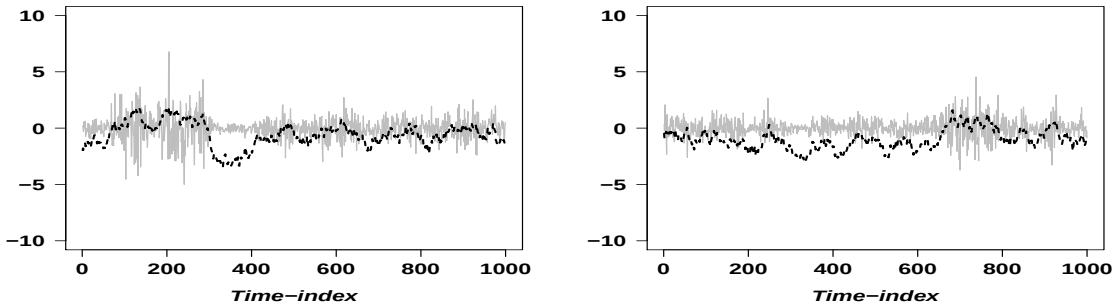
[3]Some authors adopt the alternative parametrization of the state-space model using a Gaussian approximation of the noise measurement ϵ_t, using $\mathcal{N}(-1.27, \pi^2/2)$; see Ruiz (1994). Also, a mixture of Gaussian distributions has been proposed to approximate the truly non-Gaussian distribution of ϵ_t; see, for instance, Chib, Nardari, and Shephard (2002) who use a mixture of seven Gaussian distributions.

Simulation Settings

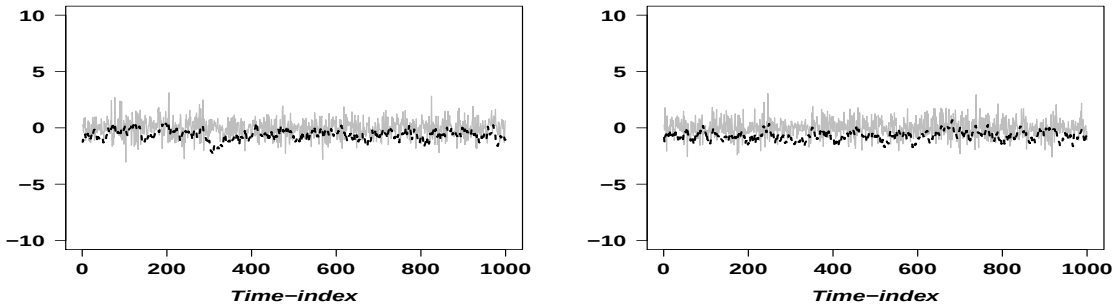
A summary list of the simulation settings used for the conducted Monte Carlo experiment is presented below:

- **Filters**: SIS, SIR and ASIR (Kalman based filters not feasible in this case). The SIS algorithm is included for the sake of illustration only.
- **Parameters settings**: The parameter μ is fixed to -0.632, the persistence parameter takes two possible values ($\phi \in \{0.9, 0.981\}$) and the uncertainty (volatility) of the volatility parameter takes also two possible values ($\sigma_\eta^2 \in \{0.194^2, 0.363^2\} = \{0.038, 0.132\}$); the chosen values respond to those typically found in literature. Thus, four different scenarios are defined by the two chosen ϕ and σ^2 settings giving raise to the four cases listed below. Keep in mind that in this first MC study, all parameters are assumed to be fixed and known.
- **Cases**: The four simulation cases defined by the chosen values of parameters $\Theta = (\mu, \phi, \sigma_\eta^2)'$ are:
  - Case 1: $\Theta = (\mu, \phi, \sigma_\eta^2)' = (-0.632, 0.981, 0.194^2)$; $0.194^2 = 0.038$.
  - Case 2: $\Theta = (\mu, \phi, \sigma_\eta^2)' = (-0.632, 0.9, 0.194^2)$; $0.194^2 = 0.038$.
  - Case 3: $\Theta = (\mu, \phi, \sigma_\eta^2)' = (-0.632, 0.981, 0.363^2)$; $0.363^2 = 0.132$.
  - Case 4: $\Theta = (\mu, \phi, \sigma_\eta^2)' = (-0.632, 0.9, 0.363^2)$; $0.363^2 = 0.132$.
- **Resampling scheme**: Stratified resampling
- **Number of replications**: $S = 100$
- **Number of particles**: $N_p \in \{200, 500, 1000, 5000\}$. Notice that in previous chapters $N_p = 5000$ particles is found to provide satisfactory estimation performance for most particle filter variants. However, we consider some intermediate values to illustrate and empirically assess the impact of increasing the number of particles when dealing with the nonlinear SARV(1) model.
- **Time series length**: $T \in \{500, 1000, 2000\}$. Herein, we aim to assess the impact of the time series length on the performance of the PF variants under study.
- **Comparison criteria**: RMSE and CPU time. Additionally, the average number (%uNp) of distinct particles at time-index $t = T$ are computed for the SIR and ASIR PF variants that use a resampling scheme. For the SIS PF variant, that does not include a resampling step, the computation of this measure does not make sense. As expected, if this measure (percentage) is computed it would be equal to 100%, but in such case most of the particles though distinct will have negligible weights.

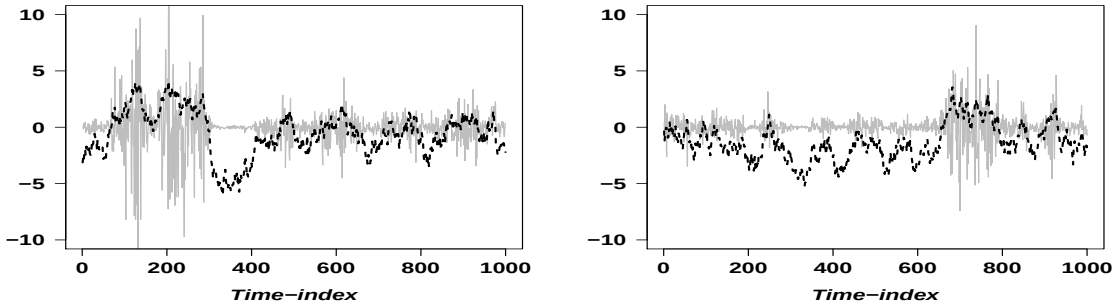
For illustrative purposes, a graphical representation of the generated univariate data y_t and corresponding state values x_t is displayed in Figure 6.4.



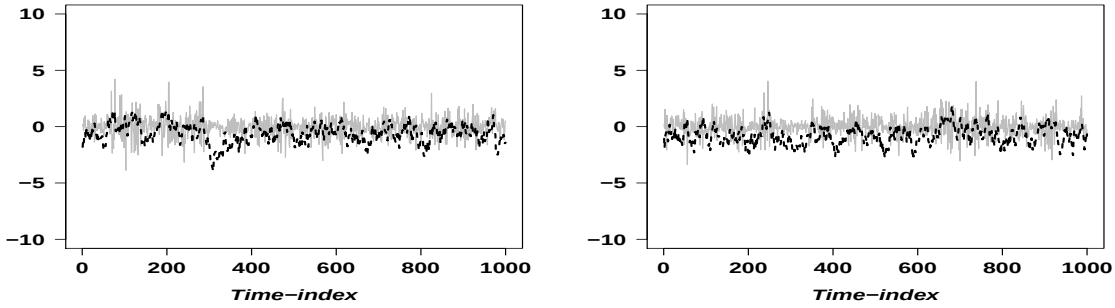
(a) Case 1: $\Theta = (\mu, \phi, \sigma_\eta^2)' = (-0.632, 0.981, 0.194^2)$; $0.194^2 = 0.038$



(b) Case 2: $\Theta = (\mu, \phi, \sigma_\eta^2)' = (-0.632, 0.9, 0.194^2)$; $0.194^2 = 0.038$



(c) Case 3: $\Theta = (\mu, \phi, \sigma_\eta^2)' = (-0.632, 0.981, 0.363^2)$; $0.363^2 = 0.132$



(d) Case 4: $\Theta = (\mu, \phi, \sigma_\eta^2)' = (-0.632, 0.9, 0.363^2)$; $0.363^2 = 0.132$

Figure 6.4: SARV(1) model: Two exemplary runs of the generated data $\boldsymbol{y}_t$ (grey/continuous) and simulated states $\boldsymbol{x}_t$ (black/dashed) for each of the four cases under study.

The plots in Figure 6.4 (each row corresponding to each one of the four case-scenarios; two exemplary runs per case) illustrate the non-stationary character of the nonlinear SARV(1) model. The generated states (volatility) x_t show a time varying behavior but, as expected, revert to a mean level according to a specific degree of persistence given by the parameter ϕ. Observe that, in panels (a) and (c), corresponding to a near non-stationary behavior, the volatility process reverts to the mean in a slower manner than when $\phi = 0.9$ (panels (b) and (d)). Also, there is a clear presence of volatility clusters (small changes and large changes are clustered together) in all of the displayed time series; this is specially portrayed in panels (a) and (c). The general pattern observed in returns and latent states seems to be mostly driven by the persistence parameter ϕ, whereas the degree of dispersion observed seems to be mostly driven by the value of the state noise variance σ_η^2.

Following, simulation results, remarks, and conclusions regarding Simulation Study I are presented.

6.5.2 Simulation Study I: Experimental Results

In Table 6.2 (complemented with Tables C.1–C.3 in Appendix C), we provide the numeric results which summarize the performance of the three filters studied in handling the estimation of the states (volatility) for the nonlinear SARV(1) model with known model parameters. Each table is organized in four vertical blocks corresponding to each of the four settings for the number of particles. Horizontally, we find three blocks corresponding to each of the filters under study. For each filter and three time series length settings, we report average and variability measures of the estimated RMSE and the percentage of the unique number of particles (%uNp) at last time-index $t = T$; for the unique number of particles, the standard deviation and not the variance is reported [4]. Notice that for the SIS filter we do not report %uNp, because its computation only makes sense for filters that adopt the resampling step: in this case the SIR and ASIR PF variants for which such percentage measures somehow the degree of degeneracy.

Additionally, in Table 6.2 corresponding to Case 1, we report the mean and standard deviation of estimated CPU times (in seconds); these numeric values are about the same in the other three case-scenarios and are thus not reported in complementary Tables C.1–C.3 in Appendix C. Be reminded that the estimated mean-CPU-elapsed time is defined as the average time (in seconds) in handling a data set containing T observations, using N_p particles.

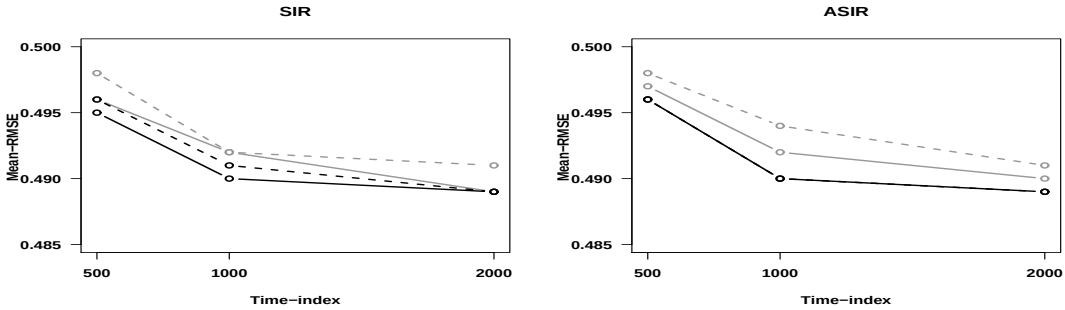
To aid in the discussion of simulation results reported in the previously mentioned tables (Table 6.2 and Tables C.1–C.3 in Appendix C), we create pictorial representations that allow us to have, at a glance, a very good idea of the main findings under this first Monte Carlo study. Specifically, in order to assess the statistical performance (in terms of the RMSE) of the three filters under study, we construct Figure 6.5 that depicts the mean-RMSE attained at the three chosen time series length settings and at the four settings used for the number of particles. The attained mean-RMSE estimates are displayed for each of the four case scenarios under scrutiny, which are defined by the chosen model parameters' values. Have in mind that in simulation study I, the ASIR is our benchmark particle filter variant.

[4] Although all displayed simulation results are rounded up to three decimal points, the reported %uNp values are rounded up to two decimal points.

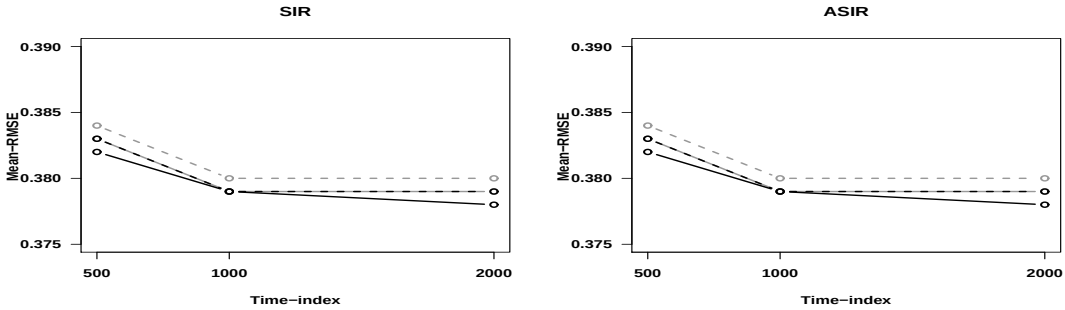
Table 6.2: Summary of simulation I results for case 1: Estimation of the states (volatility) for the SARV(1) model; $\Theta = (\mu, \phi, \sigma_\eta^2)' = (-0.632, 0.981, 0.194^2)$; $0.194^2 = 0.038$.

			$N_p = 200$		$N_p = 500$		$N_p = 1000$		$N_p = 5000$	
Filter	**T**	**Criterion**	**Mean**	**Var**	**Mean**	**Var**	**Mean**	**Var**	**Mean**	**Var**
SIS	**500**	**RMSE**	0.848	0.023	0.806	0.013	0.786	0.014	0.713	0.010
		CPU	0.200	0.014	0.311	0.018	0.503	0.026	2.179	0.075
	1000	**RMSE**	0.986	0.014	0.955	0.012	0.906	0.010	0.853	0.008
		CPU	0.407	0.036	0.619	0.026	1.016	0.033	4.385	0.119
	2000	**RMSE**	1.087	0.008	1.069	0.012	1.043	0.008	0.983	0.007
		CPU	0.809	0.025	1.249	0.035	2.033	0.133	8.911	0.151
SIR	**500**	**RMSE**	0.498	0.002	0.496	0.002	0.496	0.002	0.495	0.002
		%uNp	89.42	8.32	89.55	8.23	89.47	8.37	89.43	8.36
		CPU	0.407	0.021	0.581	0.026	0.875	0.031	3.637	0.687
	1000	**RMSE**	0.492	0.001	0.492	0.001	0.491	0.001	0.490	0.001
		%uNp	91.38	6.67	91.42	6.66	91.41	6.53	91.42	6.51
		CPU	0.806	0.025	1.158	0.034	1.775	0.034	7.592	0.830
	2000	**RMSE**	0.491	5e-04	0.489	5e-04	0.489	5e-04	0.489	5e-04
		%uNp	91.27	5.75	91.19	5.74	91.16	5.70	91.15	5.72
		CPU	1.629	0.039	2.323	0.043	3.457	0.031	14.011	0.148
ASIR	**500**	**RMSE**	0.498	0.002	0.497	0.002	0.496	0.002	0.496	0.002
		%uNp	95.93	3.09	96.11	3.14	96.12	30.03	96.07	2.84
		CPU	0.680	0.030	1.009	0.034	1.580	0.031	7.783	0.06
	1000	**RMSE**	0.494	0.001	0.492	0.001	0.490	0.001	0.490	0.001
		%uNp	96.73	2.47	96.78	2.22	96.83	2.26	96.82	2.11
		CPU	1.346	0.038	2.019	0.032	3.277	0.111	15.772	0.14
	2000	**RMSE**	0.491	5e-04	0.490	5e-04	0.489	5e-04	0.489	5e-04
		%uNp	96.64	2.36	96.68	2.10	96.73	2.00	96.70	2.09
		CPU	2.689	0.041	4.067	0.820	6.396	0.103	27.289	0.460

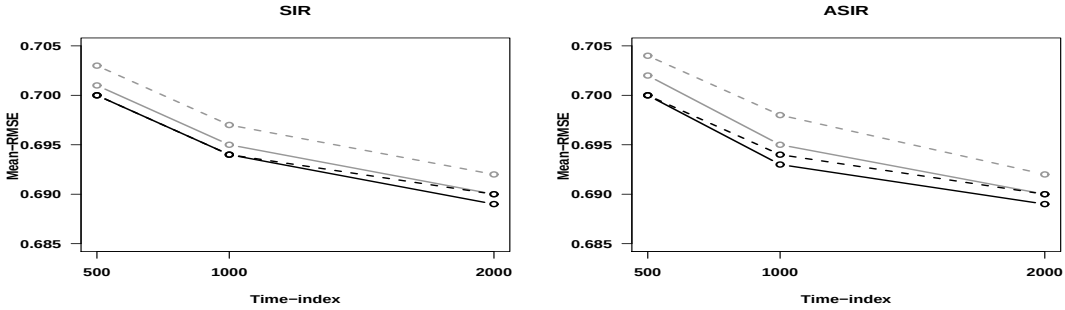
Likewise, Figure 6.8 (on page 200) accounts for the computational performance of the filters in consideration; we report values for one case (Case 1 in Table 6.2) since similar numeric values were obtained in the other three scenarios. Additionally, we construct Figure 6.7 (page 197) to reflect the degree of degeneracy present in each particle filter under study and at the same time to assess the combined impact of the number of particles and the time series length on degeneracy.



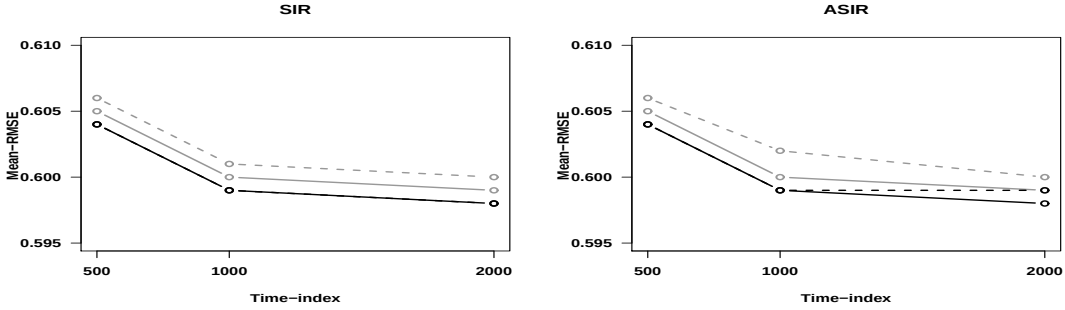
(a) Case 1: $\Theta = (\mu, \phi, \sigma_\eta^2)' = (-0.632, 0.981, 0.194^2)$; $0.194^2 = 0.038$



(b) Case 2: $\Theta = (\mu, \phi, \sigma_\eta^2)' = (-0.632, 0.9, 0.194^2)$; $0.194^2 = 0.038$



(c) Case 3: $\Theta = (\mu, \phi, \sigma_\eta^2)' = (-0.632, 0.981, 0.363^2)$; $0.363^2 = 0.132$



(d) Case 4: $\Theta = (\mu, \phi, \sigma_\eta^2)' = (-0.632, 0.9, 0.363^2)$; $0.363^2 = 0.132$

Figure 6.5: SARV(1) model: Behavior of estimated mean-RMSE for the SIR and ASIR PF variants. Assessment of the impact of the time series length (x-axis) and the number of particles ($N_p = 200$: grey/dashed; $N_p = 500$: grey/continuous; $N_p = 1000$: black/dashed, and $N_p = 5000$: black/continuous). Results shown for Cases 1–4 (SIS results are –as expected– worse and, therefore, not shown here).

6.5.3 Simulation Study I: Remarks and Conclusions

Based on simulation results reported in the above-mentioned tables and depicted on Figures 6.5–6.7, we make the following remarks and conclusions regarding the performance of the SIS, SIR and ASIR filters when handling the estimation of the states (volatility) of the nonlinear SARV(1) model with known model parameters:

First, for the nonlinear SARV(1) model at hand, we refer to the statistical performance, indicated by the mean-RMSE[5], of the three studied PF variants.

- The SIS filter, as expected, shows the worst statistical performance with respect to the competing SIR and ASIR particle filters; this holds in all four cases (results shown in Table 6.2 and Tables C.1–C.3 in Appendix C) under study irrespective of the number of particles and time series lengths used in the estimation procedure; refer to the above mentioned tables to visualize SIS RMSE values (we only graphically represent SIR and ASIR results). Indeed, in all cases, the SIS particle filter diverges showing mean-RMSE values that increase over time. This confirms, once again, that this filter is not operational since it produces a non-reliable posterior where almost all particles have negligible weights. Next, we comment on the results found for the SIR and ASIR PF variants that use a resampling scheme and are known to be operational.

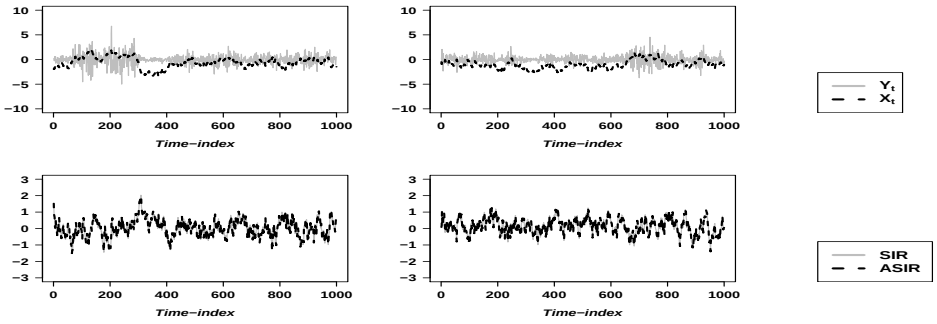
- Given a fixed T and N_p, the SIR PF variant attains (practically) the same statistical performance of the benchmark ASIR PF variant; focus on Figure 6.5. Notice that the differences observed between filters is rather minimal, with ASIR's RMSE values slightly above the corresponding SIR's RMSE.

- To study the impact of increasing the number of particle N_p or the time series length T on the statistical performance of the filters, we consider four settings for $N_p \in \{200, 500, 1000, 5000\}$ and three settings for $T \in \{500, 1000, 2000\}$. We find that, in all case scenarios:

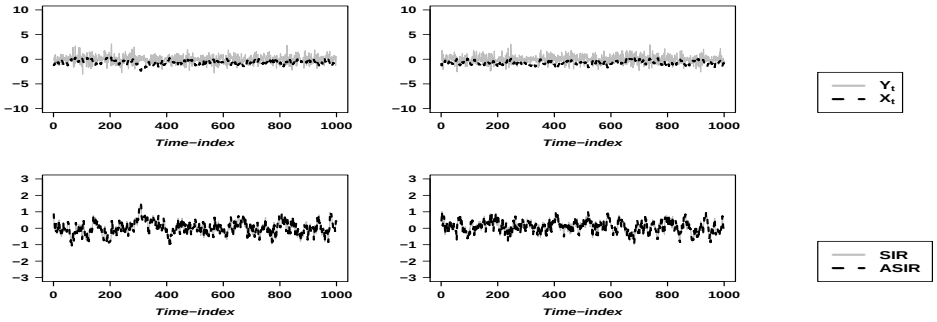
  - For a fixed number of particles N_p, the mean-RMSE of the SIR and ASIR particle filters decreases (though slightly) with the increase of the time series length.
  - For a fixed time series length T, the mean-RMSE of the SIR and ASIR particle filters tend to attain smaller values when using a larger number of particles; the differences observed are rather minimal (generally in the third decimal place).

- To better illustrate how the SIR and the ASIR PF variants yield similar statistical performance, see Figure 6.6 that was created for an exemplary run with time series length $T = 1000$, using $N_p = 5000$ particles. This figure, for each case scenario, represents the differences between estimated and true-state values $\hat{x}_{t|t} - x_t$, $t = 1, \ldots, T$, which are found to be indistinguishable between filters; focus on second row of each subfigure. Clearly, the difference between estimated and true-state values, $\hat{x}_{t|t} - x_t$, becomes larger in those scenarios with higher system noise variance.

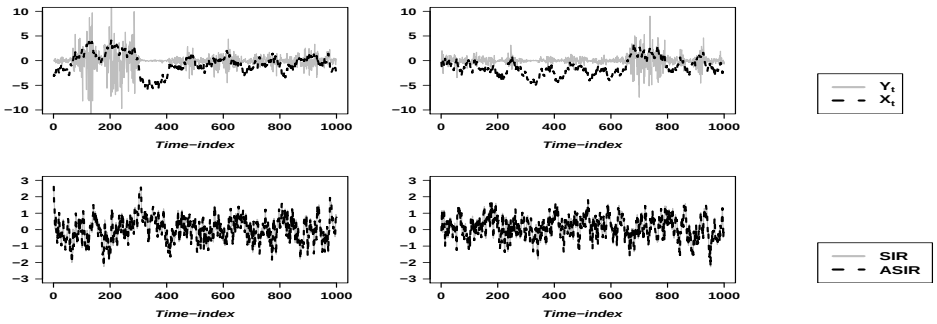
[5] Remind that Mean-RMSE and Mean(RMSE) are used interchangeably



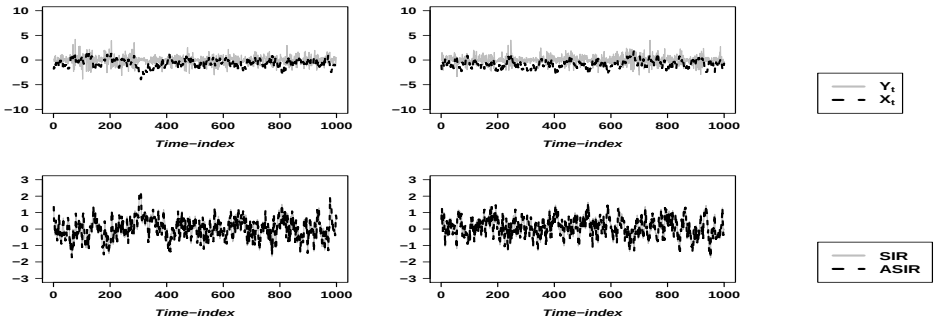
(a) Case 1: $\Theta = (\mu, \phi, \sigma_\eta^2)' = (-0.632, 0.981, 0.194^2)$; $0.194^2 = 0.038$



(b) Case 2: $\Theta = (\mu, \phi, \sigma_\eta^2)' = (-0.632, 0.9,\ 0.194^2)$; $0.194^2 = 0.038$



(c) Case 3: $\Theta = (\mu, \phi, \sigma_\eta^2)' = (-0.632, 0.981, 0.363^2)$; $0.363^2 = 0.132$



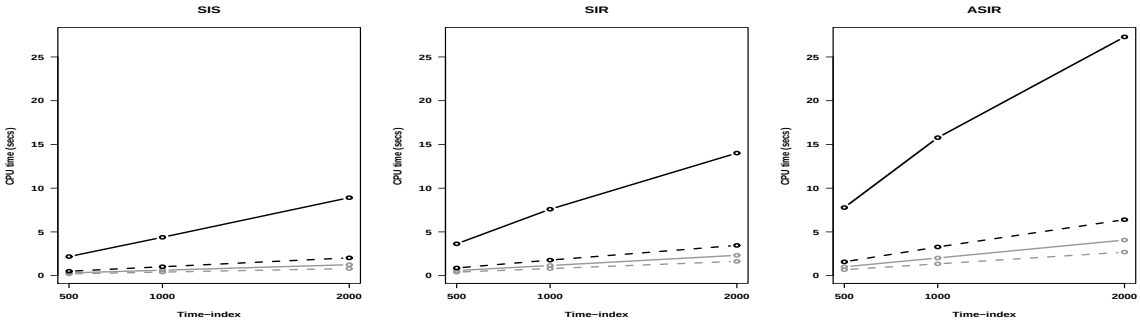
(d) Case 4: $\Theta = (\mu, \phi, \sigma_\eta^2)' = (-0.632, 0.9, 0.363^2)$; $0.363^2 = 0.132$

Figure 6.6: SARV(1) model: For each case (a)–(d), the first row represents the generated observations and states. The second row, the difference between estimated and true-state values $\hat{x}_{t|t} - x_t$, $t = 1, \ldots, T$. Results shown for $T = 1000$ and $N_p = 5000$.

- Thus, in all four scenarios under scrutiny, when estimating the states of the nonlinear SARV(1) model, the SIR achieves the same (sometimes slightly better) statistical performance than the benchmark ASIR PF. Additionally, larger mean-RMSE values are attained in those scenarios combining higher state noise variance values with higher persistence values, seemingly in that order. The non operational SIS PF, included here for illustrative reasons, is not considered in the next section and thereafter.

Second, for the nonlinear SARV(1) model at hand, focusing on the performance of the different filters in terms of the computational time, we conclude that:

- The non operational SIS filter is the least expensive algorithm with mean CPU time values around 4.39[0.12] (average[SD] time in seconds in handling a data set containing $T = 1000$ observations using $N_p = 5000$ particles), followed by the SIR (7.59[0.83]) and the ASIR (15.77[0.14]) filter. Similar results hold, as expected, in the other three case scenarios (observed slight variations in CPU times are characteristic of the simulation character of the MC studies). Further, given a fixed T and N_p, we find that the SIR computationally outperforms the ASIR, especially at larger values of the number of particles (N_p) and the time series length (T); see Figure 6.7 and refer to Table 6.2 to visualize results.



(a) Case 1: $\Theta = (\mu, \phi, \sigma_\eta^2)' = (-0.632, 0.981, 0.194^2)$; $0.194^2 = 0.038$

Figure 6.7: SARV(1) model: Behavior of the estimated mean-CPU-elapsed time in seconds for the SIS, SIR, and ASIR PF variants. Assessment of the impact of the time series length (x-axis) and the number of particles ($N_p = 200$: grey/dashed; $N_p = 500$: grey/continuous; $N_p = 1000$: black/dashed and $N_p = 5000$: black/continuous). Results shown for Case 1 but representative of all cases 1–4.

- With the aim of answering the question of how the increase of the number of particles and/or how the use of a larger size of the time series influences the computational performance of the filters, we consider the introduction of four settings for the number of particles and three settings for the time series length into the MC experiments; see values listed on page 193. We find that, in all four case scenarios:
  - For fixed number of particles N_p, the computational cost given by the CPU-elapsed times increases over time in a seemingly linear fashion.

  - For fixed time series length T, the computational cost given by the CPU-elapsed times also increases when increasing the number of particles N_p, but in a larger manner in case of the ASIR.

- Thus, in all four scenarios under scrutiny, when estimating the states of the nonlinear SARV(1) model, the SIR outperforms the computational performance of the benchmark ASIR PF. Additionally, it seems that the CPU discrepancies found among filters are greater at larger values of the number of particles (N_p) and time series length (T); especially when using a larger number of particles. The non-operational SIS PF, shown here for illustrative reasons, is not considered in the next section and thereafter.

Finally, we refer to the complementary study (found to be relevant) to investigate and somehow quantify the degree of degeneracy present in the SIR and ASIR PF variants when handling the state estimation of the nonlinear SARV(1) model; focus on Figure 6.8. We also analyze the impact of increasing N_p and T on degeneracy. That is, analyzing the reported percentage mean of the unique number of particles uNp (%uNp) at last time-index $t = T$, we conclude the following:

- As aforementioned, the non-operational SIS algorithm would always yield %uNp= 100%, but the computation of this percentage only makes sense for filters that include a resampling step. As theory indicates and confirmed in the simulation studies, under the non-operational SIS filter negligible weights are carried out from time to time. Consequently, we may end up with few or even only one particle with non-negligible weights, which beside many times are located in the wrong region of the state-space. Next, we comment on the results found which correspond to the operational SIR and ASIR PF variants.

