Linköping studies in science and technology. Thesis.
No. 1480

Rao-Blackwellised particle methods for inference and identification

Fredrik Lindsten

REGLERTEKNIK
AUTOMATIC CONTROL
LINKÖPING

Division of Automatic Control
Department of Electrical Engineering
Linköping University, SE-581 83 Linköping, Sweden
http://www.control.isy.liu.se
lindsten@isy.liu.se

Linköping 2011

This is a Swedish Licentiate's Thesis.

Swedish postgraduate education leads to a Doctor's degree and/or a Licentiate's degree.
A Doctor's Degree comprises 240 ECTS credits (4 years of full-time studies).
A Licentiate's degree comprises 120 ECTS credits,
of which at least 60 ECTS credits constitute a Licentiate's thesis.

Linköping studies in science and technology. Thesis.
No. 1480

Rao-Blackwellised particle methods for inference and identification

Fredrik Lindsten

lindsten@isy.liu.se
www.control.isy.liu.se
Department of Electrical Engineering
Linköping University
SE-581 83 Linköping
Sweden

ISBN 978-91-7393-173-1 ISSN 0280-7971 LiU-TEK-LIC-2011:19

Printed by LiU-Tryck, Linköping, Sweden 2011

Till Åsa

Abstract

We consider the two related problems of state inference in nonlinear dynamical systems and nonlinear system identification. More precisely, based on noisy observations from some (in general) nonlinear and/or non-Gaussian dynamical system, we seek to estimate the system state as well as possible unknown static parameters of the system. We consider two different aspects of the state inference problem, filtering and smoothing, with the emphasis on the latter. To address the filtering and smoothing problems, we employ sequential Monte Carlo (SMC) methods, commonly referred to as particle filters (PF) and particle smoothers (PS).

Many nonlinear models encountered in practice contain some tractable substructure. If this is the case, a natural idea is to try to exploit this substructure to obtain more accurate estimates than what is provided by a standard particle method. For the filtering problem, this can be done by using the well-known Rao-Blackwellised particle filter (RBPF). In this thesis, we analyse the RBPF and provide explicit expressions for the variance reduction that is obtained from Rao-Blackwellisation. Furthermore, we address the smoothing problem and develop a novel Rao-Blackwellised particle smoother (RBPS), designed to exploit a certain tractable substructure in the model.

Based on the RBPF and the RBPS we propose two different methods for nonlinear system identification. The first is a recursive method referred to as the Rao-Blackwellised marginal particle filter (RBMPF). By augmenting the state variable with the unknown parameters, a nonlinear filter can be applied to address the parameter estimation problem. However, if the model under study has poor mixing properties, which is the case if the state variable contains some static parameter, SMC filters such as the PF and the RBPF are known to degenerate. To circumvent this we introduce a so called "mixing" stage in the RBMPF, which makes it more suitable for models with poor mixing properties.

The second identification method is referred to as RBPS-EM and is designed for maximum likelihood parameter estimation in a type of mixed linear/nonlinear Gaussian state-space models. The method combines the expectation maximisation (EM) algorithm with the RBPS mentioned above, resulting in an identification method designed to exploit the tractable substructure present in the model.

Populärvetenskaplig sammanfattning

Vi kommer i denna avhandling att titta närmre på två relaterade problem; tillståndsskattning i olinjära system och olinjär systemidentifiering. Givet brusiga mätningar från ett olinjärt dynamiskt system, vill vi skatta systemets tillstånd och även eventuella okända, statiska systemparametrar. Vi behandlar två aspekter av tillståndsskattningsproblemet, filtrering och glättning, med fokus på det sistnämnda. För att angripa dessa båda problem använder vi oss av så kallade sekventiella Monte Carlo (SMC) metoder, ofta benämnda partikelfilter (PF) och partikelglättare.

Många olinjära modeller som man stöter på i praktiska tillämpningar innehåller en viss understruktur. Om så är fallet, är det naturligt att försöka utnyttja denna struktur för att erhålla bättre skattningar. Genom att kombinera denna idé med partikelfiltret erhålls det välkända Rao-Blackwelliserade partikelfiltret (RBPF). Ett av bidragen i denna avhandling är en analys av RBPF vilken leder till ett explicit uttryck för den variansreduktion som fås genom Rao-Blackwellisering. Dessutom betraktar vi glättningsproblemet och föreslår en Rao-Blackwelliserad partikelglättare (RBPS), vilken är utvecklad med syfte att utnyttja en viss typ av understruktur i modellen.

Baserat på RBPF och RBPS föreslår vi två olika metoder för olinjär systemidentifiering. Den första är en rekursiv metod, kallad det Rao-Blackwelliserade marginella partikelfiltret (RBMPF). Genom att utöka tillståndsvariabeln med de okända parametrarna kan ett olinjärt filter användas för parameterskattning. De statiska parametrarna kommer dock leda till att modellen får dåliga mixningsegenskaper. Detta leder i sin tur till att SMC baserade filter, såsom PF och RBPF, kommer att degenerera. För att kringgå detta problem inför vi ett så kallat “mixningssteg” i RBMPF, vilket gör filtret mer lämpligt för modeller med dåliga mixningsegenskaper.

Den andra metoden som vi föreslår går under namnet RBPS-EM, och kan användas för parameterskattning i en typ av blandade linjära/olinjära Gaussiska tillståndsmodeller. Metoden kombinerar EM algoritmen med glättning via ovannämnda RBPS. Detta resulterar i en identifieringsmetod som kan utnyttja den speciella understruktur som finns i modellen.

Acknowledgments

First of all, I would like to thank my supervisor Dr. Thomas Schön for the guidance, support and genuine enthusiasm. He has, literally, been knocking on my door, asking for another chapter to proofread or checking whether I have some new, interesting results to show him. This has been very encouraging!

My two co-supervisors, Prof. Lennart Ljung and Prof. Fredrik Gustafsson, have provided additional guidance and valuable expertise. It is also on their consciences that I came to be a member of the, truly great, group of Automatic Control at Linköping University in the first place. For this, I am very grateful. Today, this group is skillfully presided over by Prof. Svante Gunnarsson who does a great job as the head of division. The division coordinator Ninna Stensgård and her predecessors Åsa Karmelind and Ulla Salaneck, deserve my deepest gratitude for helping me with various tasks and arrangements.

I am most grateful for the financial support from the projects Calibrating Nonlinear Dynamical Models (Contract number: 621-2010-5876) and the Linneaus Center CADICS, both funded by the Swedish Research Council, and the past Strategic Research Center MOVIII, funded by the Swedish Foundation for Strategic Research.

I would also like to thank Dr. Jimmy Olsson at the Center of Mathematical Sciences, Lund University and Dr. Lennart Svensson at the Department of Signals and Systems, Chalmers University of Technology, for the time you have spent on our collaborations, which I hope will continue and intensify in the future.

The errors, unclarities and other deficiencies of this thesis would have been much more frequent if it were not for Lic. Zoran Sjanic, Lic. Christian Lyzell, Dr. David Törnqvist, Tohid Ardeshiri, Dr. Lennart Svensson and Dr. Henrik Ohlsson who have proofread various parts of the thesis. Dr. Henrik Tidefelt and Dr. Gustaf Hendeby have constructed the LaTeX-class, used for the layout of the thesis. Thank you!

The (as already mentioned) great group of Automatic Control, naturally consists of equally great people. André Carvalho Bittencourt, Peter Rosander, Thomas and all the other climbers – thanks for all the fun, both on the wall and at ground level. Lic. Daniel Petersson has been a great resource for discussing things like matrix calculus, optimisation and the sweet bitterness of an IPA at *Trädojan*. Sina Kh. Pa.[1] deserves many thanks for all the beef, whiskey and kubb games. My office buddy, referred to as The Zoran, has provided lots of laughter. Thanks to all of you and to all that I have forgotten to mention by name. Thanks to my friends from Y04C, to other friends outside the walls of the RT-corridor and to my family and my very supporting parents Anitha Lindsten and Anders Johansson!

Most of all, I thank my wonderful wife Åsa Lindsten, for reasons more numerous than the number of indices in Equation (5.7). Your patience when I have met your ideas with phrases like “Yes darling, we can do that *after* the thesis is printed”, is truly admirable. Your love is what keeps me going!

Linköping, May 2011
Fredrik Lindsten

[1] Due to space limitations, I have not written out your full surname, but you know who you are.

Contents

Notation

SPACES

Notation	Meaning
$\mathbb{N}$	Natural numbers
$\mathbb{R}$, $\mathbb{R}_+$, $\mathbb{R}_{++}$	Real numbers/nonnegative numbers/positive numbers
$\mathbb{R}^d$, $\mathbb{R}^{m\times n}$	d-dimensional Euclidian space/space of $m \times n$ matrices
$S_+(n)$	Nonnegative definite, symmetric $n \times n$ matrices
$S_{++}(n)$	Positive definite, symmetric $n \times n$ matrices

PROBABILITY AND STATE-SPACE MODELS

Notation	Meaning
$\sim$	Sampled from or distributed according to
$(\Omega, \mathcal{F}, \mathrm{P})$	Probability space
$\mathcal{B}(\mathbb{R}^d)$	Borel σ-algebra on $\mathbb{R}^d$
$\sigma(X)$	σ-algebra generated by X
$(\mathsf{X}, \mathcal{X})$	State-space
$(\mathsf{Y}, \mathcal{Y})$	Observation space
$\mathbb{F}(\mathsf{X})$	$\mathcal{X}/\mathcal{B}(\mathbb{R})$-measurable functions from X to $\mathbb{R}$
$\{X_t\}_{t\geq 1}$	State process
$\{Y_t\}_{t\geq 1}$	Measurement process
$\{\Xi_t\}_{t\geq 1}$	Nonlinear state process in CLGSS model
$\{Z_t\}_{t\geq 1}$	Linear state process in CLGSS model
Q	Transition kernel for the state process *or* process noise covariance (given by the context)
G	Measurement kernel
p	Generic density function for state-space models
$\mathrm{E}, \mathrm{Var}, \mathrm{Cov}$	Expectation/variance/covariance
$\ll$	Absolute continuity
$\xrightarrow{\mathrm{P}}$, $\xrightarrow{\mathrm{D}}$	Convergence in probability/distribution

DISTRIBUTIONS

Notation	Meaning
$\mathcal{N}(m, \Sigma)$	Multivariate Gaussian with mean m and covariance Σ
$\mathrm{Gam}(k, \theta)$	Gamma with shape k and scale θ
$\mathcal{U}([a,\, b])$	Uniform over the interval $[a,\, b]$
$\mathrm{Cat}(\{p_i\}_{i=1}^N)$	Categorical over $\{1,\, \dots,\, N\}$ with probabilities $\{p_i\}_{i=1}^N$
$\mathrm{Bin}(N, p)$	Binomial for N trials with success probability p
δ_x	Point-mass at x (Dirac δ-distribution)

OPERATORS, FUNCTIONS AND MISCELLANEOUS SYMBOLS

Notation	Meaning
$\cup$	Set union
$\cap$	Set intersection
$\mathrm{card}(S)$	Cardinality of the set S
S^{c}	Complement of S in Ω (given by the context)
$I_S(\,\cdot\,)$	Indicator function of set S
$I_{d\times d}$	d-dimensional identity matrix
A^{T}	Transpose of matrix A
$\det(A)$	Determinant of matrix A
$\mathrm{tr}(A)$	Trace of matrix A
$\mathrm{diag}(v)$	Diagonal matrix with elements of v on the diagonal
$\lVert f \rVert_\infty$	Supremum norm, $\sup_x \lvert f(x) \rvert$
$a_{m:n}$	Sequence, $\{a_m,\, a_{m+1},\, \dots,\, a_n\}$
$\cong$	Equality up to an additive constant
$\triangleq$	Definition
$:=$, $\leftarrow$	Assignment

ABBREVIATIONS

Abbreviation	Meaning
a.e.	Almost everywhere
a.s.	Almost surely/with probability 1
CLGSS	Conditionally linear Gaussian state-space
CLT	Central limit theorem
EM	Expectation maximisation
FFBSi	Forward filter/backward simulator
FFBSm	Forward filter/backward smoother
GMM	Gaussian mixture model
GPB	Generalised pseudo-Bayesian
GR	Geo-referencing
i.i.d.	Independent and identically distributed
IMM	Interacting multiple models

IMU	Inertial measurement unit
IS	Importance sampling
KF	Kalman filter, optimal filter for linear Gaussian systems
LGSS	Linear Gaussian state-space
MC	Monte Carlo
MCMC	Markov chain Monte Carlo
ML	Maximum likelihood
MPF	Marginal particle filter
PDF	Probability density function
PF	Particle filter
PS	Particle smoother
PS-EM	Particle smoother EM
RB	Rao-Blackwellised
RB-FFBSi	Rao-Blackwellised FFBSi
RBMPF	Rao-Blackwellised marginal particle filter
RBPF	Rao-Blackwellised particle filter
RBPS	Rao-Blackwellised particle smoother
RBPS-EM	Rao-Blackwellised particle smoother EM
RMSE	Root mean squared error
RS	Rejection sampling
RTS	Rauch-Tung-Striebel, optimal smoothing recursions for linear Gaussian systems
RTS-EM	Rauch-Tung-Striebel EM
SIR	Sampling importance resampling
SIS	Sequential importance sampling
SMC	Sequential Monte Carlo
SNR	Signal-to-noise ratio
SSM	State-space model
UAV	Unmanned aerial vehicle
VO	Visual odometry

1

Introduction

Assume the we have at our disposal a sensor, or a measuring device, from which we can read off values y_t at some points in time indexed by $t = 1, 2 \dots$. Based on these readings, we wish to draw conclusions about the underlying system, which has generated the measurements.

As an example, consider the often encountered problem of making predictions about the output from some system based on previous observations. Hence, assume that we have recorded the values $y_{1:t} \triangleq \{y_1, \dots, y_t\}$. Then, what is the best guess for what y_{t+1} will turn out to be? Should we simply assume that y_{t+1} will be close to the most recent recording y_t, or should we make use of older measurements as well, to account for possible trends? Such questions can be answered by using a model, which describes how to weigh the available information together to make as good predictions as possible.

For most applications, it is not possible to find models that exactly describe the measurements. There will always be fluctuations and variations in the data, not accounted for by the model. To incorporate such random components, the measurement sequence can be viewed as a realisation of a discrete-time stochastic process $\{Y_t\}_{t \geq 1}$. Hence, a model for the system is the same as a model for the stochastic process.

In this thesis we will be working with a specific class of models, known as state-space models (SSMs). The structure of an SSM can be seen as influenced by the notion of a physical system. The idea is that, at each time instant, the system is assumed to be in a certain "state". The state contains all relevant information about the system, i.e. if we would know the state of the system we would have full insight into its internal condition. However, the state is typically not known. Instead, we measure some quantities which depend on the state in some way. To exemplify the idea, let X_t be a random variable representing the state of a system at time t. An SSM for the system could then, for instance, be given by,

$$X_{t+1} = f(X_t) + V_t, \tag{1.1a}$$

$$Y_t = h(X_t) + E_t. \tag{1.1b}$$

The expression (1.1a) describes the evolution of the system state over time. The state at time $t+1$ is given by a transformation of the current state $f(X_t)$, plus some *process noise* V_t. The process noise accounts for variations in the system state, not accounted for by the model. Since the state at a consecutive time point depends on the previous state, we say that the system is dynamic and (1.1a) is known as the dynamic equation. The second part of the model, given by (1.1b), describes how the measurement Y_t depends on the state X_t and some *measurement noise* E_t. Consequently, (1.1b) is called the measurement equation. The concept of SSMs will be further discussed in Section 2.2.

In (1.1), the state process $\{X_t\}_{t\geq 1}$ is not observable; it is sometimes called latent or hidden. Instead, any conclusions that we wish to draw regarding the system, must be inferred from observations of the measurement sequence $\{Y_t\}_{t\geq 1}$. We will in this thesis be concerned with two different problems of this type.

1. **State inference:** Given a fully specified SSM for the process $\{Y_t\}_{t\geq 1}$ and based on observations $\{y_t\}_{t\geq 1}$, draw conclusions about the process itself. This could for instance be to predict future values of the process, as in the preceding example. More generally, it is the problem of estimating some past, present or future state of the system, which is not directly visible but related to the measurements through the model.

2. **System identification:** Based on observations $\{y_t\}_{t\geq 1}$, find a model for the process $\{Y_t\}_{t\geq 1}$ that can describe the observations. This problem is known as system identification, which in itself is a very broad concept. In this thesis we will consider one important part of the system identification task, namely how to estimate unknown, static parameters of the model.

As we shall see, these two problems are closely related and there is not always a clear distinction.

Remark 1.1. In the system identification literature, it is common to let the system be excited by some known input signal $\{u_t\}_{t\geq 1}$, i.e. by adding a dependence on u_t on the right hand side of (1.1). In this thesis, we will not make such dependence explicit, but this is purely for notational convenience. The identification methods that we will consider are indeed applicable also in the presence of a known input signal.

If both f and h in the model (1.1) are linear (of affine) functions, the SSM is also called linear. Reversely, if this is not the case, the model is called nonlinear. Even though there exists many relevant applications in which nonlinear models arise, the focus in the engineering community has traditionally been on linear models. One contributory factor to this, is that nonlinear models by nature are much harder to work with. However, as we develop more sophisticated computational tools and acquire more and more computational resources, we can also address increasingly more challenging problems. Inspired by this fact, this thesis puts focus on nonlinear systems and we will consider the two problems of nonlinear state inference and nonlinear system identification.

1.1 Monte Carlo and Rao-Blackwellisation

As pointed out in the previous section, the fundamental problem considered in this thesis is that of estimation. In both the state inference and the identification problem we are seeking to estimate "something", based on observations from the system. Let this something, called the estimand, be denoted θ. The estimand could for instance be a prediction of a future value, as discussed in the previous section, or some unknown parameter of the system dynamics. Based on readings from the system $y_{1:T}$ we wish to estimate the value of θ. For this cause, we construct an estimator $\hat{\theta}$ such that, in some sense, $\hat{\theta}$ is close to θ. Naturally, the estimator is a function of the observations, i.e. after having observed a measurement sequence $y_{1:T}$ we take $\hat{\theta}(y_{1:T})$ as our estimate of θ.

1.1.1 Randomised estimators and Monte Carlo

For many problems it is tricky to find an appropriate function $\hat{\theta}$, mapping the measurement sequence into an estimate of θ. For some challenging problems (e.g. the nonlinear state inference and identification problems), it can be beneficial to let the estimator depend on some auxiliary, random variable U. Hence, after having observed $U = u$ we take $\hat{\theta}^\star(y_{1:T}, u)$, which is then known as a randomised estimator, as an estimate of θ. The idea with a randomised estimator is illustrated in the following example.

Example 1.1: Randomised estimator

Let X be an unobservable random variable, dependent on the measurement sequence $Y_{1:T}$. After having observed, $Y_{1:T} = y_{1:T}$ we seek the probability that X lies in some set A. Hence, we seek the conditional probability

$$\theta \triangleq \mathrm{P}(X \in A \mid Y_{1:T}). \tag{1.2}$$

Now, assume that we do not know any analytic form for the conditional distribution of X, given $Y_{1:T}$, but that we have a way of sampling from it. We then draw N samples $\{x_i\}_{i=1}^N$ from this conditional distribution (given $Y_{1:T} = y_{1:T}$). The estimate of the conditional probability (1.2) can then be taken as the frequency of samples landing in the set A. Hence, the randomised estimator of θ is,

$$\hat{\theta}^\star(y_{1:T}, \{x_i\}_{i=1}^N) = \frac{1}{N}\sum_{i=1}^N I_A(x_i), \tag{1.3}$$

where I_A is the indicator function of the set A.

The procedure of constructing a randomised estimator as above, is an example of a so called Monte Carlo method[1]. The essence of these methods is to draw, typically a large number of random samples, which are then used to solve a mathematical problem. The most fundamental Monte Carlo method is probably that of approximating the expectation of a random variable by the sample mean of a large number of realisations of the variable. More precisely, assume that X is a random variable with distribution μ. We seek to

[1] The methods are named after the Monte Carlo casino in Monaco.

compute the expectation of some function $f(X)$, i.e.

$$\theta \triangleq \mathrm{E}[f(X)] = \int f(x)\mu(dx). \tag{1.4}$$

If the above integral is intractable, it can be approximated by drawing N i.i.d. samples $\{x_i\}_{i=1}^N$ from μ and compute an estimate of θ as,

$$\hat{\theta}^\star(\{x_i\}_{i=1}^N) = \frac{1}{N}\sum_{i=1}^N f(x_i). \tag{1.5}$$

By the strong law of large numbers, this estimate converges almost surely to the true expectation as N tends to infinity. This technique is also known as Monte Carlo integration, since we in (1.5) in fact approximate an intractable integral by using Monte Carlo sampling. The estimate (1.3) is also an example of Monte Carlo integration, with f being the indicator function of the set A and μ being the conditional probability distribution of X given $Y_{1:T}$.

1.1.2 Rao-Blackwellisation

Let us return to the original estimation problem, i.e. how to find a function $\hat{\theta}(y_{1:T})$ estimating the estimand θ. Since basically any function can be taken as an estimator, we need some way to measure the closeness of $\hat{\theta}$ to θ, to be able to find a good estimator. For this cause, assume that we have chosen a *loss function* $L(\theta, \hat{\theta})$ which is small when $\hat{\theta}$ is close to θ and vice versa. We can then say that an estimator $\hat{\theta}'$ is better that an estimator $\hat{\theta}$, if the expected loss is lower, i.e. if

$$\mathrm{E}[L(\theta, \hat{\theta}'(Y_{1:T}))] < \mathrm{E}[L(\theta, \hat{\theta}(Y_{1:T}))]. \tag{1.6}$$

In the mid 40's, Rao [1945] and Blackwell [1947] established a fundamental result in estimation theory, which has later become known as the Rao-Blackwell theorem (see also [Lehmann, 1983] p. 50). We will not review the theorem in full here, since (despite the title of this thesis) we do not need the details of it. Instead, we settle for an informal discussion on its implications. What the Rao-Blackwell theorem states is that if $\hat{\theta}$ is some estimator of θ, S is a sufficient statistic for $Y_{1:T}$ and the loss function is convex in $\hat{\theta}$, then the estimator

$$\hat{\theta}_{\mathrm{RB}}(S) = \mathrm{E}[\hat{\theta}(Y_{1:T}) \mid S] \tag{1.7}$$

is typically a better estimator of θ, and is never worse. Hence, from a crude estimator $\hat{\theta}$ we can construct a better estimator $\hat{\theta}_{\mathrm{RB}}$ according to (1.7), depending only on the sufficient statistic S. This transformation is known as Rao-Blackwellisation of the estimator $\hat{\theta}$. In this thesis, we are concerned with the implication of the Rao-Blackwell theorem for randomised estimators, which we give in a corollary.

Corollary 1.1 (Rao-Blackwellisation of randomised estimators). *For any randomised estimator of θ, there exists a non-randomised estimator which is uniformly better if the loss function is strictly convex and at least as good when it is convex.*

Proof: See [Lehmann, 1983], p. 51. □

This corollary is a direct consequence of the Rao-Blackwell theorem. If U is the random variable used to construct a randomised estimator $\hat{\theta}^\star(Y_{1:T}, U)$ (thus, U has a known distribution), then the statistic $Y_{1:T}$ is sufficient for the pair $\{Y_{1:T}, U\}$. Hence, we obtain a non-randomised estimator by Rao-Blackwellisation as,

$$\hat{\theta}^\star_{\text{RB}}(Y_{1:T}) = \mathrm{E}[\hat{\theta}^\star(Y_{1:T}, U) \mid Y_{1:T}]. \tag{1.8}$$

So, what implications does this have for the randomised estimators discussed in Section 1.1.1? To see this, let us consider the Monte Carlo integration in (1.5). Let $\{X_i\}_{i=1}^N$ be the i.i.d. random variables, distributed according to μ, of which we have observed the values $\{x_i\}_{i=1}^N$. Then, a Rao-Blackwellised estimator of the expectation (1.4) is given by,

$$\hat{\theta}^\star_{\text{RB}} = \mathrm{E}[\hat{\theta}^\star(\{X_i\}_{i=1}^N)] = \frac{1}{N}\sum_{i=1}^{N} \mathrm{E}[f(X_i)] = \mathrm{E}[f(X)]. \tag{1.9}$$

Hence, if we ask for a Rao-Blackwellisation of a Monte Carlo estimator, we are simply told; use the true value instead of the Monte Carlo estimate. In some sense, we can say that Rao-Blackwellisation is the counterpart of Monte Carlo methods. It replaces randomised sample averages with true expectations. Clearly, if it is intractable to compute the true expectation, this is the case also for the Rao-Blackwellised estimator (since they coincide). Due to this, there must be a trade-off between the application of Monte Carlo methods to construct randomised estimators, and the application of Rao-Blackwellisation to these estimators. The general idea that we will adopt in this thesis is to apply Rao-Blackwellisation to an "as high degree as possible", hopefully leading to an increased accuracy over the original Monte Carlo methods. We will return to the procedure of Rao-Blackwellisation in a sequential Monte Carlo framework in Section 3.3.

1.2 Particle methods: an application example

Before we leave this introductory chapter, let us have a look at how the Monte Carlo approach can be used to address a challenging state inference problem in a nonlinear dynamical system. For this cause, we will consider an application example, in which we seek to localise an unmanned aerial vehicle (UAV) using information from an on-board video camera.

UAVs have the potential of becoming a very useful tool, e.g. for search and rescue operations in hazardous environments. For the UAV to be able to operate autonomously, it is crucial to be able to determine its position, i.e. to estimate its state. In this example, we assume that the primary sensor for this cause is an on-board video camera, looking down on the ground. Hence, the measurements Y_t can be seen as images from the camera. By comparing these images with a preexisting map over the operational environment, we seek to estimate the position of the vehicle. Basically, this is done by comparing qualitative information from the images with the map. That is, if we see a house in the image, then we know that we are not flying over a lake; if we see a road crossing, then this will provide us with information of possible positions of the vehicle, etc.

However, to solve this problem by using a set of rules and logical inference ("if we see a house, then we must be at position A, B or $C\ldots$") can be very tricky, especially if

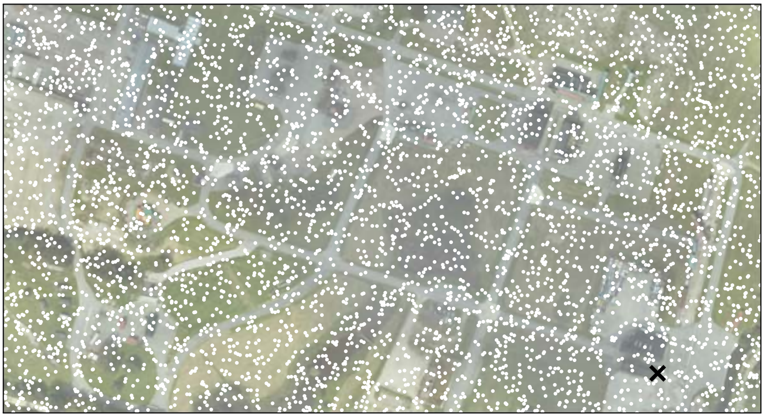

Figure 1.1: Initial particle positions, shown as white dots, spread randomly over the map. The true vehicle position at this time instant is shown as a black cross. Aerial photograph by courtesy of the UAS Technologies Lab, Artificial Intelligence and Integrated Computer Systems Division (AIICS) at the Department of Computer and Information Science (IDA), Linköping University, Linköping, Sweden.

we take into account that the measurements are uncertain, i.e. we might not be sure that it is actually a house that we see in the image. Instead, we address the problem using a sequential Monte Carlo method known as the particle filter (PF). In the PF, we propose a large number of random hypotheses of were the vehicle might be. These hypotheses are called particles, hence the name of the filter. In Figure 1.1 we show the initial hypotheses of the UAV position, randomly placed over the entire map. We then evaluate the *likelihood* that each hypothesis is true. The unlikely hypotheses are thereafter discarded, whereas the likely ones are duplicated. This is shown in Figure 1.2.

Since the vehicle is moving, we must allow the particles to move as well, to be able to track the UAV position. Basically, this is done by propagating the hypotheses through the dynamic model for the vehicle as in (1.1a). That is, if we know that the UAV is at a certain position with a certain velocity at time t, we can predict its position at time $t + 1$. This procedure, of sequentially propagating the hypotheses through time, evaluating their likelihoods and putting focus on the likely ones, is what makes the method sequential (hence the name sequential Monte Carlo, SMC). In Figure 1.3, we show how the particles are updated over time, converging to a consistent estimate of the UAV position.

The present section has been included to give a flavor for how particle methods can be used for state inference. The specific application example that we have considered is influenced by the work by Lindsten et al. [2010]. Clearly, we have left out the majority of the details. However, some of these details are provided in Section 3.3.5 of this thesis, where we return to this application example and further motivate the suitability of the PF for addressing the state inference problem. However, it should be emphasised that the main focus in this thesis is on general, particle based methods for inference and identification. Hence, this specific application is not in any way the basis or the motivation for the material of the thesis, it is merely used as an example.

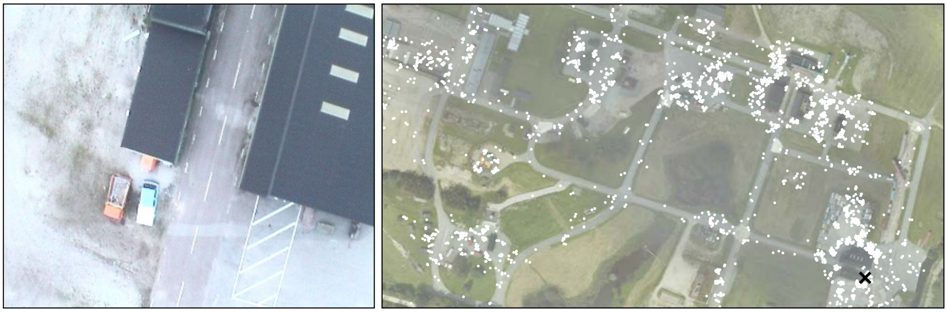

Figure 1.2: *Image from the on-board camera (left) and particle positions (right) after 1 second of flight. The image processing system on the UAV detects asphalt and buildings in the image. Hence, several of the initial hypotheses can be discarded since they do not match the image. Instead, focus is put on areas along the roads and especially near the buildings.*

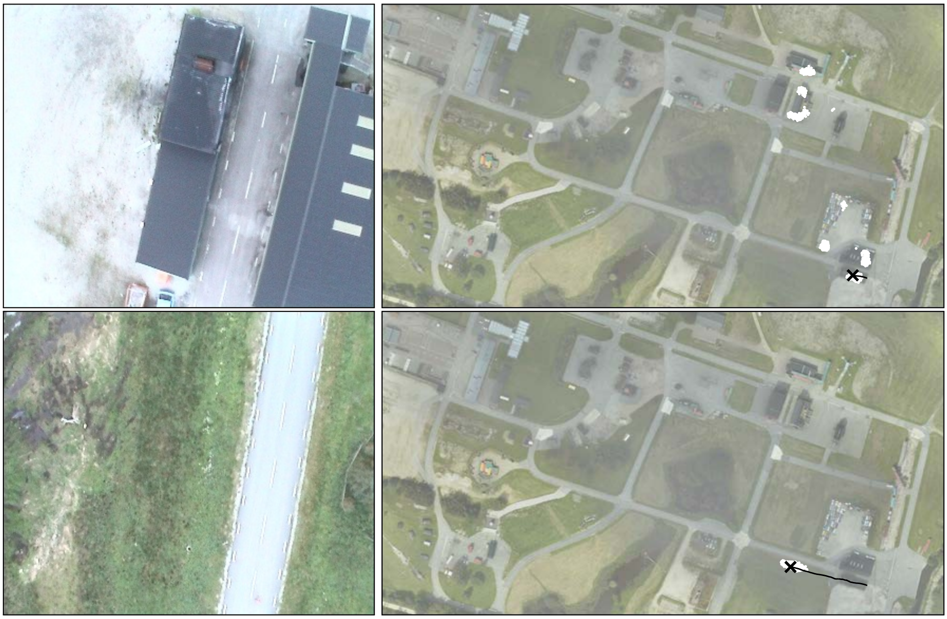

Figure 1.3: *Top row - Image from the on-board camera (left) and particle positions (right) after 5 seconds of flight. After having received several images containing asphalt and buildings, the number of possible positions of the UAV is drastically reduced. Bottom row - Image from the on-board camera (left) and particle positions (right) after 20 seconds of flight. Once the UAV proceeds along one of the roads, the remaining faulty hypotheses can be discarded since they do not match the images obtained from the camera. The true vehicle position is shown as a black cross and the vehicle trajectory as a solid line.*

1.3 Contributions

The main contribution of this thesis is the extension, development and analysis of Rao-Blackwellised Monte Carlo methods for state inference and identification problems in nonlinear dynamical systems. The thesis is based on both published and unpublished material. Most notably in the category of the unpublished work, is the Rao-Blackwellised marginal particle filter (RBMPF), derived toward the end of Chapter 3 and applied to the identification problem in Chapter 6. A second identification method, referred to as the Rao-Blackwellised particle smoother expectation maximisation (RBPS-EM) method, is presented and evaluated in Chapter 6. This method has previously been published in,

> F. Lindsten and T. B. Schön. Identification of mixed linear/nonlinear state-space models. In *Proceedings of the 49th IEEE Conference on Decision and Control (CDC)*, Atlanta, USA, December 2010.

To orientate the reader, we already now point out the two main differences between these two identification methods.

1. The RBMPF is applicable for identification of general nonlinear systems with affine parameter dependence, whereas RBPS-EM can be used for identification of mixed linear/nonlinear systems with arbitrary parameter dependence.
2. The RBMPF is a Bayesian, recursive method, whereas RBPS-EM is a maximum likelihood based, batch approach.

The RBPS-EM method is based on a novel Rao-Blackwellised particle smoother (RBPS), for state inference in mixed linear/nonlinear systems. This smoother is derived in Chapter 5 and also presented in,

> F. Lindsten and T. B. Schön. Rao-Blackwellised particle smoothers for mixed linear/nonlinear state-space models. *Submitted to IEEE Transactions on Signal Processing*, 2011.

This thesis also contains an analysis of the benefits of applying Rao-Blackwellisation to sequential Monte Carlo methods. In particular, we provide an explicit expression for the variance reduction obtained in the Rao-Blackwellised particle filter (RBPF). This analysis is presented in Chapter 4, and has previously been published in,

> F. Lindsten, T. B. Schön, and J. Olsson. An explicit variance reduction expression for the Rao-Blackwellised particle filter. In *Proceedings of the 18th World Congress of the International Federation of Automatic Control (IFAC) (accepted for publication)*, Milan, Italy, August 2011b.

Loosely connected to the material of this thesis is,

> F. Lindsten, J. Callmer, H. Ohlsson, D. Törnqvist, T. B. Schön, and F. Gustafsson. Geo-referencing for UAV navigation using environmental classification. In *Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, USA, May 2010.

In this paper, the RBPF is applied to the problem of unmanned aerial vehicle localisation using measurements from a camera, an inertial measurement unit and a barometric sensor. This application example was briefly described in the previous section and is also reviewed in Section 3.3.5.

Other published material, not included in this thesis, is,

> F. Lindsten, H. Ohlsson, and L. Ljung. Clustering using sum-of-norms regularization; with application to particle filter output computation. In *Proceedings of the 2011 IEEE Workshop on Statistical Signal Processing (SSP) (accepted for publication)*, Nice, France, June 2011a.
>
> F. Lindsten, P.-J. Nordlund, and F. Gustafsson. Conflict detection metrics for aircraft sense and avoid systems. In *Proceedings of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SafeProcess)*, Barcelona, Spain, July 2009.

1.4 Thesis outline

This thesis is structured as follows; in Chapter 2 we define the model structures that we will be working with throughout the thesis, review the fundamental filtering and smoothing recursions and present some basic Monte Carlo techniques. In Chapter 3 the SMC framework is presented and discussed. We review the RBPF and derive the novel RBMPF. In Chapter 4 some asymptotic properties of SMC methods are discussed and we analyse the potential benefits from Rao-Blackwellising the PF. We then turn to the smoothing problem in Chapter 5, where we review some existing approaches to particle smoothing and derive a new RBPS. In Chapter 6 the RBMPF and the RBPS-EM identification methods are applied to the problem of nonlinear system identification. Finally, in Chapter 7 we draw conclusions and discuss directions of future work.

2 Prerequisites

This chapter presents some background material, which forms the foundation for the content of the later chapters of the thesis. After introducing some general notation in Section 2.1 we will take a closer look at state-space models in Section 2.2. In particular, we will introduce the class of conditionally linear Gaussian state-space (CLGSS) models, which will play an important role in this thesis. In Section 2.3, we will see that the filtering and smoothing recursions provide a general, conceptually simple solution to the state inference problem. However, these recursions are typically not analytically tractable, meaning that some approximative methods are required. The focus in this thesis is on Monte Carlo based approximation, which is why we review some basic concepts from sampling theory in Section 2.4.

2.1 Notation

Let us start by going through some of the notation that will be used throughout the thesis. If the notation introduced in the present section is unfamiliar or confusing, Appendix A provides a short introduction to measure and probability theory which might be enlightening. See also any of the standard texts on probability, e.g. the excellent book by Billingsley [1995]. This section is a complement to the list of notation given prior to Chapter 1, and consequently, all notation used in the thesis is not presented here.

When relevant, all random variables are defined on a common probability space $(\Omega, \mathcal{F}, \mathrm{P})$. For a measurable space $(\mathsf{X}, \mathcal{X})$, we denote by $\mathbb{F}(\mathsf{X})$ the set of all $\mathcal{X}/\mathcal{B}(\mathbb{R})$-measurable functions from X to $\mathbb{R}$. For a measure μ on $\mathcal{X}$ and $f \in \mathbb{F}(\mathsf{X})$, we denote by $\mu(f)$ the integral $\int f \, d\mu$ (assuming that the integral exists). Hence, if μ is a probability measure, $\mu(f)$ is the expectation of the function f under the distribution μ. For $p \geq 1$, $\mathsf{L}^p(\mathsf{X}, \mu)$ denotes the set of functions $f \in \mathbb{F}(\mathsf{X})$ such that $\int |f|^p \, d\mu < \infty$.

Let $(\mathsf{X}, \mathcal{X})$ and $(\mathsf{Y}, \mathcal{Y})$ be two measurable spaces. A kernel V from X to Y is a map $V : (\mathcal{Y}, \mathsf{X}) \to \mathbb{R}_+ \cup \{\infty\}$, written $V(A \mid x)$, such that

i) for each $x \in \mathsf{X}$, the map $A \mapsto V(A \mid x)$ is a measure on $\mathcal{Y}$.

ii) for each $A \in \mathcal{Y}$, the map $x \mapsto V(A \mid x)$ is $\mathcal{X}/\mathcal{B}(\mathbb{R})$-measurable.

A kernel is called a *transition kernel* if $V(\mathsf{Y} \mid x) = 1$ for any $x \in \mathsf{X}$. Hence, for a transition kernel V and a fixed $x \in \mathsf{X}$, $V(\,\cdot \mid x)$ is a probability measure on $\mathcal{Y}$. With $f : \mathsf{X} \times \mathsf{Y} \to \mathbb{R}$, $f(x, \,\cdot\,) \in \mathsf{L}^1(\mathsf{Y}, V(\,\cdot \mid x))$ we let $V(f)$ denote the function $V(f)[x] = \int f(x,y)V(dy \mid x)$.

For two measures ν and μ, we say that ν is absolutely continuous with respect to μ, written $\nu \ll \mu$, if $\mu(A) = 0 \Rightarrow \nu(A) = 0$. Measures will often be expressed as acting on "infinitesimal sets". For instance, if $(\mathsf{X}, \mathcal{X})$ is a measurable space, ν and μ are both measures on $\mathcal{X}$ and we wish to state that p is the density of ν w.r.t. to μ, we write

$$\nu(dx) = p(x)\mu(dx). \tag{2.1a}$$

The implicit meaning of this notation is that

$$\nu(A) = \int_A p(x)\mu(dx), \qquad \text{for all } A \in \mathcal{X}. \tag{2.1b}$$

This convention is used to make it easier to determine directly from the expression for a measure, on which σ-algebra the measure is defined.

By $\mathcal{N}(m, \Sigma)$ we denote the multivariate Gaussian distribution with mean m and covariance matrix Σ. We also write $\mathcal{N}(x; m, \Sigma)$ when referring to the probability density function (PDF) of this distribution. By $\mathrm{Cat}(\{p_i\}_{i=1}^N)$ we denote the categorical (discrete) distribution with probabilities $\{p_i\}_{i=1}^N$ such that $\sum_i p_i = 1$. Hence, if $X \sim \mathrm{Cat}(\{p_i\}_{i=1}^N)$ the range of X is $\{1, \ldots, N\}$ and $\mathrm{P}(X = i) = p_i$. A few additional standard distributions are defined in the list of notation given prior to Chapter 1.

2.2 State-space models

A state-space model (SSM) provides a convenient way of modeling a stochastic process. Let the state process $\{X_t\}_{t\geq 1}$ be a discrete-time stochastic process on the state-space $(\mathsf{X}, \mathcal{X})$ (typically some power of the real line with the corresponding Borel σ-algebra). Here, X_t represents the internal state of the system at time t, and holds all information about the system at this point in time. Hence, if we know what value X_t takes, past states or measurements hold no further information about the system at time t. This is reflected in X_t being Markovian, with transition kernel $Q(dx_{t+1} \mid x_t)$ and initial distribution $\nu(dx_1)$. However, the state process is typically not known, we say that it is hidden. Instead, we observe the system through the measurement process $\{Y_t\}_{t\geq 1}$, defined on the measurable space $(\mathsf{Y}, \mathcal{Y})$. Given $X_t = x_t$, the measurement Y_t is conditionally independent of past and future states and observations, and is distributed according to the kernel

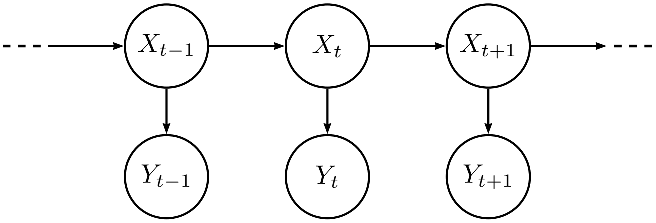

Figure 2.1: *Graphical model of an* SSM.

$G(dy_t \mid x_t)$. The SSM is thus given by,

$$X_1 \sim \nu(dx_1), \tag{2.2a}$$
$$X_{t+1} \mid \{X_t = x_t\} \sim Q(dx_{t+1} \mid x_t), \tag{2.2b}$$
$$Y_t \mid \{X_t = x_t\} \sim G(dy_t \mid x_t). \tag{2.2c}$$

A graphical model, illustrating the conditional dependencies in the SSM, is given in Figure 2.1. The model (2.2) is a fairly general representation of a dynamical system. No assumption on linearity and/or Gaussianity is made. However, as can be seen from the expressions above, the model is assumed to be time homogeneous, i.e. the kernels Q and G are not depending on t. However, this assumption is made merely to keep the notation uncluttered. In the sequel, the results presented can be seen as applicable also to time inhomogeneous models, allowing e.g. for the dependence on some known input sequence.

Alternatively, an equivalent functional representation of (2.2) may be used (with the same initial distribution ν for X_1),

$$X_{t+1} = f(X_t, V_t), \tag{2.3a}$$
$$Y_t = h(X_t, E_t). \tag{2.3b}$$

Here, the *process noise* V_t and the *measurement noise* E_t are mutually independent sequences of i.i.d. random variables.

In what follows, we will mostly be concerned with systems in which all random variables have densities w.r.t. some distributions. Hence, we make the following definitions.

Definition 2.1 (Partially dominated state-space model). The state-space model (2.2) is said to be partially dominated if there exists a probability measure μ on $\mathcal{Y}$ such that $G(\,\cdot \mid x_t) \ll \mu(\,\cdot\,)$ for all $x_t \in \mathsf{X}$. The density of $G(dy_t \mid x_t)$ w.r.t. μ will be denoted $p(y_t \mid x_t)$ and be referred to as the measurement density function.

Definition 2.2 (Fully dominated state-space model). The state-space model (2.2) is said to be fully dominated if, in addition to the conditions of Definition 2.1, there exists a probability measure λ on $\mathcal{X}$ such that $\nu \ll \lambda$ and $Q(\,\cdot \mid x_t) \ll \lambda(\,\cdot\,)$ for all $x_t \in \mathsf{X}$. The density of $Q(dx_{t+1} \mid x_t)$ w.r.t. λ will be denoted $p(x_{t+1} \mid x_t)$ and be referred to as the transition density function.

Remark 2.1 (A notational convention). In Definition 2.2 the same symbol p has been deliberately used to represent both the measurement density function and the transition density function. Which one of the two densities that is referred to is solely indicated by the arguments of the function. This abuse of notation is common, e.g. in statistical signal processing and automatic control, and shall be adopted in this thesis as well. Furthermore, the model (2.2) implicitly defines the distribution of any finite collection of variables from the processes X_t and Y_t, as well as marginals and conditionals thereof. If the model is fully dominated, these distribution will also have densities (w.r.t. some product of μ and λ), and all such densities will be denoted p (again, letting the arguments indicate which density that is referred to). This notational convention will be exemplified in Section 2.3 and is widely applied in the remaining of this thesis. Finally, in case the model is fully dominated, we will also write dx instead of $\lambda(dx)$ whenever integrating w.r.t. λ.

Remark 2.2. In many practical applications it is common to have $(\mathsf{X}, \mathcal{X}) = (\mathbb{R}^{n_x}, \mathcal{B}(\mathbb{R}^{n_x}))$ and $(\mathsf{Y}, \mathcal{Y}) = (\mathbb{R}^{n_y}, \mathcal{B}(\mathbb{R}^{n_y}))$ for some integers n_x and n_y (the state and measurements dimensions, respectively). Also, if the random variables of the model are continuous, both μ and λ can often be taken as Lebesgue measure, further motivating the convention to replace $\lambda(dx)$ with dx.

2.2.1 Linear Gaussian state-space models

A *time inhomogeneous* linear Gaussian state-space (LGSS) model is given by,

$$X_{t+1} = A_t X_t + b_t + V_t, \qquad V_t \sim \mathcal{N}(0, Q_t), \tag{2.4a}$$

$$Y_t = C_t X_t + d_t + E_t, \qquad E_t \sim \mathcal{N}(0, R_t), \tag{2.4b}$$

where the process noise V_t and the measurement noise E_t are sequences of independent Gaussian random variables. Here, A_t and C_t are sequences of matrices with appropriate dimensions, Q_t and R_t are sequences of covariance matrices and b_t and d_t are sequences of known vectors, e.g. originating from an input signal exciting the system.

Strictly speaking, (2.4) is *not* a special case of (2.2), since the latter is time homogeneous and the former is not (the kernels Q and G in (2.2) are not t-dependent). Of course, a time homogeneous special case of (2.4) is obtained if all known quantities mentioned above (A_t, C_t, etc.) are constant w.r.t. t. However, for reasons that will become clear in the next section, we choose this (more general) definition for an LGSS model.

LGSS models are without doubt the most important and most well studied class of SSMs. There are basically two reasons for this. First, LGSS models provide sufficiently accurate descriptions of many interesting dynamical systems. Second, the class of LGSS models is one of the few model classes, simple enough to allow for an analytical treatment.

Example 2.1: Partially or fully dominated SSM

Now that we have defined the class of LGSS models, let us exemplify the difference between partially and fully dominated SSMs, as defined in Definition 2.1 and Definition 2.2, respectively. Consider the time homogeneous LGSS model,

$$X_{t+1} = A X_t + V_t, \qquad V_t \sim \mathcal{N}(0, Q), \tag{2.5}$$

$$Y_t = C X_t + E_t, \qquad E_t \sim \mathcal{N}(0, R), \tag{2.6}$$

with state-space $(\mathbb{R}^{n_x}, \mathcal{B}(\mathbb{R}^{n_x}))$ and observation space $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$. Then, the measurement kernel $G(dy_t \mid x_t)$ is Gaussian (for all $x_t \in \mathbb{R}^{n_x}$) and dominated by Lebesgue measure. Hence, the model is partially dominated.

If the process noise covariance Q is full rank, then the transition kernel $Q(dx_{t+1} \mid x_t)$ is also Gaussian (for all $x_t \in \mathbb{R}^{n_x}$) and dominated by Lebesgue measure. In this case, the model is fully dominated. However, if the process noise covariance is rank deficient, then V_t has no density function (w.r.t. Lebesgue measure) and the model is *not* fully dominated. To have a rank deficient process noise covariance is common in many applications, for instance if there is a physical connection between some of the states (such as between position and velocity) or if the model is single input, single output with input and output noises.

There are many other examples of non-fully dominated SSMs, e.g. if the state-space is continuous but there is a nonzero probability that the state "jumps" to some specific point.

2.2.2 Conditionally linear Gaussian state-space models

A conditionally linear Gaussian state-space (CLGSS) model is defined as below.

Definition 2.3 (CLGSS model). Let X_t, the state process in an SSM, be partitioned according to $X_t = \{\Xi_t, Z_t\}$ and $\mathsf{X} = \mathsf{X}_\xi \times \mathsf{X}_z$. The SSM is a CLGSS model if the conditional process $\{Z_t \mid \Xi_{1:t}\}_{t\geq 1}$ is a time inhomogeneous LGSS model.

The reason for this, rather abstract definition is that there are many different functional forms (see the examples below), that all share the same fundamental property; conditioned on one part of the state, the remaining part behaves as an LGSS model. Since this is the property that we wish to exploit when constructing algorithms for this type of models, it is better to make the definition as general as possible, leading to algorithms that are more widely applicable. Since the Z-process is conditionally linear, we shall call Z_t the *linear* state whereas Ξ_t will be called the *nonlinear* state.

Remark 2.3. Note that, for most CLGSS models of interest, it is necessary to condition on the entire nonlinear trajectory $\Xi_{1:t}$ for the conditional process to be linear, i.e. to condition on just Ξ_t is not sufficient. We comment further on this in Example 2.2 below.

To explicate what kind of models that are of CLGSS type, we give two examples of such models below.

Example 2.2: Hierarchical CLGSS model

One of the most commonly seen examples of a CLGSS model is what we here denote a hierarchical CLGSS model. The name refers to the relationship between the variables Ξ_t and Z_t, where Ξ_t is at the top of the hierarchy, evolving according to a Markov kernel $Q^\xi(d\xi_{t+1} \mid \xi_t)$ independently of Z_t. The conditional dependencies of the model are illustrated in Figure 2.2. The linear state Z_t obeys an LGSS model, parameterised by Ξ_t, i.e. the model is given by,

$$\Xi_{t+1} \sim Q^\xi(d\xi_{t+1} \mid \Xi_t), \tag{2.7a}$$

$$Z_{t+1} = f(\Xi_t) + A(\Xi_t)Z_t + V_t, \tag{2.7b}$$

$$Y_t = h(\Xi_t) + C(\Xi_t)Z_t + E_t, \tag{2.7c}$$

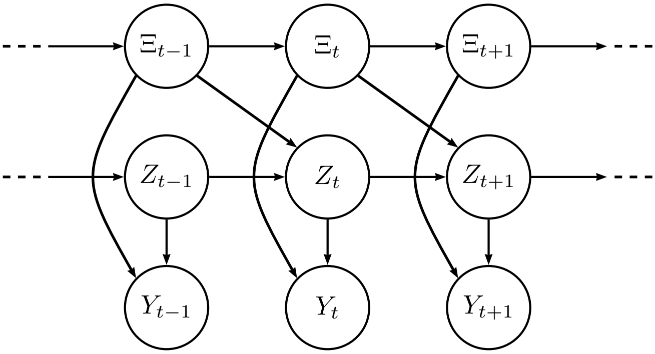

Figure 2.2: *Graphical model of a hierarchical CLGSS model.*

with Gaussian noise sources V_t and E_t. The initial distribution of the process if defined by $\Xi_1 \sim \nu^{\xi}(d\xi_1)$ and

$$Z_1 \mid \Xi_1 \sim \mathcal{N}(\bar{z}_{1|0}(\Xi_1), P_{1|0}(\Xi_1)). \tag{2.8}$$

Here we have assumed that the matrices A, C etc. are functions of the state Ξ_t, but otherwise independent of time. That is, (2.7) is a time homogeneous SSM (with state $X_t = \{\Xi_t, Z_t\}$). However, for any fixed time $t \geq 1$ and conditioned on $\Xi_{1:t}$, the sequences $\{A(\Xi_k)\}_{k=1}^t$, $\{C(\Xi_k)\}_{k=1}^t$, etc. are known. That is, conditioned on $\Xi_{1:t}$, (2.7b,c) describe a time inhomogeneous LGSS model with state Z_t.

As previously pointed out, to condition on just Ξ_t is not sufficient. In that case, the sequence $\{A(\Xi_k)\}_{k=1}^t$ (for instance) would consist of random elements, where only the final element is known.

We continue with another CLGSS example, which will play a more central role in this thesis.

Example 2.3: Mixed linear/nonlinear Gaussian state-space model

A mixed linear/nonlinear Gaussian state-space model can be expressed on functional form, according to

$$\Xi_{t+1} = f^{\xi}(\Xi_t) + A^{\xi}(\Xi_t)Z_t + V_t^{\xi}, \tag{2.9a}$$
$$Z_{t+1} = f^{z}(\Xi_t) + A^{z}(\Xi_t)Z_t + V_t^{z}, \tag{2.9b}$$
$$Y_t = h(\Xi_t) + C(\Xi_t)Z_t + E_t, \tag{2.9c}$$

where the process noise $V_t \triangleq \begin{bmatrix}(V_t^{\xi})^{\mathsf{T}} & (V_t^{z})^{\mathsf{T}}\end{bmatrix}^{\mathsf{T}}$ and the measurement noise E_t are mutually independent, white and Gaussian according to,

$$V_t \sim \mathcal{N}\left(0, Q(\Xi_t)\right), \tag{2.10a}$$
$$E_t \sim \mathcal{N}\left(0, R(\Xi_t)\right), \tag{2.10b}$$

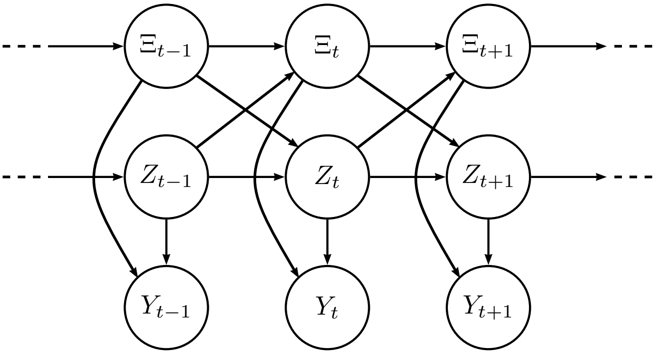

Figure 2.3: *Graphical model of a mixed linear/nonlinear Gaussian state-space model.*

with

$$Q(\Xi_t) = \begin{bmatrix} Q^{\xi}(\Xi_t) & Q^{\xi z}(\Xi_t) \\ Q^{\xi z}(\Xi_t)^{\mathsf{T}} & Q^{z}(\Xi_t) \end{bmatrix}. \tag{2.10c}$$

The initial distribution of the process if defined by $\Xi_1 \sim \nu^{\xi}(d\xi_1)$ and

$$Z_1 \mid \Xi_1 \sim \mathcal{N}(\bar{z}_{1|0}(\Xi_1), P_{1|0}(\Xi_1)). \tag{2.11}$$

For this type of model (as opposed to hierarchical CLGSS models), the evolution of the nonlinear state (2.9a) depends on the Z-process and can not simply be neglected when considering the conditional process. In fact, conditioned on $\Xi_{1:t}$ the relationship (2.9a) holds information about the Z-process and can be seen as an extra measurement. The conditional dependencies of the model are illustrated in Figure 2.3.

We will in the sequel make frequent use of a more compact reformulation of (2.9) according to

$$X_{t+1} = f(\Xi_t) + A(\Xi_t)Z_t + V_t, \tag{2.12a}$$

$$Y_t = h(\Xi_t) + C(\Xi_t)Z_t + E_t, \tag{2.12b}$$

with

$$X_t = \begin{bmatrix} \Xi_t \\ Z_t \end{bmatrix}, \qquad f(\Xi_t) = \begin{bmatrix} f^{\xi}(\Xi_t) \\ f^{z}(\Xi_t) \end{bmatrix}, \qquad A(\Xi_t) = \begin{bmatrix} A^{\xi}(\Xi_t) \\ A^{z}(\Xi_t) \end{bmatrix}. \tag{2.12c}$$

2.3 Filtering and smoothing recursions

This section will treat the fundamental problem of state inference, mentioned in Chapter 1, using the SSM setting. The material of this section is to a large extent influenced by Cappé et al. [2005], who provide a much more in-depth treatment of the subject. Assume that a partially (or fully) dominated SSM is given, according to Definition 2.1 (or Definition 2.2).

Table 2.1: Filtering, smoothing and predictive distributions and densities

	Distribution	Density
Filtering	$\Phi_{t\mid t}(dx_t)$	$p(x_t \mid y_{1:t})$
Joint smoothing	$\Phi_{1:T\mid T}(dx_{1:T})$	$p(x_{1:T} \mid y_{1:T})$
Marginal smoothing ($t \leq T$)	$\Phi_{t\mid T}(dx_t)$	$p(x_t \mid y_{1:T})$
Fixed-interval smoothing ($s < t \leq T$)	$\Phi_{s:t\mid T}(dx_{s:t})$	$p(x_{s:t} \mid y_{1:T})$
Fixed-lag smoothing (ℓ fixed)	$\Phi_{t-\ell+1:t\mid t}(dx_{t-\ell+1:t})$	$p(x_{t-\ell+1:t} \mid y_{1:t})$
k-step prediction	$\Phi_{t+k\mid t}(dx_{t+k})$	$p(x_{t+k} \mid y_{1:t})$

Since the state X_t holds all information about the system at time t, it is natural to ask what is known about the state process, given a sequence of observations $Y_{1:T} = y_{1:T}$. More precisely, what is the distribution of some state sequence $X_{s:t}$ conditioned on the measurements $Y_{1:T}$? Before we answer this question, let us make the following definition.

Definition 2.4 (Likelihood function). The likelihood function $p(y_{1:T})$ is the PDF of the measurement sequence $Y_{1:T}$.

In terms of the quantities of the model (2.2), the likelihood function can be expressed as,

$$p(y_{1:T}) = \int \cdots \int \nu(dx_1) \prod_{i=1}^{T} p(y_i \mid x_i) \prod_{i=1}^{T-1} Q(dx_{i+1} \mid x_i). \tag{2.13}$$

In the sequel, it will be implicit that all results hold only for $y_{1:T}$ in the set of probability one, for which $p(y_{1:T})$ is strictly positive.

Let us now define a family of conditional probability distributions by

$$\Phi_{s:t\mid T}(A) = \mathrm{P}(X_{s:t} \in A \mid Y_{1:T} = y_{1:T}), \tag{2.14}$$

for $A \in \mathcal{X}^{(t-s+1)}$ and for some indices s, t, $T \in \mathbb{N}$, $s \leq t$. For ease of notation, when $s = t$ we use $\Phi_{t\mid T}$ as shorthand for $\Phi_{t:t\mid T}$. There are a few special cases that are of particular interest, summarised in Table 2.1. Apart from the distributions according to (2.14), the table also provides their densities using the notational convention of Remark 2.1 on page 14. Hence, the rightmost column of the table should only be seen as valid for fully dominated models.

Remark 2.4 (Readers uncomfortable with the construction (2.14) *are encouraged to read this remark).* To define a measure according to (2.14) is not really rigorous and conditional probability is a more complicated business than what is indicated by (2.14). Basically, the problem arises since we wish to condition on the set $\{\omega : Y_{1:T}(\omega) = y_{1:T}\}$ which in many cases has probability zero. We should rather have written that $\Phi_{s:t\mid T}(dx_{s:t} \mid y_{1:T})$ is a kernel from Y^T to $\mathsf{X}^{(t-s+1)}$ such that $\Phi_{s:t\mid T}(A \mid Y_{1:T})$ is a version of the conditional probability $\mathrm{P}(X_{s:t} \in A \mid Y_{1:T})$. In (2.14), the dependence on $y_{1:T}$ on the left hand side is implicit. If one does not like the definition (2.14), the expression (2.15) can alternatively be taken as the *definition* of the Φ-family of distributions. However, it is still instructive to think of $\Phi_{s:t\mid T}$ as the distribution of $X_{s:t}$ given a fixed sequence of measurements $y_{1:T}$, and the interpretation is the same regardless of how we choose to define it.

By simple substitution or from Bayes' rule, the following result is easily obtained.

Proposition 2.1. *Let a partially (or fully) dominated* SSM *be given, with initial distribution ν, transition kernel Q and measurement density function $p(y_t \mid x_t)$. Let the likelihood function be given by* (2.13)*. Then, for any $T \geq 1$ and $k \geq 0$ and for $A \in \mathcal{X}^{(T+k)}$,*

$$\Phi_{1:T+k|T}(A) = \frac{1}{p(y_{1:T})} \int_A \nu(dx_1) \prod_{i=1}^{T} p(y_i \mid x_i) \prod_{i=1}^{T+k-1} Q(dx_{i+1} \mid x_i). \tag{2.15}$$

Proof: See e.g. Cappé et al. [2005], Proposition 3.1.4. □

By marginalisation, (2.15) provides an explicit expression for any member of the family (2.14). Hence, the state inference problem might at first glance appear to be of simple nature, since its solution is provided by (2.15). There are however two reasons for why this is not the case.

1. To be able to evaluate the distributions of Table 2.1 in a systematic manner, e.g. in a computer program, we often seek more structured expressions than what is provided by (2.15). Typically, this means to express the distribution of interest recursively. This is required also if the distribution is to be evaluated online.
2. The (multidimensional) integral appearing in (2.15) is in most cases not analytically tractable, and some approximate method of evaluation is required.

In the remainder of this section, the first of these problems will be addressed. Recursive expressions for the filtering distribution and the joint smoothing distribution will be provided. The k-step predictive distribution can straightforwardly be obtained from the filtering distribution, and will be paid no further attention. See also the discussion in Section 2.3.1. The second, more intricate problem of how to evaluate the intractable integrals will be discussed in Chapter 3 and in Chapter 5.

2.3.1 Forward recursions

Let us start by considering the filtering distribution. By marginalisation of (2.15) we have, for any $t \geq 1$,

$$\Phi_{t|t}(dx_t) = \frac{1}{p(y_{1:t})} \int_{\mathsf{X}^{t-1}} \nu(dx_1) \prod_{i=1}^{t} p(y_i \mid x_i) \prod_{i=1}^{t-1} Q(dx_{i+1} \mid x_i). \tag{2.16}$$

Here we have used the convention that $\prod_{i=1}^{0}(\,\cdot\,) = 1$.

Remark 2.5. When analysing expressions such as (2.16), some "pattern matching" is required to determine the meaning of the integral on the right hand side. We see that the filtering distribution on the left hand side is a measure on $\mathcal{X}$, indicated by the "dx_t". This means that "dx_t" should be a residue on the right hand side as well, i.e. the integral is with respect to $dx_1 \dots dx_{t-1}$ (hence over the set X^{t-1}), but dx_t is left untouched.

Furthermore, the 1-step predictive distribution (at time $t-1$) is given by,

$$\Phi_{t|t-1}(dx_t) = \frac{1}{p(y_{1:t-1})} \int\limits_{\mathsf{X}^{t-1}} \nu(dx_1) \prod_{i=1}^{t-1} p(y_i \mid x_i) \prod_{i=1}^{t-1} Q(dx_{i+1} \mid x_i). \tag{2.17}$$

Hence, by combining the above results we get the following, two-step recursion for the filtering distribution,

$$\Phi_{t|t}(dx_t) = \frac{p(y_t \mid x_t)\Phi_{t|t-1}(dx_t)}{p(y_t \mid y_{1:t-1})}, \tag{2.18a}$$

$$\Phi_{t+1|t}(dx_{t+1}) = \int\limits_{\mathsf{X}} Q(dx_{t+1} \mid x_t)\Phi_{t|t}(dx_t), \tag{2.18b}$$

where we have made use of the relation $p(y_t \mid y_{1:t-1}) = p(y_{1:t})p(y_{1:t-1})^{-1}$ and adopted the convention $\Phi_{1|0}(dx_1) = \nu(dx_1)$. As can be seen from (2.18b), the 1-step predictive distribution is given as a byproduct in the filtering recursion. Also, the k-step predictive distribution can easily be obtained from the filtering distribution similarly to (2.18b), by applying the kernel Q iteratively k times.

