Department of Neuroscience and Biomedical Engineering /
Department of Computer Science

Particle and Sigma-Point Methods for State and Parameter Estimation in Nonlinear Dynamic Systems

Juho Kokkala

Aalto University

DOCTORAL
DISSERTATIONS

Aalto University publication series
DOCTORAL DISSERTATIONS 30/2016

Particle and Sigma-Point Methods for State and Parameter Estimation in Nonlinear Dynamic Systems

Juho Kokkala

A doctoral dissertation completed for the degree of Doctor of Science (Technology) to be defended, with the permission of the Aalto University School of Science, at a public examination held at the lecture hall F239a of the school on 4 March 2016 at 12.

Aalto University
School of Science
Department of Neuroscience and Biomedical Engineering /
Department of Computer Science

Supervising professor

Professor Jouko Lampinen, Aalto University, Finland

Thesis advisor

Associate Professor Simo Särkkä, Aalto University, Finland

Preliminary examiners

Dr., Docent Dario Gasbarra, University of Helsinki, Finland
Associate Professor Gustaf Hendeby, Linköping University, Sweden

Opponent

Professor Fredrik Gustafsson, Linköping University, Sweden

Aalto University publication series
DOCTORAL DISSERTATIONS 30/2016

ISBN 978-952-60-6663-9 (printed)
ISBN 978-952-60-6664-6 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)
http://urn.fi/URN:ISBN:978-952-60-6664-6

Unigrafia Oy
Helsinki 2016

Finland

NORDIC ECOLABEL
441 697
Printed matter

Aalto University

Abstract

Aalto University, P.O. Box 11000, FI-00076 Aalto www.aalto.fi

Author
Juho Kokkala

Name of the doctoral dissertation
Particle and Sigma-Point Methods for State and Parameter Estimation in Nonlinear Dynamic Systems

Publisher School of Science

Unit Department of Neuroscience and Biomedical Engineering / Department of Computer Science

Series Aalto University publication series DOCTORAL DISSERTATIONS 30/2016

Field of research Computational Engineering

Manuscript submitted 17 November 2015 **Date of the defence** 4 March 2016

Permission to publish granted (date) 15 January 2016 **Language** English

□ **Monograph** ☒ **Article dissertation** □ **Essay dissertation**

Abstract

State-space models of dynamic systems are used to model phenomena, which evolve over time, and are observed only through noisy or incomplete measurements. An example is tracking a moving target based on sensor measurements. A state-space model consists of a probabilistic specification of the evolution of a latent state, conditional on the previous values, and a probabilistic specification of how the observations depend on the latent states. Typically, we are interested in the filtering and smoothing problems, and parameter estimation. In filtering, one computes the probability distribution of the current latent state, taking into account all observations obtained so far. Smoothing refers to updating the probability distributions of the latent states on the previous times based on new observations obtained after those times. Both filtering and smoothing problems are analytically intractable except in some special cases.

In this thesis, sigma-point and particle based filtering and smoothing methods are used. Sigma-point filters and smoothers are based on approximating the filters and smoothers by Gaussian density approximations and then further approximating certain integrals by numerical cubature rules. Particle filters and smoothers in turn use random samples to approximate the probability distributions and, under certain conditions, converge to exact filters and smoothers when the number of samples tends to infinity.

The research topics of this thesis are to develop new importance distributions for particle filters, to develop methods for static parameter estimation, and an application to animal population size estimation.

A split-Gaussian importance distribution is proposed for particle filters and compared to alternatives. In addition, in a certain time-varying Poisson regression model, it is shown that a split-Gaussian modification to a Gaussian approximation based importance distribution guarantees convergence of the particle filter when the pure Gaussian approximation does not.

A parameter estimation method for multiple target tracking models is developed based on combining recently proposed particle Markov chain Monte Carlo algorithms with the Rao-Blackwellized Monte Carlo data association algorithm. This method allows for joint estimation of the target movements and the parameters of the models, as well as the number of targets. The method is applied to estimating bear population size based on a database of field signs.

Additionally, parameter estimation in nonlinear systems with additive Gaussian noise is discussed using both direct likelihood maximization and expectation-maximization (EM), and both particle and sigma-point algorithms for the smoothing problem arising in EM.

Keywords Bayesian filtering and smoothing, state-space models, dynamic systems, particle filters, sigma-point filters

ISBN (printed) 978-952-60-6663-9 **ISBN (pdf)** 978-952-60-6664-6

ISSN-L 1799-4934 **ISSN (printed)** 1799-4934 **ISSN (pdf)** 1799-4942

Location of publisher Helsinki **Location of printing** Helsinki **Year** 2016

Pages 126 **urn** http://urn.fi/URN:ISBN:978-952-60-6664-6

A! Aalto-yliopisto

Tiivistelmä

Aalto-yliopisto, PL 11000, 00076 Aalto www.aalto.fi

Tekijä
Juho Kokkala

Väitöskirjan nimi
Partikkeli- ja sigmapistemenetelmiä epälineaaristen dynaamisten systeemien tila- ja parametriestimointiin

Julkaisija Perustieteiden korkeakoulu

Yksikkö Neurotieteen ja lääketieteellisen tekniikan laitos / Tietotekniikan laitos

Sarja Aalto University publication series DOCTORAL DISSERTATIONS 30/2016

Tutkimusala Laskennallinen tekniikka

Käsikirjoituksen pvm 17.11.2015 **Väitöspäivä** 04.03.2016

Julkaisuluvan myöntämispäivä 15.01.2016 **Kieli** Englanti

□ **Monografia** ☒ **Artikkeliväitöskirja** □ **Esseeväitöskirja**

Tiivistelmä

Dynaamisten systeemien tila-avaruusmalleilla mallinnetaan ajassa eteneviä ilmiöitä, joista saadaan epätarkkoja havaintoja. Näitä malleja voidaan soveltaa esimerkiksi liikkuvan kohteen seuraamiseen. Tila-avaruusmalli kuvaa todennäköisyysjakaumina miten systeemin tila riippuu aiemmasta tilasta ja miten kunkin ajanhetken mittaus riippuu tilasta. Tila-avaruusmallin suodatustehtävässä lasketaan tilan todennäköisyysjakauma kullakin ajanhetkellä ottaen huomioon kaikki siihen mennessä saadut mittaukset. Silotustehtävässä puolestaan nämä jakaumat päivitetään ottamaan huomioon myös myöhemmät mittaukset. Suodatus- ja silotustehtävää ei voi ratkaista analyyttisesti suljetussa muodossa kuin tietyissä erikoistapauksissa.