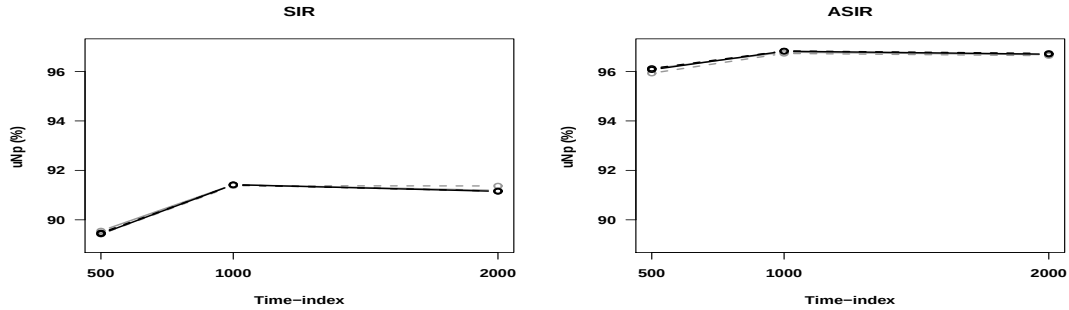
- In general, compared to the SIR PF variant, the percentage of distinct number of particles, %uNp, is higher for the benchmark ASIR algorithm. Overall the simulation settings, irrespective of the case scenario, the percentage mean number of unique particles for the SIR spans from around 84%–94%, and the benchmark ASIR from around 92%–97%.

- For fixed time series length T, regardless of the number of particles used in the estimation procedure, the percentage mean of unique particle seemingly remains stable, but the "absolute" number of unique particles is clearly larger. Of course, the higher the number of unique particles, the better, since it means that the degeneracy problem is less present.

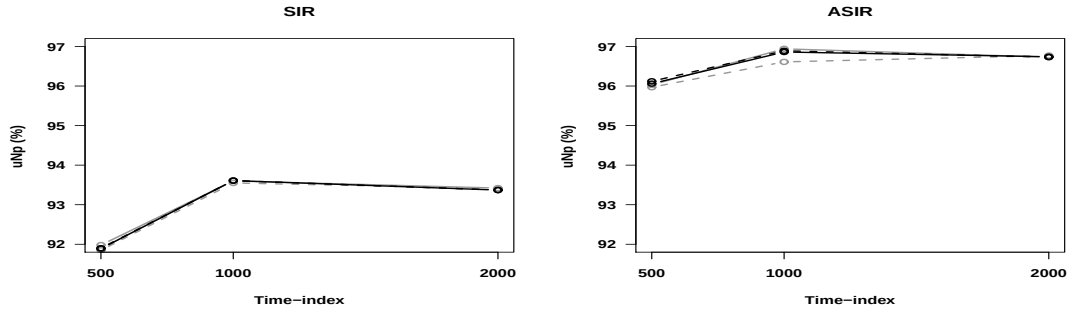
- Therefore, what the attained results confirm is that in general, the ASIR benchmark filter suffers less the degeneracy problem and that both the SIR and the ASIR particle filter variants are more affected by it at low time series length values ($T = 500$), but contrary to expected, after time series length greater than 1000 up to 2000, the percentage mean of unique particle seemingly remains stable. We find this a surprising but positive result. For instance, with $N_p = 5000$ particles, even in the worst-case scenario for the SIR/ASIR particle filters that occurs when using only $T = 500$,

we end up with about 4221 (84% of 5000)/4600 (92% of 5000) particles, which we consider big enough to produce a reliable marginal posterior distribution.

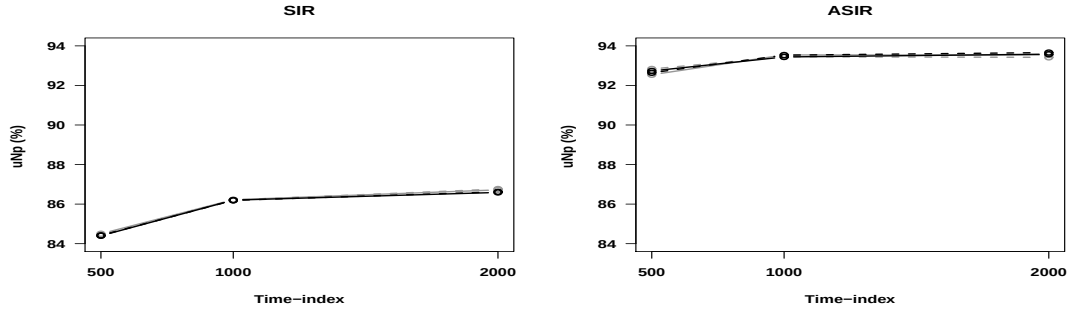
- The analyzed results also indicate that none of the filters suffer greatly the degeneracy problem, at least up to 2000 as shown. To better illustrate the non-degeneracy of the studied particle filters, see Figure 6.9 that is created for the same exemplary run considered above using $N_p = 5000$ particles. The displayed plots depict the histograms (together with the estimated posterior densities; black/ dashed) of the estimated state values at last time-index $T = 1000$: $\hat{x}_{T|T}$. In this case, each row refers to a different scenario (Cases 1–4) and each column to a different particle filter variant (SIR and ASIR). Based on this particular illustration, we can say that there is not a noticeable difference among the particle filters in question, as is shown (empirically) in the tables and related figures which have already been presented. What is confirmed is that using $N_p = 5000$ protects ourselves against the degeneracy problem, but keep in mind that sometimes even a smaller number of particles could be used.



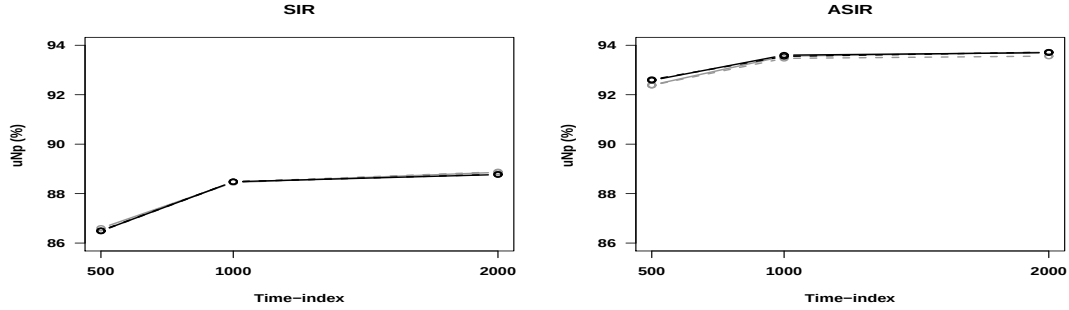
(a) Case 1: $\Theta = (\mu, \phi, \sigma_\eta^2)' = (-0.632, 0.981, 0.194^2)$; $0.194^2 = 0.038$



(b) Case 2: $\Theta = (\mu, \phi, \sigma_\eta^2)' = (-0.632, 0.9, 0.194^2)$; $0.194^2 = 0.038$

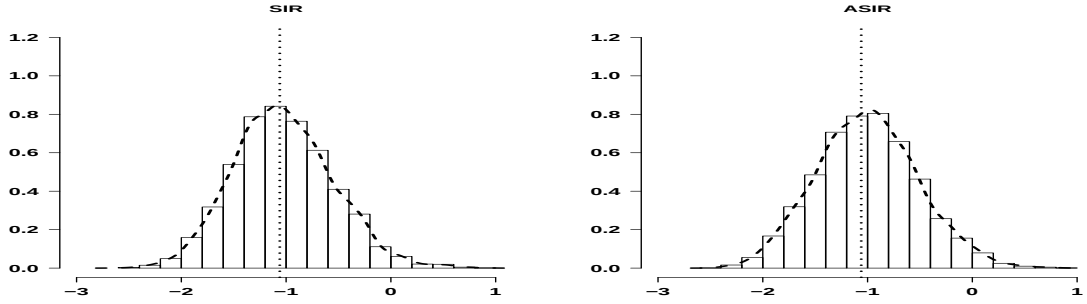


(c) Case 3: $\Theta = (\mu, \phi, \sigma_\eta^2)' = (-0.632, 0.981, 0.363^2)$; $0.363^2 = 0.132$

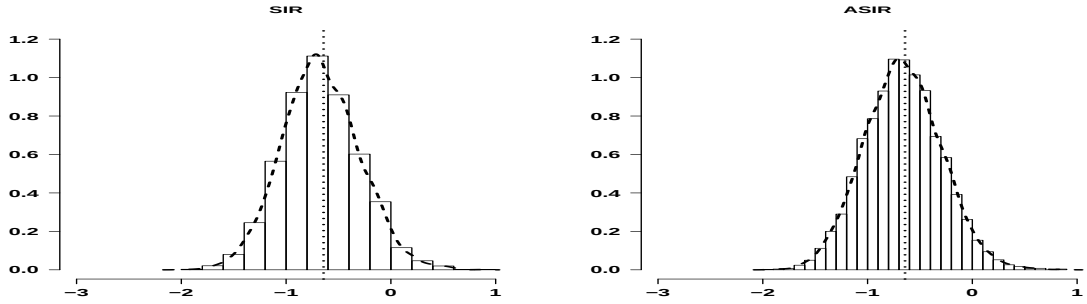


(d) Case 4: $\Theta = (\mu, \phi, \sigma_\eta^2)' = (-0.632, 0.9, 0.363^2)$; $0.363^2 = 0.132$

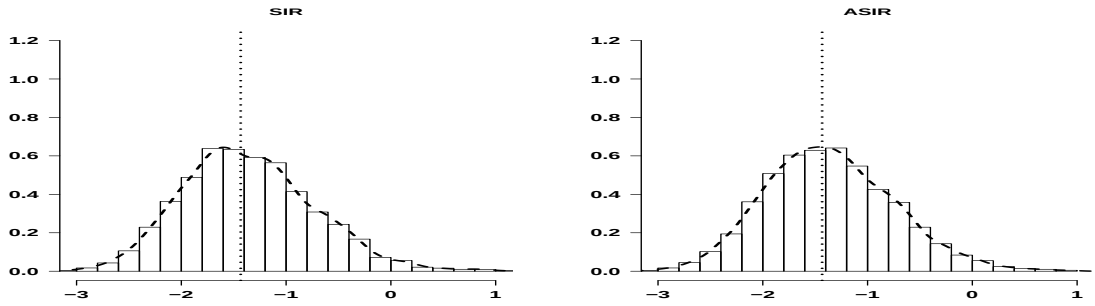
Figure 6.8: SARV(1) model: Behavior of estimated mean percentage of unique number of particles %uNp at last time-index $t = T$ for the SIR and ASIR PF variants. Assessment of the impact of the time series length (x-axis) and the number of particles ($N_p = 200$: grey/dashed; $N_p = 500$: grey/continuous; $N_p = 1000$: black/dashed and $N_p = 5000$: black/continuous). Results shown for Cases 1–4.



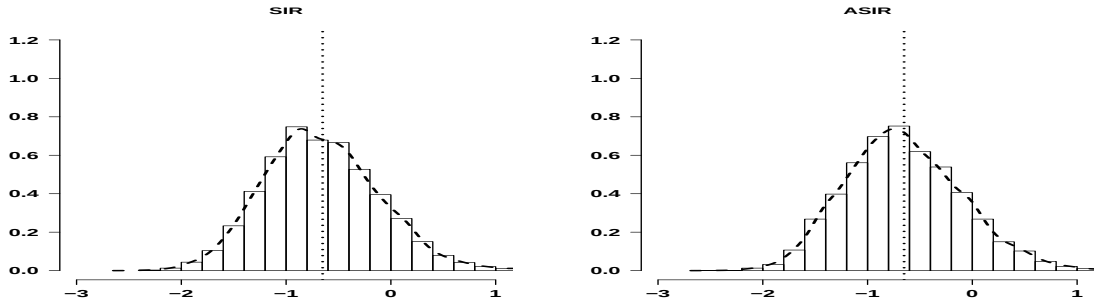
(a) Case 1: $\Theta = (\mu, \phi, \sigma_\eta^2)' = (-0.632, 0.981, 0.194^2)$; $0.194^2 = 0.038$



(b) Case 2: $\Theta = (\mu, \phi, \sigma_\eta^2)' = (-0.632, 0.9, 0.194^2)$; $0.194^2 = 0.038$



(c) Case 3: $\Theta = (\mu, \phi, \sigma_\eta^2)' = (-0.632, 0.981, 0.363^2)$; $0.363^2 = 0.132$



(d) Case 4: $\Theta = (\mu, \phi, \sigma_\eta^2)' = (-0.632, 0.9, 0.363^2)$; $0.363^2 = 0.132$

Figure 6.9: SARV(1) model: Histogram (together with the estimated posterior density; black/dashed) of estimated state values $\hat{x}_{T|T}$ via the SIR and ASIR PF variants (At time T, the true value of the generated state, x_T, is represented by a black dotted vertical line). Results shown for last data set with $T = 1000$ and $N_p = 5000$.

Thus, all results of the Monte Carlo study I, dealing solely with filtering the states of the nonlinear SARV(1) model, can be subsumed by the following summarizing points indicating the general conclusions of this study:

RMSE: The SIR PF variant attains a similar statistical performance to the benchmark ASIR PF. The SIS filter was confirmed to be a non-operational algorithm due to the inherent degeneracy drawback and potential divergence. The use of a larger number of particle N_p or a larger time series length, lead to a decrease (though slight sometimes) of the RMSE.

CPU: The SIR PF computationally outperforms the benchmark ASIR PF. For instance, with $T = 1000$ and $N_p = 5000$, the ASIR takes about twice the mean-CPU time of the SIR PF (16 vs 8 seconds).

%UnP: The mean percentage of unique particle at last time index $t = T$ is higher for the ASIR PF. Overall, %uNp spans from around 84%–94% for the SIR, and from around 92%–97% for the ASIR. These results indicate that the ASIR suffers less the degeneracy problem, but we consider that the SIR behaves reasonably well, too. Notice that, within each scenario and filter, for a time series length T larger than 1000, %uNp remains rather stable. In the same fashion, increasing N_p does not produce changes in the the value of %uNp, it remains rather stable.

Sign of degeneracy? Observe that there is no sign of degeneracy present in the estimated posterior PDF of the states (volatility), at least up to T=2000 for the model at hand.

Number of particles: Irrespective of the PF variant studied and specific case-scenarios, we recommend to use at least $N_p = 1000$ to have a reliable posterior. In previous chapters, dealing with linear and possibly non-stationary state-space models, $N_p = 5000$ particles is found to provide satisfactory estimation performance for most particle filter variants studied and simulation settings in consideration. Thus, as a rule of thumb we continue recommending the use of $N_p = 5000$ particles in the estimation procedure, regardless the filter, case-scenarios, and type of model. Indeed, simulation results confirm that using $N_p = 5000$ protects ourselves against the degeneracy problem, but also that sometimes, even a smaller number of particles could be used; in such cases, specific results and conclusions drawn listed above could be used as a guide.

Non-suitable filters: As mentioned previously, for the nonlinear SARV(1) state-space model at hand, none of the nonlinear Kalman-based filters studied (EKF, UKF, EPF, UPF) are suitable; see Acosta and Muñoz (2007). Additionally, the SIS filter is confirmed to be non-operational as it quickly leads to degeneracy and divergence. In fact, the SIS algorithm was included here only for illustrative reasons and will not be considered hereafter.

6.6 Simulation Study II: Simultaneous Estimation of States and Parameters of the Nonlinear SARV(1) Model

In contraposition to the last section, which deals solely with state estimation, the present section aims to estimate simultaneously the states and fixed parameters of the nonlinear SARV(1) model, whose original state-space representation is specified in the previous section (equations (6.3) and (6.4)). In this more realistic scenario, we place our effort in applying our SIRJ PF variant (fully explained in Chapter 5 and outlined in Algorithm 13 (on page 151) in contraposition with the widely used LW PF variant outlined in Algorithm 12 (page 148); the latter taken as a benchmark. As commented in previous sections, for the nonlinear SARV(1) model at hand, all studied Kalman-based particle filters are unsuitable.

To achieve our goal, we first specify the augmented state-space formulation for the nonlinear model at hand, where the initial state as well as the unknown model parameters are assumed to be random variables and thus are assigned a prior distribution. Then, a second Monte Carlo study is carried out to assess the statistical and computational performance of the competing (and model-suitable) PF variants: SIRJ and LW.

6.6.1 The Augmented State Space Representation

The state-space formulation of the augmented state-space vector $\boldsymbol{l}_t = (\boldsymbol{x}_t, \boldsymbol{\Theta})' = (\mu, \phi, \sigma_\eta)'$, corresponding to the stochastic autoregressive volatility model at hand, take the specific form (see equations (5.1) and (5.2) for the general formulation):

$$\boldsymbol{l}_t = \begin{bmatrix} \boldsymbol{x}_t \\ \boldsymbol{\Theta} \end{bmatrix} = \tilde{f}(\boldsymbol{l}_{t-1}, \boldsymbol{\eta}_t) = \begin{bmatrix} f(x_{t-1}, \eta_t) \\ \boldsymbol{\Theta} \end{bmatrix} = \begin{bmatrix} \mu + \phi\,(x_{t-1} - \mu) + \sigma_\eta \eta_t \\ \begin{bmatrix} \mu \\ \phi \\ \sigma_\eta^2 \end{bmatrix} \end{bmatrix}, \qquad \textit{(Transition equation)} \tag{6.8}$$

$$\boldsymbol{r_t} = \tilde{h}(\boldsymbol{l}_t, \boldsymbol{\nu}_t) = h(x_t, \nu_t) = \exp(x_t/2)\nu_t, \qquad \textit{(Measurement equation)} \tag{6.9}$$

where, as before, $\boldsymbol{x}_t$ is the latent volatility and the uncorrelated sequences $\boldsymbol{\eta_t}$ and $\boldsymbol{\nu_t}$ follow a Gaussian $\mathcal{N}(0,1)$ distribution. Thus, in the above nonlinear dynamic state-space model, the unknown parameter vector is conformed by $\Theta = (\mu, \phi, \sigma_\eta^2)'$, with stationarity restriction $\phi \in (-1, 1)$. To complete this state-space formulation, a prior distribution on the initial augmented state vector $\boldsymbol{l}_0$, must be assumed.

Following, we present the prior distributions used in this particular case.

6.6.2 A Note About the Priors Used

The prior distributions adopted for the nonlinear SARV(1) model are the ones usually found in the literature when estimating this type of models. Specifically, as many authors do, we choose a normal

prior for the original state variable x_0 and the parameter μ, a Beta prior for the persistence parameter ϕ and an Inverse Gamma prior for the transition noise variance parameter σ_η^2; see, for instance, the priors used in Chib, Nardari, and Shephard (2002), Congdon (2007), and Lopes and Tsay (2011).

In this section, for the initial state variable, a normal prior of the form $x_0 \sim N(\mu_{x_0}, \Sigma_{x_0})$ with hyper parameters μ_{x_0} and Σ_{x_0} is adopted.

Referring to the unknown model parameters:

- Since no prior information about the parameter μ is available, a diffuse Gaussian prior with mean value of -8 and variance of 25 is used: $\mu_0 \sim N(-8, 5^2)$.
- As previously mentioned, return series tend to have high persistence parameters (values of ϕ around or above 0.9) and it is for this reason that for this parameter an informative Beta prior distribution is adopted: $\phi \sim 2Be(20, 1.5) - 1$. In this case, the persistence parameter ϕ takes a prior mean and variance of about 0.86 ad 0.012, respectively.
- Since a variance parameter cannot take on negative values, a prior from the inverted Gamma family is taken. Specifically, the assumed diffuse prior for the transition noise variance parameter is an Inverse Gamma distribution formulated as $\sigma_{\eta_0}^2 \sim IG(\frac{n_0}{2}, \frac{n_0}{2} \cdot S_{\eta_0}^2)$. The chosen hyper parameters' values for this variance prior are $n_0 = 10$ and $S_{\eta_0}^2 = \sigma_\eta^2$; the last equated to the true transition noise variance value. Notice that we use a diffuse prior non-centered in the true value with prior mean value given by $\frac{n_0}{n_0-2}\sigma_\eta^2$.

  Thus, the specification of the aforementioned parameter's prior distributions with used hyper-parameters and corresponding basic statistics (prior's mean and variance) is summarized in the following Table 6.3.

Table 6.3: Summary of prior distributions specification with used hyperparameters and corresponding prior's mean and variance.

				Prior	
$\boldsymbol{\theta}$	**Prior distribution**	**Hyperparameters**		**Mean**	**Variance**
μ_0	$\mathcal{N}(a_0, b_0^2)$	$a_0 = -8$	$b_0 = 5$	-8	25
ϕ_0	$2\text{Beta}(c_0, d_0) - 1$	$c_0 = 20.5$	$d_0 = 1.5$	0.86	0.012
$\sigma_{\eta_0}^2$	$IG(\frac{n_0}{2}, \frac{n_0}{2} \cdot S_{\eta_0}^2)$	$n_0 = 10$	$S_{\eta_0}^2 = \sigma_\eta^2$	$1.25 \cdot \sigma_\eta^2$	$0.052 \cdot (\sigma_\eta^2)^2$

With respect to possible transformations of the parameters, Liu and West (2001) point out that the adoption of the normal-kernel approach with real-valued parameters is generally more appropriate. Following their recommendation, we also routinely work with the logarithmic transformation of variance parameters and with the logit transformation of any parameter restricted to a finite range. Specifically, in the case of the SARV(1) model at hand, we work with the log-transform of the transition

noise variance and the logit-transform of the persistence parameter ϕ restricted to lie in the stationary region: $|\phi| < 1$.

In the sequel, all details regarding the undergone Monte Carlo study are presented. We remark that in this thesis, as seen and applied in Chapter 4, we have also implemented the EPF and UPF filters, which are relatively more complex to implement than the SIRJ and LW particle filter variants. However, as explained in the previous section, none of the Kalman-based filters are suitable for the nonlinear SARV(1) model at hand.

6.6.3 Simulation Study II: Design and Simulation Settings

As already mentioned, in this second simulation study, we revisit the nonlinear SARV(1) model to estimate simultaneously the states and the fixed and unknown model parameters via particle filtering. In other words, using our proposed SIRJ particle filter and the benchmark LW particle filter variant, we aim to estimate not only the latent states (volatility) but also the unknown fixed-parameter vector, $(\boldsymbol{x}_t, \boldsymbol{\Theta})' = (\mu, \phi, \sigma_\eta^2)'$. Thus, departing from the SARV(1) nonlinear augmented state-space formulation given by equations (6.8) and (6.9) and adopting the prior distributions outlined previously, we proceed to carry out a second simulation study.

Simulation Design

Herein, we adopt the same general simulation design used in previous chapters. Specifically, we slightly modify the simulation procedure of Chapter 5 (dealing with the simultaneous estimation of states and parameters of the Local Level model) to tackle such estimations for the nonlinear SARV(1) model at hand. Be reminded that the statistical performance of the filters is based on the root mean square error (RMSE) criterion and that the computational performance is based on the elapsed CPU time. Also, to somehow measure the degree of degeneracy of the competing filters, the average number of unique particles (in %) at the last time-index $t = T$ is computed. Additionally, this section assesses the impact of the time series length and of increasing the number of particles on the statistical and computational performance of the two competing particle filter variants.

Next, we provide the specific simulation's settings, experimental results, remarks, and conclusions for the second simulation study.

Simulation Settings

Herein, we provide a summary list of the simulation settings used for the conducted Monte Carlo experiment:

- **Filters**: LW and SIRJ particle filter variants outlined in Algorithms 12 and 13, respectively (Kalman-based filters not suitable in this case).
- **Cases**: The four simulation scenarios are the same as defined in the last section, chosen according to typically found values in literature:

  - Case 1: $\Theta = (\mu, \phi, \sigma_\eta^2)' = (-0.632, 0.981, 0.194^2)$; $0.194^2 = 0.038$.
  - Case 2: $\Theta = (\mu, \phi, \sigma_\eta^2)' = (-0.632, 0.9, 0.194^2)$; $0.194^2 = 0.038$.
  - Case 3: $\Theta = (\mu, \phi, \sigma_\eta^2)' = (-0.632, 0.981, 0.363^2)$; $0.363^2 = 0.132$.
  - Case 4: $\Theta = (\mu, \phi, \sigma_\eta^2)' = (-0.632, 0.9, 0.363^2)$; $0.363^2 = 0.132$.

- **Resampling scheme**: Stratified resampling
- **Number of replications**: $S = 100$
- **Time series length**: A reasonably large time series length, $T = 1000$, is used. However, we also consider some intermediate values $T \in \{250, 500, 750\}$ to illustrate and empirically assess the impact of the time series length on the performance of the PF variants under study when estimating together the states and the parameters of the nonlinear SARV(1) model.
- **Number of particles**: $N_p = 5000$ and $N_p = 10000$. Herein, we aim to assess the impact of increasing the number of particles on the performance of the PF variants under study. As listed, we consider two settings for the number of particles: N_p equal to the value 5000 previously found to provide an overall satisfactory estimation performance, and a larger value $N_p = 10000$.
- **Discount parameter**: To somehow explore the impact of discount values not only inside the mentioned range on the particle filtering performance, we consider three settings of the discount factor: $\delta \in \{0.83, 0.95, 0.99\}$.
- **Comparison criteria**: RMSE and CPU time. Additionally, the average number (%) of distinct particles %uNp at last time-index $t = T$ is computed.

Following, we present the simulation results, remarks, and conclusions regarding the undergone MC study.

6.6.4 Simulation Study II: Experimental Results

In Table 6.4, we provide the numeric results which summarize the performance of the two particle filters, SIRJ and LW, used in handling the simultaneous estimation of states and parameters for the nonlinear SARV(1) model; results for $T = 1000$ reported. This table is organized in two vertical blocks distinguished by two values of the discount parameter δ: $\delta \in \{0.83, 0.95\}$. Each block itself contains two settings for the number of particles N_p: $N_p \in \{5000, 10000\}$. For each setting of the number of particles, the measures Mean(RMSE) and Var(RMSE) for the estimated states and three estimated parameters are reported. Horizontally, for each filter we also report the estimated percentage of unique number of particles (%uNp) at last time-index $t = T$; for this measure, the standard deviation and not the variance is reported. Notice that the estimated values corresponding to the discount value $\delta = 0.99$ are not reported in the table, since generally worse statistical performance (higher RMSE values) is attained at that setting.

Table 6.4: Summary of simulation II results: Estimation of the states (volatility) and parameters for the SARV(1) model with T=1000. For the states and model parameters $\Theta = (\mu, \phi, \sigma_\eta^2)'$, Mean and VAR denote the Mean(RMSE) and Var(RMSE). For the percentage of unique number of particles (%uNp) at last time-index $t = T$, Var denotes the standard deviation, SD(%uNp).

		$\delta = 0.83$				$\delta = 0.95$			
		$N_p = 5000$		$N_p = 10000$		$N_p = 5000$		$N_p = 10000$	
		Mean	**Var**	**Mean**	**Var**	**Mean**	**Var**	**Mean**	**Var**
Case 1: $\Theta = (\mu, \phi, \sigma_\eta^2)' = (-0.632, 0.981, 0.194^2)$									
LW	x_t	0.509	0.001	0.509	0.001	0.511	0.001	0.51	0.001
	μ	0.819	0.034	0.792	0.036	0.858	0.042	0.828	0.048
	ϕ	0.022	1e-04	0.022	1e-04	0.028	3e-04	0.031	3e-04
	σ_η^2	0.009	1e-06	0.009	1e-06	0.009	1e-06	0.009	1e-06
	%uNp	96.61	2.30	96.59	2.27	96.63	2.33	96.59	2.32
SIRJ	x_t	0.509	0.001	0.509	0.001	0.511	0.001	0.510	0.001
	μ	0.815	0.036	0.798	0.036	0.846	0.040	0.828	0.043
	ϕ	0.02	1e-04	0.022	2e-04	0.028	3e-04	0.03	3e-04
	σ_η^2	0.008	1e-06	0.009	1e-06	0.009	1e-06	0.009	1e-06
	%uNp	91.5	6.45	91.04	6.98	91.34	6.62	91.01	7.22
Case 2: $\Theta = (\mu, \phi, \sigma_\eta^2)' = (-0.632, 0.90, 0.194^2)$									
LW	x_t	0.429	7e-04	0.429	7e-04	0.429	6e-04	0.428	7e-04
	μ	0.606	0.004	0.596	0.004	0.661	0.004	0.634	0.004
	ϕ	0.043	1e-04	0.043	1e-04	0.048	4e-04	0.046	2e-04
	σ_η^2	0.006	1e-06	0.007	1e-06	0.008	1e-06	0.008	1e-06
	%uNp	97.2	1.98	97.13	2.02	97.18	2.01	97.14	2.1
SIRJ	x_t	0.431	6e-04	0.429	7e-04	0.431	7e-04	0.428	7e-04
	μ	0.609	0.004	0.593	0.004	0.663	0.004	0.635	0.004
	ϕ	0.047	2e-04	0.044	1e-04	0.049	2e-04	0.046	3e-04
	σ_η^2	0.007	1e-06	0.006	1e-06	0.008	1e-06	0.007	1e-06
	%uNp	93.68	4.63	93.41	5.04	93.78	4.88	93.65	4.88
Case 3: $\Theta = (\mu, \phi, \sigma_\eta^2)' = (-0.632, 0.981, 0.363^2)$									
LW	x_t	0.705	0.001	0.704	0.001	0.707	0.001	0.706	0.002
	μ	1.154	0.184	1.137	0.181	1.206	0.212	1.211	0.250
	ϕ	0.023	2e-04	0.024	2e-04	0.034	3e-04	0.034	4e-04
	σ_η^2	0.029	3e-04	0.03	3e-04	0.03	4e-04	0.031	5e-04
	%uNp	93.2	4.41	93.15	4.25	93.23	4.42	93.06	4.37
SIRJ	x_t	0.705	0.001	0.705	0.001	0.707	0.002	0.707	0.002
	μ	1.137	0.181	1.146	0.202	1.228	0.249	1.205	0.246
	ϕ	0.023	2e-04	0.024	2e-04	0.033	3e-04	0.034	3e-04
	σ_η^2	0.028	3e-04	0.029	3e-04	0.031	5e-04	0.032	4e-04
	%uNp	86.18	10.5	85.74	10.93	86.04	10.46	85.7	10.89

Table 6.4: Summary of simulation II results: Estimation of the states (volatility) and parameters for the SARV(1) model with T=1000 (continued).

		$\delta = 0.83$				$\delta = 0.95$			
		$N_p = 5000$		$N_p = 10000$		$N_p = 5000$		$N_p = 10000$	
		Mean	**Var**	**Mean**	**Var**	**Mean**	**Var**	**Mean**	**Var**
Case 4: $\Theta = (\mu, \phi, \sigma_\eta^2)' = (-0.632, 0.90, 0.363^2)$									
LW	x_t	0.628	9e-04	0.627	9e-04	0.628	9e-04	0.628	9e-04
	μ	0.674	0.008	0.659	0.008	0.722	0.007	0.702	0.007
	ϕ	0.037	1e-04	0.036	1e-04	0.038	2e-04	0.037	2e-04
	σ_η^2	0.024	1e-04	0.025	1e-04	0.026	1e-04	0.025	1e-04
	%uNp	94.14	4.09	94	4.2	94.05	4.11	93.86	4.39
SIRJ	x_t	0.629	8e-04	0.628	9e-04	0.629	9e-04	0.627	9e-04
	μ	0.68	0.008	0.654	0.008	0.741	0.007	0.703	0.008
	ϕ	0.04	1e-04	0.036	1e-04	0.039	2e-04	0.037	2e-04
	σ_η^2	0.026	1e-04	0.025	1e-04	0.026	1e-04	0.025	1e-04
	%uNp	88.97	8.43	88.62	8.71	88.85	8.49	88.51	8.89

6.6.5 Simulation Study II: Remarks and Conclusions

Based on the simulation results reported in Table 6.4, we make the following remarks and conclusions regarding the performance of the two competing particle filters (SIRJ and LW) when handling the simultaneous estimation of the states (the volatility) and parameters (level, persistence, and transition noise variance) of the nonlinear SARV(1) model:

First, we refer to the effect of the discount factor δ on the statistical performance of the two competing particle filter variants: the SIRJ and the LW. That is, to test the potential impact of the discount factor on the estimations of the states and the three model parameters of the nonlinear model at hand, focus on the comparison of the mean-RMSE estimates obtained at the three chosen discount factors, $\delta \in \{0.83, 0.95, 0.99\}$; RMSE-results for the latter are generally the worst (higher RMSE values) irrespective of the case scenario and are thus not displayed in the above mentioned table. Therefore, taking a look at the reported results we find that:

- Within each case scenario, the statistical performance of the filters, indicated by the mean-RMSE values, shows a seemingly increasing pattern as a function of the discount factor δ; conclusion valid among the three δ settings included in this second MC study. That is, for the nonlinear SARV(1) model under study, best statistical efficiency, i.e., lowest RMSE, is generally attained at lowest values of the discount factor: in this case at $\delta = 0.83$. Recall that when handling the linear and non-stationary Local Level model, considering in that case discount factor values $\delta \in \{0.83, 0.95\}$, no distinguishable influence of δ is found. This suggests that the impact of the discount factor may differ with the type of model at hand. We remark, though, that a further

Monte Carlo study must be undergone to exhaustively study the impact of the discount factor choice. Next, some more specific remarks regarding the impact of the discount factor are made:

- $\boldsymbol{x_t}$ Notice that, within each case scenario, the estimated mean-RMSE values corresponding to the states are hardly affected by the choice of the discount factor δ. This makes sense, since they are not directly affected by the choice of the discount factor δ.