The recursion (2.18) is known as the Bayesian filtering recursion. Step (2.18a) is often called the *measurement update*, since the “current” measurement y_t is taken into account. Step (2.18b) is known as the *time update*, moving the distribution forward in time, from t to $t+1$.

Assuming that the model is fully dominated, we can express the recursion in terms of densities instead, leading to,

$$p(x_t \mid y_{1:t}) = \frac{p(y_t \mid x_t)p(x_t \mid y_{1:t-1})}{p(y_t \mid y_{1:t-1})}, \tag{2.19a}$$

$$p(x_{t+1} \mid y_{1:t}) = \int p(x_{t+1} \mid x_t)p(x_t \mid y_{1:t})\, dx_t. \tag{2.19b}$$

By a similar procedure we can obtain a recursion for the joint smoothing distribution,

$$\Phi_{1:t|t}(dx_{1:t}) = \frac{p(y_t \mid x_t)\Phi_{1:t|t-1}(dx_{1:t})}{p(y_t \mid y_{1:t-1})}, \tag{2.20a}$$

$$\Phi_{1:t+1|t}(dx_{1:t+1}) = Q(dx_{t+1} \mid x_t)\Phi_{1:t|t}(dx_{1:t}), \tag{2.20b}$$

or in terms of densities,

$$p(x_{1:t} \mid y_{1:t}) = \frac{p(y_t \mid x_t)p(x_{1:t} \mid y_{1:t-1})}{p(y_t \mid y_{1:t-1})}, \tag{2.21a}$$

$$p(x_{1:t+1} \mid y_{1:t}) = p(x_{t+1} \mid x_t)p(x_{1:t} \mid y_{1:t}). \tag{2.21b}$$

The above recursion for the joint smoothing distribution will be denoted the forward recursion, since it propagates forward in time (increasing indices t). As we shall see in the coming section, it is also possible to find a time-reversed recursion for this distribution.

2.3.2 Backward recursions

Assume that we have made observations of the measurement sequence up to some "final" time T, i.e. we have observed $Y_{1:T} = y_{1:T}$. Conditioned on these measurements, the state process $\{X_t\}_{t=1}^T$ is an inhomogeneous Markov process. Under some weak assumptions (see Cappé et al. [2005], Section 3.3.2 for details), the same holds true for the time-reversed chain, starting at time T and evolving backward in time according to the so called backward kernel,

$$B_t(A \mid x_{t+1}) \triangleq \mathrm{P}(x_t \in A \mid X_{t+1} = x_{t+1}, Y_{1:T} = y_{1:T}), \tag{2.22a}$$

for $A \in \mathcal{X}$. Note that the backward kernel is time inhomogeneous, hence the dependence on t in the notation. It is not always possible to give an explicit expression for the backward kernel. However, for a fully dominated model, this can always be done, and its density is given by

$$p(x_t \mid x_{t+1}, y_{1:T}) = \frac{p(x_{t+1} \mid x_t) p(x_t \mid y_{1:t})}{\int p(x_{t+1} \mid x_t) p(x_t \mid y_{1:t}) \, dx_t}. \tag{2.22b}$$

It also holds true that $p(x_t \mid x_{t+1}, y_{1:T}) = p(x_t \mid x_{t+1}, y_{1:t})$. This fact is related to the conditional independence properties of the SSM; if we know the state at time $t + 1$, there is no further information available in the measurements $y_{t+1:T}$.

Using the backward kernel, the joint smoothing distribution can be shown to obey the following backward recursion (see e.g. [Douc et al., 2010]),

$$\Phi_{t:T|T}(dx_{t:T}) = B_t(dx_t \mid x_{t+1}) \Phi_{t+1:T|T}(dx_{t+1:T}), \tag{2.23}$$

starting with the filtering distribution $\Phi_{T|T}$ at time T. When the recursion is "complete", i.e. at $t = 1$, the joint smoothing distribution for the time interval $1, \ldots, T$ is obtained. By marginalisation of (2.23), we also obtain a recursion for the marginal smoothing distribution,

$$\Phi_{t|T}(dx_t) = \int_{\mathsf{X}} B_t(dx_t \mid x_{t+1}) \Phi_{t+1|T}(dx_{t+1}) \tag{2.24}$$

and based on this, an expression for the fixed-interval smoothing distribution,

$$\Phi_{s:t|T}(dx_{s:t}) = B_s(dx_s \mid x_{s+1}) \cdots B_{t-1}(dx_{t-1} \mid x_t) \Phi_{t|T}(dx_t). \tag{2.25}$$

The backward kernel at time t depends on the filtering distribution $\Phi_{t|t}$. This is most clearly visible in the explicit expression for its density (2.22b), where the filtering density $p(x_t \mid y_{1:t})$ appears. Hence, to utilise the backward recursion (2.23) for smoothing, the filtering distributions must first be computed for $t = 1, \ldots, T$. Consequently, this procedure is generally called *forward filtering/backward smoothing*.

Remark 2.6. As a final remark of this section, it is worth to mention that other types of recursions can be used to obtain the same, or related distributions as treated above. Most notable are the *two-filter* recursion and the *backward filtering/forward smoothing* recursion, as alternatives to (2.23) (see e.g. [Cappé et al., 2005, Chapter 3]). The reason for why (2.23) is the only recursion presented here, is that it will be used in the smoothing algorithms that will be discussed in Chapter 5, which are of the type forward filtering/backward smoothing.

2.4 Sampling theory

In the coming chapters, a few concepts from sampling theory will be frequently used. These include importance sampling (IS), sampling importance resampling (SIR) and rejection sampling (RS). For readers unfamiliar with these concepts, they will be reviewed in this section. We will not make a formal treatment of the theory. Instead, the current section aims at providing an intuition for the mechanisms underlying the different methods.

For the purpose of illustration, let μ be a probability distribution. We are interested in evaluating the expectation of some integrable function f under this distribution,

$$\mu(f) = \int f(x)\mu(dx). \tag{2.26}$$

Now, if the above integral is intractable, it can be approximated using Monte Carlo (MC) integration. This is done by sampling a sequence $\{X_i\}_{i=1}^N$ of i.i.d. random variables distributed according to μ, and computing the approximation

$$\mu(f) \approx \frac{1}{N}\sum_{i=1}^{N} f(X_i). \tag{2.27}$$

By the strong law of large numbers, this approximation converges almost surely to the true expectation as N tends to infinity.

It is convenient to introduce a distribution μ_{MC}^N, as an approximation of μ, based on the samples X_i as

$$\mu_{\text{MC}}^N = \frac{1}{N}\sum_{i=1}^{N} \delta_{X_i}, \tag{2.28}$$

where δ_x is a point-mass located at x. The approximation (2.27) is then given by $\mu_{\text{MC}}^N(f)$. Note that (2.28) is a random probability distribution.

2.4.1 Importance sampling

The problem that one often faces is that it is hard to sample from the desired distribution μ, which from now on will be denoted the target distribution. IS a way to circumvent this problem.

Let η be a probability distribution (called the proposal) such that $\mu \ll \eta$. Besides from this constraint, the distribution can be chosen arbitrarily. We then have

$$\mu(f) = \int f(x)\mu(dx) = \int f(x)\frac{d\mu}{d\eta}(x)\eta(dx) = \eta\left(f \cdot \frac{d\mu}{d\eta}\right). \tag{2.29}$$

If the proposal distribution η is chosen so that it easily can be sampled from, we can approximate $\mu(f)$ by

$$\mu(f) \approx \frac{1}{N}\sum_{i=1}^{N} \frac{d\mu}{d\eta}(Z_i)f(Z_i), \tag{2.30a}$$

where $\{Z_i\}_{i=1}^N$ is a sequence of i.i.d. random variables distributed according to η. We see that this leads to the same type of approximation as in (2.27), but the samples are weighted with the quantities,

$$\widetilde{W}_i \triangleq \frac{1}{N}\frac{d\mu}{d\eta}(Z_i), \tag{2.30b}$$

known as importance weights. This corrects for the errors introduced by sampling from the wrong distribution. The quality of the approximation (2.30) will be affected by the mismatch between the proposal and the target distributions. To get good performance from the IS method (for arbitrary, integrable test functions f), it is important that the proposal resembles the target as closely as possible.

It is often the case that the Radon-Nikodym derivative $d\mu/d\eta(x)$ can be evaluated only up to proportionality (see e.g. the application of IS for sequential Monte Carlo (SMC) in Chapter 3). Thus, assume that,

$$\frac{d\mu}{d\eta}(x) = \frac{1}{C}\kappa(x), \tag{2.31}$$

where $\kappa(x)$ can be evaluated, but C is an unknown constant. To estimate this constant, the same set of samples $\{Z_i\}_{i=1}^N$ can be used. Since

$$\int \frac{d\mu}{d\eta}(x)\eta(dx) = 1, \tag{2.32a}$$

it follows that

$$C = \int \kappa(x)\eta(dx) \approx \frac{1}{N}\sum_{i=1}^{N}\kappa(Z_i) = \sum_{i=1}^{N} W_i', \tag{2.32b}$$

where we have introduced the unnormalized importance weights

$$W_i' \triangleq \frac{1}{N}\kappa(Z_i) \propto \frac{d\mu}{d\eta}(Z_i). \tag{2.32c}$$

An approximation of the (normalised) importance weights (2.30b) is then given by

$$\widetilde{W}_i = \frac{1}{NC}\kappa(Z_i) \approx W_i \triangleq \frac{W_i'}{\sum_k W_k'}. \tag{2.33}$$

Similarly to (2.28) we can construct a point-mass distribution μ_{IS}^N, approximating μ, as

$$\mu_{\text{IS}}^N = \sum_{i=1}^{N} W_i \delta_{Z_i}. \tag{2.34}$$

The approximation (2.30) together with (2.33) is then attained as $\mu_{\text{IS}}^N(f)$. Note that, even if the constant C would be known, the weight normalisation is required for (2.34) to be interpretable as a probability distribution. We shall refer to the collection $\{Z_i, W_i\}_{i=1}^N$ as a *weighted particle system* (implicitly, with nonnegative weights W_i that sum to one, see Definition 3.1 in the next chapter). Such a system uniquely defines a probability measure according to (2.34), and we say that the system *targets* the distribution μ. The importance sampling method is summarised in Algorithm 2.1.

Algorithm 2.1 Importance sampling

Input: A target distribution μ and a proposal distribution η, s.t. $\mu \ll \eta$.
Output: A weighted particle system $\{Z_i, W_i\}_{i=1}^N$ targeting μ.

```
1: Draw $N$ i.i.d. samples $\{Z_i\}_{i=1}^N$ from the proposal.
2: Compute the unnormalised importance weights,
       $W_i' \propto \frac{d\mu}{d\eta}(Z_i), \qquad i = 1, \ldots, N.$
3: Normalise the weights,
       $W_i = \frac{W_i'}{\sum_k W_k'}, \qquad i = 1, \ldots, N.$
```

2.4.2 Sampling importance resampling

As pointed out in the previous section, the IS scheme will result in a weighted particle system $\{Z_i, W_i\}_{i=1}^N$, targeting some distribution μ. If we for some reason seek an unweighted sample, targeting the same distribution (this is for instance important in the SMC methods discussed Chapter 3), we can employ SIR.

The idea is very simple. As previously pointed out, the reason for resorting to IS (or SIR) is that we cannot straightforwardly sample from the target distribution μ directly. However, since (2.34) provides an approximation of μ, we can draw M new, i.i.d. samples from this distribution,

$$\bar{Z}_j \sim \mu_{\text{IS}}^N, \qquad j = 1, \ldots, M. \tag{2.35}$$

Since μ_{IS}^N has finite support, sampling from it is straightforward. We set $\bar{Z}_j = Z_i$ with probability W_i, i.e.

$$\mathrm{P}\left(\bar{Z}_j = Z_i \bigm| \{Z_k, W_k\}_{k=1}^N\right) = W_i, \qquad j = 1, \ldots, M. \tag{2.36}$$

This results in an equally weighted particle system $\{\bar{Z}_j, 1/M\}_{j=1}^M$ targeting the same distribution μ. The particle system defines a point-mass distribution μ_{SIR}^M approximating μ, analogously to (2.28) (with X replaced by $\bar{Z}$).

The procedure which (randomly) turns a weighted particle system into an unweighted one, is called *resampling*. The method defined by (2.36) is known as multinomial resampling, and it is the simplest and most intuitive method. However, there are other types of resampling methods that are preferable, since they introduce less variance in the approximation $\mu_{\text{SIR}}^M(f)$. We will return to this in Section 3.1.2.

2.4.3 Rejection sampling

An alternative to IS and SIR is RS. This is a sampling method which in fact generate i.i.d. samples from the target distribution μ. The main drawback with RS is that it can be computationally demanding, and there is no upper bound on the execution time required to generate a sample of fixed size. We return to these issues at the end of this section.

Algorithm 2.2 Rejection sampling

Input: A target distribution μ and a proposal distribution η, s.t. $\mu \ll \eta$, $d\mu/d\eta(x) \propto \kappa(x) < \rho$.

Output: N i.i.d. samples $\{Z_i\}_{i=1}^N$ from the target distribution.

```
L ← {1, ..., N}.
while L is not empty do
    n ← card(L).
    δ ← ∅.
    Sample independently $\{Z'_k\}_{k=1}^n \sim \eta$.
    Sample independently $\{U_k\}_{k=1}^n \sim \mathcal{U}([0,\, 1])$.
    for k = 1 to n do
        if $U_k \leq \kappa(Z'_k)/\rho$ then
            $Z_{L(k)} \leftarrow Z'_k$.
            $\delta \leftarrow \delta \cup \{L(k)\}$.
        end if
    end for
    $L \leftarrow L \setminus \delta$.
end while
```

Analogously to the IS procedure, we choose a proposal distribution η, such that $\mu \ll \eta$. As in (2.31) we assume that the Radon-Nikodym derivative $d\mu/d\eta(x)$ can be evaluated, at least up to proportionality, i.e.

$$\frac{d\mu}{d\eta}(x) = \frac{1}{C}\kappa(x), \tag{2.37}$$

where κ can be evaluated and C is a (possibly) unknown constant. Furthermore, we shall assume that the function κ is bounded from above by some known constant $\rho < \infty$. The RS procedure is then as follows. First, we draw a sample Z' from the proposal distribution, and sample a random variable U uniformly over the unit interval, i.e.

$$Z' \sim \eta, \tag{2.38a}$$

$$U \sim \mathcal{U}([0,\, 1]). \tag{2.38b}$$

The variable U serves as an indicator on whether we should accept Z' as a valid sample from the target distribution μ or not. More precisely, we accept the sample and set $Z := Z'$ if $U \leq \kappa(Z')/\rho$. If this is not the case, we reject the sample Z' and repeat the procedure (2.38) until a sample is accepted. The RS method is summarised in Algorithm 2.2, in which N samples are generated in parallel.

The procedure outlined above does indeed produces a sample Z, distributed according to the target distribution μ. To see this, let $\mathcal{X}$ be the σ-algebra on which the distributions μ and η are defined. Take $A \in \mathcal{X}$ and consider,

$$\mathrm{P}(Z \in A) = \mathrm{P}(Z' \in A \mid U \leq \kappa(Z')/\rho) = \frac{\mathrm{P}(Z' \in A \cap U \leq \kappa(Z')/\rho)}{\mathrm{P}(U \leq \kappa(Z')/\rho)}. \tag{2.39}$$

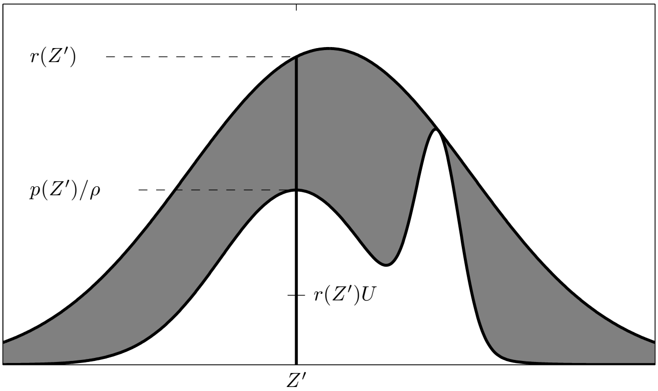

Figure 2.4: *Illustration of* RS*. A sample Z' is proposed from the density $r(x)$. The sample is accepted if $r(Z')U \leq p(Z')/\rho$.*

Since Z' is distributed according to η, the numerator in the expression above is given by,

$$\int_A \mathrm{P}(U \leq \kappa(x)/\rho)\eta(dx) = \int_A \frac{\kappa(x)}{\rho}\eta(dx) = \frac{C}{\rho}\mu(A), \tag{2.40}$$

where we have made use of the relation (2.37) and the fact that U is distributed according to (2.38b). The denominator in (2.39) can, by similar calculations, be show to equal C/ρ. Hence, we get

$$\mathrm{P}(Z \in A) = \mu(A), \tag{2.41}$$

which confirms that Z has the desired distribution. The RS approach is illustrated in Example 2.4, which also provides an intuitive explanation of the mechanism.

Example 2.4: Rejection sampling

Assume that we wish to sample from the GMM with density,

$$p(x) = \frac{3}{4}\mathcal{N}(x; -0.3, 0.7) + \frac{1}{4}\mathcal{N}(x; 1, 0.2), \tag{2.42}$$

using RS. We choose a Gaussian proposal distribution with density $r(x) = \mathcal{N}(x; 0, 1.3)$ and let ρ be a constant such that $p(x)/r(x) < \rho$. A sample Z' is draw from the proposal distribution and U is drawn uniformly from the unit interval, as in (2.38). Then, the pair $\{Z', r(Z')U\}$ can be seen as uniformly distributed over the grey area in Figure 2.4, bounded by the PDF $r(x)$. However, we only wish to accept the sample if it appears as if being a sample from the target distribution, i.e. if it falls within the white area bounded by $p(x)/\rho$. That is, we accept Z' as a sample from $p(x)$ if $U \leq p(Z')/(r(Z')\rho)$. The acceptance probability is the ratio between the areas under the curves in Figure 2.4, but since both densities are normalised, this ratio is simply $1/\rho$.

As previously mentioned, the main problem with RS is that it can be computationally expensive and that there is no upper bound on the number of executions required to draw a fixed number of samples. Clearly, the applicability of the algorithm relies on a sufficiently high acceptance probability. If the acceptance probability is low, much computational effort is spent on proposing samples that are later rejected. Now, let us see what happens when we apply RS in spaces of increasing dimension. For the sake of illustration, assume that we wish to draw samples from the d-dimensional, standard Gaussian distribution. As proposal distribution, we use a d-dimensional, zero-mean Gaussian distribution with covariance matrix $\sigma_r^2 I_{d\times d}$. For the quotient between the target and the proposal to be bounded, we require that $\sigma_r \geq 1$. Then, the lowest bound on this quotient is given by $\rho = \sigma_r^d$, which implies that the acceptance probability is at best $1/\sigma_r^d$. Hence, the acceptance probability decays exponentially as we increase the dimension of the problem, and for high-dimensional problems the method can be impractical. This issue is known as the *curse of dimensionality*.

3 Sequential Monte Carlo

In Section 2.3, recursive expressions for several distributions of interest were given. However, to compute any of these distributions, we still face the problem of evaluating their updating formulas, e.g. (2.18) or (2.20) on page 20. As previously mentioned, analytical evaluation of these formulas is tractable only in a few special cases, basically if the underlying state-space model (SSM) is linear Gaussian or with a finite state-space. In the general case, we thus need to resort to approximations. The focus in this thesis will be on Monte Carlo (MC) approximation, as outlined in Section 2.4, leading to class of methods known as sequential Monte Carlo (SMC).

The basic procedure, underlying all the methods that we will consider in the sequel, is to approximate probability measures by targeting them with weighted particle systems. Hence, let us start by making the following definition.

Definition 3.1 (Weighted particle system). A weighted particle system on a measurable space $(\mathsf{Z}, \mathcal{Z})$ is a collection of random variables $\{Z_i, W_i\}_{i=1}^N$ on a probability space $(\Omega, \mathcal{F}, \mathrm{P})$ s.t.,

i) $Z_i : \Omega \to \mathsf{Z}$ for $i = 1, \dots, N$.

ii) $W_i : \Omega \to [0, 1]$ for $i = 1, \dots, N$.

iii) $\sum_i W_i = 1$.

We say that a weighted particle system defines an empirical probability distribution μ^N on $\mathcal{Z}$ by,

$$\mu^N = \sum_{i=1}^N W_i \delta_{Z_i}. \tag{3.1}$$

3.1 A general SMC sampler

Common to many inference problems is the objective of recursively evaluating a sequence of measures. This can for instance be the sequence of filtering distributions $\Phi_{t|t}$ or the sequence of joint smoothing distributions $\Phi_{1:t|t}$ for an increasing index t, but it is not limited to these cases (see e.g. Section 3.3). When analysing different MC based algorithms for these problems, it is inconvenient to be forced to treat each problem separately. Hence, we seek a unifying framework for sequential MC approximation of any sequence of measures; this framework will be denoted SMC. This section presents a general SMC sampler, which in the coming sections will be applied to different problems, leading to well known algorithms such as the particle filter (PF) and the Rao-Blackwellised particle filter (RBPF). The language and notation used in this section is largely influenced by Douc and Moulines [2008].

As mentioned above, we are generally interested in recursively approximating a sequence of measures, which in SMC is done by targeting these measures by a sequence of weighted particle systems. However, it can be realised that it is sufficient to consider a single step in such a recursion. If we find procedures of moving from one step to the next, these procedures can then serve as building blocks to construct sequential algorithms. Hence, let ν and μ be probability measures on the measurable spaces $(\tilde{\mathsf{X}}, \tilde{\mathcal{X}})$ and $(\mathsf{X}, \mathcal{X})$, respectively. Let $\{\tilde{\xi}_i, \tilde{\omega}_i\}_{i=1}^M$ be a weighted particle system, targeting the distribution ν. The basic component of the SMC sampler is then to transform this system into another weighted particle system $\{\xi_i, \omega_i\}_{i=1}^N$, which is targeting the distribution μ. We distinguish between two different cases. If ν and μ agree on the space $(\tilde{\mathsf{X}}, \tilde{\mathcal{X}}) = (\mathsf{X}, \mathcal{X})$, i.e. they are the same measure, the transformation of the weighted particle system is called *selection*. If this is not the case, the transformation is called *mutation*. The name mutation can be motivated by the fact that we "mutate" the particle system into targeting a different distribution. Selection, on the other hand, refer to transforming the particle system in a way which does not change the target distribution. This is typically done by "selecting" among the existing particles $\{\tilde{\xi}_i, \tilde{\omega}_i\}_{i=1}^M$ to construct the new particles $\{\xi_i, \omega_i\}_{i=1}^N$.

SMC can be seen as a combination and generalisation of the sequential importance sampling (SIS) [Handschin and Mayne, 1969] and sampling importance resampling (SIR) [Rubin, 1987] methods, where mutation is related to the former and selection to the latter. The multinomial resampling method outlined in Section 2.4.2 is an example of selection, but as we shall see in Section 3.1.2 the concept is much more general.

3.1.1 Mutation

To mutate a weighted particle system targeting ν, into a new system, targeting μ, we need to know how the two measures are related to each other. Hence, let $L(d\xi \mid \tilde{\xi})$ be a (not necessarily transition) kernel from $\tilde{\mathsf{X}}$ to X, such that,

$$\mu(d\xi) = \frac{\int L(d\xi \mid \tilde{\xi})\nu(d\tilde{\xi})}{\int L(\mathsf{X} \mid \tilde{\xi})\nu(d\tilde{\xi})}. \tag{3.2}$$

Clearly, for the expression to make sense, we also require that L is such that the denominator in (3.2) is strictly positive. A kernel L satisfying (3.2) will be called a *transformation* *kernel* for the measures ν and μ, since is tells us how to transform the initial measure ν into the target measure μ. For two given measures, ν and μ, there are many such kernels. From a superficial point of view we can choose to work with either one of them. For instance, to take $L(d\xi \mid \tilde{\xi}) \equiv \mu(d\xi)$ would suffice and as we shall see in Section 3.2.4, this is actually one of the choices that are used in practice. It is important to understand that a certain choice of L may impose a specific SMC algorithm, or a class of algorithms. However, since the analyses of these methods are independent of which L we use, as long as it satisfies (3.2), general results can be obtained which apply to all of them. See also the discussion in Section 3.2.3.

Next, we introduce a *proposal kernel* $R(d\xi \mid \tilde{\xi})$, as a transition kernel from $\tilde{\mathsf{X}}$ to X. R can be chosen arbitrarily, preferably so that we easily can sample from it, with the only constraint that $L(\,\cdot \mid \tilde{\xi}) \ll R(\,\cdot \mid \tilde{\xi})$, for all $\tilde{\xi} \in \tilde{\mathsf{X}}$. We further define the *weight function* $W : (\tilde{\mathsf{X}} \times \mathsf{X}) \to \mathbb{R}_+$ by,

$$W(\tilde{\xi}, \xi) = \frac{dL(\,\cdot \mid \tilde{\xi})}{dR(\,\cdot \mid \tilde{\xi})}(\xi). \tag{3.3}$$

Given a weighted particle system $\{\tilde{\xi}_i, \tilde{\omega}_i\}_{i=1}^M$ targeting ν, we now generate new particles according to the following procedure. To allow for a varying number of particles before and after the mutation, we assume that each particle $\tilde{\xi}_i$ gives rise to α offsprings, i.e. $N = \alpha M$. Rubin [1987] suggested to use multiple draws ($\alpha \gg 1$) in the SIR context, with the motivation that an increased number of particles before the resampling step will increase the number of distinct particles after the resampling step (see Section 3.1.2). The offsprings are generated from the proposal kernel according to,

$$\xi_{\alpha(i-1)+k} \mid \{\tilde{\xi}_j, \tilde{\omega}_j\}_{j=1}^M \sim R(d\xi \mid \tilde{\xi}_i), \tag{3.4}$$

and are assigned (unnormalised) importance weights using the weight function (3.3),

$$\omega'_{\alpha(i-1)+k} = \tilde{\omega}_i W(\tilde{\xi}_i, \xi_{\alpha(i-1)+k}), \tag{3.5a}$$

for $k = 1, \ldots, \alpha$ and $i = 1, \ldots, M$. After normalisation of the weights, i.e. by setting

$$\omega_i = \frac{\omega'_i}{\sum_k \omega'_k}, \qquad i = 1, \ldots, N, \tag{3.5b}$$

we obtain a weighted particle system $\{\xi_i, \omega_i\}_{i=1}^N$ targeting the distribution μ.

Remark 3.1. We have used the terminology that a weighted particle system *targets* a certain distribution, but we have yet not defined what this means. For the time being, it is enough with the "intuitive" interpretation that the sample behaves as if it was sampled from the targeted distribution using importance sampling (IS). In Chapter 4 the concept is more clearly pinned down, by introducing the notions of consistency and asymptotic normality.

3.1.2 Selection

SMC methods are designed to approximate a sequence of distributions, and by the mutation procedure described above we have a way of moving from one distribution in the sequence to the next. If we simply concatenate such mutation steps, we end up with the SIS algorithm. However, there is a problem with this approach. The recursive updating of the weights according to (3.5) is not stable, in the sense that the variance of the normalised weights ω_i (almost always) increases over "time" [Cappé et al., 2005, page 232]. This means that, as we move forward through the sequence, one of the weights will tend to one, whereas the remaining weights all tend to zero. The weighted particle system is generally used to compute approximations similarly to (2.30) on page 22. It is clear that a particle with a very small weight will have a minor contribution to the sum. We say that the effective number of particles has decreased. Consequently, when all but one weights tend to zero, we are basically left with a single particle, which of course is not enough to do accurate MC integration. This problem is known as depletion of the particle system.

To circumvent depletion, we introduce the second basic component of SMC, namely selection. Given a weighted particle system $\{\xi_i, \omega_i\}_{i=1}^N$ targeting the distribution μ, we wish to transform this into a new system $\{\tilde{\xi}_i, \tilde{\omega}_i\}_{i=1}^M$ targeting the same distribution μ. The aim is to do this in such a way that the sample variance of $\{\tilde{\omega}_i\}_{i=1}^M$ is lower than that of $\{\omega_i\}_{i=1}^N$. In most cases in practice, this idea is driven as far as possible by minimising the weight variance, i.e. by making sure that $\tilde{\omega}_i \equiv 1/M$ after selection.

It should be noted that selection generally increase the variance of any estimator derived from the weighted particle system. It should thus be conducted with care, since it degrades the performance of the SMC method. In practical applications, it is generally a good idea to keep track of the effective number of particles, and only apply selection when this number is below some threshold. Gustafsson [2010], among others, provide technical details.

We distinguish between three different approaches to selection.

Resampling

The most common way (at least in the literature) to do selection, is by resampling. In this approach, the new particle set is constructed by random draws among the existing particles, i.e. $\tilde{\xi}_i \in \{\xi_j\}_{j=1}^N$ for $i = 1, \ldots, M$. The number of particles after resampling is fixed a priori, i.e. M is $\mathcal{F}_N$-measurable, with $\mathcal{F}_N = \sigma(\{\xi_j, \omega_j\}_{j=1}^N)$. Furthermore, the sampling is required to be unbiased; if M_j is the number of times particle j is replicated, it should be the case that,

$$\mathrm{E}[M_j \mid \mathcal{F}_N] = M\omega_j, \qquad j = 1, \ldots, N. \tag{3.6}$$

This property also implies that the resulting weights should all be equal, i.e. resampling produces an equally weighted particle system $\{\tilde{\xi}_i, 1/M\}_{i=1}^M$.

The simplest resampling scheme is multinomial resampling (see [Rubin, 1987] and the discussion in Section 2.4.2). In this method, the new particles are generated as conditionally i.i.d. samples from the empirical distribution defined by $\{\xi_j, \omega_j\}_{j=1}^N$, i.e.

$$\mathrm{P}(\tilde{\xi}_i = \xi_j \mid \mathcal{F}_N) = \omega_j, \qquad i = 1, \ldots, M. \tag{3.7}$$

The name multinomial resampling comes from the fact that the number of offsprings $\{M_j\}_{j=1}^N$ of each particle is multinomially distributed. Multinomial resampling was used in the seminal SMC paper by Gordon et al. [1993]. However, to reduce the variance induced by the selection step, several alternatives have emerged. These include residual resampling [Whitley, 1994, Liu and Chen, 1998] and stratified resampling [Kitagawa, 1996], which both can be shown to introduce less variance than multinomial resampling (see [Cappé et al., 2005, Section 7.4.2] and [Douc and Moulines, 2008]). Another alternative, often recommended as the method of choice, is systematic resampling by Carpenter et al. [1999] (also mentioned by Whitley [1994] under the name universal sampling). It is often conjectured that systematic resampling always outperforms multinomial resampling, but this is not true, as shown by Cappé et al. [2005], page 249. However, in practice it often performs well, as indicated by the empirical study by Hol et al. [2006]. See also [Hendeby, 2008] for an illustrative review of the different resampling methods.

Branching

An alternative selection strategy to resampling is branching. The difference between the two strategies, is that the latter uses a random number of particles. Hence, a branching procedure generates M particles, where M is a (non $\mathcal{F}_N$-measurable) random variable. That is, based on the particles and weights available before the selection, we cannot a priori compute the number of particles that will be given after the selection. Just as for resampling, there are many different ways to do branching, see e.g. [Crisan et al., 1999]. A simple approach, related to multinomial resampling, is binomial branching. In this method, we choose a fixed number M', which is the desired, or nominal number of particles. The number of offsprings $\{M_j\}_{j=1}^N$ of each particle are then independent draws from a binomial distribution according to,

$$M_j \sim \text{Bin}(M', \omega_j). \tag{3.8}$$

This approach is unbiased, similarly to (3.6), but as previously mentioned, it may very well be the case that,

$$\sum_{j=1}^{N} M_j \neq M'. \tag{3.9}$$

One of the main motives for using branching is to open up for parallel implementation of SMC methods. Branching methods can also be easier to implement than resampling. The drawback, of course, is that the user cannot control the number of particles. Just as with resampling, the branching procedures produce equally weighted particle systems.

Fractional reweighting

As indicated by the preceding notation, it is not necessary that the selection step generates an equally weighted particle system. Liu et al. [2001] have suggested a method which retains a fraction of the weights through the selection step. This approach has by Douc and Moulines [2008] been called fractional reweighting. This idea can then be combined with the resampling or branching methods mentioned above. Other ways to transform a weighted particle system into a new system with non-equal weights are of course also possible. We group these methods together into their own category, separate from the resampling and branching methods, simply to emphasise the fact that selection need not always lead to an equally weighted particle system.

Remark 3.2. Throughout the remaining of this thesis, we shall confine ourselves to using resampling as selection procedure. In all experiments conducted, multinomial resampling has been used. Furthermore, for notational simplicity, we shall assume that the number of particles is kept constant at N. However, it is important to remember that a varying number of particles indeed can be interesting from a practical point of view, as mentioned in Section 3.1.1. The ideas and results provided in this thesis can all be easily extended to deal with different selection procedures and a varying number of particles.

3.2 Particle filter

This section is devoted to the particle filter (PF), the flagship of the SMC methods. The PF is often credited Gordon et al. [1993], who were the first to combine the two crucial parts, mutation and selection, resulting in a functioning algorithm. The SIS method by Handschin and Mayne [1969] shows great resemblance to the PF, but lacks the selection step. Hence, the method will suffer from weight depletion. During the 90's, independent developments were made by, among others Kitagawa [1996] and Isard and Blake [1998]. The research in the area has since then steadily intensified. For an overview and further reference, see e.g. the work by Doucet et al. [2001a], Arulampalam et al. [2002], Cappé et al. [2007] and Doucet and Johansen [2011].

In Section 3.2.3, the PF is derived with the general SMC framework of Section 3.1 in mind. However, before that, we give a more intuitive presentation of the PF in Section 3.2.1. The intention is to present the material in a more easily digested form, which may also be more familiar to some readers. In that way, we provide some links between the different types of notation.

The PF is an SMC method designed to target either the filtering distribution or the joint smoothing distribution. It turns out that the algorithms will be identical for these two choices of target distributions. This is not that surprising, since the filtering distribution is one of the marginals of the joint smoothing distribution. Hence, if we attain the latter, we also have the former. Due to this, we provide the details of the derivation, only for the PF targeting the joint smoothing distribution. At the end on Section 3.2.3, we comment on the difference in point of view, if we instead wish to target the filtering distribution.

3.2.1 An intuitive preview of the PF

Throughout this section, we assume that the model is fully dominated. As mentioned above, the PF aims at approximating the joint smoothing distribution using sequences of weighted particle systems. Hence, the density of the target distribution is $p(x_{1:t} \mid y_{1:t})$. To sequentially draw samples from this distribution using IS, we choose a proposal density which factorises according to,

$$r'_t(x_{1:t} \mid y_{1:t}) = r_t(x_t \mid x_{1:t-1}, y_{1:t}) \underbrace{r'_{t-1}(x_{1:t-1} \mid y_{1:t-1})}_{\text{previous proposal}}. \tag{3.10}$$

Remark 3.3. The proposal density at time t is usually allowed to depend on the "old" state trajectory, $x_{1:t-1}$ as well as the measurement sequence up to time t, $y_{1:t}$. It is common practice to indicate this dependence, by writing the proposal as a conditional PDF as in (3.10).

The reason for why we require a factorisation according to (3.10), is to allow for a sequential algorithm. Let $\{\tilde{x}^i_{1:t-1}, \tilde{w}^i_{t-1}\}_{i=1}^N$ be a weighted particle system targeting the joint smoothing distribution at time $t-1$. Sampling from (3.10) is done by keeping the existing particle trajectories and extending them with samples at time t, i.e.

$$x_t^i \sim r_t(x_t \mid \tilde{x}_{1:t-1}^i, y_{1:t}), \tag{3.11a}$$

$$x_{1:t}^i := \{\tilde{x}_{1:t-1}^i, x_t^i\}, \tag{3.11b}$$

for $i = 1, \ldots, N$.

Since we sample from a proposal density, rather than from the target density itself, the samples need to be weighted in accordance with (2.32c) on page 23. The importance weights are given by,

$$w_t^i = \frac{p(x_{1:t}^i \mid y_{1:t})}{r_t'(x_{1:t}^i \mid y_{1:t})}. \tag{3.12a}$$

By using the forward recursion for the joint smoothing distribution (2.21) on page 20, the numerator in the expression above can be expanded, yielding

$$w_t^i \propto \frac{p(y_t \mid x_t^i)p(x_t^i \mid x_{t-1}^i)}{r_t(x_t^i \mid x_{1:t-1}^i, y_{1:t})} \underbrace{\frac{p(x_{1:t-1}^i \mid y_{1:t-1})}{r_{t-1}'(x_{1:t-1}^i \mid y_{1:t-1})}}_{=\tilde{w}_{t-1}^i}. \tag{3.12b}$$

Hence, we obtain a sequential updating formula also for the importance weights. Since we only know the weights up to proportionality, they are normalised to sum to one, as discussed in Section 2.4.1. The resulting weighted particle system $\{x_{1:t}^i, w_t^i\}_{i=1}^N$ targets the density $p(x_{1:t} \mid y_{1:t})$. To avoid depletion of the particle system, we may also choose to apply a selection scheme, e.g. one of the methods discussed in Section 3.1.2. We summarise the steps of the PF in Algorithm 3.1.

One obvious question at this point is how to choose the proposal density r_t. The simplest choice is to sample from the transition density function $p(x_t \mid x_{t-1})$, i.e. by simulating the system dynamics one time step for each particle. In this case, the weight expression (3.12b) reduces to $w_t^i \propto p(y_t \mid x_t^i)\tilde{w}_{t-1}^i$. If this choice of proposal density is combined with multinomial resampling, performed at each iteration of the algorithm, we obtain the original PF by Gordon et al. [1993], known as the bootstrap filter. Since we will refer to this specific version of the PF in the following chapters, we make an explicit definition of what is meant by the bootstrap PF.

Definition 3.2 (Bootstrap PF). The bootstrap PF is a PF according to Algorithm 3.1 in which,

i) the transition density is used as proposal, i.e. $r_t(x_t \mid x_{1:t-1}, y_{1:t}) = p(x_t \mid x_{t-1})$.

ii) selection is done by multinomial resampling at each iteration of the algorithm.

Though widely used in practice, it should be noted that the transition density function may not be the best choice of proposal. The reason is that it does not take the measurement y_t into account, when proposing particles at time t. Hence, it might be that the particles are placed in regions of the state-space with low posterior probability (given y_t). Based on this insight, it is natural to instead try to sample from the density $p(x_t \mid x_{t-1}, y_t)$. This is known as the optimal proposal function. However, this density is in most cases not

Algorithm 3.1 Particle filter (PF)

Note: This PF targets the joint smoothing distribution in a fully dominated SSM.

Input: A weighted particle system $\{x^i_{1:t-1}, w^i_{t-1}\}^N_{i=1}$ targeting $p(x_{1:t-1} \mid y_{1:t-1})$.

Output: A weighted particle system $\{x^i_{1:t}, w^i_t\}^N_{i=1}$ targeting $p(x_{1:t} \mid y_{1:t})$.

```
Selection:
1: Optionally, generate a new weighted particle system $\{\tilde{x}^i_{1:t-1}, \tilde{w}^i_{t-1}\}^N_{i=1}$ by selection,
   or set $\{\tilde{x}^i_{1:t-1}, \tilde{w}^i_{t-1}\}^N_{i=1} = \{x^i_{1:t-1}, w^i_{t-1}\}^N_{i=1}$.
Mutation:
2: Augment the sample trajectories. For $i = 1, \dots, N$,
       $x^i_t \sim r_t(x_t \mid \tilde{x}^i_{1:t-1}, y_{1:t})$,
       $x^i_{1:t} = \{\tilde{x}^i_{1:t-1}, x^i_t\}$.
3: Compute the unnormalised importance weights. For $i = 1, \dots, N$,
       $w'^i_t = \frac{p(y_t \mid x^i_t) p(x^i_t \mid x^i_{t-1})}{r_t(x^i_t \mid x^i_{1:t-1}, y_{1:t})} \tilde{w}^i_{t-1}$,
4: Normalise the weights. For $i = 1, \dots, N$,
       $w^i_t = \frac{w'^i_t}{\sum_k w'^k_t}$,
```

available to sample from. One option is then to approximate the optimal proposal, e.g. by using local linearisation of the measurement equation [Doucet et al., 2000b]. We will not review the details of this approach here, but in Section 3.3.4 we make similar calculations to approximate the optimal proposal function for the RBPF.

At the beginning of Section 3.2, we claimed that a PF targeting the filtering distribution would result in an identical algorithm as when targeting the joint smoothing distribution. So, how do we obtain a weighted particle system targeting the filtering distribution from Algorithm 3.1? The answer is very simple. The empirical distribution defined by $\{x^i_{1:t}, w^i_t\}^N_{i=1}$ is,

$$\widehat{\Phi}^N_{1:t|t}(dx_{1:t}) = \sum_{i=1}^N w^i_t \delta_{x^i_{1:t}}(dx_{1:t}). \tag{3.13}$$

By marginalising this over $dx_{1:t-1}$ we get,

$$\widehat{\Phi}^N_{t|t}(dx_t) = \int_{\mathsf{X}^{t-1}} \widehat{\Phi}^N_{1:t|t}(dx_{1:t}) = \sum_{i=1}^N w^i_t \delta_{x^i_t}(dx_t). \tag{3.14}$$

Hence, by throwing away the history of the particle trajectories, we end up with a weighted particle system $\{x^i_t, w^i_t\}^N_{i=1}$ targeting the filtering distribution. In fact, even though the PF in Algorithm 3.1 targets the joint smoothing distribution, it is in practice used mostly for filtering or fixed-lag smoothing. The reason is, as we shall see in the next section, that the particle trajectories degenerate, providing good estimates of marginals $\Phi_{t-\ell+1:t|t}$ only for small enough ℓ. How to circumvent this problem will be the topic of Chapter 5.

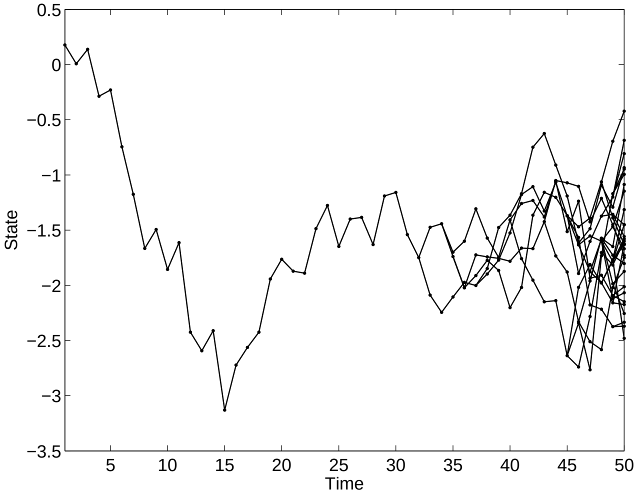

Figure 3.1: *Particle trajectories at time* $t = 50$.

3.2.2 Degeneracy

Assume that we employ a PF to target the joint smoothing distribution. At time s, we generate N particles $\{x_s^i\}_{i=1}^N$ from a proposal kernel and append these to the existing particle trajectories, according to (3.11). Assuming that all the generated particles are unique, we say that the unique particle count at time s is N. We thus have a weighted particle system $\{x_{1:s}^i, w_s^i\}_{i=1}^N$ targeting the joint smoothing distribution at time s. Now, assume that the particle trajectories are resampled at time s (e.g. due to significant weight depletion), resulting in an unweighted particle system $\{\tilde{x}_{1:s}^i, 1/N\}_{i=1}^N$. Then, there is a nonzero probability that the unique particle count at time s has decreased. This is in fact the purpose of any selection scheme, to remove particles with small weights and duplicate particles with large weights, which has the effect of decreasing the unique number of particles. Now, as we proceed through time, each consecutive resampling of the particle trajectories will cause the unique particle count at time s to decrease. Eventually the unique particle count will tend to one. In other words, for a large enough time $t \gg s$ all particle trajectories $\{x_{1:t}^i\}_{i=1}^N$ will share a common ancestor at time s (and consequently for any time prior to s). The implication of this is that the particle trajectories generated by the PF, though targeting the joint smoothing distribution, only provide accurate approximations of fixed-lag smoothing distributions for short enough lags. This problem, known as degeneracy, is further illustrated in the example below.

Example 3.1: Degeneracy

A bootstrap PF with $N = 30$ particles is used to target the joint smoothing distribution for a one-dimensional Gaussian random walk process measured in Gaussian noise. At time $t = 50$ the joint smoothing distribution is targeted by a weighted particle system $\{x_{1:50}^i, w_{50}^i\}_{i=1}^{30}$. Figure 3.1 depicts the particle trajectories over time. For any time point $s \leq 32$ all particle trajectories coincide, i.e. the unique particle count is one.

Assume, for instance, that we are interested in the smoothed estimate of the initial condition, $\mathrm{E}[X_1 \mid Y_{1:50} = y_{1:50}]$. Based on the PF trajectories, we would compute an estimate according to,

$$\hat{x}_1 = \sum_{i=1}^{30} w_{50}^i x_1^i, \tag{3.15}$$

but since x_1^i are identical for all $i = 1, \ldots, 30$, this is in effect a MC integration using a single sample. Hence, due to the degeneracy, we can not expect to obtain an accurate estimate from the PF.

Remark 3.4. It should be noted that the PF indeed generates "proper" weighted samples from the joint smoothing distribution. The problem that we face when using these samples for MC integration is that the selection procedure introduces a dependence between the samples. When the trajectories have degenerated to the extent that the unique particle count is one, the sample trajectories are perfectly correlated.

3.2.3 The PF in the general SMC framework

In Section 3.2.1 we gave a self-contained presentation of the PF. We shall now see how this fits into the general SMC framework of Section 3.1.

Transformation and proposal kernels for the joint smoothing distribution

Assume that $\{\tilde{x}_{1:t-1}^i, \tilde{w}_{t-1}^i\}_{i=1}^N$ targets the joint smoothing distribution at time $t-1$, i.e. $\Phi_{1:t-1|t-1}$. We wish to transform this system into a new weighted particle system, targeting $\Phi_{1:t|t}$. Define a sequence of kernels by,

$$L_t(dx_{1:t} \mid \tilde{x}_{1:t-1}) = p(y_t \mid x_t) Q(dx_t \mid x_{t-1}) \delta_{\tilde{x}_{1:t-1}}(dx_{1:t-1}). \tag{3.16}$$

Now, consider (3.2) in which we let L be given by (3.16) and let $\Phi_{1:t-1|t-1}$ take the role of ν. It can be straightforwardly verified that (3.2) in this case coincides with the forward recursions for the joint smoothing distribution given by (2.20) on page 20. Hence, the sequence of kernels (3.16) tells us how the joint smoothing distribution evolves over a sequence of state-spaces X^t of increasing dimension. We say that (3.16) defines a sequence of transformation kernels for the joint smoothing distribution. Note that L_t is a kernel from the space X^{t-1} to X^t. Hence, from this point of view we "decouple" the state trajectory at time $t-1$, $\tilde{x}_{1:t-1}$, from the state trajectory at time t, $x_{1:t}$. The kernel (3.16) then tells us what the distribution of $x_{1:t}$ is for a given $\tilde{x}_{1:t-1}$. However, by analysing the form of (3.16) we realise that the trajectories are indeed coupled, due to the presence of a point mass $\delta_{\tilde{x}_{1:t-1}}(dx_{1:t-1})$. This implies that the kernel (3.16), for a given $\tilde{x}_{1:t-1}$, does not "move" the old trajectory, but simply extends the distribution to the larger set X^t.

So far, we have identified the kernel describing how the joint smoothing distribution evolves (or rather *a* kernel since it is not unique; see the discussion below). Next, we seek a proposal kernel $R_t(dx_{1:t} \mid \tilde{x}_{1:t-1})$ which will be used to propose new sample trajectories in the PF. Since we require that $L_t(\,\cdot \mid \tilde{x}_{1:t-1}) \ll R_t(\,\cdot \mid \tilde{x}_{1:t-1})$ for any $\tilde{x}_{1:t-1} \in \mathsf{X}^{t-1}$, it must follow that R_t also assigns some probability mass to the singleton

$\tilde{x}_{1:t-1}$. Due to this, we restrict ourselves to proposal kernels of the form

$$R_t(dx_{1:t} \mid \tilde{x}_{1:t-1}) = r_t(dx_t \mid \tilde{x}_{1:t-1})\delta_{\tilde{x}_{1:t-1}}(dx_{1:t-1}). \tag{3.17}$$

Remark 3.5 (The proposal kernel depends on the observations). At this point, we encounter an unpleasant inconsistency in the notation used in this thesis. Still, it is believed that this notation is in least conflict with the common practice used in the literature. As pointed out in Remark 3.3 on page 34, when dealing with fully dominated models and expressing the PF in terms of densities, we let the proposal density have an *explicit* dependence on the measurement sequence $y_{1:t}$ as in (3.10). On the contrary, the proposal kernel in (3.17) has an *implicit* dependence on the measurements $y_{1:t}$, indicated only by the time index t. For instance, the density of the kernel $r_t(dx_t \mid \tilde{x}_{1:t-1})$ is thus (if it exists) given by $r_t(x_t \mid \tilde{x}_{1:t-1}, y_{1:t})$; see (3.10). The implicit dependence on the measurements for the proposal kernel, is consistent with the notation used for e.g. the filtering distribution $\Phi_{t|t}(dx_t)$ and the transformation kernel (3.16) (these too, depend implicitly on the measurements). What we seek to clarify by this remark, is that the proposal kernel/density in general is allowed to depend on the measurement, whether this is explicit in the notation or not.

To sample from the above kernel, given $\tilde{x}_{1:t-1}$, is exactly the sampling procedure described in Section 3.2.1. We simply keep the "old" trajectory up to time $t-1$ and append a sample from time t. Hence, we generate

$$x_t^i \sim r_t(dx_t \mid \tilde{x}_{1:t-1}^i) \tag{3.18}$$

and set $x_{1:t}^i := \{\tilde{x}_{1:t-1}^i, x_t^i\}$ for $i = 1, \ldots, N$. The importance weights are computed from (3.5) using the weight function (3.3), resulting in

$$w_t^i \propto \tilde{w}_{t-1}^i p(y_t \mid x_t^i) \frac{dQ(\,\cdot \mid x_{t-1}^i)}{dr_t(\,\cdot \mid x_{1:t-1}^i)}(x_t^i) \tag{3.19}$$

Note that the weight function here depends solely on the "new" particle trajectory $x_{1:t}^i$.

The above procedure shows how to mutate a weighted particle system $\{\tilde{x}_{1:t-1}^i, \tilde{w}_{t-1}^i\}_{i=1}^N$ targeting $\Phi_{1:t-1|t-1}$, into a new system $\{x_{1:t}^i, w_t^i\}_{i=1}^N$ targeting $\Phi_{1:t|t}$. We may also, if we so wish, combine this with a selection step as described in Section 3.1.2, which completes the PF.

Non-uniqueness of the transformation kernel

As previously mentioned, the transformation kernel can often be chosen in many different ways. It is thus natural to ask, is (3.16) the only kernel describing the evolution of the joint smoothing distribution? The answer is no. As pointed out in Section 3.1, another option would be to take (see also Section 3.2.4),

$$L_t'(dx_{1:t} \mid \tilde{x}_{1:t-1}) = \Phi_{1:t|t}(dx_{1:t}). \tag{3.20}$$

Then, why did we make the particular choice (3.16)? There are basically two reasons.

1. The transformation kernel should preferably be of known functional form, otherwise it is not possible to compute the weight function (3.3). This is not the case e.g. for the choice (3.20).

2. The choice of transformation kernel must be guided by the algorithm that we are aiming for. A specific choice of kernel may impose certain algorithmic properties.

In the derivation of the PF above, the transformation kernel was chosen according to (3.16) mainly to allow for a factorisation of the proposal kernel according to (3.17). Since we are aiming for a sequential algorithm, it is natural to construct the sample trajectories in the way described above, i.e. to keep the "old" trajectory and append a new sample at each time point. This requirement implies the form (3.17) for the proposal kernels, which in turn suggests the choice (3.16) for the transformation kernel.

To see that other choices of transformation kernels indeed can be of interest, let us consider the resample-move algorithm by Gilks and Berzuini [2001] (see also [Doucet et al., 2001b] for a similar approach). To increase the sample diversity after resampling, they suggest to apply a Markov chain Monte Carlo (MCMC) move on the particle trajectories. That is, we sample new trajectories from a kernel $\kappa_{t-1}(dx_{1:t-1} \mid \tilde{x}_{1:t-1})$, with invariant distribution $\Phi_{1:t-1|t-1}$, i.e.

$$\int \kappa_{t-1}(dx_{1:t-1} \mid \tilde{x}_{1:t-1})\Phi_{1:t-1|t-1}(d\tilde{x}_{1:t-1}) = \Phi_{1:t-1|t-1}(dx_{1:t-1}). \tag{3.21}$$

The effect of this additional MCMC step, is that the sequence of proposal kernels (3.17) is replaced by,

$$R''_t(dx_{1:t} \mid \tilde{x}_{1:t-1}) = r_t(dx_t \mid \tilde{x}_{1:t-1})\kappa_{t-1}(dx_{1:t-1} \mid \tilde{x}_{1:t-1}). \tag{3.22}$$

Since we require the proposal kernel to dominate the transformation kernel, the above choice of proposal kernel may not be compatible with the transformation kernel according to (3.16). However, (3.22) suggests that we instead should consider a transformation kernel according to,

$$L''_t(dx_{1:t} \mid \tilde{x}_{1:t-1}) = p(y_t \mid x_t)Q(dx_t \mid x_{t-1})\kappa_{t-1}(dx_{1:t-1} \mid \tilde{x}_{1:t-1}), \tag{3.23}$$

which, due to the $\Phi_{1:t-1|t-1}$-invariance of κ_{t-1}, also satisfies the forward recursions for the joint smoothing distribution (2.20).

Hence, the transformation kernel does not tell us how to construct SMC algorithms. On the contrary, the algorithms often arise on heuristic grounds, and are not put into the general framework until a later stage. Why then, should we bother about the transformation kernel at all, if it does not influence the construction of the algorithms? The reason is, as mentioned also in Section 3.1, to simplify an analysis of the methods. If the algorithms can be expressed in a common form, general conclusions can be drawn by analysing this unifying framework.

Filtering or joint smoothing?

Before we leave this section, we should comment on the differences in point of view, if we choose to target the filtering distribution instead of the joint smoothing distribution. In this case, we target the measure $\Phi_{t-1|t-1}(d\tilde{x}_{t-1})$ at time $t-1$ and $\Phi_{t|t}(dx_t)$ at time t, i.e. they are both measures on X. Hence, both the transformation kernel and the proposal kernel from X to X. By studying the filtering recursions (2.18) on page 20, we see that a natural choice is to take,

$$L'''_t(dx_t \mid \tilde{x}_{t-1}) = p(y_t \mid x_t)Q(dx_t \mid \tilde{x}_{t-1}). \tag{3.24}$$

Similarly, we "drop the tail" from the proposal kernel as well, which takes the (most general) form

$$R_t'''(dx_t \mid \tilde{x}_{t-1}). \tag{3.25}$$

By going through the steps of the PF, it can be easily seen that the algorithms turn out to be identical, regardless of the target being the joint smoothing distribution or the filtering distribution. The only difference is that, in the former case we keep the history of the particle trajectories, whereas in the latter we throw the trajectories away. This is in agreement with the PF presentation in Section 3.2.1.

Finally, a technical detail worth mentioning is that, for the PF targeting the joint smoothing distribution, the weight function (3.3) depends only on the "new" particle trajectories (in the notation of (3.3), the function is independent of $\tilde{\xi}$, see also (3.19)). This is not the case if the target is the filtering distribution, i.e. the weight function (3.3) will depend on both $\tilde{x}_{t-1}$ and x_t (or $\tilde{\xi}$ and ξ in the notation of (3.3)).

3.2.4 Marginal particle filter

Above, it was mentioned that (3.20) is one possible transformation kernel for the joint smoothing distribution. We also said that this choice would not be of any use, since it does not allow for evaluation of the weight function (3.3). However, this is not completely true, since an approximate evaluation of the weight function still may be feasible. This idea has been investigated for the filtering problem by Klaas et al. [2005], resulting in what they call the marginal particle filter (MPF). Klaas et al. [2005] provide two reasons for why the MPF is an interesting alternative to the PF. First, they show that the weight variance will be lower in the MPF than in the PF. Second, it is conjectured that the MPF should be more robust to errors in the transition model, which always will be present for real world problems.

Guided by the forward filtering recursion (2.18) on page 20 we define a sequence of measures by,

$$L_t(dx_t) = p(y_t \mid x_t) \int_{\mathsf{X}} Q(dx_t \mid \tilde{x}_{t-1}) \Phi_{t-1|t-1}(d\tilde{x}_{t-1}) \propto \Phi_{t|t}(dx_t). \tag{3.26}$$

With the target measure (μ in (3.2)) being $\Phi_{t|t}$, the above measure L_t serves as a transformation kernel in (3.2). However, since L_t does not depend on any "ancestor particle", we choose to call it a measure rather than a kernel. Note also that, for this choice of L, the initial measure ν is arbitrary in (3.2).

Now, assume that we have generated a weighted particle system $\{x_{t-1}^i, w_{t-1}^i\}_{i=1}^N$ targeting the filtering distribution at time $t-1$, $\Phi_{t-1|t-1}$. Guided by Klaas et al. [2005], we choose a proposal distribution according to,

$$R_t(dx_t) = \sum_{i=1}^N w_{t-1}^i r_t(dx_t \mid x_{t-1}^i). \tag{3.27}$$

Hence, the proposal distribution is a mixture, in which each component originates from one of the particles at time $t-1$. We note that sampling from (3.27) is equivalent to multinomial resampling followed by a mutation step in the PF. Hence, the difference between the MPF and the PF lies solely in the computation of the importance weights. After proposing a set of particles $\{x_t^i\}_{i=1}^N$ from (3.27), the weights should by (3.3) be computed according to,

$$w_t^i \propto \frac{dL_t}{dR_t}(x_t^i). \tag{3.28}$$

However, due to the dependence on $\Phi_{t-1|t-1}$ (and also the presence of the generally intractable integral) in (3.26), it is in general not possible to evaluate this Radon-Nikodym derivative. To circumvent this, Klaas et al. [2005] replace the filtering distribution in (3.26) with its empirical approximation, leading to an approximation of L_t,

$$\widehat{L}_t^N(dx_t) = p(y_t \mid x_t) \sum_{i=1}^N w_{t-1}^i Q(dx_t \mid x_{t-1}^i). \tag{3.29}$$

By using this approximation in (3.28), the weights can often be computed more straightforwardly.

Remark 3.6. For a fully dominated model, the MPF importance weights are given by a quotient of densities,

$$w_t^i \propto \frac{p(y_t \mid x_t^i) \sum_{j=1}^N w_{t-1}^j p(x_t^i \mid x_{t-1}^j)}{\sum_{j=1}^N w_{t-1}^j r_t(x_t^i \mid x_{t-1}^j, y_{1:t})}.$$

We summarise the MPF in Algorithm 3.2. Both the PF and the MPF can be seen as targeting the filtering distribution, and the big difference between the two is in the computation of the weights. Since the latter uses approximate weight evaluation, it is in fact not an SMC method in terms of the general framework as defined in Section 3.1. Hence, an analysis of this framework may not be applicable to the MPF, and one should proceed with care if making claims about e.g. the convergence properties of the MPF, based on the general analysis. Finally, as pointed out also by Klaas et al. [2005], it should be noted that the MPF is equivalent to the PF if the transition kernel is used as proposal, i.e. if $r_t = Q$.

3.3 Rao-Blackwellised particle filter

The application of a PF to address the filtering problem is, of course, mainly of interest when an analytic evaluation of the filtering recursions is not possible. As previously mentioned, this is the case for basically any nonlinear and/or non-Gaussian model. However, for certain problems it may be possible to evaluate some "part" of the filtering recursions analytically, and it can then be sufficient to employ particles only for the remaining, intractable "part". If this is the case, we say that there exists a tractable substructure in the model. By exploiting any such substructure, we can possibly obtain better estimates than provided by a PF, targeting the full model. This is the idea behind the Rao-Blackwellised particle filter (RBPF), suggested by Doucet et al. [2000b] and Schön et al. [2005].

Assume, as in Section 2.2.2, that the state variable can be partitioned according to $X_t = \{\Xi_t, Z_t\}$ and $\mathsf{X} = \mathsf{X}_\xi \times \mathsf{X}_z$. This suggests that we can factorise the joint smoothing

Algorithm 3.2 Marginal particle filter (MPF)

Input: A weighted particle system $\{x^i_{1:t-1}, w^i_{t-1}\}_{i=1}^N$ targeting $p(x_{1:t-1} \mid y_{1:t-1})$.
Output: A weighted particle system $\{x^i_{1:t}, w^i_t\}_{i=1}^N$ targeting $p(x_{1:t} \mid y_{1:t})$.

```
1: Draw samples from a mixture proposal. For $i = 1, \dots, N$,
       $x^i_t \sim R_t(dx_t) = \sum_{j=1}^N w^j_{t-1} r_t(dx_t \mid x^j_{t-1}).$
2: Compute the unnormalised importance weights. For $i = 1, \dots, N$,
       $w'^i_t = \frac{d\widehat{L}^N_t}{dR_t}(x^i_t) = \frac{p(y_t \mid x^i_t) \sum_{j=1}^N w^j_{t-1} p(x^i_t \mid x^j_{t-1})}{\sum_{j=1}^N w^j_{t-1} r_t(x^i_t \mid x^j_{t-1}, y_{1:t})}.$
   Note: The second equality holds for a fully dominated SSM.
3: Normalise the weights. For $i = 1, \dots, N$,
       $w^i_t = \frac{w'^i_t}{\sum_k w'^k_t},$
```

distribution according to,

$$\Phi_{1:t|t}(dx_{1:t}) = \Phi^m_{1:t|t}(d\xi_{1:t})\Phi^c_{1:t|t}(dz_{1:t} \mid \xi_{1:t}), \tag{3.30}$$

where $\{\xi_t, z_t\}$ identifies to x_t. Here, $\Phi^m_{1:t|t}$ is the marginal distribution of $\Xi_{1:t}$ given $Y_{1:t} = y_{1:t}$. Since this distribution is a marginal of the joint smoothing distribution, it will be called the state-marginal smoothing distribution. The prefix "state" is used to distinguish it from what we normally mean by the marginal smoothing distribution, i.e. $\Phi_{t|T}$ (see Table 2.1). Furthermore, $\Phi^c_{1:t|t}$ is the conditional smoothing distribution of $Z_{1:t}$ given $\Xi_{1:t} = \xi_{1:t}$ and $Y_{1:t} = y_{1:t}$.

Remark 3.7. More precisely, $\Phi^c_{1:t|t}$ is a transition kernel from X^t_ξ to X^t_z. For each fixed $\xi_{1:t}$, $\Phi^c_{1:t|t}(\,\cdot \mid \xi_{1:t})$ is a probability measure on X^t_z, and can hence be viewed as a conditional distribution. In the notation introduced in (3.30), the meaning is that $\Phi_{1:t|t}$ is the product of the measure $\Phi^m_{1:t|t}$ and the kernel $\Phi^c_{1:t|t}$.