Tässä väitöskirjassa käytetään sigmapiste- ja partikkelipohjaisia suodatus- ja silotusmenetelmiä. Sigmapistemenetelmät perustuvat suotimen ja silottimen todennäköisyysjakaumien approksimointiin gaussisilla jakaumilla ja eräiden integraalien approksimointiin numeerisilla integrointisäännöillä. Partikkelisuotimissa ja -silottimissa puolestaan käytetään satunnaisotoksia. Tietyillä ehdoilla partikkelisuotimet ja -silottimet suppenevat suodatus- ja silotustehtävän täsmällisiin ratkaisuihin, kun otoskoko lähestyy ääretöntä.

Tämän väitöskirjan tutkimuskohteet ovat importanssijakaumien kehittäminen partikkelisuodinalgoritmeihin, tila-avaruusmallien parametrien estimointimenetelmien kehittäminen sekä sovellus eläinpopulaation koon arviointiin.

Väitöskirjassa ehdotetaan split-gaussista importanssijakaumaa partikkelisuotimiin ja verrataan sitä muihin importanssijakaumiin. Aikariippuville Poisson-regressiomalleille osoitetaan, että eräs split-gaussinen importanssijakauma takaa partikkelisuotimen suppenemisen - toisin kuin gaussiseen approksimaatioon perustuva importanssijakauma.

Väitöskirjassa kehitetään monen kohteen seurantatehtäviin parametriestimointialgoritmi, joka perustuu partikkelipohjaisten Markov-ketju- Monte Carlo -algoritmien ja rao-blackwellisoidun Monte Carlo -data-assosiaation yhdistämiseen. Tällä menetelmällä voidaan samanaikaisesti estimoida mallin parametrit ja kohteiden liikkuminen sekä kohteiden lukumäärä. Menetelmää sovelletaan karhupopulaation koon arviointiin kenttähavaintojen perusteella.

Väitöskirjassa tutkitaan parametriestimointia myös additiivis-gaussisissa epälineaarisissa dynaamisissa systeemeissä. Työssä vertaillaan suoraa suurimman uskottavuuden menetelmää ja expectation-maximization -algoritmia käyttäen partikkeli- ja sigmapiste-menetelmiä.

Avainsanat bayesilainen suodatus ja silotus, tila-avaruusmallit, dynaamiset systeemit, partikkelisuotimet, sigmapistesuotimet

ISBN (painettu) 978-952-60-6663-9 **ISBN (pdf)** 978-952-60-6664-6

ISSN-L 1799-4934 **ISSN (painettu)** 1799-4934 **ISSN (pdf)** 1799-4942

Julkaisupaikka Helsinki **Painopaikka** Helsinki **Vuosi** 2016

Sivumäärä 126 **urn** http://urn.fi/URN:ISBN:978-952-60-6664-6

Preface

This work was performed at the Department of Neuroscience and Biomedical Engineering (before organizational changes, BECS). Since May 2015, I was also affiliated with the Department of Computer Science. Both departments are part of the School of Science, Aalto University, Finland. I acknowledge funding from the Academy of Finland, from the Emil Aaltonen foundation, and from the doctoral program Brain & Mind. I acknowledge the computational resources provided by the Aalto Science-IT project.

In the beginning of my journey, I was instructed by Dr. Pekka Marttinen and by Prof. Aki Vehtari. I thank Pekka and Aki for their guidance, for letting me to pursue my interests and for the decision of employing me as a doctoral student. Aki deserves thanks also for leading the Bayes group. In 2013, the journey took an interesting new direction when I started working with the advisor of this thesis, Prof. Simo Särkkä. I thank Simo for introducing me to the world of Bayesian filtering and smoothing, for all research topics and ideas he gave me, for the collaboration in our joint articles, and most importantly for being patient and understanding and encouraging me to keep going during difficulties.

I thank Prof. Jouko Lampinen for supervising my doctoral studies, and for acting some time as the head of the BECS department and now as the head of the CS department. I thank the preliminary examiners Dr. Dario Gasbarra and Prof. Gustaf Hendeby for reading my thesis and providing valuable comments, and Prof. Fredrik Gustafsson for agreeing to act as my opponent.

I thank my coauthor Arno Solin, not only for the collaboration in our joint articles and proofreading the overview part of this thesis, but also for his support and interest in my well-being, as well as for his unofficial role as the social activator of the Bayes group. I thank also all the not-yet-mentioned members/visitors of the Bayes group, including but not limited to Dr. Jouni Hartikainen, Dr. Pasi Jylänki, Toni Karvonen, Olli-Pekka Koistinen, Dr. Isambi Mbalawata, Dr. Tommi Mononen, Dr. Mari Myllymäki, Janne Ojanen, Dr. Tomi Peltola, Juho Piironen, Dr. Jaakko Riihimäki, Tuomas Sivula, and Ville Tolvanen. For providing data for Publication II, I thank Samuli Heikkinen and Dr. Mika Kurkilahti from the Natural Resources Center Finland. All administration/support personnel from NBE/BECS deserve huge thanks for their help and patience. To name a few of these people: Dr. Mikko Hakala, Mari Kaarni, Eeva Lampinen, and Henrika Wilkman. I thank Prof. Samuel Kaski and his part of the Probabilistic Machine Learning group for welcoming us into the group.

Before starting my doctoral studies, I took my first steps within the research world in the Systems Analysis Laboratory. I want to thank the colleagues from SAL, most importantly Dr. Jirka Poropudas, Prof. Kai Virtanen, and Prof. Raimo P. Hämäläinen. To go further back in time, I want to thank everyone involved with the Päivölä Mathematics programme, especially Kullervo Nieminen, Merikki Lappi, and Dr. Esa Lappi.

For various unspecified reasons more, less, or not at all related to my research, I thank the following people and entities: Johannes Haataja, Aleksi Hartikainen, Hannu Hartikainen, Katri Hartikainen, Anu Hirvonen, Juha-Matti Hirvonen, Olli-Pekka Kahilakoski, Joona Karjalainen, Atle Kivelä, Jaakko Kulhia, Sami Liedes, Teemu Liikala, Dr. Jaakko Luttinen, Eric Malmi, Astrid Pietilä, Pruikkosen kommuuni, Pässi, Pirkko Turunen, Timo Turunen, and everyone I accidentally omitted.