- As known, the three originally fixed and unknown model parameters μ, ϕ and σ_η^2 (level, persistence, and transition noise variance[6], respectively) have been jittered and thus are directly affected by the choice of the discount factor δ. The question is whether and how this choice has an impact on the quality of the estimations of those three model parameters. We find that some discrepancies are observed in the mean-RMSE of the level, persistence, and volatility of volatility parameters, especially in the first two, as further detailed below:

  - $\boldsymbol{\mu}$ The estimated mean-RMSE is always less for $\delta = 0.83$ (as compared to both $\delta = 0.95$ and $\delta = 0.99$).
  - $\boldsymbol{\phi}$ Likewise, the estimated mean-RMSE for the persistence parameter is always less for $\delta = 0.83$ (as compared to both $\delta = 0.95$ and $\delta = 0.99$).
  - $\boldsymbol{\sigma_\eta^2}$ Hardly any effect of the discount factor is observed on the estimated transition noise variance. Indeed, within each case scenario, these values seem to stay around the same value regardless of the choice of the discount factor, δ, used in the estimation procedure.

- Therefore, for the model at hand and within the range of values of δ suggested by Liu and West (2001), we recommend to use the discount factor $\delta = 0.95$. However, as aforementioned, outside such range generally best statistical performance is attained and thus we prefer to use a discount factor of $\delta = 0.83$. Notice that these results hold irrespective of the number of particles used in the estimation procedure.

Second, for a fixed discount factor δ, we focus on the comparison between the SIRJ and the benchmark LW particle filters. Take, for instance, $\delta = 0.83$ where best statistical efficiency (smallest RMSE values) is attained (similar results hold if δ is fixed at 0.95 or 0.99):

- In general, irrespective of the case scenario and number of particles used in the estimation procedure, our SIRJ PF variant attains practically the same statistical performance of the benchmark LW PF variant; refer to Table 6.4. Thus, both competing filters show similar statistical performance as indicated by the corresponding mean-RMSE values.

- To somehow study the impact of increasing the number of particles N_p on the statistical performance of the filters, we consider two settings for $N_p \in \{5000, 10000\}$. We find that, irrespective of

[6] The transition noise variance is also called volatility of volatility.

the case scenario, for fixed time series length $T = 1000$, the mean-RMSE of the SIRJ and LW particle filters tend to attain smaller values when using a larger number of particles. Notice, however, that increasing the number of particles beyond $N_p = 5000$, does not generally improve the statistical performance of the competing particle filters. According to simulation results reported in the Table 6.4, a noticeable decrease of the mean-RMSE is only observed in case of the mean-level parameter μ; in case of the states and two other parameters, differences are observed only in the third decimal place. These findings suggest that for the simultaneous estimation of the states and the parameters of the SARV(1) model, at least $N_p = 5000$ particles should be used, and also that to achieve a better overall statistical efficiency it would be more appropriate to use up to $N_p = 10000$ particles.

- Among cases, as expected, differences are observed concerning the statistical performance. Notice that mean-RMSE values differ between cases showing smaller values for $\phi = 0.9$ (as compared to $\phi = 0.981$) and $\sigma_\eta^2 = 0.194^2$ (as compared to $\sigma_\eta^2 = 0.363^2$). Further, the discrepancies observed in the estimated mean-RMSE values of the states and model parameters are detailed below:

  $\boldsymbol{x_t}$ As observed in the last section when only estimating the states, larger mean-RMSE values are attained in those case scenarios combining higher state noise variance with higher persistence parameter values, seemingly in that order: mean-RMSE values are higher in case three, followed by cases four, one and two, respectively.

  $\boldsymbol{\mu}$ For this parameter, the estimated mean-RMSE shows higher values in those case scenarios combining higher persistence parameter values ϕ with higher state noise variance, seemingly in that order: mean-RMSE values are higher in case three, followed by cases one, four and two, respectively.

  $\boldsymbol{\phi}$ For this parameter, mean-RMSE values show higher values (around 0.04-0.05) for $\phi = 0.90$ (as compared to $\phi = 0.98$ with values around 0.02-0.03). This suggests that lower mean-RMSE values are attained in those scenarios with higher persistence parameter values: cases one and three yield smaller RMSE values than cases two and four.

  $\boldsymbol{\sigma_\eta^2}$ Similarly, for this parameter, mean-RMSE values show higher values (around 0.02-0.03) for $\sigma_\eta^2 = 0.363^2$ (as compared to $\sigma_\eta^2 = 0.194^2$; values around 0.01). More specifically, likewise for the states, larger mean-RMSE are attained in those particular scenarios combining higher state noise variance values with higher persistence values, seemingly in that order: mean-RMSE values are higher in case three, followed by cases four, one and two, respectively.

- To better illustrate the discrepancies and similarities of the two competing filters, we go one step further and create an additional table for each case scenario representing the evolution of the posterior mean estimates of the parameters (μ, ϕ, σ^2) for time-indexes $T \in \{250, 500, 750, 1000\}$ and for all 100 Monte Carlo replications ; see Tables 6.5 – 6.8. In each table, for each parameter, the simulation results shown are in the format: Mean (2.5th, 97.5th percentiles) of the posterior

mean estimates at time T along 100 replications. Results are displayed for discount factor values $\delta \in \{0.83, 0.95\}$ and number of particles $N_p \in \{5000, 10000\}$. These complementary tables allow us to confirm, once again, that our SIRJ particle filter variant is valid since it is able to equate the LW filter statistical performance, as shown already in Table 6.4.

- Thus, in all four scenarios under scrutiny, when estimating the states of the nonlinear SARV(1) model, the SIRJ achieves the same statistical performance as the benchmark LW PF. In general, lower RMSE values are attained at the lowest value of the discount parameter value under consideration, say $\delta = 0.83$ (vs 0.95 and 0.99).

  Before continuing, we refer the reader to Section C.2 of Appendix C where we revisit the issue of the potential impact on estimation of the choice of the discount factor δ by including a broader range of values, including two that were prompted by an external referee. There, we consider discount factor values $\delta \in \{0.5, 0.75, 0.83, 0.90, 0.95, 0.99\}$. Naturally, if one focuses only on previously studied δ values conclusions are confirmed exactly regardless of some observed discrepancies concerning the RMSE values among cases and parameters as described beforehand. However, when considering the whole new range of δ values, not only some discrepancies are observed as before, but different conclusions might emerge. For instance the minimum mean-RMSE is not always generally attained at $\delta = 0.83$ anymore, but sometimes even at $\delta = 0.5$. Despite discrepancies, we still believe that for the model at hand $\delta = 0.83$ is generally a reasonably good choice; see Appendix C for complete reasoning.

Third, for the nonlinear SARV(1) model at hand, focusing on the performance of the different filters in terms of the computational time, we conclude that:

- Our SIRJ particle filter is the least expensive algorithm with mean CPU time values around 16.68 seconds [SD: 0.13 seconds] in handling a data set containing $T = 1000$ observations using $N_p = 5000$ particles) in comparison with the benchmark LW filter (24.93[0.08]). As expected, similar results hold in the other three case scenarios (CPU time variations proper of the simulation character of the MC studies) irrespective of the value of the discount parameter δ.

- As also expected, the attained CPU times are larger when using a higher number of particles. For instance, when handling a data set of size $T = 1000$ using $N_p = 10000$ particles, the mean-CPU times take values around 31.14[0.15] and 47.75[0.15] for the SIRJ and LW, respectively. Naturally, these results also hold in the other three case scenarios.

- For fixed time series length and number of particles, there is practically no effect of the discount factor δ on the computational performance of the filters, as expected.

- Thus, in all four scenarios under scrutiny, when simultaneously estimating the states and parameters of the nonlinear SARV(1) model, the SIRJ particle filter always outperforms the computational performance of the benchmark LW PF.

Table 6.5: Evolution of estimated parameters for all 100 MC replications and the two competing PF variants under study with $t \in \{250, 500, 750, 1000\}$. Results shown for discount factor values $\delta \in \{0.83, 0.95\}$ and $N_p \in \{5000, 10000\}$. True parameters values correspond to **Case 1:** $\Theta = (\mu, \phi, \sigma_\eta^2)' = (-0.632, 0.981, 0.194^2)$; $0.194^2 = 0.038$.

				T=250		T=500		T=750		T=1000	
δ	Filter	θ	N_p	Mean	Prob. Int.*	Mean	Prob. Int.	Mean	Prob. Int.	Mean	Prob. Int.
0.83	**LW**	μ	5000	−0.709	(-2.180, 0.532)	−0.676	(-1.754, 0.307)	−0.652	(-1.436, 0.160)	−0.671	(-1.341, -0.008)
			10000	−0.721	(-2.105, 0.445)	−0.671	(-1.821, 0.274)	−0.695	(-1.436, 0.034)	−0.707	(-1.444, -0.130)
		ϕ	5000	0.959	(0.911, 0.982)	0.967	(0.923, 0.984)	0.970	(0.931, 0.987)	0.975	(0.951, 0.987)
			10000	0.961	(0.925, 0.981)	0.969	(0.925, 0.985)	0.972	(0.938, 0.988)	0.974	(0.948, 0.987)
		σ_η^2	5000	0.044	(0.032, 0.065)	0.045	(0.031, 0.064)	0.044	(0.031, 0.063)	0.043	(0.031, 0.059)
			10000	0.046	(0.034, 0.066)	0.045	(0.030, 0.066)	0.044	(0.031, 0.064)	0.042	(0.029, 0.063)
	SIRJ	μ	5000	−0.826	(-2.316, 0.372)	−0.707	(-1.793, 0.125)	−0.748	(-1.443, -0.104)	−0.712	(-1.412, -0.10)
			10000	−0.697	(-2.125, 0.666)	−0.678	(-1.751, 0.218)	−0.703	(-1.506, 0.103)	−0.691	(-1.435, -0.07)
		ϕ	5000	0.962	(0.915, 0.985)	0.972	(0.936, 0.986)	0.974	(0.937, 0.989)	0.977	(0.950, 0.985)
			10000	0.962	(0.919, 0.981)	0.970	(0.919, 0.987)	0.972	(0.929, 0.989)	0.976	(0.945, 0.98)
		σ_η^2	5000	0.044	(0.033, 0.065)	0.044	(0.031, 0.064)	0.041	(0.029, 0.063)	0.039	(0.027, 0.05)
			10000	0.044	(0.034, 0.064)	0.044	(0.032, 0.062)	0.044	(0.031, 0.061)	0.043	(0.031, 0.06)
0.95	**LW**	μ	5000	−0.728	(-2.123, 0.485)	−0.699	(-1.735, 0.359)	−0.697	(-1.498, 0.204)	−0.693	(-1.394, 0.111)
			10000	−0.744	(-2.131, 0.514)	−0.702	(-1.774, 0.423)	−0.726	(-1.434, 0.189)	−0.725	(-1.408, 0.088)
		ϕ	5000	0.957	(0.890, 0.986)	0.968	(0.911, 0.988)	0.973	(0.922, 0.989)	0.977	(0.950, 0.991)
			10000	0.955	(0.903, 0.984)	0.968	(0.905, 0.990)	0.973	(0.921, 0.99)	0.977	(0.949, 0.989)
		σ_η^2	5000	0.044	(0.029, 0.070)	0.044	(0.029, 0.065)	0.043	(0.028, 0.064)	0.042	(0.029, 0.064)
			10000	0.046	(0.032, 0.066)	0.044	(0.030, 0.064)	0.043	(0.029, 0.06)	0.042	(0.028, 0.058)
	SIRJ	μ	5000	−0.822	(-2.190, 0.421)	−0.761	(-1.775, 0.127)	−0.767	(-1.601, -0.067)	−0.751	(-1.617, -0.114)
			10000	−0.737	(-1.982, 0.435)	−0.719	(-1.768, 0.347)	−0.738	(-1.481, 0.215)	−0.733	(-1.449, -0.001)
		ϕ	5000	0.956	(0.878, 0.985)	0.970	(0.905, 0.989)	0.974	(0.927, 0.991)	0.977	(0.953, 0.990)
			10000	0.956	(0.902, 0.985)	0.969	(0.910, 0.988)	0.973	(0.931, 0.99)	0.977	(0.951, 0.990)
		σ_η^2	5000	0.045	(0.031, 0.075)	0.044	(0.029, 0.068)	0.042	(0.029, 0.064)	0.041	(0.028, 0.063)
			10000	0.045	(0.032, 0.068)	0.044	(0.029, 0.065)	0.043	(0.029, 0.06)	0.042	(0.030, 0.060)

* Probability interval: $(2.5th, 97.5th$ percentiles).

Table 6.6: Evolution of estimated parameters for all 100 MC replications and the two competing PF variants under study with $t \in \{250, 500, 750, 1000\}$. Results shown for discount factor values $\delta \in \{0.83, 0.95\}$ and $N_p \in \{5000, 10000\}$. True parameters values correspond to **Case 2:** $\Theta = (\mu, \phi, \sigma_\eta^2)' = (-0.632, 0.90, 0.194^2)$; $0.194^2 = 0.038$.

				T=250		T=500		T=750		T=1000	
δ	Filter	θ	N_p	Mean	Prob. Int.*	Mean	Prob. Int.	Mean	Prob. Int.	Mean	Prob. Int.
0.83	**LW**	μ	5000	−0.663	(-1.105, -0.210)	−0.612	(-0.934, -0.341)	−0.618	(-0.850, -0.378)	−0.629	(-0.838, -0.473)
			10000	−0.676	(-1.094, -0.244)	−0.619	(-0.92 , -0.358)	−0.639	(-0.888, -0.383)	−0.642	(-0.864, -0.459)
		ϕ	5000	0.946	(0.887, 0.973)	0.924	(0.848, 0.965)	0.913	(0.825, 0.962)	0.910	(0.834, 0.953)
			10000	0.949	(0.877, 0.973)	0.931	(0.855, 0.964)	0.918	(0.849, 0.961)	0.903	(0.840, 0.949)
		σ_η^2	5000	0.038	(0.03 , 0.050)	0.035	(0.024, 0.045)	0.033	(0.024, 0.044)	0.032	(0.023, 0.046)
			10000	0.039	(0.031, 0.051)	0.034	(0.026, 0.047)	0.032	(0.024, 0.046)	0.031	(0.022, 0.044)
	SIRJ	μ	5000	−0.766	(-1.216, -0.356)	−0.641	(-0.981, -0.361)	−0.658	(-0.921, -0.395)	−0.642	(-0.846, -0.481)
			10000	−0.686	(-1.149, -0.248)	−0.636	(-0.96 , -0.358)	−0.651	(-0.895, -0.378)	−0.643	(-0.842, -0.478)
		ϕ	5000	0.948	(0.861, 0.976)	0.933	(0.854, 0.973)	0.917	(0.822, 0.967)	0.911	(0.836, 0.959)
			10000	0.951	(0.890, 0.975)	0.930	(0.855, 0.968)	0.913	(0.837, 0.966)	0.907	(0.823, 0.959)
		σ_η^2	5000	0.038	(0.03 , 0.053)	0.034	(0.024, 0.047)	0.031	(0.022, 0.042)	0.030	(0.021, 0.042)
			10000	0.038	(0.031, 0.050)	0.034	(0.027, 0.048)	0.033	(0.026, 0.046)	0.032	(0.024, 0.045)
0.95	**LW**	μ	5000	−0.706	(-1.144, -0.355)	−0.635	(-1.007, -0.357)	−0.637	(-0.881, -0.378)	−0.640	(-0.853, -0.475)
			10000	−0.704	(-1.148, -0.301)	−0.633	(-0.933, -0.375)	−0.641	(-0.873, -0.405)	−0.643	(-0.819, -0.479)
		ϕ	5000	0.937	(0.825, 0.982)	0.907	(0.774, 0.975)	0.899	(0.802, 0.966)	0.898	(0.779, 0.956)
			10000	0.932	(0.820, 0.979)	0.906	(0.794, 0.967)	0.899	(0.816, 0.965)	0.895	(0.802, 0.950)
		σ_η^2	5000	0.036	(0.028, 0.052)	0.033	(0.024, 0.055)	0.031	(0.021, 0.054)	0.031	(0.019, 0.049)
			10000	0.037	(0.028, 0.053)	0.033	(0.022, 0.049)	0.032	(0.022, 0.05)	0.031	(0.021, 0.046)
	SIRJ	μ	5000	−0.752	(-1.211, -0.391)	−0.650	(-0.958, -0.393)	−0.658	(-0.889, -0.438)	−0.644	(-0.844, -0.497)
			10000	−0.704	(-1.112, -0.319)	−0.642	(-0.958, -0.398)	−0.649	(-0.885, -0.435)	−0.646	(-0.834, -0.474)
		ϕ	5000	0.939	(0.790, 0.982)	0.916	(0.804, 0.978)	0.905	(0.809, 0.971)	0.902	(0.812, 0.961)
			10000	0.933	(0.804, 0.978)	0.906	(0.779, 0.969)	0.896	(0.791, 0.964)	0.895	(0.785, 0.952)
		σ_η^2	5000	0.037	(0.027, 0.053)	0.032	(0.022, 0.049)	0.030	(0.021, 0.046)	0.030	(0.020, 0.046)
			10000	0.037	(0.029, 0.050)	0.033	(0.024, 0.050)	0.032	(0.022, 0.05)	0.031	(0.021, 0.046)

* Probability interval: $(2.5th, 97.5th$ percentiles).

Table 6.7: Evolution of estimated parameters for all 100 MC replications and the two competing PF variants under study with $t \in \{250, 500, 750, 1000\}$. Results shown for discount factor values $\delta \in \{0.83, 0.95\}$ and $N_p \in \{5000, 10000\}$. True parameters values correspond to **Case 3:** $\Theta = (\mu, \phi, \sigma_\eta^2)' = (-0.632, 0.981, 0.363^2)$; $0.363^2 = 0.132$.

				T=250		T=500		T=750		T=1000	
δ	Filter	θ	N_p	Mean	Prob. Int.*	Mean	Prob. Int.	Mean	Prob. Int.	Mean	Prob. Int.
0.83	**LW**	μ	5000	−0.737	(-2.905, 1.387)	−0.727	(-2.637, 0.642)	−0.684	(-2.126, 0.506)	−0.691	(-2.068, 0.50)
			10000	−0.795	(-2.896, 1.295)	−0.761	(-2.490, 0.795)	−0.779	(-2.232, 0.399)	−0.78	(-2.027, 0.440)
		ϕ	5000	0.963	(0.922, 0.985)	0.973	(0.927, 0.987)	0.974	(0.948, 0.989)	0.977	(0.958, 0.989)
			10000	0.964	(0.925, 0.985)	0.972	(0.939, 0.987)	0.975	(0.942, 0.988)	0.977	(0.960, 0.988)
		σ_η^2	5000	0.153	(0.104, 0.217)	0.151	(0.110, 0.221)	0.148	(0.113, 0.198)	0.144	(0.109, 0.204)
			10000	0.156	(0.107, 0.223)	0.151	(0.102, 0.211)	0.148	(0.112, 0.203)	0.144	(0.11 , 0.208)
	SIRJ	μ	5000	−0.948	(-3.195, 0.981)	−0.826	(-2.679, 0.730)	−0.903	(-2.389, 0.428)	−0.818	(-2.212, 0.21)
			10000	−0.821	(-3.081, 1.358)	−0.772	(-2.544, 0.793)	−0.807	(-2.265, 0.750)	−0.78	(-2.012, 0.692)
		ϕ	5000	0.963	(0.923, 0.986)	0.973	(0.929, 0.987)	0.975	(0.944, 0.989)	0.978	(0.959, 0.990)
			10000	0.964	(0.919, 0.985)	0.973	(0.939, 0.988)	0.975	(0.939, 0.989)	0.978	(0.958, 0.989)
		σ_η^2	5000	0.152	(0.099, 0.208)	0.150	(0.108, 0.200)	0.143	(0.109, 0.205)	0.139	(0.105, 0.204)
			10000	0.154	(0.108, 0.209)	0.152	(0.105, 0.204)	0.149	(0.112, 0.200)	0.144	(0.11 , 0.200)
0.95	**LW**	μ	5000	−0.915	(-3.333, 1.228)	−0.829	(-2.891, 0.832)	−0.821	(-2.671, 0.585)	−0.795	(-2.447, 0.515)
			10000	−0.956	(-3.434, 1.488)	−0.857	(-2.950, 1.171)	−0.869	(-2.531, 0.939)	−0.846	(-2.513, 0.694)
		ϕ	5000	0.958	(0.892, 0.987)	0.972	(0.923, 0.990)	0.976	(0.941, 0.991)	0.978	(0.959, 0.989)
			10000	0.959	(0.898, 0.989)	0.972	(0.921, 0.990)	0.975	(0.933, 0.991)	0.978	(0.957, 0.99)
		σ_η^2	5000	0.155	(0.099, 0.219)	0.153	(0.105, 0.218)	0.148	(0.112, 0.209)	0.145	(0.110, 0.205)
			10000	0.158	(0.110, 0.229)	0.153	(0.108, 0.211)	0.149	(0.110, 0.223)	0.146	(0.109, 0.204)
	SIRJ	μ	5000	−0.999	(-3.699, 1.156)	−0.896	(-3.040, 1.123)	−0.921	(-2.631, 0.822)	−0.897	(-2.582, 0.566)
			10000	−0.949	(-3.561, 1.338)	−0.885	(-3.042, 1.043)	−0.925	(-2.660, 0.553)	−0.898	(-2.624, 0.537)
		ϕ	5000	0.958	(0.893, 0.989)	0.973	(0.925, 0.991)	0.975	(0.947, 0.991)	0.978	(0.956, 0.991)
			10000	0.960	(0.891, 0.988)	0.973	(0.929, 0.989)	0.974	(0.929, 0.991)	0.978	(0.957, 0.99)
		σ_η^2	5000	0.153	(0.101, 0.226)	0.152	(0.099, 0.228)	0.146	(0.103, 0.216)	0.144	(0.106, 0.224)
			10000	0.157	(0.107, 0.217)	0.155	(0.107, 0.220)	0.150	(0.112, 0.217)	0.146	(0.110, 0.209)

* Probability interval: ($2.5th$, $97.5th$ percentiles).

Table 6.8: Evolution of estimated parameters for all 100 MC replications and the two competing PF variants under study with $t \in \{250, 500, 750, 1000\}$. Results shown for discount factor values $\delta \in \{0.83, 0.95\}$ and $N_p \in \{5000, 10000\}$. True parameters values correspond to **Case 4:** $\Theta = (\mu, \phi, \sigma_\eta^2)' = (-0.632, 0.90, 0.363^2)$; $0.363^2 = 0.132$.

				T=250		T=500		T=750		T=1000	
δ	Filter	θ	N_p	Mean	Prob. Int.*	Mean	Prob. Int.	Mean	Prob. Int.	Mean	Prob. Int.
0.83	**LW**	μ	5000	−0.671	(-1.336, -0.057)	−0.608	(-1.105, -0.179)	−0.617	(-1.039, -0.245)	−0.623	(-0.943, -0.323)
			10000	−0.709	(-1.315, -0.097)	−0.620	(-1.107, -0.189)	−0.639	(-1.046, -0.311)	−0.645	(-0.941, -0.352)
		ϕ	5000	0.930	(0.857, 0.968)	0.911	(0.821, 0.955)	0.909	(0.843, 0.952)	0.912	(0.842, 0.949)
			10000	0.932	(0.861, 0.968)	0.915	(0.825, 0.957)	0.911	(0.838, 0.955)	0.908	(0.839, 0.947)
		σ_η^2	5000	0.119	(0.092, 0.164)	0.115	(0.083, 0.160)	0.113	(0.085, 0.152)	0.114	(0.083, 0.149)
			10000	0.124	(0.094, 0.167)	0.115	(0.079, 0.160)	0.113	(0.086, 0.156)	0.111	(0.081, 0.150)
	SIRJ	μ	5000	−0.807	(-1.512, -0.229)	−0.659	(-1.119, -0.215)	−0.683	(-1.063, -0.261)	−0.648	(-0.979, -0.354)
			10000	−0.689	(-1.257, -0.018)	−0.631	(-1.104, -0.235)	−0.658	(-1.027, -0.318)	−0.644	(-0.943, -0.364)
		ϕ	5000	0.935	(0.871, 0.973)	0.921	(0.835, 0.968)	0.915	(0.849, 0.961)	0.917	(0.857, 0.958)
			10000	0.931	(0.866, 0.969)	0.914	(0.837, 0.961)	0.909	(0.839, 0.958)	0.910	(0.840, 0.951)
		σ_η^2	5000	0.121	(0.089, 0.171)	0.114	(0.078, 0.165)	0.109	(0.079, 0.161)	0.107	(0.078, 0.147)
			10000	0.121	(0.089, 0.167)	0.115	(0.082, 0.161)	0.114	(0.080, 0.153)	0.113	(0.081, 0.152)
0.95	**LW**	μ	5000	−0.703	(-1.319, -0.101)	−0.620	(-1.097, -0.204)	−0.639	(-1.000, -0.295)	−0.644	(-0.942, -0.376)
			10000	−0.720	(-1.366, -0.234)	−0.645	(-1.118, -0.250)	−0.664	(-1.031, -0.333)	−0.658	(-0.957, -0.435)
		ϕ	5000	0.916	(0.793, 0.970)	0.903	(0.782, 0.956)	0.906	(0.812, 0.952)	0.909	(0.841, 0.947)
			10000	0.915	(0.809, 0.968)	0.903	(0.818, 0.959)	0.902	(0.806, 0.957)	0.905	(0.840, 0.951)
		σ_η^2	5000	0.119	(0.084, 0.174)	0.116	(0.079, 0.171)	0.116	(0.078, 0.169)	0.117	(0.080, 0.176)
			10000	0.123	(0.086, 0.186)	0.117	(0.077, 0.184)	0.117	(0.080, 0.169)	0.117	(0.083, 0.161)
	SIRJ	μ	5000	−0.774	(-1.473, -0.233)	−0.654	(-1.172, -0.245)	−0.671	(-1.106, -0.302)	−0.654	(-0.950, -0.393)
			10000	−0.712	(-1.344, -0.169)	−0.647	(-1.131, -0.236)	−0.663	(-1.024, -0.349)	−0.657	(-0.945, -0.415)
		ϕ	5000	0.918	(0.818, 0.972)	0.909	(0.810, 0.964)	0.908	(0.830, 0.967)	0.910	(0.846, 0.955)
			10000	0.913	(0.810, 0.968)	0.902	(0.792, 0.957)	0.903	(0.819, 0.958)	0.907	(0.846, 0.946)
		σ_η^2	5000	0.121	(0.083, 0.172)	0.115	(0.075, 0.166)	0.112	(0.076, 0.152)	0.113	(0.073, 0.157)
			10000	0.122	(0.089, 0.172)	0.118	(0.079, 0.171)	0.118	(0.082, 0.167)	0.118	(0.083, 0.169)

* Probability interval: ($2.5th$, $97.5th$ percentiles).

Finally, we refer to the complementary study (already found to be relevant) to investigate and somehow quantify the degree of degeneracy present in the LW and SIRJ PF variants when handling the simultaneous estimation of states and parameters of the nonlinear SARV(1) model; focus on last horizontal entry within each case scenario of Table 6.4. Additionally, we also analyze the impact of increasing number of particles on degeneracy above $N_p = 5000$ considering also $N_p = 10000$. Thus, based on the analysis of the reported percentage mean of the unique number of particles uNp (%uNp) at last time-index $t = T$ ending up with the following conclusions:

- In general, compared to the SIRJ PF variant, the percentage of distinct number of particles, %uNp, is higher for the benchmark LW algorithm. Overall the simulation settings, irrespective of the case scenario, the percentage mean number of unique particles for the SIRJ spans from around 86%–94% and the benchmark LW from around 93%–97%. Thus, this percentage varies depending on the filter and the case scenario, but seemingly remain stable regardless of the number of particles as detailed below.

- For fixed time series length $T = 1000$, regardless the number of particles used in the estimation procedure, the percentage mean of unique particle seemingly remains stable, but naturally the "absolute" number of unique particles are clearly larger when using a higher number of particles. Of course, the higher the number of unique particles, the better, since it means that the degeneracy problem is less present.

- Therefore, what the attained results indicate is that, in general, the LW PF suffers the degeneracy problem to a lesser degree than the SIRJ. Observe that for a reasonably large time series length $T = 1000$ using $N_p = 5000$ particles, even in the worst-case scenario for the SIRJ/LW particle filters that occurs in Case 3 (combining high persistence $\phi = 0.98$ and transition noise variance $\sigma^2 = 0.363^2 = 0.132$), we end up with about 4300 (86% of 5000) and 4650 (93% of 5000) particles for the SIRJ and LW particle filters, respectively. We consider that the found discrepancies in terms of degree of degeneracy are not relevant, since in both cases the final number of unique particles is big enough to produce a reliable marginal posterior distribution.

- The analyzed results thus indicate that none of the filters greatly suffer the degeneracy problem, at least up to 1000 as shown. To better illustrate that no signs of degeneracy are present in the two studied particle filters, see the Figures 6.10–6.13 that are created for one exemplary run using $N_p = 5000$ particles and discount factor $\delta = 0.83$. The displayed plots, one per case scenario, show the histograms (together with the estimated posterior densities) of the estimated state values $\hat{x}_{T|T}$ and fixed parameters $\hat{\Theta} = (\hat{\mu}, \hat{\phi}, \hat{\sigma}^2)'_{T|T}$ at last time-index $T = 1000$. In this case, each row refers to a different particle filter variant (SIRJ and LW) and each column to a different estimated variable: the states (in the first column), the mean level (in the second column), the persistence (in the third column) and the transition noise variance (in the fourth column). Based on this particular illustration, we can say that there is not a noticeable difference between the particle filters in question, as is shown (empirically) in Table 6.4 and complementary tables and figures already presented. Again, we confirm that using $N_p = 5000$ particles or more, protect ourselves against the degeneracy problem; the specific results and conclusions drawn listed above could be used as a guide.