Now, assume that the conditional distribution $\Phi^c_{1:t|t}$ is analytically tractable. It is then sufficient to employ particles for the intractable part $\Phi^m_{1:t|t}$ and make use of the analytic tractability for the remaining part. Too see the difference between the PF and RBPF, assume that we seek to estimate the expectation of some function $\varphi : \mathsf{X}^t \to \mathbb{R}$ under the joint smoothing distribution, $\Phi_{1:t|t}(\varphi) = \mathrm{E}[\varphi(\{\Xi_{1:t}, Z_{1:t}\}) \mid Y_{1:t} = y_{1:t}]$. In the PF, we would then generate a weighted particle system $\{x^i_{1:t}, w^i_t\}_{i=1}^N$ targeting $\Phi_{1:t|t}$ and compute an estimate according to,

$$\hat{\varphi}^N_{\text{PF}} = \sum_{i=1}^N w^i_t \varphi(x^i_{1:t}). \tag{3.31a}$$

In the RBPF, the key observation is that (3.30) implies that the estimand can be decomposed as $\Phi_{1:t|t}(\varphi) = \Phi^m_{1:t|t}(\Phi^c_{1:t|t}(\varphi))$. If $\Phi^c_{1:t|t}(\varphi)$ is assumed to be a known function of $\xi_{1:t}$, we can generate a weighted particle system $\{\xi^i_{1:t}, \omega^i_t\}_{i=1}^N$ targeting $\Phi^m_{1:t|t}$ and compute an estimate as,

$$\begin{aligned}\hat{\varphi}^N_{\text{RBPF}} &= \sum_{i=1}^N \omega^i_t \Phi^c_{1:t|t}(\varphi(\{\xi^i_{1:t}, \cdot\}) \mid \xi^i_{1:t}) \\ &= \sum_{i=1}^N \omega^i_t E[\varphi(\{\Xi_{1:t}, Z_{1:t}\}) \mid \Xi_{1:t} = \xi^i_{1:t}, Y_{1:t} = y_{1:t}]. \end{aligned} \tag{3.31b}$$

Moving from (3.31a) to (3.31b) resembles a Rao-Blackwellisation of the estimator (3.31a), as briefly discussed in Section 1.1.2 (see also [Lehmann, 1983]). In some sense, we move from a Monte Carlo integration to a partially analytical integration. This is also the reason for why the RBPF is called as it is. However, it is not clear that the RBPF estimator (3.31b) truly is a Rao-Blackwellisation of (3.31a), in the factual meaning of the concept. That is, it is not obvious that the conditional expectation of (3.31a) given $\{\xi^i_{1:t}\}_{i=1}^N$ results in the expression (3.31b). This is due to the nontrivial relationship between the normalised weights generated by the PF $\{w^i_t\}_{i=1}^N$, and those generated by the RBPF $\{\omega^i_t\}_{i=1}^N$. It can thus be said that the RBPF has earned its name from being inspired by the Rao-Blackwell theorem, and not because it is a direct application of it.

Still, the motivation for the RBPF is to improve the accuracy of the filter, in a similar way as what we expect from Rao-Blackwellisation. That is, any estimator derived from the RBPF is believed have lower variance than the corresponding estimator derived from the standard PF. Informally, the reason for this is that in the RBPF, the particles are spread in a lower dimensional space, yielding a denser particle representation of the underlying distribution. This conjecture will be further investigated in Section 4.2.

The RBPF is most commonly used for conditionally linear Gaussian state-space (CLGSS) models (see Section 2.2.2). In this case, the conditional distribution $\Phi^c_{1:t|t}$ is Gaussian, and can be computed using the Kalman filter (KF) recursions. Consequently, the KF updates are often shown as intrinsic steps in the presentation of the RBPF algorithm, see e.g. [Schön et al., 2005]. We shall follow this example and derive an RBPF algorithm for mixed linear/nonlinear Gaussian state-space models (defined in Example 2.3 in Section 2.2.2). This is done to exemplify the derivation of an RBPF, but remember that the RBPF is not restricted to this type of model.

Recall the mixed linear/nonlinear Gaussian state-space model given by (2.9) on page 16,

$$\Xi_{t+1} = f^\xi(\Xi_t) + A^\xi(\Xi_t)Z_t + V^\xi_t, \tag{3.32a}$$

$$Z_{t+1} = f^z(\Xi_t) + A^z(\Xi_t)Z_t + V^z_t, \tag{3.32b}$$

$$Y_t = h(\Xi_t) + C(\Xi_t)Z_t + E_t. \tag{3.32c}$$

With state-space $(\mathbb{R}^{n_x}, \mathcal{B}(\mathbb{R}^{n_x}))$ and observation space $(\mathbb{R}^{n_y}, \mathcal{B}(\mathbb{R}^{n_y}))$, this model is fully dominated by Lebesgue measure. Hence, we shall do the derivation in terms of densities.

Furthermore, the RBPF is typically used to address the filtering problem, i.e. to approximate expectations of the form,

$$\begin{aligned}\mathrm{E}\left[\varphi(\Xi_t, Z_t) \mid Y_{1:t} = y_{1:t}\right] &= \iint \varphi(\xi_t, z_t) p(\xi_t, z_t \mid y_{1:t})\, d\xi_t dz_t \\ &= \iint \varphi(\xi_t, z_t) p(z_t \mid \xi_{1:t}, y_{1:t}) p(\xi_{1:t} \mid y_{1:t})\, d\xi_{1:t} dz_t, \quad (3.33)\end{aligned}$$

for some test function φ. Hence, the task at hand can be formulated as,

1. Target $p(\xi_{1:t} \mid y_{1:t})$ with an SMC sampler, generating a sequence of weighted particle systems $\{\xi_{1:t}^i, \omega_t^i\}_{i=1}^N$.
2. Sequentially compute the sufficient statistics for the densities $p(z_t \mid \xi_{1:t}^i, y_{1:t})$ for $i = 1, \ldots, N$.

3.3.1 Updating the linear states

We shall start the derivation of the RBPF by showing how to obtain the conditional filtering density $p(z_t \mid \xi_{1:t}, y_{1:t})$ sequentially. As already stated this density will turn out to be Gaussian, and we thus only need to keep track of its first and second moment. The updating formulas will show great resemblance with the Kalman filter, which is not surprising since the conditional process $\{Z_t \mid \Xi_{1:t}\}_{t \geq 1}$ obeys an LGSS model. The only trick is that, in the eyes of the linear Z-process, the evolution of the nonlinear process (3.32a) will behave as an extra "measurement" that we need to pay attention to. However, despite this similarity with the Kalman filer, we shall derive the updating formulas from basic principles. In the process we will (if you like) also derive the Kalman filter.

The derivation will be given as a proof by induction. By the end of this section we shall see that $p(z_1 \mid \xi_1, y_1)$ is Gaussian and can thus be written according to $p(z_1 \mid \xi_1, y_1) = \mathcal{N}(z_1; \bar{z}_{1|1}(\xi_1), P_{1|1}(\xi_1))$ where we have defined $\bar{z}_{1|1}(\xi_1)$ and $P_{1|1}(\xi_1)$ as the mean and covariance of the distribution, respectively. Hence, assume that, for $t \geq 2$,

$$p(z_{t-1} \mid \xi_{1:t-1}, y_{1:t-1}) = \mathcal{N}\left(z_{t-1}; \bar{z}_{t-1|t-1}(\xi_{1:t-1}), P_{t-1|t-1}(\xi_{1:t-1})\right), \quad (3.34)$$

where the mean and covariance are functions of the nonlinear state trajectory $\xi_{1:t-1}$ (naturally, they do also depend on the measurements $y_{1:t-1}$, but we shall not make that dependence explicit). We shall now see that this implies

$$p(z_t \mid \xi_{1:t}, y_{1:t}) = \mathcal{N}\left(z_t; \bar{z}_{t|t}(\xi_{1:t}), P_{t|t}(\xi_{1:t})\right) \quad (3.35)$$

and show how we can obtain the sufficient statistics for this distribution.

Using the Markov property and the state transition density given by the model (3.32), we have,

$$\begin{aligned}&p(z_t, \xi_t \mid z_{t-1}, \xi_{1:t-1}, y_{1:t-1}) = p(z_t, \xi_t \mid z_{t-1}, \xi_{t-1}) \\ &\quad = \mathcal{N}\Bigg(\underbrace{\begin{bmatrix} \xi_t \\ z_t \end{bmatrix}}_{=x_t}; \underbrace{\begin{bmatrix} f^\xi(\xi_{t-1}) \\ f^z(\xi_{t-1}) \end{bmatrix}}_{=f(\xi_{t-1})} + \underbrace{\begin{bmatrix} A^\xi(\xi_{t-1}) \\ A^z(\xi_{t-1}) \end{bmatrix}}_{=A(\xi_{t-1})} z_{t-1}, \underbrace{\begin{bmatrix} Q^\xi(\xi_{t-1}) & Q^{\xi z}(\xi_{t-1}) \\ (Q^{\xi z}(\xi_{t-1}))^\mathsf{T} & Q^z(\xi_{t-1}) \end{bmatrix}}_{=Q(\xi_{t-1})} \Bigg),\end{aligned} \quad (3.36)$$

which is affine in z_{t-1}. Since affine transformations preserves Gaussianity (see Corollary B.1 in Appendix B), by combining (3.34) and (3.36) we get,

$$p(z_t, \xi_t \mid \xi_{1:t-1}, y_{1:t-1}) \\ = \mathcal{N}\left(\begin{bmatrix} \xi_t \\ z_t \end{bmatrix}; \underbrace{\begin{bmatrix} \alpha_{t|t-1}(\xi_{1:t-1}) \\ \zeta_{t|t-1}(\xi_{1:t-1}) \end{bmatrix}}_{=\chi_{t|t-1}(\xi_{1:t-1})}, \underbrace{\begin{bmatrix} \Sigma^{\xi}_{t|t-1}(\xi_{1:t-1}) & \Sigma^{\xi z}_{t|t-1}(\xi_{1:t-1}) \\ (\Sigma^{\xi z}_{t|t-1}(\xi_{1:t-1}))^{\mathsf{T}} & \Sigma^{z}_{t|t-1}(\xi_{1:t-1}) \end{bmatrix}}_{=\Sigma_{t|t-1}(\xi_{1:t-1})}, \right), \tag{3.37a}$$

with

$$\chi_{t|t-1}(\xi_{1:t-1}) = f + A\bar{z}_{t-1|t-1}, \tag{3.37b}$$

$$\Sigma_{t|t-1}(\xi_{1:t-1}) = Q + AP_{t-1|t-1}A^{\mathsf{T}}; \tag{3.37c}$$

to keep the notation simple, the dependencies on ξ_{t-1} and $\xi_{1:t-1}$ have been dropped from the right hand side. This is simply a prediction of the state at time t, conditioned on $\xi_{1:t-1}$ and $y_{1:t-1}$. In (3.37b) the system dynamics is simulated and (3.37c) shows how the uncertainty in the prediction depends on the process noise and the prior uncertainty in the linear state.

By marginalisation of (3.37a) we obtain (Theorem B.1),

$$p(\xi_t \mid \xi_{1:t-1}, y_{1:t-1}) = \mathcal{N}\left(\xi_t; \alpha_{t|t-1}(\xi_{1:t-1}), \Sigma^{\xi}_{t|t-1}(\xi_{1:t-1})\right), \tag{3.38}$$

and by conditioning (3.37a) on ξ_t (Theorem B.2) we get,

$$p(z_t \mid \xi_{1:t}, y_{1:t-1}) = \mathcal{N}\left(z_t; \bar{z}_{t|t-1}(\xi_{1:t}), P_{t|t-1}(\xi_{1:t})\right), \tag{3.39a}$$

with

$$\bar{z}_{t|t-1}(\xi_{1:t}) = \zeta_{t|t-1} + L_t(\xi_t - \alpha_{t|t-1}), \tag{3.39b}$$

$$P_{t|t-1}(\xi_{1:t-1}) = \Sigma^{z}_{t|t-1} - L_t\Sigma^{\xi z}_{t|t-1}, \tag{3.39c}$$

$$L_t(\xi_{1:t-1}) = (\Sigma^{\xi z}_{t|t-1})^{\mathsf{T}}(\Sigma^{\xi}_{t|t-1})^{-1}. \tag{3.39d}$$

The above expressions constitute the time update of the filter. The prediction of the nonlinear state, which will be used during sampling (see Section 3.3.2), is given by (3.38). Once the nonlinear state trajectory is augmented with a new sample we can condition the prediction of the linear state on this sample, according to (3.39). In doing so we provide some information about the linear state, through the connection between the linear and the nonlinear parts of the state vector. From (3.39) we see that the estimate is updated accordingly and that the covariance is reduced. This update is very similar to a Kalman filter measurement update, and is therefore sometimes denoted the "extra measurement update" of the RBPF. However, note that we have not used any information about the current measurement y_t up to this point. This is what we will do next.

From the measurement likelihood given by model (3.32), and the conditional independence properties of the model, we have

$$p(y_t \mid \xi_{1:t}, z_t, y_{1:t-1}) = p(y_t \mid \xi_t, z_t) = \mathcal{N}\left(y_t; h(\xi_t) + C(\xi_t)z_t, R(\xi_t)\right), \tag{3.40}$$

which is affine in z_t. Again appealing to Corollary B.1 and using the result (3.39) we obtain the measurement prediction density,

$$p(y_t \mid \xi_{1:t}, y_{1:t-1}) = \mathcal{N}\left(y_t; \hat{y}_t(\xi_{1:t}), S_t(\xi_{1:t})\right), \tag{3.41a}$$

with

$$\hat{y}_t(\xi_{1:t}) = h + C\bar{z}_{t|t-1}, \tag{3.41b}$$

$$S_t(\xi_{1:t}) = R + CP_{t|t-1}C^{\mathsf{T}}, \tag{3.41c}$$

and also the posterior of z_t conditioned on y_t,

$$p(z_t \mid \xi_{1:t}, y_{1:t}) = \mathcal{N}\left(z_t; \bar{z}_{t|t}(\xi_{1:t}), P_{t|t}(\xi_{1:t})\right), \tag{3.42a}$$

with

$$\bar{z}_{t|t}(\xi_{1:t}) = \bar{z}_{t|t-1} + K_t(y_t - \hat{y}_t), \tag{3.42b}$$

$$P_{t|t}(\xi_{1:t}) = P_{t|t-1} - K_t C P_{t|t-1}, \tag{3.42c}$$

$$K_t(\xi_{1:t}) = P_{t|t-1}C^{\mathsf{T}}S_t^{-1}. \tag{3.42d}$$

Now, if we define $y_{1:0} \triangleq \emptyset$, so that $p(z_1 \mid \xi_{1:1}, y_{1:0}) = p(z_1 \mid \xi_1)$ and analogously for other densities, we see that the expression (3.39a) coincides with the prior (2.11) on page 17 at $t = 1$. The computations in (3.39) to (3.42) will thus hold at $t = 1$, which in turn confirms the validity of the induction assumption, $p(z_1 \mid \xi_1, y_1) = \mathcal{N}(z_1; \bar{z}_{1|1}(\xi_1), P_{1|1}(\xi_1))$.

3.3.2 Sampling nonlinear state trajectories

In the previous section, we obtained closed form expressions for the conditional filtering density for the linear state, $p(z_t \mid \xi_{1:t}, y_{1:t})$. However, it remains to find the state-marginal smoothing density, $p(\xi_{1:t} \mid y_{1:t})$. Due to the nonlinear dependence on Ξ_t in the model (3.32), this density is not available in closed form. Instead, we target it with an SMC sampler.

Let us assume that $t \geq 2$. Sampling at time $t = 1$ can be done by straightforward modifications of the results given here. First, using Bayes' rule we note the following about the target density,

$$\begin{aligned} p(\xi_{1:t} \mid y_{1:t}) &\propto p(y_t \mid \xi_{1:t}, y_{1:t-1})p(\xi_{1:t} \mid y_{1:t-1}) \\ &= p(y_t \mid \xi_{1:t}, y_{1:t-1})p(\xi_t \mid \xi_{1:t-1}, y_{1:t-1})p(\xi_{1:t-1} \mid y_{1:t-1}). \end{aligned} \tag{3.43}$$

Second, similarly to (3.10) we choose a proposal density which factorises according to,

$$r'_t(\xi_{1:t} \mid y_{1:t}) = r_t(\xi_t \mid \xi_{1:t-1}, y_{1:t})\underbrace{r'_{t-1}(\xi_{1:t-1} \mid y_{1:t-1})}_{\text{previous proposal}}. \tag{3.44}$$

Given a weighted particle system $\{\tilde{\xi}^i_{1:t-1}, \tilde{\omega}^i_{t-1}\}_{i=1}^N$ targeting $p(\xi_{1:t-1} \mid y_{1:t-1})$, sample trajectories are constructed as in (3.11),

$$\xi^i_t \sim r_t(\xi_t \mid \tilde{\xi}^i_{1:t-1}, y_{1:t}), \tag{3.45a}$$

$$\xi^i_{1:t} := \{\tilde{\xi}^i_{1:t-1}, \xi^i_t\}, \tag{3.45b}$$

for $i = 1, \dots, N$. Using (3.38) and (3.41), the importance weights are then given by,

$$\omega_t^i = \frac{p(\xi_{1:t}^i \mid y_{1:t})}{r_t'(\xi_{1:t}^i \mid y_{1:t})} \propto \frac{p(y_t \mid \xi_{1:t}^i, y_{1:t-1}) p(\xi_t^i \mid \xi_{1:t-1}^i, y_{1:t-1})}{r_t(\xi_t^i \mid \xi_{1:t-1}^i, y_{1:t})} \underbrace{\frac{p(\xi_{1:t-1}^i \mid y_{1:t-1})}{r_{t-1}'(\xi_{1:t-1}^i \mid y_{1:t-1})}}_{=\tilde{\omega}_{t-1}^i}$$

$$= \frac{\mathcal{N}\big(y_t; \hat{y}_t(\xi_{1:t}^i), S_t(\xi_{1:t}^i)\big) \mathcal{N}\big(\xi_t^i; \alpha_{t|t-1}(\xi_{1:t-1}^i), \Sigma_{t|t-1}^{\xi}(\xi_{1:t-1}^i)\big)}{r_t(\xi_t^i \mid \xi_{1:t-1}^i, y_{1:t})} \tilde{\omega}_{t-1}^i. \tag{3.46}$$

Since we only know the weights up to proportionality, they are normalised to sum to one. Finally, a selection strategy (e.g. resampling) should be applied to the RBPF just as for the regular PF.

3.3.3 RBPF algorithm

We summarise the RBPF in Algorithm 3.3. To simplify the notation; for functions in argument ξ_t or $\xi_{1:t}$, e.g. $R(\xi_t)$ and $\bar{z}_{t|t}(\xi_{1:t})$, we write $R_t^i \triangleq R(\xi_t^i)$ and $\bar{z}_{t|t}^i \triangleq \bar{z}_{t|t}(\xi_{1:t}^i)$, etc. We also make the following definition of an augmented weighted particle system.

Definition 3.3 (Augmented weighted particle system). An augmented weighted particle system targeting a factorised density $p(z \mid \xi)p(\xi)$, is a collection of quadruples $\{\xi^i, \omega^i, \bar{z}^i, P^i\}_{i=1}^N$ s.t.,

i) $\{\xi^i, \omega^i\}_{i=1}^N$ is a weighted particle system targeting $p(\xi)$.

ii) The conditional density $p(z \mid \xi)$ is Gaussian, with $p(z \mid \xi^i) = \mathcal{N}(z; \bar{z}^i, P^i)$ for $i = 1, \dots, N$.

Furthermore, in the interest of giving a somewhat more compact presentation, the algorithm is only given for time $t \geq 2$. Initialisation at time $t = 1$ can be done by straightforward modifications of the steps of the algorithm.

3.3.4 Proposal construction by local linearisation

Choosing a proposal kernel for the RBPF can be done similarly as for the PF. However, there are some differences. To start with, since the nonlinear state process is not necessarily Markovian, sampling from the transition kernel might not be an option in the RBPF. One alternative is to sample from the 1-step predictive distribution for the nonlinear process. For a mixed linear/nonlinear Gaussian state-space model, this distribution is Gaussian and given by (3.38). For this choice of proposal, we get a cancellation in the weight expression (3.46), just as for the bootstrap PF (see Definition 3.2). Hence, the 1-step predictive distribution can be seen as an analogue of the bootstrap kernel in the PF. We thus make the following definition.

Definition 3.4 (Bootstrap RBPF). The bootstrap RBPF for mixed linear/nonlinear Gaussian state-space models is an RBPF according to Algorithm 3.3 in which,

i) the 1-step predictive distribution for the nonlinear process, given by (3.38), is used as proposal, i.e. $r_t(\xi_t \mid \xi_{1:t-1}, y_{1:t}) = p(\xi_t \mid \xi_{1:t-1}, y_{1:t-1})$.

ii) selection is done by multinomial resampling at each iteration of the algorithm.

Algorithm 3.3 RBPF for mixed linear/nonlinear Gaussian state-space models

```
Input:  An augmented weighted particle system $\{\xi^i_{1:t-1}, \omega^i_{t-1}, \bar{z}^i_{t-1|t-1}, P^i_{t-1|t-1}\}_{i=1}^N$,
        targeting $p(z_{t-1} \mid \xi_{1:t-1}, y_{1:t-1})p(\xi_{1:t-1} \mid y_{1:t-1})$.
Output: An augmented weighted particle system $\{\xi^i_{1:t}, \omega^i_t, \bar{z}^i_{t|t}, P^i_{t|t}\}_{i=1}^N$, targeting
        $p(z_t \mid \xi_{1:t}, y_{1:t})p(\xi_{1:t} \mid y_{1:t})$.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Selection:
  1: Optionally, generate a new weighted particle system $\{\tilde{\xi}^i_{1:t-1}, \tilde{\omega}^i_{t-1}\}_{i=1}^N$ by selection,
     or set $\{\tilde{\xi}^i_{1:t-1}, \tilde{\omega}^i_{t-1}\}_{i=1}^N = \{\xi^i_{1:t-1}, \xi^i_{t-1}\}_{i=1}^N$.
Sampling:
  2: Augment the sample trajectories. For $i = 1, \ldots, N$,
         $\xi^i_t \sim r_t(\xi_t \mid \tilde{\xi}^i_{1:t-1}, y_{1:t})$,
         $\xi^i_{1:t} = \{\tilde{\xi}^i_{1:t-1}, \xi^i_t\}$.
Prediction:
  3: Predict the state and condition the linear state on the newly drawn particles $\{\xi^i_t\}_{i=1}^N$.
     For $i = 1, \ldots, N$,
         $\alpha^i_{t|t-1} = f^{\xi,i}_{t-1} + A^{\xi,i}_{t-1}\bar{z}^i_{t-1|t-1}$,
         $\bar{z}^i_{t|t-1} = f^{z,i}_{t-1} + A^{z,i}_{t-1}\bar{z}^i_{t-1|t-1} + (\Sigma^{\xi z,i}_{t|t-1})^{\mathsf{T}}(\Sigma^{\xi,i}_{t|t-1})^{-1}(\xi^i_t - \alpha^i_{t|t-1})$,
         $P^i_{t|t-1} = \Sigma^{z,i}_{t|t-1} - (\Sigma^{\xi z,i}_{t|t-1})^{\mathsf{T}}(\Sigma^{\xi,i}_{t|t-1})^{-1}(\Sigma^{\xi z,i}_{t|t-1})$,
     with
         $\Sigma^i_{t|t-1} = Q^i_{t-1} + A^i_{t-1}P^i_{t-1|t-1}(A^i_{t-1})^{\mathsf{T}}$.
Weighting:
  4: Evaluate the unnormalised importance weights. For $i = 1, \ldots, N$,
         $\omega'^{\,i}_t = \dfrac{\mathcal{N}\big(y_t; \hat{y}^i_t, S^i_t\big)\mathcal{N}\big(\xi^i_t; \alpha^i_{t|t-1}, \Sigma^{\xi,i}_{t|t-1}\big)}{r_t(\xi^i_t \mid \xi^i_{1:t-1}, y_{1:t})}\tilde{\omega}^i_{t-1}$,
     with
         $\hat{y}^i_t = h^i_t + C^i_t\bar{z}^i_{t|t-1}$,
         $S^i_t = R^i_t + C^i_t P^i_{t|t-1}(C^i_t)^{\mathsf{T}}$.
  5: Normalise the importance weights. For $i = 1, \ldots, N$, set $\omega^i_t = \omega'^{\,i}_t / \sum_k \omega'^{\,k}_t$.
Update the linear states:
  6: Compute the sufficient statistics for the linear states, given the current measurement.
     For $i = 1, \ldots, N$,
         $\bar{z}^i_{t|t} = \bar{z}^i_{t|t-1} + K^i_t(y_t - \hat{y}^i_t)$,
         $P^i_{t|t} = P^i_{t|t-1} - K^i_t C^i_t P^i_{t|t-1}$,
         $K^i_t = P^i_{t|t-1}(C^i_t)^{\mathsf{T}}(S^i_t)^{-1}$.
```

However, as pointed out in Section 3.2.1, the bootstrap kernel might not be the most suitable choice in practice. The reason is that the current measurement y_t is not taken into account when proposing particles at time t. Due to this, we will now consider an alternative way of constructing a proposal kernel for mixed linear/nonlinear Gaussian state-space models. The construction is based on a local linearisation of the measurement equation (3.32c). The same technique has been used by Doucet et al. [2000b] to construct a proposal kernel for the PF. To start with, note that for some matrix valued function $M : \mathbb{R}^m \mapsto \mathbb{R}^{p \times n}$,

$$M(x) = \begin{pmatrix} M_1(x) \\ \vdots \\ M_p(x) \end{pmatrix}, \tag{3.47}$$

and a constant n-vector v, the Jacobian matrix of the function $M(x)v$ is given by,

$$\frac{\partial (M(\,\cdot\,)v)}{\partial x} = \begin{pmatrix} v^{\mathsf{T}} J_{M_1^{\mathsf{T}}}(x) \\ \vdots \\ v^{\mathsf{T}} J_{M_p^{\mathsf{T}}}(x), \end{pmatrix} \tag{3.48}$$

where $J_{M_i^{\mathsf{T}}}$ is the Jacobian matrix of the function $M_i^{\mathsf{T}} : \mathbb{R}^m \to \mathbb{R}^n$.

Now, consider a first order Taylor expansion of the measurement equation (3.32c), around some point $\bar{x}_t = \{\bar{\xi}_t, \bar{z}_t\}$. Observe that we need to linearise the measurement function in the z_t-direction as well, to get rid of the cross terms between ξ_t and z_t. We thus have,

$$\begin{aligned} h(\xi_t) + C(\xi_t)z_t &\approx h(\bar{\xi}_t) + C(\bar{\xi}_t)\bar{z}_t + \left[J_h(\bar{\xi}_t) + \Gamma(\bar{\xi}_t, \bar{z}_t)\right](\xi_t - \bar{\xi}_t) + C(\bar{\xi}_t)(z_t - \bar{z}_t) \\ &= h(\bar{\xi}_t) + \left[J_h(\bar{\xi}_t) + \Gamma(\bar{\xi}_t, \bar{z}_t)\right](\xi_t - \bar{\xi}_t) + C(\bar{\xi}_t)z_t, \end{aligned} \tag{3.49a}$$

where we have defined,

$$\Gamma(\bar{\xi}_t, \bar{z}_t) \triangleq \frac{\partial (C(\,\cdot\,)\bar{z}_t)}{\partial \xi_t}_{|\xi_t = \bar{\xi}_t} = \begin{pmatrix} \bar{z}_t^{\mathsf{T}} J_{C_1^{\mathsf{T}}}(\bar{\xi}_t) \\ \vdots \\ \bar{z}_t^{\mathsf{T}} J_{C_{n_y}^{\mathsf{T}}}(\bar{\xi}_t) \end{pmatrix}. \tag{3.49b}$$

From (3.32c) and (3.49) we have that,

$$Y_t \approx h(\bar{\xi}_t) + \left[J_h(\bar{\xi}_t) + \Gamma(\bar{\xi}_t, \bar{z}_t)\right](\Xi_t - \bar{\xi}_t) + C(\bar{\xi}_t)Z_t + E_t. \tag{3.50}$$

Hence, we have that the density function $p(y_t \mid \xi_{1:t}, z_t, y_{1:t-1}) = p(y_t \mid \xi_t, z_t)$ is approximately Gaussian and affine in $x_t = \{\xi_t, z_t\}$. In fact, this density is Gaussian and affine in z_t, but the linearisation has the effect of removing any cross terms between ξ_t and z_t.

From (3.39a) we have that $Z_t \mid \{\Xi_{1:t}, Y_{1:t-1}\}$ is Gaussian. Furthermore, let us assume that the measurement noise covariance R is independent of Ξ_t (recall from (2.10b) on page 16 that we otherwise allow for a dependence). Then, (3.50) is an affine transformation of a Gaussian variable. Hence, we get an approximate Gaussian density for Y_t, according to,

$$p(y_t \mid \xi_{1:t}, y_{1:t-1}) \approx \mathcal{N}(y_t; \hat{y}_t'(\xi_{1:t}), S_t'(\xi_{1:t-1})), \tag{3.51a}$$

with,

$$\hat{y}'_t(\xi_{1:t}) = h(\bar{\xi}_t) + \left[J_h(\bar{\xi}_t) + \Gamma(\bar{\xi}_t, \bar{z}_t)\right](\xi_t - \bar{\xi}_t) + C(\bar{\xi}_t)\bar{z}_{t|t-1}(\xi_{1:t}), \tag{3.51b}$$

$$S'_t(\xi_{1:t-1}) = R + C(\bar{\xi}_t)P_{t|t-1}(\xi_{1:t-1})C(\bar{\xi}_t)^{\mathsf{T}}. \tag{3.51c}$$

From (3.39b) we see that the last term in (3.51b) has an affine dependence on ξ_t, which is then also the case for $\hat{y}'_t(\xi_{1:t})$. We can thus write,

$$\hat{y}'_t(\xi_{1:t}) = H_t\xi_t + d_t, \tag{3.52a}$$

$$H_t(\xi_{1:t-1}) \triangleq J_h(\bar{\xi}_t) + \Gamma(\bar{\xi}_t, \bar{z}_t) + C(\bar{\xi}_t)L_t, \tag{3.52b}$$

$$d_t(\xi_{1:t-1}) \triangleq h(\bar{\xi}_t) - \left[J_h(\bar{\xi}_t) + \Gamma(\bar{\xi}_t, \bar{z}_t)\right]\bar{\xi}_t + C(\bar{\xi}_t)\left[\zeta_{t|t-1} - L_t\alpha_{t|t-1}\right]. \tag{3.52c}$$

Finally, (3.38) and (3.51) is again (approximately) an affine transformation of a Gaussian variable (this time Ξ_t). Hence, we get

$$p(\xi_t \mid \xi_{1:t-1}, y_{1:t}) \approx \mathcal{N}(\xi_t; m_t(\xi_{1:t-1}), \Pi_t(\xi_{1:t-1})), \tag{3.53a}$$

with

$$\begin{aligned} m_t(\xi_{1:t-1}) &= \Pi_t(H_t^{\mathsf{T}}(S'_t)^{-1}(y_t - d_t) + (\Sigma^{\xi}_{t|t-1})^{-1}\alpha_{t|t-1}) \\ &= \alpha_{t|t-1} + \Sigma^{\xi}_{t|t-1}H_t^{\mathsf{T}}(S'_t + H_t\Sigma^{\xi}_{t|t-1}H_t^{\mathsf{T}})^{-1}(y_t - d_t - H_t\alpha_{t|t-1}), \end{aligned} \tag{3.53b}$$

$$\begin{aligned} \Pi_t(\xi_{1:t-1}) &= \left[(\Sigma^{\xi}_{t|t-1})^{-1} + H_t^{\mathsf{T}}(S'_t)^{-1}H_t\right]^{-1} \\ &= \Sigma^{\xi}_{t|t-1} - \Sigma^{\xi}_{t|t-1}H_t^{\mathsf{T}}(S'_t + H_t\Sigma^{\xi}_{t|t-1}H_t^{\mathsf{T}})^{-1}H_t\Sigma^{\xi}_{t|t-1}. \end{aligned} \tag{3.53c}$$

A natural choice of linearisation point is the 1-step prediction, i.e. $\{\bar{\xi}_t, \bar{z}_t\} = \{\alpha_{t|t-1}, \zeta_{t|t-1}\}$. For this choice, the expression (3.52c) reduces to

$$d_t(\xi_{1:t-1}) = h(\alpha_{t|t-1}) + C(\alpha_{t|t-1})\zeta_{t|t-1} - H_t\alpha_{t|t-1}. \tag{3.54a}$$

This further implies that, in (3.53b), we get

$$y_t - d_t - H_t\alpha_{t|t-1} = y_t - h(\alpha_{t|t-1}) - C(\alpha_{t|t-1})\zeta_{t|t-1}. \tag{3.54b}$$

Guided by the approximation of the optimal proposal in (3.53a), we can use as proposal density,

$$r_t(\xi_t \mid \xi_{1:t-1}, y_{1:t}) = \mathcal{N}(\xi_t; m_t(\xi_{1:t-1}), \Pi_t(\xi_{1:t-1})). \tag{3.55}$$

The same density is then also used to compute the weights according to (3.46).

3.3.5 Application example: RBPF for UAV localisation

A numerical illustration of how the the RBPF performs in comparison to the PF and the KF is postponed to Section 5.3.5, where we at the same time consider the smoothing problem. However, in this section we will illustrate the use of the RBPF in an application example. The problem that we will consider here is unmanned aerial vehicle (UAV) localisation through visual odometry (VO) and geo-referencing (GR). The present section is a short summary of the material previously published in [Lindsten et al., 2010].

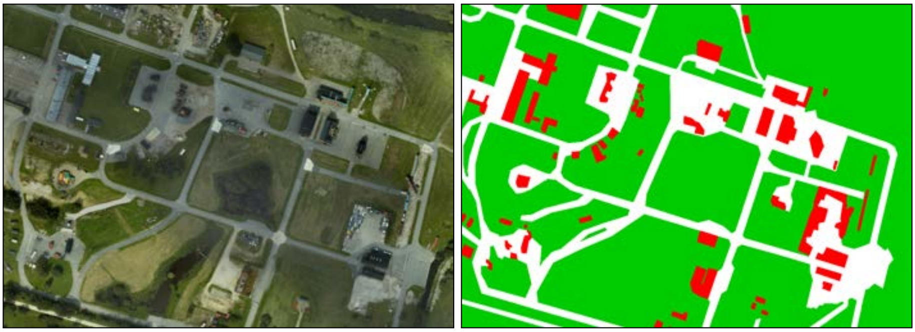

Figure 3.2: *Map over the operational environment (left) and a manually classified reference map with grass, asphalt and houses as prespecified classes (right). Aerial photograph by courtesy of the UAS Technologies Lab, Artificial Intelligence and Integrated Computer Systems Division (AIICS) at the Department of Computer and Information Science (IDA), Linköping University, Linköping, Sweden.*

The work is motivated by the fact that navigation of commercial UAVs today is depending on global navigation satellite systems, e.g. GPS. However, to solely rely on GPS is associated with a risk. When operating close to obstacles, reflections can make the GPS signal unreliable. It is also easy to jam the GPS making it vulnerable to malicious attacks. Due to the possibility of signal failure, a drift free backup system might be necessary.

As shown by Törnqvist et al. [2009], a sensory setup using an inertial measurement unit (IMU) together with vision from an on-board camera enables accurate pose estimates through the process of VO fused with the IMU measurements. In VO, a set of distinct landmarks are tracked between consecutive camera frames. By assuming that the landmarks are static, this provides a measurement of the velocity of the own vehicle. However, without any absolute position reference the estimated position of the UAV will always suffer from a drift.

In [Lindsten et al., 2010] we proposed to use an existing, preclassified map over the operational environment, as shown in Figure 3.2. By matching the images from the on-board camera with this map, we obtain an additional measurement which can be used to remove the drift.

Without going into any details (these can can found in [Lindsten et al., 2010]), the matching is done in the following way. First, once we obtain an image from the onboard camera, we use an image segmentation technique by Felzenszwalb and Huttenlocher [2004] to divide the image into uniform regions, called *superpixels*. For each superpixel, we then extract a descriptor, consisting of color and texture information. These descriptors are then fed into a trained neural network classifier. The output from the classifier is an assignment of each superpixel into one of three prespecified classes; grass, asphalt and buildings.

Once we have classified the image into its environmental content, we need to match it with the preclassified map of the operational environment. As argued in [Lindsten et al., 2010],

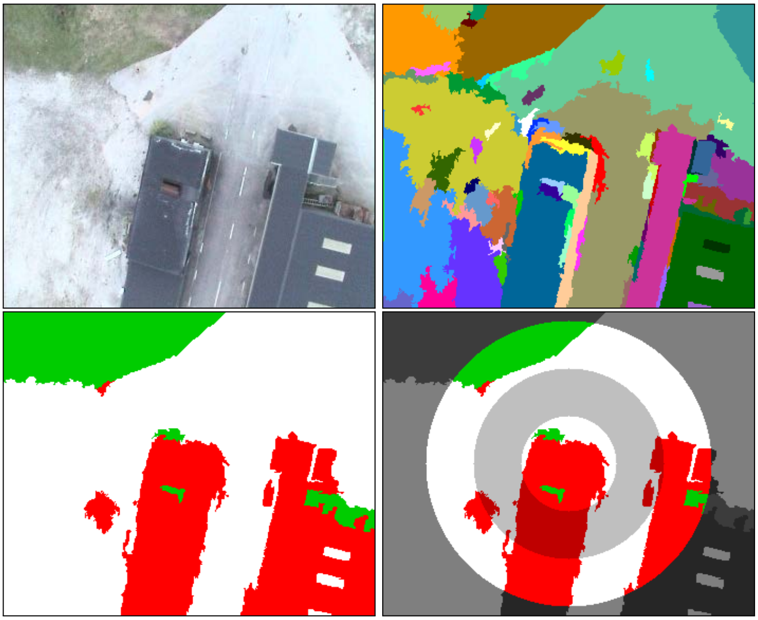

Figure 3.3: *Image from on-board camera (top-left), extracted superpixels (top-right), superpixels classified as grass, asphalt or house (bottom-left) and three circular regions used for computing the class histograms (bottom-right).*

to avoid introducing instability in the observer, we seek a measurement which is invariant to the rotation of the image. To attain this, we extract several circular regions from the image. For each such region we then compute a class histogram, i.e. the proportions of the three different classes in the region. The above described procedure is illustrated in Figure 3.3.

Finally, the class histograms are matched with class histograms computed from the map. To evaluate the likelihood that the vehicle is located at a certain horisontal position, this position is projected onto the map. The class histograms from the corresponding point in the map are thereafter compared with the class histograms extracted from the on-board camera image. If there is a good resemblance, the likelihood is high and vice versa. As shown in [Lindsten et al., 2010], this relationship can be expressed as a nonlinear measurement equation,

$$Y_{\mathrm{GR},t} = h_{\mathrm{GR}}(X_{p,t}) + E_{\mathrm{GR},t}, \tag{3.56}$$

where the measurement $Y_{\mathrm{GR},t}$ consist of the class histograms extracted from the on-board image, the function h_{GR} is given as a look-up table, related to the reference map, and the measurement noise $E_{\mathrm{GR},t}$ is modelled as zero-mean Gaussian with a time-varying covariance, related to the classification uncertainty.

Using the reference map shown in Figure 3.2 and the classification result from Figure 3.3, the resulting likelihood evaluated over the entire map is illustrated in Figure 3.4. We

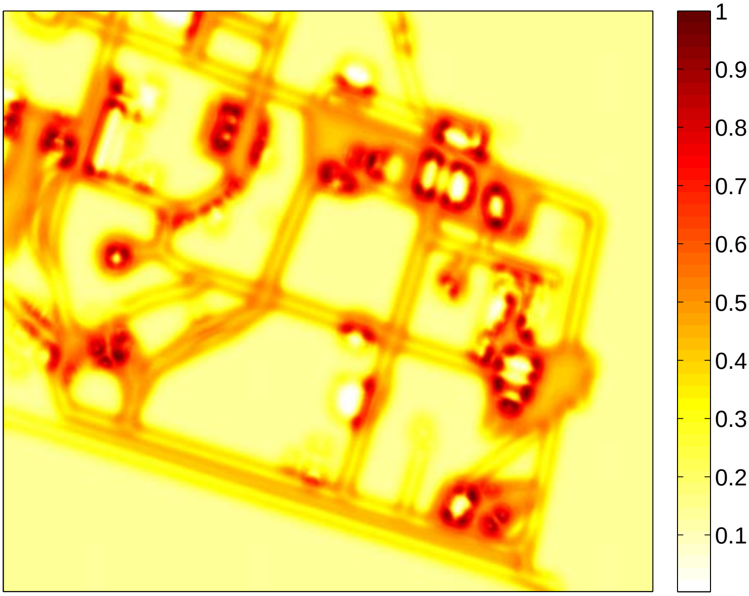

Figure 3.4: *Computed likelihood over the reference map.*

see that the likelihood is high in regions where we have both asphalt and houses in the reference map, since this is the case for the classified image. Along the roads, where the reference map consists of asphalt and grass but no houses, the likelihood is lower but still significantly above zero. This is desired, since the houses in the on-board image very well could be incorrectly classified. Finally, in regions where the reference map solely consists of grass, the matching is very poor and the likelihood is close to zero.

Combining the geo-reference measurement (3.56) with the VO framework used by Törnqvist et al. [2009], we end up with the following dynamic model, used in the navigation system of the UAV. The vehicle state consists of position $X_{p,t}$, velocity $X_{v,t}$, acceleration $X_{a,t}$, a quaternion $X_{q,t}$ representing the orientation of the UAV and its angular velocity $X_{\omega,t}$. The state vector is also augmented with bias states for acceleration $B_{a,t}$ and angular velocity $B_{\omega,t}$ to account for sensor imperfections.

The dynamic model turns out to be mixed linear/nonlinear Gaussian, enabling the application of the RBPF. Hence, the state vector is divided into nonlinear states Ξ_t and linear states Z_t,

$$\Xi_t = \begin{bmatrix} X_{p,t}^{\mathsf{T}} & X_{q,t}^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}, \tag{3.57a}$$

$$Z_t = \begin{bmatrix} X_{v,t}^{\mathsf{T}} & X_{a,t}^{\mathsf{T}} & B_{\omega,t}^{\mathsf{T}} & B_{a,t}^{\mathsf{T}} & X_{\omega,t}^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}. \tag{3.57b}$$

VO is incorporated into the estimation problem by tracking a set of landmarks $M_t = \{M_{j,t}\}_{j=1}^{J_t}$ in consecutive frames. The landmark positions in an absolute coordinate system are included in the linear part of the state vector. In summary, the dynamic model of the system is given by,

$$\Xi_{t+1} = f^{\xi}(\Xi_t) + A^{\xi}(\Xi_t)Z_t + G^{\xi}(\Xi_t)V_t^{\xi}, \tag{3.58a}$$

$$Z_{t+1} = A^{z}(Z_t)Z_t + G^{z}(\Xi_t)V_t^{z}, \tag{3.58b}$$

$$M_{j,t+1} = M_{j,t}, \qquad j = 1, \ldots, J_t, \tag{3.58c}$$

where the process noises V_t^{ξ} and V_t^{z} are assumed white and Gaussian with zero means.

The landmarks are initiated from distinct Harris corners in the on-board images and tracked between frames using normalised cross correlation. This gives rise to a measurement available at 4 Hz (the image frequency), given by

$$Y_{\mathrm{VO},t} = h_{\mathrm{VO}}(\Xi_t) + C_{\mathrm{VO}}(\Xi_t)M_t + E_{\mathrm{VO},t}. \tag{3.59}$$

The vehicle is also equipped with an IMU and a barometric sensor, operating at 20 Hz, yielding a second measurement,

$$Y_{\mathrm{IMU},t} = h_{\mathrm{IMU}}(\Xi_t) + C_{\mathrm{IMU}}(\Xi_t)Z_t + E_{\mathrm{IMU},t}. \tag{3.60}$$

The measurement noises $E_{\mathrm{VO},t}$ and $E_{\mathrm{IMU},t}$ are assumed white and Gaussian with zero means. Finally, we have a third measurement, also available at 4 Hz, from the geo-referencing according to (3.56).

Before we continue with experimental result, we note that the model described above has several properties making the RBPF a suitable choice of filter. First, the model is highly nonlinear, especially through the geo-reference measurement (3.56). Also, this measurement model is available only as a look-up table. The model can be evaluated pointwise, but linearisation can be quite problematic. This rules out deterministic filters such as the extended KF. On the contrary, any particle based method can straightforwardly handle the measurement (3.56). However, a standard PF will suffer from the high dimensionality of the state-space. At any given time t, the state vector has dimension[1] $21 + 3J_t$ where J_t is the number of landmarks tracked at time t. However, all but six of these "states" are conditionally linear Gaussian. Hence, by using the RBPF, we only need to spread the particles on a six-dimensional manifold embedded in the full, high-dimensional state-space.

Now, to test the UAV localisation system, data was collected during a 400 m test flight in southern Sweden, using an unmanned Yamaha RMAX helicopter. Figure 3.5 shows a map over the area with the UAVs true flight trajectory (a Kalman filtered GPS signal) illustrated with circles. Without using the GPS signal, we employ a bootstrap RBPF with $N = 500$ particles to estimate the vehicle position. First, we do not make use of the geo-reference measurement, i.e. we use the VO solution by Törnqvist et al. [2009]. The result is plotted as a dashed line in the figure. We can see that the estimate is fairly accurate, but as expected it suffers from a drift. In the same plot, also the solution using both VO and geo-referencing is shown as a solid line. The estimated trajectory in this case is very close to the ground truth, and much of the drift has been removed.

[1] When counting the dimension of the state-space we note that the quaternion, though represented by four numbers, belongs to the special orthogonal group $\mathsf{SO}(3)$ and is thus three-dimensional.

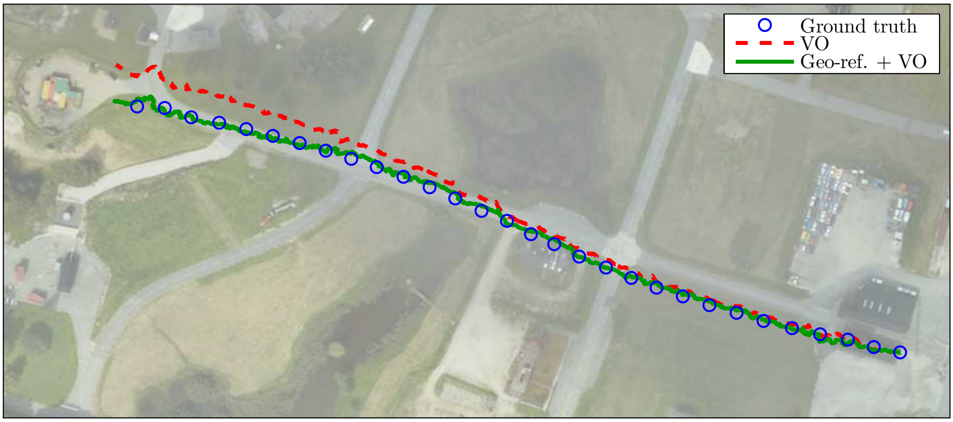

Figure 3.5: *True trajectory illustrated with circles and the estimated trajectories with (solid line) and without (dashed line) geo-referencing.*

3.4 Rao-Blackwellised marginal particle filter

We shall continue to study the RBPF of the previous section for CLGSS models. As previously mentioned, the RBPF is typically used to address the filtering problem. To be able to exploit the conditional tractability of the model, the filtering density is, in accordance with (3.33), expressed as,

$$p(x_t \mid y_{1:t}) = \int p(z_t \mid \xi_{1:t}, y_{1:t}) p(\xi_{1:t} \mid y_{1:t})\, d\xi_{1:t-1}. \tag{3.61}$$

From (3.35), we have that the first factor of the integrand is Gaussian and analytically tractable. We also note (the important fact) that the mean and the covariance of this Gaussian are functions of the nonlinear state trajectory $\xi_{1:t}$. The second factor of the integrand, i.e. the state-marginal smoothing density $p(\xi_{1:t} \mid y_{1:t})$, is targeted using an SMC sampler. If this yields a good approximation of the state-marginal smoothing distribution, (3.61) provides a way to approximate the filtering distribution.

However, a problem with this approach is that it is often not straightforward to obtain good approximations of the state-marginal smoothing distribution for large t, due to the degeneracy problem discussed in Section 3.2.2. To get around this problem, one often relies on the mixing properties of the system. More precisely, the state Z_t is supposed to be more or less independent of $\{\Xi_s,\ s \leq t - \ell\}$ for some lag ℓ. In that case, we only need to keep track of the fixed-lag, state-marginal smoothing density $p(\xi_{t-\ell+1:t} \mid y_{1:t})$. As pointed out in Section 3.2.2, this density can be readily approximated using an SMC sampler, as long as the lag is not too large.

Clearly, the success of this approach heavily depends on how good the mixing assumption is. If the system is slowly mixing, e.g. if the Z-process contains some static parameter, the dependence of $Z_t \mid \{\Xi_{t-\ell+1:t}, Y_{1:t}\}$ on $\{\Xi_s,\ s \leq t - \ell\}$ can be non-negligible. That is, if the approximation of the density $p(\xi_{1:t-\ell} \mid y_{1:t})$ is poor, using (3.61) to approximate the filtering distribution can give very poor results. The issue is illustrated in Example 3.2.

Example 3.2: RBPF for a partially static system

The first order LGSS system,

$$\Xi_{t+1} = a\Xi_t + V_t, \tag{3.62a}$$

$$Y_t = \Xi_t + E_t, \tag{3.62b}$$

is simulated for $T = 10000$ time steps. The system parameter a has the value -0.8. The process and measurement noises V_t and E_t, are mutually independent sequences of zero-mean Gaussian variables with variances 0.1 and 1, respectively. The initial distribution for the state process is zero-mean Gaussian with variance 0.1.

Now, assume that a is an unknown parameter. It can then be incorporated into the state vector, and a mixed linear/nonlinear Gaussian state-space model for the system is given by,

$$\Xi_{t+1} = \Xi_t Z_t + V_t, \tag{3.63a}$$

$$Z_{t+1} = Z_t, \tag{3.63b}$$

$$Y_t = \Xi_t + E_t. \tag{3.63c}$$

The parameter a is modeled as a Gaussian random variable, by assigning an initial distribution to the linear state process, $Z_1 \sim \mathcal{N}(1, 3)$. A bootstrap RBPF using $N = 100$ particles is applied to the data. The estimate of the state Z_t, together with the estimated 3σ-confidence intervals, are shown in Figure 3.6.

The initial distribution for Z_1 should have negligible effect at time $T = 10000$. Hence, we expect the estimate to converge to the "true" value -0.8. As can be seen in the figure, this is not the case. Also, the estimated confidence interval is way too small, i.e. the covariance of the linear state is underestimated. The intuitive explanation is that the RBPF sample trajectories, in some sense, degenerate faster than the estimate of Z_t converges.

We note that the signal to noise ratio (SNR) in this particular example is fairly poor (the measurement noise variance is a factor ten times the process noise variance). Hence, the convergence of the estimate is expected to be slow; a large amount of measurement data is needed to make accurate state inference. Based on this argument, we thus expect that the use of the RBPF for state estimation in slowly mixing systems, is particularly problematic when the SNR is low.

Here, we propose an alternative to (3.61), which is to factorise the filtering density as,

$$p(x_t \mid y_{1:t}) = p(z_t \mid \xi_t, y_{1:t})p(\xi_t \mid y_{1:t}). \tag{3.64}$$

The marginal filtering density $p(\xi_t \mid y_{1:t})$ can be approximated using SMC without suffering from degeneracy. Thus, an approximation of the filtering distribution based on the factorisation (3.64), does not rely on the mixing properties of the system. However, as opposed to $p(z_t \mid \xi_{1:t}, y_{1:t})$ given in (3.35), the density

$$p(z_t \mid \xi_t, y_{1:t}), \tag{3.65}$$

is in general non-Gaussian and intractable. The problem we face is thus to find a good approximation of (3.65), while still enjoying the benefits of a Rao-Blackwellised setting. The resulting filter will be denoted the Rao-Blackwellised marginal particle filter (RBMPF).

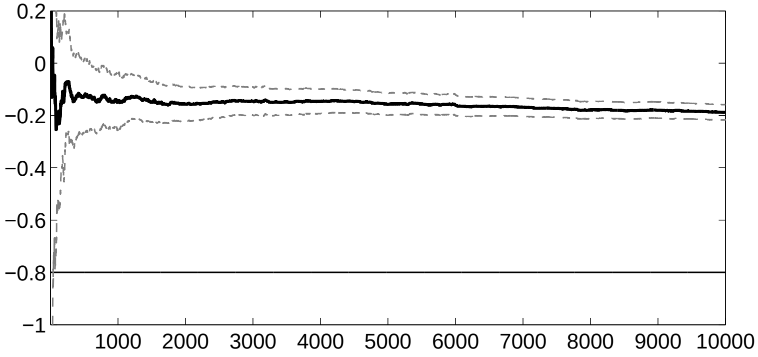

Figure 3.6: *RBPF estimate of Z_t (thick black line) and the estimated 3σ-confidence interval (grey dotted lines), as function of time t. The "true" value is -0.8.*

Before we go on with the details of this approximation, it should be mentioned that there have been previous contributions in this direction. Both Jianjun et al. [2007] and Smal et al. [2007] have proposed filters under the name RBMPF (or strictly speaking, MRBPF in the former case). However, these are both focused on combining the RBPF with the MPF by Klaas et al. [2005] (see also Section 3.2.4), without addressing the problems arising for slowly mixing systems. In the terminology introduced below, they can be seen as combining marginal sampling (Alternative S2) to sample from the marginal filtering distribution, with ancestral dependence (Alternative G1) for approximating the conditional filtering distribution. We comment further on this in Section 3.4.3.

We start the presentation of the RBMPF with a discussion on how to sample from the marginal filtering distribution. This material is more or less a restatement of the PF and the MPF approaches to sampling from the filtering distribution, but with the marginal filtering distribution in focus. The reason for providing the discussion here, is to allow for a more thorough treatment of the different approaches to the RBMPF. After this, we turn to the more central problem of approximating the conditional filtering density (3.65).

3.4.1 Sampling from the marginal

In the RBMPF we seek to target the marginal filtering density $p(\xi_t \mid y_{1:t})$ with an SMC sampler, as indicated by (3.64). Here, we mention two different alternatives to do this. The first is analogous to the PF, as we draw sample trajectories and simply discard the history of these trajectories. The second is analogous to the MPF (see Section 3.2.4), in the sense that we target the marginal filtering distribution "directly".

Alternative S1: Auxiliary variable sampling

One option to sample from the marginal filtering distribution is to perform sampling exactly as for the RBPF. The sampling procedure described in Section 3.3.2 produces a weighted particle system $\{\xi^i_{1:t}, \omega^i_t\}_{i=1}^N$ targeting the state-marginal smoothing density $p(\xi_{1:t} \mid y_{1:t})$. By discarding the history of the particle trajectories, we are left with a weighted particle system $\{\xi^i_t, \omega^i_t\}_{i=1}^N$ targeting the marginal filtering density $p(\xi_t \mid y_{1:t})$.

This is in analogy with the discussion about the PF in Section 3.2.1. As mentioned there, whether the PF targets the filtering distribution or the joint smoothing distribution is a matter of point of view.

Observe that, even if we sample the nonlinear states identically in the RBPF and the RBMPF, the latter can still circumvent the problems with slowly mixing systems as pointed out above. The reason is, that the problems arising in the RBPF are not due to poor approximations of the marginal filtering density appearing in (3.64), but due to poor approximations of the state-marginal smoothing density appearing in (3.61).

This approach to sampling shall be referred to as *auxiliary variable sampling*. The reason is that the sampling procedure alternatively can be explained in terms of auxiliary variables, as we now shall see. Recall that we wish to sample from the marginal filtering density, which can be expanded according to,

$$p(\xi_t \mid y_{1:t}) = \int \frac{p(y_t \mid \xi_{t-1:t}, y_{1:t-1})p(\xi_t \mid \xi_{t-1}, y_{1:t-1})}{p(y_t \mid y_{1:t-1})} p(\xi_{t-1} \mid y_{1:t-1})\, d\xi_{t-1}. \tag{3.66}$$

Let $\{\xi_{t-1}^j, \omega_{t-1}^j\}_{j=1}^N$ be a weighted particle system targeting $p(\xi_{t-1} \mid y_{1:t-1})$. By using the empirical distribution defined by this particle system in (3.66) we get,

$$p(\xi_t \mid y_{1:t}) \approx \sum_{j=1}^{N} \omega_{t-1}^j \frac{p(y_t \mid \xi_{t-1}^j, \xi_t, y_{1:t-1})p(\xi_t \mid \xi_{t-1}^j, y_{1:t-1})}{p(y_t \mid y_{1:t-1})}. \tag{3.67}$$

Now, instead of trying to sample "directly" from the above mixture (see Alternative S2 below), we aim at sampling a pair $\{J(i), \xi_t^i\}$, where the random variable $J(i)$ is an index into the mixture. Sampling from a mixture like (3.67), is naturally thought of as a two-step procedure. First, we choose one of the components at random, i.e. we draw an ancestor particle ξ_{t-1}^j. Second, we draw a new particle ξ_t^i from this specific component. The variable $J(i)$, denoted an auxiliary variable, can then be seen as the index of this ancestor particle.

A pair $\{J(i), \xi_t^i\}$ is proposed by first sampling the auxiliary variable,

$$J(i) \sim \mathrm{Cat}\left(\{\omega_{t-1}^j\}_{j=1}^N\right), \tag{3.68a}$$

and then, conditioned on this index, draw a new particle from some proposal density

$$\xi_t^i \sim r_t(\xi_t \mid \xi_{t-1}^{J(i)}, y_{1:t}). \tag{3.68b}$$

The pair is assigned an importance weight,

$$\omega_t^i \propto \frac{p(y_t \mid \xi_{t-1}^{J(i)}, \xi_t^i, y_{1:t-1})p(\xi_t^i \mid \xi_{t-1}^{J(i)}, y_{1:t-1})}{r_t(\xi_t^i \mid \xi_{t-1}^{J(i)}, y_{1:t})}, \tag{3.69}$$

and we can thereafter discard the auxiliary variable $J(i)$. Since we only know the weights up to proportionality, they are normalised to sum to one. By repeating the procedure N times, we obtain a weighted particle system $\{\xi_t^i, \omega_t^i\}_{i=1}^N$ targeting the marginal filtering distribution at time t.

As already pointed out, this sampling procedure is analogous to the sampling conducted in the standard RBPF, using multinomial resampling followed by a mutation step. Here, resampling is replaced by the auxiliary variable sampling (3.68a), and mutation corresponds to drawing the new particles in (3.68b). However, the observant reader might have noticed that there is a slight difference between the weight expression for the RBPF (3.46) and for the RBMPF (3.69). In the former, the densities are conditioned on the full history of the nonlinear state trajectory $\Xi_{1:t-1}$, whereas in the latter the conditioning is on just Ξ_{t-1}. As argued in Section 2.2.2, it is in general not sufficient to condition on the nonlinear state at a single time point, to retain the analytically tractable substructure in a CLGSS model. Hence, the densities appearing in the weight expression (3.69) are in the general case not available in closed form.

However, this will in fact not be an issue in the RBMPF setting. The reason is that the conditional filtering density $p(z_{t-1} \mid \xi_{t-1}, y_{1:t-1})$ is approximated by a Gaussian at time $t - 1$ (see Section 3.4.2). Given this approximation, conditioning on just Ξ_{t-1} in the RBMPF, will have the "same effect" as conditioning on $\Xi_{1:t-1}$ in the RBPF. Hence, the densities appearing in the weight expression (3.69) will indeed be available for evaluation, under the above mentioned Gaussian approximation. It can be said that the whole idea with the RBMPF, is to replace the conditioning on the nonlinear state trajectory, with a conditioning on the nonlinear state at a single time point.

Before we leave this section, it should be said that the idea of using auxiliary variables is not restricted to the RBMPF. The same idea can also be used to describe the inner workings of the PF, targeting the filtering distribution. In fact, auxiliary variables have been used by Pitt and Shephard [1999] to design the auxiliary particle filter. This is often seen as a "look-ahead" method, used to improve the performance of the filter. The idea is to use a different proposal distribution for the auxiliary variables than (3.68a). By modifying the weights to incorporate information about the current measurement y_t, before we sample the auxiliary variables, we are more likely to draw ancestor particles with a good fit to the current observation. In the presentation above, we did not make use of the "look-ahead" ideas of Pitt and Shephard [1999], even if this indeed is an interesting option. On the contrary, the auxiliary variable sampling presented here is, as previously pointed out, equivalent to using multinomial resampling followed by mutation in the standard RBPF.

Alternative S2: Marginal sampling

The second alternative for sampling in the RBMPF that we will consider, is to target the mixture density (3.67) directly. This is in complete analogy with the MPF by Klaas et al. [2005] (see also Section 3.2.4). That is, given a weighted particle system $\{\xi_{t-1}^j, \omega_{t-1}^j\}_{j=1}^N$ targeting $p(\xi_{t-1} \mid y_{1:t-1})$, we construct a proposal as a mixture density according to,

$$r_t'(\xi_t \mid y_{1:t}) = \sum_{j=1}^{N} \omega_{t-1}^j r_t(\xi_t \mid \xi_{t-1}^j, y_{1:t}). \tag{3.70}$$

From this, we draw a set of new particles $\{\xi_t^i\}_{i=1}^N$. To compute the importance weights, i.e. the quotient between the target and the proposal densities, we make use of the empirical approximation of the target density (3.67), resulting in,

$$\omega_t^i \propto \frac{\sum_{j=1}^N \omega_{t-1}^j p(y_t \mid \xi_{t-1}^j, \xi_t^i, y_{1:t-1}) p(\xi_t^i \mid \xi_{t-1}^j, y_{1:t-1})}{\sum_{j=1}^N \omega_{t-1}^j r_t(\xi_t^i \mid \xi_{t-1}^j, y_{1:t})}. \tag{3.71}$$

Finally, since we only know the weights up to proportionality, they are normalised to sum to one.

3.4.2 Gaussian mixture approximations

We now turn to the more central problem in the RBMPF, namely to find an approximation of the density (3.65). The general idea that we will employ is to approximate it as Gaussian. Hence, let us assume that $p(z_{t-1} \mid \xi_{t-1}, y_{1:t-1}) \approx \widehat{p}(z_{t-1} \mid \xi_{t-1}, y_{1:t-1})$ for some $t \geq 2$, with,

$$\widehat{p}(z_{t-1} \mid \xi_{t-1}, y_{1:t-1}) \triangleq \mathcal{N}\left(z_{t-1}; \bar{z}_{t-1|t-1}(\xi_{t-1}), P_{t-1|t-1}(\xi_{t-1})\right), \tag{3.72}$$

for some mean and covariance functions, $\bar{z}_{t-1|t-1}$ and $P_{t-1|t-1}$, respectively. At time $t = 2$, no approximation is in fact needed, since (3.72) then coincides with (3.34).

Remark 3.8. We will in the sequel use $\widehat{p}$ as a generic symbol for any density, which is an approximation of some density p. Just as for p (see Remark 2.1 on page 14), we will let the argument of $\widehat{p}$ indicate which density that is referred to, so that $\widehat{p}(\,\cdot\,) \approx p(\,\cdot\,)$.

Just as in the standard RBPF, if we augment the conditioning on the nonlinear state to $\xi_{t-1:t}$, and make a time update and measurement update of (3.72), we obtain

$$\widehat{p}(z_t \mid \xi_{t-1}, \xi_t, y_{1:t}) = \mathcal{N}\left(z_t; \tilde{z}_{t|t}(\xi_{t-1:t}), \widetilde{P}_{t|t}(\xi_{t-1:t})\right), \tag{3.73}$$

for some mean and covariance functions, $\tilde{z}_{t|t}$ and $\widetilde{P}_{t|t}$, respectively. For the case of mixed linear/nonlinear Gaussian state-space models, this is in analogy with the RBPF updating steps given in Section 3.3.1. However, note that under the assumption (3.72), the Gaussianity of (3.73) holds true for all types of CLGSS models.

The problem is that once we "remove" the conditioning on ξ_{t-1}, the Gaussianity is lost. Hence, to obtain a recursion, i.e. to end up with (3.72) with time index $t-1$ replaced by t, we need to find a Gaussian approximation of $p(z_t \mid \xi_t, y_{1:t})$ based on (3.73). Below, we provide to alternative approaches.