I thank my parents for raising and supporting me. I thank my brothers for all our shared experiences. I thank my wife Essi for her love and also for her patience with the demands of this thesis.

Finally, I thank God.

Otaniemi, February 5, 2016,

Juho Kokkala

Contents

List of Publications

This thesis consists of an overview and of the following publications which are referred to in the text by their Roman numerals.

I Juho Kokkala, Arno Solin, and Simo Särkkä. Expectation Maximization Based Parameter Estimation by Sigma-Point and Particle Smoothing. In *The 17th International Conference on Information Fusion (FUSION)*, 8 pages, Salamanca, Spain, July 2014.

II Juho Kokkala and Simo Särkkä. Combining Particle MCMC with Rao-Blackwellized Monte Carlo Data Association for Parameter Estimation in Multiple Target Tracking. *Digital Signal Processing*, Volume 47, Pages 84-95, December 2015.

III Juho Kokkala and Simo Särkkä. On the (Non-)Convergence of Particle Filters with Gaussian Importance Distributions. In *Proceedings of the 17th IFAC Symposium on System Identification (SYSID)*, (IFAC-PapersOnline, Volume 48, Issue 28), Pages 793–798. Beijing, China, October 2015.

IV Juho Kokkala and Simo Särkkä. Split-Gaussian Particle Filter. In *Proceedings of 23rd European Signal Processing Conference (EUSIPCO)*, Nice, France, Pages 484–488. August 2015.

V Juho Kokkala, Arno Solin, and Simo Särkkä. Sigma-Point Filtering and Smoothing Based Parameter Estimation in Nonlinear Dynamic Systems. Accepted for publication in *Journal of Advances in Information*

Fusion, arXiv preprint arXiv:1504.06173. 14 pages. 2016.

Author's Contribution

Publication I: "Expectation Maximization Based Parameter Estimation by Sigma-Point and Particle Smoothing"

Kokkala and Solin wrote the article jointly with constructive comments by Särkkä. In writing, Kokkala had the main responsibility in the parts describing particle methods while Solin had the main responsibility in the parts describing sigma-point methods. Kokkala implemented the particle smoother parts of the numeric experiments.

Publication II: "Combining Particle MCMC with Rao-Blackwellized Monte Carlo Data Association for Parameter Estimation in Multiple Target Tracking"

Kokkala wrote the first version of the manuscript, which was jointly revised by both authors. Kokkala worked out the details of the algorithms, implemented them by extending and modifying code from an existing toolbox, and designed the experiments. The general idea was originally suggested by Särkkä.

Publication III: "On the (Non-)Convergence of Particle Filters with Gaussian Importance Distributions"

Kokkala wrote the first version of the manuscript, which was then revised by both authors jointly. Kokkala suggested the specific idea of using split-Gaussian importance distributions for the Poisson regression model and developed the proofs (partially based on notes by Särkkä, and the proof of Theorem 5 is by Särkkä). Kokkala designed and implemented the numeric experiment. The general idea of motivating the split-Gaussian importance distributions by convergence considerations was originally proposed by Särkkä.

Publication IV: "Split-Gaussian Particle Filter"

Kokkala wrote the first version of the manuscript, which was then revised by both authors jointly. Kokkala designed and implemented the experiments. Särkkä suggested the idea of using a split-Gaussian importance distribution in particle filters.

Publication V: "Sigma-Point Filtering and Smoothing Based Parameter Estimation in Nonlinear Dynamic Systems"

This is a significant extension of Publication I. Kokkala designed and implemented the experiments jointly with Solin. All three authors participated in writing.

1. Introduction

Dynamic systems are used to model phenomena evolving over time. Mathematical models of dynamic systems may be used to solve the problems of estimation and control. Estimation refers to deducing what can be known about the state of a dynamic system based on noisy and incomplete observations collected over time. Control refers to the problem of applying inputs to the system to steer it to behave in a desired manner. Estimation in dynamic systems is covered in various textbooks, (see, e.g., Jazwinski, 1970; Gelb, 1974; Anderson and Moore, 1979; Bar-Shalom et al., 2004; Särkkä, 2013).

In this thesis, we focus on methods for estimation, and do not consider control. However, the final motivation of gaining more information about a system may still be to enable a decision-maker to make better decisions, even though in a particular piece of work the connection to decision-making may be temporarily ignored. For introductions to control theory, see, e.g., Kirk (1970); Stengel (1986); Glad and Ljung (2000).

The models and methods considered in this thesis are formulated in the probabilistic, or Bayesian, state-space modeling framework (see, e.g., Särkkä, 2013; Cappé et al., 2005; Jazwinski, 1970). In Bayesian inference (see, e.g., Gelman et al., 2013), all uncertainty about the system of interest is modeled as randomness, such that incomplete knowledge is modeled using probability distributions. The probabilistic state-space models used for dynamic systems specify the evolution of a latent, unobserved, state of the system as probability distributions of the state conditional on the state at a previous time. Furthermore, the model specifies how the observations depend on the state by defining probability distributions of the observations conditional on the state. See Figure 1.1 for a graphical representation.

The filtering problem refers to finding the probability distribution of

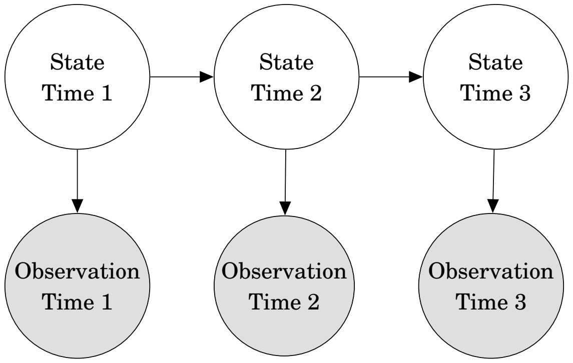

Figure 1.1. Graphical representation of a state-space model. The state at a particular time step influences the state at the next time step as well as the observation at the current time step.

the state conditional on the observations obtained so far. Filtering algorithms find these distributions in a sequential manner, that is, construct the probability distribution of the state by updating the previous probability distribution of the state, taking into account the possible evolution of the state as well as the new observation. Figure 1.2 illustrates a filtering problem where the state is the unknown location of an object and the observations are sensor measurements.