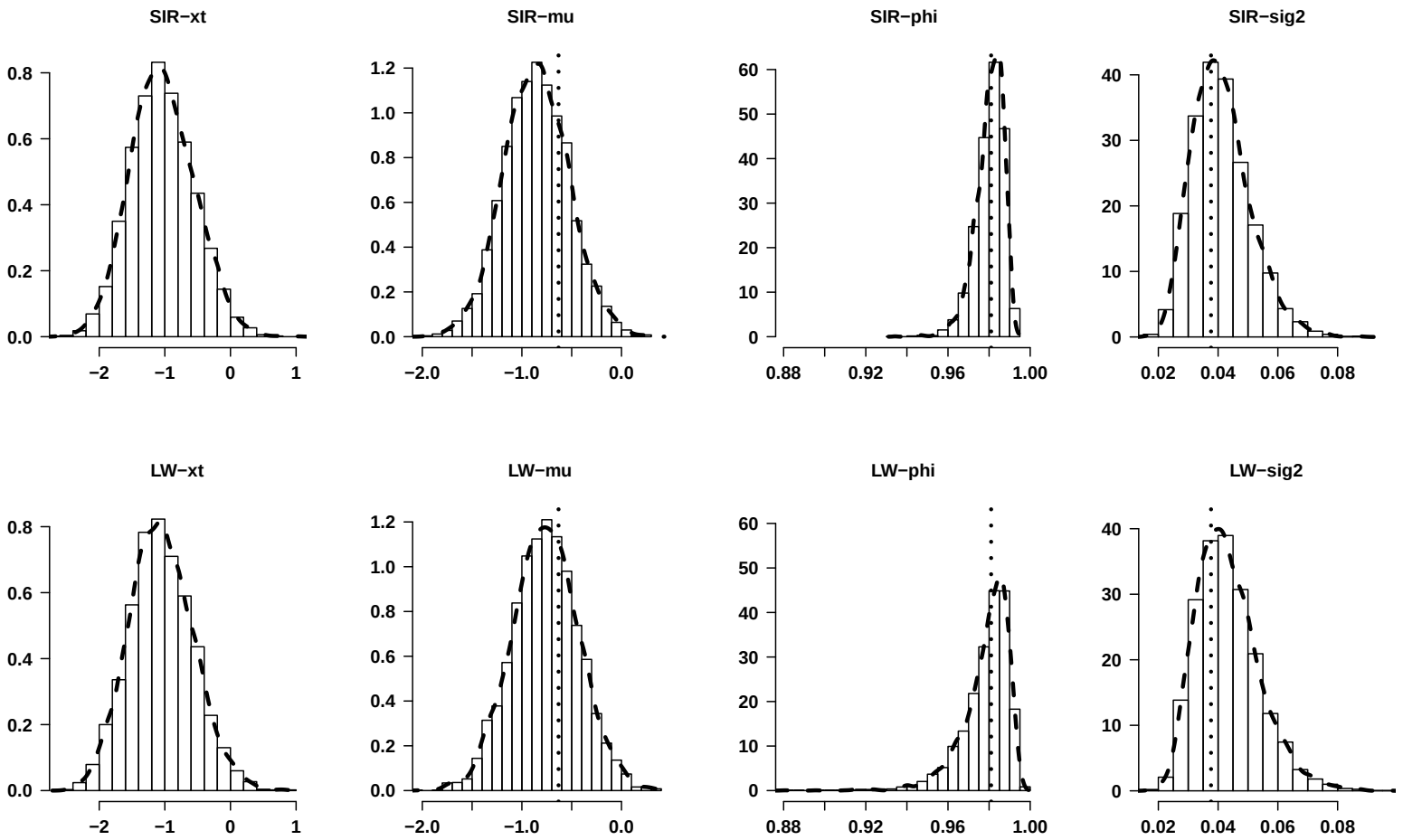


Figure 6.10: Illustration for last exemplar run and last time index $T = 1000$: Histograms representing the estimated posterior distributions of: the states (first column), the level parameter μ (second column), the persistence parameter ϕ (third column) and the transition noise variance σ_η^2 (fourth column) for the SARV(1) model. Notice that each row refers to a different PF variant. Results shown for Case 1, $N_p = 5000$ and $\delta = 0.83$.

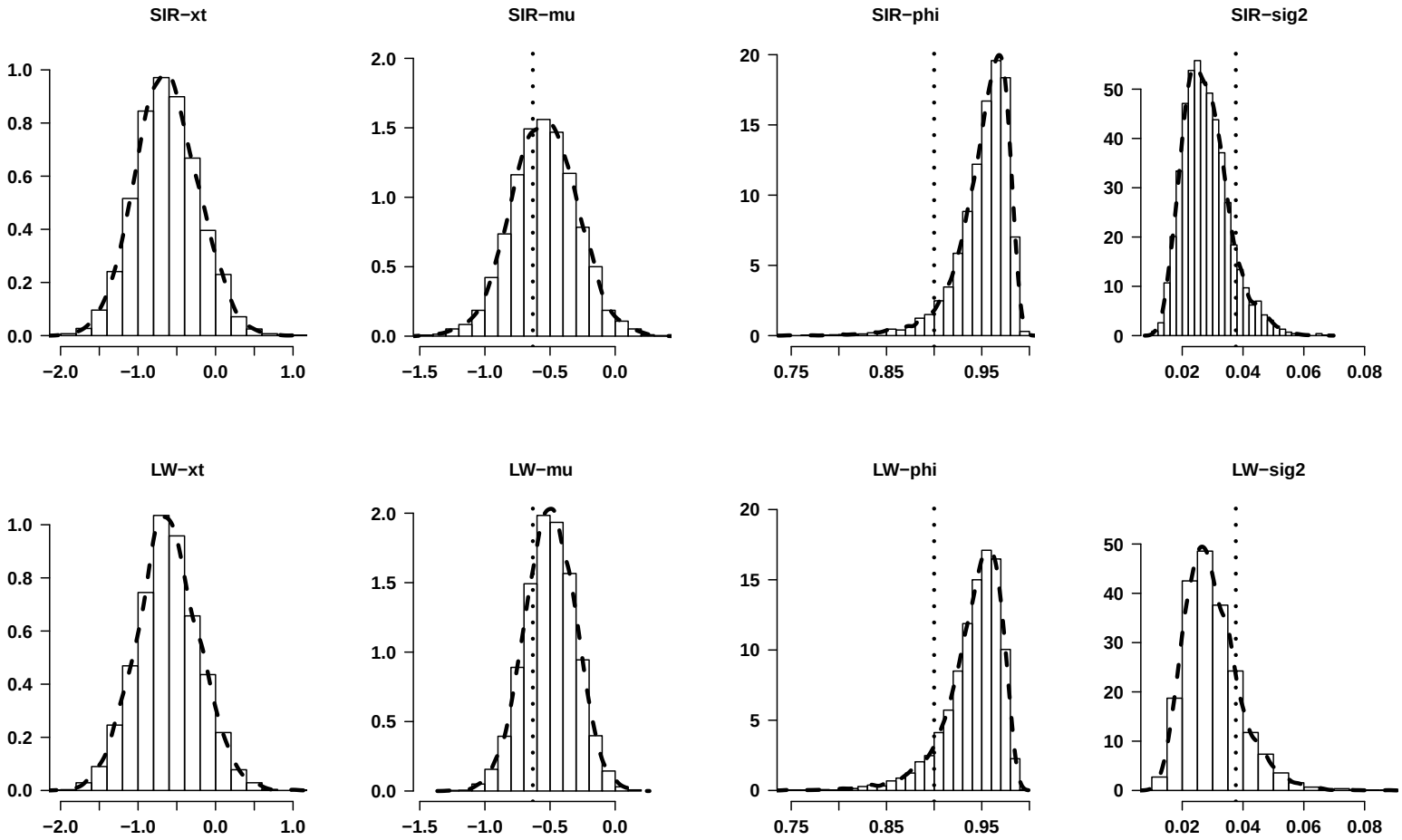


Figure 6.11: Illustration for last exemplar run and last time index $T = 1000$: Histograms representing the estimated posterior distributions of: the states (first column), the level parameter μ (second column), the persistence parameter ϕ (third column) and the transition noise variance σ_η^2 (fourth column) for the SARV(1) model. Notice that each row refers to a different PF variant. Results shown for Case 2, $N_p = 5000$ and $\delta = 0.83$.

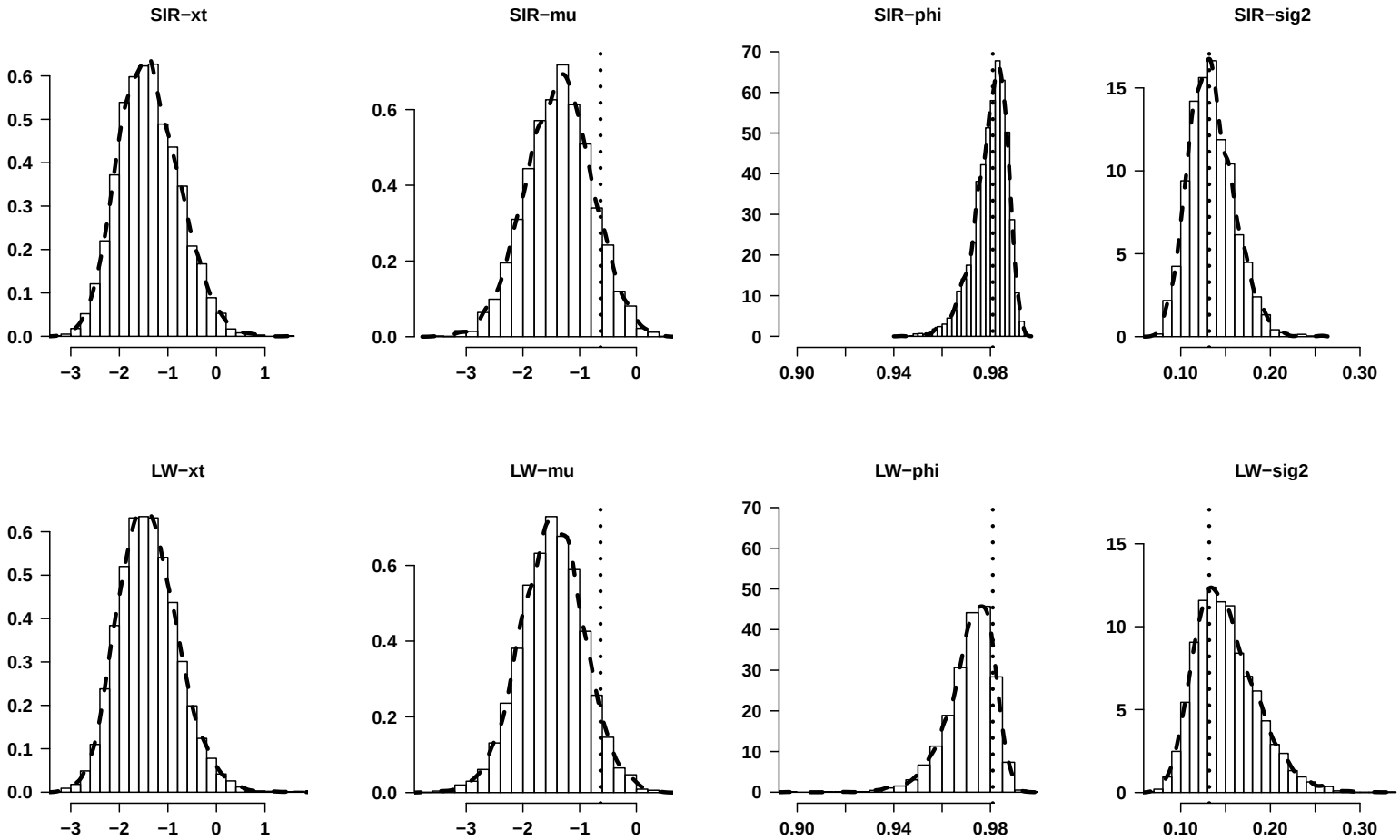


Figure 6.12: Illustration for last exemplar run and last time index $T = 1000$: Histograms representing the estimated posterior distributions of: the states (first column), the level parameter μ (second column), the persistence parameter ϕ (third column) and the transition noise variance σ_η^2 (fourth column) for the SARV(1) model. Notice that each row refers to a different PF variant. Results shown for Case 3, $N_p = 5000$ and $\delta = 0.83$.

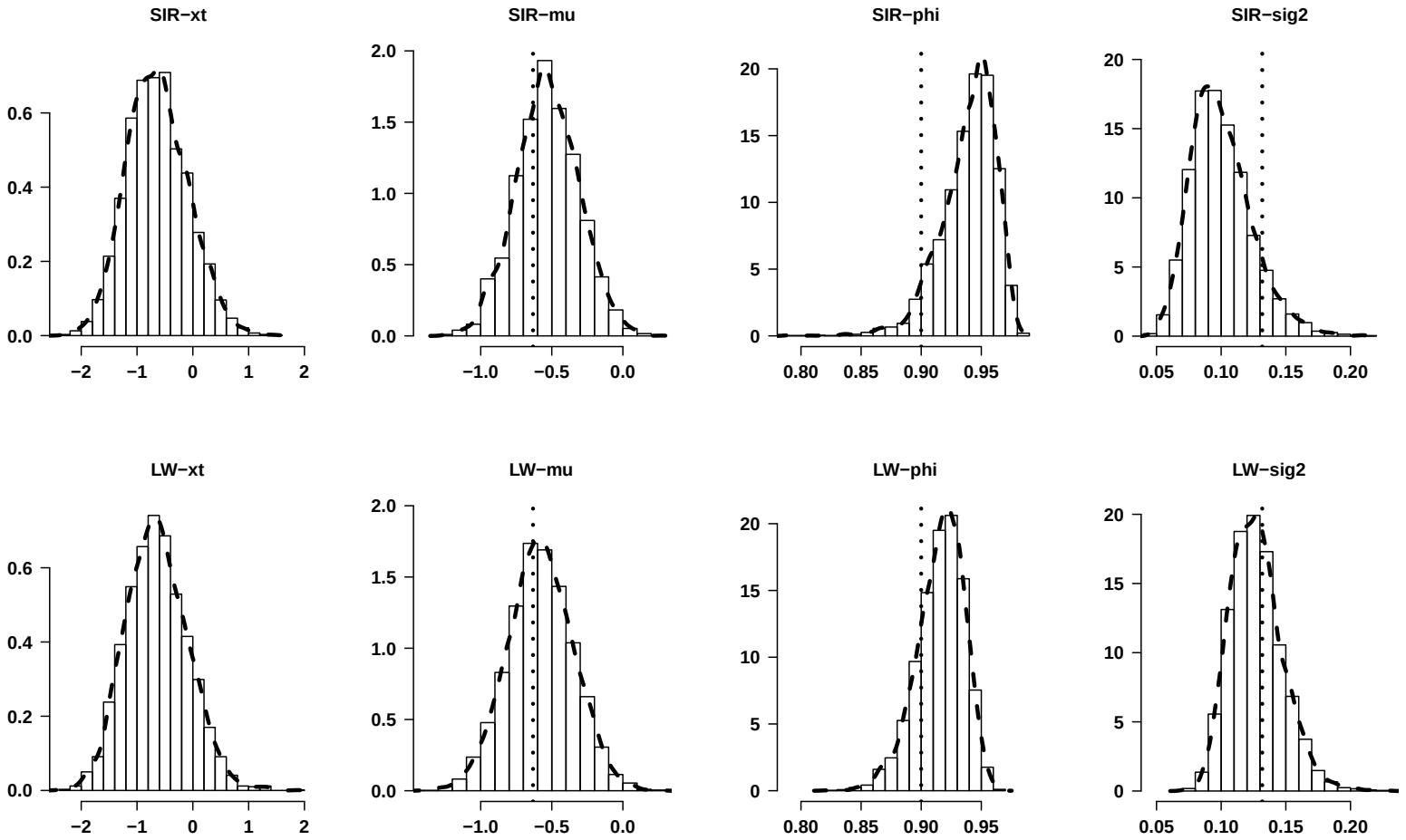


Figure 6.13: Illustration for last exemplar run and last time index $T = 1000$: Histograms representing the estimated posterior distributions of: the states (first column), the level parameter μ (second column), the persistence parameter ϕ (third column) and the transition noise variance σ_η^2 (fourth column) for the SARV(1) model. Notice that each row refers to a different PF variant. Results shown for Case 4, $N_p = 5000$ and $\delta = 0.83$.

- For the model at hand, to be on the safe side, we recommend using at least $N_p = 5000$ particles to end up with a reliable posterior. However, as a rule of thumb, when estimating simultaneously the states and parameters of the nonlinear SARV(1) model, $N_p = 10000$ will provide better mean-RMSE results irrespective of all simulation settings in consideration. As done in previous section, dealing solely with filtering the states, we continue recommending to use at least $N_p = 5000$ particles to protect ourselves against degeneracy. Simulation results also suggest to use 10000 particles in the estimation procedure regardless of the filter type and case scenarios.

Thus, all results of the Monte Carlo study II, dealing with the simultaneous estimation of states and parameters of the nonlinear SARV(1) model, can be subsumed by the following summarizing points indicating the general conclusions of this second study:

RMSE: Our SIRJ PF variant attains a similar statistical performance to the benchmark LW PF. The Kalman-based particle filters explored in this work are confirmed to be non-suitable as they yield null Kalman-Gain values, indicating that estimates are not really updated over time. The use of a larger number of particle N_p or a larger time series length generally lead to a decrease (though slight sometimes) of the RMSE. An increase of the number of particles beyond $N_p = 5000$ does not necessarily improve the statistical performance for all the parameters, but in general using $N_p = 10000$ provides better statistical efficiency in all case-scenarios.

CPU: The SIRJ PF computationally outperforms the benchmark LW PF variant. For instance, with $T = 1000$ and $N_p = 5000$, the LW PF takes about 1.5 times the mean-CPU time of the SIRJ PF (25 vs 17 seconds).

%UnP: The mean percentage of unique particle at last time index $t = T$ is higher for the LW PF. Overall, %uNp spans from around 86%–94% for the SIRJ PF, and from around 94%–97% for the LW PF. These results indicate that the LW PF suffers less the degeneracy problem, but we consider that the SIRJ behaves reasonably well too. Notice that, within each scenario and filter, for a time series length of size $T = 1000$, %uNp remains rather stable when increasing the number of particles used in the estimation procedure.

Sign of degeneracy? Observe that neither the estimated posterior PDF of the states (volatility), nor the posterior PDF of the model parameters show signs of degeneracy; at least up to $T = 1000$ for the model at hand. Be reminded that when filtering solely the states of this nonlinear SARV(1) model, no signs of degeneracy were observed up to $T = 2000$.

Number of particles: For the model at hand, to be on the safe side, we recommend using at least $N_p = 5000$ particles to end up with a reliable posterior. However, as a rule of thumb, when estimating simultaneously the states and parameters of the nonlinear SARV(1) model, $N_p = 10000$ will provide better mean-RMSE results irrespective of all simulation settings in consideration.

Impact of discount factor δ: In general, lower RMSE values are attained at the lowest setting of the discount factor, say at $\delta = 0.83$ (vs $0.95, 0.99$). Hereafter, unless stated otherwise, only $\delta = 0.83$ will be used.

Non-suitable filters: As mentioned, for the nonlinear SARV(1) state-space model at hand none of the two nonlinear Kalman-based PF variants (EPFJ, UPFJ) explored in this work are suitable; see Acosta and Muñoz (2007).

Following, we present the results, remarks, and conclusions regarding the application – to volatile financial data – of our proposed SIRJ particle filter variant and the widely used LW particle filter variant; the latter is taken as a benchmark. We remind that the stratified resampling scheme is adopted and a discount parameter of value $\delta = 0.83$ is chosen.

6.7 Application to Volatility in Financial Data

This section deals with the application of the SIRJ and LW particle filter variants to two real data sets from the financial area, where the nonlinear SARV(1) model is adopted. In other words, the present section aims to apply these two filters to simultaneously estimate the latent states (volatility) and unknown parameters $\Theta = (\mu, \phi, \sigma_\eta^2)'$ of the nonlinear SARV(1) model, specified in equations (6.3) and (6.4), under the assumption of Gaussian measurement and transition noises.

Be reminded that our empirical work is based on weekday daily returns of the Spanish financial index IBEX 35 and the Europe Brent spot prices. Both time series were taken in the period from January 2002 to July 2012 comprising 2670 and 2669 observations, respectively. These two real data sets are already presented in Section 6.2 to illustrate the stylized features of volatile financial returns series. The reader may refer to Table 6.1 for summary statistics and to Figures 6.1 and 6.2 for corresponding graphical displays.

Next, we present the filtering estimation results and main findings, first for the IBEX 35 data set and later on for the Brent data set. Before continuing, we remark that the choice of the nonlinear SARV(1) model is based on its known importance to model stochastic volatility within the financial markets. That is, the undergone applications represent a set of application examples to illustrate the estimation ability of the particle filters studied, but it is not our aim in this work to provide a procedure for model selection nor to exploit in depth the economical aspects of the two return series studied. We, however attempt to provide –apart from the statistical estimation results– some economical explanation of the results found.

6.7.1 Application to the IBEX 35 Data: Results and Remarks

Herein, we report the estimation results and related remarks regarding the implementation of the two particle filter variants studied to estimate the latent states (volatility) and the unknown parameters (mean level, persistence, volatility of volatility) that underlie in the daily returns of the IBEX 35 data.

The SIRJ particle filter variant was shown (via MC studies) to be able to equate the statistical performance of the benchmark and widely used LW PF variant, and with a lower computational cost. Thus, our aim is twofold: to verify previously found results and to illustrate the estimation ability of both particle filter variants when using real data from the financial area.

To better portray the estimated filtering results obtained for the IBEX 35 returns series, a table and a series of plots are constructed. Figure 6.14 depicts in the top panel, the evolution of the posterior states' estimates (measure of volatility expressed as $\hat{\sigma}_v = \exp(\hat{x}_t/2)$) yield by the SIRJ (grey/continuos) and the benchmark LW (black/continuous) filter. The bottom panel shows the evolution of the IBEX 35 returns during the period under study. Focusing on the top panel of this figure, we verify that both particle filters under study show practically the same behavior when estimating the states (volatility). In fact, we can barely distinguish one color from the other, meaning that there is close agreement in the statistical performance of our SIRJ particle filter and the LW particle filter when filtering the states (volatility).

Taking a global look at Figure 6.14, we find evidence that the estimated measure of volatility values $\hat{\sigma}_t$ reasonably captures the dynamic behavior of variability present in the IBEX 35 returns. Notice that periods where the returns (bottom panel) show higher volatility are reflected (in top panel) by the higher spikes in the estimates of volatility of returns $\hat{\sigma}_t$. In the same fashion, periods in which the returns show lower volatility do not have such large spikes in the estimates of volatility of returns $\hat{\sigma}_t$. Additionally, observe how larger negative values are present in the IBEX 35 returns within the years 2008-2010 and notice that this period includes the origin of the world economic crisis and the official onset of the Spanish financial crisis which is still going on. Specifically, the IBEX 35 shows in January 2008 a negative return in excess of about 8%, in October a negative return in excess of about 10% (black Friday), and in May 2010 a negative return in excess of about 7%.

To assess how the unknown parameters estimates behave, we construct Table 6.9 and Figure 6.15. The table displays estimation results related to the posterior estimates of the unknown parameters μ, ϕ and σ_η^2 at chosen time-indexes (first available date at the start of the chosen year). For each filter and each parameter, we report the corresponding posterior mean estimate together with the credible interval (2.5th and 97.5th percentiles) and the length of the credible interval. As expected, these results are a subset of those graphically displayed in Figure 6.15 (on page 227), representing the evolution of estimated values (posterior mean) for the parameters $\Theta = (\mu, \phi, \sigma_\eta^2)'$ and corresponding sequential credible intervals. Specifically, the top, middle and bottom panels represent the paths of the estimated parameters μ, ϕ and σ_η^2, respectively. Notice that SIRJ and LW estimates are represented by grey and black colors, respectively. More specifically, the sequential posterior mean estimate of a parameter is depicted by a continuous line while the corresponding credible interval is represented by a dashed line. As discussed before, both particle filters under study show similar performance when estimating sequentially the states (volatility) of the SARV(1) model. A different picture emerges, however, when looking at the sequential estimates of the unknown parameters $\Theta = (\mu, \phi, \sigma_\eta^2)'$ for this nonlinear model. Following, we present an analysis of the estimation results for the parameters.

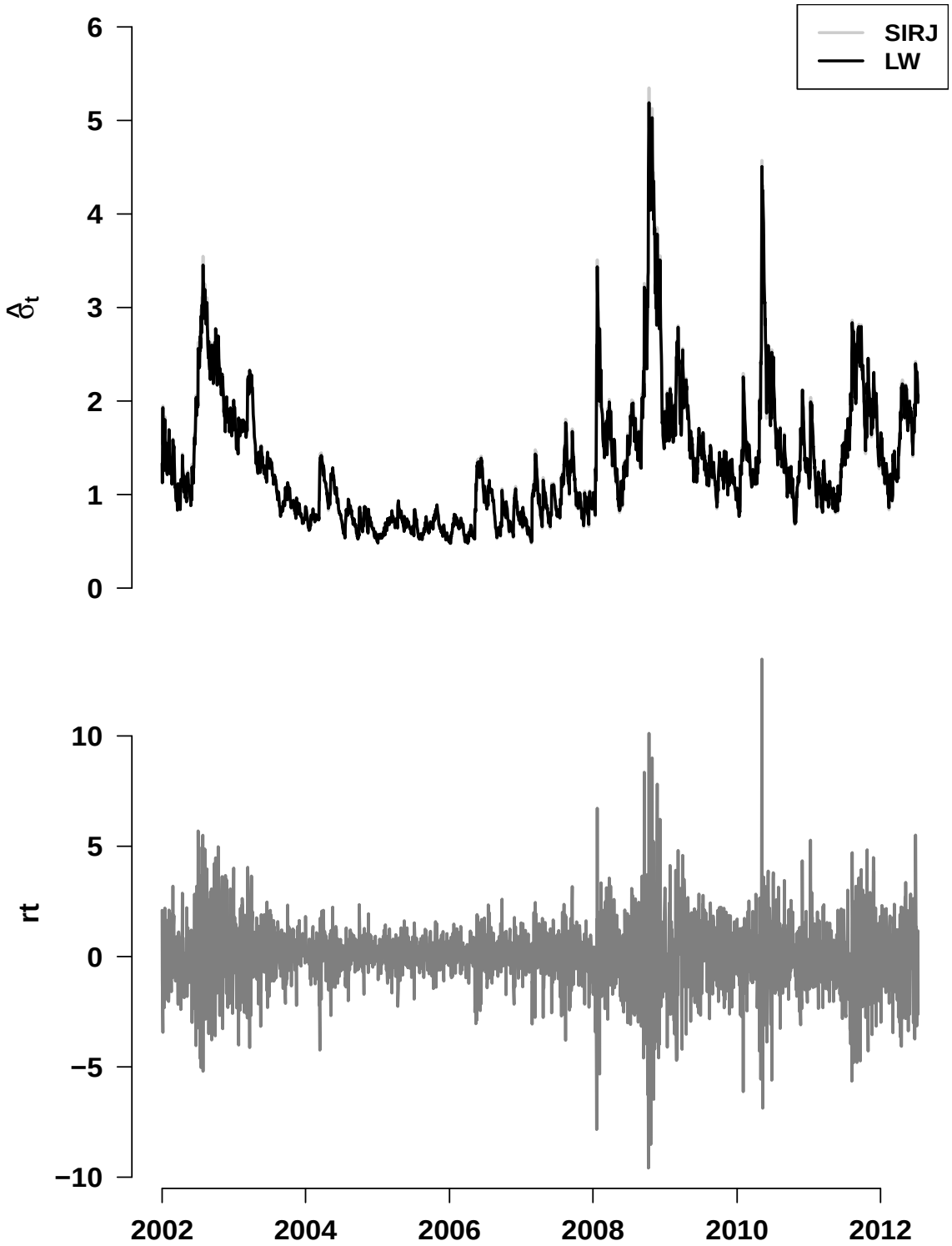


Figure 6.14: Nonlinear SARV(1) model fitted to the IBEX 35 returns. Top panel: Evolution of estimated posterior values of the states (measure of volatility expressed as $\hat{\sigma}_t = \exp(\hat{x}_t/2)$) yield by the SIRJ (grey/continuos) and the benchmark LW (black/continuous) particle filters; results shown for $N_p = 10000$ and discount factor $\delta = 0.83$. The bottom panel shows the IBEX 35 returns in the period under study.

Table 6.9: Evolution of estimated parameters $\Theta = (\mu, \phi, \sigma_\eta^2)'$ for IBEX 35 data and the two PF variants under study with $t \in \{250, 500, 1008, 1515, 2022, 2536, 2668, 2670\}$.

		LW			SIRJ		
Date (Day/Month/Year)	$\boldsymbol{\theta}$	**Mean**	**Cred. Int.***	$\boldsymbol{\Delta}_{\mathbf{CI}}^{**}$	**Mean**	**Cred. Int.**	$\boldsymbol{\Delta}_{\mathbf{CI}}$
January 2, 2003	μ	1.408	(-0.609, 3.466)	4.075	1.411	(-0.543, 3.339)	3.882
	ϕ	0.984	(0.954, 0.997)	0.043	0.982	(0.947, 0.997)	0.05
	σ_η^2	0.019	(0.006, 0.049)	0.043	0.019	(0.005, 0.047)	0.042
January 2, 2004	μ	0.444	(-1.039, 1.929)	2.968	0.426	(-0.783, 1.655)	2.438
	ϕ	0.987	(0.964, 0.997)	0.033	0.987	(0.963, 0.997)	0.034
	σ_η^2	0.016	(0.006, 0.034)	0.028	0.015	(0.004, 0.039)	0.035
January 3, 2006	μ	−0.448	(-1.737, 0.839)	2.576	−0.33	(-1.412, 0.746)	2.158
	ϕ	0.989	(0.972, 0.997)	0.025	0.989	(0.974, 0.997)	0.023
	σ_η^2	0.017	(0.007, 0.034)	0.027	0.016	(0.005, 0.037)	0.032
January 2, 2008	μ	−0.32	(-1.124, 0.494)	1.618	−0.346	(-1.378, 0.676)	2.054
	ϕ	0.984	(0.954, 0.997)	0.043	0.987	(0.971, 0.996)	0.025
	σ_η^2	0.028	(0.01, 0.066)	0.056	0.032	(0.012, 0.07)	0.058
January 4, 2010	μ	−0.012	(-0.716, 0.702)	1.418	−0.098	(-1.131, 0.942)	2.073
	ϕ	0.989	(0.977, 0.996)	0.019	0.99	(0.982, 0.996)	0.014
	σ_η^2	0.03	(0.017, 0.05)	0.033	0.031	(0.016, 0.054)	0.038
January 3, 2012	μ	0.163	(-0.472, 0.781)	1.253	0.005	(-0.993, 1.02)	2.013
	ϕ	0.989	(0.977, 0.995)	0.018	0.989	(0.981, 0.995)	0.014
	σ_η^2	0.031	(0.019, 0.047)	0.028	0.034	(0.02, 0.054)	0.034
July 10, 2012	μ	0.194	(-0.379, 0.778)	1.157	0.086	(-0.885, 0.998)	1.883
	ϕ	0.989	(0.977, 0.996)	0.019	0.99	(0.982, 0.995)	0.013
	σ_η^2	0.029	(0.017, 0.044)	0.027	0.031	(0.019, 0.05)	0.031
July 12, 2012	μ	0.19	(-0.383, 0.751)	1.134	0.083	(-0.853, 1.005)	1.858
	ϕ	0.989	(0.977, 0.996)	0.019	0.99	(0.982, 0.995)	0.013
	σ_η^2	0.029	(0.017, 0.044)	0.027	0.031	(0.019, 0.05)	0.031

* Credible interval: ($2.5th$, $97.5th$ percentiles); ** Interval length of credible interval.

For the IBEX 35 returns data, the SIRJ and LW sequential posterior mean-estimates are very similar for the mean level (μ) and transition noise σ_η^2 parameters, but differ more in some periods for the persistence parameter ϕ. Specifically, both filters show some disagreement in estimating the persistence parameter from the beginning of the study period until around year 2004, closely agree within the period from 2004 until 2006, differ again from 2006 until around the end of 2008, but agree again towards the end of the study period (July, 2012). We observe that the disagreement between filtering estimates –yielded by both, the SIRJ and the benchmark LW particle filter– becomes more obvious when focusing on the estimated credible intervals.

These results go in parallel with what is stated in Stroud, Polson, and Müller (2004) in the sense that large negative returns "…can lead to filtered parameters estimates changing abruptly and provides a useful testing ground for particle filters". Based on results portrayed in top panel of Figure 6.14 and in Figure 6.15, we consider that the two implemented particle filters are able to react to large negative

returns by producing abrupt changes in the estimated values of some model parameters, but recover afterwards without suffering the sometimes unavoidable degeneracy problem; focus on bottom panel of last figure depicting the estimation of the state noise volatility σ_η^2. The general pattern observed in the evolution of the estimated parameters resembles the one shown in Stroud, Polson, and Müller (2004) that propose another particle filter variant called 'Practical filter', but contrary to them we find no signs of degeneracy as they report for the S&P500 data. Next, as done throughout this work, we somehow quantify the degree of degeneracy present in the particle filters studied.

Thus, for the SARV(1) model fitted to the IBEX 35 returns, we provide (in percentages) a measure of the unique number of particles uNp (%uNp) at last time-index $T = 2670$. For the SIRJ and LW PF variants, using $N_p = 10000$ particles, the yielded percentages are about 91% and 97%, respectively. Notice that the values obtained in this particular case for the unique number of particles lie within the range of values obtained through simulations in previous Section 6.6, refer to page 216. These results together with the graphical displays in Figure 6.16 empirically show that both particle filters studied –SIRJ and LW– avoid the inherent potential degeneracy drawback present in the particle filtering methodology. Hence, we consider that both competing particle filters lead to reliable estimated posterior distributions for the parameters. We find these results very encouraging, since –as nowadays is very well-known– the estimation of fixed parameters has posed (and still does) great difficulties, because many times the estimated filtering distributions either degenerate to a few or a single particle or suffer another kind of degeneracy called sample-impoverishment. The undergone application empirically shows that the Liu and West (2001) jittering strategy helps to avoid the degeneracy drawback.