Alternative G1: Ancestral dependence

For both sampling alternatives presented in Section 3.4.1, we note that each proposed particle at time t can be traced back to an ancestor particle at time $t-1$. Using auxiliary variable sampling (Alternative S1), the index of the ancestor to particle ξ_t^i is simply the auxiliary variable $J(i)$, given by (3.68a). In the marginal sampling approach (Alternative S2), the ancestor is not as clearly visible. However, sampling from the proposal mixture (3.70) is, from an implementation point of view, done by first choosing a component at random, and then drawing a sample from this component. We can then take the index of this component as the ancestor particle index.

Hence, let $J(i)$ be the index of the ancestor to the particle ξ_t^i. By exploiting the ancestral dependence, we can straightforwardly approximate (3.65) as,

$$\widehat{p}(z_t \mid \xi_t^i, y_{1:t}) \triangleq \widehat{p}(z_t \mid \xi_t^i, \xi_{t-1}^{J(i)}, y_{1:t}), \tag{3.74}$$

for $i = 1, \ldots, N$. The right hand side in the above expression is Gaussian and given by (3.73). In other words, we set

$$\bar{z}_{t|t}(\xi_t^i) := \tilde{z}_{t|t}(\xi_{t-1}^{J(i)}, \xi_t^i), \tag{3.75a}$$

$$P_{t|t}(\xi_t^i) := \widetilde{P}_{t|t}(\xi_{t-1}^{J(i)}, \xi_t^i), \tag{3.75b}$$

for $i = 1, \ldots, N$.

Before we go on, let us pause for a moment and consider the meaning of this approximation. From an implementation point of view, (3.74) and (3.75) implies that there is no change in the updating formulas for the linear states, compared to the RBPF. Once we iterate over $t = 1, 2 \ldots$, (3.74) further implies that $\widehat{p}(z_t \mid \xi_t, y_{1:t}) = p(z_t \mid \xi_{1:t}, y_{1:t})$. Hence, the approximation of (3.65) using ancestral dependence consists of simply neglecting the dependence on $\Xi_{1:t-1}$.

In fact, this is exactly the approximation that we would obtain by marginalisation of the empirical distribution given by the RBPF. The weighted particle system $\{\xi_{1:t}^i, \omega_t^i\}_{i=1}^N$ produced by the RBPF defines an empirical distribution approximating the state-marginal smoothing distribution $\Phi_{1:t|t}^m$. Similarly to (3.14), we can marginalise this empirical distribution to find an approximation of the filtering distribution,

$$\widehat{\Phi}_{t|t}^N(d\xi_t, dz_t) = \int_{\mathsf{X}_\xi^{t-1}} \Phi_{t|t}^c(dz_t \mid \xi_{1:t}) \widehat{\Phi}_{1:t|t}^{m,N}(d\xi_{1:t}) = \sum_{i=1}^N \omega_t^i \Phi_{t|t}^c(dz_t \mid \xi_{1:t}^i) \delta_{\xi_t^i}(d\xi_t). \tag{3.76}$$

By taking the conditional of the approximate filtering distribution on the left hand side, this further implies that

$$\widehat{p}(z_t \mid \xi_t^i, y_{1:t}) = p(z_t \mid \xi_{1:t}^i, y_{1:t}). \tag{3.77}$$

Hence, the approximation of (3.65) obtained by the ancestral dependence approximation, is the same as that given by the RBPF. In fact, if auxiliary variable sampling (Alternative S1 in Section 3.4.1) is combined with ancestral dependence for approximating the conditional filtering density (3.65), the RBMPF is identical to the RBPF. Consequently, approximation using ancestral dependence will not be of any use when dealing with slowly mixing models, as the RBMPF will clearly suffer from the same issues as the RBPF.

Alternative G2: Mixing

We now present an alternative way of approximating the conditional density (3.65), which we shall call mixing. We start by noting that the density can be written,

$$p(z_t \mid \xi_t, y_{1:t}) = \int p(z_t \mid \xi_{t-1:t}, y_{1:t}) p(\xi_{t-1} \mid \xi_t, y_{1:t})\, d\xi_{t-1}. \tag{3.78}$$

Now, assume that we wish to evaluate (3.78), conditioned on some particle ξ_t^i. It can then be realised that the ancestral dependence approximation, described in the previous section, consist of approximating the above integral using MC integration with a single particle. That is, we plug the approximation,

$$p(\xi_{t-1} \mid \xi_t^i, y_{1:t})\, d\xi_{t-1} \approx \delta_{\xi_{t-1}^{J(i)}}(d\xi_{t-1}), \tag{3.79}$$

into the integral, resulting in the approximation (3.74) of the sought density (3.65).

From this point of view, a more natural approach would be to make use of the complete particle system $\{\xi_{t-1}^j, \omega_{t-1}^j\}_{j=1}^N$, to evaluate the integral. Thus, consider the second factor of the integrand in (3.78),

$$p(\xi_{t-1} \mid \xi_t, y_{1:t}) = \frac{p(y_t \mid \xi_{t-1:t}, y_{1:t-1}) p(\xi_t \mid \xi_{t-1}, y_{1:t-1})}{p(\xi_t, y_t \mid y_{1:t-1})} p(\xi_{t-1} \mid y_{1:t-1}). \tag{3.80}$$

The weighted particle system $\{\xi_{t-1}^j, \omega_{t-1}^j\}_{j=1}^N$ defines an empirical distribution, approximating the marginal filtering distribution at time $t-1$,

$$p(\xi_{t-1} \mid y_{1:t-1})\, d\xi_{t-1} \approx \widehat{\Phi}_{t-1|t-1}^{m,N}(d\xi_{t-1}) = \sum_{j=1}^N \omega_{t-1}^j \delta_{\xi_{t-1}^j}(d\xi_{t-1}). \tag{3.81}$$

By plugging this into (3.80) and (3.78), conditioned on ξ_t^i, we obtain

$$p(z_t \mid \xi_t^i, y_{1:t}) \approx \sum_{j=1}^N \gamma_t^{j,i} p(z_t \mid \xi_{t-1}^j, \xi_t^i, y_{1:t}), \tag{3.82a}$$

with,

$$\gamma_t^{j,i} = \frac{\omega_{t-1}^j p(y_t \mid \xi_{t-1}^j, \xi_t^i, y_{1:t-1}) p(\xi_t^i \mid \xi_{t-1}^j, y_{1:t-1})}{\sum_{k=1}^N \omega_{t-1}^k p(y_t \mid \xi_{t-1}^k, \xi_t^i, y_{1:t-1}) p(\xi_t^i \mid \xi_{t-1}^k, y_{1:t-1})}. \tag{3.82b}$$

Furthermore, by the Gaussianity assumption (3.73), we see that (3.82) is a Gaussian mixture model (GMM). Recall that we seek to approximate the left hand side of (3.82a) with a single Gaussian. To keep the full GMM representation is generally not an option, since this would result in a mixture with a number of components increasing exponentially over time. Hence, we propose to approximate the GMM with a single Gaussian, using moment matching. From (3.73), the mean and covariance of the GMM (3.82a) are given by,

$$\bar{z}_{t|t}(\xi_t^i) = \sum_{j=1}^N \gamma_t^{j,i} \tilde{z}_{t|t}^{j,i}, \tag{3.83a}$$

$$P_{t|t}(\xi_t^i) = \sum_{j=1}^N \gamma_t^{j,i} \left(\widetilde{P}_{t|t}^{j,i} + (\tilde{z}_{t|t}^{j,i} - \bar{z}_{t|t}^i)(\tilde{z}_{t|t}^{j,i} - \bar{z}_{t|t}^i)^{\mathsf{T}} \right), \tag{3.83b}$$

respectively. Here we have used the shorthand notation $\tilde{z}_{t|t}^{j,i}$ instead of $\tilde{z}_{t|t}(\xi_{t-1}^j, \xi_t^i)$, etc. In conclusion, the above results provide a Gaussian approximation of (3.65) according to,

$$\widehat{p}(z_t \mid \xi_t^i, y_{1:t}) \triangleq \mathcal{N}\left(z_t; \bar{z}_{t|t}(\xi_t^i), P_{t|t}(\xi_t^i)\right). \tag{3.84}$$

This approximation procedure, called mixing, is summarised and exemplified in Algorithm 3.4, where the RBMPF is applied to a mixed linear/nonlinear Gaussian state-space model.

Algorithm 3.4 RBMPF for mixed linear/nonlinear Gaussian state-space models

```
Input: An augmented weighted particle system $\{\xi^i_{t-1}, \omega^i_{t-1}, \bar{z}^i_{t-1|t-1}, P^i_{t-1|t-1}\}_{i=1}^N$,
       (approximately) targeting $p(z_{t-1} \mid \xi_{t-1}, y_{1:t-1})p(\xi_{t-1} \mid y_{1:t-1})$.
Output: An augmented weighted particle system $\{\xi^i_t, \omega^i_t, \bar{z}^i_{t|t}, P^i_{t|t}\}_{i=1}^N$, (approximately)
        targeting $p(z_t \mid \xi_t,\ y_{1:t})p(\xi_t \mid y_{1:t})$.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Sampling:
1: Draw ancestor particle indices, $J(i) \sim \mathrm{Cat}\left(\{\omega^j_{t-1}\}_{j=1}^N\right)$, for $i = 1, \ldots, N$.
2: Propose new particles, $\xi^i_t \sim r_t(\xi_t \mid \xi^{J(i)}_{t-1}, y_{1:t})$, for $i = 1, \ldots, N$.
Prediction:
3: For $j = 1, \ldots, N$,
       $\alpha^j_{t|t-1} = f^{\xi,j}_{t-1} + A^{\xi,j}_{t-1}\bar{z}^j_{t-1|t-1},$
       $\zeta^j_{t|t-1} = f^{z,j}_{t-1} + A^{z,j}_{t-1}\bar{z}^j_{t-1|t-1},$
       $P^j_{t|t-1} = \Sigma^{z,j}_{t|t-1} - (\Sigma^{\xi z,j}_{t|t-1})^\mathsf{T}(\Sigma^{\xi,j}_{t|t-1})^{-1}(\Sigma^{\xi z,j}_{t|t-1}),$
   with
       $\Sigma^j_{t|t-1} = Q^j_{t-1} + A^j_{t-1}P^j_{t-1|t-1}(A^j_{t-1})^\mathsf{T}.$
Moment matching:
4: for $i = 1$ to $N$ do
5:     Condition the linear state on the particle $\xi^i_t$. For $j = 1, \ldots, N$,
           $\tilde{z}^{j,i}_{t|t-1} = \zeta^j_{t|t-1} + (\Sigma^{\xi z,j}_{t|t-1})^\mathsf{T}(\Sigma^{\xi,j}_{t|t-1})^{-1}(\xi^i_t - \alpha^j_{t|t-1}).$
6:     Condition the linear state on the current measurement. For $j = 1, \ldots, N$,
           $\tilde{z}^{j,i}_{t|t} = \tilde{z}^{j,i}_{t|t-1} + K^{j,i}_t(y_t - \hat{y}^{j,i}_t),$
           $\widetilde{P}^{j,i}_{t|t} = P^j_{t|t-1} - K^{j,i}_t C^i_t P^j_{t|t-1},$
       with
           $\hat{y}^{j,i}_t = h^i_t + C^i_t\tilde{z}^{j,i}_{t|t-1},$
           $S^{j,i}_t = R^i_t + C^i_t P^j_{t|t-1}(C^i_t)^\mathsf{T}$
           $K^{j,i}_t = P^j_{t|t-1}(C^i_t)^\mathsf{T}(S^{j,i}_t)^{-1}.$
7:     For $j = 1, \ldots, N$, compute the mixture weights $\gamma^{j,i}_t$ according to (3.82b), using
       Gaussian approximations of the densities according to,
           $\widehat{p}(y_t \mid \xi^j_{t-1}, \xi^i_t, y_{1:t-1}) = \mathcal{N}\left(y_t; \hat{y}^{j,i}_t, S^{j,i}_t\right),$
           $\widehat{p}(\xi^i_t \mid \xi^j_{t-1}, y_{1:t-1}) = \mathcal{N}\left(\xi^i_t; \alpha^j_{t|t-1}, \Sigma^{\xi,j}_{t|t-1}\right).$
8:     Compute the mean $\bar{z}^i_{t|t}$ and covariance $P^i_{t|t}$ of the GMM according to (3.83).
9: end for
Weighting:
10: For $i = 1, \ldots, N$, compute the weights $\omega^i_t$ according to Alternative S1 (3.69), or
    according to Alternative S2 (3.71), using Gaussian approximations of the involved
    densities as in step 7 above.
```

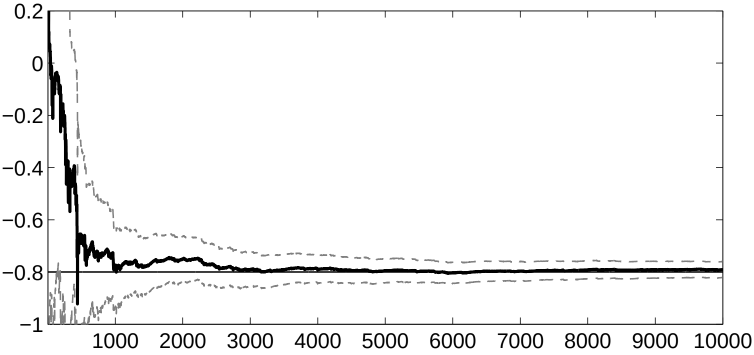

Figure 3.7: *RBMPF estimate of Z_t (thick black line) and the estimated 3σ-confidence interval (grey dotted lines), as function of time t. The "true" value is -0.8.*

3.4.3 Discussion

Before we proceed with a discussion on some of the properties of the RBMPF, let us return to Example 3.2, to see how the RBMPF performs. A more thorough numerical evaluation of the RBMPF is given in Section 6.2, where the RBMPF is applied to the problem of nonlinear system identification.

Example 3.3: RBMPF for a partially static system

To cope with the degeneracy problems that arose for the RBPF in Example 3.2, an RBMPF is applied to the same data. As for the RBPF, a bootstrap RBMPF with $N = 100$ particles is used. The RBMPF uses auxiliary variable sampling (Alternative S1) together with mixing for Gaussian mixture approximation (Alternative G2), as in Algorithm 3.4. The estimate of the state Z_t, together with the estimated 3σ-confidence intervals, are shown in Figure 3.7. In this example, it is clear that the RBMPF succeeds much better than the RBPF, in estimating the linear state.

We emphasise that this example is provided as an illustration of the concept, and not an evaluation of the RBMPF. We have only considered one realisation of data, which of course is not enough to draw any general conclusions. A more substantial numerical evaluation of the RBMPF is given in Section 6.2.

As indicated by Example 3.3, and also by the results in Section 6.2, there is indeed a large potential gain in using the RBMPF instead of the RBPF for certain problems. This gain is related to how the Gaussian approximations for the conditional filtering distribution (3.65) are computed. In Section 3.4.2 we argued that, among the two approximations discussed here, only mixing (Alternative G2) will circumvent the problems that arose in the RBPF. The reason is that the mixing approximation causes an increase in the estimated uncertainty about the linear states, which is otherwise underestimated. Since we, in this thesis, have proposed the RBMPF as a way to deal with slowly mixing systems, it shall from now on be assumed that the mixing approximation is the default choice, whenever referring to the RBMPF.

***Table 3.1:** RBMPF combinations*

Sec. 3.4.1	Sec. 3.4.2	Comment
S1	G1	Equivalent to the RBPF
S2	G1	Jianjun et al. [2007] and Smal et al. [2007]
S1	G2	RBMPF proposed in this thesis (Algorithm 3.4)
S2	G2	

The RBMPF by Jianjun et al. [2007] and that of Smal et al. [2007], are both focused on combining the RBPF with the MPF by Klaas et al. [2005]. They both use marginal sampling (Alternative S2) combined with ancestral dependence (Alternative G1). Consequently, they will suffer from the same drawbacks as the RBPF when applied to slowly mixing systems. However, neither are they designed to deal with such systems. The reason for why the filters proposed by Jianjun et al. [2007] and Smal et al. [2007], are sorted into the RBMPF framework of this thesis, is to emphasise the differences, and the different objectives, of these methods compared to the RBMPF proposed here. See also Table 3.1, where different RBMPF combinations are summarised.

The above mentioned benefits of the RBMPF does (of course) come at a price. Most notably, the RBMPF has $O(N^2)$ complexity. This is easily seen in Algorithm 3.4, where indices i and j both range from 1 to N. In fact, just as the RBPF can be seen as using N parallel Kalman filters, the RBMPF uses N^2 Kalman filters. In this way, by viewing each particle as a separate model, the RBMPF very much resembles the 2nd order, generalised pseudo-Bayesian (GPB2) multiple model filter. Guided by this insight, we could also derive an RBMPF similar to the 1st order generalised pseudo-Bayesian (GPB1) multiple model filter (see [Bar-Shalom et al., 2001] for the two GPB filters). This would reduce the complexity to $O(N)$, but at the cost of coarser approximations, likely to degrade the performance of the filter. A third approach in this direction, is to start from the IMM filter by Blom [1984] and Blom and Bar-Shalom [1988]. The IMM filter is a popular choice for multiple model filtering, since it has lower complexity than GPB2 (still quadratic, but smaller constants), but is known to have similar performance [Blom and Bar-Shalom, 1988]. However, it is not clear that the ideas underlying the IMM filter, can be straightforwardly generalised to the RBMPF. This issue requires further attention.

Another way to reduce the complexity of the algorithm is by numerical approximations of the mixture models. Due to the exponential decay of the Gaussian components, truncation might aid in making fast, sufficiently accurate, evaluations of the GMM moments. A related approach is that of Gray and Moore [2000, 2003], used for fast, nonparametric density estimation. Also, fast summation methods, similar to the ideas underlying the fast Gauss transform by Greengard and Strain [1991], Greengard and Sun [1998] and the improved fast Gauss transform by Yang et al. [2003], might be of use. However, as discussed by Boyd [2010], truncation methods should in general have more to offer than fast summation methods, for Gaussian components which are quickly decaying.

Finally, another option is of course to seek alternative approximations of the conditional filtering distribution (3.65), not based on a GMM as in (3.82). By doing so, one can possibly find good approximations, which can be evaluated more efficiently than the ones presented here.

4

Asymptotic properties of SMC methods

In this chapter we consider the convergence properties of sequential Monte Carlo (SMC) methods. In Section 4.1 we review some existing convergence results and discuss their different properties. The main contribution of this chapter lies in Section 4.2, where we analyse the asymptotic variance of the Rao-Blackwellised particle filter (RBPF) and compare it to that of the standard particle filter (PF), with the purpose of answering the question; how much do we gain from using an RBPF instead of a PF?

4.1 Convergence

Assume that we wish to compute the expectation of some function $\varphi : \mathsf{X} \to \mathbb{R}$ under the filtering distribution, i.e. we seek $\Phi_{t|t}(\varphi)$. Assume further that we estimate this quantity by employing a PF, generating a weighted particle system $\{x_t^i, w_t^i\}_{i=1}^N$ targeting $\Phi_{t|t}$, and construct a randomised estimator according to,

$$\hat{\varphi}_{\text{PF}}^N = \sum_{i=1}^N w_t^i \varphi(x_t^i). \tag{4.1}$$

A natural question to ask is; will this estimator converge to the true expectation as we increase the number of particles?

There exists a vast amount of literature dedicated to answering this question, and there are still many results that remain to be found. The reason is that the question is not as simple as it first appears. If we dig a little deeper into the problem we may elaborate the question and ask;

- What type of convergence can be guaranteed and at what rate will the estimator convergence?

- What constraints do we need to impose on the underlying model (in this case the SSM) for the convergence results to hold?
- What constraints do we need to impose on the test function φ for the results to hold?

Due to the high variety of different aspects of convergence, it is hard to present a thorough overview of the existing results. There is an interplay between the strengths of the results and the strengths of the assumptions. A full spectrum of results, ranging from weak to strong, can be found in the literature. In this section, we give a brief survey of some of the aspects of SMC convergence, but we emphasise that this overview is in no way complete.

4.1.1 Consistency and central limits for general SMC

The first observation that we make, which is a very comforting one, is that SMC methods do converge as the number of particles tend to infinity, and that they do so very generally. However, in this claim, we do not consider the strength, or the rate of the convergence. To begin with, we will focus on asymptotic results on consistency (i.e. convergence in probability) and asymptotic normality of weighted particle systems. The material of this section is based on the (quite recent) work by Douc and Moulines [2008], basically because these are the results that we will use in Section 4.2. However, it should be noted that many convergence results for SMC methods were established prior to this work; see the references below.

Before we go on, we introduce some additional notation that will be used in this chapter. First of all, we note that so far in this thesis, we have mostly considered weighted particle systems with a fixed number of particles. In this section, the idea is to analyse the properties of these systems as the number of particles tend to infinity. Hence, we must allow the number of particles N to vary. We will indicate this by considering a sequence of integers $\{N_n\}_{n=1}^{\infty}$, corresponding to the number of particles. Naturally, this sequence is increasing and divergent, so that $N_n \to \infty$ as $n \to \infty$, but it is not necessarily the case that $N_n = n$.

Furthermore, the particle systems generated by an SMC method follow different laws depending on the number of particles used. Hence, we are in fact dealing with triangular arrays of random variables, i.e. constructs of the form,

$$\begin{array}{llll} Z^{1,1} & & & \\ Z^{2,1} & Z^{2,2} & & \\ Z^{3,1} & Z^{3,2} & Z^{3,3} & \\ \vdots & \vdots & \vdots & \ddots \end{array}$$

A generic weighted particle system will thus, in this section (and also in Section 4.2), be written $\{Z^{n,i}, W^{n,i}\}_{i=1}^{N_n}$. Though cumbersome, it is important to keep the dependence on n in the notation (cf. Definition 3.1).

Finally, convergence in distribution and convergence in probability are denoted by $\xrightarrow{\text{D}}$ and $\xrightarrow{\text{P}}$, respectively.

Now, let us introduce the concept of weighted particle system consistency, borrowed from

Douc and Moulines [2008].

Definition 4.1 (Consistency). A weighted particle system $\{Z^{n,i}, W^{n,i}\}_{i=1}^{N_n}$ on Z is said to be consistent for the probability measure μ and the set $\mathsf{C} \subseteq \mathsf{L}^1(\mathsf{Z}, \mu)$ if,

$$\sum_{i=1}^{N_n} W^{n,i} \varphi(Z^{n,i}) \xrightarrow{\text{P}} \mu(\varphi), \qquad \text{for any } \varphi \in \mathsf{C}, \tag{4.2a}$$

$$\max_{i=1,\,\dots,\,N_n} W^{n,i} \xrightarrow{\text{P}} 0, \tag{4.2b}$$

as $n \to \infty$.

Hence, if a weighted particle system is consistent for some probability measure μ, then the MC estimate of the expectation of a function $\varphi \in \mathsf{C}$, converges in probability to the true expectation as the number of particles tends to infinity. Furthermore, by (4.2b), we note the definition of consistency also requires that the individual contribution to the MC estimate from each particle should tend to zero.

Douc and Moulines [2008] show that the general SMC framework, as outlined in Section 3.1, produces consistent particle systems. More precisely, if we start with a consistent weighted particle system, then any step of an SMC method will retain the consistency. We reproduce two of the main results from [Douc and Moulines, 2008].

Theorem 4.1 (Mutation: preservation of consistency). *Let ν and μ be probability measures on the measurable spaces $(\tilde{\mathsf{X}}, \tilde{\mathcal{X}})$ and $(\mathsf{X}, \mathcal{X})$, respectively. Let L be a transformation kernel for ν and μ according to* (3.2). *Let $\{\tilde{\xi}^{n,i}, \tilde{\omega}^{n,i}\}_{i=1}^{M_n}$ be a weighted particle system, consistent for $(\nu, \tilde{\mathsf{C}})$ and assume that $L(\mathsf{X} \mid \cdot\,) \in \tilde{\mathsf{C}}$. Then, the weighted particle system $\{\xi^{n,i}, \omega^{n,i}\}_{i=1}^{N_n}$ generated by mutation according to* (3.4) *and* (3.5) *is consistent for (μ, C) with, $\mathsf{C} = \{\varphi \in \mathsf{L}^1(\mathsf{X}, \mu), L(|f|) \in \tilde{\mathsf{C}}\}$.*

Proof: See [Douc and Moulines, 2008], Theorem 1. □

Theorem 4.2 (Selection: preservation of consistency). *Let μ be a probability measure on the measurable space $(\mathsf{X}, \mathcal{X})$ and let $\{\xi^{n,i}, \omega^{n,i}\}_{i=1}^{N_n}$ be a weighted particle system, consistent for (μ, C). Then, the equally weighted particle system $\{\tilde{\xi}^{n,i}, 1/M_n\}_{i=1}^{M_n}$ generated by either multinomial or residual resampling (see Section 3.1.2) is consistent for (μ, C).*

Proof: See [Douc and Moulines, 2008], Theorem 3. □

It should be mentioned that Douc and Moulines [2008] provide similar consistency results for several additional selection schemes, in the categories of branching and fractional reweighting.

As mentioned above, the results on consistency tell us that general SMC methods do converge, in the sense that the sequence of random variables $\{\sum_{i=1}^{N_n} \omega^{n,i} \varphi(\xi^{n,i})\}_{n=1}^{\infty}$ converges in probability to the expectation $\mu(\varphi)$. It is also possible to assess the asymptotic distribution for this sequence of random variables, which provides a central limit theorem (CLT) for SMC methods. This theory has been studied and gradually developed by Del Moral and Miclo [2000], Gilks and Berzuini [2001], Del Moral [2004], Chopin [2004], Künsch [2005] and Douc and Moulines [2008]. Here, we continue to review the results by Douc and Moulines [2008], and introduce the concept of asymptotic normality.

Definition 4.2 (Asymptotic normality). Let μ be a probability measure and γ a finite measure, both on the measurable space $(\mathsf{Z}, \mathscr{Z})$. Let $\mathsf{A} \subseteq \mathsf{L}^1(\mathsf{Z}, \mu)$ and $\mathsf{W} \subseteq \mathsf{L}^1(\mathsf{Z}, \gamma)$ be sets of real-valued, measurable functions on Z. Let σ be a real nonnegative function on A and $\{a_n\}_{n=1}^{\infty}$ a nondecreasing real sequence diverging to infinity.

A weighted particle system $\{Z^{n,i}, W^{n,i}\}_{i=1}^{N_n}$ on Z is said to be asymptotically normal for $(\mu, \mathsf{A}, \mathsf{W}, \sigma, \gamma, \{a_n\})$ if,

$$a_n \sum_{i=1}^{N_n} W^{n,i} \left(\varphi(Z^{n,i}) - \mu(\varphi)\right) \xrightarrow{\text{D}} \mathcal{N}(0, \sigma^2(\varphi)), \qquad \text{for any } \varphi \in \mathsf{A}, \tag{4.3a}$$

$$a_n^2 \sum_{i=1}^{N_n} (W^{n,i})^2 \varphi(Z^{n,i}) \xrightarrow{\text{P}} \gamma(\varphi), \qquad \text{for any } \varphi \in \mathsf{W}, \tag{4.3b}$$

$$a_n \max_{i=1,\ldots,N_n} W^{n,i} \xrightarrow{\text{P}} 0, \tag{4.3c}$$

as $n \to \infty$.

Here, the convergence rate is given by the divergent sequence $\{a_n\}_{n=1}^{\infty}$, which in general need not be equal to $\{\sqrt{N_n}\}_{n=1}^{\infty}$. However, for the results that we will encounter in the sequel, this is indeed the case. We review two additional theorems given by Douc and Moulines [2008], which establish the asymptotic normality for general SMC methods.

Theorem 4.3 (Mutation: preservation of asymptotic normality). *Suppose that the assumptions of Theorem 4.1 hold. Assume in addition that the weighted particle system* $\{\tilde{\xi}^{n,i}, \tilde{\omega}^{n,i}\}_{i=1}^{M_n}$ *is asymptotically normal for* $(\nu, \tilde{\mathsf{A}}, \tilde{\mathsf{W}}, \tilde{\sigma}, \tilde{\gamma}, \{a_n\})$ *and that* $R(W^2) \in \tilde{\mathsf{W}}$. *Then, the weighted particle system* $\{\xi^{n,i}, \omega^{n,i}\}_{i=1}^{N_n}$ *generated by mutation according to* (3.4) *and* (3.5) *is asymptotically normal for* $(\mu, \mathsf{A}, \mathsf{W}, \sigma, \gamma, \{a_n\})$ *with,* $\gamma(f) \triangleq M_n {N_n}^{-1} \tilde{\gamma}(R(W^2 f))/[\nu(L(\mathsf{X} \mid \cdot))]^2$,

$$\sigma^2(f) = \frac{\tilde{\sigma}^2(L(\bar{f})) + M_n {N_n}^{-1} \gamma\left(R\left([W\bar{f} - R(W\bar{f})]^2\right)\right)}{[\nu(L(\mathsf{X} \mid \cdot))]^2}, \tag{4.4a}$$

$$\bar{f} = f - \mu(f), \tag{4.4b}$$

and

$$\mathsf{A} \triangleq \{f : L(|f|) \in \tilde{\mathsf{A}}, R(W^2 f^2) \in \tilde{\mathsf{W}}\}, \tag{4.4c}$$

$$\mathsf{W} \triangleq \{f : R(W^2 |f|) \in \tilde{\mathsf{W}}\}. \tag{4.4d}$$

Proof: See [Douc and Moulines, 2008], Theorem 2. □

Theorem 4.4 (Selection: preservation of asymptotic normality). *Suppose that the assumptions of Theorem 4.2 hold. Assume in addition that the weighted particle system*

$\{\xi^{n,i}, \omega^{n,i}\}_{i=1}^{N_n}$ is asymptotically normal for $(\mu, \mathsf{A}, \mathsf{W}, \sigma, \gamma, \{a_n\})$ and that $a_n^{-2} N_n \xrightarrow{\mathrm{P}} \beta^{-1} \in \mathbb{R}_+$. Let $M_n = \ell N_n$ for some $\ell \in \mathbb{R}_+$. Then, the equally weighted particle system $\{\tilde{\xi}^{n,i}, 1/M_n\}_{i=1}^{M_n}$ generated by multinomial resampling is asymptotically normal for $(\mu, \tilde{\mathsf{A}}, \mathsf{C}, \tilde{\sigma}, \tilde{\gamma}, \{a_n\})$ with $\tilde{\gamma} = \beta \ell^{-1} \mu$, $\tilde{\mathsf{A}} \triangleq \{f : f \in \mathsf{A}, f^2 \in \mathsf{C}\}$ and

$$\tilde{\sigma}^2(f) = \beta \ell^{-1} \mathrm{Var}_{\mu}(f) + \sigma^2(f). \tag{4.5}$$

Proof: See [Douc and Moulines, 2008], Theorem 4. □

Douc and Moulines [2008] provide similar theorems for several additional selection methods, such as residual resampling and different branching procedures.

Related to the material presented in this section is the work by Douc et al. [2009], in which consistency and asymptotic normality for the auxiliary PF is established (see also the work by Johansen and Doucet [2008]). Furthermore, Douc et al. [2010] analyse the smoothing problem (which we will discuss in the subsequent chapter) and provide central limit theorems for certain particle smoothers.

4.1.2 Non-asymptotic and strong convergence

The convergence results reviewed in the previous section were all asymptotic, i.e. they considered the limits of the particle approximations as the number of particles tended to infinity. Another, common approach to analysing the convergence of SMC methods is to fix the number of particles N. The error (in some appropriate sense) is then bounded by a function decreasing in N, yielding a qualitative result on the convergence rate of the method. In this section, we provide a brief and informal discussion on this type of results. For a more in-depth theoretical treatment, we refer to the various articles and books, referenced below.

A common approach is to consider the L^p-error of an estimator derived from the PF (or a similar method). For $\varphi \in \mathsf{L}^p(\mathsf{X}, \Phi_{t|t})$ the L^p-error can typically be bounded according to,

$$\mathrm{E}\left[|\widehat{\Phi}_{t|t}^N(\varphi) - \Phi_{t|t}(\varphi)|^p \right]^{1/p} \leq \frac{C_{\varphi}(t,p)}{\sqrt{N}}, \tag{4.6}$$

where $C_{\varphi}(t,p)$ is a φ-dependent function of t and p, but independent of the number of particles N. This type of results are provided by e.g. Del Moral and Miclo [2000] and Del Moral [2004] for bounded functions φ. In this case, it holds that $C_{\varphi}(t,p) = C(t,p)\|\varphi\|_{\infty}$ for some function C and where $\|\cdot\|_{\infty}$ is the supremum norm. Hu et al. [2008, 2011] extend these results, for $p \geq 2$, to unbounded functions φ. It should be noted that to enable this, they are forced to make certain algorithmic modifications. However, as pointed out by Hu et al. [2008], similar modifications have previously been introduced on heuristic grounds, to obtain a more practically applicable algorithm. Furthermore, Douc et al. [2009] provide results on the L^p-error for the auxiliary PF (for bounded test functions).

Related to L^p-bounds are exponential deviation inequalities, typically of the form,

$$\mathrm{P}\left(|\widehat{\Phi}_{t|t}^N(\varphi) - \Phi_{t|t}(\varphi)| \geq \epsilon \right) \leq A_{\varphi}(t) e^{-N\epsilon^2 / B_{\varphi}(t)}, \tag{4.7}$$

for some $A_\varphi(\,\cdot\,)$, $B_\varphi(\,\cdot\,)$, and $\epsilon > 0$. This type of results are (for bounded functions φ) given by e.g. Del Moral and Miclo [2000] and Douc et al. [2010] (the results by Douc et al. [2010] also apply to certain particle smoothers). Furthermore, from (4.7) and by the Borel-Cantelli lemma, almost sure convergence follows;

$$\mathrm{P}\left(\lim_{N\to\infty} \widehat{\Phi}^N_{t|t}(\varphi) = \Phi_{t|t}(\varphi)\right) = 1. \tag{4.8}$$

It should be noted that the bounds given above are in general not uniform in the time parameter t. That is, there is no guarantee that the functions A_φ, B_φ and C_φ do not increase over time. To obtain time uniform bounds, some mixing assumptions on the model are needed. The most common assumption, is to bound the transition density function from above and below. Under this assumption, time uniform bounds have been established by for instance Del Moral and Guionnet [2001], Le Gland and Oudjane [2004] and Künsch [2005]. However, this is a strong assumption, which can be satisfied basically only if the state-space is compact. Quite recently, Handel [2009] has established time uniform results under weaker mixing assumptions. However, the expense of this is that the results are weakened and that no convergence rate is obtained. To find stronger, time uniform bounds under weak mixing assumption (if such bounds exist) is a topic for future work.

4.2 Variance reduction for the RBPF

In Section 3.3 we presented the RBPF, an SMC method designed to exploit a type of conditional, tractable substructure in the model at hand. We motivated the RBPF by claiming that it will improve the accuracy of the filter, in the sense that any estimator derived from the RBPF will intuitively have lower variance than the corresponding estimator derived from the standard PF. Informally, the reason for this is that in the RBPF, the particles are spread in a lower dimensional space, yielding a denser particle representation of the underlying distribution. The improved accuracy is also something that is experienced by practitioners. However, it can be argued that it is still not beneficial to resort to Rao-Blackwellisation in all cases. The reason is that the RBPF in general is computationally more expensive per particle, compared to the standard PF. For instance, for an RBPF targeting a conditionally linear Gaussian state-space (CLGSS) model, each particle is equipped with a Kalman filter (KF), all which need to be updated at each iteration. Hence, for a fixed computational effort, we can choose to either use Rao-Blackwellisation or to run a standard PF, but instead increase the number of particles. Both these alternatives will reduce the variance of the estimators. Hence, it is important to understand and to be able to quantify how large variance reduction we can expect from the RBPF, in order to make suitable design choices for any given problem.

In this section we will study the asymptotic (in the number of particles) variances for the RBPF and the PF. We provide an explicit expression for the difference between the variance of an estimator derived from the PF and the variance of the corresponding estimator derived from the RBPF. The material of the present section has previously been published in [Lindsten et al., 2011b].

Of course, there has been previous work regarding the variance reduction for the RBPF.

Doucet et al. [2000b] motivates the RBPF by concluding that the weight variance will be lower than for the PF, but they do not consider the variances of any estimators. This is done by Chopin [2004], who, under certain assumptions, concludes that the variance of an estimator based on the PF always is at least as high as for the RBPF. However, no explicit expression for the difference is given, and the test functions considered are restricted to one partition of the state-space. Doucet et al. [2000a] also analyse the RBPF and the reduction of asymptotic variance. However, they only consider an importance sampling setting and neglect the important selection step. Karlsson et al. [2005] studies the problem empirically, by running simulations on a specific example. Here, they have also analysed the number of computations per iteration in the RBPF and the PF, respectively.

4.2.1 PF and RBPF revisited

Before we go on with the results on variance reduction, let us revisit the PF of Section 3.2.3 and the RBPF of Section 3.3 to establish the notation used in this section.

We shall assume that a PF is used to target the joint smoothing distribution $\Phi_{1:t|t}$. Let L_t be a transformation kernel for this distribution, i.e. a kernel from X^{t-1} to X^t such that,

$$\Phi_{1:t|t}(dx_{1:t}) = \frac{\Phi_{1:t-1|t-1}L_t(dx_{1:t})}{\Phi_{1:t-1|t-1}L_t(\mathsf{X}^t)}. \tag{4.9}$$

In the mutation step of the filter, we mutate the particle trajectories at time $t-1$ into new particle trajectories at time t, by sampling from a proposal kernel (a transition kernel from X^{t-1} to X^t),

$$R_t(dx_{1:t} \mid \tilde{x}_{1:t-1}). \tag{4.10}$$

Recall that the proposal kernel very well may depend on the measurement sequence $y_{1:t}$, but we do not make this dependence explicit. The kernels L_t and R_t are chosen such that $L_t(\,\cdot \mid \tilde{x}_{1:t-1}) \ll R_t(\,\cdot \mid \tilde{x}_{1:t-1})$ for all $\tilde{x}_{1:t-1} \in \mathsf{X}^{t-1}$. Furthermore, we shall assume that the weight function according to (3.3) on page 31, here given by,

$$W_t'(x_{1:t}) = \frac{dL_t(\,\cdot \mid \tilde{x}_{1:t-1})}{dR_t(\,\cdot \mid \tilde{x}_{1:t-1})}(x_{1:t}), \tag{4.11}$$

only depends on the "new" trajectory $x_{1:t}$. As discussed in Section 3.2.3 this is often the case when the target is the joint smoothing distribution. In particular, it is true whenever the proposal kernel has the form given by (3.17) on page 39, i.e. when the proposal is such that we keep the old trajectory up to time $t-1$ and simply append a sample at time t.

Let us define a measure on $\mathcal{X}^t$ according to,

$$\pi_{1:t|t}(dx_{1:t}) \triangleq \int\limits_{\mathsf{X}^{t-1}} R_t(dx_{1:t} \mid \tilde{x}_{1:t-1})\Phi_{1:t-1|t-1}(d\tilde{x}_{1:t-1}). \tag{4.12}$$

This can be seen as the "proposed joint smoothing distribution" at time t under the proposal kernel R_t. In the analysis that follows, it is convenient to replace (4.11) with another weight function defined by,

$$W_t(x_{1:t}) \triangleq \frac{d\Phi_{1:t|t}}{d\pi_{1:t|t}}(x_{1:t}). \tag{4.13}$$

In fact, with $C_t = \Phi_{1:t-1|t-1}L_t(\mathsf{X}^t)$ we may write (4.9) as,

$$\begin{aligned}\Phi_{1:t|t}(dx_{1:t}) &= \frac{1}{C_t}\int\limits_{\mathsf{X}^{t-1}} L_t(dx_{1:t} \mid \tilde{x}_{1:t-1})\Phi_{1:t-1|t-1}(d\tilde{x}_{1:t-1}) \\ &= \frac{1}{C_t}\frac{dL_t(\,\cdot \mid \tilde{x}_{1:t-1})}{dR_t(\,\cdot \mid \tilde{x}_{1:t-1})}(x_{1:t})\pi_{1:t|t}(dx_{1:t}),\end{aligned} \tag{4.14}$$

which implies that,

$$W_t(x_{1:t}) = \frac{1}{C_t}W'_t(x_{1:t}), \qquad \pi_{1:t|t}\text{-a.s.} \tag{4.15}$$

Hence, (4.13) is an “unnormalised” weight function, since it equals (4.11) up to a constant. However, whether we use (4.11) or (4.13) will not make any difference in the actual PF algorithm, since either way, the weights are normalised to sum to one. As previously pointed out, the reason for why we prefer (4.13) over (4.11) is because it is more convenient to work with.

Remark 4.1. Alternatively, we may go the other way around and define a function according to (4.13). Then, taking $L_t(dx_{1:t} \mid \tilde{x}_{1:t-1}) \triangleq W_t(x_{1:t})R_t(dx_{1:t} \mid \tilde{x}_{1:t-1})$ gives a plausible transformation kernel in (4.9) with $\Phi_{1:t-1|t-1}L_t(\mathsf{X}^t) = 1$, meaning that $W_t = W'_t$.

The second crucial step of an SMC method is selection, as discussed in Section 3.1.2. As mentioned there, many different selection strategies are available. Here, we shall assume that selection is done by multinomial resampling which is performed at each iteration of the algorithm. This gives a PF for the joint smoothing distribution which we summarise in Algorithm 4.1. However, results similar to those presented in Section 4.2.3 could be obtained for other types of PFs as well, such as the auxiliary PF by Pitt and Shephard [1999] and PFs with more sophisticated selection schemes.

Now, as in Section 3.3 we assume that there is an analytically tractable substructure in the model. Thus, we partition the state variable as $X_t = \{\Xi_t, Z_t\}$ with $\mathsf{X} = \mathsf{X}_\xi \times \mathsf{X}_z$, and factorise the joint smoothing distribution according to,

$$\Phi_{1:t|t}(dx_{1:t}) = \Phi^m_{1:t|t}(d\xi_{1:t})\Phi^c_{1:t|t}(dz_{1:t} \mid \xi_{1:t}), \tag{4.16}$$

where $\{\xi_t, z_t\}$ identifies to x_t. Here, $\Phi^m_{1:t|t}$ is the state-marginal smoothing distribution of $\Xi_{1:t}$ and $\phi^c_{1:t|t}$ is the conditional joint smoothing distribution of $Z_{1:t}$ given $\Xi_{1:t} = \xi_{1:t}$. The conditional distribution is assumed to be analytically tractable, typically Gaussian or with finite support.

Remark 4.2. More precisely, as pointed out in Remark 3.7 on page 43, $\Phi^c_{1:t|t}$ is a kernel from X^t_ξ to X^t_z. For each fixed $\xi_{1:t}$, $\Phi^c_{1:t|t}(\,\cdot \mid \xi_{1:t})$ is a measure on X^t_z, and can hence be viewed as a conditional distribution. In the notation used in (4.16), the meaning is that $\Phi_{1:t|t}$ is the product of the measure $\Phi^m_{1:t|t}$ and the kernel $\Phi^c_{1:t|t}$. In the remainder of this section we shall make frequent use of a Fubini like theorem for such products, see e.g. [Uglanov, 1991].

Algorithm 4.1 Particle filter (PF)

```
Input:  A weighted particle system $\{x^{N,i}_{1:t-1}, w^{N,i}_{t-1}\}_{i=1}^{N}$ targeting $\Phi_{1:t-1|t-1}$.
Output: A weighted particle system $\{x^{N,i}_{1:t}, w^{N,i}_{t}\}_{i=1}^{N}$ targeting $\Phi_{1:t|t}$.
Selection:
1: Generate an equally weighted particle system by multinomial resampling,
       $\mathrm{P}(J(i) = j \mid \{x^{N,k}_{1:t-1}, w^{N,k}_{t-1}\}_{k=1}^{N}) = w^{N,j}_{t-1}, \qquad i = 1, \ldots, N.$
2: Set $\tilde{x}^{N,i}_{1:t-1} = x^{N,J(i)}_{1:t-1}$ for $i = 1, \ldots, N$.
Mutation:
3: Sample new particle trajectories from a proposal kernel according to,
       $x^{N,i}_{1:t} \sim R_t(dx_{1:t} \mid \tilde{x}^{N,i}_{1:t-1}), \qquad i = 1, \ldots, N.$
4: Compute the unnormalised importance weights using the weight function (4.13),
       $w'^{N,i}_{t} \propto W_t(x^{N,i}_{1:t}), \qquad i = 1, \ldots, N$
5: Normalise the weights.
       $w^{N,i}_{t} = \frac{w'^{N,i}_{t}}{\sum_k w'^{N,k}_{t}}, \qquad i = 1, \ldots, N.$
```

Instead of running a PF targeting the "full" joint smoothing distribution, we have the option to target the state-marginal smoothing distribution $\Phi^m_{1:t|t}$ with an SMC sampler, and then make use of an analytical expression for $\Phi^c_{1:t|t}$. Hence, we choose a proposal kernel $R^m_t(d\xi_{1:t} \mid \tilde{\xi}_{1:t-1})$ from X^{t-1}_{ξ} to X^{t}_{ξ}, such that $\Phi^m_{1:t|t} \ll \pi^m_{1:t|t}$ and define a weight function $W^m_t(\xi_{1:t})$ analogously to (4.13). The measure $\pi^m_{1:t|t}$ on $\mathcal{X}^t_{\xi}$ is defined analogously to (4.12).

A weighted particle system $\{\xi^{N,i}_{1:t}, \omega^{N,i}_{t}\}_{i=1}^{N}$, targeting $\Phi^m_{1:t|t}$, can then be generated in the same manner as in Algorithm 4.1. We simply replace the weighted particle systems $\{x^{N,i}_{\cdot}, w^{N,i}_{\cdot}\}_{i=1}^{N}$ with $\{\xi^{N,i}_{\cdot}, \omega^{N,i}_{\cdot}\}_{i=1}^{N}$ and Φ_t, R_t, W_t with Φ^m_t, R^m_t, W^m_t, respectively (again, superscript m for marginal). The resulting filter is what we call the Rao-Blackwellised particle filter.

Remark 4.3. As pointed out in Section 3.3, the most common way to present the RBPF is for CLGSS models. In this case, the conditional joint smoothing distribution $\Phi^c_{1:t|t}$ is Gaussian and can be computed using the KF recursions. Consequently, the KF updates are often shown as intrinsic steps in the presentation of the RBPF algorithm. This was the case, e.g. for the RBPF presented in Algorithm 3.3, designed for mixed linear/nonlinear Gaussian SSMs. In this chapter, we adopt a more general view and simply see the RBPF as a "regular" SMC method targeting the state-marginal smoothing distribution $\Phi^m_{1:t|t}$. We then assume that the conditional distribution $\Phi^c_{1:t|t}$ is available by some means (for the CLGSS case, this would of course be by the KF), but it is not important for the results of this chapter what those means are. Hence, in the view adopted in this chapter, there is no fundamental difference between the PF and the RBPF. They are simply two SMC methods, targeting different distributions.

4.2.2 Problem formulation

The PF and the RBPF can both be used to estimate expectations under the joint smoothing distribution. Assume that we, for some function $f \in \mathsf{L}^1(\mathsf{X}^t, \Phi_{1:t|t})$, seek the expectation $\Phi_{1:t|t}(f)$. For the PF we use the natural estimator,

$$\hat{f}^N_{\text{PF}} \triangleq \sum_{i=1}^{N} w_t^{N,i} f(x_{1:t}^{N,i}). \tag{4.17}$$

For the RBPF we use the fact that $\Phi_{1:t|t}(f) = \Phi^m_{1:t|t}(\Phi^c_{1:t|t}(f))$, and define the estimator,

$$\hat{f}^N_{\text{RBPF}} \triangleq \sum_{i=1}^{N} \omega_t^{N,i} \Phi^c_{1:t|t}\left(f(\{\xi_{1:t}^{N,i}, \cdot\}) \;\middle|\; \xi_{1:t}^{N,i} \right). \tag{4.18}$$

The question then arise, how much better is (4.18) compared to (4.17)?

One analysis of this question, sometimes seen in the literature, is to simply consider a decomposition of variance,

$$\underbrace{\mathrm{Var}(f)}_{\text{PF}} = \underbrace{\mathrm{Var}(\mathrm{E}[f \mid \Xi_{1:t}])}_{\text{RBPF}} + \underbrace{\mathrm{E}[\mathrm{Var}(f \mid \Xi_{1:t})]}_{\geq 0}. \tag{4.19}$$

Here, the last term is claimed to be the variance reduction obtained in the RBPF. The decomposition is of course valid, the problem is that it does not answer our question. What we have in (4.19) is simply an expression for the variance of the test function f, it does not apply to the *estimators* (4.17) and (4.18).

Remark 4.4. It is not hard to see why the "simplified" analysis (4.19) has been considered. If the PF would produce i.i.d. samples from the target distribution (which it does not), then the analysis would be correct. More precisely, for i.i.d. samples, the central limit theorem states that the asymptotic variance of an estimator of a test function f, coincides with the variance of the test function itself (up to a factor $1/N$). However, as we have already pointed out, the PF does not produce i.i.d. samples. This is due to the selection step, in which a dependence between the particles is introduced. At the end of Section 4.2.5, one of the inadequacies of (4.19) will be pointed out.

Hence, we are interested in the asymptotic variances of (4.17) and (4.18). These are given in the following two theorems (slight modifications of what has previously been given by Douc and Moulines [2008]), in which we claim asymptotic normality of the weighted particle systems generated by the PF and the RBPF, respectively.

Theorem 4.5 (Asymptotic normality of the PF). *Assume that the initial particle system $\{x_1^{N,i}, w_1^{N,i}\}_{i=1}^N$ is asymptotically normal for $(\Phi_{1|1}, \mathsf{A}_1, \mathsf{W}_1, \sigma_1, \Phi_{1|1}, \{\sqrt{N}\})$. Define recursively the sets*

$$\mathsf{A}_t \triangleq \{f \in \mathsf{L}^2(\mathsf{X}^t, \Phi_{1:t|t}) : R_t(W_t f) \in \mathsf{A}_{t-1}, R_t(W_t^2 f^2) \in \mathsf{W}_{t-1}\}, \tag{4.20a}$$

$$\mathsf{W}_t \triangleq \{f \in \mathsf{L}^1(\mathsf{X}^t, \Phi_{1:t|t}) : R_t(W_t^2 |f|) \in \mathsf{W}_{t-1}\}. \tag{4.20b}$$

Assume that, for any $t \geq 1$, $R_{t+1}(W_{t+1}^2) \in \mathsf{W}_t$. Then, for any $t \geq 1$, the weighted particle system $\{x_{1:t}^{N,i}, w_t^{N,i}\}_{i=1}^N$ generated by the PF in Algorithm 4.1 is asymptotically normal for $(\Phi_{1:t|t}, \mathsf{A}_t, \mathsf{W}_t, \sigma_t, \Phi_{1:t|t}, \{\sqrt{N}\})$. The asymptotic variance is, for $f \in \mathsf{A}_t$, *given by*

$$\sigma_t^2(f) = \sigma_{t-1}^2\left(R_t(W_t\bar{f})\right) + \Phi_{1:t-1|t-1}\left[R_t\left((W_t\bar{f})^2\right)\right], \tag{4.21a}$$

$$\bar{f} = f - \Phi_{1:t|t}(f). \tag{4.21b}$$

Proof: The proof is given by induction and is a direct consequence of Theorem 4.3 and Theorem 4.4 (see also [Douc and Moulines, 2008], Theorem 10). □

Theorem 4.6 (Asymptotic normality of the RBPF). *Under analogous conditions and definitions as in Theorem 4.5, for any $t \geq 1$ the weighted particle system $\{\xi_{1:t}^{N,i}, \omega_t^{N,i}\}_{i=1}^N$ generated by the RBPF, is asymptotically normal for $(\Phi^m_{1:t|t}, \mathsf{A}^m_t, \mathsf{W}^m_t, \tau_t, \Phi^m_{1:t|t}, \{\sqrt{N}\})$. The asymptotic variance is, for $g \in \mathsf{A}^m_t$, given by*

$$\tau_t^2(g) = \tau_{t-1}^2\left(R_t^m(W_t^m\bar{g}) + \Phi^m_{1:t-1|t-1}\left[R_t^m\left((W_t^m\bar{g})^2\right)\right], \tag{4.22a}$$

$$\bar{g} = g - \Phi^m_{1:t|t}(g). \tag{4.22b}$$

Proof: The proof is given by induction and is a direct consequence of Theorem 4.3 and Theorem 4.4 (see also [Douc and Moulines, 2008], Theorem 10). □

Recall from Remark 4.3 that the PF and the RBPF are really just two SMC methods, targeting different distributions, hence the similarity between the two theorems above. Actually, we could have sufficed with one, more general, theorem applicable to both filters. The reason for why we have chosen to present them separately is for clarity and to introduce all the required notation.

As previously pointed out, the RBPF will intuitively produce better estimates than the PF, i.e. we expect $\tau_t^2(\Phi^c_{1:t|t}(f)) \leq \sigma_t^2(f)$. Let us therefore define the variance difference,

$$\Delta_t(f) \triangleq \sigma_t^2(f) - \tau_t^2(\Phi^c_{1:t|t}(f)). \tag{4.23}$$

Now, the problem that we are concerned with is to find an explicit expression for this quantity. This will be provided in the next section.

4.2.3 The main result

To analyse the variance difference (4.23) we will need the following assumption (similar to what is used by Chopin [2004]).

Assumption A1. For each $\tilde{\xi}_{1:t-1} \in \mathsf{X}_\xi^{t-1}$, the two measures

$$\int_{\mathsf{X}_z^{t-1}} R_t(dx_{1:t} \mid \{\tilde{\xi}_{1:t-1}, \tilde{z}_{1:t-1}\})\Phi^c_{1:t-1|t-1}(d\tilde{z}_{1:t-1} \mid \tilde{\xi}_{1:t-1}) \tag{4.24}$$

and

$$a_t(\xi_{1:t})\pi^c_{1:t|t}(dz_{1:t} \mid \xi_{1:t})R^m_t(d\xi_{1:t} \mid \tilde{\xi}_{1:t-1}) \tag{4.25}$$

agree on $\mathcal{X}^t$, for some positive function $a_t : \mathsf{X}_\xi^t \to \mathbb{R}_{++}$ and some transition kernel $\pi^c_{1:t|t}$ from X_ξ^t to X_z^t, for which $\Phi^c_{1:t|t}(\,\cdot \mid \xi_{1:t}) \ll \pi^c_{1:t|t}(\,\cdot \mid \xi_{1:t})$.

The basic meaning of this assumption is to create a connection between the proposal kernels R_t and R_t^m. It is natural that we need some kind of connection. Otherwise the asymptotic variance expressions (4.21a) and (4.22a) would be completely decoupled, and it would not be possible to draw any conclusions from a comparison. Still, as we shall see in the next section, Assumption A1 is fairly weak.

We are now ready to state the main result of this section.

Theorem 4.7. *Under Assumption A1, and using the definitions from Theorem 4.5 and Theorem 4.6, for any* $f \in \tilde{\mathsf{A}}_t$*,*

$$\Delta_t(f) = \Delta_{t-1}(R_t(W_t\bar{f})) + \Phi^m_{1:t-1|t-1}\left[R_t^m\left(\left(\tfrac{1-a_t}{a_t}\right)(W_t^m\bar{\psi})^2 + a_t\mathrm{Var}_{\pi^c_{1:t|t}}(W_t\bar{f})\right)\right], \tag{4.26}$$

where

$$\bar{\psi} = \Phi^c_{1:t|t}(f) - \Phi_{1:t|t}(f), \tag{4.27a}$$

$$\tilde{\mathsf{A}}_t = \{f \in \mathbb{F}(\mathsf{X}^t) : \Phi^c_{1:t|t}(f) \in \mathsf{A}_t^m\} \cap \mathsf{A}_t. \tag{4.27b}$$

Proof: See Appendix 4.A. □

4.2.4 Relationship between the proposal kernels

To understand the results given in the previous section, we shall have a closer look at the relationship between the proposal kernels imposed by Assumption A1. We shall do this for a certain family of proposal kernels. More precisely, similarly to (3.17) on page 39 we assume that the kernels can be written

$$R_t(dx_{1:t} \mid \tilde{x}_{1:t-1}) = r_t(dx_t \mid x_{1:t-1})\delta_{\tilde{x}_{1:t-1}}(dx_{1:t-1}), \tag{4.28a}$$

$$R_t^m(d\xi_{1:t} \mid \tilde{\xi}_{1:t-1}) = r_t^m(d\xi_t \mid \xi_{1:t-1})\delta_{\tilde{\xi}_{1:t-1}}(d\xi_{1:t-1}). \tag{4.28b}$$

Informally, this means that when a particle trajectory ($x_{1:t}^{N,i}$ or $\xi_{1:t}^{N,i}$) is sampled at time t, we keep the "old" trajectory up to time $t-1$ and simply append a sample from time t. As discussed in Section 3.2.3, this is the case for most PFs when targeting the joint smoothing distribution, but not all.

Furthermore, let r_t be factorised as

$$r_t(dx_t \mid x_{1:t-1}) = q_t^c(dz_t \mid \xi_{1:t}, z_{1:t-1})q_t^m(d\xi_t \mid \xi_{1:t-1}, z_{1:t-1}). \tag{4.29}$$

Assume that $q_t^m(\,\cdot \mid \xi_{1:t-1}, z_{1:t-1}) \ll r_{t-1}^m(\,\cdot \mid \xi_{1:t-1})$ for any $\{\xi_{1:t-1}, z_{1:t-1}\} \in \mathsf{X}^{t-1}$ and define the kernel

$$\begin{aligned}\nu_t(dz_{1:t} \mid \xi_{1:t}) \triangleq{} & \frac{dq_t^m(\,\cdot \mid \xi_{1:t-1}, z_{1:t-1})}{dr_t^m(\,\cdot \mid \xi_{1:t-1})}(\xi_t) \\ & \times q_t^c(dz_t \mid \xi_{1:t}, z_{1:t-1})\Phi^c_{1:t-1|t-1}(dz_{1:t-1} \mid \xi_{1:t-1}).\end{aligned} \tag{4.30}$$

It can now be verified that the choice

$$a_t(\xi_{1:t}) = \int_{\mathsf{X}_z^t} \nu_t(dz_{1:t} \mid \xi_{1:t}), \tag{4.31}$$

$$\pi^c_{1:t|t}(dz_{1:t} \mid \xi_{1:t}) = \frac{\nu_t(dz_{1:t} \mid \xi_{1:t})}{a_t(\xi_{1:t})}, \tag{4.32}$$

satisfies Assumption A1, given that $\Phi^c_{1:t|t}(\,\cdot \mid \xi_{1:t}) \ll \pi^c_{1:t|t}(\,\cdot \mid \xi_{1:t})$.

Hence, the function a_t takes the role of a normalisation of the kernel ν_t to obtain a transition kernel $\pi^c_{1:t|t}$. One interesting fact is that, from (4.26), we cannot guarantee that $\Delta_t(f)$ is nonnegative for arbitrary functions a_t. At first this might seem counterintuitive, since it would mean that the variance is higher for the RBPF than for the PF. The explanation lies in that Assumption A1, relating the proposal kernels in the two filters, is fairly weak. In other words, we have not assumed that the proposal kernels are "equally good". For instance, say that the optimal proposal kernel is used in the PF, whereas the RBPF uses a poor kernel. It is then no longer clear that the RBPF will outperform the PF. In the example below, we shall see that if both filters use their respective bootstrap proposal kernel, a case when the term "equally good" makes sense, then $\Delta_t(f)$ will indeed be nonnegative. However, for other proposal kernels, it is not clear that there is an analogue between the PF and the RBPF in the same sense.

Example 4.1: Bootstrap kernels

Let $Q(dx_t \mid x_{t-1})$ be the Markov transition kernel of the state process $\{X_t\}_{t\geq 1}$. In the bootstrap PF (see Definition 3.2) we choose the proposal kernel according to (4.28a) with

$$r_{t-1}(dx_t \mid x_{1:t-1}) = Q(dx_t \mid x_{t-1}), \tag{4.33}$$

where, for $A \in \mathcal{X}$,

$$Q(A \mid X_{t-1}) = \mathrm{P}(X_t \in A \mid X_{t-1}) = \mathrm{P}(X_t \in A \mid X_{1:t-1}, Y_{1:t-1}). \tag{4.34}$$

The second equality follows from the Markov property of the process. In the bootstrap RBPF (see Definition 3.4), we use a proposal kernel according to (4.28b) with

$$r^m_t(A \mid \Xi_{1:t-1}) = \mathrm{P}(\Xi_t \in A \mid \Xi_{1:t-1}, Y_{1:t-1}), \tag{4.35}$$

for $A \in \mathcal{X}_\xi$.

It can be then be verified that, for these choices of proposal kernels, Assumption A1 is fulfilled with,

$$a_t \equiv 1, \tag{4.36a}$$

and

$$\pi^c_{1:t|t}(A \mid \Xi_{1:t}) = \mathrm{P}(Z_{1:t} \in A \mid \Xi_{1:t}, Y_{1:t-1}), \tag{4.36b}$$

for $A \in \mathcal{X}_z^t$. Hence, $\pi^c_{1:t|t}$ is indeed the predictive distribution of $Z_{1:t}$ conditioned on $\Xi_{1:t}$ and based on the measurements up to time $t-1$. In this case we can also write $\pi_{1:t|t}(dx_{1:t}) = \pi^m_{1:t|t}(d\xi_{1:t})\pi^c_{1:t|t}(dz_{1:t} \mid \xi_{1:t})$, which highlights the connection between the predictive distributions in the two filters. In this case, due to (4.36a), the variance difference (4.26) can be simplified to

$$\Delta_t(f) = \Delta_{t-1}(R_t(W_t\bar{f})) + \Phi^m_{1:t-1|t-1}\left[R^m_t\left(\mathrm{Var}_{\pi^c_{1:t|t}}(W_t\bar{f})\right)\right]. \tag{4.37}$$

Hence, $\Delta_t(f)$ can be written as a sum (though, we have expressed it in a recursive form here) in which each term is an expectation of a conditional variance. It is thus ensured to be nonnegative.

4.2.5 Discussion

In Theorem 4.7 we gave an explicit expression for the difference in asymptotic variance between the PF and the RBPF. This expression can be used as a guideline for when it is beneficial to apply Rao-Blackwellisation, and when it is not. The variance expressions given in this chapters are asymptotic. Consequently, they do not apply exactly to the variances of the estimators (4.17) and (4.18), for a finite number of particles. Still, a reasonable approximation of the accuracy of the estimator (4.17) is,

$$\mathrm{Var}\left(\hat{f}^N_{\text{PF}}\right) \approx \frac{\sigma^2_t(f)}{N}, \tag{4.38}$$

and similarly for (4.18),

$$\mathrm{Var}\left(\hat{f}^N_{\text{RBPF}}\right) \approx \frac{\tau^2_t(\Phi^c_{1:t|t}(f))}{N}. \tag{4.39}$$

Now, assume that the computational effort required by the RBPF, using M particles, equals that required by the PF, using N particles (thus, $M < N$ since, in general, the RBPF is more computationally demanding than the PF per particle). We then have,

$$\frac{\mathrm{Var}\left(\hat{f}^N_{\text{PF}}\right)}{\mathrm{Var}\left(\hat{f}^M_{\text{RBPF}}\right)} \approx \frac{M}{N}\left(1 + \frac{\Delta_t(f)}{\tau^2_t(\Phi^c_{1:t|t}(f))}\right). \tag{4.40}$$

Whether or not this quantity is greater than one tells us if it is beneficial to use Rao-Blackwellisation. The crucial point is then to compute the ratio $\Delta_t(f)/\tau^2_t(\Phi^c_{1:t|t}(f))$, which in itself is a challenging problem. However, it is possible that this ratio can be efficiently estimated, e.g. from a single run of the RBPF.

As a final remark, for the special case discussed in Example 4.1, the variance difference (4.37) resembles the last term in the expression (4.19). They are both composed of an expectation of a conditional variance. One important difference though, is that the dependence on the weight function W_t is visible in (4.37). As an example, if the test function is restricted to $f \in \mathsf{L}^1(\mathsf{X}^t_\xi, \Phi^m_{1:t|t})$ the gain in variance indicated by (4.19) would be zero (since $\mathrm{Var}(f(\Xi_{1:t}) \mid \Xi_{1:t}) \equiv 0$), but this is not the case for the actual gain (4.37).