While the Bayesian framework provides a logically coherent way to formulate the problems of estimating the state and identifying the parameters, a significant drawback is that the required conditional probability distributions are not usually easy to compute. Analytic solutions that are exact and possible to evaluate quickly exist only in some cases. For example, the filtering problem of a linear dynamic system with Gaussian noise can be solved using the Kalman filter (Kalman, 1960; Ho and Lee, 1964). In the more general case, one has to resort to computational methods that only approximate the exact probability distributions that would be implied by the model.

In this thesis, we use nonlinear Kalman filters (or sigma-point filters) (e.g. Ito and Xiong, 2000) and particle filters (e.g. Doucet et al., 2001; Ristic et al., 2004). Nonlinear Kalman filtering is based on using certain simplifying, but not completely accurate, assumptions while computing the solution of the filtering problem. These assumptions enable the use of similar equations as in the Kalman filter.

When more accurate solutions are desired at the expense of increased computational load, or when the model is such that the nonlinear Kalman filtering framework is not suitable, particle filters may be used instead.

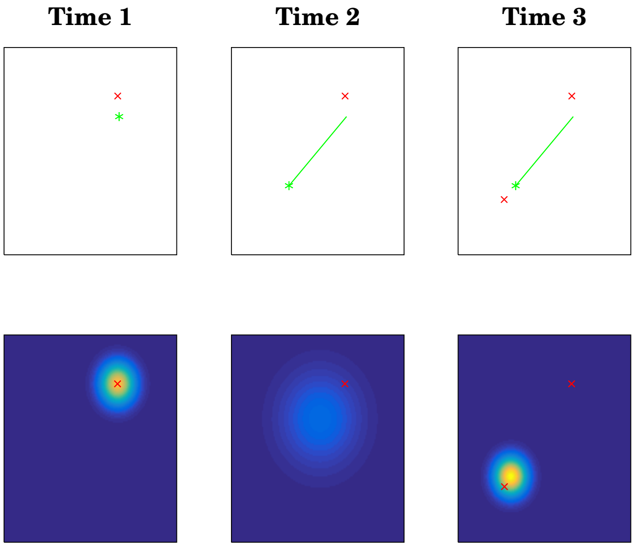

Figure 1.2. Tracking a target based on noisy measurements (the red crosses). The upper row shows the true evolution of the target (the green star), while the lower row shows probability distributions based on the measurements (yellow indicates a region of high probability). At the time of the first measurement (leftmost column), the target is estimated to be close to the location. As time passes, the location becomes more uncertain, i.e., the probability distribution becomes wider. After the second measurement, the new probability distribution for the location is a compromise between the probability distribution before the measurement (middle column) and the measurement.

Particle filters do not make the simplifying assumptions that the nonlinear Kalman filters do. Instead, they are based on representing the filtering distributions as a finite set of weighted sample values known as particles. Each particle contains a possible state value as well as a weight describing how likely that particular value is. Particle filters are Monte Carlo methods, that is, the samples are drawn randomly. The particle filter algorithms are designed so that, under certain technical conditions, using more particles will likely produce a more accurate solution and when the number of particles approaches infinity, the solutions approach the exact filtering distributions. However, the drawback is that typically many particles and thus a high computational load is required to obtain good approximations, compared to the relatively fast sigma-point filters.

Besides the state estimation problem, the dynamic system itself may be unknown, in which case system identification is called for. As a specific example, the state-space model may depend on parameters whose values are not known. In this case, identifying the system means estimating the parameters. In the Bayesian framework, the parameters, too, are modeled as random variables with probability distributions, and parameter estimation is understood to mean finding a conditional probability distribution for the parameters conditional on the observations. For parameter estimation, we consider both the Bayesian approach and maximum likelihood estimation.

The methodology development research topics of this thesis are i) static parameter estimation in state-space models and ii) developing importance distributions for particle filters. For static parameter estimation, we develop a new Bayesian parameter estimation algorithm for multiple target tracking problems, that is based on combining the recently proposed particle Markov chain Monte Carlo methods (Andrieu et al., 2010) with a multiple target tracking algorithm (Särkkä et al., 2007). In addition, we consider maximum-likelihood based point estimation using expectation–maximization (EM, see Dempster et al., 1977) in connection with sigma-point and particle filters. The particle filter algorithm development is concentrated on proposing the so-called split-Gaussian importance distribution (Geweke, 1989), and showing how it is motivated by convergence considerations in a certain state-space model. Besides methodology development, an additional third research topic is animal population size estimation, namely, estimation of the bear population in Finland based on recorded observations. This animal population size estimation problem serves as a motivating example for the parameter estimation work.

The remainder of the overview part of this thesis is structured as follows. The relevant prerequisite material is reviewed in Chapters 2–3. Of these, Chapter 2 presents the ideas of Bayesian inference and state-space models, and discusses Kalman filtering and nonlinear Kalman filtering for linear and nonlinear systems with additive Gaussian noise. Chapter 3 in turn presents a review of particle filtering and particle filter based smoothing and parameter estimation methods. The contributions of this thesis are summarized in Chapter 4. Finally, Chapter 5 contains discussion.

2. Bayesian and Nonlinear Kalman Filtering and Smoothing

2.1 Bayesian Statistics

In the Bayesian approach to statistical modeling (see, e.g., Gelman et al., 2013), all uncertainty that is taken into account is modeled as randomness using probability theory. Updating beliefs about any unknown phenomenon in light of new information is then based on computing conditional probabilities. The reader is referred to textbooks such as Jacod and Protter (2003) for an introduction to probability theory, which is not presented in this thesis. In this thesis, we use similar shorthand notation as in Särkkä (2013). Bold letters are used to refer to both (possibly vector-valued) random variables and their realizations. $p(\mathbf{x})$ refers to the probability density function of the random variable $\mathbf{x}$ (or to the value of the density at $\mathbf{x}$). Similarly, $p(\mathbf{x} \mid \mathbf{y})$ refers to the probability density function of the conditional distribution of $\mathbf{x}$ conditional on $\mathbf{y}$. Note that the notation is overloaded such that, for example, $p(\mathbf{x})$ and $p(\mathbf{y})$ are two different functions. $\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$ refers to the density of the multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ evaluated at argument $\mathbf{x}$. Integrals such as $\int p(\mathbf{x}) \, \mathrm{d}\mathbf{x}$ refer to integrating over the support of $\mathbf{x}$. Furthermore, if $\mathbf{x}$ is a discrete random variable, $p(\mathbf{x})$ refers to the probability mass function and the corresponding integral notation refers to summation.