Finally, for both particle filters we quantify the computational cost by reporting the elapsed CPU time to handle the sequential simultaneous estimation of states (volatility) and parameters for the SARV(1) model fitted to the IBEX 35 data set of length $T = 2670$. Hence, for the SIRJ and LW PF variants, using $N_p = 10000$ particles and $T = 2670$ observations, the obtained elapsed CPU-times (in seconds) are about 125.2 and 196.89, respectively. This means that in this particular case, the LW PF is about 1.6 times slower than the SIRJ.

Next, on page 229, we present the estimation results, remarks and conclusions regarding the application of the competing SIRJ and LW particle filters to the Europe Brent returns series.

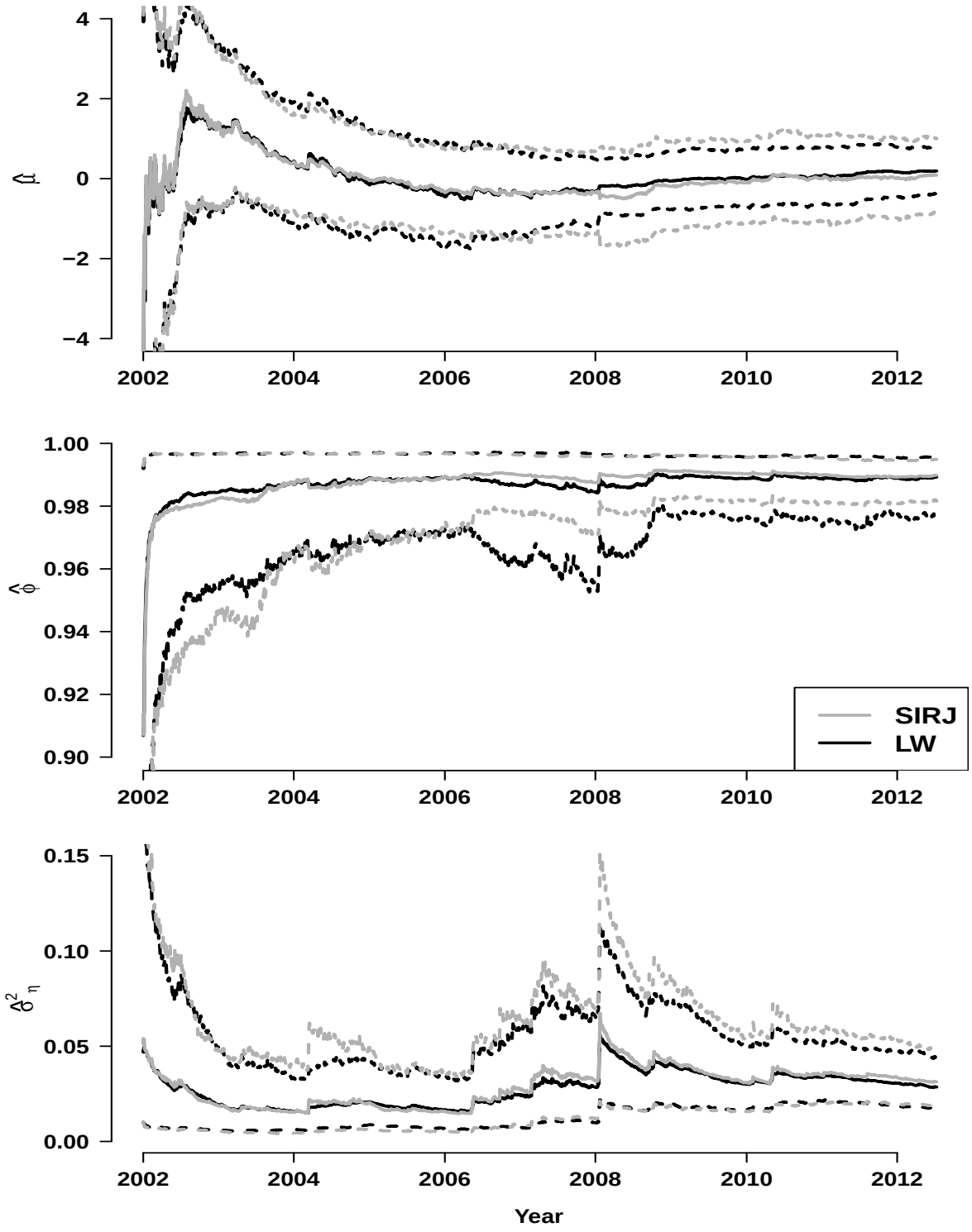


Figure 6.15: Nonlinear SARV(1) model fitted to the IBEX 35 returns. Evolution of estimated values (posterior mean) of the model parameters: μ (level parameter/top panel), ϕ (persistence parameter/middle panel) and volatility variance parameter σ_η^2 (bottom panel) yield by the SIRJ (grey/continuos) and the benchmark LW (black/continuous) particle filters. Corresponding 95% credible intervals (2.5th and 97.5th percentiles) for the parameters are represented by grey/dashed lines for the SIRJ (black/dashed for LW). Results are shown for $N_p = 10000$ and discount factor $\delta = 0.83$.

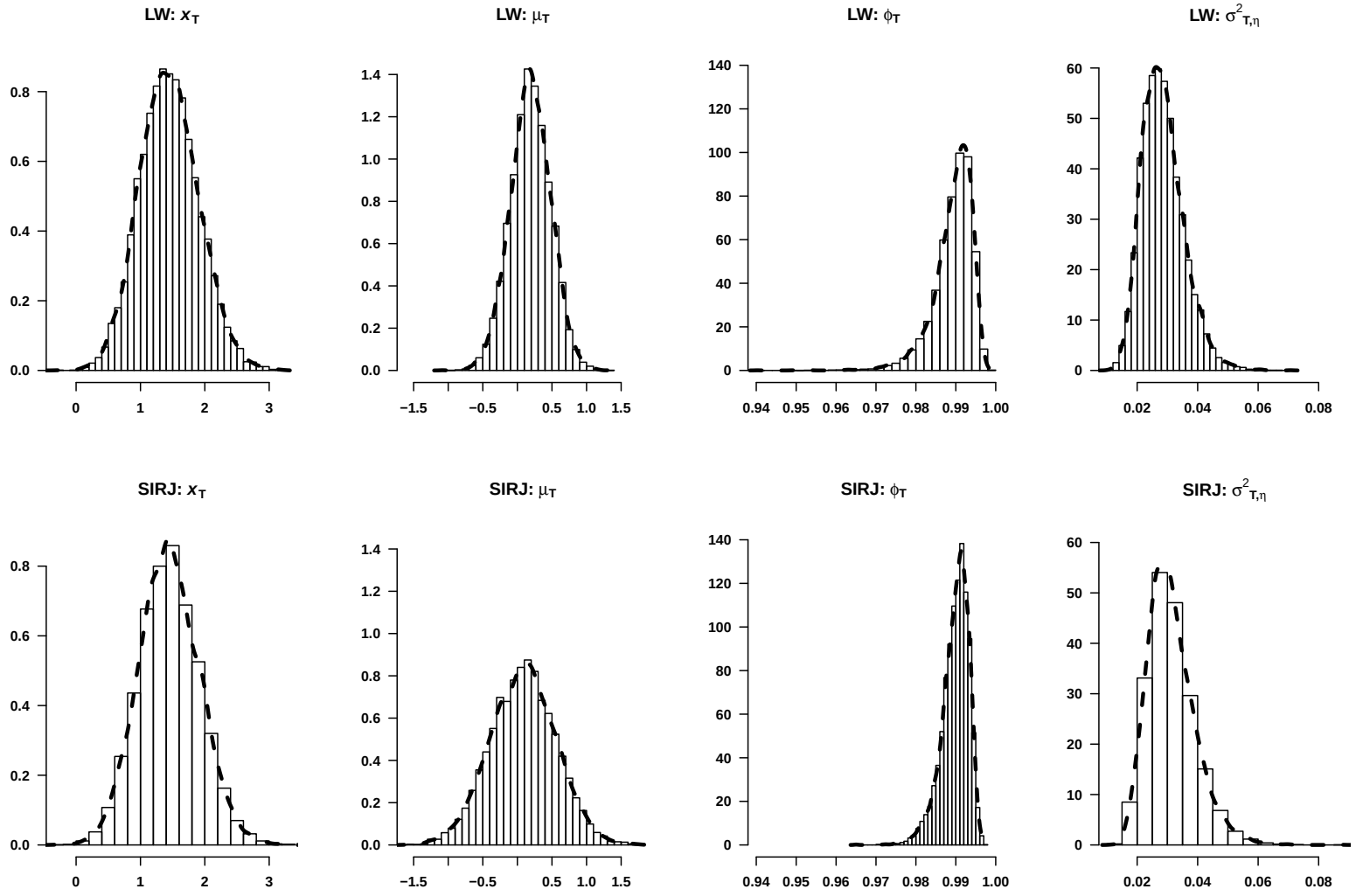


Figure 6.16: Illustration of non degeneracy in SIRJ and LW PF variants at last time index $T = 2670$: Histograms (together with overlaid estimated density curves) representing the estimated posterior distributions of: the states (first column), the level parameter μ (second column), the persistence parameter ϕ (third column) and the transition noise variance σ^2_η (fourth column) for the SARV(1) model fitted to the IBEX 35 returns. Notice that each row refers to a different PF variant. Results shown for $N_p = 10000$ and discount factor $\delta = 0.83$.

6.7.2 Application to the Brent Data: Results and Remarks

Herein, likewise for the IBEX 35 returns data, we report the estimation results and related remarks regarding the implementation of the two particle filter variants studied to estimate the latent states and the three unknown parameters that underlie in the daily returns of the Brent data. Thus, the same type of table/figures presented before for the IBEX 35 estimation results are constructed for the Brent data at hand.

For the Brent return series, Figure 6.17 displays in the top panel, the evolution of the posterior states' estimates (measure of volatility expressed as $\hat{\sigma}_v = \exp(\hat{x}_t/2)$) yield by the SIRJ (denoted by a grey/continuos line as said before) and the benchmark LW (black/continuous) filter. The bottom panel shows the evolution of the Europe Brent returns during the period under study. Focusing on the top panel of this figure, it is also verified that both particle filters under study show practically the same behavior when estimating the states (volatility) for the fitted nonlinear SARV(1) model. In fact, we can barely distinguish one color from the other, meaning that there is close agreement in the statistical performance of our SIRJ particle filter and the widely used LW particle filter when filtering the states (volatility).

Taking a global look at Figure 6.17, we also find evidence that the estimated volatility of returns $\hat{\sigma}_v$ (top panel) reasonably reflects the periods of higher volatility present in Brent returns (bottom panel); we observe how periods where Brent returns are more volatile coincide with higher spikes on the estimated volatility returns $\hat{\sigma}_v$. Notice that for the Brent data, the three largest negative returns (in excess of about 12-17%) are produced within years from 2008 to 2010, which coincide with the onset and first years of the still present economic crisis. Another period where large negative returns (in excess of about 8-9%) are present is within years from 2003 through 2007. With the aim of providing an economic explanation, we look for some key events occurring in these years and that maybe responsible for the high spikes of volatility observed; such key events are framed within two major theories: the market law of supply and demand and the speculation. Next, we provide a description of those related market-key-events occurred during the years from 2003 to 2012.

2003 – 2008 The oil price starts rising in 2003 (Iraq invasion takes place), reaches the psychological barrier of 60 $ in 2005, continues to rise up to around 70 $ in 2006 (Israel-Lebanon war takes place), and it is in 2007 that the oil price begins to escalate until reaching a historic maximum value of about 144 $ in 2008. The reasons for this escalation are naturally multifactorial, but as aforementioned are believed to be related to the law of supply and demand on the one hand, and speculation on the other. It is said that in this period, countries like China and India began to demand more oil, but there was less supply. Besides, it is also believed that the so-called "fear of peak oil" made speculation to flourish, raising the spot price of oil.

2008 – 2012 The collapse of oil prices seems to be related to the onset of the economic crisis in 2008. This drop in oil prices after the crisis is believed to be related to the so-called credit crunch and the entry of many countries into a recession, resulting in a reduction in oil demand.

To assess how the unknown parameters estimates behave, we construct Table 6.10 and Figure 6.18. The table displays estimation results related to the posterior estimates of the unknown parameters μ, ϕ and σ_η^2 at chosen time-indexes (first available date at the start of the chosen year). For each filter and each parameter, we report the corresponding posterior mean estimate together with the credible interval (2.5th and 97.5th percentiles) and the length of the credible interval. As expected, these results are a subset of those graphically displayed in Figure 6.18, representing the evolution of estimated values (posterior mean) for the parameters $\Theta = (\mu, \phi, \sigma_\eta^2)'$ and corresponding sequential credible intervals. Specifically, the top, middle and bottom panels represent the paths of the estimated parameter μ, ϕ and σ_η^2, respectively. Notice that SIRJ and LW estimates are represented by grey and black colors, respectively. More specifically, the sequential posterior mean estimate of a parameter is depicted by a continuous line while the corresponding credible interval is represented by a dashed line.

Table 6.10: Evolution of estimated parameters $\Theta = (\mu, \phi, \sigma_\eta^2)'$ for Brent data and the two PF variants under study with $t \in \{255, 513, 1031, 1536, 2041, 2541, 2669\}$.

		LW			**SIRJ**		
Date (Day/Month/Year)	$\boldsymbol{\theta}$	**Mean**	**Cred. Int.***	$\boldsymbol{\Delta_{\text{CI}}^{**}}$	**Mean**	**Cred. Int.**	$\boldsymbol{\Delta_{\text{CI}}}$
January 2, 2003	μ	1.374	(0.165, 2.574)	2.409	1.4	(0.109, 2.689)	2.58
	ϕ	0.96	(0.855, 0.994)	0.139	0.961	(0.867, 0.994)	0.127
	σ_η^2	0.028	(0.007, 0.076)	0.069	0.028	(0.005, 0.087)	0.082
January 2, 2004	μ	1.502	(0.771, 2.231)	1.46	1.465	(0.793, 2.15)	1.357
	ϕ	0.943	(0.796, 0.992)	0.196	0.953	(0.856, 0.991)	0.135
	σ_η^2	0.023	(0.007, 0.057)	0.05	0.027	(0.009, 0.062)	0.053
January 3, 2006	μ	1.393	(1.023, 1.754)	0.731	1.455	(1.109, 1.809)	0.7
	ϕ	0.884	(0.621, 0.982)	0.361	0.918	(0.795, 0.976)	0.181
	σ_η^2	0.02	(0.007, 0.044)	0.037	0.023	(0.009, 0.047)	0.038
January 2, 2008	μ	1.232	(1.013, 1.45)	0.437	1.282	(1.076, 1.5)	0.424
	ϕ	0.846	(0.563, 0.967)	0.404	0.908	(0.799, 0.967)	0.168
	σ_η^2	0.017	(0.007, 0.034)	0.027	0.018	(0.008, 0.036)	0.028
January 4, 2010	μ	1.433	(1.219, 1.647)	0.428	1.4	(1.198, 1.599)	0.401
	ϕ	0.969	(0.948, 0.983)	0.035	0.969	(0.945, 0.984)	0.039
	σ_η^2	0.03	(0.016, 0.051)	0.035	0.03	(0.016, 0.052)	0.036
January 3, 2012	μ	1.371	(1.168, 1.577)	0.409	1.353	(1.153, 1.551)	0.398
	ϕ	0.967	(0.943, 0.982)	0.039	0.965	(0.935, 0.983)	0.048
	σ_η^2	0.031	(0.019, 0.05)	0.031	0.029	(0.015, 0.052)	0.037
July 10, 2012	μ	1.352	(1.144, 1.557)	0.413	1.344	(1.141, 1.544)	0.403
	ϕ	0.969	(0.945, 0.984)	0.039	0.967	(0.942, 0.983)	0.041
	σ_η^2	0.031	(0.019, 0.047)	0.028	0.028	(0.014, 0.049)	0.035

* Credible interval: $(2.5th, 97.5th$ percentiles); ** Interval length of credible interval.

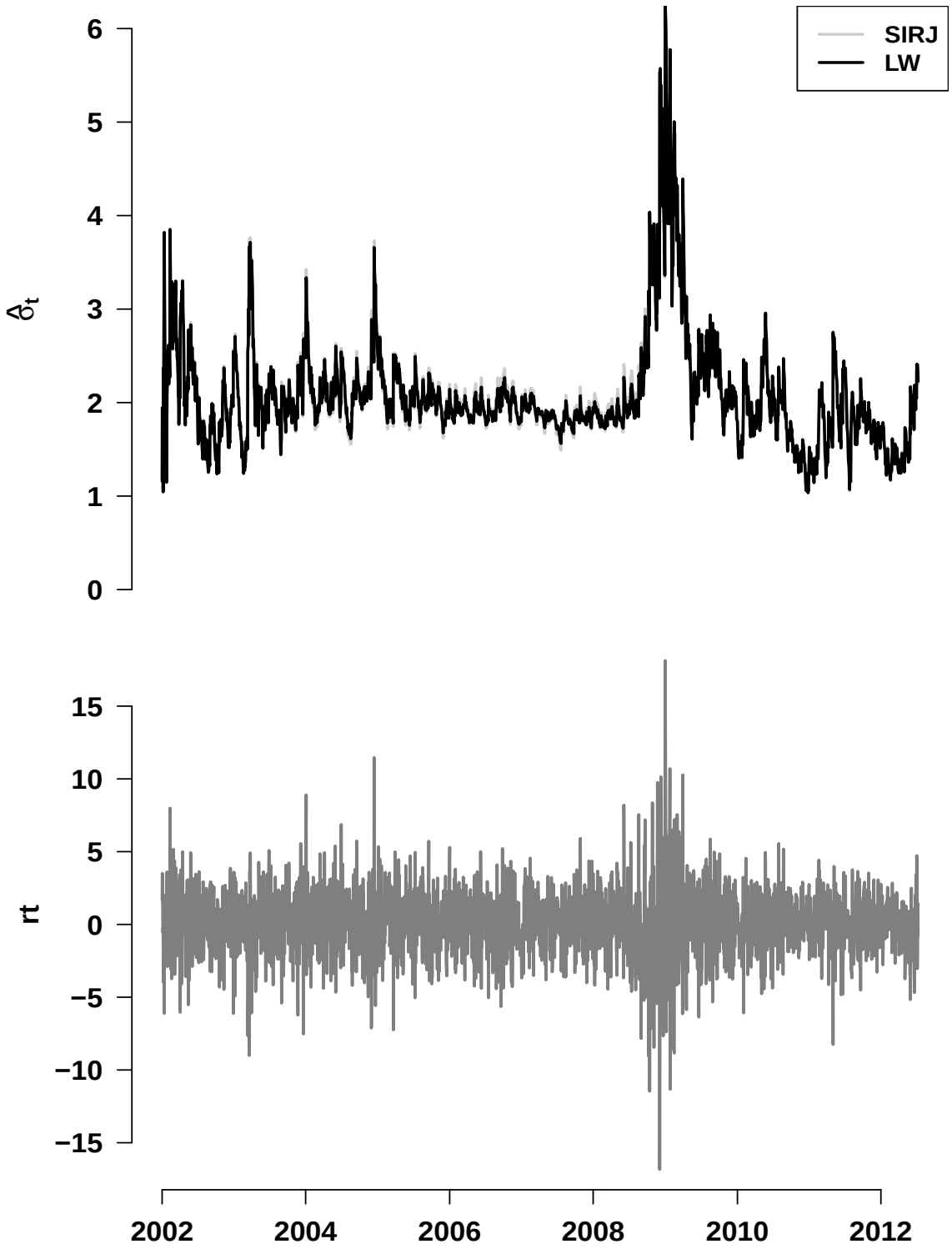


Figure 6.17: Nonlinear SARV(1) model fitted to the Brent returns. Top panel: Evolution of estimated posterior values of the states (measure of volatility expressed as $\hat{\sigma}_t = \exp(\hat{x}_t/2)$) yield by the SIRJ (grey/continuos) and the benchmark LW (black/continuous) particle filters; results shown for Np=10000 and discount factor $\delta = 0.83$. The bottom panel shows the Brent returns in the period under study.

Although both particle filters under study –the SIRJ and the LW PF– show similar statistical performance when estimating the states (volatility) of the nonlinear SARV(1) model, some differences emerge when looking at the sequential parameters estimates; being more obvious for the persistence parameter ϕ. Following, we present an analysis of the estimation results for the parameters.

For the Brent returns data, the SIRJ and LW posterior mean estimates are very similar for the parameters μ and σ_η^2, but differ clearly in some periods for the persistence parameter ϕ. Specifically, both filters show close agreement in estimating the persistence parameter at the beginning of the period of study, start to differ in year 2003 until the end of 2008, and since then agree again until the end of the study period (July, 2012). We observe that the disagreement between these filtering estimates –yielded by both, the SIRJ and the benchmark LW particle filter– becomes even more obvious when focusing on the estimated credible intervals.

Based on results portrayed in top panel of Figure 6.17 and in Figure 6.18, we consider that both implemented particle filters are able to react to large negative returns by producing high spikes on the estimated states (volatility) and also abrupt changes in the estimated values of some model parameters, but recover afterwards without suffering the sometimes unavoidable degeneracy problem; focus on bottom panel of last figure depicting the estimation of the state noise volatility σ_η^2. The general pattern observed in the evolution of the estimated parameters resembles the one shown in Stroud, Polson, and Müller (2004) that propose another particle filter variant called 'Practical filter', but contrary to them we find no signs of degeneracy as they report for the S&P500 data. We remark that contrary to the IBEX 35 returns data, the differences observed between the SIRJ and the LW particle filters when estimating the persistence parameter are more pronounced for the Brent returns data. Next, as done throughout this work, we somehow quantify the degree of degeneracy present in the particle filters studied.

Thus, for the SARV(1) model fitted to the Brent returns, we provide (in percentages) a measure of the unique number of particles uNp (%uNp) at last time-index $T = 2669$. For the SIRJ and LW PF variants, using $N_p = 10000$ particles, the yielded percentages are about 93% and 97%, respectively. Notice that the values obtained in this particular case for the unique number of particles lie within the range of values obtained through simulations in previous Section 6.6, refer to page 216. These results together with the graphical displays in Figure 6.19 empirically show that both particle filters studied –SIRJ and LW– avoid the inherent potential degeneracy drawback present in the particle filtering methodology. Hence, we consider that both competing particle filters lead to reliable estimated posterior distributions for the parameters. Once again, it is empirically shown that the Liu and West (2001) jittering strategy within the context of particle filtering can be able to avoid the degeneracy drawback.

Finally, for both particle filters we quantify the computational cost by reporting the elapsed CPU time to handle the sequential simultaneous estimation of states (volatility) and parameters for the SARV(1) model fitted to the Europe Brent data set of length $T = 2669$. For the SIRJ and LW PF variants, using $N_p = 10000$ and $T = 2669$, the obtained elapsed CPU times in seconds are about 124.9 and 192.7, respectively. This means that in this particular case, the LW PF is about 1.5 times slower than the SIRJ.

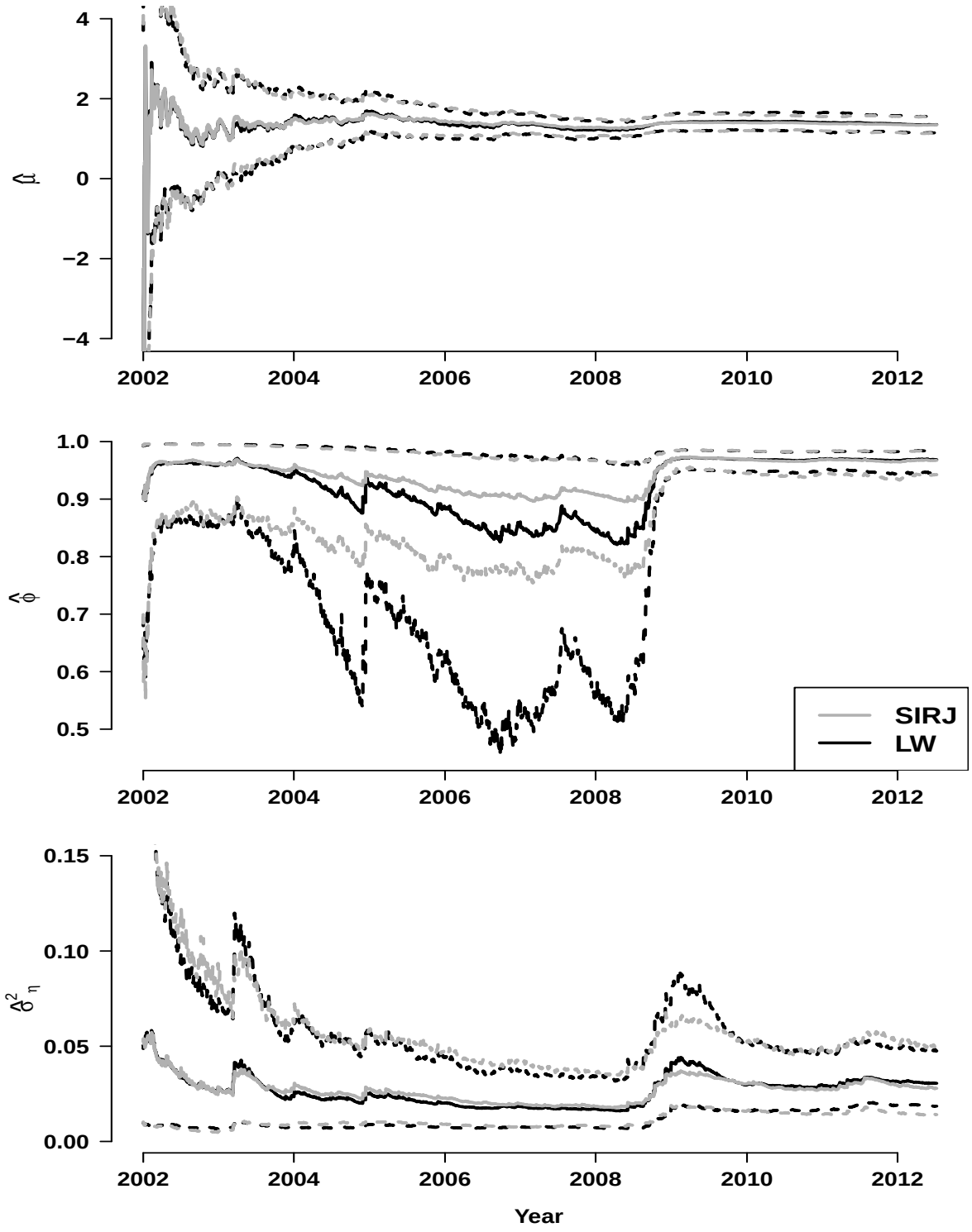


Figure 6.18: Nonlinear SARV(1) model fitted to the Brent returns. Evolution of estimated values (posterior mean) of the model parameters: μ (level parameter/top panel), ϕ (persistence parameter/middle panel) and volatility variance parameter σ_η^2 (bottom panel) yield by the SIRJ (grey/continuos) and the benchmark LW (black/continuous) particle filters. Corresponding 95% credible intervals (2.5th and 97.5th percentiles) for the parameters are represented by grey/dashed lines for the SIRJ (black/dashed for LW). Results are shown for $N_p = 10000$ and discount factor $\delta = 0.83$.

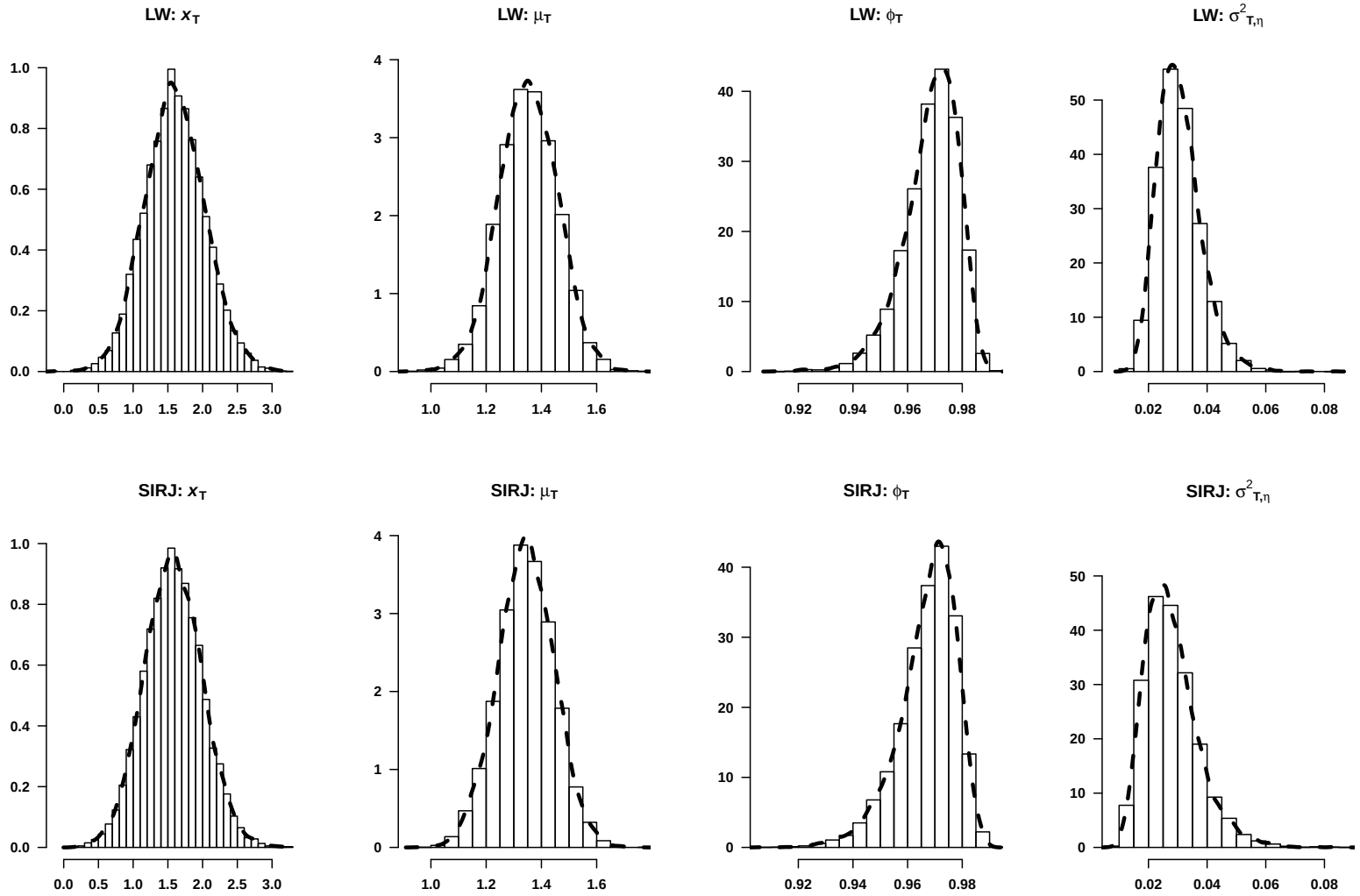


Figure 6.19: Illustration of non degeneracy in SIRJ and LW PF variants at last time index $T = 2670$: Histograms (together with overlaid estimated density curves) representing the estimated posterior distributions of: the states (first column), the level parameter μ (second column), the persistence parameter ϕ (third column) and the transition noise variance σ^2_η (fourth column) for the SARV(1) model fitted to the Brent returns. Notice that each row refers to a different PF variant. Results shown for $N_p = 10000$ and discount factor $\delta = 0.83$.

6.7.3 SARV(1) Model Validation

For model diagnostics we can use simple and easy-to-use statistical tools to check model adequacy. The model diagnostics are based on the standardized observations, denoted as $e_t = y_t/\sigma_t$, where σ_t is the estimated volatility measure. Notice that the return volatility estimates σ_t, are themselves based on the estimates of the parameters $\Theta = (\mu, \phi, \sigma_\eta^2)'$ in the system equation for the nonlinear SARV(1) model.

First, for both data sets (IBEX 35 and Brent), Table 6.11 displays a summary statistics of corresponding residuals, including skewness and kurtosis. Second, Normal quantile-to-quantile plots (Q-Q plots) for the residuals e_t are depicted in Figure 6.20. These Q-Q plots together with measures of skewness and kurtosis are used to check the validity of the distributional assumptions made. Third, to complement the valuable information provided by Normal Q-Q plots and measures of skewness and kurtosis, some statistical tests are entertained on the residuals. Finally, to check whether residuals are autocorrelated or not, the Ljung-Box Q-statistic (at the twentieth lag) is computed for the residuals and squared residuals. A result of an statistical test when found significant at a significance level $\alpha = 0.05$, is indicated by the symbol '*'.