Appendix

4.A Proof of Theorem 4.7

Let Assumption A1 be satisfied. We shall start by determining the relationship between the weight functions W_t and W_t^m. Consider

$$\begin{aligned}\Phi_{1:t|t}(dx_{1:t}) &= \frac{d\Phi_{1:t|t}}{d\pi_{1:t|t}}(x_{1:t})\pi_{1:t|t}(dx_{1:t}) \\ &= W_t(x_{1:t}) \int\limits_{\mathsf{X}^{t-1}} R_t(dx_{1:t} \mid \tilde{x}_{1:t-1})\Phi_{1:t-1|t-1}(d\tilde{x}_{1:t-1}) \qquad (4.41)\end{aligned}$$

where we have made use of the definitions in (4.12) and (4.13). Furthermore, from the factorisation of $\Phi_{1:t-1|t-1}$ (4.16) and Assumption A1 we get,

$$\begin{aligned}&\Phi_{1:t|t}(dx_{1:t}) = W_t(x_{1:t}) \\ &\quad\times \int\limits_{\mathsf{X}_\xi^{t-1}} \Big(\Phi^m_{1:t-1|t-1}(d\tilde{\xi}_{1:t-1}) \int\limits_{\mathsf{X}_z^{t-1}} R_t(dx_{1:t} \mid \tilde{x}_{1:t-1})\Phi^c_{1:t-1|t-1}(d\tilde{z}_{1:t-1} \mid \tilde{\xi}_{1:t-1})\Big) \\ &= a_t(\xi_{1:t})W_t(x_{1:t}) \int\limits_{\mathsf{X}_\xi^{t-1}} \underbrace{\Phi^m_{1:t-1|t-1}(d\tilde{\xi}_{1:t-1})R_t^m(d\xi_{1:t} \mid \tilde{\xi}_{1:t-1})}_{\text{integrates to } \pi^m_{1:t|t}(d\xi_{1:t})} \pi^c_{1:t|t}(dz_{1:t} \mid \xi_{1:t}) \\ &= a_t(\xi_{1:t})W_t(x_{1:t})\pi^m_{1:t|t}(d\xi_{1:t})\pi^c_{1:t|t}(dz_{1:t} \mid \xi_{1:t}); \qquad (4.42)\end{aligned}$$

recall that $\pi^m_{1:t|t}$ is defined analogously to (4.12). However, we may also write

$$\Phi_{1:t|t} = \frac{d\Phi^m_{1:t|t}}{d\pi^m_{1:t|t}} \frac{d\Phi^c_{1:t|t}}{d\pi^c_{1:t|t}} \pi^m_{1:t|}\pi^c_{1:t|t}. \qquad (4.43)$$

Hence, we have two candidates for the Radon-Nikodym derivative of $\Phi_{1:t|t}$ with respect to $\pi^m_{1:t|t}\pi^c_{1:t|t}$ which, $\pi^m_{1:t|t}\pi^c_{1:t|t}$-a.s. implies,

$$a_t(\xi_{1:t})W_t(x_{1:t}) = W^m_t(\xi_{1:t})\frac{d\Phi^c_{1:t|t}(\,\cdot\mid\xi_{1:t})}{d\pi^c_{1:t|t}(\,\cdot\mid\xi_{1:t})}(z_{1:t}). \tag{4.44}$$

Now, consider an arbitrary $\varphi \in \tilde{\mathsf{A}}_t$. Using (4.16) and Assumption A1 we may write

$$\Phi_{1:t-1|t-1}\left[R_t(\varphi)\right] = \Phi^m_{1:t-1|t-1}\left[R^m_t\left(\,\cdot\,, a_t\pi^c_{1:t|t}(\varphi)\right)\right]. \tag{4.45}$$

Comparing (4.45) and (4.21a), we see that we can let φ take the role of $(W_t\bar{f})^2$. Hence, consider

$$\pi^c_{1:t|t}\left((W_t\bar{f})^2\right) = \left(\pi^c_{1:t|t}(W_t\bar{f})\right)^2 + \mathrm{Var}_{\pi^c_{1:t|t}}(W_t\bar{f}), \tag{4.46}$$

where, using (4.44) we have $\pi^m_{1:t|t}$-a.s.,

$$\begin{aligned}\pi^c_{1:t|t}(W_t\bar{f}) &= \int \frac{W^m_t(\xi_{1:t})}{a_t(\xi_{1:t})}\frac{d\Phi^c_{1:t|t}(\,\cdot\mid\xi_{1:t})}{d\pi^c_{1:t|t}(\,\cdot\mid\xi_{1:t})}(z_{1:t})\bar{f}(\{\xi_{1:t}, z_{1:t}\})\pi^c_{1:t|t}(dz_{1:t}\mid\xi_{1:t})\\ &= \frac{W^m_t(\xi_{1:t})}{a_t(\xi_{1:t})}\Phi^c_{1:t|t}(\bar{f}) = \frac{W^m_t(\xi_{1:t})}{a_t(\xi_{1:t})}\bar{\psi}(\xi_{1:t}).\end{aligned} \tag{4.47}$$

Here we have made use of the definition of $\bar{\psi}$ in (4.27a), yielding

$$\Phi^c_{1:t|t}(\bar{f}) = \Phi^c_{1:t|t}(f - \Phi_{1:t|t}(f)) = \bar{\psi}(\xi_{1:t}). \tag{4.48}$$

Combining (4.46) and (4.47) we get, $\pi^m_{1:t|t}$-a.s.,

$$a_t(\xi_{1:t})\pi^c_{1:t|t}\left((W_t\bar{f})^2\right) = \frac{\left(W^m_t(\xi_{1:t})\bar{\psi}(\xi_{1:t})\right)^2}{a_t(\xi_{1:t})} + a_t(\xi_{1:t})\mathrm{Var}_{\pi^c_{1:t|t}}(W_t\bar{f}). \tag{4.49}$$

Using (4.23), (4.21a), (4.22a) and the above results, the difference in asymptotic variance can now be expressed as,

$$\begin{aligned}\Delta_t(f) &= \sigma^2_{t-1}\left(R_t(W_t\bar{f})\right) - \tau^2_{t-1}\left(R^m_t(W^m_t\bar{\psi})\right)\\ &\quad + \Phi_{1:t-1|t-1}\left[R_t\left((W_t\bar{f})^2\right)\right] - \Phi^m_{1:t-1|t-1}\left[R^m_t\left((W^m_t\bar{\psi})^2\right)\right]\\ &= \sigma^2_{t-1}\left(R_t(W_t\bar{f})\right) - \tau^2_{t-1}\left(R^m_t(W^m_t\bar{\psi})\right)\\ &\quad + \Phi^m_{1:t-1|t-1}\left[R^m_t\left(\left(\tfrac{1}{a_t} - 1\right)(W^m_t\bar{\psi})^2 + a_t\mathrm{Var}_{\pi^c_{1:t|t}}(W_t\bar{f})\right)\right];\end{aligned} \tag{4.50}$$

recall that $\pi^m_{1:t|t} = \Phi^m_{1:t-1|t-1}R^m_t$ which ensures that we, due to the expectation w.r.t. $\Phi^m_{1:t-1|t-1}R^m_t$ in (4.50), can make use of the equality in (4.49).

Finally, consider

$$
\begin{aligned}
&\Phi^c_{1:t-1|t-1}(R_t(W_t\bar{f}))\\
&= \int_{\mathsf{X}^t} \Big(W_t(x_{1:t})\bar{f}(x_{1:t}) \int_{\mathsf{X}_z^{t-1}} R_t(dx_{1:t} \mid \tilde{x}_{1:t-1})\Phi^c_{1:t-1|t-1}(d\tilde{z}_{1:t-1} \mid \tilde{\xi}_{1:t-1})\Big)\\
&= \int_{\mathsf{X}_\xi^t}\int_{\mathsf{X}_z^t} a_t(\xi_{1:t})W_t(x_{1:t})\bar{f}(x_{1:t})R_t^m(d\xi_{1:t} \mid \tilde{\xi}_{1:t-1})\pi^c_{1:t|t}(dz_{1:t} \mid \xi_{1:t})\\
&= \int_{\mathsf{X}_\xi^t} \left(W_t^m(\xi_{1:t})R_t^m(d\xi_{1:t} \mid \tilde{\xi}_{1:t-1}) \int_{\mathsf{X}_z^t} \bar{f}(x_{1:t}) \frac{d\Phi^c_{1:t|t}(\,\cdot \mid \xi_{1:t})}{d\pi^c_{1:t|t}(\,\cdot \mid \xi_{1:t})}(z_{1:t})\pi^c_{1:t|t}(dz_{1:t} \mid \xi_{1:t}) \right)\\
&= R^m_{t-1}(W_t^m\Phi^c_{1:t|t}(\bar{f})) = R_t^m(W_t^m\bar{\psi}), \qquad \pi^m_{1:t|t}\text{-a.s.}
\end{aligned}
\tag{4.51}
$$

The second equality follows from Assumption A1 and the third follows $\pi^m_{1:t|t}\pi^c_{1:t|t}$-a.s. from (4.44). Hence,

$$
\sigma^2_{t-1}\left(R_t(W_t\bar{f})\right) - \tau^2_{t-1}\left(R_t^m(W_t^m\bar{\psi})\right) = \Delta_{t-1}(R_t(W_t\bar{f})),
\tag{4.52}
$$

which completes the proof. □

5

Particle smoothing

As discussed in Section 3.2.2, the particle filter (PF) suffers from degeneration of the particle trajectories. Due to this, it does in general not provide accurate approximations of the various smoothing distributions that we might be interested in, other than for fixed-lag smoothing with a short enough lag (see Table 2.1 on page 18). One way to get around this problem is to complement the forward filter with a second recursion, evolving in the time-reversed direction. From this idea, two competing methods have evolved.

The first is known as a the two-filter approach, since it is based on one filter moving forward in time and one filter moving backward in time. When the two filters meet "somewhere in the middle", the information is merged, enabling computation of smoothed estimates. The second approach is based on the forward/backward recursions presented in Section 2.3. Here, we start by applying a forward filter to the entire data sequence. Once this is complete, it is supplemented with a backward smoothing pass, in which the output from the forward filter is updated. The focus of the present chapter is on particle based forward filtering/backward smoothing, i.e. the second approach mentioned above. For further reference regarding two-filter smoothing, see e.g. [Briers et al., 2010, Fearnhead et al., 2010].

The main contribution of this chapter lies in Section 5.3, where a novel Rao-Blackwellised particle smoother (RBPS) is derived.

5.1 Forward filter/backward smoother

The forward/backward approach to smoothing can be used to supplement the PF with a backward recursion, enabling approximations of the marginal as well as the joint smoothing distributions. This is the topic of the present section. We will make use of the backward recursion presented in Section 2.3.2. The key ingredient here is the backward kernel defined by (2.22a) on page 21. To allow for explicit expressions in what follows, we shall assume that the model is fully dominated and make use of the explicit expression for the density of the backward kernel given by (2.22b) on page 21.

As previously pointed out, the backward kernel depends on the filtering distribution. The basic idea underlying particle based forward filtering/backward smoothing (FFBSm), is to make use of the PF approximation of the filtering distribution to approximate the backward kernel. This is also the reason for why the FFBSm can succeed where the PF fails. Even though the PF provides poor approximations of the marginal and joint smoothing distributions, it can generally provide accurate approximations of the filtering distribution, which is all that is needed to compute the backward kernel. Now, let us assume that a PF (or some similar method) has been applied to a measurement sequence $y_{1:T}$ of fixed length T. For each time $t = 1, \dots, T$, we have obtained a weighted particle system $\{x_t^i, w_t^i\}_{i=1}^N$, approximating the filtering distribution at time t with an empirical distribution according to,

$$\widehat{\Phi}_{t|t}^N(dx_t) = \sum_{i=1}^N w_t^i \delta_{x_t^i}(dx_t). \tag{5.1}$$

By the distribution given above and the expression for the backward kernel density (2.22b), we can approximate the backward kernel by,

$$\widehat{B}_t^N(dx_t \mid x_{t+1}) \triangleq \sum_{i=1}^N \frac{w_t^i p(x_{t+1} \mid x_t^i)}{\sum_k w_t^k p(x_{t+1} \mid x_t^k)} \delta_{x_t^i}(dx_t). \tag{5.2}$$

In the coming sections, we will see how this approximation can be used to transform the output from the PF, to instead target the various smoothing distributions of interest.

5.1.1 FFBSm for joint smoothing

By expanding the backward recursion (2.23) on page 21, the joint smoothing distribution can be expressed as,

$$\Phi_{1:T|T}(dx_{1:T}) = \left(\prod_{t=1}^{T-1} B_t(dx_t \mid x_{t+1}) \right) \Phi_{T|T}(dx_T). \tag{5.3}$$

An empirical filtering distribution at time T is given by (5.1), and the backward kernels can be approximated by (5.2). By plugging these approximations into (5.3), an empirical joint smoothing distribution is given by,

$$\begin{aligned}\widehat{\Phi}_{1:T|T}^N(dx_{1:T}) &\triangleq \sum_{i_1=1}^N \cdots \sum_{i_T=1}^N \underbrace{\left(\prod_{t=1}^{T-1} \frac{w_t^{i_t} p(x_{t+1}^{i_{t+1}} \mid x_t^{i_t})}{\sum_k w_t^k p(x_{t+1}^{i_{t+1}} \mid x_t^k)} \right) w_T^{i_T}}_{\triangleq w_{1:T|T}(i_1,\dots,i_T)} \delta_{x_1^{i_1} \cdots x_T^{i_T}}(dx_{1:T}) \\ &= \sum_{i_1=1}^N \cdots \sum_{i_T=1}^N w_{1:T|T}(i_1, \dots, i_T) \delta_{x_1^{i_1} \cdots x_T^{i_T}}(dx_{1:T}). \end{aligned} \tag{5.4}$$

The expression above defines a point-mass distribution on the space X^T, and the cardinality of its support is N^T. The meaning of the distribution can be understood in the following way. For each time $t = 1, \ldots, T$, the particles $\{x_t^i\}_{i=1}^N$ generated by the PF is a set in the space X of cardinality N. By "picking" one particle from each time index, we obtain a particle trajectory, i.e. a point in the space X^T,

$$\{x_1^{i_1}, \ldots, x_T^{i_T}\} \in \mathsf{X}^T. \tag{5.5}$$

By letting all of the indices $i_1, i_2, \ldots, i_T$ vary from 1 to N, we get in total N^T such trajectories. The empirical distribution (5.4) assigns, to each such trajectory, a probability $w_{1:T|T}(i_1, \ldots, i_T)$.

Even though (5.4) provides a closed form approximation of the joint smoothing distribution, it is impractical for any real problem of interest. The reason is of course, that evaluating the discrete probabilities of the distribution is an $O(N^T)$ operation, both in terms of computational complexity and storage. However, even though it is not practically applicable on its own, the distribution (5.4) still provides interesting means for approximating the joint smoothing distribution. We will return to this in Section 5.2, but before that we turn our attention to the marginal smoothing problem.

5.1.2 FFBSm for marginal and fixed-interval smoothing

As pointed out above, evaluating the distribution (5.4) is in general not feasible, since the cardinality of its support is N^T. However, assume that we are only interested in the sequence of marginal smoothing distributions $\Phi_{t|T}$, for $t = 1, \ldots, T$. Based on the particles from the forward pass of the PF, each marginal smoothing distribution will be approximated by a weighted particle system with no more than N particles. We thus expect that the computational complexity of approximating this sequence, is lower than what is required to evaluate (5.4).

To see that this indeed is the case, we shall make use of the backward recursion for the marginal smoothing distribution given by (2.24) on page 21. Assume that we have available a weighted particle system $\{x_{t+1}^j, w_{t+1|T}^j\}_{j=1}^N$, targeting $\Phi_{t+1|T}$. At time $t = T-1$, this is provided by the PF by letting $w_{T|T}^j := w_T^j$ for $j = 1, \ldots, N$. Plugging the empirical distribution defined by this particle system, and the approximation of the backward kernel (5.2), into the recursion (2.24) results in,

$$\widehat{\Phi}_{t|T}^N(dx_t) \triangleq \sum_{j=1}^N w_{t+1|T}^j \sum_{i=1}^N \frac{w_t^i p(x_{t+1}^j \mid x_t^i)}{\sum_k w_t^k p(x_{t+1}^j \mid x_t^k)} \delta_{x_t^i}(dx_t) = \sum_{i=1}^N w_{t|T}^i \delta_{x_t^i}(dx_t), \tag{5.6a}$$

where we have defined the smoothing weights,

$$w_{t|T}^i \triangleq w_t^i \sum_{j=1}^N w_{t+1|T}^j \frac{p(x_{t+1}^j \mid x_t^i)}{\sum_k w_t^k p(x_{t+1}^j \mid x_t^k)}. \tag{5.6b}$$

Remark 5.1. Note that the smoothing weights are "self-normalised" since,

$$\sum_{i=1}^N w_{t|T}^i = \sum_{j=1}^N w_{t+1|T}^j \frac{\sum_i w_t^i p(x_{t+1}^j \mid x_t^i)}{\sum_k w_t^k p(x_{t+1}^j \mid x_t^k)} = 1.$$

Algorithm 5.1 Marginal FFBSm [Doucet et al., 2000b]

Input: A sequence of weighted particle systems $\{x_t^i, w_t^i\}_{i=1}^N$ targeting the filtering distributions $\Phi_{t|t}$, for $t = 1, \ldots, T$.

Output: A sequence of weighted particle systems $\{x_t^i, w_{t|T}^i\}_{i=1}^N$ targeting the marginal smoothing distributions $\Phi_{t|T}$, for $t = 1, \ldots, T$.

```
Initialise the smoothing weights. For $i = 1, \ldots, N$, set $w_{T|T}^i = w_T^i$.
for $t = T - 1$ to $1$ do
    For $j = 1, \ldots, N$, set $v_t^j = \sum_{k=1}^N w_t^k p(x_{t+1}^j \mid x_t^k)$.
    For $i = 1, \ldots, N$, compute the smoothing weights,
        $$w_{t|T}^i = w_t^i \sum_{j=1}^N w_{t+1|T}^j \frac{p(x_{t+1}^j \mid x_t^i)}{v_t^j}.$$
end for
```

Hence, we have obtained a weighted particle system $\{x_t^i, w_{t|T}^i\}_{i=1}^N$, that targets $\Phi_{t|T}$. This smoother, targeting the sequence of marginal smoothing distributions, has previously been proposed by Doucet et al. [2000b]. We summarise the steps of the procedure in Algorithm 5.1. The complexity of this algorithm is $O(N^2(T-1))$, which is a significant reduction from the $O(N^T)$ complexity of evaluating the full empirical joint smoothing distribution (5.4). Still, a computational cost growing quadratically with the number of particles, might be prohibitive for certain problems.

Remark 5.2. Note that the marginal FFBSm keeps the forward filter particles $\{x_t^i\}_{i=1}^N$ unchanged. It simply updates the importance weights of these particles, to target the marginal smoothing distribution rather than the filtering distribution. The same goes for all "versions" of the FFBSm presented in this section. The smoothing consists of computing new weights, and does not "move" the particles generated by the forward filter.

For many problems (see e.g. Section 6.3), it is not sufficient to work with the marginal smoothing distributions. Instead, we seek to approximate some fixed-interval smoothing distribution $\Phi_{s:t|T}$ for $s < t$ (not to be confused with fixed-lag smoothing, see Table 2.1). This can be obtained in a similar manner as above, based on the expression (2.25) on page 21. By plugging (5.2) and (5.6a) into (2.25), we get

$$\widehat{\Phi}_{s:t|T}^N(dx_{s:t}) \triangleq \sum_{i_s=1}^N \cdots \sum_{i_t=1}^N \underbrace{\left(\prod_{u=s}^{t-1} \frac{w_u^{i_u} p(x_{u+1}^{i_{u+1}} \mid x_u^{i_u})}{\sum_k w_u^k p(x_{u+1}^{i_{u+1}} \mid x_u^k)} \right) w_{t|T}^{i_t}}_{\triangleq w_{s:t|T}(i_s, \ldots, i_t)} \delta_{x_s^{i_s} \cdots x_t^{i_t}}(dx_{s:t}). \tag{5.7}$$

This can be seen as a reduced version of (5.4), and by taking $s = 1$ and $t = T$, we indeed recover the same expression. Let $\ell = t - s + 1$ be the length of the interval. Then, the complexity of evaluating the fixed-interval smoothing weights $w_{s:t|T}$ is $O(N^\ell)$. If this is done for all $t = \ell, \ldots, T$, the total complexity of performing fixed-interval smoothing is $O(N^\ell(T-\ell+1)+N^2(T-\ell))$. This includes the computation of the marginal smoothing weights according to (5.6b).

Algorithm 5.2 Two-step fixed-interval FFBSm

Input: A sequence of weighted particle systems $\{x_t^i, w_t^i\}_{i=1}^N$ targeting the filtering distributions $\Phi_{t|t}$, for $t = 1, \ldots, T$.

Output: A sequence of weighted particle systems $\{x_{t:t+1}^{i,j}, w_{t:t+1|T}^{i,j}\}_{i,j=1}^N$ targeting the 2-step, fixed-interval smoothing distributions $\Phi_{t:t+1|T}$, for $t = 1, \ldots, T-1$.

```
Initialise the smoothing weights. For $i = 1, \ldots, N$, set $w_{T|T}^i = w_T^i$.
for $t = T-1$ to $1$ do
  for $j = 1$ to $N$ do
    Set $v_t^j = \sum_{k=1}^N w_t^k p(x_{t+1}^j \mid x_t^k)$.
    For $i = 1, \ldots, N$, compute the 2-step smoothing weights,
        $$w_{t:t+1|T}^{i,j} = w_t^i w_{t+1|T}^j \frac{p(x_{t+1}^j \mid x_t^i)}{v_t^j}.$$
  end for
  Compute the marginal smoothing weights. For $i = 1, \ldots, N$,
        $$w_{t|T}^i = \sum_{j=1}^N w_{t:t+1|T}^{i,j}.$$
end for
```

We illustrate the fixed-interval FFBSm in Algorithm 5.2, for the special case of 2-step fixed-interval smoothing. The reason for this restriction is to simplify the notation. Also, 2-step smoothing is a common special case of the fixed-interval smoothing problem, used e.g. by Schön et al. [2011] (see also Section 6.3).

5.2 Forward filter/backward simulator

Another approach to particle smoothing, very much related to FFBSm, is forward filtering/backward simulation (FFBSi). This can be seen as a modification of the FFBSm, in which the backward recursion is done by random sampling. The drawback with this is that the backward sampling introduces extra variance in the estimators. However, by applying backward simulation, we can target the joint smoothing distribution with a computational cost which is significantly lower than for the FFBSm. We start in Section 5.2.1 by reviewing the original FFBSi formulation, which has $O(TN^2)$ complexity. In Section 5.2.2, we then turn to a novel modification of the FFBSi, resulting in an equivalent algorithm, which under appropriate assumptions can be shown to reach linear complexity in the number of particles.

5.2.1 Standard FFBSi

Let us return to the empirical approximation of the joint smoothing distribution (5.4). As previously pointed out, evaluating the discrete probabilities of this distribution is impractical for any problem of interest. However, an idea proposed by Doucet et al. [2000c] and Godsill et al. [2004], is to draw sample trajectories from this empirical distribution, and in that way approximate the joint smoothing distribution. Clearly, this can not be done by simple evaluation of the discrete probabilities, since again, this has $O(N^T)$ complexity.

The key to circumvent this problem lies in the construction of the distribution (5.4). Recall that this was done by plugging (5.1) and (5.2) into (5.3), which in turn could be traced back to (2.23) on page 21. Analogously, we can plug the particle approximations into the backward recursion (2.23) directly, resulting in,

$$\widehat{\Phi}^N_{t:T|T}(dx_{t:T}) = \widehat{B}^N_t(dx_t \mid x_{t+1})\widehat{\Phi}^N_{t+1:T|T}(dx_{t+1:T}). \tag{5.8}$$

We note that the above expression defines the evolution of a Markov chain. More clearly phrased, (5.4) is the terminal distribution of a time-reversed, inhomogeneous Markov chain with initial distribution $\widehat{\Phi}^N_{T|T}(dx_T)$ and transition kernel $\widehat{B}^N_t(dx_t \mid x_{t+1})$. Hence, sampling from the distribution (5.4) is equivalent to sampling a trajectory from the Markov chain given by (5.8). Recall that the forward filtering pass is assumed to be completed, for $t = 1, \ldots, T$, and the sequence of weighted particle systems generated by the forward filter can thus be seen as fixed. Hence, let $\mathcal{F}^N_{1:T}$ be the σ-algebra generated by the measurement sequence $Y_{1:T}$ and the full collection of weighted particle systems produced by the forward filter, i.e.

$$\mathcal{F}^N_{1:T} = \sigma\left(Y_{1:T}, \{\{x^i_t, w^i_t\}^N_{i=1}, t = 1, \ldots, T\}\right). \tag{5.9}$$

Now, given $\mathcal{F}^N_{1:T}$, assume that $\{\tilde{x}^j_{t+1:T}\}^M_{j=1}$ is a set of i.i.d.[1] samples from the empirical distribution $\widehat{\Phi}^N_{t+1:T|T}$. To each such *backward trajectory*, we then append a sample from the kernel (5.2),

$$\tilde{x}^j_t \sim \widehat{B}^N_t(dx_t \mid \tilde{x}^j_{t+1}), \tag{5.10a}$$

$$\tilde{x}^j_{t:T} := \{\tilde{x}^j_t, \tilde{x}^j_{t+1:T}\}, \tag{5.10b}$$

for $j = 1, \ldots, M$. At time $t = 1$, the backward trajectories $\{\tilde{x}^j_{1:T}\}^M_{j=1}$ are i.i.d. samples from the empirical joint smoothing distribution (5.4). Sampling from the empirical backward kernel according to (5.10a) can be done straightforwardly, since it has finite support with cardinality N,

$$\widehat{B}^N_t(dx_t \mid \tilde{x}^j_{t+1}) = \sum_{i=1}^N \underbrace{\frac{w^i_t p(\tilde{x}^j_{t+1} \mid x^i_t)}{\sum_{k=1}^N w^k_t p(\tilde{x}^j_{t+1} \mid x^k_t)}}_{\triangleq \tilde{w}^{i,j}_{t|T}} \delta_{x^i_t}(dx_t). \tag{5.11}$$

Hence, to sample from the kernel, we only need to compute the N discrete probabilities $\{\tilde{w}^{i,j}_{t|T}\}^N_{i=1}$ and set $\tilde{x}^j_t := x^i_t$ with probability $\tilde{w}^{i,j}_{t|T}$ (see also Algorithm 5.3). The procedure (5.10) is called backward simulation, and the resulting particle smoother is known as a forward filter/backward simulator (FFBSi). Similarly to the PF and the FFBSm, the FFBSi defines an empirical approximation of the joint smoothing distribution, according to,

$$\widetilde{\Phi}^M_{1:T|T}(dx_{1:T}) \triangleq \frac{1}{M}\sum_{j=1}^M \delta_{\tilde{x}^j_{1:T}}(dx_{1:T}). \tag{5.12}$$

[1] When using the phrase i.i.d. in the context of backward simulation, we implicitly mean i.i.d. given the σ-algebra $\mathcal{F}^N_{1:T}$.

As opposed to the FFBSm, the sample trajectories in the FFBSi are unweighted. This is basically due to the fact that the backward simulation produces i.i.d. samples from the empirical joint smoothing distribution and no weighting is therefore needed.

To further motivate the validity of the FFBSi, we make the following observation. Let $\varphi : \mathsf{X}^T \to \mathbb{R}$ be some test function, of which we seek to compute the expectation under the joint smoothing distribution, i.e. we seek $\Phi_{1:T|T}(\varphi)$. We assume that an analytical evaluation of this expectation is intractable, and a particle method is employed to estimate it. By (5.4), the FFBSm estimator is given by,

$$\widehat{\Phi}^N_{1:T|T}(\varphi) = \sum_{i_1=1}^{N} \cdots \sum_{i_T=1}^{N} w_{1:T|T}(i_1, \dots, i_T)\varphi(x_1^{i_1}, \dots, x_T^{i_T}). \tag{5.13}$$

As previously pointed out, this estimator is, unfortunately, also intractable. However, the FFBSi provides an unbiased estimator of (5.13), i.e.

$$\widehat{\Phi}^N_{1:T|T}(\varphi) = \mathrm{E}\left[\widetilde{\Phi}^M_{1:T|T}(\varphi) \;\middle|\; \mathcal{F}^N_{1:T}\right], \tag{5.14}$$

where $\widetilde{\Phi}^M_{1:T|T}(\varphi) = \sum_{j=1}^{M} \varphi(\tilde{x}^j_{1:T})$. Here, the expectation is taken w.r.t. the random components of the backward simulation.

Remark 5.3. From the relation (5.14), we see that the FFBSm estimator in fact can be seen as a Rao-Blackwellised version of the FFBSi estimator, or the other way around, FFBSi is an “anti-Rao-Blackwellisation” of FFBSm. Rao-Blackwellisation often aims at reducing the variance of an estimator, but generally at the cost of increased computational complexity. Here, we go the other way, and (significantly) reduce the complexity of the FFBSm estimator by instead employing FFBSi. However, due to the randomness of the backward simulation, this will also increase the variance of the estimator.

Remark 5.4. The term Rao-Blackwellisation, as used above, should not be confused with Rao-Blackwellisation of the PF, as described in Section 3.3, and more importantly Rao-Blackwellisation of the FFBSi, as we shall discuss in Section 5.3. In Remark 5.3, we spoke of “Rao-Blackwellisation w.r.t. the random components of the backward simulation”. In the RBPF and the RB-FFBSi (see Section 5.3), we instead deal with “Rao-Blackwellisation w.r.t. one partition of the state variable”.

We summarise the FFBSi in Algorithm 5.3, and illustrate the backward simulation procedure in Example 5.1 below. Also, note that the FFBSi targets the joint smoothing distribution, but (since they are marginals thereof) it can also be used to estimate expectations under the marginal or fixed-interval smoothing distributions.

Example 5.1: Backward simulation

We illustrate the backward simulation using an example with synthetic data. In Figure 5.1, we show the particle trajectories generated by a forward PF in a one-dimensional experiment. The dots show the particle positions for the $N = 4$ particles over $T = 5$ time steps and their sizes represent the particle weights. The dots are connected, to illustrate the ancestral dependence of the particles. All particles at time $t = 5$ share a common ancestor at time $t = 2$, i.e. the particle trajectories have degenerated.

In Figure 5.2 we show the simulation of a single backward trajectory. In the upper left plot, the backward trajectory is initiated by sampling from the forward filter particles at

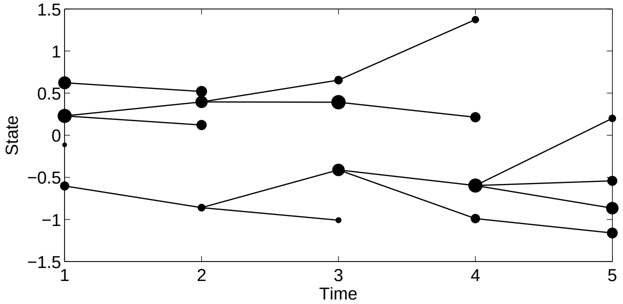

Figure 5.1: *Particle trajectories for $N = 4$ particles over $T = 5$ time steps after a completed forward filtering pass. The sizes of the dots represent the particle weights.*

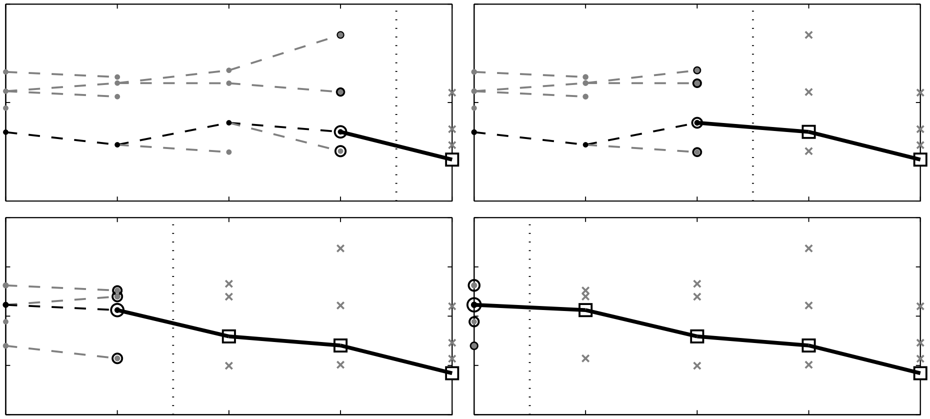

Figure 5.2: *Backward simulation of a single backward trajectory. See the text for details.*

time $t = 5$. The probability of sampling a particle is given by its importance weight. The initiated backward trajectory is shown as a square. The particle weights at $t = 4$ are thereafter recomputed according to Row 5 in Algorithm 5.3. The smoothing weights are shown as circles, whereas the filter weights are illustrated with dots. Another particle is then drawn and added to the backward trajectory. In the upper right and lower left plots, the trajectory is appended with new particles at $t = 3$ and $t = 2$, respectively. Finally, in the lower right plot, a final particle is appended at $t = 1$, forming a complete backward trajectory. Observe that the backward trajectory differs from the ancestral line of the forward filter particle as shown in Figure 5.1.

As can be seen in Algorithm 5.3, the smoothing weights in the FFBSi need to be computed for index i ranging from 1 to N and index j ranging from 1 to M. Hence, the total complexity of the algorithm is $O(NMT)$. The number of backward trajectories M, generated by Algorithm 5.3 is arbitrary. However, to obtain accurate MC integration from these tra-

Algorithm 5.3 Standard FFBSi [Godsill et al., 2004]

Input: A sequence of weighted particle systems $\{x_t^i, w_t^i\}_{i=1}^N$ targeting the filtering distributions $\Phi_{t|t}$, for $t = 1, \ldots, T$.

Output: A collection of backward trajectories $\{\tilde{x}_{1:T}^j\}_{j=1}^M$ targeting the joint smoothing distribution $\Phi_{1:T|T}$.

```
Initialise the backward trajectories. For $j = 1, \ldots, M$,
    $I(j) \sim \mathrm{Cat}\left(\{w_T^i\}_{i=1}^N\right)$,
    $\tilde{x}_T^j = x_T^{I(j)}$.
for $t = T - 1$ to $1$ do
    for $j = 1$ to $M$ do
        Set $v_t^j = \sum_{k=1}^N w_t^k p(\tilde{x}_{t+1}^j \mid x_t^k)$.
        Compute the smoothing weights. For $i = 1, \ldots, N$,
            $\tilde{w}_{t|T}^{i,j} = w_t^i \frac{p(\tilde{x}_{t+1}^j \mid x_t^i)}{v_t^j}$.
        Sample from the empirical backward kernel,
            $I(j) \sim \mathrm{Cat}\left(\{\tilde{w}_{t|T}^{i,j}\}_{i=1}^N\right)$,
            $\tilde{x}_t^j = x_t^{I(j)}$.
        Append the sample to the backward trajectory, $\tilde{x}_{t:T}^j = \{\tilde{x}_t^j, \tilde{x}_{t+1:T}^j\}$.
    end for
end for
```

jectories, it is clear that M should be large. Assume, for instance, that the FFBSi is used to estimate the expectation of some test function under the marginal smoothing distribution. To get similar performance as the marginal FFBSm of Algorithm 5.1, we expect that $M \approx N$ backward trajectories (or rather $M > N$, see the comments below) are needed. Hence, the computational complexity of the FFBSi is comparable with that of the marginal and 2-step fixed-interval FFBSm. If fixed-interval smoothing, with an interval length of three or more, is required, the complexity of the FFBSi will be significantly lower than that of the FFBSm. Furthermore, and more importantly, it is possible to reduce the computational cost of the FFBSi to grow only linearly with the number of backward trajectories drawn. How to achieve this will be the topic of Section 5.2.2.

As argued before, the backward trajectories $\{\tilde{x}_{1:T}^j\}_{j=1}^M$ are i.i.d. samples from the empirical joint smoothing distribution (5.4). Also, if we extract particles from the backward trajectories at a single time point t, i.e. $\{\tilde{x}_t^j\}_{j=1}^M$, we are left with i.i.d. samples from the empirical marginal smoothing distribution (5.6a). Let us again assume that the FFBSi is employed to solve the marginal smoothing problem, and assume that we draw $M = N$ backward trajectories. Then, using the FFBSi to generate an (unweighted) particle system targeting $\Phi_{t|T}$, is equivalent to conducting multinomial resampling of the weighted particle system generated by the marginal FFBSm. Hence, for $M = N$, the variance of an FFBSi estimator will in general be larger than that of the corresponding FFBSm estimator (see also Remark 5.3). However, when put in relation to the fast implementation of the FFBSi, which we will discuss in the next section, this is a minor drawback and the FFBSi is indeed a strong alternative also for marginal smoothing. In fact, for a fixed computational cost, the FFBSi can for many problems produce better estimates than the FFBSm. This is possible since, by using the fast implementation of the FFBSi, we can afford a larger number of particles already in the forward filter.

5.2.2 Fast FFBSi

There exist several different approaches to reduce the computational complexity of particle smoothing, based on both numerical approximations and algorithmic modifications. Klaas et al. [2006] employ so called N-body algorithms to reduce the complexity of the marginal FFBSm to $O(TN \log N)$. These methods impose further approximations, though the tolerance can usually be specified beforehand. There have also been developments based on the two-filter algorithm by Briers et al. [2010]. In its original formulation, this method is quadratic in the number of particles. However, Fearnhead et al. [2010] have proposed a modified two-filter algorithm with linear complexity. Another idea by Fearnhead [2005], is to use quasi-random numbers for particle smoothing. The smoother proposed by Fearnhead [2005], which is restricted to one-dimensional problems, has quadratic complexity, but at the same time a quadratic decrease in variance, thanks to the use of quasi-random numbers. However, here we will focus on the fast FFBSi proposed by Douc et al. [2010]. This algorithm is equivalent to the original FFBSi by Godsill et al. [2004], presented above, but has a complexity growing only linearly with the number of particles.

The key insight made by Douc et al. [2010], is that we do not need to evaluate all the smoothing weights $\{\tilde{w}_{t|T}^{i,j}\}_{i=1}^{N}$ to be able to sample from the empirical backward kernel (5.11). To convince ourselves that there indeed is room for improvements here, note that we in the FFBSi in Algorithm 5.3 evaluate N smoothing weights in Step 5, draw a single sample from the categorical distribution in Step 6 and then discard all the weights. Instead of making this full evaluation of the categorical distribution, we can take a rejection sampling approach (see Section 2.4.3). For this to be applicable, we shall assume that the transition density function is bounded from above,

$$p(x_{t+1} \mid x_t) \leq \rho, \qquad x_{t+1}, x_t \in \mathsf{X}, \tag{5.15}$$

which is true for many commonly used density functions.

Now, we wish to sample an index $I(j)$, corresponding to the forward filter particle which is to be appended to the j:th backward trajectory. The target distribution is categorical over the index space $\{1, \ldots, N\}$, with probabilities $\{\tilde{w}_{t|T}^{i,j}\}_{i=1}^{N}$ (which we have not computed yet). As proposal, we take another categorical distribution over the same index space, with (known) probabilities $\{w_t^i\}_{i=1}^{N}$. That is, we propose samples based on the filter weights, rather than the smoothing weights. Now, assume that a sample index $I(j)$ is proposed for the j:th backward trajectory. To compute the acceptance probability, we consider the quotient between the target and the proposal, as in (2.37) on page 25. Using

Algorithm 5.4 Fast FFBSi [Douc et al., 2010]

Input: A sequence of weighted particle systems $\{x_t^i, w_t^i\}_{i=1}^N$ targeting the filtering distributions $\Phi_{t|t}$, for $t = 1, \ldots, T$.

Output: A collection of backward trajectories $\{\tilde{x}_{1:T}^j\}_{j=1}^M$ targeting the joint smoothing distribution $\Phi_{1:T|T}$.

```
1:  Initialise the backward trajectories,
        $\{I(j)\}_{j=1}^M \sim \mathrm{Cat}\left(\{w_T^i\}_{i=1}^N\right),$
        $\tilde{x}_T^j = x_T^{I(j)}, \qquad j = 1, \ldots, M.$
2:  for $t = T - 1$ to $1$ do
3:    $L \leftarrow \{1, \ldots, M\}$
4:    while $L$ is not empty do
5:      $n \leftarrow \mathrm{card}(L)$.
6:      $\delta \leftarrow \emptyset$.
7:      Sample independently $\{C(k)\}_{k=1}^n \sim \mathrm{Cat}(\{w_t^i\}_{i=1}^N)$.
8:      Sample independently $\{U(k)\}_{k=1}^n \sim \mathcal{U}([0,\, 1])$.
9:      for $k = 1$ to $n$ do
10:       if $U(k) \leq p(\tilde{x}_{t+1}^{L(k)} \mid x_t^{C(k)})/\rho$ then
11:         $I(L(k)) \leftarrow C(k)$.
12:         $\delta \leftarrow \delta \cup \{L(k)\}$.
13:       end if
14:     end for
15:     $L \leftarrow L \setminus \delta$.
16:   end while
17:   Append the samples to the backward trajectories. For $j = 1, \ldots, N$,
          $\tilde{x}_t^j = x_t^{I(j)},$
          $\tilde{x}_{t:T}^j = \{\tilde{x}_t^j, \tilde{x}_{t+1:T}^j\}.$
18: end for
```

the definition of the smoothing weights from (5.11), we get,

$$\frac{\tilde{w}_{t|T}^{I(j),j}}{w_t^{I(j)}} = \frac{1}{\sum_{k=1}^N w_t^k p(\tilde{x}_{t+1}^j \mid x_t^k)} p(\tilde{x}_{t+1}^j \mid x_t^{I(j)}), \tag{5.16}$$

which implies that the sample should be accepted with probability $p(\tilde{x}_{t+1}^j \mid x_t^{I(j)})/\rho$ (cf. Section 2.4.3). The fast FFBSi is given in Algorithm 5.4. We also provide MATLAB code in Listing 5.1.

Douc et al. [2010] have shown that, for $M = N$ the complexity of Algorithm 5.4 converges in probability to $O(NT)$ as N tends to infinity (we refer the reader to [Douc et al., 2010] for the details). Informally, we can say that, for a large number of particles N, the fast FFBSi reaches linear complexity (in the number of particles). However, it is worth to note that there is no upper bound on the number of times that the while-loop at Row 4 may

```
 1 % INPUT:
 2 %   x_pf                 - T*1 cell array, each element is an nx*N matrix
 3 %                          with forward filter particles at time t.
 4 %   w_pf                 - T*1 cell array, each element is an 1*N matrix
 5 %                          with forward filter particle weights.
 6 %   T, M, N, nx,         - Constants, see text for explanation.
 7 %   R_max, rho
 8 % OUTPUT:
 9 %   x_ffbsi              - T*1 cell array, each element is an nx*N matrix
10 %                          with backward trajectories at time t.
11
12 [~, I] = histc(rand(M,1), [0 cumsum(w_pf{T})]); % Sample I ~ Cat(w_pf{T})
13 x_ffbsi{T} = x_pf{t}(:,I);
14 for(t = (T) : (-1) : 1)
15     x_ffbsi{t} = zeros(nx, M);
16     counter = 0;
17     bins = [0 cumsum(w_pf{t})];              % Precomputation
18     L = 1:M;
19     while(~isempty(L) && (counter < R_max))
20         n = length(L);
21         [~, C] = histc(rand(n,1), bins);     % Sample C ~ Cat(w_pf{t})
22         U = rand(1, n);                      % Sample U ~ U([0,1])
23
24         x_t1 = x_ffbsi{t+1}(:,L);            % x_{t+1}^k for k in L
25         x_t =  x_pf{t}(:, C);                % x_t^i     for i in C
26
27         p = transition_density_function(x_t1, x_t);
28         accept = (U <= p/rho);               % Accepted draws
29
30         % L is the index list of samples at time t that still need
31         % assignment ("smoothing particles"). C is the index list of
32         % candidate samples at time t ("filter particles"). That is, the
33         % forward filter particle with (random) index C(k) is either
34         % accepted as the smoothing particle with index L(k), or not.
35
36         x_ffbsi{t}(:,L(accept)) = x_t(:,C(accept));
37         L = L(~accept);                      % Remove accepted indices
38         counter = counter+1;
39     end
40     if(~isempty(L)) % Timeout!
41         for(j = L)
42             xj_t1 = x_ffbsi{t+1}(:,j);       % j:th backward trajectory
43             p = transition_density_function(repmat(xj_t1, 1, N), x_pf{t});
44             w_ji = w_pf{t}.*p;               % Compute smoothing weights
45             w_ji = w_ji/sum(w_ji);           % Normalise
46             I = find(rand(1) < cumsum(w_ji),1,'first');
47             x_ffbsi{t}(:,j) = x_pf{t}(:,I);
48         end
49     end
50 end
```

Listing 5.1: *MATLAB code for fast FFBSi. We have assumed that a function* `transition_density_function(x_t1, x_t)` *is available, where* `x_t1` *and* `x_t` *are $n_x \times N$ matrices (n_x being the state dimension). The function computes the transition density function value $p(x_{t+1} \mid x_t)$ for each pair of columns in the two matrices, and returns the result as a $1 \times N$ row vector. Also, the "timeout check", as described at the end of Section 5.2.2, is included on lines 40–49.*

be executed. Empirical studies indicate that most of the time required by Algorithm 5.4, is spent on just a few particles. In other words, the cardinality of L decreases fast in the beginning (we get a lot of accepted samples), but can linger for a long time close to zero. This can for instance occur when just a single backward trajectory remains, for which all forward filter particles gets low acceptance probabilities. To circumvent this, a "timeout check" can be added to Algorithm 5.4. Hence, let R_{max} be the maximum allowed number of executions of the while-loop at Row 4. If L is not empty after R_{max} iterations, we make an exhaustive evaluation of the smoothing weights for the remaining elements in L, i.e. as in Algorithm 5.3 but with j ranging only over the remaining indices in L. By empirical studies, such a timeout check can drastically reduce the execution time of Algorithm 5.4, and seems to be crucial for its applicability for certain problems. A sensible value for R_{max} seems to be in the range $M/3$ to $M/2$. Such a "timeout check" is included in the MATLAB code presented in Listing 5.1.

Finally, for Algorithm 5.4 to reach linear complexity, we note that the sampling at Row 7 must be conducted prior to the for-loop at Row 9. That is, when proposing indices $\{C(k)\}_{k=1}^{n}$ from the categorical distribution with probabilities $\{w_t^i\}_{i=1}^{N}$, we draw the samples "all at once". The reason is that drawing N i.i.d. samples from a categorical distribution with support at N points can be done in $O(N)$. However, if we instead draw a single sample from the same distribution, this costs $O(\log N)$, and since we then need to repeat this N times the total complexity will be $O(N \log N)$ [Douc et al., 2010].

5.3 Rao-Blackwellised FFBSi

We will now return to the factorised models considered in Section 3.3 and derive an analogue of the Rao-Blackwellised particle filter (RBPF), but with the smoothing problem in mind. Hence, we seek a Rao-Blackwellised particle smoother (RBPS). The RBPS which we shall consider here uses an FFBSi for sampling backward trajectories for the nonlinear state. Consequently, this specific RBPS will be denoted RB-FFBSi. Furthermore, we will focus the derivation on CLGSS models. In particular, we will study mixed linear/nonlinear Gaussian state-space models, as in Section 3.3. An RB-FFBSi designed for hierarchical CLGSS models (see Example 2.2) has previously been proposed by Fong et al. [2002]. However, this is only applicable to the model (2.9) on page 16, in the special case $A^\xi \equiv Q^\xi \equiv 0$, i.e. when the nonlinear state dynamics are independent of the linear states. We will in the present section present a novel RB-FFBSi, capable of handling mixed linear/nonlinear Gaussian state-space models with full interconnection between the state variables. This smoother is a reformulation of some of the material presented in [Lindsten and Schön, 2010] and [Lindsten and Schön, 2011]. For a thorough discussion on how the proposed RB-FFBSi is related to the one derived by Fong et al. [2002], we refer to [Lindsten and Schön, 2011]. It should also be mentioned that Briers et al. [2010] have proposed an RBPS, based on the two-filter algorithm. However, this algorithm is also restricted to hierarchical CLGSS models (i.e. $A^\xi \equiv Q^\xi \equiv 0$).

Assume that the RBPF given in Algorithm 3.3 has been applied to a fix sequence of measurements $y_{1:T}$. Hence, we have generated a sequence of weighted particle systems $\{\xi_{1:t}^i, \omega_t^i\}_{i=1}^{N}$, targeting the state-marginal smoothing densities $p(\xi_{1:t} \mid y_{1:t})$, for

$t = 1, \ldots, T$. For each of these particle trajectories, we have also evaluated the sufficient statistics $\{\bar{z}_{t|t}(\xi^i_{1:t}), P_{t|t}(\xi^i_{1:t})\}_{i=1}^N$ for the conditional filtering distributions of the linear states, with Gaussian densities,

$$p(z_t \mid \xi^i_{1:t}, y_{1:t}) = \mathcal{N}\left(z_t; \bar{z}_{t|t}(\xi^i_{1:t}), P_{t|t}(\xi^i_{1:t})\right), \qquad i = 1, \ldots, N. \tag{5.17}$$

As indicated by (5.17), the mean and covariance of these Gaussians are functions of the nonlinear state trajectory. This implies that if we take a different path, when traversing through the nonlinear partition of the state-space, this will influence our belief about the linear states. However, in the backward simulation, we will in general sample backward trajectories which differ from the forward trajectories. Hence, we can not allow ourselves to condition on the entire forward nonlinear state trajectory. Put in another way, and as we shall see in what follows, the derivation of the RB-FFBSi will require the conditional filtering densities $p(z_t \mid \xi^i_t, y_{1:t})$.

By marginalisation, the RBPF does indeed provide an approximation of the filtering distribution,

$$\begin{aligned}\Phi_{t|t}(dx_t) &= \int \Phi^c_{t|t}(dz_t \mid \xi_{1:t}) \Phi^m_{1:t|t}(d\xi_{1:t}) \\ &\approx \sum_{i=1}^N w^i_t \mathcal{N}\left(dz_t; \bar{z}_{t|t}(\xi^i_{1:t}), P_{t|t}(\xi^i_{1:t})\right) \delta_{\xi^i_t}(d\xi_t),\end{aligned} \tag{5.18}$$

which suggests the following approximation.

Approximation 5.1. *Let ξ^i_t belong to the set of* RBPF *particles at time t. Then, the conditional of the filtering density is approximately Gaussian, according to*

$$p(z_t \mid \xi^i_t, y_{1:t}) \approx \widehat{p}(z_t \mid \xi^i_t, y_{1:t}) \triangleq \mathcal{N}(z_t; \bar{z}_{t|t}(\xi^i_{1:t}), P_{t|t}(\xi^i_{1:t})). \tag{5.19}$$

Note that both (5.18) and (5.19) are approximations, as opposed to (5.17) which is exact.

Remark 5.5. The above approximation corresponds to what was denoted the "ancestral dependence approximation" in the presentation of the RBMPF in Section 3.4. As argued there, this approximation is only good when the system under study is mixing sufficiently fast. In other words, the approximation is only good when the RBPF performs well, which is natural since it arises by marginalisation of the RBPF. We could alternatively employ some other approximation, instead of Approximation 5.1, e.g. based on mixing as suggested in Section 3.4.2. This would be preferable for slowly mixing systems, but comes with the increased computational cost as discussed in Section 3.4.3. We will not consider this alternative further in this section. See also the discussion in Section 5.3.3.

During the derivation of the RB-FFBSi, we will consider the 2-step fixed-interval smoothing problem. The reason is, that this is exactly the problem that we will encounter in the RBPS based identification method of Section 6.3. The derivation could straightforwardly be extended to joint smoothing, and the reason why this is not done is simply for notational convenience. We thus seek to compute expectations of the form,

$$
\begin{aligned}
&\mathrm{E}\left[\varphi(\Xi_{t:t+1}, Z_{t:t+1}) \mid Y_{1:T} = y_{1:T}\right] \\
&\quad = \iint \varphi(\xi_{t:t+1}, z_{t:t+1}) p(\xi_{t:t+1}, z_{t:t+1} \mid y_{1:T})\, d\xi_{t:t+1} dz_{t:t+1} \\
&\quad = \iint \varphi(\xi_{t:t+1}, z_{t:t+1}) p(z_{t:t+1} \mid \xi_{t:T}, y_{1:T}) p(\xi_{t:T} \mid y_{1:T})\, d\xi_{t:T} dz_{t:t+1}, \qquad (5.20)
\end{aligned}
$$

for some test function φ. Clearly, we will also, at the same time, address the marginal smoothing problem. It can be instructive to also consider the approximation of the marginal smoothing densities explicitly (in addition to the 2-step fixed-interval smoothing densities). Hence, the task at hand can be formulated as,

1. Target $p(\xi_{t:T} \mid y_{1:T})$ with a backward simulator, generating a collection of backward trajectories $\{\tilde{\xi}^j_{t:T}\}_{j=1}^M$ for $t = T : -1 : 1$[2].

2. Sequentially (backward in time) evaluate the sufficient statistics for the densities,

   i) $p(z_t \mid \tilde{\xi}^j_{t:T}, y_{1:T})$ for $t = T : -1 : 1$,

   ii) $p(z_{t:t+1} \mid \tilde{\xi}^j_{t:T}, y_{1:T})$ for $t = (T-1) : -1 : 1$,

   for $j = 1, \ldots, M$.

At time $t = T$, this can be done by simply resampling the forward RBPF particles and appeal to (5.19),

$$
\begin{aligned}
\{I(j)\}_{j=1}^M &\sim \mathrm{Cat}\left(\{w_T^i\}_{i=1}^N\right), & & (5.21a)\\
\tilde{\xi}^j_T &:= \xi_T^{I(j)}, & j = 1, \ldots, M, & \quad (5.21b)\\
\tilde{z}^j_{T|T} &:= \bar{z}_{T|T}\left(\xi_{1:T}^{I(j)}\right), & j = 1, \ldots, M, & \quad (5.21c)\\
\widetilde{P}^j_{T|T} &:= P_{T|T}\left(\xi_{1:T}^{I(j)}\right), & j = 1, \ldots, M. & \quad (5.21d)
\end{aligned}
$$

Hence, assume that we have sampled backward trajectories $\{\tilde{\xi}^j_{t+1:T}\}_{j=1}^N$ and that the conditional filtering densities are approximately given by,

$$
\widehat{p}(z_{t+1} \mid \tilde{\xi}^j_{t+1:T}, y_{1:T}) = \mathcal{N}\left(z_{t+1}; \tilde{z}^j_{t+1|T}, \widetilde{P}^j_{t+1|T}\right), \qquad (5.22)
$$

for $j = 1, \ldots, M$. We will now show how to complete the recursions at time t.

5.3.1 Backward simulation

As opposed to the (full) state process $\{X_t\}_{t\geq 1}$, the nonlinear process $\{\Xi_t\}_{t\geq 1}$ in a mixed linear/nonlinear Gaussian state-space model is non-Markov. The same applies to the time-reversed process. Thus, there exist no Markovian backward kernel for the time-reversed, nonlinear process. However, it is still possible to perform backward simulation. We note

[2] Here, we have adopted the MATLAB like syntax for sequences $\{b : -1 : a\} \triangleq \{b,\, b-1,\, \ldots,\, a+1,\, a\}$.

that the target density can be factorised as,

$$p(\xi_{t:T} \mid y_{1:T}) = p(\xi_t \mid \xi_{t+1:T}, y_{1:T}) \underbrace{p(\xi_{t+1:T} \mid y_{1:T})}_{\text{previous target}}. \tag{5.23}$$

Hence, nonlinear backward trajectories can be sampled according to,

$$\tilde{\xi}_t^j \sim p(\xi_t \mid \tilde{\xi}_{t+1:T}^j, y_{1:T}), \tag{5.24a}$$

$$\tilde{\xi}_{t:T}^j := \{\tilde{\xi}_t^j, \tilde{\xi}_{t+1:T}^j\}. \tag{5.24b}$$

However, it turns out that it is in fact easier to sample from the joint distribution with density (see Appendix 5.A),

$$p(z_{t+1}, \xi_{1:t} \mid \tilde{\xi}_{t+1:T}^j, y_{1:T}) = p(\xi_{1:t} \mid z_{t+1}, \tilde{\xi}_{t+1:T}^j, y_{1:T}) p(z_{t+1} \mid \tilde{\xi}_{t+1:T}^j, y_{1:T}). \tag{5.25}$$

Using (5.22), the second factor in (5.25) is approximately Gaussian, and we can easily sample,

$$\widetilde{Z}_{t+1}^j \sim \mathcal{N}\left(z_{t+1}; \tilde{z}_{t+1|T}^j, \widetilde{P}_{t+1|T}^j\right). \tag{5.26}$$

For the first factor of (5.25), by using the conditional independence properties of the model, we get

$$p(\xi_{1:t} \mid z_{t+1}, \xi_{t+1:T}, y_{1:T}) = p(\xi_{1:t} \mid z_{t+1}, \xi_{t+1}, y_{1:t}). \tag{5.27}$$

This result follows from the fact that, conditioned on the state at time $t+1$, $\{\Xi_{t+1}, Z_{t+1}\}$, there is no extra information available in the states at time $\tau > t+1$ or in the measurements at time $\tau > t$. Furthermore, from Bayes' rule we get,

$$p(\xi_{1:t} \mid z_{t+1}, \xi_{t+1}, y_{1:t}) \propto p(z_{t+1}, \xi_{t+1} \mid \xi_{1:t}, y_{1:t}) p(\xi_{1:t} \mid y_{1:t}). \tag{5.28}$$

Hence, we arrive at a distribution that can be readily approximated by the forward filter. The RBPF provides an approximation of the state-marginal smoothing distribution,

$$p(\xi_{1:t} \mid y_{1:t})\, d\xi_{1:t} \approx \widehat{\Phi}_{1:t|t}^{m,N}(d\xi_{1:t}) = \sum_{i=1}^{N} \omega_t^i \delta_{\xi_{1:t}^i}(d\xi_{1:t}), \tag{5.29}$$

which together with (5.27) and (5.28) yields,

$$p(\xi_{1:t} \mid \widetilde{Z}_{t+1}^j, \tilde{\xi}_{t+1:T}^j, y_{1:T})\, d\xi_{1:t} \approx \sum_{i=1}^{N} \tilde{\omega}_{t|T}^{i,j} \delta_{\xi_{1:t}^i}(d\xi_{1:t}), \tag{5.30a}$$

with

$$\tilde{\omega}_{t|T}^{i,j} \triangleq \frac{\omega_t^i p(\widetilde{Z}_{t+1}^j, \tilde{\xi}_{t+1}^j \mid \xi_{1:t}^i, y_{1:t})}{\sum_k \omega_t^k p(\widetilde{Z}_{t+1}^j, \tilde{\xi}_{t+1}^j \mid \xi_{1:t}^k, y_{1:t})}. \tag{5.30b}$$

The density involved in the above weight expression is available in the RBPF and is, according to (3.37) on page 46, given by

$$p(z_{t+1}, \xi_{t+1} \mid \xi_{1:t}^i, y_{1:t}) = \mathcal{N}\left(x_{t+1}; f_t^i + A_t^i \bar{z}_{t|t}^i, Q_t^i + A_t^i P_{t|t}^i (A_t^i)^\mathsf{T}\right). \tag{5.31}$$

Algorithm 5.5 RB-FFBSi: Backward simulation

Input: • RBPF data: An augmented weighted particle system $\{\xi^i_{1:t}, \omega^i_t, \bar{z}^i_{t|t}, P^i_{t|t}\}_{i=1}^N$ targeting $p(z_t \mid \xi_{1:t}, y_{1:t})p(\xi_{1:t} \mid y_{1:t})$.
• RB-FFBSi data: Backward trajectories, augmented with the first and second moments of the linear state, $\{\tilde{\xi}^j_{t+1:T}, \tilde{z}^j_{t+1|T}, \widetilde{P}^j_{t+1|T}\}_{j=1}^M$, (approximately) targeting $p(z_{t+1} \mid \xi_{t+1:T}, y_{1:T})p(\xi_{t+1:T} \mid y_{1:T})$.

Output: Indices $\{I(j)\}_{j=1}^M$.

```
for $j = 1$ to $N$ do
    Sample $\widetilde{Z}^j_{t+1} \sim \mathcal{N}\left(z_{t+1}; \tilde{z}^j_{t+1|T}, \widetilde{P}^j_{t+1|T}\right)$.
    Using (5.31), set $\nu^j_t = \sum_{k=1}^N \omega^i_t p(\widetilde{Z}^j_{t+1}, \tilde{\xi}^j_{t+1} \mid \xi^i_{1:t}, y_{1:t})$.
    Using (5.31), compute the smoothing weights. For $i = 1, \dots, N$,
        $$\tilde{\omega}^{i,j}_{t|T} = \omega^i_t \frac{p(\widetilde{Z}^j_{t+1}, \tilde{\xi}^j_{t+1} \mid \xi^i_{1:t}, y_{1:t})}{\nu^j_t}.$$
    Sample an index from the categorical distribution defined by the smoothing weights,
        $$I(j) \sim \mathrm{Cat}\left(\{\tilde{\omega}^{i,j}_{t|T}\}_{i=1}^N\right).$$
end for
```

Here we have employed the shorthand notation, $f^i_t = f(\xi^i_t)$ etc. For each backward trajectory, i.e. for $j = 1, \dots, M$, we can now sample an index

$$I(j) \sim \mathrm{Cat}\left(\{\tilde{\omega}^{i,j}_{t|T}\}_{i=1}^N\right), \tag{5.32}$$

corresponding to the forward filter particle that is to be appended to the j:th backward trajectory. Since (5.30) defines a distribution over the RBPF particle *trajectories*, we will in fact obtain a sample $\xi^{I(j)}_{1:t}$ from the space of trajectories X^t_ξ. However, by simply discarding $\xi^{I(j)}_{1:t-1}$ and also the auxiliary variable $\widetilde{Z}^j_{t+1}$, we end up with a sample $\tilde{\xi}^j_t := \xi^{I(j)}_t$, approximately distributed according to (5.24a), which can then be appended to the nonlinear backward trajectory as in (5.24b).

We summarise the sampling procedure described above in Algorithm 5.5. Just as the original FFBSi formulation presented in Section 5.2, the backward simulator given by Algorithm 5.5 has $O(NMT)$ complexity. However, we can straightforwardly adapt the fast backward simulation technique by Douc et al. [2010], presented in Section 5.2.2, to the RB-FFBSi. Here, the target distribution is categorical with probabilities $\{\tilde{\omega}^{i,j}_{t|T}\}_{i=1}^N$. Again, we use the forward filter weights $\{\omega^i_t\}_{i=1}^N$ to define a proposal distribution. The target and proposal weights are related according to (5.30b). Similarly to (5.15) we bound the quotient between the two. However, since we are now dealing with a Gaussian density according to (5.31), we can give explicit expressions for the bounds, according to,

$$\rho_t \triangleq (2\pi)^{-\frac{n_x}{2}} \max_{i=1,\dots,N} \left[\det\left(Q^i + A^i P^i_{t|t} (A^i)^{\mathsf{T}}\right)^{-\frac{1}{2}}\right]. \tag{5.33}$$

Algorithm 5.6 RB-FFBSi: Fast backward simulation

Input: • RBPF data: An augmented weighted particle system $\{\xi^i_{1:t}, \omega^i_t, \bar{z}^i_{t|t}, P^i_{t|t}\}_{i=1}^N$ targeting $p(z_t \mid \xi_{1:t}, y_{1:t})p(\xi_{1:t} \mid y_{1:t})$.
• RB-FFBSi data: Backward trajectories, augmented with the first and second moments of the linear state, $\{\tilde{\xi}^j_{t+1:T}, \tilde{z}^j_{t+1|T}, \tilde{P}^j_{t+1|T}\}_{j=1}^M$, (approximately) targeting $p(z_{t+1} \mid \xi_{t+1:T}, y_{1:T})p(\xi_{t+1:T} \mid y_{1:T})$.