Gelman et al. (2013) describe the Bayesian data analysis framework as consisting of three steps:

1. Setting up a probability model.

2. Conditioning on observed data.

3. Evaluating the model.

The first step refers to defining a joint probability distribution over all observable (both observed and not yet observed) as well as all unobservable quantities that are taken into account in the analysis. In the second step, based on the rules of probability theory, the conditional probability distribution of the unobserved quantities of interest is computed conditional on the observed data. The third step is outside the formal Bayesian reasoning framework and consists of checking that the results of the first two steps are reasonable. In the viewpoint of Gelman et al. (2013), the process may be iterated so that the model definition (the first step above) is changed if the model evaluation step shows that the model is not satisfactory.

In a typical Bayesian analysis, one is interested in estimating some parameters $\boldsymbol{\theta}$ given data y that somehow depend on the parameters. The full probability model, that is, the joint distribution $p(\mathrm{y}, \boldsymbol{\theta})$, is often defined by defining the observation model $p(\mathrm{y} \mid \boldsymbol{\theta})$, that is, the probability distribution for the data if the parameter would be fixed, and a prior distribution $p(\boldsymbol{\theta})$ for the parameter. The prior distribution $p(\boldsymbol{\theta})$ may be selected so that it reflects existing information about the parameter. In the second step above, the conditioning on observed data then updates the analyst's information about the parameter by computing $p(\boldsymbol{\theta} \mid \mathrm{y})$ using Bayes' rule,

$$p(\boldsymbol{\theta} \mid \mathrm{y}) = \frac{p(\boldsymbol{\theta})\, p(\mathrm{y} \mid \boldsymbol{\theta})}{\int p(\boldsymbol{\theta})\, p(\mathrm{y} \mid \boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta}}. \tag{2.1}$$

Conditional independence is a useful concept for building Bayesian models. For example, the previous setting may be extended to consider future observations $\tilde{\mathrm{y}}$. One may assume that the observed data y and the future observation $\tilde{\mathrm{y}}$ are conditionally independent conditional on the parameter, so that the full probability model factorizes as $p(\boldsymbol{\theta}, \mathrm{y}, \tilde{\mathrm{y}}) = p(\boldsymbol{\theta})\, p(\mathrm{y} \mid \boldsymbol{\theta})\, p(\tilde{\mathrm{y}} \mid \boldsymbol{\theta})$. Then, predicting the future observations based on the observed data, which in the Bayesian framework means computing the posterior predictive distribution $p(\tilde{\mathrm{y}} \mid \mathrm{y})$, may be performed as

$$p(\tilde{\mathrm{y}} \mid \mathrm{y}) = \int p(\tilde{\mathrm{y}} \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta} \mid \mathrm{y})\, \mathrm{d}\boldsymbol{\theta}, \tag{2.2}$$

that is, by computing a weighted average of the observation models, weighting by the posterior distribution of the parameter.

Unfortunately, the integrals appearing in computing the posterior distributions and posterior predictive distributions are not available in closed form, except in some specific probability models. Therefore, much of the practical work in Bayesian statistics is related to developing computational methods for approximating the distributions. A commonly used family of computational methods is Markov chain Monte Carlo (MCMC), which refers to constructing Markov chains that move in the space of the unobserved quantity of interest, so that the chain produces a sequence of dependent samples from the posterior distribution. Eventually, after the chain has been run long enough, the sequence of visited states may be used as an approximation to the posterior distribution.

2.2 State-Space Models

State-space models (Särkkä, 2013; Cappé et al., 2005) are probabilistic models for situations where a sequence of observed variables is interpreted as noisy or indirect observations of an unobserved phenomenon evolving over time. For example, recorded sightings and field observations of bears depend on the true unknown locations of the bears. The unobserved phenomenon is modeled as a sequence of random variables, called the latent state.

A state-space model consists of a specification of the probabilistic evolution of the latent state, denoted here by $\mathbf{x}_k \in \mathcal{X}$ for time step $k \in \mathbb{N}$, as well as a probability model for the observations (also called measurements), denoted here by $\mathbf{y}_k \in \mathcal{Y}$ for time step $k \in \mathbb{N}$. The state is usually assumed to contain all relevant information carried from the past. Thus, the state sequence has the Markov property $p(\mathbf{x}_k \mid \mathbf{x}_{0:k-1}) = p(\mathbf{x}_k \mid \mathbf{x}_{k-1})$. The observations are assumed to be conditionally independent given the latent state sequence. Furthermore, given the latent states, each observation depends only on the latent state of the time of the observation. Then, the joint probability distribution over T time steps and the initial state at time 0 factorizes as

$$p(\mathbf{y}_{1:T}, \mathbf{x}_{0:T}) = p(\mathbf{x}_0) \prod_{k=1}^{T} p(\mathbf{x}_k \mid \mathbf{x}_{k-1}) \prod_{k=1}^{T} p(\mathbf{y}_k \mid \mathbf{x}_k), \tag{2.3}$$

where $p(\mathbf{x}_0)$ is the prior distribution of the initial value of the latent state, the densities $p(\mathbf{x}_k \mid \mathbf{x}_{k-1})$ are called the dynamic model, and the densities $p(\mathbf{y}_k \mid \mathbf{x}_k)$ are called the observation model (or the measurement model).

We may also denote the model as

$$\begin{aligned}
&\mathbf{x}_0 \sim p(\mathbf{x}_0),\\
&\mathbf{x}_k \sim p(\mathbf{x}_k \mid \mathbf{x}_{k-1}),\ k = 1, 2, \ldots,\\
&\mathbf{y}_k \sim p(\mathbf{y}_k \mid \mathbf{x}_k),\ k = 1, 2, \ldots,
\end{aligned} \tag{2.4}$$

where the conditional independence properties are left implicit.

Typically, one is interested in estimating the current value of the latent state given the observations obtained so far. Following the Bayesian viewpoint, this means computing the probability distribution $p(\mathbf{x}_k \mid \mathbf{y}_{1:k})$, known as the filtering distribution. Furthermore, in contrast to general Bayesian modeling, the interest lies in computing these distributions sequentially, that is, by somehow using the already computed filtering distribution $p(\mathbf{x}_k \mid \mathbf{y}_{1:k})$ to obtain the next filtering distribution $p(\mathbf{x}_{k+1} \mid \mathbf{y}_{1:k+1})$ in a computationally efficient way. This problem is known as the filtering problem.