The validation of the fitted SARV(1) model to both data sets (IBEX 35 and Brent) is based on the analysis of residuals obtained via the SIRJ and LW PF variants. The main results are:

Skewness: The IBEX 35 residuals show significant skewness values for both particle filters in consideration. The skewness for Brent residuals, however, is found to be non-significant.

Kurtosis: The IBEX 35 residuals show significant kurtosis values for both particle filters. The kurtosis for Brent residuals, however, is found to be non-significant.

Q(20) and Q2(20): The test results lead to the non-rejection of the null hypothesis of serially uncorrelated residuals (Lag= 20), for the two data sets and two particle filters.

Extreme residuals: In both cases, the model seems to be robust from extreme values. Observe that only one (0.04%) of the Brent residuals obtained via the LW PF variant is greater than 3.5 standard deviations; this value is about 3.6.

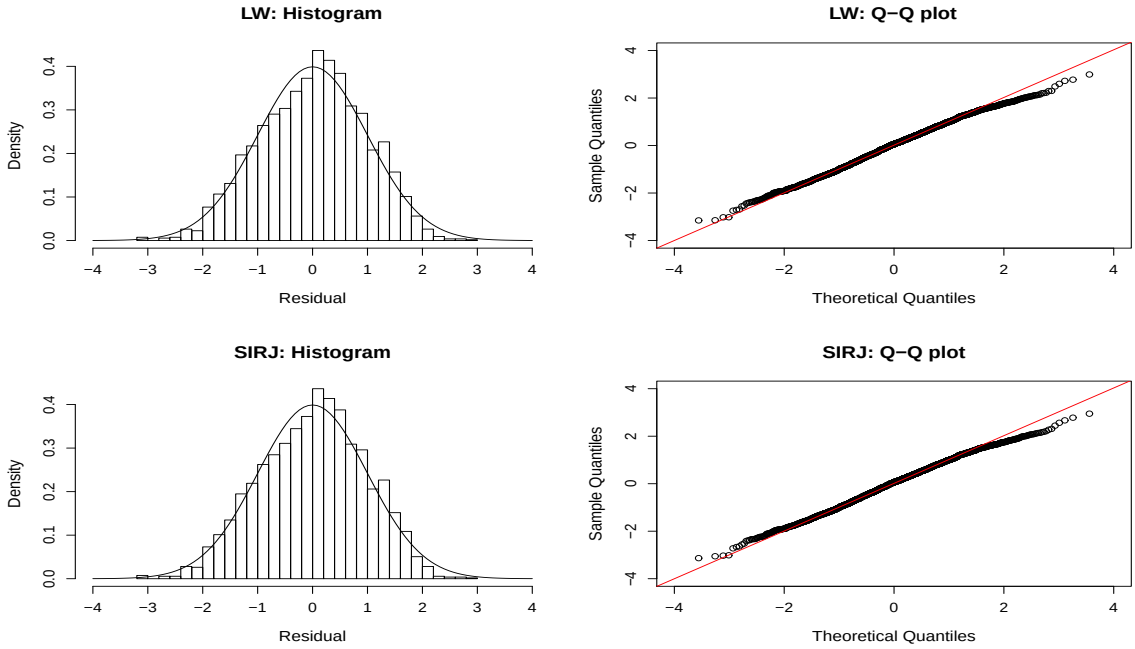
Based on the above results of skewness and kurtosis, the IBEX 35 residuals do not meet the assumption of normality for the measurement noise; see also the Normal Q-Q plots. The Brent residuals are reasonably normally distributed as indicated by results of skewness and kurtosis; see also the Normal Q-Q plots. We conclude that the SARV(1) model provides a good fit for the Brent data set. In contrast, for the IBEX 35, the fitted SARV(1) model with Gaussian measurement noise confirms to be non appropriate to capture completely the kurtosis. Notice, however, that the kurtosis decreases more than two thirds from 9.04. In none of the cases, a significant serial correlation is found.

Table 6.11: Summary statistics of daily returns residuals of the Spanish IBEX 35 financial index and the Europe Brent spot price

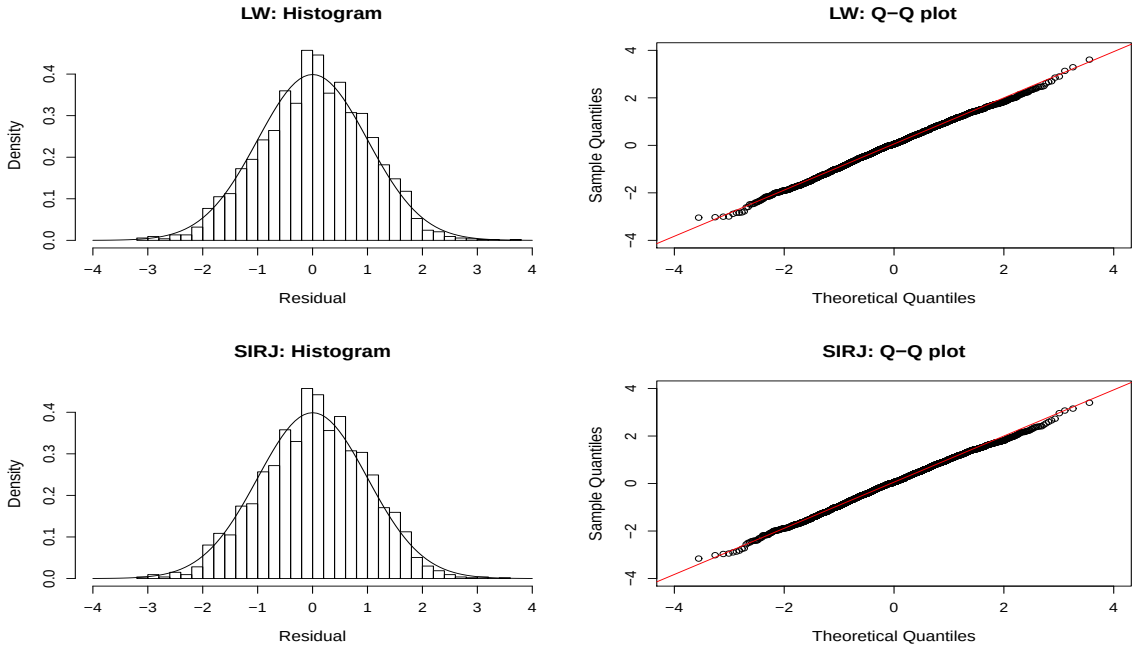
	IBEX 35 (n = 2670)			**Brent (n = 2669)**		
		Residuals			**Residuals**	
Statistics	r_t	**LW**	**SIRJ**	r_t	**LW**	**SIRJ**
Mean	−0.009	0.01	0.01	0.06	0.038	0.038
Stdev	1.549	0.957	0.952	2.268	0.964	0.958
Minimum	−9.586	−3.153	−3.134	−16.832	−3.044	−3.16
Maximum	13.484	2.991	2.948	18.13	3.608	3.401
Skewness	0.151*	−0.141*	−0.139*	−0.029	−0.104	−0.108
Kurtosis	9.04*	2.644*	2.628*	7.474*	2.916	2.88
Autocorrelations r_t						
Q(20)[b]	56.407*	24.933	24.791	47.053*	19.467	19.397
Autocorrelations r_t^2						
Q2(20)[b]	1529.925*	20.198	21.018	922.903*	27.078	26.559
N. obs (%) > |3.5|		0 (0%)	0 (0%)		1 (0.04%)	0 (0%)

[a] r_t: denotes the return series at time t
[b] Q(20) and Q2(20): Ljung-Box statistics (Lag= 20) to test the autocorrelation of the original and squared returns series: r_t and r_t^2, respectively (critical value = 31.41)
[c] r2(k): Order k autocorrelation of squared observations r_t^2
* Significant at %5 level

Next chapter presents the final discussion and future lines of research.



(a) IBEX 35 residual analysis



(b) Europe Brent residual analysis

Figure 6.20: Q-Q plots and histograms of the residuals for a SARV(1) model estimated via the SIRJ and LW particle filters: Europe Brent data (top) and IBEX 35 data (bottom).

CHAPTER 7

DISCUSSION, CONTRIBUTIONS, AND FUTURE LINES OF RESEARCH

The present chapter deals with the discussion, summary of contributions and future lines of research. We not only highlight the strengths of the particle filtering methodology, but also pinpoint its limitations.

7.1 Discussion

This work considers the adoption of the particle filtering methodology to tackle the estimation of the states as well as of states and parameters simultaneously for linear/nonlinear, Gaussian/non-Gaussian, stationary/non-stationary dynamic state-space models. The latter, the simultaneous estimation of states and fixed model parameters, is a more challenging problem and still a very active area of research; see, for instance, Niemi (2009), Carvalho, Johannes, Lopes, and Polson (2010), Andrieu, Doucet, and Holenstein (2010), Lopes and Tsay (2011), and Doucet and Johansen (2011). The present work deals with univariate latent states and a multivariate vector (order up to three) of fixed model parameters.

Following, one finds a discussion of how the jittering ideas arrive as a possible solution to the so-called sample impoverishment drawback.

7.1.1 How Do the Jittering Ideas Arrive?

The particle filtering methodology is proven to be –since around 1993– a very reliable approach for filtering solely the states; see references in Table 2.2 of this dissertation and also Doucet (1998), Hürzeler and Künsch (1998), Fearnhead (1998), Doucet, Godsill, and Andrieu (2000), Doucet, de Freitas, and

Gordon (2001), and Arulampalam, Maskell, Gordon, and Clapp (2002). Up to our knowledge, one of the first attempts to use a particle filter for the simultaneous estimation of states and model parameters is due to Kitagawa (1998) that proposes the so-called Self Organizing filter (SO); see Algorithm 11 in Chapter 5 (on page 141). We believe that this filter represents an important step towards attempting to solve the still difficult problem of estimating simultaneously the states and *fixed* parameters of a dynamic state-space model, but though the method might yield good results it usually suffers degeneracy in a very high degree due to the presence of fixed parameters. Indeed, degeneracy becomes more acute when dealing with the simultaneous estimation of the original state vector and fixed model parameters, because in this situation the particles associated with the parameters do not adequately regenerate and may end up stacked in a small and possibly wrong subregion of the parameters state-space support. For instance, when attempting to estimate the autoregressive parameter ϕ of the AR(1) plus noise model (as illustrated in Chapter 5), experimental results confirm that when one applies the SO particle filter, the estimated parameter collapses to a unique value despite the large number (20000) of particles used, resulting in an inadequate posterior distribution. This drawback is even pointed out in Kitagawa and Sato (2001) where the SO particle filter variant is revisited and where the authors state that the SO particle filter variant originally introduced in Kitagawa (1998) was not really able to cope with the fixed parameter estimation problem unless an artificial noise is added to the parameters evolution model. As a possible solution, they propose to add such an artificial noise in the form suggested already by Gordon, Salmond, and Smith (1993) in the context of states estimation only.

The proposed solution does give rise to an improved version of the original SO particle filter variant introduced by Kitagawa (1998), but it also presents some limitations. When assuming a dynamic evolution for the fixed parameters, the researcher must choose the magnitude of the variance to impose in the added noise term, but how exactly this should be achieved without introducing the so-called loss of information did not remain clear at that time. Questions arise like how small the variance of the added artificial noise must be or whether it should also be estimated as part of the already augmented state vector, not to mention that its election asks for more researcher expertise. We remark that: (1) the potential degeneracy drawback is not exclusive to the SO particle filter; the same problem would occur if one uses the augmented state vector and applies the ASIR particle filter variant for the simultaneous estimation of states and parameters (described in Chapter 2), and (2) the original SO particle filter works adequately if the model parameters are truly time–varying, as shown in Kitagawa et al. (2002).

The addition of this artificial noise certainly prevents the parameters' particles to collapse to a few unique or even to a single particle. However, a criticism of combining the SO PF variant with the artificial evolution of the parameters in the form suggested in Gordon, Salmond, and Smith (1993), is that parameters are fixed but are imposed to be time varying. Additionally, how this can be undertaken without significantly changing the model at hand poses a problem, because by imposing fixed parameters to be time varying makes the variability to increase over time, and thus the loss of information problem arises. This is the main critic of some authors like Liu and West (2001) that do agree to add an artificial noise, but it should be done so as to avoid the loss of information. The solution to the sample impoverishment drawback proposed by these authors appeals to modified Kernel density es-

timation methods (originally used by West (1993) in another context) in order to diversify *old* particles and produce regenerated fixed model parameters particles. Specifically, they propose their particle filter variant named LW PF in this work.

The basic feature of the LW PF variant is the combination of the ASIR PF (proposed in 1999 by Pitt and Shephard to estimate solely latent states) with the addition of an artificial noise for the fixed parameters using Kernel smoothing via shrinkage, as detailed in Section 5.5 of Chapter 5. Thus, the addition of the diversification step using Kernel smoothing and shrinkage ideas –the jittering step– makes it possible to efficiently extend the use of existing particle filter variants for filtering only latent states to also estimate both latent states and fixed parameters. This PhD thesis takes this widely used classic PF variant as a benchmark.

Our incursion into the simulation-based methodology called particle filtering began with the papers of Kitagawa (1996) (which deals solely with filtering latent states) and Kitagawa (1998), the author of aforementioned SO PF variant, and this somehow justifies the origin of our proposed Sampling Importance Resampling plus Jittering (SIRJ) PF variant to estimate simultaneously latent states and fixed parameters of a previously specified dynamic state-space model. The SIRJ PF variant that we propose combines the SO PF of Kitagawa (1998) with the Kernel smoothing and shrinkage ideas in Liu and West (2001); see Section 5.6 of Chapter 5. The SIRJ is proven in this work to reach the same statistical performance of the widely used LW PF taken as a benchmark; the reader may refer to the MC studies and related results in Chapters 5 and 6, dealing with the linear but non-stationary Local Level model and the nonlinear SARV(1) model. We consider this our most relevant contribution. Additionally, in Chapter 5, we explore and study the combination of a Kalman-based PF variant, called by us KPFJ, in the context of a linear and Gaussian but non-stationary Local Level model obtaining reasonably good results as detailed there.

The jittering step thus aims to provide general efficiency and specifically attempts to avoid any of the two forms of degeneracy inherently present in particle filters: (1) the collapse of the particles to a single one, or (2) the collapse to a very few unique number of particles. As known, the degeneracy is more prone to occur when estimating simultaneously the states and fixed parameters, because being fixed they do not evolve dynamically and may be trapped in a small and possibly wrong part of their support region. Naturally, when a filter suffers greatly any form of degeneracy, it precludes obtaining good estimators of any interesting feature of the posterior distribution. In other words, the estimated posterior PDF (the main job of PFs) of the variable of interest (states and/or parameters) is not reliable and the filter is thus inefficient.

All together, the results obtained along this work prove that the ideas of Kernel smoothing via shrinkage render good results for diversifying fixed parameters particles and thus avoid (or at least postpone) the sample impoverishment drawback. At the same time, MC results suggest that these ideas could work well under particle filter variants apart from the ASIR; this lead us to propose the EPFJ (being the KPFJ a special case) and the UPFJ particle filter variants, though the latter one was not entertained in the present work and remains as a theoretical proposal for future work. We remark,

however, that a detailed pseudo-code of this filter and all other studied algorithms are provided and implemented; if a filter is later on not applied (or reported), it is because of its unsuitability due to the type of model at hand.

The various MC studies undertaken, which we consider exhaustive enough, yield the following general findings with regard to the key factors under investigation within the particle filtering methodology.

7.1.2 General Findings

This work also assesses the impact that key factors –such as the signal-to-noise ratio, the resampling scheme, the number particles, and the length of the time series data (T), the discount factor δ– may have in the filtering performance of the algorithms. Extensive MC studies are run to test the filtering performance of all (whenever suitable) filters under study in both a linear and in a nonlinear context. The comparison criteria is based mainly on the typically used RMSE criterion, but also on the elapsed CPU time, and on a measure specially defined to somehow quantify the degree of degeneracy (which is a measure of the unique number of particles at last time-step, denoted by %uNp) present in the particle filters. As known, efficient particle filter variants must have a reasonably large effective sample size (ESS) to produce a reliable posterior PDF, but is also important to quantify how distinct the (ending) number of particles are; our degree of degeneracy measure accounts for both cases. Thus, we consider that the need of a measure of degeneracy is justified per se, but also go in line with the suggestions presented in the discussion of paper Andrieu, Doucet, and Holenstein (2010). We find relevant to complement the RMSE results with such (or other) measure of degeneracy because sometimes filters show similar RMSE, but differ on the quality of the estimated posterior in terms of degeneracy, leading again to unreliable results. Particle filters estimate posterior PDFs and these must be first reliable enough to carry out any further computations. We find, however, that generally particle filter variants show worse performance since they suffer greatly the degeneracy with very few unique number of particles observed at the end of the time series. In principle, an increase of the number of particles used in the estimation procedure helps to fight the degeneracy issue, but we also found that some filters benefit of this strategy more than others. That is, the reaction of filters to an increase of the number of particles seem to be filter-dependent, and we even suggest that also the complexity of the model plays a role.

With regard to the resampling scheme, either the stratified or residual resampling implementation schemes are found to be a valid methodological option, since both are variance reduction techniques, which becomes even more desirable when the resampling step is performed more often or at every point of time. Notice that the use of resampling strategies that induce a reduction of the MC variation of the particles somehow compensates for the fact that the addition of a resampling scheme always increases the variance of the particles. Nowadays, particle filter variants based on multinomial resampling are known to be not as efficient as the residual and stratified resampling, because they are more subject to MC variation; see, for instance, Kitagawa (1996), Bolic (2004), and the discussion of Andrieu,

Doucet, and Holenstein (2010). Some authors still adopt a multinomial resampling scheme, but we believe (confirmed by our simulations and literature) that the choice of a variance reduction resampling strategy does not play a minor role in the efficiency of the filters.

As mentioned throughout this work, we choose to use the stratified resampling scheme –found to be the best when compared to the multinomial one by Kitagawa (1996)– for two reasons: (1) our incursion to the particle filtering methodology began with that paper and (2) it is generally computationally less costly than its counterpart residual resampling. Both, residual and stratified resampling are widely accepted and used schemes.

Referring to the impact of the time series length, we find that though more observations naturally bring more information, the use of the added information is both problem and type of filter-dependent. In general, as expected, an increase of the time-series length is related to a decrease of the RMSE, though sometimes the reduction is not so relevant. Likewise, an increase of the number of particles, as intuition suggests, generally produces more accurate estimations, but in some cases the impact is not that relevant. The simulation results show that the improvement yielded is also affected by other factors such as the complexity of the model (linear or not, Gaussian or not, stationary or not; type of nonlinearity), the chosen particle filter variant (fully adapted or not), and the specific simulation settings of the model at hand (signal-to-noise-ratio (SNR) values and specific values of involved parameters, for instance).

Be reminded that, within the non-simulation framework, the famous Kalman Filterprovides the best filtering solution in the presence of a linear and Gaussian state-space model with known model parameters. In the same framework and under nonlinearity, the EKF and the UKF could provide optimal solutions; the former when nonlinearity is not so high and the latter when higher nonlinearity is present. Notice, however, that both non-simulation based nonlinear filters might struggle with large departures of normality. Regardless of the fact that we conducted MC studies (Chapter 3 dealing solely with the state estimation of linear and Gaussian state-space models) using particle filters in cases where the Kalman filter is known to be optimal, we naturally recommend only to resort to the computationally more demanding particle filters when the Kalman-based filters prove to be unsuitable. The main reason for choosing to adopt the particle filtering methodology in such a linear and Gaussian context is that we aimed to closely study special issues related to particle filters in a context where we "knew" the filtering answer in exact form. That allowed us to characterize how different PF variants behave under case-scenarios defined, for instance, by different SNR settings, or when using a certain number of particles and/or time series length. We consider that those extensive simulations shed some light about how competing particle filters behave in terms of statistical performance and degree of degeneracy, which is at the end what determines the quality of the estimated posterior PDF. Additionally, it allowed us to envision the potential performance and use of the competing PF variants to estimate not only the states but also the involved model parameters.

When estimating solely the states (Chapters 3 and 4) via the adopted particle filtering methodology, the Monte Carlo results suggest that using $N_p = 5000$ works well (in terms of RMSE and degree of

degeneracy) irrespective of the SNR for all filters, except for the KPF and ASIR at extremely low and high SNR settings, respectively. The reader may refer to specific results/remarks in corresponding chapters, when interested in a particular PF variant; in some cases even a smaller number of particles could be used.

When the simultaneous estimation of states and parameters is a target (Chapters 5 and 6), either $N_p = 5000$ or $N_p = 10000$ is recommended, specially the latter value to guarantee a reliable enough estimation of the posterior PDF of the states and fixed parameters. Again, this number of particles could be less when using a particular particle filter variant or type of model at hand. Since this work deals with a univariate state vector and a multivariate vector of parameters (order up to three), we consider that the particle filters studied using the recommended number of particles are not too costly (in terms of CPU time and memory requirements). Naturally, there is a trade-off between more accuracy and computer requirements as N_p increases, which will surely get worse as the dimension of the state and/or the parameter vector increases.

A nice byproduct of the performed MC studies regarding the degeneracy issue is that we found that generally irrespective of the time series length and number of particles used in the simulation procedure the percentage of unique number of particles at last time-index t maintains rather stable within a specific particle filter variant and case scenario; clearly, as N_p increases the absolute number of unique particles also increases.

The study of the potential impact on estimation of the chosen value for the discount parameter δ applies only in Chapters 5 and 6, dealing with the estimation of fixed model parameters. The simulation results in Chapter 5 –dealing with the estimation of the latent state (level) and two unknown variance parameters for the linear and non-stationary local level model– indicate that there is practically no difference in the quality of the estimations if one uses a δ value of 0.83 or 0.95. Likewise, in Chapter 6 –dealing with the estimation of the states (volatility) and the three parameters (μ, ϕ, σ_η^2) of the nonlinear SARV(1) model– we consider δ values of 0.83, 0.95, and 0.99 and find that generally (irrespective of the case-scenario and filter-type) lower mean-RMSE values are attained for a discount factor $\delta = 0.83$.

Under the referral procedure we were asked to assess the impact of much lower, say 0.5, or larger, say 0.999, discount factor values. To respond to such request we focused on the SARV(1) were some discrepancies on the estimation was already detected as a function of the chosen discount factor. Specifically, we revisit the estimation of states and fixed parameters of the nonlinear SARV(1) model using discount factor values of $\delta \in \{0.5, 0.75, 0.83, 0.9, 0.95, 0.99\}$ and represent the observed results in Figures C.1 and C.2; notice that we consider a larger set of requested δ values. The findings obtained through this complementary MC study are thoroughly explained in Appendix C and allow us to state that in general $\delta = 0.83$ yields very good statistical performance; see details in Section C.2.

After the above discussion of the obtained results, we provide a list with the most important contributions of this thesis.

7.2 Contributions

The main contributions (mostly in order of importance) of this thesis are:

- The most relevant contribution of this work is the proposed hybrid approach named by us SIRJ, which combines the self organizing (SO) filter of Kitagawa (1998) with the Kernel smoothing and shrinkage ideas presented in Liu and West (2001). We consider that this PF variant is a competing alternative to the well-established and widely used particle filter variant of Liu and West (2001), named LW PF in this work.

- Implementation of all filters under study (pseudo-codes with a unified notation are provided) using the R language (R Development Core Team 2013) and including the implementation of the residual and stratified resampling schemes. The specific code for the different filters are going to be available upon request to the PhD author.

- An overview of the most classic existing sequential Monte Carlo methods, named particle filters, for estimating solely the states (in Chapter 2) and for the simultaneous estimation of states and fixed parameters (Chapter 5). Additionally, in Chapter 2, we provide a complete coverage of the non-simulation based Kalman filter, the extended and the unscented Kalman filter. All these methods used for state estimation are scattered in the literature and we have put our effort into unifying notation for the sake of consistency, readability and comparability. This work provides, in Chapter 2, corresponding pseudo-codes for all filters studied for state estimation only: the SIR, ASIR, EPF and UPF. Likewise, in Chapter 5 we provide pseudo-codes for all three proposed filters to tackle the simultaneous estimation of states and fixed parameters: the SIRJ, EPFJ (being the KPFJ a special case) and the UPFJ. The corresponding pseudo-code for the existing LW PF variant, taken as a benchmark, is also given.

- Exhaustive MC studies to test the filtering performance of the competing PF variants in a linear and nonlinear context are designed and carried out. Other key aspects, within the particle filtering methodology, like the choice of a resampling scheme and the degeneracy problem are also addressed.

- For state estimation in a linear and Gaussian context, we explore the use of the so-called KPF and SIRopt PF variants as special cases of existing PF variants; the former is a special case of the EPF and the latter of the SISR approach. The distinguished feature of the KPF is that it uses as a proposal PDF the normal distribution obtained via the KF. Likewise, the SIRopt uses a fully adapted proposal; the reader may refer to Chapters 2 and 3. In this part, dealing with a linear and Gaussian context, we compare in total five filters: four are simulation-based (SIR, ASIR, SIRopt and KPF) and one is non-simulation based (the optimal KF). The originality herein is on designing exhaustive MC experiments to place under the same "umbrella" filters scattered in the literature and using many times different settings that avoid a direct comparison. As a byproduct, we

were able to assess the impact on estimation of key factors such as the signal-to-noise-ratio, the resampling scheme, the number of particles, and the time series length.

- For state estimation in a nonlinear context (Chapter 4), we assess the competing performance of the SIR, ASIR, EPF, UPF in contraposition with the EKF and UKF. This MC exercise allowed us to confirm that when dealing with a SSM with a complex structure (being not only nonlinear but non-Gaussian and non-stationary), the particle filtering methodology is a better estimation alternative to the non-simulation based EKF and UKF approaches. Additionally, we confirm that indeed the UPF is able to outperform other PF variants using a small number of particles. Our contribution herein is on extending the MC experiment carried out by the authors of the UPF, so to also include in the comparison the ASIR filter, to consider the stratified resampling scheme apart from the residual, and by adding the elapsed CPU time as a measure of computational efficiency apart from the statistical criterion based on the RMSE. We remark that the nonlinear model was artificially created by the authors of the UPF filter to assess its statistical performance in contraposition with other existing algorithms, but had not further practical interest. That is the reason why we do not consider this model afterwards and do not place further effort on studying, for instance, how it might behave if more observations are used. We do explore however the impact that an increase of the number of particles can have on the statistical and computational performance of the competing filters. We found, for instance, that the SIR and ASIR show similar statistical performance for a small number of particles ($N_p = 200$), but that both react –as expected, the RMSE become smaller, but at different rates– to increasing the number of particles, being both able to equate the UPF performance at the expense of using more particles. We found also that although similar statistical performance was generally attained when using either residual or stratified resampling, the former yielded higher computational cost.

- Extension (and later implementation) of existing particle filter variants previously used for state estimation only, to be able to handle the simultaneous estimation of states and fixed model parameters. As possible extensions we propose the aforementioned SIRJ PF variant, but also the so-called EPFJ (being the KPFJ a special case) or UPFJ filters; the latter remaining as a theoretical proposal. The well-established and widely used Liu West particle filter is used as a benchmark. We address key aspects like the time series length, the increase of the number of particles, the sample impoverishment problem, and the impact of choice of a jittering parameter on the quality of the estimations. We found that the proposed SIRJ PF performs on par with the benchmark LW filter. Another relevant result in this part is the rather stable behavior observed in the degree of degeneracy of studied filters regardless of the number of particles used in the estimation procedure. This allowed us to provide a rule of thumb for the number of particles to be used: in a linear and Gaussian context (LLM and AR(1) plus noise), 5000 particles seemed to be a reasonable value irrespective of the 13 case scenarios in consideration and type of filters compared. For the nonlinear case (SARV(1) model), at least 5000 should be used. Specifically, if only estimating the states, we recommend to use 5000 particles and when estimating together states and param-

eters at least 5000, but preferably 10000 to be in the safe side; this irrespective of the type of filter and four case-scenarios in consideration.

- Application of the two PF methods already compared to two real data sets containing high volatility data: The IBEX 35 returns index and the Brent spot price series (Chapter 6). This allowed us to assess how studied filters, found to have equivalent performance in simulations, behave in real world environments. Therein, evaluation of the goodness of fit of the nonlinear SARV(1) model used for the two financial time series is included. We consider that the application highlights the potentiality of PFs to be used in realistic situations.

- As aforementioned, the study of the impact on estimation of the discount factor suggests that the value of $\delta = 0.83$ leads to reasonably good results when compared with other values listed before. Indeed, for the SARV(1) model, as a rule of thumb, we recommend to use δ values in the range $\delta \in [0.75, 0.90]$, for which lower values of the mean-RMSE are obtained, irrespective of the case-scenarios. For the local level model, the simulation results suggest that there is practically no difference in the quality of the estimations when using discount factors $\delta = 0.83$, $\delta = 0.95$, or even $\delta = 0.99$. Further research is needed, however, to confirm the role that the discount factor has on estimation.

Since we consider the SIRJ PF to be our most important contribution, we want to state the following:

The SIRJ PF shows (via MC studies) to have a good performance in comparison to the well-known LW PF variant, both computationally and statistically for the models considered in this work. The LW PF was found to suffer in lesser degree the degeneracy drawback than the SIRJ, though we consider that the discrepancies found are not relevant. With respect to the computational performance, the LW was found to take approximately twice the time of the SIRJ PF. As concluded before, we have shown that our proposed SIRJ PF variant is able to equate the performance of the benchmark LW PF with a lower computational cost, and, if needed, more particles could be used for the SIRJ to improve efficiency.

Additionally, in general, due to the ease of computation and portability, the particle filters being sequential methods do not impose the storage of all data, feature that makes them more attractive/desirable. Naturally, particle filters –like any MC sampling method– are subject to both statistical and computational errors. We do not want to suggest that PFs do not have limitations. Thus, we want to pinpoint the limitations found in the present work and suggest possible further lines of research.

7.3 Limitations and Future Lines of Research

Possibly the three main limitations of the present work are:

1. Particle filters, like other approaches, suffer the so-called “curse of dimensionality”. An already detected partial solution consists of adopting a Rao-Blackwellized approach when feasible; that

is, the state-space vector can be decomposed in a linear and nonlinear part and then, for instance, one can use the KF for the linear part and particle filtering for the nonlinear. This helps to reduce the dimension of the state-space vector that is estimated via the more computationally costly particle filtering methodology. It seems that the new approach of Andrieu et al. (2010) can help in this respect also.

2. We found that particle filters require having prior knowledge about the possible region containing the true parameters in order to specify a proper prior distribution for the fixed parameters. A particle filter is not able to adequately converge to the true value if it is not contained in the support of the prior PDF. In practice, this rarely poses a huge problem, since usually the practitioner has some expert knowledge about the data and/or model at hand.

3. Throughout this work, extensive and systematic Monte Carlo studies have been carried out, but sometimes even more simulation scenarios could have been used. For instance, in part I of the thesis, sometimes we consider not so large values of the time series length T. We want to remark, however, that generally a great variety of settings is chosen for the performed simulation studies, and that this is done with the aim to be exhaustive enough having the specific simulation goal in mind.

As future lines of research, we envision the following:

1. The assessment of the behavior of the UPFJ in practice. In other words, look for a model that is complex enough that not only makes feasible the implementation of most studied particle filters, but also that has an impact from the application point of view. Try to improve the efficiency of the implementation of the UPFJ, which is actually too costly, is also in mind. We consider the EPFJ and UPFJ worth further study.

2. The extension of the current MC studies comparing the SIRJ and the LW PF variants, by also incorporating the so-called Particle Learning particle filter variant, which is based on sufficient statistics; see, for instance, Lopes and Tsay (2011), Lopes et al. (2011), and Prado and Lopes (2013).

3. The performance of an even more exhaustive study of the potential impact of the discount factor considering, for instance, other types of models. The results obtained, for the non-stationary local level model and the nonlinear SARV(1) models, suggest that better statistical performance is generally attained at δ values outside the range recommended and/or typically used in the literature. However, further study is needed in this direction.

4. The application suggests the presence of fat tails on the residuals of the IBEX 35 data set. In that case, the use of a more heavy tailed distribution for the measurement noise is desirable as done, for instance, in Muñoz, Márquez, and Acosta (2007), where a Student-t distribution was entertained.

5. The development of an R package including all filters studied is also a target, once some filters efficiency is further tackled (on-going work).

To close this work, we want to express the following.