Output: Indices $\{I(j)\}_{j=1}^M$.

```
for $j = 1$ to $N$ do
    Sample $\tilde{Z}^j_{t+1} \sim \mathcal{N}\left(z_{t+1}; \tilde{z}^j_{t+1|T}, \tilde{P}^j_{t+1|T}\right)$.
end for
$L \leftarrow \{1, \dots, M\}$
while $L$ is not empty do
    $n \leftarrow \mathrm{card}(L)$.
    $\delta \leftarrow \emptyset$.
    Sample independently $\{C(k)\}_{k=1}^n \sim \mathrm{Cat}(\{w^i_t\}_{i=1}^N)$.
    Sample independently $\{U(k)\}_{k=1}^n \sim \mathcal{U}([0, 1])$.
    for $k = 1$ to $n$ do
        if $U(k) \le p\left(\tilde{Z}^{L(k)}_{t+1}, \tilde{\xi}^{L(k)}_{t+1} \mid \xi^{C(k)}_{1:t}, y_{1:t}\right) / \rho_t$ then
            $I(L(k)) \leftarrow C(k)$.
            $\delta \leftarrow \delta \cup \{L(k)\}$.
        end if
    end for
    $L \leftarrow L \setminus \delta$.
end while
```

A fast backward simulator, adapted to the RB-FFBSi, is given in Algorithm 5.6. We emphasise that, in terms of input and output, this algorithm is equivalent to Algorithm 5.5. Also, note that the discussion in Section 5.2.2 applies also the fast backward simulator given here. For instance, the proposed "timeout check" which (on heuristic grounds) was suggested to obtain further speedup in practice, can be used also in Algorithm 5.6.

5.3.2 Smoothing the linear states

When traversing backward in time, we also need to update the sufficient statistics for the linear states. Since, according to (5.26), the smoothing distribution for the linear states is required during the backward simulation, we need to update it sequentially (backward in time). In accordance with (5.22), we will approximate it with a Gaussian density,

$$p(z_t \mid \tilde{\xi}^j_{t:T}, y_{1:T}) \approx \hat{p}(z_t \mid \tilde{\xi}^j_{t:T}, y_{1:T}) = \mathcal{N}\left(z_t; \tilde{z}^j_{t|T}, \tilde{P}^j_{t|T}\right); \tag{5.34}$$

at time $t = T$, the approximation is given by (5.19). The mean and covariance of this Gaussian density will be determined by fusing the information available in the forward RBPF at time t, with the (existing) marginal smoothing distribution for the linear states at time $t+1$. We start by noting that, by the conditional independence properties of an SSM,

$$p(z_t \mid z_{t+1}, \xi_{t:T}, y_{1:T}) = p(z_t \mid z_{t+1}, \xi_t, \xi_{t+1}, y_{1:t}). \tag{5.35}$$

Furthermore, from Bayes' rule we have,

$$p(z_t \mid z_{t+1}, \xi_t, \xi_{t+1}, y_{1:t}) \propto p(z_{t+1}, \xi_{t+1} \mid z_t, \xi_t, y_{1:t}) p(z_t \mid \xi_t, y_{1:t}). \tag{5.36}$$

We recognise the first factor of (5.36) as the transition density, which for a mixed linear/nonlinear Gaussian state-space model is Gaussian and affine in z_t (see (2.9) on page 16),

$$\begin{aligned} p(z_{t+1}, \xi_{t+1} \mid z_t, \xi_t, y_{1:t}) &= p(z_{t+1}, \xi_{t+1} \mid z_t, \xi_t) \\ &= \mathcal{N}\left(\underbrace{\begin{bmatrix} \xi_{t+1} \\ z_{t+1} \end{bmatrix}}_{=x_{t+1}}; \underbrace{\begin{bmatrix} f^\xi(\xi_t) \\ f^z(\xi_t) \end{bmatrix}}_{=f(\xi_t)} + \underbrace{\begin{bmatrix} A^\xi(\xi_t) \\ A^z(\xi_t) \end{bmatrix}}_{=A(\xi_t)} z_t, \underbrace{\begin{bmatrix} Q^\xi(\xi_t) & Q^{\xi z}(\xi_t) \\ (Q^{\xi z}(\xi_t))^\mathsf{T} & Q^z(\xi_t) \end{bmatrix}}_{=Q(\xi_t)} \right). \end{aligned} \tag{5.37}$$

The second factor of (5.36) is the conditional of the filtering density. By replacing this by its Gaussian approximation (5.19), we arrive at an affine transformation of a Gaussian variable, which itself is Gaussian. Hence, let ξ_t^i belong to the set of RBPF particles at time t. Then, by (5.19), (5.36), (5.37) and Corollary B.1 in Appendix B we get,

$$p(z_t \mid z_{t+1}, \xi_t^i, \xi_{t+1}, y_{1:t}) \approx \widehat{p}(z_t \mid z_{t+1}, \xi_t^i, \xi_{t+1}, y_{1:t}) \triangleq \mathcal{N}\left(z_t; \mu_{t|t}^i(x_{t+1}), \Pi_{t|t}^i\right), \tag{5.38}$$

where we have introduced,

$$\mu_{t|t}^i(x_{t+1}) \triangleq \Pi_{t|t}^i \left((A_t^i)^\mathsf{T} (Q_t^i)^{-1} \left(\begin{bmatrix} \xi_{t+1}^\mathsf{T} & z_{t+1}^\mathsf{T} \end{bmatrix}^\mathsf{T} - f_t^i \right) + (P_{t|t}^i)^{-1} \bar{z}_{t|t}^i \right), \tag{5.39a}$$

$$\begin{aligned} \Pi_{t|t}^i &\triangleq \left((P_{t|t}^i)^{-1} + (A_t^i)^\mathsf{T} (Q_t^i)^{-1} A_t^i \right)^{-1} \\ &= P_{t|t}^i - P_{t|t}^i (A_t^i)^\mathsf{T} \left(Q_t^i + A_t^i P_{t|t}^i (A_t^i)^\mathsf{T} \right)^{-1} A_t^i P_{t|t}^i. \end{aligned} \tag{5.39b}$$

To expand the expression (5.39a) we introduce,

$$\begin{bmatrix} W^\xi(\xi_t) & W^z(\xi_t) \end{bmatrix} \triangleq A(\xi_t)^\mathsf{T} Q(\xi_t)^{-1}. \tag{5.40}$$

Explicit expressions for the functions W^ξ and W^z can be given in terms of the process noise precision,

$$W^\xi(\xi_t) = A^\xi(\xi_t)^\mathsf{T} \Lambda^\xi(\xi_t) + A^z(\xi_t)^\mathsf{T} \Lambda^{\xi z}(\xi_t)^\mathsf{T}, \tag{5.41a}$$

$$W^z(\xi_t) = A^\xi(\xi_t)^\mathsf{T} \Lambda^{\xi z}(\xi_t) + A^z(\xi_t)^\mathsf{T} \Lambda^z(\xi_t), \tag{5.41b}$$

where

$$\begin{bmatrix} \Lambda^\xi(\xi_t) & \Lambda^{\xi z}(\xi_t) \\ (\Lambda^{\xi z}(\xi_t))^\mathsf{T} & \Lambda^z(\xi_t) \end{bmatrix} = Q(\xi_t)^{-1}. \tag{5.41c}$$

By plugging (5.40) into (5.39a) we get,

$$\begin{aligned} \mu_{t|t}^i(x_{t+1}) &= \Pi_{t|t}^i \left(W_t^{\xi,i}(\xi_{t+1} - f_t^{\xi,i}) + W_t^{z,i} z_{t+1} - W_t^{z,i} f_t^{z,i} + (P_{t|t}^i)^{-1} \bar{z}_{t|t}^i \right) \\ &= \Pi_{t|t}^i W_t^{z,i} z_{t+1} + c_{t|t}^i(\xi_{t+1}), \end{aligned} \tag{5.42a}$$

where we have defined

$$c^i_{t|t}(\xi_{t+1}) \triangleq \Pi^i_{t|t}\left(W^{\xi,i}_t(\xi_{t+1} - f^{\xi,i}_t) - W^{z,i}_t f^{z,i}_t + (P^i_{t|t})^{-1}\bar{z}^i_{t|t}\right). \tag{5.42b}$$

We seek to combine (5.38) with the smoothing distribution for the linear states at time $t+1$ given by (5.22). To enable this, we need the following Gaussian approximation.

Approximation 5.2. *Let $\tilde{\xi}^j_{t:T}$ be any particle trajectory, sampled during the backward simulation as described in Section 5.3.1. Let $I(j)$ be the index of the FF particle that was appended at time t, i.e. $\tilde{\xi}^j_{t:T} = \{\xi^{I(j)}_t, \tilde{\xi}^j_{t:T}\}$. Then,*

$$p(z_{t+1} \mid \xi^{I(j)}_t, \tilde{\xi}^j_{t+1:T}, y_{1:T}) \approx \widehat{p}(z_{t+1} \mid \tilde{\xi}^j_{t+1:T}, y_{1:T}) = \mathcal{N}\left(z_{t+1}; \tilde{z}^j_{t+1|T}, \widetilde{P}^j_{t+1|T}\right). \tag{5.43}$$

The implication of this approximation is that we assume that the smoothing estimate for z_{t+1} is independent of which forward filter particle $\xi^{I(j)}_t$ that is appended to the backward trajectory. This approximation can be motivated by the fact that a particle $\xi^{I(j)}_t$ is more likely to be drawn if it has a good fit to the current smoothing trajectory. Hence, it should not affect the smoothing estimate at time $t+1$ to any significant extent. See also the discussion in Section 5.3.3.

Now, from (5.35) and (5.38), we have that the density $p(z_t \mid z_{t+1}, \xi^{I(j)}_t, \tilde{\xi}^j_{t+1:T}, y_{1:T})$ is approximately Gaussian. Furthermore, from (5.42a) we see that it has an affine dependence on z_{t+1}. By Theorem B.3 this yields, together with (5.43), a Gaussian approximation of $p(z_{t:t+1} \mid \xi^{I(j)}_t, \tilde{\xi}^j_{t+1:T}, y_{1:T})$ as,

$$\widehat{p}(z_{t:t+1} \mid \xi^{I(j)}_t, \tilde{\xi}^j_{t+1:T}, y_{1:T}) = \mathcal{N}\left(\begin{bmatrix} z_t \\ z_{t+1} \end{bmatrix}; \begin{bmatrix} \tilde{z}^j_{t|T} \\ \tilde{z}^j_{t+1|T} \end{bmatrix}, \begin{bmatrix} \widetilde{P}^j_{t|T} & M^j_{t|T} \\ (M^j_{t|T})^{\mathsf{T}} & \widetilde{P}_{t+1|T} \end{bmatrix}\right), \tag{5.44}$$

with

$$\tilde{z}^j_{t|T} = \Pi^{I(j)}_{t|t} W^{z,I(j)}_t \tilde{z}^j_{t+1|T} + c^{I(j)}_{t|t}(\tilde{\xi}^j_{t+1}), \tag{5.45a}$$

$$\widetilde{P}^j_{t|T} = \Pi^{I(j)}_{t|t} + M^j_{t|T}(W^{z,I(j)}_t)^{\mathsf{T}}\Pi^{I(j)}_{t|t}, \tag{5.45b}$$

$$M^j_{t|T} = \Pi^{I(j)}_{t|t} W^{z,I(j)}_t \widetilde{P}^j_{t+1|T}. \tag{5.45c}$$

Finally, by marginalisation of the above density (Theorem B.1) we get an approximation of the marginal, conditional smoothing density,

$$\widehat{p}(z_t \mid \xi^{I(j)}_t, \tilde{\xi}^j_{t+1:T}, y_{1:T}) = \mathcal{N}\left(z_t; \tilde{z}^j_{t|T}, \widetilde{P}^j_{t|T}\right). \tag{5.46}$$

The resulting RB-FFBSi is summarised in Algorithm 5.7. In conclusion, we note that the algorithm results in an approximation of the 2-step, fixed-interval smoothing distribution,

$$\begin{aligned} \Phi_{t:t+1|T}(dx_{t:t+1}) &\approx \widetilde{\Phi}^{\text{RB},M}_{t:t+1|T}(dx_{t:t+1}) \\ &\triangleq \sum_{j=1}^{M} \mathcal{N}\left(d\begin{bmatrix} z_t \\ z_{t+1} \end{bmatrix}; \begin{bmatrix} \tilde{z}^j_{t|T} \\ \tilde{z}^j_{t+1|T} \end{bmatrix}, \begin{bmatrix} \widetilde{P}^j_{t|T} & M^j_{t|T} \\ (M^j_{t|T})^{\mathsf{T}} & \widetilde{P}_{t+1|T} \end{bmatrix}\right) \delta_{\tilde{\xi}^j_{t:t+1}}(d\xi_{t:t+1}). \end{aligned} \tag{5.47}$$

Algorithm 5.7 RB-FFBSi for mixed linear/nonlinear Gaussian state-space models

Input: A sequence of augmented weighted particle systems $\{\xi^i_{1:t}, \omega^i_t, \bar{z}^i_{t|t}, P^i_{t|t}\}_{i=1}^N$ targeting $p(z_t \mid \xi_{1:t}, y_{1:t})p(\xi_{1:t} \mid y_{1:t})$, for $t = 1, \ldots, T$.

Output: Smoothed nonlinear state trajectories $\{\tilde{\xi}^j_{1:T}\}_{j=1}^M$, with the corresponding sufficient statistics for the linear states, $\{\tilde{z}^j_{t|T}, \widetilde{P}^j_{t|T}, M^j_{t|T}\}_{j=1}^M$ for $t = 1, \ldots, T$ ($M^j_{t|T}$ only for $t < T$).

```
Initialise the smoother by resampling the RBPF at time $T$, i.e. generate $\{\tilde{\xi}^j_T, \tilde{z}^j_{T|T}, \widetilde{P}^j_{T|T}\}_{j=1}^M$ according to (5.21).
for $t = T - 1$ to $1$ do
    Sample indices $\{I(j)\}_{j=1}^M$ according to Algorithm 5.6 (or Algorithm 5.5).
    Augment the backward trajectories. For $j = 1, \ldots, M$,
        $\tilde{\xi}^j_{t:T} = \{\tilde{\xi}^j_t, \tilde{\xi}^j_{t+1:T}\}$.
    Update the sufficient statistics for the linear states. For $j = 1, \ldots, M$,
        $\tilde{z}^j_{t|T} = \Pi^{I(j)}_{t|t} W^{z,I(j)}_t \tilde{z}^j_{t+1|T} + c^{I(j),j}_{t|t}$,
        $\widetilde{P}^j_{t|T} = \Pi^{I(j)}_{t|t} + M^j_{t|T}(W^{z,I(j)}_t)^{\mathsf{T}} \Pi^{I(j)}_{t|t}$,
        $M^j_{t|T} = \Pi^{I(j)}_{t|t} W^{z,I(j)}_t \widetilde{P}^j_{t+1|T}$,
    where
        $c^{I(j),j}_{t|t} = \Pi^{I(j)}_{t|t} \left( W^{\xi,I(j)}_t (\tilde{\xi}^j_{t+1} - f^{\xi,I(j)}_t) - W^{z,I(j)}_t f^{z,I(j)}_t + (P^{I(j)}_{t|t})^{-1} z^{I(j)}_{t|t} \right)$,
        $\Pi^{I(j)}_{t|t} = P^{I(j)}_{t|t} - P^{I(j)}_{t|t} (A^{I(j)}_t)^{\mathsf{T}} \left( Q^{I(j)}_t + A^{I(j)}_t P^{I(j)}_{t|t} (A^{I(j)}_t)^{\mathsf{T}} \right)^{-1} A^{I(j)}_t P^{I(j)}_{t|t}$,
    and
        $\begin{bmatrix} W^{\xi,I(j)_t} & W^{z,I(j)}_t \end{bmatrix} = (A^{I(j)}_t)^{\mathsf{T}} (Q^{I(j)}_t)^{-1}$.
end for
```

5.3.3 Discussion

During the derivation of the RB-FFBSi in the previous section, we were forced to make several Gaussian approximations. However, these can all be traced back to Approximation 5.1 and Approximation 5.2. To understand why we need to use these approximations, consider the conditional distribution appearing in (5.17). This is the basis for both the RBPF and the RBPS, stating that as long as we traverse along (and condition on) a nonlinear state trajectory $\xi^i_{1:t}$, the conditional distribution is Gaussian. Clearly, the purpose of smoothing through backward simulation is to "update" the trajectories generated by the forward filter (if we do not allow for any change of the trajectories, we will not gain anything from smoothing). When we no longer have fixed nonlinear state trajectories, the Gaussianity implied by (5.17) is lost. To circumvent this, we make use of Approximation 5.1 and Approximation 5.2. Informally, the meaning of both these approximations is that, conditioned on the present, we do not care about the past. Hence, the justification of the involved approximations is closely related to the mixing properties of the system. For slowly mixing (or non-mixing) systems, the proposed smoother should be used with care. However, if the underlying system is mixing sufficiently fast, good performance has been experienced in empirical studies, as we shall see in Section 5.3.5 and also in Section 6.3.2.

5.3.4 A special case

As previously mentioned, the RBPS derived by Fong et al. [2002], and also the one given by Briers et al. [2010], are restricted to hierarchical CLGSS models. Similarly, the RBPS presented above is derived explicitly for mixed linear/nonlinear Gaussian state-space models, defined by (2.9) on page 16. However, for the special case $A^\xi \equiv Q^{\xi z} \equiv 0$, the model (2.9) on page 16 coincides with a hierarchical CLGSS model, as defined by (2.7), with a Gaussian kernel Q^ξ. We will now take a closer look at this special case, and see that we recover a smoothing recursion which shows great similarities with the one derived by Fong et al. [2002].

Since the process noise covariance $Q(\xi_t)$ is now block diagonal we get the precision matrices $\Lambda^\xi(\xi_t) = Q^\xi(\xi_t)^{-1}$, $\Lambda^z(\xi_t) = Q^z(\xi_t)^{-1}$ and $\Lambda^{\xi z}(\xi_t) \equiv 0$. Furthermore, since $A^\xi(\xi_t) \equiv 0$, we get from (5.41),

$$W^\xi(\xi_t) \equiv 0, \tag{5.48a}$$

$$W^z(\xi_t) = A^z(\xi_t)^\mathsf{T} Q^z(\xi_t)^{-1}, \tag{5.48b}$$

which in (5.42b) and (5.39b) gives

$$c^i_{t|t}(\xi_{t+1}) = -\Pi^i_{t|t} W^{z,i}_t f^{z,i}_t + \Pi^i_{t|t}(P^i_{t|t})^{-1} \bar{z}^i_{t|t}, \tag{5.49a}$$

and

$$\Pi^i_{t|t} = P^i_{t|t} - P^i_{t|t}(A^{z,i}_t)^\mathsf{T} \left(Q^{z,i}_t + A^{z,i}_t P^i_{t|t} (A^{z,i}_t)^\mathsf{T} \right)^{-1} A^{z,i}_t P^i_{t|t} = P^i_{t|t} - T^i_t A^{z,i}_t P^i_{t|t}, \tag{5.49b}$$

where we have defined

$$T^i_t \triangleq P^i_{t|t}(A^{z,i}_t)^\mathsf{T} \left(Q^{z,i}_t + A^{z,i}_t P^i_{t|t} (A^{z,i}_t)^\mathsf{T} \right)^{-1} = P^i_{t|t}(A^{z,i}_t)^\mathsf{T}(P^i_{t+1|t})^{-1}. \tag{5.49c}$$

The last equality follows from (3.37) and (3.39c) on page 46.

Now, consider the product,

$$\begin{aligned}
\Pi^i_{t|t} W^{z,i}_t &= P^i_{t|t}(A^{z,i}_t)^\mathsf{T}(Q^{z,i}_t)^{-1} - T^i_t A^{z,i}_t P^i_{t|t}(A^{z,i}_t)^\mathsf{T}(Q^{z,i}_t)^{-1} \\
&= P^i_{t|t}(A^{z,i}_t)^\mathsf{T} \left(I_{n_z \times n_z} - (P^i_{t+1|t})^{-1} A^{z,i}_t P^i_{t|t}(A^{z,i}_t)^\mathsf{T} \right) (Q^{z,i}_t)^{-1} \\
&= P^i_{t|t}(A^{z,i}_t)^\mathsf{T}(P^i_{t+1|t})^{-1} \underbrace{\left(P^i_{t+1|t} - A^{z,i}_t P^i_{t|t}(A^{z,i}_t)^\mathsf{T} \right)}_{=Q^{z,i}_t} (Q^{z,i}_t)^{-1} \\
&= P^i_{t|t}(A^{z,i}_t)^\mathsf{T}(P^i_{t+1|t})^{-1} = T^i_t.
\end{aligned} \tag{5.50}$$

We can now rewrite the updating formulas (5.45) as,

$$\begin{aligned}\tilde{z}^{j}_{t|T} &= T_t^{I(j)}\tilde{z}^{j}_{t+1|T} - T_t^{I(j)} f_t^{z,I(j)} + \Pi_{t|t}^{I(j)}(P_{t|t}^{I(j)})^{-1}\bar{z}_{t|t}^{I(j)} \\ &= \bar{z}_{t|t}^{I(j)} + T_t^{I(j)}\left(\tilde{z}^{j}_{t+1|T} - f_t^{z,I(j)} - A_t^{z,I(j)}\bar{z}_{t|t}^{I(j)}\right) \\ &= \bar{z}_{t|t}^{I(j)} + T_t^{I(j)}\left(\tilde{z}^{j}_{t+1|T} - \bar{z}_{t+1|t}^{I(j)}\right), \end{aligned} \tag{5.51a}$$

where the last equality follows from (3.37) and (3.39b). Furthermore,

$$M^{j}_{t|T} = T_t^{I(j)}\widetilde{P}^{j}_{t+1|T}, \tag{5.51b}$$

and finally

$$\begin{aligned}\widetilde{P}^{j}_{t|T} &= P_{t|t}^{I(j)} - T_t^{I(j)} A_t^{z,I(j)} P_{t|t}^{I(j)} + T_t^{I(j)}\widetilde{P}^{j}_{t+1|T}(T_t^{I(j)})^{\mathsf{T}} \\ &= \Big/ A_t^{z,i} P_{t|t}^{i} = P_{t+1|t}^{i}(T_t^{i})^{\mathsf{T}} \Big/ \\ &= P_{t|t}^{I(j)} - T_t^{I(j)}\left(P_{t+1|t}^{I(j)} - \widetilde{P}^{j}_{t+1|T}\right)(T_t^{I(j)})^{\mathsf{T}}. \end{aligned} \tag{5.51c}$$

The above expressions for $\tilde{z}_{t|T}$ and $\widetilde{P}_{t|T}$ can be recognised as the Rauch-Tung-Striebel (RTS) recursions for the smoothing estimate in LGSS models [Rauch et al., 1965]. That is, for hierarchical CLGSS models, the RB-FFBSi can be seen as an FFBSi working on the nonlinear state process, equipped with an RTS smoother for smoothing of the linear states. This is expected since, for hierarchical CLGSS models, the nonlinear state process evolves independently of the linear state process. We can thus decouple the problem into one nonlinear smoothing problem (addressed with an FFBSi) and one linear smoothing problem (addressed with an RTS smoother). This is not the case for mixed linear/nonlinear Gaussian state-space models. For such models, there is a interconnection between the dynamics of the linear and the nonlinear states. This leads to the more "complicated" updating formulas (5.45), which are not straightforward modifications of the RTS formulas.

The RB-FFBSi for this special case is very similar to the one derived by Fong et al. [2002]. The difference is that Fong et al. [2002] applies backward simulation jointly for the nonlinear, as well as the linear states. That is, the output from their smoother is a collection of smoothing trajectories $\{\tilde{x}^{j}_{1:t}\} = \{\tilde{\xi}^{j}_{1:t}, \tilde{z}^{j}_{1:t}\}$. To perform this joint backward simulation, they apply a one-step RTS smoothing of the linear state, in each step of the backward recursion. On the contrary, the output from the RB-FFBSi proposed here, is a collection of *nonlinear* backward trajectories, together with means and covariances for the linear states, as indicated by Algorithm 5.7. We discuss the differences and similarities between the two approaches further in [Lindsten and Schön, 2011].

5.3.5 Numerical illustrations

In this section we will illustrate the RB-FFBSi given in Algorithm 5.7 in numerical examples. Since the smoother is based on a forward/backward recursion, we will also provide some results for the forward RBPF, given in Algorithm 3.3. A bootstrap RBPF is used for forward filtering; see Definition 3.4.

Two different examples will be presented, first we consider a linear Gaussian system and thereafter a mixed linear/nonlinear system. The purpose of including a linear Gaussian example is to gain confidence in the presented smoother. This is possible since, for this case, there are closed form solutions available for the filtering and smoothing densities via the Kalman filter (KF) and the RTS smoother, respectively.

For both the linear and the mixed linear/nonlinear examples, we can also address the inference problems using standard particle methods. To solve the filtering problem, we then use the bootstrap PF by Gordon et al. [1993]. The smoothing problem is thereafter addressed using the fast FFBSi by Douc et al. [2010], discussed in Section 5.2.2.

As a measure of evaluation, we use the time-averaged root mean squared error (RMSE). For a sequence of estimates $\{\hat{x}_t^k\}_{t=1}^T$ of the estimands $\{x_t^k\}_{t=1}^T$, over $k = 1, \dots, K$ experiments, this is defined as,

$$\text{RMSE} \triangleq \frac{1}{T}\sum_{t=1}^{T}\sqrt{\frac{1}{K}\sum_{k=1}^{K}(\hat{x}_t^k - x_t^k)^2}. \tag{5.52}$$

Example 5.2: RB-FFBSi: 2^nd order LGSS system

We start the evaluation of the RB-FFBSi on a linear, second order system, according to

$$\begin{pmatrix}\Xi_{t+1}\\ Z_{t+1}\end{pmatrix} = \begin{pmatrix}0.8 & 0.1\\ 0 & 1\end{pmatrix}\begin{pmatrix}\Xi_t\\ Z_t\end{pmatrix} + V_t, \qquad V_t \sim \mathcal{N}(0, Q), \tag{5.53a}$$

$$Y_t = \Xi_t + E_t, \qquad E_t \sim \mathcal{N}(0, R), \tag{5.53b}$$

with $Q = 0.01 I_{2\times 2}$ and $R = 0.1$. The initial state of the system is Gaussian according to

$$\begin{pmatrix}\Xi_1\\ Z_1\end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix}0\\ 5\end{pmatrix}, \begin{pmatrix}10^{-6} & 0\\ 0 & 10^{-6}\end{pmatrix}\right). \tag{5.54}$$

In the RBPF and the RB-FFBSi, the first state Ξ_t is treated as if it is nonlinear, whereas the second state Z_t is treated as linear.

The comparison was made by a Monte Carlo study over 100 realisations of data $y_{1:T}$ from the system (5.53), each consisting of $T = 200$ samples (measurements). The three filters, KF, PF and RBPF, and thereafter the three smoothers, RTS, (fast) FFBSi and (fast) RB-FFBSi, were run in parallel. The PF and RBPF both employed $N = 50$ particles and the FFBSi and RB-FFBSi used $M = 50$ backward trajectories.

Table 5.1 and Table 5.2 gives the RMSEs for the three filters and smoothers, respectively.

Table 5.1: *RMSEs for filters*

Filter	ξ_t	z_t
PF	0.16	0.41
RBPF	0.15	0.36
KF	0.15	0.36

Table 5.2: *RMSEs for smoothers*

Smoother	ξ_t	z_t
FFBSi	0.14	0.32
RB-FFBSi	0.13	0.25
RTS	0.12	0.24

The results are as expected. First, smoothing clearly decreases the RMSEs when compared to filtering. Second, Rao-Blackwellisation has the desired effect of decreasing the RMSE when compared to standard particle methods. The RBPF and the RB-FFBSi perform close to the optimal KF and RTS, respectively. When looking at the “linear” state z_t, the PF and the FFBSi result in significantly larger errors in this example.

The key difference between the PF/FFBSi and the RBPF/RB-FFBSi is that in the former, the particles have to cover the distribution in two dimensions. In the RBPF/RB-FFBSi we marginalise out one of the dimensions and thus only need to deal with, what appears as, a one-dimensional problem using particles.

We continue with an example with a mixed linear/nonlinear Gaussian system. Since the system is nonlinear, the optimal filter and smoother are not available. We thus make the comparison only between the PF/FFBSi and the RBPF/RB-FFBSi.

Example 5.3: RB-FFBSi: 4th order mixed linear/nonlinear Gaussian system

Consider the fourth order mixed linear/nonlinear Gaussian system, where three of the states are conditionally linear Gaussian, given by,

$$\Xi_{t+1} = \arctan \Xi_t + \begin{pmatrix}1 & 0 & 0\end{pmatrix} Z_t + V_t^\xi, \tag{5.55a}$$

$$Z_{t+1} = \begin{pmatrix}1 & 0.3 & 0 \\ 0 & 0.92 & -0.3 \\ 0 & 0.3 & 0.92\end{pmatrix} Z_t + V_t^z, \tag{5.55b}$$

$$Y_t = \begin{pmatrix}0.1\Xi_t^2 \operatorname{sign}(\Xi_t) \\ 0\end{pmatrix} + \begin{pmatrix}0 & 0 & 0 \\ 1 & -1 & 1\end{pmatrix} Z_t + E_t, \tag{5.55c}$$

with $V_t = \begin{bmatrix}V_t^\xi & (V_t^z)^\mathsf{T}\end{bmatrix}^\mathsf{T} \sim \mathcal{N}(0, Q)$, $Q = 0.01 I_{4\times 4}$ and $E_t \sim N(0, R)$, $R = 0.1 I_{2\times 2}$. The initial distribution for the system is $X_1 \sim \delta_0(dx_1)$. The Z-system is oscillatory and marginally stable, with poles in 1 and $0.92 \pm 0.3i$. The linear Z-variables are connected to the nonlinear Ξ-system through $Z_{1,t}$.

Again, 100 realisations of data $y_{1:T}$ were generated, each consisting of $T = 200$ samples. Table 5.3 and Table 5.4 present the RMSE values for the PF and RBPF, and for the FFBSi and RB-FFBSi, respectively. The PF and RBPF both employed $N = 50$ particles and the FFBSi and RB-FFBSi used $M = 50$ backward trajectories.

Table 5.3: *RMSEs for filters*

Filter	ξ_t	$z_{1,t}$	$z_{2,t}$	$z_{3,t}$
PF	1.12	0.66	0.28	0.21
RBPF	0.45	0.29	0.21	0.18

Table 5.4: *RMSEs for smoothers*

Smoother	ξ_t	$z_{1,t}$	$z_{2,t}$	$z_{3,t}$
FFBSi	1.08	0.60	0.21	0.20
RB-FFBSi	0.33	0.16	0.11	0.14

The benefits of using Rao-Blackwellisation becomes even more evident in this, more challenging, problem. Since we can marginalise over three out of the four “dimensions”, Rao-Blackwellisation allows us to handle this four-dimensional system using only 50 particles. To see how the results are affected by the number of particles, we run the same experiment again, using $N = M = 200$ particles/backward trajectories. The results are summarised in Table 5.5 and in Table 5.6.

Table 5.5: *RMSEs for filters*

Filter	ξ_t	$z_{1,t}$	$z_{2,t}$	$z_{3,t}$
PF	0.43	0.29	0.21	0.18
RBPF	0.40	0.27	0.20	0.17

Table 5.6: *RMSEs for smoothers*

Smoother	ξ_t	$z_{1,t}$	$z_{2,t}$	$z_{3,t}$
FFBSi	0.32	0.16	0.13	0.15
RB-FFBSi	0.28	0.14	0.12	0.13

When we increase the number of particles, the difference between the PF/FFBSi and the RBPF/RB-FFBSi naturally becomes less pronounced. However, we note that the performance of the RBPF/RB-FFBSi using $N = M = 50$ is similar to that of the PF/FFBSi using $N = M = 200$.

Appendix

5.A Sampling in the RB-FFBSi

The sampling step in the RB-FFBSi at time t appends a new sample $\tilde{\xi}_t^j$ to a backward trajectory $\tilde{\xi}_{t+1:T}^j$. Hence, from (5.24a) we see that we wish to draw samples from the distribution $p(\xi_t \mid \tilde{\xi}_{t+1:T}^j, y_{1:T})$. In this appendix we shall see why it is easier to instead sample from the join distribution with density $p(\xi_{1:t}, z_{t+1} \mid \tilde{\xi}_{t+1:T}^j, y_{1:T})$ and thereafter discard everything but $\tilde{\xi}_t^j$.

First of all we note that the backward simulation makes use of the forward filter particles, i.e. we only sample among the particles generated by the forward filter. This means that our target distribution can be written as a weighted point-mass distribution according to

$$p(\xi_t \mid \tilde{\xi}_{t+1:T}^j, y_{1:T})\, d\xi_t \approx \sum_{i=1}^N \theta_t^{i,j} \delta_{\xi_t^i}(d\xi_t), \tag{5.56}$$

with some, yet unspecified, weights $\theta_t^{i,j}$. Clearly, the tricky part is to compute these weights, once we have them the sampling is trivial.

To see why it is indeed hard to compute the weights, we consider the joint distribution with density $p(\xi_{1:t}, z_{t+1} \mid \tilde{\xi}_{t+1:T}^j, y_{1:T})$. Following the steps in (5.25)–(5.30), this distribution is approximately

$$\begin{aligned} p(\xi_{1:t}, &z_{t+1} \mid \tilde{\xi}_{t+1:T}^j, y_{1:T})\, d\xi_{1:t} dz_{t+1} \\ &\approx p(z_{t+1} \mid \tilde{\xi}_{t+1:T}^j, y_{1:T}) \frac{\sum_{i=1}^N \omega_t^i p(z_{t+1}, \tilde{\xi}_{t+1}^j \mid \xi_{1:t}^i, y_{1:t})}{\sum_{k=1}^N \omega_t^k p(z_{t+1}, \tilde{\xi}_{t+1}^j \mid \xi_{1:t}^k, y_{1:t})} \delta_{\xi_{1:t}^i}(d\xi_{1:t}) dz_{t+1} \\ &= \sum_{i=1}^N p(z_{t+1} \mid \tilde{\xi}_{t+1:T}^j, y_{1:T}) \tilde{\omega}_{t|T}^{i,j}(z_{t+1}) \delta_{\xi_{1:t}^i}(d\xi_{1:t}) dz_{t+1}, \end{aligned} \tag{5.57}$$

where we have introduced the z_{t+1}-dependent weights,

$$\tilde{\omega}_{t|T}^{i,j}(z_{t+1}) \triangleq \frac{\omega_t^i p(z_{t+1}, \tilde{\xi}_{t+1}^j \mid \xi_{1:t}^i, y_{1:t})}{\sum_k \omega_t^k p(z_{t+1}, \tilde{\xi}_{t+1}^j \mid \xi_{1:t}^k, y_{1:t})}. \tag{5.58}$$

To obtain (5.56) we can marginalise (5.57) over $\xi_{1:t-1}$ and z_{t+1}, which results in

$$p(\xi_t \mid \tilde{\xi}_{t+1:T}^j, y_{1:T})\, d\xi_t \approx \sum_{i=1}^N \underbrace{\int p(z_{t+1} \mid \tilde{\xi}_{t+1:T}^j, y_{1:T}) \tilde{\omega}_{t|T}^{i,j}(z_{t+1})\, dz_{t+1}}_{=\theta_t^{i,j}} \delta_{\xi_t^i}(d\xi_t). \tag{5.59}$$

Hence, if we want to sample "directly" from $p(\xi_t \mid \tilde{\xi}_{t+1:T}^j, y_{1:T})$ we need to evaluate the (likely to be intractable) integrals involved in (5.59). If we instead sample from the joint density $p(\xi_{1:t}, z_{t+1} \mid \tilde{\xi}_{t+1:T}^j, y_{1:T})$ we can use the fact that the marginal density $p(z_{t+1} \mid \tilde{\xi}_{t+1:T}^j, y_{1:T})$ is (approximately) Gaussian, and hence easy to sample from. We then only need to evaluate $\tilde{\omega}_{t|T}^{i,j}(z_{t+1})$ at a single point, which is clearly much simpler than evaluating the integrals in (5.59).

6 Nonlinear system identification

So far, we have only considered the state inference problem, i.e. how to estimate the state of a dynamical system given a sequence of measurements. This chapter addresses the related problem of system identification, i.e. how to infer knowledge about the system itself based on the measurements.

6.1 Introduction

System identification is in itself a very broad concept. It is the art of finding, or identifying, models that can describe dynamical systems. This includes a wide variety of tasks, ranging from how to choose appropriate model structures, to how to design input signals enabling an accurate identification. For a thorough treatment of the field, we refer to the standard textbooks by Ljung [1999] and Söderström and Stoica [1989].

Here, we will use the word identification in a more narrow sense, referring to the problem of parameter estimation. Hence, we assume that a model structure is given, but that it is parameterised by some unknown parameters. The task at hand, is then to estimate these parameters based on measurements from the system.

Furthermore, as the title of this chapter indicates, the focus will be on nonlinear (and/or non-Gaussian) systems. Identification of nonlinear systems is probably one of the currently most active areas within the system identification community [Ljung and Vicino, 2005, Ljung, 2008]. This is basically due to its relevance and challenging nature. The presence of nonlinearities suggests that Monte Carlo (MC) integration techniques, as discussed in the previous chapters, can be applied. In fact, during the last decade or so, identification methods based on sequential Monte Carlo (SMC) and related techniques, have appeared at an increasing rate and with increasingly better performance. The two overview papers by Andrieu et al. [2004] and by Kantas et al. [2009], and the recent results by Schön et al. [2011], Olsson et al. [2008], Gopaluni [2008] and Poyiadjis et al. [2009] provide a good introduction to these ideas. In the present chapter, we will continue this line of work and apply the Rao-Blackwellised marginal particle filter (RBMPF) of Section 3.4 and the Rao-Blackwellised particle smoother (RBPS) of Section 5.3 to the problem of nonlinear system identification.

Let us return to the SSM (2.2) on page 13. To perform state inference in this model, we have throughout the previous chapters assumed that the model is fully known. That is, in the filtering and smoothing recursions of Section 2.3, we allowed ourselves to evaluate model quantities such as the distribution ν and the kernel Q. Now, assume that the model is not fully known, but parameterised by some unknown $\theta \in \mathsf{X}_\theta$[1],

$$X_1 \sim \nu_\theta(dx_1), \tag{6.1a}$$

$$X_{t+1} \mid \{X_t = x_t\} \sim Q_\theta(dx_{t+1} \mid x_t), \tag{6.1b}$$

$$Y_t \mid \{X_t = x_t\} \sim G_\theta(dy_t \mid x_t). \tag{6.1c}$$

Based on a sequence of observations $Y_{1:T} = y_{1:T}$, we wish to find an appropriate value for θ. The meaning of the word appropriate in this context, will be clarified in the coming sections. In particular, we will start by introducing the maximum likelihood (ML) criterion, and thereafter the expectation maximisation (EM) algorithm which can be used to solve the ML problem. We then introduce an alternative Bayesian criterion. Based on these two criteria, we propose two nonlinear identification methods. First, in Section 6.2 we discuss Bayesian identification using the RBMPF developed in Section 3.4. Second, in Section 6.3 we consider ML identification using the EM algorithm combined with the RBPS of Section 5.3. In Section 6.3.3 we discuss how this latter approach potentially can be used for identification of Wiener systems. Finally, in Section 6.4 we discuss some properties of the two methods, and of particle based identification methods in general.

6.1.1 Maximum likelihood

Recall the definition of the likelihood function, as the probability density function (PDF) of the measurement sequence. In a parameterised model, the likelihood function will also depend on θ. In fact, since the measurement sequence $y_{1:T}$ is assumed to be fixed, the word function now takes a different meaning. That is, we view the likelihood function as a mapping from the parameter space to the real line,

$$p_\theta(y_{1:T}) : \mathsf{X}_\theta \to \mathbb{R}. \tag{6.2}$$

A sensible approach to parameter identification, is to find a value of θ which maximises the likelihood function. That is, we seek a parameter value for which the observed measurement sequence is "as likely as possible"; this idea is known as maximum likelihood. Hence, we define the ML estimator as,

$$\hat{\theta}^{\mathrm{ML}} = \underset{\theta \in \mathsf{X}_\theta}{\arg\max}\, p_\theta(y_{1:T}). \tag{6.3}$$

The ML criterion was proposed, analysed and popularised by Sir Ronald Aylmer Fisher (1890–1962) in the early 20th century [Fisher, 1912, 1921, 1922]. However, the idea can be traced back even further to, among others, Gauss, Hagen and Edgeworth [Hald, 1999].

[1] Here, X_θ is the set of possible parameter values.

Aldrich [1997] provides a historical discussion on Fisher and the making of ML. Due to its appealing theoretical properties, it has a long tradition in many fields of science, including that of system identification.

When it comes to solving the ML problem using MC integration, a common choice is to apply the EM algorithm, see e.g. the work by Andrieu et al. [2004], Olsson et al. [2008], Schön et al. [2011] and Gopaluni [2008] and the textbooks by McLachlan and Krishnan [2008] and Cappé et al. [2005]. The EM algorithm will be the topic of the next section, and will also be the "method of choice" for addressing the ML problem in Section 6.3. However, an often overlooked alternative is to perform direct maximisation of the likelihood function. In fact, for any model in which the EM algorithm is applicable, Fisher's and Louis' identities (see e.g. [Cappé et al., 2005, page 353]), state that the gradient and the Hessian of the log-likelihood function are also available. This opens up for maximisation of the likelihood function by e.g. MC based steepest ascent or Newton methods. See also the work by Andrieu et al. [2004], Poyiadjis et al. [2005] and Poyiadjis et al. [2009].

6.1.2 Expectation maximisation

The EM algorithm by Dempster et al. [1977] is one of the statistician's standard tools for addressing the ML problem. McLachlan and Krishnan [2008] provides a thorough treatment of the method an discuss its properties and a variety of extensions. We will in this section present the EM algorithm in a similar fashion as it was derived by Dempster et al. [1977]. Hence, the presentation is not exclusive for SSMs, but applies to a wider class of models. In Section 6.3 we will then combine the EM algorithm with the RBPS of Section 5.3, resulting in a method for identification of mixed linear/nonlinear Gaussian SSMs. The application to general SSMs is thoroughly described by e.g. Cappé et al. [2005] and Schön et al. [2011]. Gibson and Ninness [2005] apply the EM algorithm to the special case of linear Gaussian state-space (LGSS) models, and provide details on how to make a robust implementation of the method. Smith and Robinson [2000] discuss the similarities and differences between the EM algorithm and the subspace identification algorithm (see e.g. [Van Overschee and De Moor, 1996]) for identification of LGSS models.

The EM algorithm is a method for ML estimation in incomplete data models. The word *incomplete* refers to the underlying dependence of the observed variable on some hidden or latent variable. In other words, assume that Y is a random variable, for which we observe the value $Y = y$. We then postulate that there exist some *latent* variable Z which can not be observed (compare with an SSM, in which the measurement process is observed, but the state process is not). The pair $\{Z, Y\}$ is known as the complete data, whereas the observed variable Y is the incomplete data. The latent variable Z is really a design variable, and how to choose it is connected to how the variable Y is modeled. In general, the latent variable should be chosen such that "if it would be known, then the identification problem would be simpler". Now, assume that the joint distribution of $\{Z, Y\}$, for a given parameter θ, admits a density $f_\theta(z, y)$ w.r.t. some product measure $\lambda \times \mu$. The likelihood function, i.e. the marginal density function for the observed variable, is then given by,

$$g_\theta(y) = \int f_\theta(z, y)\lambda(dz). \tag{6.4}$$

The ML problem is, in analogy with (6.3), given by

$$\widehat{\theta}^{\mathrm{ML}} = \operatorname*{arg\,max}_{\theta \in \mathsf{X}_\theta} g_\theta(y). \tag{6.5}$$

To maximise the above expression we can, by monotonicity, equally well maximise the log-likelihood function given by,

$$\log g_\theta(y) = \log f_\theta(z, y) - \log p_\theta(z \mid y), \tag{6.6}$$

where we have introduced the conditional density of Z given Y,

$$p_\theta(z \mid y) = \frac{f_\theta(z, y)}{g_\theta(y)}. \tag{6.7}$$

Now, for each value of $\theta \in \mathsf{X}_\theta$, this conditional density is a PDF w.r.t. λ. Hence, by multiplying (6.6) with $p_{\theta'}(z \mid y)$ and integrating w.r.t. λ, we get,

$$\begin{aligned}\log g_\theta(y) &= \int \log f_\theta(z, y) p_{\theta'}(z \mid y) \lambda(dz) - \int \log p_\theta(z \mid y) p_{\theta'}(z \mid y) \lambda(dz) \\ &= \underbrace{\mathrm{E}_{\theta'}[\log f_\theta(Z, y) \mid Y = y]}_{\triangleq \mathcal{Q}(\theta, \theta')} - \underbrace{\mathrm{E}_{\theta'}[\log p_\theta(Z \mid y) \mid Y = y]}_{\triangleq \mathcal{V}(\theta, \theta')}. \end{aligned} \tag{6.8}$$

Before we go on, we note that we need to impose some extra conditions for the above expressions to be well defined. Hence, we make the following assumption.

Assumption A1. The family of densities $\{f_\theta\}_{\theta \in \mathsf{X}_\theta}$ is such that,

i) for any $\theta \in \mathsf{X}_\theta$ the likelihood function $g_\theta(y)$ is positive and finite.

ii) for any $(\theta, \theta') \in \mathsf{X}_\theta \times \mathsf{X}_\theta$ the conditional densities $p_\theta(z \mid y)$ and $p_{\theta'}(z \mid y)$ have the same support.

Assumption A1(i) is needed for (6.6) and (6.7) to make sense. Assumption A1(ii) ensures that the integrals in (6.8) are well defined. In fact, we should interpret the integration to be only over the support of $p_{\theta'}(z \mid y)$, a set on which both $p_\theta(z \mid y)$ and $f_\theta(z, y)$ are ensured to be strictly positive under Assumption A1(ii).

Remark 6.1. The above assumptions may at first seem like technicalities which are not important from a practitioner's point of view. However, especially Assumption A1(ii) can indeed affect the implementation aspects of the EM algorithm. What the assumption states is that the support of the conditional density of the latent variable Z, given the observed variable Y, may not be parameter dependent. For certain problems, this may restrict the freedom that one has in choosing the latent variable. As an example, Wills et al. [2011] consider identification of SSMs on innovation form. For the EM algorithm to be applicable for this specific problem, a "clever choice" of the latent variables is required.

The main ingredient of the EM algorithm is the $\mathcal{Q}$-function, defined in (6.8). The algorithm is based on the following proposition, which states that an increase in the $\mathcal{Q}$-function implies an increase in the log-likelihood function.

Proposition 6.1 (The fundamental EM inequality). *Under Assumption A1 and for any* $(\theta, \theta') \in \mathsf{X}_\theta \times \mathsf{X}_\theta$*,*

$$\log g_\theta(y) - \log g_{\theta'}(y) \geq \mathcal{Q}(\theta, \theta') - \mathcal{Q}(\theta', \theta'), \tag{6.9a}$$

where the inequality is strict unless $p_\theta(\,\cdot \mid y)$ *and* $p_{\theta'}(\,\cdot \mid y)$ *are equal* λ*-a.e. This further implies,*

$$\mathcal{Q}(\theta, \theta') \geq \mathcal{Q}(\theta', \theta') \Rightarrow \log g_\theta(y) \geq \log g_{\theta'}(y). \tag{6.9b}$$

The proposition is a direct consequence of Jensen's inequality

Lemma 6.1 (Jensen's inequality). *If* φ *is a convex function defined over an open interval* I *and* X *is a random variable with* $\mathrm{P}(X \in I) = 1$ *and finite expectation, then*

$$\varphi(\mathrm{E}[X]) \leq \mathrm{E}[\varphi(X)]. \tag{6.10}$$

If φ *is strictly convex, the inequality is strict unless* X *is constant a.s.*

Proof: See e.g. [Lehmann, 1983], page 50. □

Proof of Proposition 6.1: Consider the difference between the log-likelihood function values evaluated at two different parameters θ and θ',

$$\log g_\theta(y) - \log g_{\theta'}(y) = \left(\mathcal{Q}(\theta, \theta') - \mathcal{Q}(\theta', \theta')\right) + \left(\mathcal{V}(\theta', \theta') - \mathcal{V}(\theta, \theta')\right), \tag{6.11}$$

where

$$\mathcal{V}(\theta', \theta') - \mathcal{V}(\theta, \theta') = \int \log \frac{p_{\theta'}(z \mid y)}{p_\theta(z \mid y)} p_{\theta'}(z \mid y) \lambda(dz). \tag{6.12}$$

The above expression is recognised as the Kullback-Leibler divergence between $p_{\theta'}(z \mid y)$ and $p_\theta(z \mid y)$ [Kullback and Leibler, 1951]. By Jensen's inequality,

$$\begin{aligned}\int \log \frac{p_{\theta'}(z \mid y)}{p_\theta(z \mid y)} p_{\theta'}(z \mid y) \lambda(dz) &= \mathrm{E}_{\theta'}\left[-\log \frac{p_\theta(Z \mid y)}{p_{\theta'}(Z \mid y)} \,\middle|\, Y = y \right] \\ \geq -\log \mathrm{E}_{\theta'}\left[\frac{p_\theta(Z \mid y)}{p_{\theta'}(Z \mid y)} \,\middle|\, Y = y \right] &= -\log \int p_\theta(z \mid y) \lambda(dz) = 0, \end{aligned} \tag{6.13}$$

which completes the proof. □

The property (6.9a) suggests that the $\mathcal{Q}$-function can be used as a surrogate for the log-likelihood function in the ML problem. This is exploited in the EM algorithm, given in Algorithm 6.1. A sequence of parameter estimates $\{\theta_k\}_{k \geq 1}$ is constructed in such a way that $\mathcal{Q}(\theta_{k+1}, \theta_k) \geq \mathcal{Q}(\theta_k, \theta_k)$. Thus, (6.9b) implies that the sequence of log-likelihood function values $\{\log g_{\theta_k}(y)\}_{k \geq 1}$ is non-decreasing, meaning that the EM algorithm is a monotone optimisation algorithm. Furthermore, it can be shown that if the sequence of parameter estimates converge to some point $\hat{\theta}^{\mathrm{EM}}$, this is a stationary point also for the log-likelihood function. However, to ensure that the sequence of parameter estimates indeed converges, further conditions need to be imposed. We refer the reader to the paper by Wu [1983] and Chapter 3 of the book by McLachlan and Krishnan [2008] for convergence

Algorithm 6.1 EM algorithm [Dempster et al., 1977]

Input: An initial parameter estimate $\theta_1 \in \mathsf{X}_\theta$.

Output: A parameter estimate $\hat{\theta}^{\mathrm{EM}}$ which is (close to) a local maximiser of the likelihood function.

```
$k \leftarrow 1$.
while not converged do
    E-step (expectation): Compute the $\mathcal{Q}$-function,
        $\mathcal{Q}(\theta, \theta_k) = \mathrm{E}_{\theta_k}[\log f_\theta(Z, y) \mid Y = y] = \int \log f_\theta(z, y) p_{\theta_k}(z \mid y) \lambda(dz).$
    M-step (maximisation): $\theta_{k+1} = \arg\max_{\theta \in \mathsf{X}_\theta} \mathcal{Q}(\theta, \theta_k)$.
    $k \leftarrow k + 1$.
end while
$\hat{\theta}^{\mathrm{EM}} = \theta_k$.
```

results regarding the EM algorithm. In practice, the algorithm is often run until either the increase in the log-likelihood function value, or the difference between two consecutive parameter estimates, is below some threshold. Also, it is common to specify a maximum number of iterations beforehand.

Remark 6.2. We will in general express the M-step of the EM algorithm as a maximisation of the $\mathcal{Q}$-function. However, from (6.9b) we note that it is in fact enough to increase the value of the $\mathcal{Q}$-function to guarantee an increase in the likelihood function. Hence, the M-step can be replaced by an approximate maximisation, and it is not that crucial that the next parameter iterate is close to any true maximiser. However, it is clear that the accuracy of the maximisation can effect the convergence of the algorithm. An EM algorithm with an approximate M-step is sometimes called generalised EM.

6.1.3 Bayesian identification

An alternative view on probability bears the name of the British statistician and reverend Thomas Bayes (1702–1761). The Bayesian probabilist uses the term probability to measure the degree of belief in some hypothesis, which is then said to be true with a certain probability. Before we obtain any "measurements" regarding the validity of the hypothesis, we believe that it is true with some *a priori* probability. After receiving new information, we reevaluate our belief in the hypothesis, which gives rise to an *a posteriori* probability that it is true. This is in contrast with the frequentistic view on probability, to which e.g. Fisher's concept of ML belongs, in which the probability of an event is seen as the frequency of observing the event in a long-run experiment.

Bayes [1764][2] treated the problem of Bayesian inference, but only considered uniform priors [Stiegler, 1982]. The ideas that we today refer to as Bayesian, were to a large extent pioneered and popularised by the French mathematician Pierre-Simon Laplace (1749–

[2] Bayes' essay was published after his death. The essay was found by Richard Price who edited and presented the work. Interestingly enough, Price writes in his introductory remarks to the essay, that he believes that Bayes' theorem helps to prove the existence of a deity [Bayes, 1764].

1827). In a memoir, produced at the age of 25 and supposedly unaware of Bayes' work, Laplace [1774] discovered the more general form of "Bayes' theorem". Stiegler [1986] writes the following about Laplace's memoir:

> *"The influence of this memoir was immense. It was from here that 'Bayesian' ideas first spread through the mathematical world, as Bayes's own article was ignored until 1780 and played no important role in scientific debate until the 20th century. It was also this article of Laplace's that introduced the mathematical techniques for the asymptotic analysis of posterior distributions that are still employed today. And it was here that the earliest example of optimum estimation can be found, the derivation and characterization of an estimator that minimized a particular measure of posterior expected loss. After more than two centuries, we mathematicians, statisticians cannot only recognize our roots in this masterpiece of our science, we can still learn from it."*

Today, the Bayesian ideas form a popular approach to statistical inference. See for instance the book by Denison et al. [2002] for an overview of Bayesian methods in regression and classification problems. The Bayesian approach to system identification is, among others, discussed by Peterka [1981] and Ninness and Henriksen [2010]. Basically, the main difference between a frequentistic and a Bayesian approach to inference, is that in the latter all unknown quantities of the model are seen as random variables. Hence, for the system identification problem, we assume that the parameter θ is a random variable, distributed according to some known *prior distribution* $\pi_{\theta|0}$. This distribution summarises our a priori knowledge about the parameter, i.e. what we know before we make any measurements on the system. Such prior information is sometimes naturally available, e.g. due to physical constraints or insight into the system dynamics based on experience. In other cases, the prior can be completely artificial and introduced simply to enable the application of Bayesian methods. In such cases, it is common to choose a prior which is as uninformative as possible. After observing a measurement sequence $Y_{1:T} = y_{1:T}$, we wish to find the *posterior distribution* of the parameters, defined by,

$$\pi_{\theta|T}(A) = \mathrm{P}(\theta \in A \mid Y_{1:T} = y_{1:T}), \tag{6.14}$$

and Bayes' rule states that

$$\pi_{\theta|T}(d\theta) \propto p_\theta(y_{1:T})\pi_{\theta|0}(d\theta). \tag{6.15}$$

In contrast to the ML approach, in which we sought the point estimate (6.3) under which the observations were as likely as possible, the objective in the Bayesian setting is thus to find a complete probability distribution for the parameter. The posterior distribution summarises everything we know about the parameter, based on the a priori knowledge and the information available in the measurements. Once we have this distribution, we can ask questions like; what is the probability that the parameter lies in a given interval? Hence, it could be claimed that a Bayesian method provides a richer source of information, than for instance ML. However, it should then be remembered that this "additional information" is highly dependent on the prior distribution that we choose. Hence, if we choose the prior poorly, the posterior distribution can be very misleading.

6.2 RBMPF for identification

We shall now see that the RBMPF introduced in Section 3.4 can be a useful tool in solving nonlinear identification problems in a Bayesian setting. We start by introducing a general augmented state-space approach to recursive Bayesian identification. We then evaluate the RBMPF on numerical data in a simulation study.

6.2.1 Augmented state-space approach to identification

As pointed out in the previous section, we can view the parameter θ as a random variable with some prior distribution $\pi_{\theta|0}$. Equivalently, by letting $\Theta_1 = \theta$ and $\Theta_{t+1} = \Theta_t$ for $t \geq 1$, we can view it as a non-varying stochastic process. We can then rewrite the model (6.1) under study as,

$$X_{t+1} \mid \{X_t = x_t, \Theta_t = \theta_t\} \sim Q_{\theta_t}(dx_{t+1} \mid x_t), \tag{6.16a}$$

$$\Theta_{t+1} \mid \{\Theta_t = \theta_t\} \sim \delta_{\theta_t}(d\theta_{t+1}), \tag{6.16b}$$

$$Y_t \mid \{X_t = x_t, \Theta_t = \theta_t\} \sim G_{\theta_t}(dy_t \mid x_t), \tag{6.16c}$$

with initial distribution,

$$\Theta_1 \sim \pi_{\theta|0}(d\theta), \tag{6.16d}$$

$$X_1 \mid \{\Theta_1 = \theta_1\} \sim \nu_{\theta_1}(dx_1). \tag{6.16e}$$

Hence, we have reduced the parameterised model (6.1), to a non-parameterised SSM by augmenting the state to include also the parameters. That is, the state process is taken as $\{X_t, \Theta_t\}_{t\geq 1}$ on the augmented state-space $\mathsf{X} \times \mathsf{X}_\theta$. Clearly, $\Theta_t = \theta$ for any $t \geq 1$. Consequently, this Bayesian parameter identification problem can be solved analogously to the state inference problem. More precisely, the posterior parameter distribution is available as a marginal of the filtering distribution,

$$\pi_{\theta|T}(d\theta) = \int_{\mathsf{X}} \Phi_{T|T}(dx_T, d\theta). \tag{6.17}$$

Since the filtering problem is often solved sequentially, the above expression tells us that the posterior parameter distribution can be computed sequentially as well. That is, to obtain the posterior distribution (6.17) we do not need to process a batch of data $y_{1:T}$ "all at once". Instead, we can compute a sequence of posterior parameter distributions $\pi_{\theta|t}$ for $t \geq 1$. This enables on-line parameter identification, in which the parameter estimate is updated sequentially as more and more data becomes available. This is known as recursive identification; see for instance [Ljung and Söderström, 1983].

Now, if the "original" SSM (6.1) is nonlinear, the same hold for (6.16) and we thus need a nonlinear filter to address the identification problem. Here, the focus will be on SMC methods. One option is then to use the particle filter (PF), discussed in Section 3.2. Unfortunately, due to the static evolution of the Θ-state, a direct application of the PF to this specific problem is bound to fail. The reason is that the exploration of the parameter space is restricted to the first time instant. Once the particles (in the parameter space) are sampled at time $t = 1$, their positions are fixed. At consecutive time instants, the particles will be reweighted and resampled, but not moved to new positions (see Remark 6.3 below). Another approach would be to marginalise out the Θ-state (assuming that this can be done), by employing a Rao-Blackwellised particle filter (RBPF). However, as discussed in Section 3.4, the RBPF will still suffer from degeneracy due to the static Θ-process. In Section 3.4 we proposed the novel RBMPF, equipped with a mixing procedure for updating the linear states, as a way to circumvent these problems with the RBPF. This will be our approach also to the Bayesian identification problem considered in this section. In fact, the RBMPF of Section 3.4 has been developed with SSMs such as (6.16) in mind.

Remark 6.3. An alternative approach to enable the application of the PF and RBPF to a static parameter SSM is to add some artificial dynamic evolution to the Θ-state, and hope that this has negligible effect on the estimates. The artificial dynamics are often of random walk type, with a small and possibly decaying (over time) variance. This technique is sometimes called roughening or jittering. It is employed by for instance Gordon et al. [1993], Kitagawa [1998] and Liu and West [2001], using the PF. Similarly, Schön and Gustafsson [2003] use jittering noise in an RBPF setting. A related idea, investigated by e.g. Stavropoulos and Titterington [2001] and Musso et al. [2001], is to use kernel smoothing of the point-mass distributions arising in the PF.

For the RBMPF to be applicable, the SSM must be conditionally linear Gaussian (recall Definition 2.3 on page 15). We will in particular consider the special case where the kernels Q_θ and G_θ in (6.16), are Gaussian and have an affine dependence on the parameter θ. This special case can thus be expressed as,

$$\begin{aligned} X_{t+1} &= f(X_t) + A(X_t)\Theta_t + V_t, && \text{(6.18a)}\\ \Theta_{t+1} &= \Theta_t, && \text{(6.18b)}\\ Y_t &= h(X_t) + C(X_t)\Theta_t + E_t, && \text{(6.18c)} \end{aligned}$$

with

$$\begin{aligned} V_t &\sim \mathcal{N}(0, Q(X_t)), && \text{(6.18d)}\\ E_t &\sim \mathcal{N}(0, R(X_t)). && \text{(6.18e)} \end{aligned}$$

Furthermore, we assume that the prior parameter distribution (6.16d) is Gaussian and that the initial distribution for the X-process (6.16e) is independent of θ. This model is mixed linear/nonlinear Gaussian (and thus a CLGSS model) and we can apply the RBMPF of Algorithm 3.4 to do simultaneous state and parameter inference (i.e. recursive identification) in the model. This identification method will be evaluated in the coming section.

Remark 6.4. The model (6.18) is not the most general CLGSS model in the family of models defined by (6.16). We could for instance allow either Q or G to be non-Gaussian, if they at the same time are independent of θ. More generally, we could allow for a partitioning of the state X_t or the measurement Y_t into two parts, one Gaussian and parameter dependent and one non-Gaussian and parameter independent. The reason for why we choose to work with the model (6.18) is, again, for notational convenience.

6.2.2 Numerical results

In this section we evaluate the RBMPF method for recursive identification on simulated data. We will consider two examples. First a simple LGSS system, included to gain confidence in the proposed method. We then turn to a nonlinear, challenging identification problem.

The RBMPF will be compared with the RBPF, where the latter uses jittering noise as discussed in Remark 6.3. As suggested by Schön and Gustafsson [2003] and Gustafsson and Hriljac [2003], we apply Gaussian jittering noise with decaying variance on both states and parameters. That is, in the RBPF we modify the model (6.18) by adding jittering noises J_t^x and J_t^θ to the right hand sides of (6.18a) and (6.18b), respectively. These artificial noise sources have variances that decay linearly over time, i.e.

$$J_t^x \sim \mathcal{N}\left(0, (\sigma_x^2/t) I_{n_x \times n_x}\right), \tag{6.19a}$$

$$J_t^\theta \sim \mathcal{N}\left(0, (\sigma_\theta^2/t) I_{n_\theta \times n_\theta}\right). \tag{6.19b}$$

Of course, the jittering noises are internal to the RBPF and are not used when simulating data from the systems.

Example 6.1: RBMPF: 1st order LGSS system

Consider the following first order LGSS system with one unknown parameter,

$$X_{t+1} = \theta X_t + 0.3 U_t + V_t, \tag{6.20a}$$

$$Y_t = X_t + E_t. \tag{6.20b}$$

Here, we have assumed that there exists a known input U_t to the system, which is a realisation of a zero-mean, unit variance Gaussian white noise process. The input is added to excite the system dynamics, making the parameter θ more easily identifiable. The process noise V_t and the measurement noise E_t are zero-mean Gaussian white noise processes, with variances 0.1 and 1, respectively. The initial state of the system is zero-mean Gaussian with variance 0.1.

The comparison was made by a Monte Carlo study over 100 realisations of input and output data $\{u_{1:T}, y_{1:T}\}$ from the system (6.20), each consisting of $T = 1000$ samples (measurements). The true value of the parameter was set to $\theta^\star = -0.8$. Since the identification methods considered here are Bayesian, we model the parameter as a random variable, $\Theta \sim \mathcal{N}(1, 3)$.

The bootstrap RBMPF and two versions of the bootstrap RBPF were run i parallel, all using $N = 200$ particles. The first RBPF did not use any jittering noise and the second RBPF used jittering noise according to (6.19) with $\sigma_x^2 = \sigma_\theta^2 = 0.1$. Figure 6.1 illustrates the parameter estimates from the two RBPFs, over the 100 realisations of data. The corresponding plot for the RBMPF is given in Figure 6.2. As expected, the RBPF without any jittering noise is sensitive to particle degeneracy and several estimates converge to “erroneous” values. This is circumvented by adding jittering noise, but as can be seen in Figure 6.1 this introduces extra variance to the estimates. It is possible that the RBPF will perform better if the jittering noise variance is tuned more carefully. However, to rely on an “optimal” tuning of this parameter is not particularly satisfactory, since such tuning is hard to do in a practical application. In this specific example, the RBMPF performs much better, as can be seen in Figure 6.2. This is also confirmed by Table 6.1, where the Monte Carlo means and standard deviations of the parameter estimates at time $t = T = 1000$ are summarised.