In principle, the filtering problem can be solved by iteratively using the following equations, known as the filtering equations:

$$p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}) = \int p(\mathbf{x}_{k-1} \mid \mathbf{y}_{1:k-1})\, p(\mathbf{x}_k \mid \mathbf{x}_{k-1})\, \mathrm{d}\mathbf{x}_{k-1}, \tag{2.5}$$

$$p(\mathbf{x}_k \mid \mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k \mid \mathbf{x}_k)\, p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1})}{\int p(\mathbf{y}_k \mid \mathbf{x}_k)\, p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1})\, \mathrm{d}\mathbf{x}_k}, \tag{2.6}$$

where the first equation marginalizes $p(\mathbf{x}_{k-1}, \mathbf{x}_k \mid \mathbf{y}_{1:k-1})$ over $\mathbf{x}_{k-1}$ to obtain the so-called prediction distribution. The second equation is the update step, which computes the filtering distribution from the prediction distribution via Bayes' rule. For some state-space models, these equations are analytically tractable and the filtering problem is thus solved by iteratively applying the filtering equations. However, in the general case the integrals involved in these equations are not analytically tractable, wherefore numerical approximation methods, such as the particle and sigma-point filters discussed in this thesis, are employed.

2.3 Kalman Filter

Kalman (1960) considered optimal state estimation in dynamic systems of the following form:

$$\begin{aligned}
\mathbf{x}_k &= \mathbf{A}_{k-1}\, \mathbf{x}_{k-1} + \mathbf{q}_{k-1},\\
\mathbf{y}_k &= \mathbf{H}_k\, \mathbf{x}_k + \mathbf{r}_k,
\end{aligned} \tag{2.7}$$

where $\mathbf{x}_k \in \mathbb{R}^{d_x}$, $\mathbf{y}_k \in \mathbb{R}^{d_y}$, $\mathbf{A}_{k-1} \in \mathbb{R}^{d_x \times d_x}$ and $\mathbf{H}_k \in \mathbb{R}^{d_y \times d_x}$ are known matrices, $\mathbf{q}_{k-1}$ is a zero-mean process noise, and $\mathbf{r}_k$ is a zero-mean measurement noise. The noises are assumed to be independent with known covariances. In this thesis, we use the Bayesian interpretation of the Kalman filter (Ho and Lee, 1964), which is based on assuming that the noises are Gaussian and that the initial state $\mathbf{x}_0$ has a Gaussian prior distribution. With these Gaussianity assumptions, the linear dynamic system (Eq. 2.7) corresponds to the general probabilistic state-space model (Eq. 2.4) with

$$\begin{aligned} \mathbf{x}_0 &\sim \mathcal{N}(\mathbf{x}_0 \mid \mathbf{m}_0,\ \mathbf{P}_0), \\ \mathbf{x}_k &\sim \mathcal{N}(\mathbf{x}_k \mid \mathbf{A}_{k-1}\,\mathbf{x}_{k-1},\ \mathbf{Q}_{k-1}), \\ \mathbf{y}_k &\sim \mathcal{N}(\mathbf{y}_k \mid \mathbf{H}_k\,\mathbf{x}_k,\ \mathbf{R}_k). \end{aligned} \tag{2.8}$$

For this model, the prediction and update equations can be computed in closed form. For the prediction equation (Eq. 2.5), assuming

$$p(\mathbf{x}_{k-1} \mid \mathbf{y}_{1:k-1}) = \mathcal{N}(\mathbf{x}_{k-1} \mid \mathbf{m}_{k-1}, \mathbf{P}_{k-1}) \tag{2.9}$$

implies

$$p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}) = \mathcal{N}(\mathbf{x}_k \mid \mathbf{m}_k^-, \mathbf{P}_k^-), \tag{2.10}$$

where the mean and covariance of the prediction distribution are obtained by the following matrix equations:

$$\begin{aligned} \mathbf{m}_k^- &= \mathbf{A}_{k-1}\,\mathbf{m}_{k-1}, \\ \mathbf{P}_k^- &= \mathbf{A}_{k-1}\,\mathbf{P}_{k-1}\,\mathbf{A}_{k-1}^\top + \mathbf{Q}_{k-1}. \end{aligned} \tag{2.11}$$

Furthermore, for the update equation (2.6), assuming $p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}) = \mathcal{N}(\mathbf{x}_k \mid \mathbf{m}_k^-, \mathbf{P}_k^-)$ implies

$$p(\mathbf{x}_k \mid \mathbf{y}_{1:k}) = \mathcal{N}(\mathbf{x}_k \mid \mathbf{m}_k, \mathbf{P}_k), \tag{2.12}$$

where the mean and covariance of the filtering distribution are obtained by the following matrix equations:

$$\begin{aligned} \mathbf{S}_k &= \mathbf{H}_k\,\mathbf{P}_k^-\,\mathbf{H}_k^\top + \mathbf{R}_k, \\ \mathbf{K}_k &= \mathbf{P}_k^-\,\mathbf{H}_k^\top\,\mathbf{S}_k^{-1}, \\ \mathbf{m}_k &= \mathbf{m}_k^- + \mathbf{K}_k\,(\mathbf{y}_k - \mathbf{H}_k\,\mathbf{m}_k^-), \\ \mathbf{P}_k &= \mathbf{P}_k^- - \mathbf{K}_k\,\mathbf{S}_k\,\mathbf{K}_k^\top. \end{aligned} \tag{2.13}$$

The Kalman filter algorithm then consists of iterating the prediction step (Eq. 2.11) and the update step (Eq. 2.13).

2.4 Nonlinear Kalman Filters

Consider the following nonlinear dynamic system with additive Gaussian noise:

$$\begin{aligned}\mathbf{x}_k &= \mathbf{f}_{k-1}(\mathbf{x}_{k-1}) + \mathbf{q}_{k-1},\\ \mathbf{y}_k &= \mathbf{h}_k(\mathbf{y}_{k-1}) + \mathbf{r}_k,\end{aligned} \tag{2.14}$$

where $\mathbf{x}_k \in \mathbb{R}^{d_x}, \mathbf{y}_k \in \mathbb{R}^{d_y}$, and $\mathbf{f}_{k-1} : \mathbb{R}^{d_x} \mapsto \mathbb{R}^{d_x}$ and $\mathbf{h}_k : \mathbb{R}^{d_x} \mapsto \mathbb{R}^{d_y}$ are some known, possibly nonlinear, functions. The noises $\mathbf{q}_{k-1}$ and $\mathbf{r}_k$ are assumed zero-mean Gaussian and independent as in the model defined by Equation (2.8). In contrast to the linear dynamic system (Eq. 2.7), the filtering problem for this nonlinear dynamic system is generally intractable. However, one may attempt to obtain approximate solutions to the filtering problem by approximating the system by a linear system such that Kalman filter type recursions may be applied. One such algorithm is the extended Kalman filter (EKF, Jazwinski, 1970), where at each prediction step one forms a local linearization of $\mathbf{f}_{k-1}$ around the mean $\mathbf{m}_{k-1}$ and at each update step one forms a local linearization of $\mathbf{h}_k$ around the predicted mean $\mathbf{m}_k^-$.