The adopted particle filtering methodology though proves to be very efficient, it suffers an inherent drawback: the so-called degeneracy problem. Luckily, a lot of effort (also ours) has been placed to surmount (or at least postpone) this drawback that could be problem specific. Thus, the use of any variant of the particle filters implies the need to invest certain amount of time to carry out "tunning" tasks. For instance, it is relevant to somehow monitor the degree of degeneracy present within each particle filter variant, to choose the best importance PDF as possible, or to choose the best resampling strategy. Our experience with the application of this approach indicates that to achieve good performance, the determination of the number of enough particles is important, but also problem and type of filter specific. Although sometimes we provide a rule of thumb for the number of particles to be used in the estimation procedure, this determination requires the performance of exhaustive Monte Carlo studies. Naturally, this "tunning" tasks only become necessary when we are dealing with problems that have been not previously exhaustively studied.

From a pedagogical perspective, our intention is that this work serves as a point of reference for practitioner researchers interested in getting to know the particle filtering methodology, since all filters are first described in detail, then corresponding pseudo-codes are written and finally they all are implemented in the R language. Moreover, key classic and more recent references within the particle filtering methodology are pointed out. Detailed explanations of important issues within this methodology are addressed, like the choice of the proposal distribution, the resampling scheme and its different implementation procedures, and how to undertake the potential degeneracy drawback of PFs.

REFERENCES

Abanto-Valle, C., H. Migon, and H. Lopes (2010). Bayesian modeling of financial returns: A relationship between volatility and trading volume. *Applied Stochastic Models in Business and Industry 26*(2), 172–193.

Acosta, L. M., M. Martí-Recober, and M. P. Muñoz (2003). Autoregressive parameter's estimation via particle filtering. In *Actas del 27 Congreso de Estadística e Investigación Operativa*, pp. 1947–1953.

Acosta, L. M., M. Martí-Recober, and M. P. Muñoz (2004). Our walk through particle filtering: Simultaneous estimation of state and parameters. In 6^{th} *World Congress of the Bernoulli Society*.

Acosta, L. M. and M. P. Muñoz (2007). Benchmark study of some particle filter variants in a nonlinear non-gaussian framework: Application to a stochastic volatility model parameters estimation. In *BISP5: Fith Workshop on Bayesian Inference in Stochastic Processes*.

Akaike, H. (1974). Markovian representation of stochastic processes and its application to the analysis of autoregressive moving average processes. *Journal of time Series Analysis, Annals of the Institute of Statistical Mathematics 26*, 363–387.

Andersen, T., T. Bollerslev, F. Diebold, and H. Ebens (2001). The distribution of realised stock return volatility. *Journal of Financial Economics 61*(1), 43–76.

Anderson, B. and J. Moore (1979). *Optimal Filtering*. Englewood Cliffs, NJ: Prentice Hall.

Anderson, H., K. Nam, and F. Vahid (1999). Asymmetric nonlinear smooth transition GARCH models. In P. Rothman (Ed.), *Non Linear Time Series Analysis of Economic and Financial Data*, pp. 191–207. Amsterdam: Kluwer.

Andrieu, C., A. Doucet, and R. Holenstein (2010). Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society. Series B 72*(3), 269–342.

Arulampalam, S., S. Maskell, N. Gordon, and T. Clapp (2002). A tutorial on particle filters for online non-linear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing 50*(2), 174–188.

Bera, A. K. and M. L. Higgins (1995). On ARCH models: Properties, estimation and testing. In L. Oxley, D. George, C. Roberts, and S. Sayer (Eds.), *Surveys in Econometrics*, pp. 215–272. Oxford: Basil Blackwell.

Bergman, N. (1999). *Recursive Bayesian Estimation: Navigation and Tracking Applications*. Ph. D. thesis, Department of Electrical Engineering, Linköping, Sweden. Dissertation No 579.

Black, F. (1976). Studies in stock price volatility changes. In *American Statistical Association, Proceedings of the Business and Economic Statistics Section*, pp. 177–181.

Bolic, M. (2004). *Architectures for Efficient Implementations of Particle Filters*. Ph. D. thesis, Stony Brook University.

Bollerslev, T. (1986). Generalized autoregressive conditional heterocedasticity. *Journal of Econometrics 31*(3), 307–327.

Bollerslev, T., R. Chou, and K. Kroner (1992). Arch modeling in finance: A review of the theory and empirical evidence. *Journal of Econometrics 52*(1-2), 5–59.

Bollerslev, T., R. Engle, and D. Nelson (1994). ARCH models. In R. Engle and D. McFadden (Eds.), *Handbook of Econometrics IV*, pp. 2961–3038. Amsterdam: Elsevier Science.

Box, G. and G. Jenkins (1976). *Time Series Analysis, Forecasting and Control*. San Francisco: Holden-Day: Academic Press.

Box, G., G. Jenkins, and G. Reinsel (1994). *Time Series Analysis: Forecasting and Control*. Englewood Cliffs, NJ: Prentice-Hall.

Brockwell, P. and R. Davis (1996). *Time Series: Theory and Methods*. Berlin: Springer-Verlag.

Carlin, B., N. Polson, and D. Stoffer (1992). A monte carlo approach to nonnormal and nonlinear state-space modeling. *Journal of the American Statistical Association 87*(418), 493–500.

Carnero, M., D. Peña, and E. Ruiz (2001). Is stochastic volatility more flexible than GARCH? Technical Report W.P. 01–08, Universidad Carlos III de Madrid.

Carnero, M., D. Peña, and E. Ruiz (2004). Persistence and kurtosis in GARCH and stochastic volatility models. *Journal of Econometrics 2*(2), 319–342.

Carpenter, J., P. Clifford, and P. Fearnhead (1999). An improved particle filter for non-linear problems. *IEEE Proceedings on Radar and Sonar Navigation 146*(1), 2–7.

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

Carvalho, C. M. and H. F. Lopes (2007). Simulation-based sequential analysis of markov switching stochastic volatility models. *Computational Statistics & Data Analysis 51*(9), 4526–4542.

Chib, S., F. Nardari, and N. Shephard (2002). Markov chain Monte Carlo methods for stochastic volatility models. *Journal of Econometrics 108*, 231–316.

Congdon, P. (2007). *Applied Bayesian Modelling*. Wiley Series in Probability and Statistics.

de Freitas, J., M. Niranjan, A. Gee, and A. Doucet (2000). Sequential Monte Carlo methods to train neural network models. *Neural Computation 12*(4), 955–993.

Dickey, D. and W. Fuller (1979). Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association 74*(366), 427–431.

Diebold, F. and J. Lopez (1995). Modelling volatility dynamics. In K. Hoover (Ed.), *Macroeconomics: Developments, Tensions and Prospects*, pp. 427–472. Boston: Kluwer.

Ding, Z., C. Granger, and R. Engle (1993). A long memory property of stock market returns and a new model. *Journal of Empirical Finance 1*(1), 83–106.

Doucet, A. (1998). On sequential simulation-based methods for Bayesian filtering. Technical Report CUED/F-INFENG/TR.310, Department of Engineering, University of Cambridge.

Doucet, A., N. de Freitas, and N. Gordon (2001). *Sequential Monte Carlo Methods in Practice*. Springer-Verlag.

Doucet, A., S. Godsill, and C. Andrieu (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing 10*(3), 197–208.

Doucet, A. and A. M. Johansen (2011). A tutorial on particle filtering and smoothing: fifteen years later.

Doucet, A. and V. Tadic (2003). Parameter estimation in general state-space models using particle methods. *Annals of the Institute of Statistical Mathematics 55*(2), 409–422.

Durbin, J. and S. Koopman (2001). *Time Series Analisis by State Space Methods*. Oxford: Oxford University Press.

Engle, R. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of the United Kingdom inflation. *Econometrica 50*(4), 987–1007.

Engle, R. and A. Patton (2001). What good is a volatility model? Technical Report C22, NYU Stern School of Business and University of California, California, San Diego.

Fama, E. (1965). The behavior of stock market prices. *Journal of Business 38*(1), 34–105.

Fearnhead, P. (1998). *Sequential Monte Carlo Methods in Filter Theory*. Ph. D. thesis, Merton College, University of Oxford.

Flury, T. and N. Shephard (2008). Bayesian inference based only on simulated likelihood: particle filter analysis of dynamic economic models. Economics Series Working Papers 413, University of Oxford, Department of Economics.

Franses, P. and D. Van Dijk (2000). *Non-linear time series models in empirical finance*. Cambridge, United Kingdom: Cambridge University Press.

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

Geweke, J. (1989). Bayesian inference in econometrics models using Monte Carlo integration. *Econometrica 57*, 1317–1339.

Ghysels, E., A. Harvey, and E. Renault (1996). Stochastic volatility. Technical Report 95s–49, CIRANO: Centre Interuniversitaire de Recherche en Analyse des Organisations, Montréal.

Gómez, V. and A. G. Maravall (1994). Estimation, prediction and interpolation for nonstationary series with the Kalman filter. *Journal of the American Statistical Association 89*, 611–624.

Gordon, N. J., D. Salmond, and A. Smith (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE. Proceedings-F 140*(2), 107–110.

Hamilton, J. and R. Susmel (1994). Autoregressive conditional heterokedasticity and changes in regime. *Journal of Econometrics 64*(1-2), 307–333.

Harvey, A. (1996). *Forecasting, Structural Time Series and the Kalman Filter*. Cambridge, United Kingdom: Cambridge University Press.

Higuchi, T. (1997). Monte Carlo filter using the genetic algorithm operators. *Journal of Statistical Computation and Simulation 59*(1), 1–23.

Hürzeler, M. and H. Künsch (1998). Monte Carlo approximations for general state-space models. *Journal of Computational and Graphical Statistics 7*, 175–193.

Jacquier, E., N. Polson, and P. Rossi (1994). Bayesian analysis of stochastic volatility models. *Journal of Business & Economic Statistics 12*(4), 69–87.

Jazwinski, A. (1970). *Stochastic Processes and Filtering Theory*, Volume 64 of *Mathematics in Science and Engineering*. New York: Academic Press.

Julier, S. (2002, May). The scaled unscented transformation. In *Proceedings of the IEEE American Control Conference*, Anchorage, AK, USA, pp. 4555–4559.

Julier, S. and J. Uhlmann (1997). A new extension of the Kalman filter to nonlinear systems. In *Proc. of AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls*, Orlando, FL, USA.

Julier, S. and J. Uhlmann (2004). Invited paper: Unscented filtering and nonlinear estimation. *Proceedings of the IEEE 92*(3), 401–422.

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

Kalman, R. and R. Bucy (1961). New results in linear filtering and prediction theory. *Journal of Basic Engineering 83*, 95–108.

Kanazawa, K., D. Koller, and S. Russel (1995). Stochastic simulation algorithms for dynamic probabilistic networks. In *Proceedings of the Eleventh Annual Conference on Uncertainty in AI*, UAI, pp. 346–351.

Kim, S., N. Shephard, and S. Chib (1998). Stochastic volatility: likelihood inference and comparison with ARCH models. *Review of Economic Studies 65*, 361–393.

Kitagawa, G. (1987). Non-Gaussian state-space modeling of nonstationary time series. *Journal of the American Statistical Association 82*, 1032–1063.

Kitagawa, G. (1996). Monte Carlo filter and smoother for non-Gaussian nonlinear state-space model. *Journal of Computational and Graphical Statistics 5*, 1–25.

Kitagawa, G. (1998). Self organizing state-space model. *Journal of the American Statistical Association 93*, 1203–1215.

Kitagawa, G. and S. Sato (2001). Monte Carlo smoothing and self organizing state-space model. In A. Doucet, d. N., and D. Gordon (Eds.), *Sequential Monte Carlo Methods in Practice*, pp. 177–195. Springer Verlag.

Kitagawa, G., T. Takanami, A. Kuwano, Y. Murai, and H. Shimamura (2002). Extraction of signal from high dimensional time series: Analysis of ocean bottom seismograph data. In S. Arikawa and A. Shinohara (Eds.), *Progress in Discovery Science*, LNAI 2281, Berlin, Heidelberg, pp. 449–458. Springer–Verlag.

Li, W. and K. Lam (1995). Modelling the asymmetry in stock returns by a threshold ARCH model. *The Statistician 44*(3), 333–341.

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

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

Ljung, G. and G. Box (1978). On a measure of a lack of fit in time series models. *Biometrika 65*(2), 297–303.

Lopes, H., C. Carvalho, M. Johannes, and N. Polson (2011). Particle learning for sequential bayesian computation. *Bayesian Statistics 9*, 317–360.

Lopes, H. and R. Tsay (2011). Particle filters and bayesian inference in financial econometrics. *Journal of Forecasting 30*, 168–209.

Loudon, G., W. Watt, and P. Yadav (2000). An empirical analysis of alternative parametric ARCH models. *Journal of Applied Econometrics 15*, 117–136.

MacCormick, J. and A. Blake (1999). A probabilistic exclusion principle for tracking multiple objects. In *Proceedings of the International Conference on Computer Vision*, pp. 572–578.

Mandelbrot, B. (1963). The variation of certain speculative prices. *Journal of Business 36*(4), 394–419.

Van der Merwe, R., A. Doucet, N. de Freitas, and E. Wan (2001, December). The unscented particle filter. In T. K. Leen, T. G. Dietterich, and V. Tresp (Eds.), *Advances in Neural Information Processing Systems (NIPS–13)*. MIT Press.

Meinhold, R. and N. Singpurwalla (1989). Robustification of Kalman filter models. *Journal of the American Statistical Association 84*, 479–486.

Mora-Galan, A., A. Perez, and E. Ruiz (2004, November). Stochastic volatility models and the Taylor effect. Technical Report Statistics and Econometrics Series 15, W.P. 04-63, Universidad Carlos III de Madrid, Calle Madrid, 28903 Getafe (Spain).

Márquez, M. D. (2002). *Modelo SETAR aplicado a la volatilidad de la rentabilidad de las acciones: algoritmos para su identificación*. Ph. D. thesis, Universitat Politècnica de Catalunya, Barcelona España.

Márquez, M. D., M. P. Muñoz, C. Villazón, M. Martí-Recober, and L. M. Acosta (2005). Rendimiento y volatilidad del IBEX 35: Capturando las asimetrías y el exceso de curtosis. Technical Report DR 2005/1, Universitat Politècnica de Catalunya.

Muñoz, M. P. (1988). Estimació dels parametres de models ARMA(p,q) mitjançant algorismes de filtrage optim, tesis doctoral. Technical report, Universitat Politècnica de Catalunya, Barcelona, España.

Muñoz, M. P., R. Egozcue, and M. Martí-Recober (1988). Estimació del pol y de la variancia del soroll d'un model AR(1) mitjançant filtratge no-lineal. *Questió 12*, 21–42.

Muñoz, M. P., M. D. Márquez, and L. M. Acosta (2007). Forecasting volatility by means of threshold models. *Journal of Forecasting 26*, 343–363.

Muñoz, M. P., M. D. Márquez, M. Martí-Recober, C. Villazón, and L. M. Acosta (2004). Stochastic volatility and TAR-GARCH models: Evaluation based on simulations and financial time series. In *Physica-Verlag/Springer ISBN: 3-7908-1554-3 COMPSTAT 2004*.

Muñoz, M. P., J. Pagès, and M. Martí-Recober (1988). Estimation of arma process parameters and noise variance by means of a non linear filtering algorithm. In P.-V. Heidelberg (Ed.), *COMPSTAT 1988 - Proceedings in Computational Statistics*, Volume 1, República Federal de Alemania (la), pp. 357–362.

Niemi, J. B. (2009). *Bayesian Analysis and Computational Methods for Dynamic Modeling*. Ph. D. thesis, Duke University, Department of Statistical Science.

Pellegrini, S. (2009). *Predicción en modelos de componentes inobservables condicionalmente heteroscedásticos*. Ph. D. thesis, Universidad Carlos III de Madrid, Departamento de Estadística.

Perron, P. (1988). Trends and random walks in macroeconomic time series. *Journal of Economic Dynamics and Control 12*, 297–332.

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

Pollock, D. (2003). Recursive estimation in econometrics. *Journal of Computational Statistics and Data Analysis 44*, 37–75.

Prado, R. and H. Lopes (2013). Sequential parameter learning and filtering in structured autoregressive state-space models. *Statistics and Computing 23*, 43–57.

R Development Core Team (2013). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. ISBN 3-900051-07-0.

Ripley, B. (1987). *Stochastic Simulation*. New York: Wiley.

Rodriguez, A. F. (2010). *Bootstrapping Unobserved Component Models*. Ph. D. thesis, Universidad Carlos III de Madrid, Departamento de Estadística.

Ruiz, E. (1994). Quasi-maximum likelihood estimation of stochastic volatility models. *Journal of Econometrics 63*(1), 289–306.

Shephard, N. and A. Harvey (1990). On the probability of estimating a deterministic component in the local level model. *London School of Economics, Journal of Time Series Análisis 11*(4), 339–347.

Shephard, N. and M. Pitt (1995, December). Likelihood analysis of non-gaussian parameter-driven models. Technical report, Department of Statistics, University of Oxford, OX1 3TG and Nuffield College, OX1 1NF, UK.

Shumway, R. (1988). *Applied Statistical Time Series Analysis*. New Jersey: Prentice Hall.

Shumway, R. and D. Stoffer (2000). *Applied Statistical Time Series Analysis*. Springer Verlag.

Shumway, R. and D. Stoffer (2006). *Time Series Analysis ans Its Applications. With R examples* (Second Edition ed.). Springer Verlag.

Smith, A. and A. Gelfand (1992). Bayesian statistics without tears: a sampling resampling perspective. *American Statistician 46*(4), 84–88.

Stock, J. and M. Watson (2007). Why has u.s. inflation become harder to forescast? *Journal of Money, Credit and Banking 39*, 13–33.

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.

Stroud, J., N. G. Polson, and P. Müller (2004). *State Space and Unobserved Components Models*, Chapter Practical Filtering for Stochastic Volatility Models, pp. 236–247. Cambridge University Press.

Tanizaki, H. (1991). *Nonlinear Filters: Estimation and Applications*. Ph. D. thesis, University of Pennsilvania.

Tanizaki, H. (1996). *Nonlinear Filters, Estimation and Applications*. New York: Springer.

Tanizaki, H. (2001). Estimation of unknown parameters in nonlinear and non-Gaussian state-space models. *Journal of Statistical Planning and Inference 96*, 301–323.

Tanizaki, H. and R. Mariano (1996). Nonlinear filters based on Taylor series expansions. *Communications in Statistics, Theory and Methods 25*, 1261–1282.

Tanizaki, H. and R. Mariano (1998). Nonlinear and nonnormal state-space modeling with Monte-Carlo stochastic simulations. *Journal of Econometrics 83*, 263–290.

Taylor, S. (1986). *Modeling Financial Time Series*. New York: John Wiley.

Taylor, S. J. (1994). Modelling stochastic volatility: A review and comparative study. *Mathematical Finance 4*, 183–204.

Trapletti, A. and K. Hornik (2012). *tseries: Time Series Analysis and Computational Finance*. R package version 0.10-30.

Tsay, R. (2002). *Analysis of Financial Time Series*. Wiley Series in Probability and Statistics. Hoboken, NJ, USA: Wiley-Interscience.

Van der Merwe, R. (2004). *Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models*. Ph. D. thesis, Oregon Health and Science University.

Van der Merwe, R., A. Doucet, N. de Freitas, and E. Wan (2000, August). The unscented particle filter. Technical Report CUED/F-INFENG/TR.380, Department of Engineering, University of Cambridge.

Wan, E. and R. Van der Merwe (2000). The Unscented Kalman Filter for nonlinear estimation. In *Proceedings of Symposium 2000 on Adaptive Systems for Signal Processing, Communication and Control*, Alberta, Canada.

Wan, E. and R. Van der Merwe (2001). The unscented Kalman filter. In S. Haykin (Ed.), *Kalman Filtering and Neural Networks*. Wiley Publishing.

Wei, W. (1994). *Time series Analysis:Univariate and Multivariate Methods*. Redwood City, Ca.: Addison Wesley.

West, M. (1993). Approximating posterior distributions by mixtures. *Journal of Royal Statistical Society 55*, 409–422.

West, M. and J. Harrison (1989). *Bayesian Forecasting and Dynamic Models*. New York Inc.: Springer-Verlag.

West, M. and J. Harrison (1997). *Bayesian Forecasting and Dynamic Models*. New York Inc.: Springer-Verlag.

Wishner, R., J. Tabaczynski, and M. Athans (1969). A comparison of three non-linear filters. *Automatica 5*, 487–496.

Wood, S. (2004). Stable and efficient multiple smoothing parameter estimation for generalized additive models. *Journal of the American Statistical Association 99*, 673–686.

Zakoian, J. (1994). Threshold heteroskedastic models. *Journal of Economic Dynamics and Control 18*(5), 931–955.

Zoeter, O., A. Ypma, and T. Heskes (2004). Improved unscented Kalman smoothing for stock volatility estimation. In *Machine Learning for Signal Processing, 2004. Proceedings of the 2004 14th IEEE Signal Processing Society Workshop*, pp. 143–152.

APPENDIX A

COMPLEMENTARY SIMULATION STUDY ISSUES

In this thesis, two sketches are provided for a better illustration of the simulation design and for a better understanding of the defined performance criteria. The first sketch illustrates the criteria for comparing the non-simulation based filters; see Figure 3.2 of Chapter 3. Similarly, Section A.1 includes a sketch of the comparison criteria used for the simulation based filters; see Figure A.1.

Section A.2 displays the main programm code in R language for estimating the states of the synthetic nonlinear model considered in Chapter 5. In all MC studies carried out throughout this research, the same general skeleton is used for the main program. Its choice is based on and inspired by the authors of the unscented particle filter (Van derMerwe et al. 2001) and on the technical report of van der Merwe et al. (2000) that develop a program-Demo, which is the base for the results presented in van der Merwe et al. (2001). We remark, however, that the full implementation of every specific filter under study is performed by the author of this PhD thesis. In this part, additionally, the original MATLAB instructions for implementing the residual resampling algorithm are reported, which were taken from the website `http://vismod.media.mit.edu/pub/yuanqi/mcep/residualR.m` [last visited: September 2013]. These instructions are later on written in R language by the author of this thesis.

A.1 Sketch of Performance Criteria for Particle Filter Variants

			Time index (t)				
Set	**Filter(f)**		**1**	.....	**T**		**Comparison criteria**
		NP(j)					
		1	$\hat{x}_1^{f(j)}$	.....	$\hat{x}_T^{f(j)}$		
		2	$\hat{x}_1^2$	.....	$\hat{x}_T^2$		
1	**PF's**	$\vdots$	$\vdots$	$\vdots$	$\vdots$		
		$\vdots$	$\vdots$	$\vdots$	$\vdots$		
		M	$\hat{x}_1^M$	.....	$\hat{x}_T^M$		
			$\Downarrow$		$\Downarrow$		
			$\sum_{j=1}^{M} \hat{x}_{1,[1]}^{f(j)} / M$	.....	$\sum_{j=1}^{M} \hat{x}_{T,[1]}^{f(j)} / M$		
			$\Downarrow$		$\Downarrow$		
		$\hat{\boldsymbol{x}}_{t,[1]}^f =$	$\hat{\boldsymbol{x}}_{1,[1]}^f$	.....	$\hat{\boldsymbol{x}}_{T,[1]}^f$	$\rightarrow$	$\text{RMSE}_{[1]}^f$ $\quad$ $\text{CPU}_{[1]}^f$
$\vdots$				$\vdots$			$\vdots$
$\vdots$				$\vdots$			$\vdots$
		NP(j)					
		1	$\hat{x}_1^{f(j)}$	.....	$\hat{x}_T^{f(j)}$		
		2	$\hat{x}_1^2$	.....	$\hat{x}_T^2$		
S	**PF's**	$\vdots$	$\vdots$	$\vdots$	$\vdots$		
		$\vdots$	$\vdots$	$\vdots$	$\vdots$		
		M	$\hat{x}_1^M$	.....	$\hat{x}_T^M$		
			$\Downarrow$		$\Downarrow$		
			$\sum_{j=1}^{M} \hat{x}_{1,[S]}^{f(j)} / M$	.....	$\sum_{j=1}^{M} \hat{x}_{T,[S]}^{f(j)} / M$		
			$\Downarrow$		$\Downarrow$		
		$\hat{\boldsymbol{x}}_{t,[S]}^f =$	$\hat{\boldsymbol{x}}_{1,[S]}^f$	.....	$\hat{\boldsymbol{x}}_{T,[S]}^f$	$\rightarrow$	$\text{RMSE}_{[S]}^f$ $\quad$ $\text{CPU}_{[S]}^f$
							$\Downarrow$ $\quad$ $\Downarrow$
							Mean(RMSE)_f $\quad$ Mean(CPU)_f
							Var(RMSE)_f

Table A.1: Sketch II: Comparison criteria of simulation based filters

A.2 Main programm code for estimating the states of a synthetic nonlinear model: Benchmark Implementation

```
# PURPOSE : Benchmark study to assess the performance (advantages and possible drawbacks) in state
# estimation of the following nonlinear filters:
#   I    EKF:   Extended Kalman Filter
#   II   UKF:   Unscented Kalman Filter
#   III  SIR:   Sampling Importance Resampling Particle Filter variant, using transition prior as proposal
#   IV   EPF:   Extended Particle Filter variant, using the EKF as proposal
#   V    UPF:   Unscented Particle Filter variant,using the UKF as proposal
# Extra for Simulation II
#   VI  ASIR:   Auxiliary Sampling Importance Resampling Particle Filter variant

# Used Model:   Synthetic nonlinear non-Gaussian dynamic model taken from literature (used by UPF authors).
# Some ideas taken from program-Demo which is the base of UPF authors.
# We remark that the whole implementation of the filters is the student's full responsibility and are going
# to conform an R-Package.

# PhD Student: Lesly Acosta      (lesly.acosta@upc.edu)
# DATE:        2004-today

# Get ready to start:
#####################################################################
# Charge needed libraries
# Charge all needed functions corresponding to the above filters, the function for generating the data plus
# the two functions for the deterministic and residual resampling.
set.seed(seed)             # set seed so results are reproducible

# Data generation Step
#####################################################################
Sets <- 100                # Number of realizations of the synthetic nonlinear model
T <- 60                    # Time series maximum length
obswnv <- 1e-5             # True Gaussian measurement noise variance
a<- 3                      # True Gamma state noise shape parameter
b<- 1/2                    # True Gamma state noise scale parameter; mean=3/2, variance=3/4
w <- 4e-2                  # Constant and fixed known parameter used in Transition equation
phi1 <- 0.5
phi2 <- 0.2                # Constant and fixed known parameters used in Measurement equation
phi3 <- 0.5
thr <- T/2

for (nset in 1:Sets){
   SIN <- genera.SIN(T)
   write(SIN$yt, paste(path, "DATASINxt/ytSIN.", nset, sep=""))
   write(SIN$xt, paste(path, "DATASINxt/xtSIN.", nset, sep=""))
}

# Filtering Step
#####################################################################
ini <- date()
Np <- 200                  # Number of particles
                           # Note: We check later how its increase affects performance
resStrat <- 2              # Residual resampling= 1; Deterministic resampling =2
P0 <- 3/4                  # Initial system state variance
```

```
# Values used by EKF/UKF
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
R <- obswnv                #R denotes measurement noise variance
Q <- 3/4                   #Q denotes state noise variance

# Values used by SIR PF
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Rpf <- obswnv
Qpf <- 2*3/4

# Values used by EPF
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Rpfekf <- 1e-1
Qpfekf <- 10*3/4

# Values used by UPF
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Rpfukf <- 1e-1
Qpfukf <- 2*3/4
alpha <- 1                  # Scale parameter
beta  <- 0                  # Non-negative parameter that incorporates prior knowledge of the distribution
kappa <- 2                  # Secondary scaling parameter, usually set to $0$.
                            # To guarantee a positive definite cov-matrix, a non-negative value must be chosen.

# Values used by ASIR PF
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Rpfasir <- obswnv
Qpfasir <- 2*3/4
K <- Np

# Memory allocation for process and output variables
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

# EKF
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
muXt_ekfSet <- matrix(NA, Sets, T)
VXt_ekfSet  <- matrix(NA, Sets, T)
rmsError_ekf <- rep(NA, Sets)
time_ekf <- rep(NA, Sets)

# UKF
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
muXt_ukfSet <- matrix(NA, Sets, T)
VXt_ukfSet  <- matrix(NA, Sets, T)
rmsError_ukf <- rep(NA, Sets)
time_ukf <- time_ekf

# SIR
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
muXt_pfSet  <- matrix(NA, Sets, T)
rmsError_pf <- rep(NA, Sets)
time_pf <- rep(NA, Sets)
```

```
# EPF
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
muXt_pfekfSet  <- matrix(NA, Sets, T)
rmsError_pfekf <- rep(NA, Sets)
time_pfekf <- rep(NA, Sets)

# UPF
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
muXt_pfukfSet  <- matrix(NA, Sets, T)
rmsError_pfukf <- rep(NA, Sets)
time_pfukf  <- rep(NA, Sets)

## Extra Simulation II
# ASIR
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
muXt_pfasirSet  <- matrix(NA, Sets, T)
rmsError_pfasir <- rep(NA, Sets)
time_pfasir <- rep(NA, Sets)

iniekf <- date()

# MAIN LOOP for EKF
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
for (nset in 1:Sets){
   SIN$yt <- scan(paste(path, "DATASINxt/ytSIN.", nset, sep=""))
   SIN$xt <- scan(paste(path, "DATASINxt/xtSIN.", nset, sep=""))
   #EKF calling
   out.EKF<-SIN.EKF(T, SIN$yt)
   ##############################
   ##-- CALCULATE PERFORMANCE --#
   ##############################
   muXt_ekfSet[nset,] <- out.EKF$muEKF
   rmsError_ekf[nset] <- sqrt(sum((SIN$xt-muXt_ekfSet[nset, ])^2)/T)
   time_ekf[nset] <- out.EKF$timeEKF
   VXt_ekfSet[nset, ] <- out.EKF$PEKF
}

finekf<-date()
# Calculate mean of RMSE errors for EKF
mean_RMSE_ekf <- round(mean(rmsError_ekf), 4)
# Calculate variance of RMSE errors
var_RMSE_ekf  <- round(var(rmsError_ekf), 4)
# Calculate mean of execution time
mean_time_EKF  <- round(mean(time_ekf), 4)

# MAIN LOOP for UKF
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
for (nset in 1:Sets){
   SIN$yt <- scan(paste(path, "DATASINxt/ytSIN.", nset, sep=""))
   SIN$xt <- scan(paste(path, "DATASINxt/xtSIN.", nset, sep=""))
   #UKF calling
   out.UKF <- SIN.UKF(T, SIN$yt)
```

```
   ##############################
   ##-- CALCULATE PERFORMANCE --#
   ##############################
   muXt_ukfSet[nset,] <- out.UKF$muUKF
   rmsError_ukf[nset] <- sqrt(sum((SIN$xt-muXt_ukfSet[nset, ])^2)/T)
   time_ukf[nset] <- out.UKF$timeUKF
   VXt_ukfSet[nset, ] <- out.UKF$PUKF
}

# Calculate mean of RMSE errors for UKF
mean_RMSE_ukf <- round(mean(rmsError_ukf), 4)
# Calculate variance of RMSE errors
var_RMSE_ukf  <- round(var(rmsError_ukf), 4)
# Calculate mean of execution time
mean_time_UKF <- round(mean(time_ukf), 4)

# MAIN LOOP for SIR PF
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
for (nset in 1:Sets){
   SIN$yt <- scan(paste(path, "DATASINxt/ytSIN.", nset, sep=""))
   SIN$xt <- scan(paste(path, "DATASINxt/xtSIN.", nset, sep=""))
   out.PF <- SIN.PF(SIN$yt, Np, Rpf)                              #PF function call
   muXt_pfSet[nset,] <- out.PF$muPF
   rmsError_pf[nset] <- sqrt(sum((SIN$xt-muXt_pfSet[nset, ])^2)/T)
   time_pf[nset] <- out.PF$timePF
}
##############################
##-- CALCULATE PERFORMANCE --#
##############################
# Calculate mean of RMSE errors for PF
mean_RMSE_pf <- round(mean(rmsError_pf), 4)
# Calculate variance of RMSE errors
var_RMSE_pf  <- round(var(rmsError_pf), 4)
# Calculate mean of execution time
mean_time_PF <- round(mean(time_pf), 4)

# MAIN LOOP for EPF
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
for (nset in 1:Sets){
   SIN$yt <- scan(paste(path, "DATASINxt/ytSIN.", nset, sep=""))
   SIN$xt <- scan(paste(path, "DATASINxt/xtSIN.", nset, sep=""))
   out.PFEKF <- SIN.PFmitEKFprop(SIN$yt, Np, obswnv)              #EPF function call
   muXt_pfekfSet[nset,] <- out.PFEKF$muPFEKF
   rmsError_pfekf[nset] <- sqrt(sum((SIN$xt-muXt_pfekfSet[nset, ])^2)/T)
   time_pfekf[nset] <- out.PFEKF$timePFEKF
}
##############################
##-- CALCULATE PERFORMANCE --#
##############################
# Calculate mean of RMSE errors for PFEKF
mean_RMSE_pfekf <- round(mean(rmsError_pfekf), 4)
# Calculate variance of RMSE errors
var_RMSE_pfekf  <- round(var(rmsError_pfekf), 4)
# Calculate mean of execution time
mean_time_PFEKF <- round(mean(time_pfekf), 4)
```

```
# MAIN LOOP for UPF
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
for (nset in 1:Sets){
   SIN$yt <- scan(paste(path,"DATASINxt/ytSIN.", nset, sep=""))
   SIN$xt <- scan(paste(path,"DATASINxt/xtSIN.", nset, sep=""))
   out.PFUKF <- SIN.PFmitUKFprop(SIN$yt, Np, obswnv)            #UPF function call
   muXt_pfukfSet[nset,] <- out.PFUKF$muPFUKF
   rmsError_pfukf[nset] <- sqrt(sum((SIN$xt-muXt_pfukfSet[nset, ])^2)/T)
   time_pfukf[nset] <- out.PFUKF$timePFUKF
}
##############################
##-- CALCULATE PERFORMANCE --#
##############################
# Calculate mean of RMSE errors for PFUKF
mean_RMSE_pfukf <- round(mean(rmsError_pfukf), 4)
# Calculate variance of RMSE errors
var_RMSE_pfukf  <- round(var(rmsError_pfukf), 4)
# Calculate mean of execution time
mean_time_PFUKF <- round(mean(time_pfukf), 4)

# MAIN LOOP for ASIR
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
for (nset in 1:Sets){
   SIN$yt <- scan(paste(path, "DATASINxt/ytSIN.", nset, sep=""))
   SIN$xt <- scan(paste(path, "DATASINxt/xtSIN.", nset, sep=""))
   out.PFASIR <- SIN.PFmitASIRprop(SIN$yt,Np,K,Rpfasir,nset)     #ASIR PF function call
   muXt_pfasirSet[nset,] <- out.PFASIR$muPFASIR
   rmsError_pfasir[nset] <- sqrt(sum((SIN$xt-muXt_pfasirSet[nset, ])^2)/T)
   time_pfasir[nset] <- out.PFASIR$timePFASIR
}

##############################
##-- CALCULATE PERFORMANCE --#
##############################
# Calculate mean of RMSE errors for PFASIR
mean_RMSE_pfasir <- round(mean(rmsError_pfasir), 4)
# Calculate variance of RMSE errors
var_RMSE_pfasir  <- round(var(rmsError_pfasir), 4)
# Calculate mean of execution time
mean_time_PFASIR <- round(mean(time_pfasir), 4)

#Print out simulation results
#################################################################
print(paste("(seed, REsScheme, NRuns) = ", seed, resStrat, Runs))
print(paste("Method    ", "mean_RMSE", "var_RMSE", "mean_time"))
print(paste("EKF       ", mean_RMSE_ekf,    var_RMSE_ekf,    mean_time_EKF))
print(paste("UKF       ", mean_RMSE_ukf,    var_RMSE_ukf,    mean_time_UKF))
print(paste("PF        ", mean_RMSE_pf,     var_RMSE_pf,     mean_time_PF))
print(paste("PF_EKF    ", mean_RMSE_pfekf,  var_RMSE_pfekf,  mean_time_PFEKF))
print(paste("PF_UKF    ", mean_RMSE_pfukf,  var_RMSE_pfukf,  mean_time_PFUKF))
print(paste("PF_ASIR   ", mean_RMSE_pfasir, var_RMSE_pfasir, mean_time_PFASIR))
fin<-date()# End of Main R language programm code
```