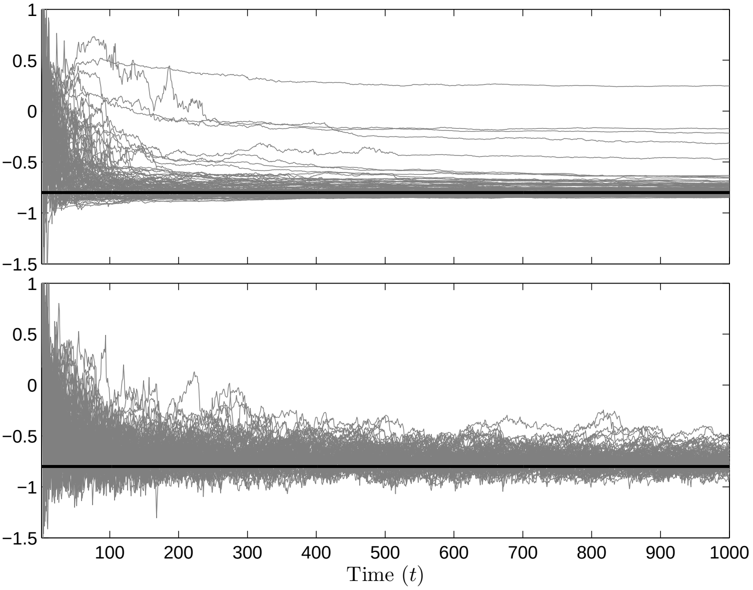

Figure 6.1: *Parameter estimates for the RBPFs without jittering noise (top) and with jittering noise (bottom). The grey lines illustrate the estimates over the 100 realisations of data. The true parameter value is -0.8, indicated with a solid black line.*

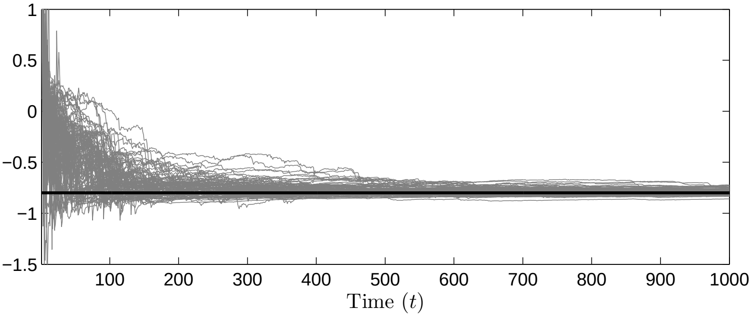

Figure 6.2: *Parameter estimates for the RBMPF. The grey lines illustrate the estimates over the 100 realisations of data. The true parameter value is -0.8, indicated with a solid black line.*

***Table 6.1:** Monte Carlo means and standard deviations*

Method	Mean	Std. dev.
True value ($\theta^\star$)	-0.8	–
RBPF w/o jittering	-0.749	0.148
RBPF w. jittering (variance $0.1/t$)	-0.746	0.089
RBMPF	-0.792	0.026

We continue with a more challenging, nonlinear identification problem.

Example 6.2: RBMPF: 1st order nonlinear system

Consider the first order nonlinear system,

$$X_{t+1} = aX_t + b\frac{X_t}{1+X_t^2} + c\cos(1.2t) + V_t, \tag{6.21a}$$

$$Y_t = dX_t^2 + E_t, \tag{6.21b}$$

where the process noise is given by $V_t \sim \mathcal{N}(0, 0.01)$, the measurement noise is given by $E_t \sim \mathcal{N}(0, 0.1)$ and the initial state of the system is $X_1 \sim \delta_0(dx_1)$. The true parameters are given by $\theta^\star = \begin{pmatrix} a & b & c & d \end{pmatrix}^\mathsf{T} = \begin{pmatrix} 0.5 & 25 & 8 & 0.05 \end{pmatrix}^\mathsf{T}$. This system has been studied e.g. by Andrade Netto et al. [1979] and Gordon et al. [1993] and has become something of a benchmark example for nonlinear filtering. Schön et al. [2011] considers the same system for nonlinear identification, but based on the ML approach using the EM algorithm. They use the same parameter values as we do here, with the exception that they let the process noise be identically zero. Another difference is that they parameterise and estimate also the process and measurement noise variances[3].

The evaluation was made by a Monte Carlo study over 100 realisations of data $y_{1:T}$ from the system (6.21), each consisting of $T = 200$ samples (measurements). The parameters were modeled as Gaussian random variables,

$$\Theta_1 \sim \mathcal{N}\left(\bar{\theta}_{1|0}, \operatorname{diag}\left(\begin{pmatrix} 0.5 & 25 & 8 & 0.05 \end{pmatrix}^\mathsf{T}\right)\right). \tag{6.22}$$

Here, $\bar{\theta}_{1|0} = \begin{pmatrix} \bar{a}_{1|0} & \bar{b}_{1|0} & \bar{c}_{1|0} & \bar{d}_{1|0} \end{pmatrix}^\mathsf{T}$ corresponds to the initial parameter estimate. This vector was chosen randomly for each Monte Carlo simulation, so that each parameter was uniformly distributed over the interval ± 50 % from its true value.

The RBMPF and four versions of the RBPF were run i parallel, all using $N = 500$ particles. The first RBPF did not use any jittering noise, whereas the remaining three versions used jittering noise according to (6.19) with ($\sigma_x^2 = \sigma_\theta^2 = \sigma^2$), $\sigma^2 = 0.01$, $\sigma^2 = 0.1$ and $\sigma^2 = 1$, respectively. In all filters we use local linearisation of the measurement equation for proposal construction (see Section 3.3.4).

Table 6.2 summarises the results from the different filters, in terms of the Monte Carlo means and standard deviations for the parameter estimates extracted at the final time point

[3] The reason for this difference in parameterisation is that the RBMPF is applicable only if the model is conditionally linear Gaussian in the parameters.

$t = T = 200$. Based on these values, we conclude that jittering noise with $\sigma^2 = 0.1$ provides the best tuning for the RBPF, among the values considered here. The results from this filter over the 100 realisations of data, are given in Figure 6.3. A similar plot for the RBMPF is provided in Figure 6.4. It should be noted that the plots are deliberately zoomed in on the true parameter values, so that the estimates at the later time points are clearly visible. At the beginning of the experiments ($t \leq 20$ or so) there are a lot of fluctuations in the estimates, not visible in these figures. It is clear that the jittering noise in the RBPF introduces extra variance to the estimates and also that it slows down the convergence, when compared to the RBMPF.

Table 6.2: *Monte Carlo means and standard deviations*

Method	$a\,(\times 10^{-1})$	b	c	$d\,(\times 10^{-2})$
True value ($\theta^\star$)	5	25	8	5
RBPF ($\sigma^2 = 0$)	4.98 ± 0.110	24.8 ± 3.27	7.96 ± 0.551	5.19 ± 0.918
RBPF ($\sigma^2 = 0.01$)	4.95 ± 0.105	25.0 ± 2.29	7.99 ± 0.322	5.14 ± 0.476
RBPF ($\sigma^2 = 0.1$)	4.86 ± 0.171	22.8 ± 1.32	7.62 ± 0.230	5.89 ± 0.393
RBPF ($\sigma^2 = 1$)	4.68 ± 0.286	18.8 ± 1.11	6.28 ± 1.062	8.68 ± 0.791
RBMPF	5.00 ± 0.030	25.1 ± 0.80	8.03 ± 0.128	4.97 ± 0.220

However, what we have not mentioned so far is that filter divergence was experienced during several identification experiments. By filter divergence, we mean that at some time point, all unnormalised particle weights turned out to be numerically zero. This can occur if the particle support does not provide a good representation of the true support of the filtering distribution. That is, for some reason, the particles are located in the "wrong" part of the state-space. If this occurs, the filter can not proceed. Instead, it was terminated and discarded from the experiment. Hence, the results given in Table 6.2 and in Figure 6.3 and 6.4 are based only on the non-diverged experiments. In Table 6.3 we give the number of filter divergences that occurred during the 100 Monte Carlo runs[4].

Table 6.3: *Number of divergences over 100 experiments*

Method	Divergences
RBPF ($\sigma^2 = 0$)	16
RBPF ($\sigma^2 = 0.01$)	1
RBPF ($\sigma^2 = 0.1$)	0
RBPF ($\sigma^2 = 1$)	0
RBMPF	16

As can be seen, the RBMPF (and also the RBPF without jittering noise) got a lot of divergences, whereas the RBPFs with jittering noise appear to be more robust. It is not that surprising that jittering provides some robustification of the filter. By adding jittering

[4] Note that no divergences occurred during the experiments in Example 6.1

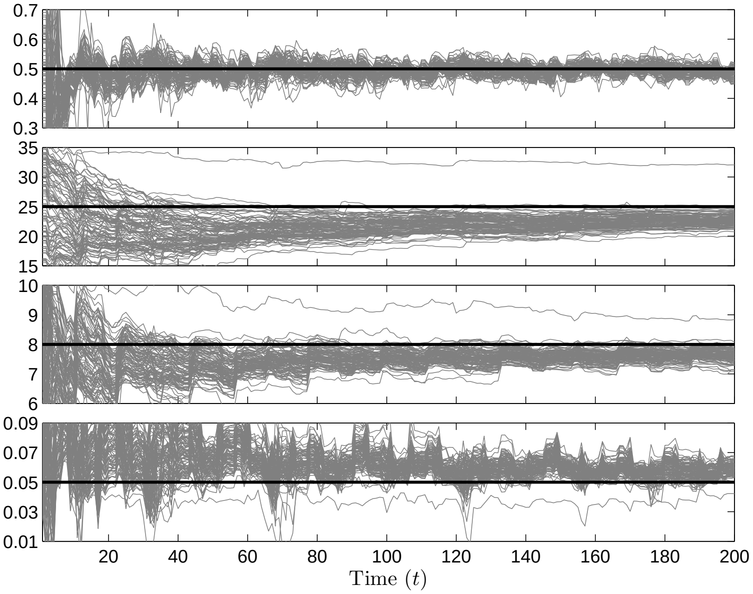

Figure 6.3: Estimates of the parameters a, b, c and d (from top to bottom) for the RBPF using jittering noise with $\sigma_x^2 = \sigma_\theta^2 = 0.1$. The grey lines illustrate the estimates over the different realisations of data. The true parameter values are shown as thick black lines.

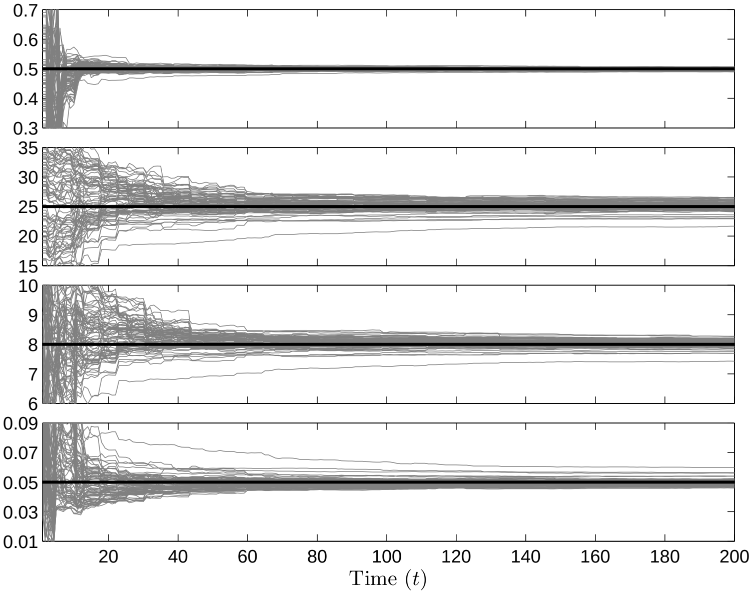

Figure 6.4: Estimates of the parameters a, b, c and d (from top to bottom) for the RBMPF. The grey lines illustrate the estimates over the different realisations of data. The true parameter values are shown as thick black lines.

noise, we basically say that we are not really confident in the estimates provided by the filter. The effect of this is to make the variations in the weight function less pronounced, and in particular to make the weight function decay more slowly toward zero. The RBMPF lacks this type of robustification, which is likely to be one reason for the figures reported in Table 6.3.

Clearly, the high number of divergences for the RBMPF is an unacceptable property of the method, which is not yet fully understood. However, as we will discuss in Section 6.4.3, the lack of robustness is potentially a general problem, common to many particle based identification methods. Hence, the problems experienced here might not be related directly to the RBMPF approach, but rather to the fact that the RBMPF is a particle based identification method. We continue this discussion in Section 6.4.3, where we also suggest some possible approaches to robustify the method.

6.3 RBPS-EM

We will now turn our attention to identification of mixed linear/nonlinear Gaussian state-space models using the EM algorithm. A central component in computing approximations of the $\mathcal{Q}$-function is the RBPS derived in Section 5.3. Hence, the resulting identification algorithm will be referred to as RBPS-EM. The material presented in this section is based on work by Lindsten and Schön [2010].

6.3.1 The RBPS-EM identification method

We assume that we are given a (fully dominated) mixed linear/nonlinear Gaussian state-space model according to (2.9) on page 16, parameterised by $\theta \in \mathsf{X}_\theta$. All components of the model (i.e. the transition and measurement functions, the noise covariances and the initial distributions) may depend on the parameter. In the compact notation introduced in (2.12), we can thus express the model as,

$$X_{t+1} = f_\theta(\Xi_t) + A_\theta(\Xi_t)Z_t + V_t, \tag{6.23a}$$

$$Y_t = h_\theta(\Xi_t) + C_\theta(\Xi_t)Z_t + E_t, \tag{6.23b}$$

where $X_t = \begin{bmatrix}\Xi_t^{\mathsf{T}} & Z_t^{\mathsf{T}}\end{bmatrix}^{\mathsf{T}}$. The process noise and the measurement noise are given by,

$$V_t \sim \mathcal{N}\left(0, Q_\theta(\Xi_t)\right), \tag{6.23c}$$

$$E_t \sim \mathcal{N}\left(0, R_\theta(\Xi_t)\right), \tag{6.23d}$$

respectively. The initial distribution of the process is defined by $\Xi_1 \sim p_\theta(\xi_1)$ and

$$Z_1 \mid \{\Xi_1 = \xi_1\} \sim \mathcal{N}\left(\bar{z}_{\theta,1|0}(\xi_1), P_{\theta,1|0}(\xi_1)\right). \tag{6.23e}$$

We take the ML approach, as discussed in Section 6.1.1, and employ the EM algorithm (see Section 6.1.2). The first design choice that we need to make is to define the latent variables. As pointed out in Section 6.1.2, these should consist of the missing, or unobserved, data of the model. For an SSM, a natural choice is thus to let the latent variables be given by the (unobserved) state process $X_{1:T}$.

Recall from (6.8) that the $\mathcal{Q}$-function is defined as the conditional expectation of the complete data, log-likelihood,

$$\begin{aligned}\mathcal{Q}(\theta,\theta') &= \mathrm{E}_{\theta'}[\log p_\theta(X_{1:T}, y_{1:T}) \mid Y_{1:T} = y_{1:T}] \\ &= \int \log p_\theta(x_{1:T}, y_{1:T}) p_{\theta'}(x_{1:T} \mid y_{1:T})\, dx_{1:T}. \end{aligned} \tag{6.24}$$

Hence, the $\mathcal{Q}$-function is given by an expectation under the joint smoothing distribution. However, due to the special structure of an SSM, the complete data log-likelihood can be expanded according to,

$$\begin{aligned}\log p_\theta(x_{1:T}, y_{1:T}) &= \log p_\theta(x_{1:T}) + \log p_\theta(y_{1:T} \mid x_{1:T}) \\ &= \log p_\theta(x_1) + \sum_{t=1}^{T-1} \log p_\theta(x_{t+1} \mid x_t) + \sum_{t=1}^{T} \log p_\theta(y_t \mid x_t). \end{aligned} \tag{6.25}$$

This suggests that we can decompose the $\mathcal{Q}$-function as,

$$\mathcal{Q}(\theta,\theta') = I_1(\theta,\theta') + I_2(\theta,\theta') + I_3(\theta,\theta'), \tag{6.26}$$

where we have defined,

$$I_1(\theta,\theta') \triangleq \mathrm{E}_{\theta'}\left[\log p_\theta(X_1) \mid Y_{1:T} = y_{1:T}\right], \tag{6.27a}$$

$$I_2(\theta,\theta') \triangleq \sum_{t=1}^{T-1} \mathrm{E}_{\theta'}\left[\log p_\theta(X_{t+1} \mid X_t) \mid Y_{1:T} = y_{1:T}\right], \tag{6.27b}$$

$$I_3(\theta,\theta') \triangleq \sum_{t=1}^{T} \mathrm{E}_{\theta'}\left[\log p_\theta(y_t \mid X_t) \mid Y_{1:T} = y_{1:T}\right]. \tag{6.27c}$$

Hence, we only need to compute expectations under the marginal smoothing distribution (for I_1 and I_3) and the 2-step, fixed-interval smoothing distribution (for I_2).

However, before we turn to the actual computation of these expectations, we note that the expressions (6.27) can be expanded even further. This is enabled by the fact that both the transition and the measurement density functions are Gaussian. To avoid a repeated and notationally cumbersome presentation, we shall restrict ourselves to one of the terms above, namely I_2 defined in (6.27b). The terms I_1 and I_3 follow analogously.

Using the relationship $x^{\mathsf{T}}Mx = \operatorname{tr}(Mxx^{\mathsf{T}})$, the transition density function can be expressed as,

$$\begin{aligned}-2\log p_\theta(x_{t+1}|x_t) &= -2\log p_{V,\theta}\left(x_{t+1} - f(\xi_t) - A(\xi_t)z_t\right) \\ &\cong \log\det Q(\xi_t) + \operatorname{tr}\left(Q(\xi_t)^{-1}\ell_2(\xi_{t:t+1}, z_{t:t+1})\right), \end{aligned} \tag{6.28a}$$

where

$$\ell_2(\xi_{t:t+1}, z_{t:t+1}) \triangleq \left(x_{t+1} - f(\xi_t) - A(\xi_t)z_t\right)\left(x_{t+1} - f(\xi_t) - A(\xi_t)z_t\right)^{\mathsf{T}}, \tag{6.28b}$$

and $\cong$ means equality up to an additive constant, independent of the parameters θ. For notational convenience we have dropped the dependence on θ from the right hand side, but remember that f, A, Q, h, C, R, $\bar{z}_{1|0}$, $P_{1|0}$ and $p(\xi_1)$ may in fact be θ-dependent.

Inserting this into (6.27b) results in,

$$I_2(\theta, \theta') \cong -\frac{1}{2}\sum_{t=1}^{T-1} \mathrm{E}_{\theta'}\left[\log\det Q(\Xi_t) + \operatorname{tr}\left(Q(\Xi_t)^{-1}\ell_2(X_{t:t+1})\right) \mid Y_{1:T} = y_{1:T}\right]. \tag{6.29}$$

The expectations under the 2-step fixed-interval smoothing distribution, involved in the expression above, are in general intractable. Hence, to proceed from here, we make use of the RBPS approximation (5.47) on page 106. This provides an approximation of (6.29) according to,

$$\hat{I}_2(\theta, \theta') = -\frac{1}{2M}\sum_{t=1}^{T-1}\sum_{j=1}^{M}\left(\log\det Q(\tilde{\xi}_t^j) + \operatorname{tr}\left(Q(\tilde{\xi}_t^j)^{-1}\hat{\ell}_{2,t}^j\right)\right), \tag{6.30}$$

where

$$\hat{\ell}_{2,t}^j \triangleq \mathrm{E}_{\theta'}\left[\ell_2(\tilde{\xi}_{t:t+1}^j, Z_{t:t+1}) \,\middle|\, \Xi_{1:T} = \tilde{\xi}_{1:T}^j, Y_{1:T} = y_{1:T}\right]. \tag{6.31}$$

Observe that the expectation here is taken only over the (approximately) Gaussian Z-variables, conditioned on the nonlinear Ξ-variables. The nontrivial parts of the conditional expectation above are the terms,

$$\mathrm{E}_{\theta'}[Z_t Z_t^{\mathsf{T}} \mid \Xi_{1:T} = \tilde{\xi}_{1:T}^j, Y_{1:T} = y_{1:T}] = \tilde{z}_{t|T}^j \tilde{z}_{t|T}^{j\,T} + \widetilde{P}_{t|T}^j, \tag{6.32a}$$

$$\mathrm{E}_{\theta'}[Z_t Z_{t+1}^{\mathsf{T}} \mid \Xi_{1:T} = \tilde{\xi}_{1:T}^j, Y_{1:T} = y_{1:T}] = \tilde{z}_{t|T}^j \tilde{z}_{t+1|T}^{j\,T} + M_{t|T}^j, \tag{6.32b}$$

where $\tilde{z}_{t|T}^j$, $\widetilde{P}_{t|T}^j$ and $M_{t|T}^j$ are given in (5.45) on page 106. Analogously, we obtain approximations of I_1 and I_3, according to,

$$\hat{I}_1(\theta, \theta') = -\frac{1}{2M}\sum_{j=1}^{M}\left(\log\det P_{1|0}(\tilde{\xi}_1^j) + \operatorname{tr}\left(P_{1|0}(\tilde{\xi}_1^j)^{-1}\hat{\ell}_1^j\right) - 2\log p(\tilde{\xi}_1^j)\right), \tag{6.33a}$$

$$\hat{I}_3(\theta, \theta') = -\frac{1}{2M}\sum_{t=1}^{T}\sum_{j=1}^{M}\left(\log\det R(\tilde{\xi}_t^j) + \operatorname{tr}\left(R(\tilde{\xi}_t^j)^{-1}\hat{\ell}_{3,t}^j\right)\right), \tag{6.33b}$$

where,

$$\hat{\ell}_1^j \triangleq \mathrm{E}_{\theta'}\left[\ell_1(\tilde{\xi}_1^j, Z_1) \,\middle|\, \Xi_{1:T} = \tilde{\xi}_{1:T}^j, Y_{1:T} = y_{1:T}\right], \tag{6.34a}$$

$$\hat{\ell}_{3,t}^j \triangleq \mathrm{E}_{\theta'}\left[\ell_3(\tilde{\xi}_t^j, Z_t) \,\middle|\, \Xi_{1:T} = \tilde{\xi}_{1:T}^j, Y_{1:T} = y_{1:T}\right], \tag{6.34b}$$

$$\ell_1(\xi_1, z_1) \triangleq \left(z_1 - \bar{z}_{1|0}(\xi_1)\right)\left(z_1 - \bar{z}_{1|0}(\xi_1)\right)^{\mathsf{T}}, \tag{6.34c}$$

$$\ell_3(\xi_t, z_t) \triangleq \left(y_t - h(\xi_t) - C(\xi_t)z_t\right)\left(y_t - h(\xi_t) - C(\xi_t)z_t\right)^{\mathsf{T}}. \tag{6.34d}$$

Putting the pieces together we obtain an RBPS based approximation of the $\mathcal{Q}$-function, given by,

$$\widehat{\mathcal{Q}}(\theta, \theta') = \hat{I}_1(\theta, \theta') + \hat{I}_2(\theta, \theta') + \hat{I}_3(\theta, \theta') \approx \mathcal{Q}(\theta, \theta') + \text{const.} \tag{6.35}$$

By replacing the $\mathcal{Q}$-function in the EM algorithm with this approximation, we end up with the RBPS-EM identification method, summarised in Algorithm 6.2.

Algorithm 6.2 RBPS-EM [Lindsten and Schön, 2010]

Input: A measurement sequence $y_{1:T}$, a parameterised model in the form (6.23), an initial parameter estimate $\theta_1 \in \mathsf{X}_\theta$ and some termination criterion.

Output: A parameter estimate $\widehat{\theta}^{\text{RBPS-EM}}$.

```
$k \leftarrow 1$.
while not converged do
    Parameterise the model (6.23) using the current parameter estimate $\theta_k$.
    Smoothing: Run the RB-FFBSi (Algorithm 5.7) and store the backward trajectories $\{\tilde{\xi}^j_{1:T}\}_{j=1}^M$ and the corresponding sufficient statistics for the linear states $\{\tilde{z}^j_{t|T}, \widetilde{P}^j_{t|T}, M^j_{t|T}\}_{j=1}^M$ for $t = 1, \dots, T$ ($M^j_{t|T}$ only for $t < T$).
    Approximate the Q-function: Let,
        $$\widehat{\mathcal{Q}}(\theta, \theta_k) = \hat{I}_1(\theta, \theta_k) + \hat{I}_2(\theta, \theta_k) + \hat{I}_3(\theta, \theta_k),$$
    where $\hat{I}_1$, $\hat{I}_2$ and $\hat{I}_3$ are given by (6.33a), (6.30) and (6.33b), respectively.
    Maximisation: Set,
        $$\theta_{k+1} = \arg\max_{\theta \in \mathsf{X}_\theta} \widehat{\mathcal{Q}}(\theta, \theta_k)$$
    $k \leftarrow k + 1$.
end while
$\widehat{\theta}^{\text{RBPS-EM}} = \theta_k$.
```

6.3.2 Numerical results

In this section we will evaluate the RBPS-EM identification method on simulated data. Two different examples will be presented, first with a linear Gaussian system and thereafter with a mixed linear/nonlinear Gaussian system. The example systems can be recognised from the evaluation of the RB-FFBSi presented in Section 5.3.5. However, here we have parameterised the systems with supposedly unknown parameters.

As before, the purpose of including a linear Gaussian example is to gain confidence in the proposed method. For this case, there are closed form solutions available for all the involved calculations (see [Gibson and Ninness, 2005] for all the details in a very similar setting). The smoothing densities can for this case be explicitly calculated using the RTS recursions [Rauch et al., 1965]. The resulting identification method, combining the RTS smoother with the EM algorithm, will be denoted RTS-EM.

For both the linear and nonlinear examples, we can clearly also address the estimation problem using standard PS based methods, as is done by Schön et al. [2011]. More precisely, we use the Fast FFBSi by Douc et al. [2010] (see Section 5.2) for computing the expectations in the EM algorithm. This approach will be denoted PS-EM. Finally, we have the option to employ the proposed RBPS-EM method presented in Algorithm 6.2.

For all methods, the maximisation of the $\widehat{\mathcal{Q}}$-function (i.e. the M-step of the algorithm) is performed using a BFGS quasi-Newton method, see e.g. [Nocedal and Wright, 2000, Chapter 6]. The gradients of the cost function are approximated using finite differences. Note that we do not need to solve the optimisation problem in the M-step, just find a parameter value which increases the value of the $\widehat{\mathcal{Q}}$-function (see Remark 6.2 on page 120)[5].

Example 6.3: RBPS-EM: 2nd order LGSS system

Consider the linear, second order system with a single unknown parameter θ given by,

$$\begin{pmatrix}\Xi_{t+1}\\ Z_{t+1}\end{pmatrix} = \begin{pmatrix}0.8 & \theta\\ 0 & 1\end{pmatrix}\begin{pmatrix}\Xi_t\\ Z_t\end{pmatrix} + V_t, \qquad V_t \sim \mathcal{N}(0, Q), \tag{6.36a}$$

$$Y_t = \Xi_t + E_t, \qquad E_t \sim \mathcal{N}(0, R), \tag{6.36b}$$

with $Q = 0.01 I_{2\times 2}$ and $R = 0.1$. The initial state of the system is Gaussian according to

$$\begin{pmatrix}\Xi_1\\ Z_1\end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix}0\\ 5\end{pmatrix}, \begin{pmatrix}10^{-6} & 0\\ 0 & 10^{-6}\end{pmatrix}\right). \tag{6.37}$$

In RBPS-EM, the first state Ξ_t is treated as if it is nonlinear, whereas the second state Z_t is treated as linear.

The comparison was made by a Monte Carlo study over 100 realisations of data $y_{1:T}$ from the system (6.36), each consisting of $T = 200$ samples (measurements). The true value of the parameter was set to $\theta^\star = 0.1$. The three identification methods, RTS-EM, PS-EM and RBPS-EM were run in parallel for 200 iterations of the EM algorithm. The smoothers used the same settings as in Example 5.2. In particular, the particle methods all used $N = M = 50$ particles/backward trajectories. The initial parameter estimate θ_1 was set to 0.2 in all experiments.

Table 6.4 gives the Monte Carlo means and standard deviations for the parameter estimates.

Table 6.4: *Monte Carlo means and standard deviations*

Method	Mean ($\times 10^{-2}$)	Std. dev. ($\times 10^{-2}$)
RTS-EM [Gibson and Ninness, 2005]	10.01	0.61
PS-EM [Schön et al., 2011]	10.04	1.59
RBPS-EM [Lindsten and Schön, 2011]	10.03	0.63

On average, all methods converge to values very close to the true parameter value 0.1. The major difference is in the standard deviations of the estimated parameter. For RTS-EM and RBPS-EM, the standard deviations are basically identical, whereas for PS-EM it is more than twice as high. This is in agreement with the results in Example 5.2. There we saw that for this specific example, the RB-FFBSi and the RTS smoother had similar performance, both superior to the FFBSi.

We continue with an example with a mixed linear/nonlinear Gaussian system, similar to the one considered in Example 5.3. Since the system is nonlinear, RTS-EM is not applicable. We thus make the comparison only between PS-EM and RBPS-EM.

[5] For the linear model in Example 6.3 we could have solved the optimisation problem in the M-step analytically, but for simplicity we employed a numerical optimisation routine for this example as well. The effects of this on the results should be negligible.

Example 6.4: RBPS-EM: 4th order mixed linear/nonlinear Gaussian system

Consider the fourth order mixed linear/nonlinear Gaussian system, where three of the states are conditionally linear Gaussian, given by,

$$\Xi_{t+1} = \arctan \Xi_t + \begin{pmatrix} a & 0 & 0 \end{pmatrix} Z_t + V_t^{\xi}, \tag{6.38a}$$

$$Z_{t+1} = \begin{pmatrix} 1 & 0.3 & 0 \\ 0 & b\cos(c) & -b\sin(c) \\ 0 & b\sin(c) & b\cos(c) \end{pmatrix} Z_t + V_t^z, \tag{6.38b}$$

$$Y_t = \begin{pmatrix} 0.1\Xi_t^2 \operatorname{sign}(\Xi_t) \\ 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 1 & -1 & 1 \end{pmatrix} Z_t + E_t, \tag{6.38c}$$

with $V_t = \begin{bmatrix} V_t^{\xi} & (V_t^z)^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}} \sim \mathcal{N}(0, Q)$, $Q = 0.01 I_{4\times 4}$ and $E_t \sim N(0, R)$, $R = 0.1 I_{2\times 2}$. The initial distribution for the system is $X_1 \sim \delta_0(dx_1)$. The system is parameterised by $\theta = \begin{pmatrix} a & b & c \end{pmatrix}^{\mathsf{T}}$ and the true parameter vector is $\theta^\star = \begin{pmatrix} 1 & 0.968 & 0.315 \end{pmatrix}^{\mathsf{T}}$. The Z-system is oscillatory and marginally stable, with poles in 1 and $0.92 \pm 0.3i$. The linear Z-variables are connected to the nonlinear Ξ-system through $Z_{1,t}$.

First, we assume that the Z-system is known, i.e. we are only concerned with finding the parameter a connecting the two systems. Again, we consider a Monte Carlo study with 100 realisations of data $y_{1:T}$, each consisting of $T = 200$ samples. The parameter a was thereafter identified by running RBPS-EM and PS-EM in parallel for 500 iterations. Both methods used $N = M = 50$ particles/backward trajectories. The initial parameter estimate was chosen randomly from a uniform distribution over the interval $[0,\ 2]$ for each simulation.

As for the RBMPF experiments (see Section 6.2.2) we again encountered divergences of the identification procedures. By a divergence we mean that one of the particle methods used in the identification, at some stage ran into numerical problems caused by the particle weights being all numerically zero. If this occurred, the method was terminated and the result discarded from the experiment. Hence, only the results from the non-diverged experiments are reported below. For the 100 realisations of data, PS-EM got 19 divergences and RBPS-EM got 3 divergences. We discuss this issue further in Section 6.4.3.

Figure 6.5 illustrates the convergence of the parameter estimates for the two methods. For RBPS-EM, the Monte Carlo mean and the standard deviation of the final parameter estimate was $\hat{a}_{500}^{\text{RBPS-EM}} = 0.996 \pm 0.066$. For PS-EM the corresponding figures were $\hat{a}_{500}^{\text{PS-EM}} = 1.05 \pm 0.145$. Also in this example, the parameter variance is much higher for PS-EM than for RBPS-EM, which is obvious from Figure 6.5 as well.

Now, let us assume that all the parameters $\theta = \begin{pmatrix} a & b & c \end{pmatrix}^{\mathsf{T}}$ are unknown. Once again, PS-EM and RBPS-EM were run in parallel on 100 realisations of data $y_{1:T}$ with $T = 200$. To ensure that the algorithms had time to converge, we increased the number of iterations in the EM algorithm to 1000. Also, since we expect the problem to be more challenging, we increased the number of particles and simulated backward trajectories to $N = M = 200$ for both methods. Apart from this, all the settings were as before. The parameters were initialised randomly from uniform distributions, a in the interval $[0,\ 2]$, b in the interval $[0, 1]$ and c in the interval $[0, \pi/2]$ (i.e. the poles of the Z-system were initiated randomly in the first and the fourth quadrants of the unit circle in the complex plane).

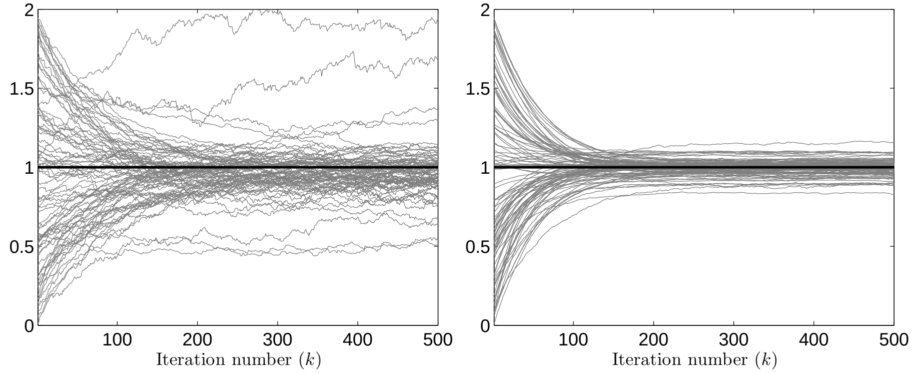

Figure 6.5: *Estimates of the parameter a as functions of iteration number k for PS-EM (left) and RBPS-EM (right). Each grey line corresponds to one realisation of data. The true parameter value is $a^\star = 1$, indicated with a thick black line.*

As before, we encountered divergences due to numerical problems in the particle methods. For the 100 realisations of data, PS-EM had 33 divergences and RBPS-EM had 6 divergences. In the results reported below, only the non-diverged experiments are used.

Figure 6.6 and Figure 6.7 illustrate the convergence of the parameter estimates for PS-EM and for RBPS-EM, respectively. We notice a couple of interesting facts about these results. First, if we consider the results from RBPS-EM in Figure 6.7, the method seems to be very slow to converge for some data realisations. The estimates (in particular for parameter c) lingers for hundreds of iterations of the EM algorithm, before rapidly moving into the "correct" area. This property is not experienced for PS-EM, illustrated in Figure 6.6. However, to further investigate this peculiarity we have used two different line styles when plotting the estimates for RBPS-EM in Figure 6.7. The solid lines correspond to data realisations for which both RBPS-EM *and* PS-EM did not diverge. The dashed lines, on the other hand, correspond to data realisations for which PS-EM (but clearly not RBPS-EM) diverged. Hence, for these experiments we did not get any estimates at all for PS-EM. As can be seen in the figure, the experiments that are slow to converge all correspond to data realisations for which PS-EM encountered numerical problems leading to a divergence. Hence, it seems as if the convergence speed for any fixed data realisation, is related to the tendency of running into numerical problems. This property requires further investigation and is left as future work.

A second interesting fact which can be seen in the figures is related to the variances of the parameter estimates. For any single experiment, it is clear that the MC variance for RBPS-EM is lower than for PS-EM, i.e. there are less "fluctuations" in the parameter estimates for any single data realisation (corresponding to a single grey line in the figures). However, when considering the "total" variance of the parameter estimates over the 100 experiments, they are pretty much the same. This is confirmed by the results given in Table 6.5, where the Monte Carlo means and standard deviations of the final parameter estimates (at $k = 1000$) over the different data realisations are given. This result can be understood by noting that there are different sources of variance for the randomised esti-

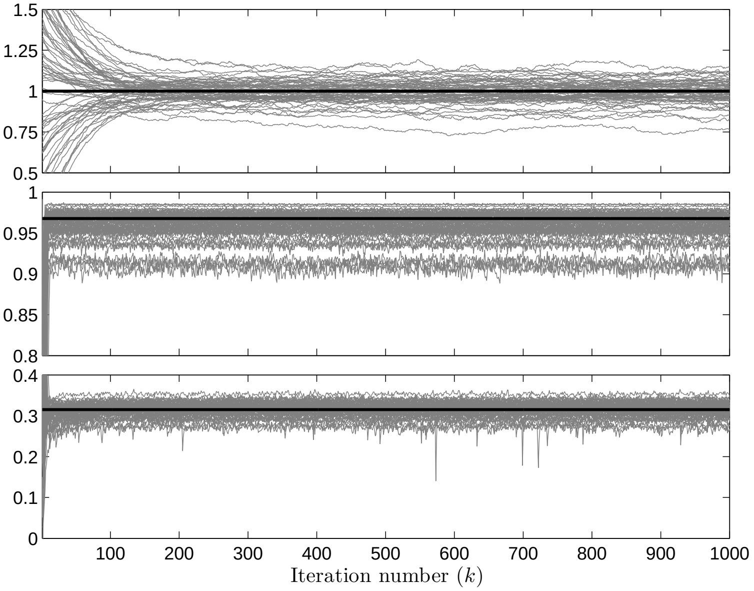

Figure 6.6: *Estimates of the parameters a, b and c (from top to bottom) for PS-EM, plotted versus the iteration number k. The grey lines illustrate the estimates over the different realisations of data. The true parameter values are shown as thick black lines.*

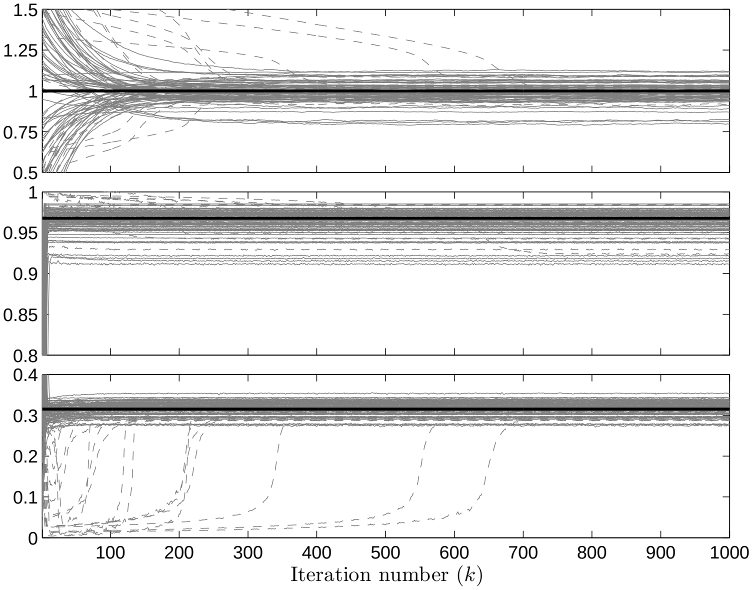

Figure 6.7: *Estimates of the parameters a, b and c (from top to bottom) for RBPS-EM, plotted versus the iteration number k. The grey lines illustrate the estimates over the different realisations of data (see the text for details). The true parameter values are shown as thick black lines.*

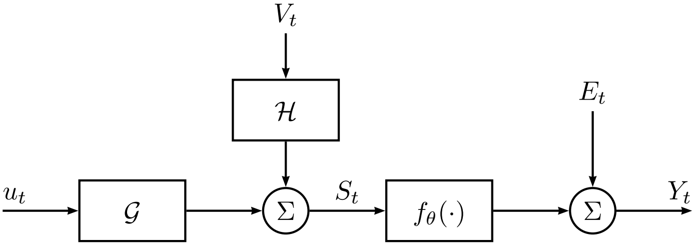

Figure 6.8: *Block diagram of a fairly general class of Wiener systems.*

mators generated by PS-EM and RBPS-EM. The particle methods that are used will introduce some MC variance, but there is also variance originating from the randomness in the data. Only the first of these variances can be reduced by using RBPS-EM instead of PS-EM. In the experiments leading up to the results reported in Figure 6.5, we used very few particles ($N = M = 50$). Consequently, the MC variance was quite high and we gained a lot from using RBPS-EM in place of PS-EM. For the results given in Table 6.5 we used more particles ($N = M = 200$), which reduced the MC variance and made the difference between the two methods less pronounced. We discuss this further in Section 6.4.1.

Table 6.5: *Monte Carlo means and standard deviations*

Method	a	b	c
True value ($\theta^\star$)	1	0.968	0.315
PS-EM	0.995 ± 0.0702	0.960 ± 0.0168	0.315 ± 0.0170
RBPS-EM	0.991 ± 0.0578	0.963 ± 0.0153	0.315 ± 0.0165

6.3.3 Wiener system identification

We will now discuss one potential application of the RBPS-EM identification method given in the previous section, namely identification of Wiener systems[6]. The material of the present section can be put in the category of “future work”. Hence, we will throughout this section pose a set of questions, rather than trying to provide any answers to them.

A Wiener system is a linear dynamical system with a static, nonlinear transformation of the output. A fairly general class of Wiener systems is given by the block diagram shown in Figure 6.8. Here, u_t is an input signal to the linear system $\mathcal{G}$, V_t is a process noise which is colored by the linear system $\mathcal{H}$, f_θ is a static nonlinearity and Y_t is the measured output. The process noise V_t and the measurement noise E_t are assumed to be mutually independent Gaussian white noise processes.

Wiener models are used in a range of different practical applications, such as process industry [Norquay et al., 1999] and biology [Hunter and Korenberg, 1986]. Consequently,

[6] The idea presented in this section can straightforwardly be applied also for Hammerstein-Wiener system, i.e. with static nonlinearities on both input and output.

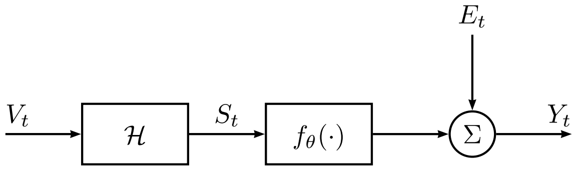

Figure 6.9: *Block diagram of a class of Wiener systems with no deterministic input.*

there is an extensive literature on identification of Wiener systems, see e.g. the work by Wigren [1993], Westwick and Verhaegen [1996], Hagenblad et al. [2008] and Wills and Ljung [2010].

To simplify the presentation, we will again assume that no deterministic input is present, and consider a reduced class of Wiener systems illustrated in Figure 6.9. Based on observations $Y_{1:T} = y_{1:T}$ we now seek to estimate the linear system $\mathcal{H}$ and also the nonlinear mapping f_θ, which we assume is parameterised by $\theta \in \mathsf{X}_\theta$. This problem is sometimes referred to as the blind identification problem (see e.g. [Abed-Meraim et al., 1997]), since we only measure the output from the system.

Wills et al. [2011] considers this problem, using a fully parameterised SSM for the linear block $\mathcal{H}$ and a PS-EM identification method. However, due to the structure that is present in the problem, it is also possible to employ the RBPS-EM method. This can be seen by writing the linear block $\mathcal{H}$ on observer canonical form, which is always possible if the system is observable. Hence, the Wiener system can be modelled as,

$$X_{t+1} = \begin{pmatrix} -a_1 & 1 & 0 & \cdots & 0 \\ -a_2 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_{n_x-1} & 0 & 0 & \cdots & 1 \\ -a_{n_x} & 0 & 0 & \cdots & 0 \end{pmatrix} X_t + \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n_x-1} \\ b_{n_x} \end{pmatrix} V_t, \tag{6.39a}$$

$$S_t = \begin{pmatrix} 1 & 0 & \cdots & 0 \end{pmatrix} X_t, \tag{6.39b}$$

$$Y_t = f_\theta(S_t) + E_t, \tag{6.39c}$$

where $\{X_t\}_{t \geq 1}$ is the state process for the linear system $\mathcal{H}$. Now, due to the structure of (6.39b) we see that only the first state $X_{1,t}$ enters the static nonlinearity. Hence, if we define,

$$\Xi_t = X_{1,t}, \tag{6.40a}$$

$$Z_t = \begin{pmatrix} X_{2,t} & \cdots & X_{n_x,t} \end{pmatrix}^{\mathsf{T}}, \tag{6.40b}$$

the model (6.39) can be recognised as being mixed linear/nonlinear Gaussian.

Guided by (6.39) we may also consider an alternative parameterisation of the Wiener model according to,

$$X_{t+1} = AX_t + V_t', \tag{6.41a}$$

$$S_t = \begin{pmatrix} 1 & 0 & \cdots & 0 \end{pmatrix} X_t, \tag{6.41b}$$

$$Y_t = f_\theta(S_t) + E_t, \tag{6.41c}$$

with

$$V_t' \sim \mathcal{N}(0, Q'), \tag{6.41d}$$

$$E_t \sim \mathcal{N}(0, R). \tag{6.41e}$$

Hence, we consider a fully parameterised dynamic equation (the matrices $A \in \mathbb{R}^{n_x \times n_x}$ and $Q' \in S_+(n_x)$ are fully parameterised), but we keep the structure given by (6.41b). That is, the "C-matrix" of the linear block is fixed to keep the mixed linear/nonlinear structure. Clearly, (6.41) contains (6.39) as a special case, and is thus general enough to contain any observable system of order n_x.

One reason for why the parameterisation (6.41) might be preferable over (6.39) is to obtain a faster convergence of the identification procedure. In empirical studies, it has been experienced that the EM algorithm converges more slowly if more structure is introduced in the parameterisation of the model. Hence, if the same data is used to identify two different models using the parameterisations (6.39) and (6.41), respectively, the method using (6.41) tends to converge faster. This difference in convergence speed has been experienced even though the two methods converge basically to the same linear model, only with different state realisations. The reason for why the convergence speed is influenced by the parameterisation in this way is not fully understood, and it is a topic which requires further attention.

Furthermore, there is another major difference between the parameterisations (6.39) and (6.41). In the latter, we parameterise the process noise covariance Q' as an arbitrary nonnegative definite matrix. If we further assume that this matrix is full rank, then the model (6.41) is fully dominated. This is not the case for the model given by (6.39). More precisely, let the (scalar) process noise V_t in (6.39a) be given by $V_t \sim \mathcal{N}(0, q)$. Then, with

$$B = \begin{pmatrix} b_1 & \cdots & b_{n_x} \end{pmatrix}^\mathsf{T}, \tag{6.42}$$

if follows that the additive noise term in (6.39a) is given by $BV_t \sim \mathcal{N}(0, qBB^\mathsf{T})$, where the covariance matrix $Q = qBB^\mathsf{T}$ is clearly of rank one. Due to this, the model is not fully dominated and we will therefore encounter problems when applying an FFBSm or an FFBSi particle smoother (see Chapter 5).

To get some intuition for why this will be problematic, assume that a PF has been applied to a linear system with a singular process noise covariance. Assume further that we apply an FFBSi smoother to simulate "smoothing trajectories" backward in time. At time $t+1$ we have obtained a backward trajectory $\tilde{x}_{t+1:T}$ and wish to append a sample from time t. In the FFBSi we would then draw a sample from the set of forward filter particles $\{x_t^i\}_{i=1}^N$ with probabilities given by the smoothing weights $\{\tilde{w}_{t|T}^i\}_{i=1}^N$; see (5.10) and (5.11) on page 92. However, if the process noise covariance is singular, it can be realised that all but one of these probabilities almost surely will be zero. The reason for this is that for a given x_t^i, the transition kernel $Q(dx_{t+1} \mid x_t^i)$ will put all "probability mass" on a (low-dimensional) subspace. The probability that $\tilde{x}_{t+1}$ lies in this subspace is zero, unless $\tilde{x}_{t+1}$ in fact was generated conditioned on x_t^i in the forward filtering pass. The effect of this is that any smoothing trajectory that is sampled in the backward simulation will almost surely be *identical* to one of the particle trajectories generated by the forward filter. Hence, we do not gain anything by applying an FFBSi (or an FFBSm). This too, i.e. how particle methods can be used to address the smoothing problem in this type of degenerated models, is a problem which requires further attention.

Finally, we mention another peculiarity which arises when working with particle methods in models with singular process noise. Assume that we apply an RBPF to the model (6.39), using a state partitioning according to (6.40). Assume further that the initial linear state Z_1 is known, meaning that the covariance function $P_{1|0}$ in (2.11) on page 17 is identically zero. It is then easy to verify that the covariance function for the linear state $P_{t|t-1}$ will remain zero for any $t \geq 1$, given that the process noise covariance Q is of rank one. Hence, for such models, the RBPF reduces to a standard PF, and the two methods will result in identical estimates regardless of the dimension of the system. What implications this has for the possible benefits from using Rao-Blackwellisation in Wiener system identification, is also a topic for future work.

6.4 Discussion

We conclude this chapter on particle based, nonlinear system identification by commenting on some of the encountered properties and peculiarities of the identification methods.

6.4.1 RBPS-EM variance reduction

The basic motivation for using the RBPS-EM identification method instead of PS-EM, is to reduce the variance of the parameter estimates. The intuition behind the method is as follows. The particle based smoothers give rise to some MC variance in the state estimates. This variance is, in some way, propagated through the EM algorithm, introducing an MC variance to the (randomised) estimators defined by the algorithm. Assume that, by using the RBPS instead of the PS, we manage to reduce the MC variance of the state estimates (cf. the variance reduction of the RBPF discussed in Chapter 4). Then, this should have the effect of reducing the MC variance of the parameter estimates as well.

However, we must remember that if we do not fix the measurement sequence, the ML estimator (6.3) is in it self a random variable (which in most cases has a nonzero variance). Hence, the variance of the parameter estimates generated by the EM algorithm will typically be nonzero, even if we solve the smoothing problem exactly (cf. Example 6.3 in which the RTS smoother is exact). Informally, we can thus divide the variance of an estimator generated by a particle based EM algorithm into two parts, one originating from the variations in the data and one coming from the MC nature of the particle method. It is only the latter part, i.e. the MC variance, that we can hope to reduce by using RBPS-EM instead of PS-EM. How large the MC variance is compared to the variance of the ML estimator, is likely to be strongly problem dependent. Consequently, this should be the case also for the potential benefits of using RBPS-EM instead of PS-EM.

6.4.2 RBMPF vs. RBPS-EM

In this chapter we have discussed two different methods for nonlinear system identification. We have chosen not to compare the methods with each other, and the reason for this is that they have quite different properties. In this section we make a short summary of these properties, to serve as a guideline for how and when the two methods are applicable.

Type of identification

The two methods use different identification criteria. RBMPF is a Bayesian method in which the parameters are modelled as Gaussian random variables. Consequently, we also need to specify some prior (Gaussian) distribution for the parameters. The RBPS-EM method, on the other hand, is based on an ML criterion. Furthermore, the RBMPF is a recursive identification procedure, whereas RBPS-EM is a batch method.

Type of models

The RBMPF is designed for identification of (typically) nonlinear systems with an affine parameter dependence. The state process or the measurement process is allowed to be non-Gaussian, if it at the same time is parameter independent (see Remark 6.4 on page 123). The RBPS-EM method is developed for identification of mixed linear/nonlinear Gaussian state-space models (or more generally CLGSS models), and the parameter dependence can be quite arbitrary. However, if there is a “complicated” nonlinear parameter dependence in the model, the M-step of the algorithm will become more challenging.

Computational complexity

One of the main drawbacks with the RBMPF is that its computational complexity is quadratic in the number of particles, i.e. the complexity is of order $O(N^2T)$. In Section 3.4.3 we discussed some potential ways in which this complexity can be reduced. When it comes to RBPS-EM, by using the fast backward simulation technique in the RB-FFBSi (see Algorithm 5.6), the complexity of a single smoothing pass is (roughly) $O(NT)$. However, since RBPS-EM is an iterative method, the total computational complexity of an identification experiment is $O(KNT)$, where K is the total number of iterations in the EM algorithm. Consequently, RBPS-EM is also a rather slow method. It would be interesting to investigate possible speedups of the method, to increase its practical applicability.

6.4.3 Robustness of particle based identification methods

In Section 6.2.2 and Section 6.3.2 we saw that both the RBMPF and the RBPS-EM method were non-robust to variations in the simulated data. The effect of this was that the particle weights, at some stage of the algorithm, all turned out to be numerically zero. When this occurred the method could not proceed and we got a “divergence”.

This “robustness issue” is potentially a general problem, common to many particle based identification methods. In this section we will discuss the issue further, hopefully providing some insight into the problem. We will also propose some possible directions for future work, with the incentive of developing robust, particle based identification methods.

We start with an example regarding ML parameter estimation in a nonlinear system.

Example 6.5: Non-robust likelihood computation

Let us consider the nonlinear system (6.21) once again. Assume that the parameters a, b and c are known, i.e. the only unknown parameter of the system which we wish to estimate is d. We take a ML approach and use a grid based optimisation method to maximise the log-likelihood function. That is, we grid the parameter space over the interval $[0.01,\, 0.1]$ using 19 equally spaced grid points (recall that the true parameter value is 0.05). We then compute an estimate of the log-likelihood function value at each grid point. To enable this we observe that the log-likelihood function can be written as,

$$\log p_\theta(y_{1:T}) = \sum_{t=1}^{T} \log p_\theta(y_t \mid y_{1:t-1}), \tag{6.43a}$$

where

$$p_\theta(y_t \mid y_{1:t-1}) = \int p_\theta(y_t \mid x_t) p_\theta(x_t \mid y_{1:t-1})\, dx_t. \tag{6.43b}$$

To estimate the quantities (6.43b) we employ a bootstrap PF. Hence, for each $t = 1, \ldots, T$ we generate an equally weighted particle system $\{x_t^i, 1/N\}_{i=1}^N$ targeting the 1-step predictive density $p_\theta(x_t \mid y_{1:t-1})$ (this particle system is obtained after resampling and mutation, but before the samples are weighted using the measurement density function). By plugging the empirical distribution defined by this particle system into (6.43b), we get an approximation according to,

$$p_\theta(y_t \mid y_{1:t-1}) \approx \frac{1}{N} \sum_{i=1}^{N} p_\theta(y_t \mid x_t^i) = \frac{1}{N} \sum_{i=1}^{N} w_t'^{\,i}. \tag{6.44}$$

Hence, the quantity $p_\theta(y_t \mid y_{1:t-1})$ can be approximated by the sum of the unnormalised particle weights at time t.

Now, if the likelihood function in some part of the parameter space is close to zero, we thus expect that this is the case also for the unnormalised importance weight. When computing these weights in the PF we may, due to the randomness of the method and the insufficiency of the numerical precision, find that they turn out to be all equal to zero. If this occurs, it is not possible to proceed and the filter is terminated.

In Figure 6.10 we plot the estimated log-likelihood function value over the grid in the parameter space. Using $N = 100$ particles, we were only able to estimate the log-likelihood function over the interval $0.015 \leq d \leq 0.06$. Outside of this interval (the grey area in the figure), the PF was terminated prematurely since the unnormalised weight sum turned out to be numerically zero. No estimate of the log-likelihood function value was thus obtained for the grid points in the grey area. If we increase the number of particles to $N = 1000$, we are able to estimate the log-likelihood function over a larger interval in the parameter space without running into numerical problems, but still not for all grid points.

Many identification methods, e.g. the RBMPF and the RBPS-EM presented in this thesis, use some initialisation of the parameters, from which the estimates are updated iteratively.

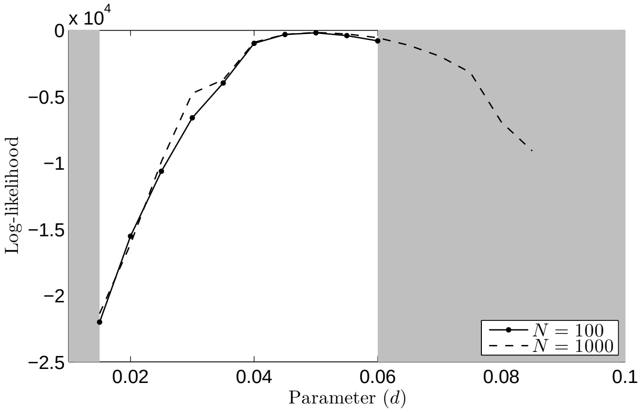

***Figure 6.10:** Estimated log-likelihood function over a grid of parameter values using two PFs with $N = 100$ particles (solid-dotted line) and $N = 1000$ particles (dashed line), respectively. The grey area shows the region in which the PF using $N = 100$ particles encountered numerical problems in the weight evaluation.*

Now, assume that the parameters are initialised to values for which the log-likelihood function is small. Based on the above example, we are then likely to experience a “divergence” of the identification method. In the example we considered ML estimation. However, the effect of “being located” in a part of the parameter space where the likelihood function is close to zero, should reasonably be the same for other approaches as well, such as the Bayesian RBMPF. Furthermore, we note that these robustness issues should be even worse if the likelihood function is “peaky” around a certain value. This property is otherwise something that is desirable, since it generally implies that the variance of the parameter estimates will be low.

Informally, to get around these robustness issues, we would like to find a way of moving from the “bad” areas of the parameter space into a “good” area, without encountering numerical problems on the way. Let us consider the identification problem of Example 6.5 again. In this example, the measurement noise was zero-mean Gaussian with variance 0.1. Hence, when computing the unnormalised weights in (6.44), we evaluate a Gaussian density at N points. Now, due to the exponential decay of the Gaussian density, if all these points are far from the mean, the unnormalised weights are likely to turn out to be numerically zero. One idea to circumvent this problem, is to replace the Gaussian density in the weight evaluation with some other, similar function which decays more slowly. This will in effect mean that we model the measurement noise using some non-Gaussian distribution. This approach is illustrated in the example below.

Example 6.6: Robust likelihood computation

Consider the identification experiment in Example 6.5 again. The measurement noise is known to be zero-mean Gaussian with variance 0.1. However, to circumvent the numerical problems encountered in Example 6.5, we model the measurement noise as being zero-mean Student’s t-distributed with (the same) variance 0.1 and 3 degrees of freedom. Student’s t-distribution (also referred to as simply the t-distribution) resembles the Gaus-

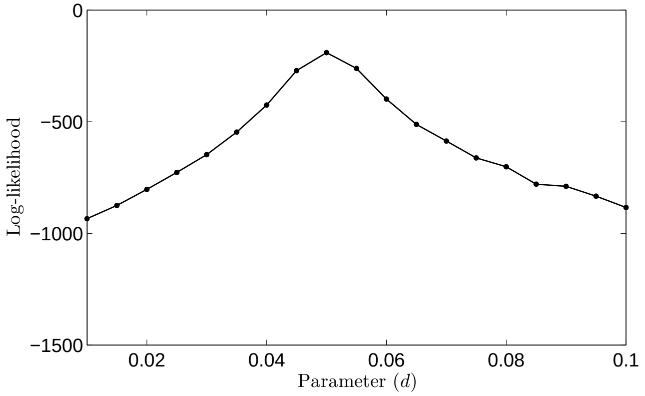

Figure 6.11: *Estimated log-likelihood function over a grid of parameter values using a PF with $N = 100$ particles. The Gaussian measurement density function has been replaced by Student's t-density in the weight evaluation in the PF.*

sian distribution, but is more heavy tailed. It is sometimes described as a generalisation of the Gaussian distribution, which is robust to outliers.

Student's t-distribution is used in place of the Gaussian distribution when computing the weights in the PF. Otherwise, the experiment setup is just as in Example 6.5 and we use the same data (i.e. the data is generated using a Gaussian measurement noise). Figure 6.11 shows the estimated log-likelihood function for parameter values in the interval $[0.01, 0.1]$ (evaluated at the grid points). The function value can be evaluated at all grid points, without encountering any numerical problems. The maximum value is attained (just as in Example 6.5) for $d = 0.05$. It should be noted that there is a big difference in the scaling of the vertical axis, between Figure 6.10 and Figure 6.11. The estimated log-likelihood function using Student's t-distribution, is in fact much flatter than when the Gaussian distribution is used. The maximum function value (attained at $d = 0.05$) is roughly -200 for both approaches.

In the example above, we saw that we got a numerically more robust evaluation of the log-likelihood function, by modelling the measurement noise as Student's t-distributed. In this example, the maximum of the likelihood was attained for the same parameter value as when the original model was used. Still, an obvious criticism against this approach is that we deliberately use an erroneous model for the process, which naturally should affect the parameter estimates "negatively". One possible way to get around this, is to introduce an auxiliary parameter ν, corresponding to the degrees of freedom of the t-distribution. The motivation for this is that Student's t-distribution can be seen as a generalisation of the Gaussian distribution. As ν goes to infinity, the t-distribution tends to a Gaussian. Hence, by initialising ν to some small value (say 2–3), we get a heavy tailed distribution which should be more robust to numerical problems. However, since the system in fact is Gaussian, we expect ν to increase as we get closer to a maximum of the likelihood function, making the modified model more similar to the original model.

In “purely” particle based identification methods (e.g. PS-EM), this approach can straightforwardly be applied, since they in general do not require Gaussian measurement noise. However, for the identification methods considered in this thesis, we do require a certain structure, which may not be compatible with a non-Gaussian measurement noise. Take for instance the RBPS-EM method presented in Section 6.3. This method is designed for identification of mixed linear/nonlinear Gaussian state-space models as in (6.23). For such models, the measurement noise is in general assumed to be Gaussian.

Remark 6.5. A mixed linear/nonlinear Gaussian state-space model remains in the class of CLGSS models even for non-Gaussian measurement noise, if we at the same time require that the measurement equation is independent of Z_t. See also Remark 6.4 on page 123.

However, there are some ways to incorporate a t-distributed noise also in this model class. A first, simple approach is to use the t-distribution only when evaluating the particle weights. When e.g. updating the linear states in the RBPF, we keep a Gaussian representation as before. The effects of this approach on the estimates obtained from the RBPF and RBPS, need further investigation. A second idea is based on an alternative interpretation of the t-distribution. Let τ be a Gamma distributed random variable with shape parameter $\nu/2$ and scale parameter $2\lambda/\nu$, for some ν, $\lambda > 0$. Let the random variable E, conditioned on τ, be zero-mean Gaussian with variance τ^{-1}. Then, the marginal distribution of E is zero-mean Student’s t, with ν degrees of freedom and precision parameter λ. This fact can be used to model the measurement noise in the model (6.23) as t-distributed, without violating the mixed linear/nonlinear Gaussian structure of the model. This is achieved by writing the model as,

$$X_{t+1} = f_\theta(\Xi_t) + A_\theta(\Xi_t)Z_t + V_t, \tag{6.45a}$$

$$\tau_{t+1} \sim \text{Gam}\left(\frac{\nu}{2}, \frac{2\lambda_\theta(\Xi_t)}{\nu}\right), \tag{6.45b}$$

$$Y_t = h_\theta(\Xi_t) + C_\theta(\Xi_t)Z_t + E_t, \tag{6.45c}$$

where $X_t = \begin{bmatrix}\Xi_t^{\mathsf{T}} & Z_t^{\mathsf{T}}\end{bmatrix}^{\mathsf{T}}$ and the process noise and measurement noise are given by,

$$V_t \sim \mathcal{N}(0, Q_\theta(\Xi_t)), \tag{6.45d}$$

$$E_t \sim \mathcal{N}\left(0, \tau_t^{-1}\right), \tag{6.45e}$$

respectively. Here we have assumed that the measurement Y_t is scalar, but the approach can be straightforwardly extended to vector valued measurements. By including τ_t in the nonlinear part of the state, the model above is mixed linear/nonlinear Gaussian[7]. The precision parameter (function) $\lambda_\theta(\,\cdot\,)$ is related to the variance of the t-distributed measurement noise E_t, i.e. if this variance is assumed to be known (or of known structure), $\lambda_\theta(\,\cdot\,)$ can be chosen accordingly. The degrees of freedom ν can, as mentioned above, be seen as an auxiliary parameter which is estimated alongside θ. Alternatively, we can let the degrees of freedom be a deterministic, increasing (in the iteration number of the algorithm) sequence, to exploit the fact that the “true” noise is known to be Gaussian. One drawback with this method is that the nonlinear state dimension is increased by the

[7] Strictly speaking, (6.45) is not mixed linear/nonlinear Gaussian as defined by (6.23), due to the non-Gaussianity of (6.45b). However, since this state equation is independent of Z_t the model is still a CLGSS.

dimension of τ_t, which in the general case would equal the measurement dimension. It remains to investigate the potential benefits from this approach, in terms of robustification of e.g. RBPS-EM.