In the remainder of this section, we focus on assumed density Gaussian filters (Ito and Xiong, 2000). The assumed density Gaussian filter is derived by approximating the required distributions by Gaussian distributions with matching means and covariances in the prediction and update step, respectively. In practice, only the mean and covariance of the target distribution then needs to be computed at each step of the algorithm. However, the integrals required to evaluate the means and covariances in the assumed density Gaussian filter are generally intractable, too. A tractable approximative filter algorithm is then obtained by replacing the integrals by cubature rules, which leads to the so-called sigma-point filters. Alternatively, the sigma-point filters may be derived, for example, based on so-called weighted statistical linear regression (van der Merwe and Wan, 2003). However, in this thesis we focus only on the aforementioned cubature interpretation.

2.4.1 Assumed Gaussian Density Filtering

Ito and Xiong (2000) (see also Kushner, 1967) developed the assumed density Gaussian filter as follows. The prediction step (Eq. 2.5) is approximated by assuming that $p(\mathbf{x}_{k-1} \mid \mathbf{y}_{1:k-1}) \approx \mathcal{N}(\mathbf{x}_{k-1} \mid \mathbf{m}_{k-1}, \mathbf{P}_{k-1})$. The prediction equation is then replaced by

$$\begin{aligned} &p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}) \\ &\quad \approx \int \mathcal{N}(\mathbf{x}_k \mid \mathbf{f}_{k-1}(\mathbf{x}_{k-1}), \mathbf{Q}_{k-1})\, \mathcal{N}(\mathbf{x}_{k-1} \mid \mathbf{m}_{k-1}, \mathbf{P}_{k-1})\, \mathrm{d}\mathbf{x}_{k-1}. \end{aligned} \tag{2.15}$$

The mean and covariance of this prediction distribution are then obtained by

$$\begin{aligned} \mathbf{m}_k^- &= \int \mathbf{f}_{k-1}(\mathbf{x}_{k-1})\, \mathcal{N}(\mathbf{x}_{k-1} \mid \mathbf{m}_{k-1}, \mathbf{P}_{k-1})\, \mathrm{d}\mathbf{x}_{k-1}, \\ \mathbf{P}_k^- &= \mathbf{Q}_{k-1} \\ &\quad + \int (\mathbf{f}_{k-1}(\mathbf{x}_{k-1}) - \mathbf{m}_k^-)\, (\mathbf{f}_{k-1}(\mathbf{x}_{k-1}) - \mathbf{m}_k^-)^\mathsf{T}\, \mathcal{N}(\mathbf{x}_{k-1} \mid \mathbf{m}_{k-1}, \mathbf{P}_{k-1})\, \mathrm{d}\mathbf{x}_{k-1}. \end{aligned} \tag{2.16}$$

In the update step, a Gaussian approximation is used for the joint distribution $p(\mathbf{x}_k, \mathbf{h}_k(\mathbf{x}_k) \mid \mathbf{y}_{1:k-1})$. Assuming the prediction distribution is $\mathcal{N}(\mathbf{x}_k \mid \mathbf{m}_k^-, \mathbf{P}_k^-)$, the expectation of $\mathbf{h}_k(\mathbf{x}_k)$ is

$$\boldsymbol{\mu}_k = \mathbb{E}[\mathbf{h}_k(\mathbf{x}_k)] = \int \mathbf{h}_k(\mathbf{x}_k)\, \mathcal{N}(\mathbf{x}_k \mid \mathbf{m}_k^-, \mathbf{P}_k^-)\, \mathrm{d}\mathbf{x}_k, \tag{2.17}$$

and the covariance of $\mathbf{h}_k(\mathbf{x}_k)$ is

$$\begin{aligned} &\mathbb{E}[(\mathbf{h}_k(\mathbf{x}_k) - \boldsymbol{\mu}_k)\, (\mathbf{h}_k(\mathbf{x}_k) - \boldsymbol{\mu}_k)^\mathsf{T}] \\ &\quad = \int (\mathbf{h}_k(\mathbf{x}_k) - \boldsymbol{\mu}_k)\, (\mathbf{h}_k(\mathbf{x}_k) - \boldsymbol{\mu}_k)^\mathsf{T}\, \mathcal{N}(\mathbf{x}_k \mid \mathbf{m}_k^-, \mathbf{P}_k^-)\, \mathrm{d}\mathbf{x}_k, \end{aligned} \tag{2.18}$$

and the cross-covariance of $\mathbf{h}_k(\mathbf{x}_k)$ and $\mathbf{x}_k$ is

$$\begin{aligned} &\mathbb{E}[(\mathbf{x}_k - \mathbf{m}_k^-)(\mathbf{h}_k(\mathbf{x}_k) - \boldsymbol{\mu}_k)^\mathsf{T}] \\ &\quad = \int (\mathbf{x}_k - \mathbf{m}_k^-)\, (\mathbf{h}_k(\mathbf{x}_k) - \boldsymbol{\mu}_k)^\mathsf{T}\, \mathcal{N}(\mathbf{x}_k \mid \mathbf{m}_k^-, \mathbf{P}_k^-)\, \mathrm{d}\mathbf{x}_k. \end{aligned} \tag{2.19}$$