The original MATLAB instructions for implementing the residual resampling algorithm can be found in `http://vismod.media.mit.edu/pub/yuanqi/mcep/residualR.m` [last visited: September 2013]. These instructions were written in R language by the author of this thesis.

```
function outIndex = residualR(inIndex,q);
% PURPOSE : Performs the resampling stage of the SIR
%           in order(number of samples) steps. It uses Liu's
%           residual resampling algorithm and Niclas' magic line.
% INPUTS  : - inIndex = Input particle indices.
%           - q = Normalised importance ratios.
% OUTPUTS : - outIndex = Resampled indices.
% AUTHORS  : Arnaud Doucet and Nando de Freitas - Thanks for the acknowledgement.
% DATE     : 08-09-98

if nargin < 2, error('Not enough input arguments.'); end

[S,arb] = size(q);  % S = Number of particles.

% RESIDUAL RESAMPLING:
% ====================
N_babies= zeros(1,S);
% first integer part
q_res = S.*q'; %' %multiply all normalized weights q' by No. of particles S
N_babies = fix(q_res); % integer part of q_res

% residual number of particles to sample
N_res=S-sum(N_babies);
if (N_res~=0)
  q_res=(q_res-N_babies)/N_res;
  cumDist= cumsum(q_res);
  % generate N_res ordered random variables uniformly distributed in [0,1]
  u = fliplr(cumprod(rand(1,N_res).^(1./(N_res:-1:1))));
  j=1;
  for i=1:N_res
    while (u(1,i)>cumDist(1,j))
      j=j+1;
    end
    N_babies(1,j)=N_babies(1,j)+1;
  end;
end;

% COPY RESAMPLED TRAJECTORIES:
% ============================
index=1;
for i=1:S
  if (N_babies(1,i)>0)
    for j=index:index+N_babies(1,i)-1
      outIndex(j) = inIndex(i);
    end;
  end;
  index= index+N_babies(1,i);
end
```

APPENDIX B

COMPLEMENTARY GRAPHICAL DISPLAYS

The plots in Sections B.1 and B.2 aim to better illustrate the results reported in Tables 3.2 (on page 58) and 3.4 (page 83), for the local level model (Simulation study I with $\phi = 1$) and the AR(1) plus noise model (Simulation II with $\phi \in \{0.3, 0.8\}$), respectively. These plots are defined by the 13 values of of the signal-to-noise ratio, and the three values of ϕ chosen; making a total of 39 figures.

Recall that in each figure, for each signal-to-noise ratio setting, we show a graphical illustration of the generating process (observations and states) as well as of the filtering performance. For three particular sets of data, in the first row of the upper panel of these 13 figures we plot together the evolution of the generated observations y_t and states $x_{t|t}, t = 1, \ldots, T$. In the second row of this panel, the evolution of the difference between the estimated state values and corresponding true state values $(\hat{x}_{t|t} - x_t, t = 1, \ldots, T)$ obtained via the studied filters, is displayed. In the bottom panel, only for the last exemplar run and last time-index T, we present the marginal posterior densities obtained via the four PF variants under study in contraposition with the exact Gaussian posterior density yielded by the gold standard KF.

The 13 plots for the local level model are presented in Section B.1 and the 26 plots for the AR(1) plusnoise model in Section B.2. All these plots are shown on the website `http://www-eio.upc.edu/~lacosta/AppendixB.pdf` [last visited: September 2013].

APPENDIX C

COMPLEMENTARY MATERIAL FOR SARV(1) MODEL

This Appendix contains additional material for Chapter 6. Section C.1 complements the numeric results in Section 6.5.2 of Chapter 6 by summarizing the performance of three studied filters (SIS, SIR, and ASIR) in handling the estimation of the states for the SARV(1) model; the model parameters are assumed to be known. Section C.2 contains a complementary MC study of the potential impact on estimation of the chosen value for the discount factor δ needed in the jittering step.

C.1 Simulation Results for Cases 2–4.

The following three Tables C.1–C.3 complement the numeric results reported/plotted in Section 6.5.2 of Chapter 6 summarizing the performance of the three studied filters (SIS, SIR and ASIR) in handling the estimation of the states (volatility) for the SARV(1) model when the model parameters are known. Notice that these three tables do not report the CPU-times as they take about the same values as the ones reported in Table 6.2 corresponding to Case 1.

Table C.1: Summary of simulation I results for Case 2: Estimation of the states (volatility) for the SARV(1) model; $\Theta = (\mu, \phi, \sigma_\eta^2)' = (-0.632, 0.90, 0.194^2)$; $0.194^2 = 0.038$

			$Np = 200$		$Np = 500$		$Np = 1000$		$Np = 5000$	
Filter	**T**	**Criterion**	**Mean**	**Var**	**Mean**	**Var**	**Mean**	**Var**	**Mean**	**Var**
SIS	**500**	**RMSE**	0.481	0.002	0.473	0.002	0.462	0.002	0.440	0.002
	1000	**RMSE**	0.517	0.001	0.503	0.001	0.498	0.001	0.483	0.001
	2000	**RMSE**	0.545	0.001	0.535	0.001	0.533	5e-04	0.517	0.001
SIR	**500**	**RMSE**	0.384	0.001	0.383	0.001	0.383	0.001	0.382	0.001
		%uNp	91.85	7.01	91.98	6.84	91.90	6.97	91.89	6.95
	1000	**RMSE**	0.380	5e-04	0.379	4e-04	0.379	4e-04	0.379	4e-04
		%uNp	93.55	4.90	93.60	4.94	93.61	4.81	93.61	4.84
	2000	**RMSE**	0.380	2e-04	0.379	2e-04	0.379	2e-04	0.378	2e-04
		%uNp	93.41	5.05	93.42	4.91	93.37	4.95	93.37	4.97
ASIR	**500**	**RMSE**	0.384	0.001	0.383	0.001	0.383	0.001	0.382	0.001
		%uNp	95.97	3.69	96.03	3.50	96.12	3.39	96.06	3.30
	1000	**RMSE**	0.380	4e-04	0.379	4e-04	0.379	4e-04	0.379	4e-04
		%uNp	96.61	2.53	96.94	2.28	96.89	2.18	96.86	2.25
	2000	**RMSE**	0.380	2e-04	0.379	2e-04	0.379	2e-04	0.378	2e-04
		%uNp	96.77	2.41	96.73	2.49	96.73	2.41	96.74	2.37

Table C.2: Summary of simulation I results for Case 3: Estimation of the states (volatility) for the SARV(1) model; $\Theta = (\mu, \phi, \sigma_\eta^2)' = (-0.632, 0.981, 0.363^2)$; $0.363^2 = 0.132$

			$Np = 200$		$Np = 500$		$Np = 1000$		$Np = 5000$	
Filter	**T**	**Criterion**	**Mean**	**Var**	**Mean**	**Var**	**Mean**	**Var**	**Mean**	**Var**
SIS	**500**	**RMSE**	1.662	0.138	1.559	0.076	1.498	0.094	1.362	0.054
	1000	**RMSE**	1.918	0.067	1.878	0.084	1.819	0.072	1.723	0.054
	2000	**RMSE**	2.176	0.067	2.131	0.072	2.071	0.055	1.969	0.061
SIR	**500**	**RMSE**	0.703	0.003	0.701	0.002	0.700	0.002	0.700	0.002
		%uNp	84.42	11.44	84.50	11.45	84.42	11.84	84.40	11.72
	1000	**RMSE**	0.697	0.001	0.695	0.001	0.694	0.001	0.694	0.001
		%uNp	86.21	10.69	86.22	10.51	86.19	10.39	86.20	10.35
	2000	**RMSE**	0.692	0.001	0.690	0.001	0.690	0.001	0.689	0.001
		%uNp	86.76	7.50	86.72	7.82	86.62	7.95	86.59	8.00
ASIR	**500**	**RMSE**	0.704	0.003	0.702	0.003	0.700	0.003	0.700	0.002
		%uNp	92.83	5.03	92.54	5.06	92.64	4.93	92.73	4.72
	1000	**RMSE**	0.698	0.001	0.695	0.001	0.694	0.001	0.693	0.001
		%uNp	93.45	4.03	93.54	4.01	93.53	3.95	93.44	4.02
	2000	**RMSE**	0.692	0.001	0.690	0.001	0.690	0.001	0.689	0.001
		%uNp	93.42	3.89	93.54	3.39	93.66	3.47	93.58	3.33

Table C.3: Summary of simulation I results for Case 4: Estimation of the states (volatility) for the SARV(1) model; $\Theta = (\mu, \phi, \sigma_\eta^2)' = (-0.632, 0.90, 0.363^2)$; $0.363^2 = 0.132$

			$Np = 200$		$Np = 500$		$Np = 1000$		$Np = 5000$	
Filter	**T**	**Criterion**	**Mean**	**Var**	**Mean**	**Var**	**Mean**	**Var**	**Mean**	**Var**
SIS	**500**	**RMSE**	0.922	0.007	0.914	0.006	0.909	0.008	0.860	0.007
	1000	**RMSE**	1.002	0.004	0.978	0.003	0.980	0.004	0.945	0.004
	2000	**RMSE**	1.052	0.002	1.032	0.001	1.033	0.002	1.004	0.002
SIR	**500**	**RMSE**	0.606	0.002	0.605	0.001	0.604	0.001	0.604	0.001
		%uNp	86.48	10.68	86.60	10.57	86.50	10.95	86.49	10.86
	1000	**RMSE**	0.601	0.001	0.600	0.001	0.599	0.001	0.599	0.001
		%uNp	88.47	8.81	88.45	8.66	88.48	8.52	88.48	8.45
	2000	**RMSE**	0.600	4e-04	0.599	4e-04	0.598	4e-04	0.598	4e-04
		%uNp	88.88	6.97	88.86	7.00	88.78	7.13	88.77	7.16
ASIR	**500**	**RMSE**	0.606	0.002	0.605	0.002	0.604	0.002	0.604	0.001
		%uNp	92.38	5.84	92.39	5.59	92.61	5.60	92.58	5.44
	1000	**RMSE**	0.602	0.001	0.600	0.001	0.599	0.001	0.599	0.001
		%uNp	93.47	4.47	93.56	4.10	93.55	4.17	93.60	4.21
	2000	**RMSE**	0.600	4e-04	0.599	4e-04	0.599	4e-04	0.598	4e-04
		%uNp	93.56	4.46	93.72	3.76	93.72	3.76	93.70	3.73

Next, we address the issue of the potential impact on estimation of the chosen value for the discount factor δ needed in the jittering step.

C.2 Revisiting the Impact of the Discount Factor δ

Under the referral procedure, we were suggested to carry out a further study of some aspects already presented along our research work and to include "a discussion of such aspects in the dissertation". One of such aspects is the following:

> "(3) Revisit the discount factor effect in the simulation studies presented in Chapter 5. Only two discount factors were considered (0.83 and 0.95, why such values?) and based on the results presented there seems to be none or little effect of the discount factor value. What happens if much lower, say 0.5, or larger, say 0.999, discount factors are considered?"

In order to answer the stated question and to place ourselves in the right context, **first**, we remark that at the moment of the revision the contents of Chapter 5 were complete but those of Chapter 6 were not, and thus the results regarding the estimation of states and parameters for the SARV(1) model were not available for the referees. Be reminded that the discount factor issue only applies to Chapters 5 and 6 were the jittering step is needed. In those chapters, we already partially address the issue about the potential impact on estimation of the chosen value for the discount factor δ. Most of the publications consulted about the use of the LW PF variant and consequently using a discount factor δ, refer back to the authors of the LW PF variant, Liu and West (2001). Therein, it is stated not only that the discount factor δ must lie in the $(0,1]$ interval with typically used values in the range 0.95-0.995, but also that a higher discount factor –around 0.99– will be relevant. Since our proposed SIRJ PF also relies on the choice of a discount factor, we proceed to justify why to use specific values for the discount factor δ. At the start of our research, we closely followed the recommendation given by the authors in Liu and West (2001) (and in many other related publications), and that is the reason for initially choosing a discount factor with value $\delta = 0.95$ or even 0.99; we focus at that time on diversifying parameter particles using values lying within the range of values recommended by the literature consulted.

When looking more closely at the formula for the diversification and obtained estimation results, we began to wonder if values different from the typically chosen could be used in order to diversify a bit more the fixed parameters particles, and then we tried different values for δ suggesting that $\delta = 0.83$ was a reasonable good choice to do diversify the particles as targeted but controlling not to smooth them to much. However, it was not until recently that we found a publication confirming that the choice of a discount factor outside the typically recommended range $(0,1]$ was a sound possibility for other authors also; see the article of Carvalho and Lopes (2007). On page 4536, the authors report the use of a discount factor $\delta = 0.85$[1], and somehow justify their choice by referring back to an argument presented in the book of West and Harrison (1997) stating that

[1] They report the use of $\delta = 0.85$, but I believe there is a typo error; they also mention the use of values of $\delta \in (0.50, 0.99)$.

> "(...) δ should be chosen between 0.8 and 0.99 as a function of the amount of information that the modeller is willing to preserve in the filtering process."

This argument theoretically justifies the use of $\delta = 0.83$ in our work, but our choices are also well justified by the MC studies carry out in Chapters 5 and 6.

Second, the simulation results in Chapter 5 –dealing with the estimation of the latent state (level) and two unknown variance parameters for the linear and non-stationary local level model– indicate that there is practically no difference in the quality of the estimations if one uses a δ value of 0.83 or 0.95. Likewise, in Chapter 6 –dealing with the estimation of the states (volatility) and the three parameters (μ, ϕ, σ_η^2) of the nonlinear SARV(1) model– we consider δ values of 0.83, 0.95 and 0.99 and find that generally (irrespective of the case-scenarios and filter-type) lower mean-RMSE values are attained for a discount factor $\delta = 0.83$; for that reason, in that chapter we chose to represent in the shown figures only the results corresponding to $\delta = 0.83$.

Third, to answer the last part of the question we will focus on the SARV(1) were some discrepancies on the estimation was already detected as a function of the chosen discount factor. Specifically, we revisit the estimation of states and fixed parameters of the nonlinear SARV(1) model using discount factor values of $\delta \in \{0.5, 0.75, 0.83, 0.9, 0.95, 0.99\}$ and represent the observed results in Figures C.1–C.2; notice that we consider a larger set of requested δ values. The shown plots basically confirm previously mentioned findings. Following, we list the findings suggested by these figures (results shown for $T = 1000$ and $N_p = 5000$) representing the attained mean-RMSE values when handling the nonlinear SARV(1) model at specified values of the discount factor for the SIRJ and LW PF variants:

x_t Within each case, the statistical quality of the estimated states (mean-RMSE) is practically not affected by the chosen discount factor δ. Notice that both competing filters show equal performance. Across case scenarios, it is observed that larger mean-RMSE values are obtained in case three, followed by cases four, one and two, respectively. These results confirm those reported already in Chapter 6 when considering only three values for δ.

μ For this parameter, within each case, an effect on the mean-RMSE is observed as a function of the chosen discount factor. Notice that both competing filters show practically equal performance. Across cases, larger mean-RMSE values are obtained in case three, followed by cases one, four, and two, respectively; confirming again Chapter 6 findings.

ϕ For this parameter, within each case, an effect on the mean-RMSE is observed as a function of the chosen discount factor. Notice that differences between the SIRJ and LW are more clearly seen (sometimes SIRJ-RMSE greater than LW-RMSE or viceversa), but they are practically not relevant since when they occur they are rather small; in the third decimal place. Across cases, there is not an overall clear pattern, but Chapter 6 findings still hold. We observe that for cases 1 and 3, a minimum mean-RMSE is attained at $\delta = 0.83$. For cases 2 and 4, however, one observes that the mean-RMSE increases with δ.

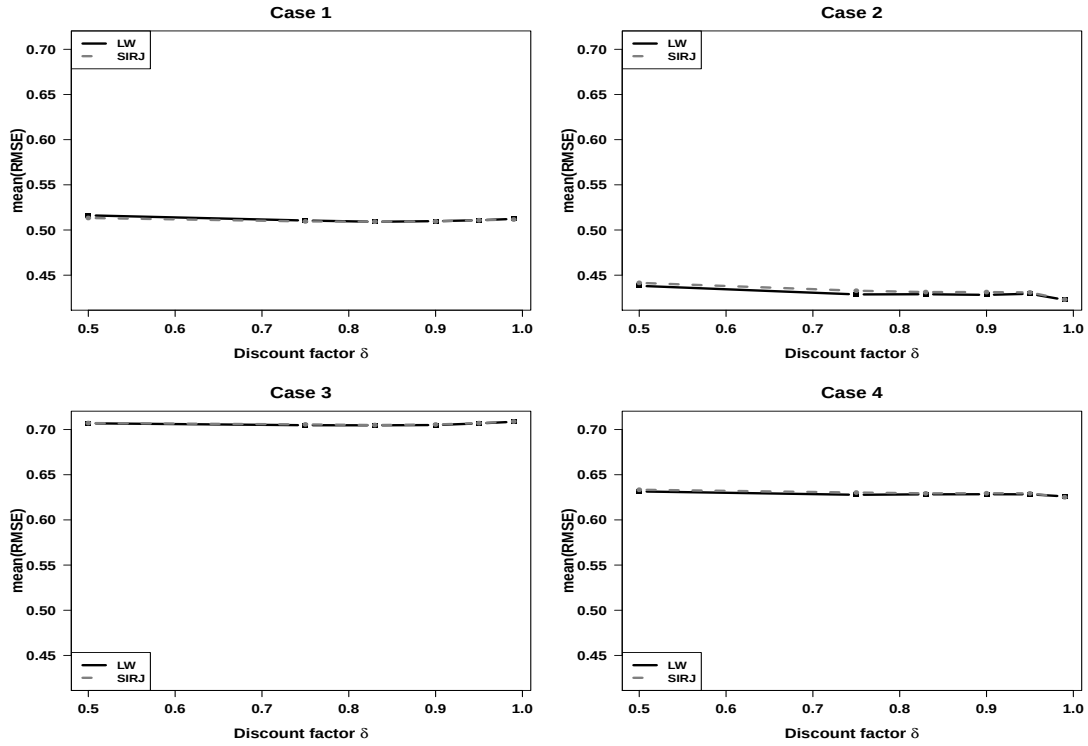
$\boldsymbol{\sigma_\eta^2}$ For this parameter, within each case, obtained mean-RMSE is practically not affected by the choice of the discount factor. Notice also that both competing filters show practically equal performance; differences between filters when occur are rather small; in the third decimal place. Across case scenarios, it is observed that larger mean-RMSE values are obtained in case 3, followed by cases 4, 1 and 2, respectively. Again, these results confirm those reported already in Chapter 6.

All the above indicate that there is not a unified pattern across both parameters and cases; focus on the varied shapes of the figures. If one focuses again on comparing the two filters, they tend to behave very similarly at most δ values; specially at larger values. Next, we go one step further and detail the situations when differences are observed, being them relevant or not, in order to provide a rule of thumb for chosing δ.

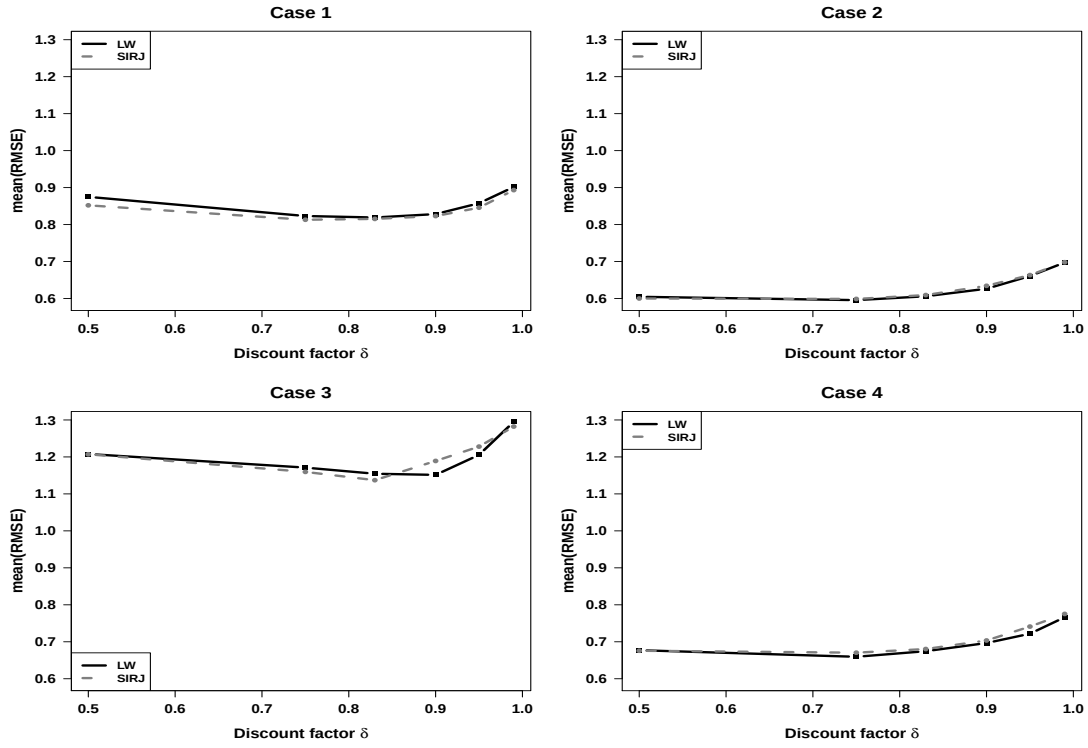
- For $\delta > 0.90$, both competing filters show closer agreement, but the attained mean-RMSE are generally the highest (for model parameters); in this small range of values it seems that generally, the larger the discount factor δ, the higher the RMSE. On the contrary, some discrepancies between filters begin to be observed for lower discount factors, say $\delta \leq 0.90$, but in general lower mean-RMSE are obtained.

- Based on plotted results, if we had to provide a recommendation, for the model at hand we would recommend as a rule of thumb to use δ values in the range $\delta \in [0.75, 0.90]$, were irrespective of the case-scenarios lower values of the mean-RMSE are obtained. We consider that our choice of $\delta = 0.83$ is reasonably justified.

Finally, from a theoretical point of view, the lower the discount factor δ, the greater the degree of smoothness imposed to the fixed particles. Likewise, the larger the δ value, the lower the degree of smoothness; if, for instance, $\delta = 1$ was used, then no jittering would be performed. We consider that the choice of $\delta = 0.83$ allows us to properly diversify the fixed parameters particles and at the same time to obtain very good statistical performance.

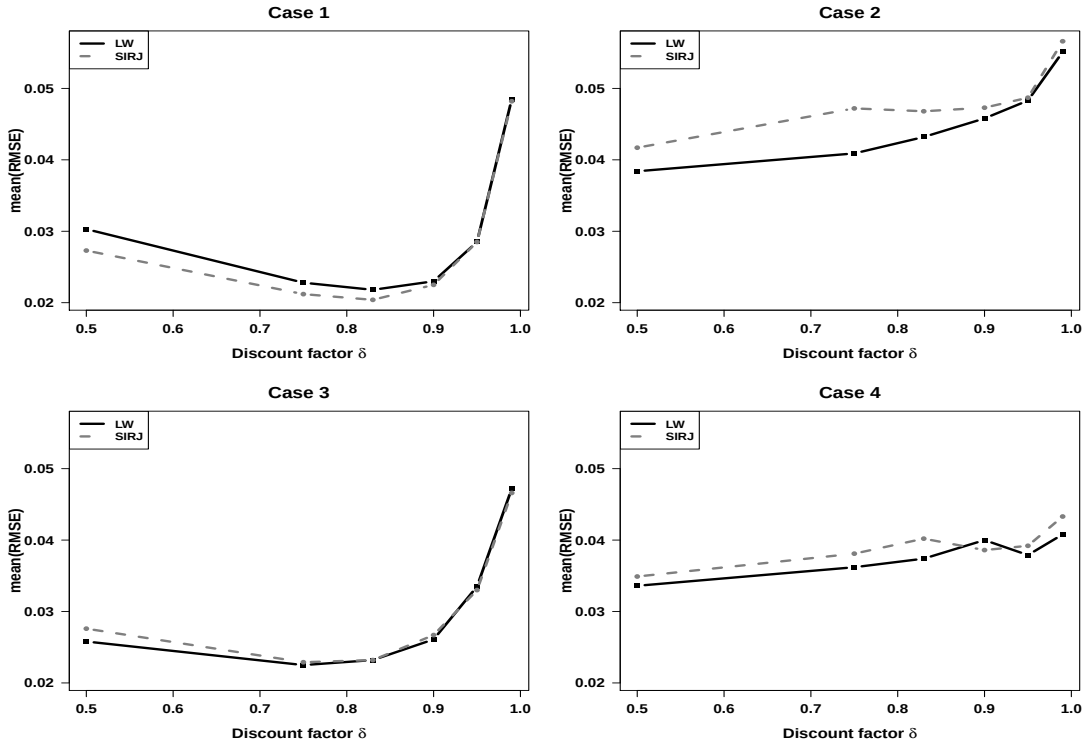


(a) Latent state (volatility): $\boldsymbol{x_t}$

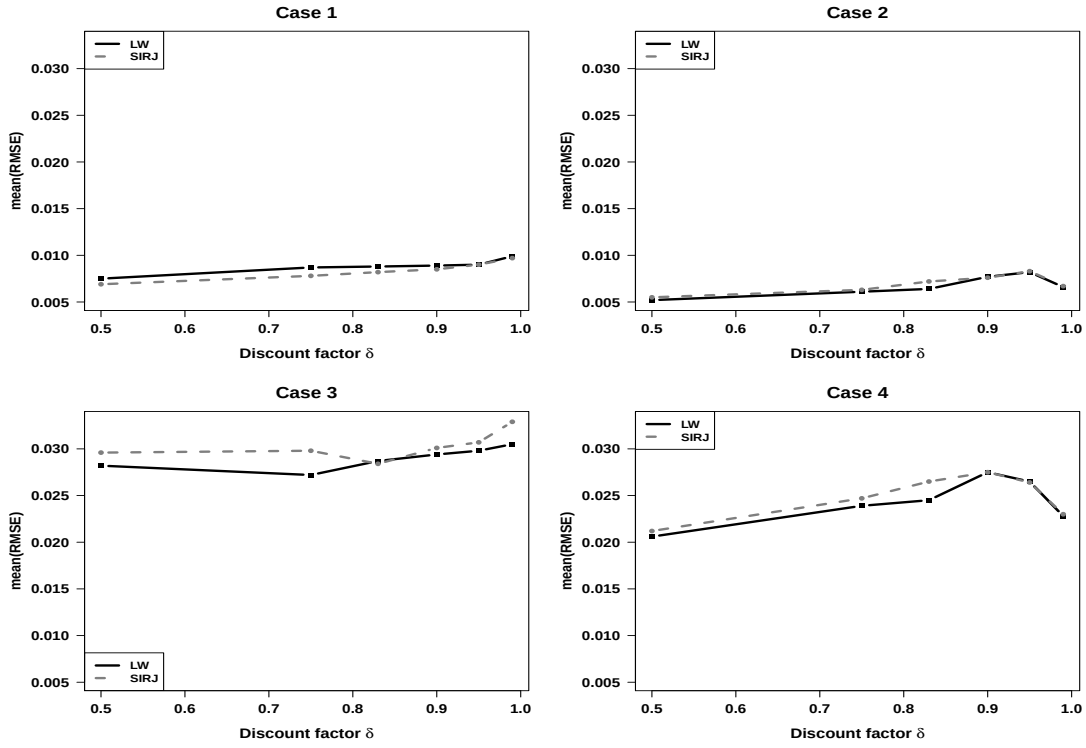


(b) Mean level parameter: μ

Figure C.1: SARV(1) model: Impact of discount factor δ on the estimation of the states $\boldsymbol{x}$ (volatility in upper panel) and the mean-level parameter μ comparing the SIRJ and LW PF variants. Results shown for $T = 1000$ and $N_p = 5000$.



(a) Autoregressive parameter ϕ



(b) Transition variance parameter: σ_η^2

Figure C.2: SARV(1) model: Impact of discount factor δ on the estimation of the persistence and the volatility of volatility parameters ϕ (upper panel) and σ_η^2 comparing the SIRJ and LW PF variants. Results shown for $T = 1000$ and $N_p = 5000$.

The above stated findings, seem to indicate that $\delta = 0.83$ can be a good choice for the discount factor δ; at least for the models entertained in this work. Thus, we must admit that though we consider this a very positive and promising result, we cannot and do not want to extrapolate this to any model. The MC results suggest that the right choice for the discount factor seemingly depends on the type of model at hand. Thus, our suggestion is that the practitioner must entertain different values using as a guide previously found results. Said that, an in-deep study of the effect of the discount factor on the quality of the estimation is a matter of further study.