As a final remark of this section, a possible generalisation of (6.45) is to include a dynamic evolution of the τ-state in (6.45b). It would be interesting to analyse the effect of this on the descriptive properties of the model.

7

Concluding remarks

In this chapter, we summarise the conclusions drawn from the results and analyses presented in the previous chapters. We also lay out some directions for future work.

7.1 Conclusions

We have considered Rao-Blackwellisation of particle methods for the two related problems of state inference and parameter estimation in nonlinear dynamical systems. The basic idea underlying the methods presented and analysed in this thesis, is to exploit a certain type of tractable substructure in the model under study. More precisely, we considered a class of models for which the joint smoothing density can be factorised according to,

$$p(\xi_{1:t}, z_{1:t} \mid y_{1:t}) = p(z_{1:t} \mid \xi_{1:t}, y_{1:t}) p(\xi_{1:t} \mid y_{1:t}), \tag{7.1}$$

and where the conditional density $p(z_{1:t} \mid \xi_{1:t}, y_{1:t})$ is analytically tractable. Here, $x_t = \{\xi_t, z_t\}$ is the state of the system, ξ_t is the “nonlinear state” and z_t is the “linear state”. The primary method built on this idea is the Rao-Blackwellised particle filter (RBPF) by Doucet et al. [2000a] and Schön et al. [2005]. In Section 3.3 we gave a self-contained derivation of the RBPF. The key enabler of the RBPF is that a sequential Monte Carlo method can be used to sequentially generate “nonlinear” weighted sample *trajectories* $\{\xi^i_{1:t}, \omega^i_t\}_{i=1}^N$, targeting the state-marginal smoothing density $p(\xi_{1:t} \mid y_{1:t})$. This means that the conditional filtering densities $p(z_t \mid \xi^i_{1:t}, y_{1:t})$ (for each particle trajectory $i = 1, \ldots, N$) can be computed sequentially as well. For a conditionally linear Gaussian state-space (CLGSS) model, this is done by equipping each particle (i.e. each nonlinear state trajectory) with a Kalman filter.

Besides filtering, we have in this thesis also been concerned with the smoothing problem.

For this problem, it is not as straightforward to make use of the Rao-Blackwellisation idea. The reason is that in a smoothing pass, we typically wish to change or update the nonlinear state trajectories generated by the filter. This means that we break the conditions under which the filtering density for the linear states is analytically tractable. In Section 5.3 we derived a Rao-Blackwellised particle smoother (RBPS) based on the forward filtering/backward simulation idea. To circumvent the above mentioned problems we were forced to make certain approximations (see Section 5.3.3). As mentioned above, the RBPS of Section 5.3 is based on the forward/backward recursions. However, it is likely that we would encounter similar challenges for other types of smoothing as well. Take for instance the two-filter formula. If we were to design a Rao-Blackwellised two-filter particle smoother, we need to construct a Rao-Blackwellised backward filter. However, just because the model contains a tractable substructure in the forward direction, this need not necessarily be the case for the time-reversed model. In conclusion, as opposed to the RBPF, there is no single "natural" way to construct an RBPS and how to apply the Rao-Blackwellisation idea for particle smoothing is still to a large extent an open research question. This is discussed further in [Lindsten and Schön, 2011].

The main motivation behind the RBPF or any RBPS, is to obtain better estimates than what is provided by a particle filter (PF) or a particle smoother (PS). In particular, we wish to reduce the variances of the estimates. In Section 4.2 we analysed the asymptotic variance for the RBPF, compared it with that of the standard PF and computed an explicit expression for the variance reduction. For the case of a bootstrap PF and a bootstrap RBPF, we could directly conclude that the asymptotic variance of the RBPF never is larger than that of the PF. Still, as argued in Section 4.2 it is not always beneficial to use Rao-Blackwellisation. The reason is that the RBPF in general is computationally more expensive per particle than the PF. Hence, for a fixed computational capacity, we can either employ the RBPF, or use a PF with more particles. Unfortunately, we were not able to draw any stronger conclusions or to supply any simple rules-of-thumb regarding this issue, based on the variance reduction expression. The reason is that, even though we obtained an explicit expression, it is not straightforward to compute the variance reduction for a given model. As mentioned in Section 4.2.5, one possibility is to estimate the variance reduction, e.g. by using an RBPF.

The main focus in this thesis has been on nonlinear system identification, using Rao-Blackwellised particle methods. In particular, we presented two different identification methods, the Rao-Blackwellised marginal particle filter (RBMPF) and RBPS expectation maximisation (RBPS-EM). In Chapter 6 we made a numerical evaluation of the methods. The RBMPF was compared with the RBPF using jittering noise, which has previously been suggested for recursive parameter estimation [Schön and Gustafsson, 2003]. One of the main advantages of the RBMPF over the RBPF is that it does not require any tuning of a jittering noise. We also noted that the variances of the RBMPF estimates were lower than those of the RBPF estimates, supposedly because the RBMPF avoids an additional variance in the parameter estimates introduced by the jittering. One of the main drawbacks of the RBMPF is that it seems to be non-robust, in the sense that we got several divergences of the method due to numerical problems. This robustness issue was discussed in more detail in Section 6.4.3, where we argued that it in fact can be a general problem for particle based identification methods.

The RBPS-EM method has been designed as an alternative to PS-EM for identification of models with a tractable substructure. The motivation is to reduce the variance of the parameter estimates. Consequently, the RBPS-EM method was primarily compared with PS-EM. From the numerical results (see Section 6.3.2) we conclude that the parameter variance indeed can be reduced by using Rao-Blackwellisation of the underlying particle smoother. However, we also note that how large this variance reduction is, relative to the total variance of the estimates, seems to be highly problem dependent. As for the RBMPF, we encountered numerical problems in the RBPS-EM and the PS-EM methods. However, the number of divergences was generally lower for RBPS-EM than for PS-EM, even in the cases where the variances of the parameter estimates did not differ much between the methods.

To summarise, Rao-Blackwellisation can be a very useful technique when addressing state inference and identification problems using particle methods. However, how large the possible benefits are, seems to be highly problem dependent. To understand the applicability and the potential of this technique, much work remains to be done, which leads us into the next section.

7.2 Future work

Throughout this thesis, we have encountered a range of different problems which require further attention. Perhaps most notably is the robustness issue with particle based identification methods discussed in Section 6.4.3. Future work on this topic can hopefully result in generally applicable procedures to robustify the identification methods.

The different state inference and identification methods discussed in the thesis, can most likely be improved in various ways. In Section 3.4.3 we mentioned possible modifications of the RBMPF, e.g. with the incentive to reduce the computational complexity of the method. To sort out the details of these modifications, and also to analyse their implications, is a topic for future work. Furthermore, it would be interesting to address the "RBMPF problem" using a different approximation procedure than what we used in Section 3.4. One possible approach is to employ a forward-smoother for additive functionals (see e.g. [Cappé et al., 2005, Del Moral et al., 2010]) to estimate the sufficient statistics for the conditional filtering distribution (see also [Smidl, 2011]).

The RB-FFBSi presented in Section 5.3 is, as pointed out in the previous section, only one possible approach to Rao-Blackwellised particle smoothing. A very interesting topic for future work is to sort out in which way the tractable substructures used in the RBPF, can be exploited also in an RBPS. Furthermore, during the derivation of the RB-FFBSi in Section 5.3 we were forced to make certain approximations. To what extent such approximations are needed is not fully understood and requires further investigation.

Regarding the RBPS-EM identification method, there are several possible directions for future work. Besides a robustification of the method, as mentioned above, it would be interesting to analyse and compare the different sources of variance of the parameter estimates. By doing so, we can hopefully determine whether or not there is a large potential gain in using RBPS-EM instead of PS-EM, for a given problem. Furthermore, as discussed in Section 6.3.3, the RBPS-EM method can possibly be used for identification of Wiener systems. However, to sort out the details of this approach is another topic for future work. Here, we also encountered models with singular process noise. For such models, the application of FFBSm and FFBSi type of particle smoothers will be problematic. This is in itself an interesting issue which requires further attention.

Finally, there are many possible directions for future work, which we have not mentioned in the previous chapters. One such direction is to analyse the class of models for which the RBPF is applicable. Given a nonlinear model, we may ask if it is possible to find some change of variables, which will transform this model into e.g. a CLGSS model. To have a systematic way of finding such transformations, if they exist, would be of great value. Another idea is to use the RBPF and the RB-FFBSi together with the particle MCMC method by Andrieu et al. [2010] and Olsson and Rydén [2010]. If this is feasible, it will result in a Bayesian identification method as an alternative to the RBPS-EM method for identification of mixed linear/nonlinear Gaussian state-space models.

A

Measure, integration and probability

This appendix provides a very brief introduction to measure, integration and probability. The purpose is to make the thesis more self-contained and accessible for readers without any measure theoretic background. Hopefully, this short introduction will provide sufficient insight for a comfortable understanding of the material of the thesis. For further reference on measure theory and probability, see any of the standard textbooks on the subject, e.g. by Billingsley [1995], Chung [2001] or Loève [1977, 1978]. See also the book by Royden and Fitzpatrick [2010] for a treatment of measure theory in a non-probabilistic setting.

A.1 Measure

The fundamental concept in measure theory is that of measuring the sizes of sets. The most natural example is how to assign length, area or volume to a set in one-, two- or three-dimensional Euclidian space. As a generalisation of these concepts, a measure μ is a set-function, assigning real numbers to sets in some space Ω. The value assigned to the set $A \subseteq \Omega$, denoted $\mu(A)$, is in some sense the "size" of A under μ.

However, before we go on with the definition of a measure, let us reflect over the domain of a set-function μ. Let Ω be an arbitrary space of points ω. Then, the domain of μ is naturally some class of subsets of Ω. The most extensive class is of course that of all subsets of Ω, and at first it may be tempting to let this class be the domain of μ. However, it is in general not possible to construct a satisfactory theory for such extensive classes of sets. Consequently, we need to restrict the domain of μ to some suitable class $\mathcal{F}$ of subsets of Ω. It is natural to require that this restriction is made in such a way that Ω itself belongs to $\mathcal{F}$, and also that $\mathcal{F}$ is closed under countable set-theoretic operations. More generally, we will consider classes of subsets with the following properties,

i) $\Omega \in \mathcal{F}$.

ii) $A \in \mathcal{F} \Rightarrow A^{\mathrm{c}} \in \mathcal{F}$.

iii) $A_1, A_2, \ldots \in \mathcal{F} \Rightarrow A_1 \cup A_2 \cup \ldots \in \mathcal{F}$.

Equivalently, we could have replaced condition (iii) with

iii′) $A_1, A_2, \ldots \in \mathcal{F} \Rightarrow A_1 \cap A_2 \cap \ldots \in \mathcal{F}$.

A class $\mathcal{F}$ of subsets of Ω with these properties is known as a σ-algebra and the pair $(\Omega, \mathcal{F})$ is called a measurable space. Generally, we will assume that the domain of any set function is a σ-algebra.

Let us now return to the concept of a measure. As mentioned above, a set-function μ on $\mathcal{F}$ assigns a real number to any set $A \in \mathcal{F}$. However, all such set-functions do not coincide with what we intuitively mean by a measure. Hence, it is natural to impose further constraint on μ for it to be called a measure. We thus make the following definition.

Definition A.1 (Measure). A set-function μ on a σ-algebra $\mathcal{F}$ is a measure if:

i) $\mu(\emptyset) = 0$.

ii) $\mu(A) \in \mathbb{R}_+ \cup \{\infty\}$ for $A \in \mathcal{F}$.

iii) If $A_1, A_2, \ldots$ is a *disjoint* sequence of $\mathcal{F}$-sets, then

$$\mu\left(\bigcup_{k=1}^{\infty} A_k\right) = \sum_{k=1}^{\infty} \mu(A_k).$$

The first two conditions in the definition are rather natural; the empty set has measure zero and an arbitrary set has a nonnegative measure. Usually, we also allow for sets of infinite measure, as indicated by the definition. The third condition is countable additivity. What the condition says is that, for any disjoint sequence of $\mathcal{F}$-sets, measuring the union of all sets is the same thing as measuring them individually and summing up the results. This is a generalisation of the natural additivity property of e.g. length and area. However, it should be noted that countable additivity, as in the definition above, is a stronger property than finite additivity.

If $\mu(\Omega) < \infty$, μ is said to be *finite*, and naturally, if $\mu(\Omega) = \infty$ it is said to be *infinite*. If Ω can be partitioned into some finite or countable collection of $\mathcal{F}$-sets $A_1, A_2, \ldots$, such that $\mu(A_k) < \infty$ for all k, then μ is said to be *σ-finite*.

If the following two examples, we introduce two important special cases of measures.

Example A.1: Counting measure

Let Ω be a countable set and let $\mathcal{F}$ be the class of all subsets of Ω. For $A \in \mathcal{F}$, let $\mu(A)$ be the number of elements in A, or $\mu(A) = \infty$ if A is not finite. This measure is (one example of) *counting measure*.

Example A.2: Lebesgue measure

Let Ω be d-dimensional Euclidian space $\mathbb{R}^d$. A bounded d-dimensional rectangle is a set given by,

$$A = \{ \begin{pmatrix} x_1 & \cdots & x_d \end{pmatrix}^{\mathsf{T}} : a_i < x_i \le b_i, i = 1, \ldots, d \}, \tag{A.1}$$

for some $-\infty < a_i < b_i < \infty$, $i = 1, \ldots, d$. Let $\mathcal{F}$ be the smallest[1] σ-algebra, containing all bounded rectangles. We say that $\mathcal{F}$ is *generated* by the bounded rectangles. This σ-algebra is known as the *Borel* σ-algebra on $\mathbb{R}^d$, and will generally be written $\mathcal{B}(\mathbb{R}^d)$. A set $A \in \mathcal{B}(\mathbb{R}^d)$ is known as a Borel set. This class of sets is very general. For instance, it contains all open and all closed subsets of $\mathbb{R}^d$.

It can be shown that there exists a unique measure λ on $\mathcal{B}(\mathbb{R}^d)$ which assigns to (A.1) the volume,

$$\lambda(A) = \prod_{i=1}^{d} (b_i - a_i). \tag{A.2}$$

This measure is the d-dimensional *Lebesgue measure*.

A.2 Integration

Tightly coupled to measure theory is integration. The aim of the present section is to introduce the integral of a function $f : \Omega \to \mathbb{R}$ w.r.t. a measure μ, written,

$$\mu(f) = \int f \, d\mu = \int f(\omega) \mu(d\omega). \tag{A.3}$$

Let us first assume that the function f is nonnegative and simple[2]. Let $\{x_1, \ldots, x_N\}$ be the range of f and let A_k be the set on which it takes on the value x_k, i.e.

$$A_k = \{\omega : f(\omega) = x_k\}, \qquad k = 1, \ldots, N. \tag{A.4}$$

If we define the indicator function of a set A according to,

$$I_A(\omega) = \begin{cases} 1 & \text{if } \omega \in A, \\ 0 & \text{otherwise}, \end{cases} \tag{A.5}$$

we can write f as,

$$f(\omega) = \sum_{k=1}^{N} x_k I_{A_k}(\omega). \tag{A.6}$$

Now, assume further that $\{A_k\}_{k=1}^N$ is a partitioning of Ω into $\mathcal{F}$-sets where $(\Omega, \mathcal{F})$ is a measurable space and μ is a measure on $\mathcal{F}$. Then, we define the integral of f w.r.t. μ

[1] That is, $\mathcal{F}$ is the intersection of all σ-algebras containing the bounded rectangles.

[2] A function is called simple if it has a finite range.

according to,

$$\int f\,d\mu = \sum_{k=1}^{N} x_k \mu(A_k), \tag{A.7}$$

where, if necessary, the convention "$0 \cdot \infty = 0$" is used. If f is a step function on $\mathbb{R}$ and μ is Lebesgue measure, then the expression above is a Riemann sum.

Clearly, it is not satisfactory to have a definition valid only for simple functions. The basic idea, used to extend the definition to a non-simple function f, is to find a sequence of simple functions f_n converging to f. Then, the integral of f can be defined as the limit of $\int f_n\,d\mu$. We thus need to find an appropriate class of functions, for which such converging sequences exist. For this cause, we make the following definition.

Definition A.2 (Measurable function). Let $(\Omega, \mathcal{F})$ be a measurable space and let f be a function from Ω to $\mathbb{R}$. Then, f is called measurable $\mathcal{F}$ (or simply measurable) if, for every $H \in \mathcal{B}(\mathbb{R})$, $f^{-1}(H) = \{\omega : f(\omega) \in H\} \in \mathcal{F}$, that is if the inverse image of H lies in $\mathcal{F}$.

For instance, all continuous functions $f : \mathbb{R}^d \to \mathbb{R}$ are measurable. As another special case, a function f taking on a countable number of distinct values $x_1, x_2, \dots$ on the sets $A_1, A_2, \dots$, is measurable if and only if $A_k \in \mathcal{F}$ for all k.

Now, if f is a nonnegative, measurable function, then there exists (see [Billingsley, 1995], Theorem 13.5) a non-decreasing sequence of nonnegative, simple, measurable functions f_n, such that

$$f(\omega) = \lim_{n\to\infty} f_n(\omega). \tag{A.8}$$

We can then define the integral of f w.r.t. μ according to,

$$\int f\,d\mu = \lim_{n\to\infty} \int f_n\,d\mu. \tag{A.9}$$

If we allow for infinite values, the limit on the right hand side will always exist since the sequence $\{f_n\}$ is non-decreasing. Furthermore, it can be shown that given any two sequences (with the properties mentioned above) converging to f, their integrals will have the same limits. Hence, to define the integral according to (A.9) makes sense, since the value of right hand side is independent of the exact sequence of functions f_n used. The way of constructing the integral as the limit of the integrals of a sequence of simple functions, resembles the construction of the Riemann integral as the limit of a sequence of Riemann sums.

Finally, if f is an arbitrary (not necessarily nonnegative), measurable function, we can divide it into its positive and negative parts,

$$f = f^+ - f^-, \tag{A.10}$$

where both f^+ and f^- are nonnegative. Then, the integral of f is taken as

$$\int f\,d\mu = \int f^+\,d\mu - \int f^-\,d\mu, \tag{A.11}$$

unless both terms on the right hand side are infinite. If this is the case, the integral of f w.r.t. μ is not defined. If both terms on the right hand side in the expression above are finite, or equivalently if $\int |f| \, d\mu$ is finite, the function f is said to be integrable w.r.t. μ. Note, however, that the integral of f may exist even if it is not integrable, since we allow for the values $\pm\infty$.

Remark A.1. Integration with respect to Lebesgue measure λ is often written using $d\omega$ rather than $\lambda(d\omega)$. That is, if $(\Omega, \mathcal{F})$ is Euclidian, f is a function on Ω and λ is Lebesgue measure, we write $\int f(\omega) \, d\omega \triangleq \int f(\omega)\lambda(d\omega) = \int f \, d\lambda$.

We can also define integration over a set A using the natural construction,

$$\int_A f \, d\mu \triangleq \int I_A f \, d\mu. \tag{A.12}$$

Assume that p is a nonnegative, measurable function. We can then define a measure ν on $\mathcal{F}$ according to,

$$\nu(A) = \int_A p \, d\mu. \tag{A.13}$$

If the measures ν and μ are related as above, the function p is said to be a *density* of ν w.r.t. μ. Integration w.r.t. ν can be done by substituting $d\nu$ with $p \, d\mu$. That is, for f integrable w.r.t. ν, or nonnegative, it holds that,

$$\int f \, d\nu = \int fp \, d\mu. \tag{A.14}$$

If ν is defined according to (A.13), then $\mu(A) = 0$ clearly implies $\nu(A) = 0$. This property is known as *absolute continuity* and is written $\nu \ll \mu$. An interesting fact, known as the Radon-Nikodym theorem, is that absolute continuity is in fact a sufficient condition for ν to have a density w.r.t. μ.

Theorem A.1 (Radon-Nikodym). *Let μ and ν be σ-finite measures on $\mathcal{F}$ and assume that $\nu \ll \mu$. Then there exists a nonnegative, measurable function p (a density) s.t. $\nu(A) = \int_A p \, d\mu$ for all $A \in \mathcal{F}$. This density is unique, except on a set of μ-measure zero, that is for any two such densities p and p', $\mu(\{\omega : p(\omega) \neq p'(\omega)\}) = 0$.*

Proof: See [Billingsley, 1995], Theorem 32.2. □

The density p is called the Radon-Nikodym derivative and is often written,

$$p(\omega) = \frac{d\nu}{d\mu}(\omega). \tag{A.15}$$

A.3 Probability

In this section we consider the specialisation of measure and integration theory to probability. We will now view the space Ω as a "sample space", meaning that its elements ω

are possible outcomes of some random experiment. For instance, $\Omega = \{1, 2, 3, 4, 5, 6\}$ can represent the possible outcomes from a roll of a die. A subset A of Ω is then called an *event* and we seek to answer the question; what is the probability of the event A, i.e. the probability that ω falls into A once the experiment is performed? Intuitively, this should be related to the "size" or measure of A.

More formally, let $(\Omega, \mathcal{F})$ be a measurable space and let P be a measure on $\mathcal{F}$. Then, P is called a *probability measure* or a *probability distribution* if $\mathrm{P}(\Omega) = 1$. For $A \in \mathcal{F}$, P assigns the probability $\mathrm{P}(A)$ to the event A. The triple $(\Omega, \mathcal{F}, \mathrm{P})$ is called a probability space. If $S \in \mathcal{F}$ and $\mathrm{P}(S) = 1$, then S is a *support* of P.

Example A.3: Dirac distribution

Consider the probability measure on $(\Omega, \mathcal{F})$ defined by,

$$\delta_{\omega'}(A) = \begin{cases} 1 & \text{if } \omega' \in A, \\ 0 & \text{otherwise.} \end{cases} \tag{A.16}$$

This is known as a point-mass distribution or a Dirac distribution, and it assigns all "probability mass" to the singleton ω'. Note the similarity between the point-mass distribution and the indicator function (A.5), the difference being whether we view the set A or the point ω' as the argument. Integration w.r.t. Dirac measure follows the well known rule

$$\int f(\omega)\delta_{\omega'}(d\omega) = f(\omega'). \tag{A.17}$$

In a probability context, a measurable function $X : \Omega \to \mathbb{R}$ is known as a *random variable*. If a random experiment is carried out and results in ω, then $X(\omega)$ is the value taken on by the random variable. The *distribution* or *law* of a random variable is a probability measure μ on $\mathcal{B}(\mathbb{R})$ defined by,

$$\mu(A) = \mathrm{P}(\{\omega : X(\omega) \in A\}), \qquad A \in \mathcal{B}(\mathbb{R}). \tag{A.18}$$

Hence, $\mu(A)$ is the probability that X falls in the set A. When dealing with probabilities as above, it is often convenient to introduce the simplified notation,

$$\mathrm{P}(X \in A) \triangleq \mathrm{P}(\{\omega : X(\omega) \in A\}). \tag{A.19}$$

Example A.4: Continuous random variable

Let μ be the distribution of a random variable X on some probability space $(\Omega, \mathcal{F}, \mathrm{P})$. Assume that μ is absolutely continuous w.r.t. Lebesgue measure. Then, by the Radon-Nikodym theorem we can write, for $A \in \mathcal{B}(\mathbb{R})$,

$$\mathrm{P}(X \in A) = \mu(A) = \int_A p(x)\, dx, \tag{A.20}$$

for some function p, known as a *probability density function* (PDF).

Example A.5: Dirac distribution (continued)

If the random variable X is the constant function $X(\omega) \equiv x'$, then the distribution of X is $\delta_{x'}$, i.e. the random variable X (naturally) assigns all "probability mass" to the singleton x'.

Remark A.2. Here we have, for the sake of illustration, defined random variables as taking values in $\mathbb{R}$. In the main part of this thesis, the term random variable is used more generally for $\mathcal{F}/\mathcal{X}$-measurable mappings from Ω to some arbitrary space X, with $(\mathsf{X}, \mathcal{X})$ being a measurable space. However, we then often consider functions of these random variables $f : \mathsf{X} \to \mathbb{R}$, meaning that the composition $f \circ X : \omega \mapsto f(X(\omega))$ is a random variable as defined in this section.

As a final concept of this appendix, we introduce the *expected value* of a random variable. If the random variable X on the probability space $(\Omega, \mathcal{F}, \mathrm{P})$ is integrable or nonnegative, we define its expectation as,

$$\mathrm{E}[X] \triangleq \int X(\omega) P(d\omega). \tag{A.21}$$

Also, if f is a real function of a real variable, we can compute the expectation of $f(X)$ as,

$$\mathrm{E}[f(X)] = \mu(f) = \int f(X(\omega)) P(d\omega) = \int f(x) \mu(dx), \tag{A.22a}$$

where the last equality follows from a change of variables. If, as in Example A.4, μ has a density p w.r.t. Lebesgue measure, then the expectation above can further be expressed as

$$\mathrm{E}[f(X)] = \int f(x) p(x)\, dx. \tag{A.22b}$$

B

The multivariate Gaussian distribution

In this appendix we shall give a few results on how the multivariate Gaussian distribution can be manipulated. The following theorems and corollary gives us all the tools needed to derive the expressions for the so called linear states z_t in this work. The statements are given without proofs, since the proofs are easily found in standard textbooks on the subject.

B.1 Partitioned Gaussian variables

We start by giving two results on partitioned Gaussian variables. Let us, without loss of generality, assume that the random vector X, its mean μ and its covariance Σ can be partitioned according to,

$$X = \begin{bmatrix} X_a \\ X_b \end{bmatrix}, \qquad \mu = \begin{bmatrix} \mu_a \\ \mu_b \end{bmatrix}, \qquad \Sigma = \begin{bmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{bmatrix} \tag{B.1}$$

where for reasons of symmetry $\Sigma_{ba} = \Sigma_{ab}^{\mathsf{T}}$. It is also useful to write down the partitioned information matrix,

$$\Lambda = \Sigma^{-1} = \begin{bmatrix} \Lambda_{aa} & \Lambda_{ab} \\ \Lambda_{ba} & \Lambda_{bb} \end{bmatrix}, \tag{B.2}$$

since this form will provide simpler expressions below. Note that, since the inverse of a symmetric matrix is also symmetric, we have $\Lambda_{ab} = \Lambda_{ba}^{\mathsf{T}}$.

We now provide two important and very useful theorems for partitioned Gaussian variables. These theorems concern the two operations marginalisation and conditioning.

Theorem B.1 (Marginalisation). *Let the random vector X be Gaussian distributed and partitioned according to* (B.1)*, then the marginal density $p(x_a)$ is given by,*

$$p(x_a) = \mathcal{N}\left(x_a\,;\,\mu_a, \Sigma_{aa}\right). \tag{B.3}$$

Theorem B.2 (Conditioning). *Let the random vector X be Gaussian distributed and partitioned according to* (B.1)*, then the conditional density $p(x_a \mid x_b)$ is given by,*

$$p(x_a \mid x_b) = \mathcal{N}\left(x_a\,;\,\mu_{a|b}, \Sigma_{a|b}\right), \tag{B.4a}$$

where

$$\mu_{a|b} = \mu_a + \Sigma_{ab}\Sigma_{bb}^{-1}(x_b - \mu_b), \tag{B.4b}$$

$$\Sigma_{a|b} = \Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba}, \tag{B.4c}$$

which using the information matrix can be written,

$$\mu_{a|b} = \mu_a - \Lambda_{aa}^{-1}\Lambda_{ab}(x_b - \mu_b), \tag{B.4d}$$

$$\Sigma_{a|b} = \Lambda_{aa}^{-1}. \tag{B.4e}$$

B.2 Affine transformations

In the previous section we dealt with partitioned Gaussian densities, and derived the expressions for the marginal and conditional densities expressed in terms of the parameters of the joint density. We shall now take a different starting point, namely that we are given the marginal density $p(x_a)$ and the conditional density $p(x_b \mid x_a)$ (affine in x_a) and derive expressions for the joint density $p(x_a, x_b)$, the marginal density $p(x_b)$ and the conditional density $p(x_a \mid x_b)$.

Theorem B.3 (Affine transformation). *Assume that X_a, as well as X_b conditioned on X_a, are Gaussian distributed with densities,*

$$p(x_a) = \mathcal{N}\left(x_a\,;\,\mu_a, \Sigma_a\right), \tag{B.5a}$$

$$p(x_b \mid x_a) = \mathcal{N}\left(x_b\,;\,Mx_a + b, \Sigma_{b|a}\right), \tag{B.5b}$$

where M is a matrix and b is a constant vector (of appropriate dimensions). The joint density of X_a and X_b is then given by,

$$p(x_a, x_b) = \mathcal{N}\left(\begin{bmatrix} x_a \\ x_b \end{bmatrix}\,;\,\begin{bmatrix} \mu_a \\ M\mu_a + b \end{bmatrix}, R\right), \tag{B.5c}$$

with

$$R = \begin{bmatrix} M^{\mathsf{T}}\Sigma_{b|a}^{-1}M + \Sigma_a^{-1} & -M^{\mathsf{T}}\Sigma_{b|a}^{-1} \\ -\Sigma_{b|a}^{-1}M & \Sigma_{b|a}^{-1} \end{bmatrix}^{-1} = \begin{bmatrix} \Sigma_a & \Sigma_a M^{\mathsf{T}} \\ M\Sigma_a & \Sigma_{b|a} + M\Sigma_a M^{\mathsf{T}} \end{bmatrix}. \tag{B.5d}$$

Combining the results in Theorem B.1, B.2 and B.3 we also get the following corollary.

Corollary B.1 (Affine transformation – marginal and conditional). *Assume that X_a, as well as X_b conditioned on X_a, are Gaussian distributed with densities,*

$$p(x_a) = \mathcal{N}\left(x_a\,;\,\mu_a, \Sigma_a\right), \tag{B.6a}$$

$$p(x_b \mid x_a) = \mathcal{N}\left(x_b\,;\,Mx_a + b, \Sigma_{b|a}\right), \tag{B.6b}$$

where M is a matrix and b is a constant vector (of appropriate dimensions). The marginal density of X_b is then given by,

$$p(x_b) = \mathcal{N}\left(x_b\,;\,\mu_b, \Sigma_b\right), \tag{B.6c}$$

with

$$\mu_b = M\mu_a + b, \tag{B.6d}$$

$$\Sigma_b = \Sigma_{b|a} + M\Sigma_a M^{\mathsf{T}}. \tag{B.6e}$$

The conditional density of X_a given X_b is

$$p(x_a \mid x_b) = \mathcal{N}\left(x_a\,;\,\mu_{a|b}, \Sigma_{a|b}\right), \tag{B.6f}$$

with

$$\mu_{a|b} = \Sigma_{a|b}\left(M^{\mathsf{T}}\Sigma_{b|a}^{-1}(x_b - b) + \Sigma_a^{-1}\mu_a\right) = \mu_a + \Sigma_a M^{\mathsf{T}}\Sigma_b^{-1}(x_b - b - M\mu_a), \tag{B.6g}$$

$$\Sigma_{a|b} = \left(\Sigma_a^{-1} + M^{\mathsf{T}}\Sigma_{b|a}^{-1}M\right)^{-1} = \Sigma_a - \Sigma_a M^{\mathsf{T}}\Sigma_b^{-1}M\Sigma_a. \tag{B.6h}$$

Bibliography

K. Abed-Meraim, W. Qiu, and Y. Hua. Blind system identification. *Proceedings of the IEEE*, 85(8):1310–1322, 1997. Cited on page 138.

J. Aldrich. R. A. Fisher and the making of maximum likelihhod 1912–1922. *Statistical Science*, 12(3):162–176, 1997. Cited on page 117.

M. L. Andrade Netto, L. Gimeno, and M. J. Mendes. A new spline algorithm for non-linear filtering of discrete time systems. In *Proceedings of the 7th Triennial World Congress*, pages 2123–2130, Helsinki, Finland, 1979. Cited on page 126.

C. Andrieu, A. Doucet, S. S. Singh, and V. B. Tadić. Particle methods for change detection, system identification, and contol. *Proceedings of the IEEE*, 92(3):423–438, March 2004. Cited on pages 115 and 117.

C. Andrieu, A. Doucet, and R. Holstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B*, 72(3):269–342, 2010. Cited on page 150.

M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002. Cited on page 34.

Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation.* John Wiley & Sons, 2001. Cited on page 66.

T. Bayes. An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London*, 53:370–418, 1764. Cited on page 120.

P. Billingsley. *Probability and measure.* Wiley series in probability and mathematical statistics. John Wiley & Sons, 3rd edition edition, 1995. Cited on pages 11, 151, 154, and 155.

D. Blackwell. Conditional expectation and unbiased sequential estimation. *Annals of Mathematical Statistics*, 18(1):105–110, 1947. Cited on page 4.

H. A. P. Blom. An efficient filter for abruptly changing systems. In *Proceedings of the*

23rd IEEE Conference on Decision and Control (CDC), Las Vegas, USA, December 1984. Cited on page 66.

H. A. P. Blom and Y. Bar-Shalom. The interacting multiple model algorithm for systems with Markovian switching coefficients. *IEEE Transactions on Automatic Control*, 33 (8):780–783, 1988. Cited on page 66.

J. P. Boyd. The uselessness of the fast Gauss transform for summing Gaussian radial basis function series. *Journal of Computational Physics*, 229:1311–1326, 2010. Cited on page 66.

M. Briers, A. Doucet, and S. Maskell. Smoothing algorithms for state-space models. *Annals of the Institute of Statistical Mathematics*, 62(1):61–89, February 2010. Cited on pages 87, 96, 99, and 108.

O. Cappé, E. Moulines, and T. Rydén. *Inference in Hidden Markov Models*. Springer, 2005. Cited on pages 17, 19, 21, 32, 33, 117, and 149.

O. Cappé, S. J. Godsill, and E. Moulines. An overview of existing methods and recent advances in sequential Monte Carlo. *Proceedings of the IEEE*, 95(5):899–924, 2007. Cited on page 34.

J. Carpenter, P. Clifford, and P. Fearnhead. Improved particle filter for nonlinear problems. *IEE Proceedings Radar, Sonar and Navigation*, 146(1):2–7, 1999. Cited on page 33.

N. Chopin. Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference. *The Annals of Statistics*, 32(6):2385–2411, 2004. Cited on pages 72, 75, and 79.

K. L. Chung. *A course in probability theory*. Academic Press, 3rd edition edition, 2001. Cited on page 151.

D. Crisan, P. Del Moral, and T. Lyons. Discrete filtering using branching and interacting particle systems. *Markov Processes and Related Fields*, 5(3):293–318, 1999. Cited on page 33.

P. Del Moral. *Feynman-Kac Formulae - Genealogical and Interacting Particle Systems with Applications*. Probability and its Applications. Springer, 2004. Cited on pages 72 and 73.

P. Del Moral and A. Guionnet. On the stability of interacting processes with applications to filtering and genetic algorithms. *Annales de l'Institut Henri Poincaré*, 37(2):155–194, 2001. Cited on page 74.

P. Del Moral and L. Miclo. Branching and interacting particle systems approximations of Feynman-Kac formulae with applications to non-linear filtering. In J. Azéma, M. Ledoux, M. Émery, and M. Yor, editors, *Séminaire de Probabilités XXXIV*, Lecture Notes in Mathematics, pages 1–145. Springer, 2000. Cited on pages 72, 73, and 74.

P. Del Moral, A. Doucet, and S. S. Singh. Forward smoothing using sequential Monte Carlo. Technical Report CUED/F-INFENG/TR 638, Cambridge University Engineering Department, Cambridge, UK, 2010. Cited on page 149.

A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977. Cited on pages 117 and 120.

D. G. T. Denison, C. G. Holmes, B. K. Mallick, and A. F. M. Smith. *Bayesian Methods for Nonlinear Classification and Regression.* John Wiley & Sons, 2002. Cited on page 121.

R. Douc and E. Moulines. Limit theorems for weighted samples with applications to sequential Monte Carlo. *The Annals of Statistics*, 36(5):2344–2376, 2008. Cited on pages 30, 33, 70, 71, 72, 73, 78, and 79.

R. Douc, E. Moulines, and J. Olsson. Optimality of the auxiliary particle filter. *Probability and Mathematical Statistics*, 29:1–28, 2009. Cited on page 73.

R. Douc, A. Garivier, E. Moulines, and J. Olsson. Sequential Monte Carlo smoothing for general state space hidden Markov models. *Submitted to Annals of Applied Probability*, 2010. Cited on pages 21, 73, 74, 96, 97, 99, 103, 110, and 132.

A. Doucet and A. Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. In D. Crisan and B. Rozovsky, editors, *The Oxford Handbook of Nonlinear Filtering.* Oxford University Press, 2011. Cited on page 34.

A. Doucet, N. de Freitas, K. Murphy, and S. Russell. Rao-Blackwellised particle filtering for dynamic Bayesian networks. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, pages 176–183, Stanford, USA, July 2000a. Cited on pages 75 and 147.

A. Doucet, S. J. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000b. Cited on pages 36, 42, 50, 75, and 90.

A. Doucet, S. J. Godsill, and M. West. Monte Carlo filtering and smoothing with application to time-varying spectral estimation. In *Proceedings of the 2000 IEEE International Conference on Computer Vision (ICCV)*, Istanbul , Turkey, June 2000c. Cited on page 91.

A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice.* Springer Verlag, New York, USA, 2001a. Cited on page 34.

A. Doucet, N. Gordon, and V. Krishnamurthy. Particle filters for state estimation of jump Markov linear systems. *IEEE Transactions on Signal Processing*, 49(3):613–624, March 2001b. Cited on page 40.

P. Fearnhead. Using random quasi-Monte-Carlo within particle filters, with application to financial time series. *Journal of Computational and Graphical Statistics*, 14(4):751–769, 2005. Cited on page 96.

P. Fearnhead, D. Wyncoll, and J. Tawn. A sequential smoothing algorithm with linear computational cost. *Biometrika*, 97(2):447–464, 2010. Cited on pages 87 and 96.

P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004. Cited on page 52.

R. A. Fisher. On an absolute criterion for fitting frequency curves. *Messenger of Mathematics*, 41:155–160, 1912. Cited on page 116.

R. A. Fisher. On the“probable error” of a coefficient of correlation deduced from a small sample. *Metron*, 1:3–32, 1921. Cited on page 116.

R. A. Fisher. On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London Series A*, 222:309–368, 1922. Cited on page 116.

W. Fong, S. J. Godsill, A. Doucet, and M. West. Monte Carlo smoothing with application to audio signal enhancement. *IEEE Transactions on Signal Processing*, 50(2):438–449, February 2002. Cited on pages 99, 108, and 109.

S. Gibson and B. Ninness. Robust maximum-likelihood estimation of multivariable dynamic systems. *Automatica*, 41(10):1667–1682, 2005. Cited on pages 117, 132, and 133.

W. R. Gilks and C. Berzuini. Following a moving target – Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 63(1):127–146, 2001. Cited on pages 40 and 72.

S. J. Godsill, A. Doucet, and M. West. Monte Carlo smoothing for nonlinear time series. *Journal of the American Statistical Association*, 99(465):156–168, March 2004. Cited on pages 91, 95, and 96.

R. B. Gopaluni. Identification of nonlinear processes with known model structure using missing observations. In *Proceedings of the 17th IFAC World Congress*, Seoul, South Korea, July 2008. Cited on pages 116 and 117.

N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 140(2):107 –113, April 1993. Cited on pages 32, 34, 35, 110, 123, and 126.

A. G. Gray and A. W. Moore. ’n-body’ problems in statistical learning. In *Proceedings of the 2000 Conference on Neural Information Processing Systems (NIPS)*, Denver, USA, November 2000. Cited on page 66.

A. G. Gray and A. W. Moore. Nonparametric density estimation: Toward computational tractability. In *Proceedings of the SIAM International Conference on Data Mining*, San Francisco, USA, May 2003. Cited on page 66.

L. Greengard and S. Strain. The fast Gauss transform. *SIAM Journal on Scientific and Statistical Computing*, 12(1):79–94, 1991. Cited on page 66.

L. Greengard and X. Sun. A new version of the fast Gauss transform. *Documenta Mathematica*, Extra Volume ICM III:575–584, 1998. Cited on page 66.

F. Gustafsson. *Statistical Sensor Fusion*. Studentlitteratur, 2010. Cited on page 32.

F. Gustafsson and P. Hriljac. Particle filters for prediction of chaos. In *Proceedings of the 13th IFAC Symposium on System Identification (SYSID)*, Rotterdam, The Netherlands, September 2003. Cited on page 124.

A. Hagenblad, L. Ljung, and A. Wills. Maximum likelihood identification of Wiener models. *Automatica*, 44(11):2697–2705, 2008. Cited on page 138.

A. Hald. On the history of maximum likelihood in relation to inverse probability and least squares. *Statistical Science*, 14(2):214–222, 1999. Cited on page 116.

R. van Handel. Uniform time average consistency of Monte Carlo particle filters. *Stochastic Processes and their Applications*, 119(11):3835–3861, 2009. Cited on page 74.

J. Handschin and D. Mayne. Monte Carlo techniques to estimate the conditional expectation in multi-stage non-linear filtering. *International Journal of Control*, 9(5):547–559, May 1969. Cited on pages 30 and 34.

G. Hendeby. *Performance and Implementation Aspects of Nonlinear Filtering*. PhD thesis, Linköping University, 2008. Cited on page 33.

J. D. Hol, T. B. Schön, and F. Gustafsson. On resampling algorithms for particle filters. In *Proceedings of the Nonlinear Statistical Signal Processing Workshop*, Cambridge, UK, September 2006. Cited on page 33.

X.-L. Hu, T. B. Schön, and L. Ljung. A basic convergence result for particle filtering. *IEEE Transactions on Signal Processing*, 56(4):1337–1348, 2008. Cited on page 73.

X.-L. Hu, T. B. Schön, and L. Ljung. A general convergence result for particle filtering. *IEEE Transactions on Signal Processing*, 2011. Cited on page 73.

I. W. Hunter and M. J. Korenberg. The identification of nonlinear biological systems: Wiener and Hammerstein cascade models. *Biological Cybernetics*, 55:135–144, 1986. Cited on page 137.

M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998. Cited on page 34.

Y. Jianjun, Z. Jianqiu, and M. Klaas. The marginal Rao-Blackwellized particle filter for mixed linear/nonlinear state space models. *Chinese Journal of Aeronautics*, 20:346–352, 2007. Cited on pages 58 and 66.

A. M. Johansen and A. Doucet. A note on auxiliary particle filters. *Statistics & Probability Letters*, 78(12):1498–1504, 2008. Cited on page 73.

N. Kantas, A. Doucet, S.S. Singh, and J.M. Maciejowski. An overview of sequential Monte Carlo methods for parameter estimation in general state-space models. In *Proceedings of the 15th IFAC Symposium on System Identification*, pages 774–785, Saint-Malo, France, July 2009. Cited on page 115.

R. Karlsson, T. B. Schön, and F. Gustafsson. Complexity analysis of the marginalized particle filter. *IEEE Transactions on Signal Processing*, 53(11):4408–4411, 2005. Cited on page 75.

G. Kitagawa. Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5(1):1–25, 1996. Cited on pages 32 and 34.

G. Kitagawa. A self-organizing state-space model. *Journal of the American Statistical Association*, 93(443):1203–1215, September 1998. Cited on page 123.

M. Klaas, N. de Freitas, and A. Doucet. Toward practical n^2 Monte Carlo: the marginal particle filter. In *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence (UAI)*, Edinburgh, UK, July 2005. Cited on pages 41, 42, 58, 60, and 66.

M. Klaas, M. Briers, N. de Freitas, A. Doucet, S. Maskell, and D. Lang. Fast particle smoothing: if I had a million particles. In *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, USA, June 2006. Cited on page 96.

S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951. Cited on page 119.

H. R. Künsch. Recursive Monte Carlo filters: algorithms and theoretical analysis. *The Annals of Statistics*, 33(5):1983–2021, 2005. Cited on pages 72 and 74.

P. S. Laplace. Mémoire sur la probabilité des causes par les évènemens. *Mémoires de mathématique et de physique presentés á l'Académie royale des sciences par divers savants & lus dans ses assemblées*, 6:621–656, 1774. Cited on page 121.

F. Le Gland and N. Oudjane. Stability and uniform approximation of nonlinear filters using the Hilbert metric and application to particle filters. *The Annals of Applied Probability*, 14(1):144–187, 2004. Cited on page 74.

E. L. Lehmann. *Theory of Point Estimation*. Probability and mathematical statistics. John Wiley & Sons, New York, USA, 1983. Cited on pages 4, 44, and 119.

F. Lindsten and T. B. Schön. Identification of mixed linear/nonlinear state-space models. In *Proceedings of the 49th IEEE Conference on Decision and Control (CDC)*, Atlanta, USA, December 2010. Cited on pages 99, 129, and 132.

F. Lindsten and T. B. Schön. Rao-Blackwellised particle smoothers for mixed linear/nonlinear state-space models. *Submitted to IEEE Transactions on Signal Processing*, 2011. Cited on pages 99, 109, 133, and 148.

F. Lindsten, P.-J. Nordlund, and F. Gustafsson. Conflict detection metrics for aircraft sense and avoid systems. In *Proceedings of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SafeProcess)*, Barcelona, Spain, July 2009. Not cited.

F. Lindsten, J. Callmer, H. Ohlsson, D. Törnqvist, T. B. Schön, and F. Gustafsson. Georeferencing for UAV navigation using environmental classification. In *Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, USA, May 2010. Cited on pages 6, 51, 52, and 53.

F. Lindsten, H. Ohlsson, and L. Ljung. Clustering using sum-of-norms regularization; with application to particle filter output computation. In *Proceedings of the 2011*

IEEE Workshop on Statistical Signal Processing (SSP) (accepted for publication), Nice, France, June 2011a. Not cited.

F. Lindsten, T. B. Schön, and J. Olsson. An explicit variance reduction expression for the Rao-Blackwellised particle filter. In *Proceedings of the 18th World Congress of the International Federation of Automatic Control (IFAC) (accepted for publication)*, Milan, Italy, August 2011b. Cited on page 74.

J. Liu and M. West. Combined parameter and state estimation in simulation-based filtering. In A. Doucet, N. de Freitas, and N. Gordon, editors, *Sequential Monte Carlo Methods in Practice*. Springer Verlag, 2001. Cited on page 123.

J. S. Liu and R. Chen. Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93(443):1032–1044, 1998. Cited on page 32.

J. S. Liu, R. Chen, and T. Logvinenko. A theoretical framework for sequential importance sampling and resampling. In A. Doucet, N. de Freitas, and N. Gordon, editors, *Sequential Monte Carlo Methods in Practice*. Springer Verlag, 2001. Cited on page 33.

L. Ljung. *System identification, Theory for the user*. System sciences series. Prentice Hall, Upper Saddle River, NJ, USA, second edition, 1999. Cited on page 115.

L. Ljung. Perspectives on system identification. In *Proceedings of the 17th IFAC World Congress*, pages 7172–7184, Seoul, South Korea, July 2008. Plenary lecture. Cited on page 115.

L. Ljung and T. Söderström. *Theory and Practice of Recursive Identification*. MIT Press, 1983. Cited on page 122.

L. Ljung and A. Vicino, editors. *Special Issue on System Identification*, volume 50 (10). *IEEE Transactions on Automatic Control*, 2005. Cited on page 115.

M. Loève. *Probability Theory I*. Springer, 4th edition edition, 1977. Cited on page 151.

M. Loève. *Probability Theory II*. Springer, 4th edition edition, 1978. Cited on page 151.

G. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. Whiley Series in Probability and Statistics. John Wiley & Sons, New York, USA, second edition, 2008. Cited on pages 117 and 119.

C. Musso, N. Oudjane, and F. Le Gland. Improving regularised particle filters. In A. Doucet, N. de Freitas, and N. Gordon, editors, *Sequential Monte Carlo Methods in Practice*. Springer Verlag, 2001. Cited on page 123.

B. Ninness and S. Henriksen. Bayesian system identification via Markov chain Monte Carlo techniques. *Automatica*, 46(1):40–51, 2010. Cited on page 121.

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Operations Research. Springer, New York, USA, 2000. Cited on page 132.

S. J. Norquay, A. Palazoglu, and J. A. Romagnoli. Application of Wiener model predictive control (WMPC) to a pH neutralization experiment. *IEEE Transactions on Control Systems Technology*, 7(4):437–445, 1999. Cited on page 137.

J. Olsson and T. Rydén. Metropolising forward particle filtering backward sampling and Rao-Blackwellisation of Metropolised particle smoothers. Technical Report 2010:15, Mathematical Sciences, Lund University, Lund, Sweden, 2010. Cited on page 150.

J. Olsson, R. Douc, O. Cappé, and E. Moulines. Sequential Monte Carlo smoothing with application to parameter estimation in nonlinear state-space models. *Bernoulli*, 14(1): 155–179, 2008. Cited on pages 116 and 117.

V. Peterka. Bayesian system identification. *Automatica*, 17(1):41–53, 1981. Cited on page 121.

M. K. Pitt and N. Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599, 1999. Cited on pages 60 and 76.

G. Poyiadjis, A. Doucet, and S. S. Singh. Maximum likelihhod parameter estimation in general state-space models using particle methods. In *Proceedings of the American Statistical Association*, Minneapolis, USA, August 2005. Cited on page 117.

G. Poyiadjis, A. Doucet, and S.S. Singh. Sequential monte carlo computation of the score and observed information matrix in state-space models with application to parameter estimation. Technical Report CUED/F-INFENG/TR 628, Cambridge University Engineering Department, Cambridge, UK, May 2009. Cited on pages 116 and 117.

C. R. Rao. Information and the accuracy attainable in the estimation of statistical parameters. *Bulletin of the Calcutta Mathematical Society*, 37:81–91, 1945. Cited on page 4.

H. E. Rauch, F. Tung, and C. T. Striebel. Maximum likelihood estimates of linear dynamic systems. *AIAA Journal*, 3(8):1445–1450, August 1965. Cited on pages 109 and 132.

H. Royden and P. Fitzpatrick. *Real analysis*. Pearson, 4th edition edition, 2010. Cited on page 151.

D. B. Rubin. A noniterative sampling/importance resampling alternative to the data augmentation algorithm for creating a few imputations when fractions of missing information are modest: The SIR algorithm. *Journal of the American Statistical Association*, 82(398):543–546, June 1987. Comment to Tanner and Wong: The Calculation of Posterior Distributions by Data Augmentation. Cited on pages 30, 31, and 32.

T. Schön and F. Gustafsson. Particle filters for system identification of state-space models linear in either parameters or states. In *Proceedings of the 13th IFAC Symposium on System Identification (SYSID)*, pages 1287–1292, Rotterdam, The Netherlands, September 2003. Cited on pages 123, 124, and 148.

T. Schön, F. Gustafsson, and P.-J. Nordlund. Marginalized particle filters for mixed linear/nonlinear state-space models. *IEEE Transactions on Signal Processing*, 53(7):2279–2289, July 2005. Cited on pages 42, 44, and 147.

T. B. Schön, A. Wills, and B. Ninness. System identification of nonlinear state-space models. *Automatica (to appear)*, 2011. Cited on pages 91, 116, 117, 126, 132, and 133.

I. Smal, K. Draegestein, N. Galjart, W. Niessen, and E. Meijering. Rao-Blackwellized marginal particle filtering for multiple object tracking in molecular bioimaging. In *Proceedings of the 20th International Conference on Information Processing in Medical Imaging*, Kerkrade, The Netherlands, July 2007. Cited on pages 58 and 66.

V. Smidl. Forgetting in marginalized particle filtering and its relation to forward smoothing. Technical Report LiTH-ISY-R-3009, Department of Electrical Engineering, Linköping University, Linköping, Sweden, May 2011. Cited on page 149.

G. A. Smith and A. J. Robinson. A comparison between the EM and subspace identification algorithms for time-invariant linear dynamical systems. Technical Report CUED/F-INFENG/TR 345, Cambridge University Engineering Department, Cambridge, UK, 2000. Cited on page 117.

T. Söderström and P. Stoica. *System Identification*. Prentice Hall, 1989. Cited on page 115.

P. Stavropoulos and M. Titterington. Improved particle filters and smoothing. In A. Doucet, N. de Freitas, and N. Gordon, editors, *Sequential Monte Carlo Methods in Practice*. Springer Verlag, 2001. Cited on page 123.

S. M. Stiegler. Thomas bayes's bayesian inference. *Journal of the Royal Statistical Society. Series A*, 145:250–258, 1982. Cited on page 120.

S. M. Stiegler. Laplace's 1774 memoir on inverse probability. *Statistical Science*, 1(3): 359–363, 1986. Cited on page 121.

D. Törnqvist, T. B. Schön, R. Karlsson, and F. Gustafsson. Particle filter SLAM with high dimensional vehicle model. *Journal of Intelligent and Robotic Systems*, 55(4):249–266, 2009. Cited on pages 52, 54, and 55.

A. V. Uglanov. Fubini's theorem for vector-valued measures. *Math. USSR Sbornik*, 69 (2):453–463, 1991. Cited on page 76.

P. Van Overschee and B. De Moor. *Subspace Identification for Linear Systems: Theory, Implementation, Applications*. Kluwer Academic Publishers, 1996. Cited on page 117.

D. Westwick and M. Verhaegen. Identifying MIMO Wiener systems using subspace model identification methods. *Signal Processing*, 52(2):235–258, 1996. Cited on page 138.

D. Whitley. A genetic algorithm tutorial. *Statistics and Computing*, 4:65–85, 1994. Cited on pages 32 and 33.

T. Wigren. Recursive prediction error identification using the nonlinear Wiener model. *Automatica*, 29(4):1011–1025, 1993. Cited on page 138.

A. Wills and L. Ljung. Wiener system identification using the maximum likelihood method. In F. Giri and E. W. Bai, editors, *Block-Oriented Nonlinear System Identification, Lecture Notes in Control and Information Sciences*. Springer, 2010. Cited on page 138.

A. Wills, T. B. Schön, L. Ljung, and B. Ninness. Blind identification of Wiener models. In *Proceedings of the 18th World Congress of the International Federation of Automatic Control (IFAC) (accepted for publication)*, Milan, Italy, August 2011. Cited on pages 118 and 138.

C. F. J. Wu. On the convergence properties of the EM algorithm. *The Annals of Statistics*, 11(1):95–103, 1983. Cited on page 119.

C. Yang, R. Duraiswami, N. A. Gumerov, and L. Davis. Improved fast Gauss transform and efficient kernel density estimation. In *Proceedings of the 9th IEEE International Conference on Computer Vision (ICCV)*, Nice, France, October 2003. Cited on page 66.

Licentiate Theses
Division of Automatic Control
Linköping University

P. Andersson: Adaptive Forgetting through Multiple Models and Adaptive Control of Car Dynamics. Thesis No. 15, 1983.

B. Wahlberg: On Model Simplification in System Identification. Thesis No. 47, 1985.

A. Isaksson: Identification of Time Varying Systems and Applications of System Identification to Signal Processing. Thesis No. 75, 1986.

G. Malmberg: A Study of Adaptive Control Missiles. Thesis No. 76, 1986.

S. Gunnarsson: On the Mean Square Error of Transfer Function Estimates with Applications to Control. Thesis No. 90, 1986.

M. Viberg: On the Adaptive Array Problem. Thesis No. 117, 1987.

K. Ståhl: On the Frequency Domain Analysis of Nonlinear Systems. Thesis No. 137, 1988.

A. Skeppstedt: Construction of Composite Models from Large Data-Sets. Thesis No. 149, 1988.

P. A. J. Nagy: MaMiS: A Programming Environment for Numeric/Symbolic Data Processing. Thesis No. 153, 1988.

K. Forsman: Applications of Constructive Algebra to Control Problems. Thesis No. 231, 1990.

I. Klein: Planning for a Class of Sequential Control Problems. Thesis No. 234, 1990.

F. Gustafsson: Optimal Segmentation of Linear Regression Parameters. Thesis No. 246, 1990.

H. Hjalmarsson: On Estimation of Model Quality in System Identification. Thesis No. 251, 1990.

S. Andersson: Sensor Array Processing; Application to Mobile Communication Systems and Dimension Reduction. Thesis No. 255, 1990.

K. Wang Chen: Observability and Invertibility of Nonlinear Systems: A Differential Algebraic Approach. Thesis No. 282, 1991.

J. Sjöberg: Regularization Issues in Neural Network Models of Dynamical Systems. Thesis No. 366, 1993.

P. Pucar: Segmentation of Laser Range Radar Images Using Hidden Markov Field Models. Thesis No. 403, 1993.

H. Fortell: Volterra and Algebraic Approaches to the Zero Dynamics. Thesis No. 438, 1994.

T. McKelvey: On State-Space Models in System Identification. Thesis No. 447, 1994.

T. Andersson: Concepts and Algorithms for Non-Linear System Identifiability. Thesis No. 448, 1994.

P. Lindskog: Algorithms and Tools for System Identification Using Prior Knowledge. Thesis No. 456, 1994.

J. Plantin: Algebraic Methods for Verification and Control of Discrete Event Dynamic Systems. Thesis No. 501, 1995.

J. Gunnarsson: On Modeling of Discrete Event Dynamic Systems, Using Symbolic Algebraic Methods. Thesis No. 502, 1995.

A. Ericsson: Fast Power Control to Counteract Rayleigh Fading in Cellular Radio Systems. Thesis No. 527, 1995.

M. Jirstrand: Algebraic Methods for Modeling and Design in Control. Thesis No. 540, 1996.

K. Edström: Simulation of Mode Switching Systems Using Switched Bond Graphs. Thesis No. 586, 1996.

J. Palmqvist: On Integrity Monitoring of Integrated Navigation Systems. Thesis No. 600, 1997.

A. Stenman: Just-in-Time Models with Applications to Dynamical Systems. Thesis No. 601, 1997.

M. Andersson: Experimental Design and Updating of Finite Element Models. Thesis No. 611, 1997.

U. Forssell: Properties and Usage of Closed-Loop Identification Methods. Thesis No. 641, 1997.

M. Larsson: On Modeling and Diagnosis of Discrete Event Dynamic systems. Thesis No. 648, 1997.

N. Bergman: Bayesian Inference in Terrain Navigation. Thesis No. 649, 1997.

V. Einarsson: On Verification of Switched Systems Using Abstractions. Thesis No. 705, 1998.

J. Blom, F. Gunnarsson: Power Control in Cellular Radio Systems. Thesis No. 706, 1998.

P. Spångéus: Hybrid Control using LP and LMI methods – Some Applications. Thesis No. 724, 1998.

M. Norrlöf: On Analysis and Implementation of Iterative Learning Control. Thesis No. 727, 1998.

A. Hagenblad: Aspects of the Identification of Wiener Models. Thesis No. 793, 1999.

F. Tjärnström: Quality Estimation of Approximate Models. Thesis No. 810, 2000.

C. Carlsson: Vehicle Size and Orientation Estimation Using Geometric Fitting. Thesis No. 840, 2000.

J. Löfberg: Linear Model Predictive Control: Stability and Robustness. Thesis No. 866, 2001.

O. Härkegård: Flight Control Design Using Backstepping. Thesis No. 875, 2001.

J. Elbornsson: Equalization of Distortion in A/D Converters. Thesis No. 883, 2001.

J. Roll: Robust Verification and Identification of Piecewise Affine Systems. Thesis No. 899, 2001.

I. Lind: Regressor Selection in System Identification using ANOVA. Thesis No. 921, 2001.

R. Karlsson: Simulation Based Methods for Target Tracking. Thesis No. 930, 2002.

P.-J. Nordlund: Sequential Monte Carlo Filters and Integrated Navigation. Thesis No. 945, 2002.

M. Östring: Identification, Diagnosis, and Control of a Flexible Robot Arm. Thesis No. 948, 2002.

C. Olsson: Active Engine Vibration Isolation using Feedback Control. Thesis No. 968, 2002.

J. Jansson: Tracking and Decision Making for Automotive Collision Avoidance. Thesis No. 965, 2002.

N. Persson: Event Based Sampling with Application to Spectral Estimation. Thesis No. 981, 2002.

D. Lindgren: Subspace Selection Techniques for Classification Problems. Thesis No. 995, 2002.

E. Geijer Lundin: Uplink Load in CDMA Cellular Systems. Thesis No. 1045, 2003.

M. Enqvist: Some Results on Linear Models of Nonlinear Systems. Thesis No. 1046, 2003.

T. Schön: On Computational Methods for Nonlinear Estimation. Thesis No. 1047, 2003.

F. Gunnarsson: On Modeling and Control of Network Queue Dynamics. Thesis No. 1048, 2003.

S. Björklund: A Survey and Comparison of Time-Delay Estimation Methods in Linear Systems. Thesis No. 1061, 2003.

M. Gerdin: Parameter Estimation in Linear Descriptor Systems. Thesis No. 1085, 2004.

A. Eidehall: An Automotive Lane Guidance System. Thesis No. 1122, 2004.

E. Wernholt: On Multivariable and Nonlinear Identification of Industrial Robots. Thesis No. 1131, 2004.

J. Gillberg: Methods for Frequency Domain Estimation of Continuous-Time Models. Thesis No. 1133, 2004.

G. Hendeby: Fundamental Estimation and Detection Limits in Linear Non-Gaussian Systems. Thesis No. 1199, 2005.

D. Axehill: Applications of Integer Quadratic Programming in Control and Communication. Thesis No. 1218, 2005.

J. Sjöberg: Some Results On Optimal Control for Nonlinear Descriptor Systems. Thesis No. 1227, 2006.

D. Törnqvist: Statistical Fault Detection with Applications to IMU Disturbances. Thesis No. 1258, 2006.

H. Tidefelt: Structural algorithms and perturbations in differential-algebraic equations. Thesis No. 1318, 2007.

S. Moberg: On Modeling and Control of Flexible Manipulators. Thesis No. 1336, 2007.

J. Wallén: On Kinematic Modelling and Iterative Learning Control of Industrial Robots. Thesis No. 1343, 2008.

J. Harju Johansson: A Structure Utilizing Inexact Primal-Dual Interior-Point Method for Analysis of Linear Differential Inclusions. Thesis No. 1367, 2008.

J. D. Hol: Pose Estimation and Calibration Algorithms for Vision and Inertial Sensors. Thesis No. 1370, 2008.

H. Ohlsson: Regression on Manifolds with Implications for System Identification. Thesis No. 1382, 2008.

D. Ankelhed: On low order controller synthesis using rational constraints. Thesis No. 1398, 2009.

P. Skoglar: Planning Methods for Aerial Exploration and Ground Target Tracking. Thesis No. 1420, 2009.

C. Lundquist: Automotive Sensor Fusion for Situation Awareness. Thesis No. 1422, 2009.

C. Lyzell: Initialization Methods for System Identification. Thesis No. 1426, 2009.

R. Falkeborn: Structure exploitation in semidefinite programming for control. Thesis No. 1430, 2010.

D. Petersson: Nonlinear Optimization Approaches to $\mathcal{H}_2$-Norm Based LPV Modelling and Control. Thesis No. 1453, 2010.

Z. Sjanic: Navigation and SAR Auto-focusing in a Sensor Fusion Framework. Thesis No. 1464, 2011.

K. Granström: Loop detection and extended target tracking using laser data. Thesis No. 1465, 2011.

J. Callmer: Topics in Localization and Mapping. Thesis No. 1489, 2011.