Now, as $(\mathbf{x}_k, \mathbf{h}_k(\mathbf{x}_k))$ is assumed jointly Gaussian and the measurement noise is independent additive Gaussian, the Kalman filter update equations (Eq. 2.13) apply as follows:

$$\begin{aligned} \boldsymbol{\mu}_k &= \mathbb{E}[\mathbf{h}_k(\mathbf{x}_k)], \\ \mathbf{S}_k &= \mathbb{E}[(\mathbf{h}_k(\mathbf{x}_k) - \boldsymbol{\mu}_k)\, (\mathbf{h}_k(\mathbf{x}_k) - \boldsymbol{\mu}_k)^\mathsf{T}] + \mathbf{R}_k, \\ \mathbf{C}_k &= \mathbb{E}[(\mathbf{x}_k - \mathbf{m}_k^-)\, (\mathbf{h}_k(\mathbf{x}_k) - \boldsymbol{\mu}_k)^\mathsf{T}], \\ \mathbf{K}_k &= \mathbf{C}_k\, \mathbf{S}_k^{-1}, \\ \mathbf{m}_k &= \mathbf{m}_k^- + \mathbf{K}_k\, (\mathbf{y}_k - \boldsymbol{\mu}_k), \\ \mathbf{P}_k &= \mathbf{P}_k^- - \mathbf{K}_k\, \mathbf{S}_k\, \mathbf{K}_k^\mathsf{T}, \end{aligned} \tag{2.20}$$

where the expectations are evaluated using Equations (2.17, 2.18, 2.19). The assumed density Gaussian filter algorithm then operates by iterating the prediction step (Eq. 2.16) and the update step (Eq. 2.20).

2.4.2 Cubature Integration by Sigma-Points

The assumed density Gaussian filter discussed in the previous section requires the evaluation of various integrals of functions of the state x weighted by a Gaussian density (Eqs. 2.16, 2.17, 2.18, 2.19). In the general case, these integrals cannot be evaluated in closed form. Practically viable filtering algorithms may be obtained by replacing these integrals by numeric approximations. Especially, the sigma-point filters are based on using cubature (numeric integration) rules of the form

$$\int \mathbf{g}(\mathbf{x})\, \mathcal{N}(\mathbf{x} \mid \mathbf{m}, \mathbf{P})\, \mathrm{d}\mathbf{x} \approx \sum_i w_i\, \mathbf{x}_i, \tag{2.21}$$

where the $\mathbf{x}_i$ are a finite collection of so-called sigma-points and the w_i are corresponding scalar weights. By change of variables $\mathbf{x} = \mathbf{m} + \mathbf{L}\,\mathbf{z}$ where $\mathbf{L}$ is such that $\mathbf{P} = \mathbf{L}\,\mathbf{L}^\mathsf{T}$, for example based on the Cholesky decomposition of $\mathbf{P}$, the integral can be expressed as weighted by the standard normal distribution as

$$\int \mathbf{g}(\mathbf{m} + \mathbf{L}\,\mathbf{z})\, \mathcal{N}(\mathbf{z} \mid \mathbf{0}, \mathbf{I})\, \mathrm{d}\mathbf{z}. \tag{2.22}$$

This change of variables motivates selecting the sigma-points $\mathbf{x}_i$ based on transforming a set of unit sigma-points $\boldsymbol{\xi}_i$ by

$$\mathbf{x}_i = \mathbf{m} + \mathbf{L}\,\boldsymbol{\xi}_i. \tag{2.23}$$

Different sigma-point methods are then obtained by different selections of the weights and the unit sigma-points for approximating the integrals (Eq. 2.22). In this thesis, we consider the Gauss–Hermite rules as well as the unscented transform and higher order symmetric spherical-radical rules. These are reviewed briefly in the following.

For approximating one-dimensional Gaussian integrals

$$\int g(z)\, \mathcal{N}(0, 1)\, \mathrm{d}z \approx \sum_i w_i\, \xi_i, \tag{2.24}$$

the Gauss-Hermite quadrature rule selects the weights and points so that when m points are used, the cubature is exact when g is a polynomial of at most degree $2m - 1$ (Wu et al., 2006, and references therein). The multidimensional Gauss-Hermite rule is then defined by letting the set of sigma-points be the Cartesian product of the dimension-wise one-dimensional Gauss-Hermite points. With m one-dimensional points there are then m^{d_x} sigma-points in total. The weights are set equal to the products of the corresponding one-dimensional weights.

The unscented transform (Julier et al., 1995, 2000) is a third order method in the sense that it gives the correct value for the integral when the integrand is a polynomial of degree at most three. The scaled version of the unscented transform proposed by Julier (2002) results by setting

$$\begin{aligned} \boldsymbol{\xi}_0 &= \mathbf{0}, \\ \boldsymbol{\xi}_i &= \sqrt{\lambda + d_x}\,\mathbf{e}_i, i = 1, \dots, d_x, \\ \boldsymbol{\xi}_i &= -\sqrt{\lambda + d_x}\,\mathbf{e}_{i-d_x}, i = n+1, \dots, 2\,d_x, \end{aligned} \tag{2.25}$$

where $\mathbf{e}_i$ is the ith unit vector and $\lambda = \alpha^2(d_x + \kappa) - d_x$ where α, κ are parameters of the method. The weights are

$$\begin{aligned} w_0 &= \frac{\lambda}{d_x + \lambda}, \\ w_i &= \frac{1}{2\,(d_x + \lambda)}, i = 1, \dots, 2\,d_x. \end{aligned} \tag{2.26}$$

For evaluation of the covariance terms, w_0 is replaced by $\frac{\lambda}{d_x+\lambda} + 1 - \alpha^2 - \beta$, where β is an additional parameter of the method.

As Solin (2010) pointed out, the symmetric spherical-radial cubature rule (Arasaratnam and Haykin, 2009) is obtained as a special case of the unscented transform by setting $\alpha = \pm 1,\ \beta = 0,\ \kappa = 0$ so that

$$\begin{aligned} w_i &= \frac{1}{2\,d_x},\ i = 1, \dots, 2\,d_x, \\ \boldsymbol{\xi}_i &= \sqrt{d_x}\,\mathbf{e}_i,\ i = 1, \dots, d_x, \\ \boldsymbol{\xi}_i &= -\sqrt{d_x}\,\mathbf{e}_{i-d_x},\ i = d_x + 1, \dots, 2\,d_x. \end{aligned} \tag{2.27}$$

Higher-order symmetric cubature rules that are exact for polynomials upto degree $p = 5, 7, \dots$ can be constructed, for example, following McNamee and Stenger (1967).

2.5 Smoothing

2.5.1 Fixed-Interval Smoothing

Besides the filtering problem, one may also be interested in updating the estimates of the previous states in light of new observations, known as the smoothing problem. In this thesis, we concentrate on the fixed-interval smoothing problem, that is, finding the marginal distribution of the state at each time step $k = 0, \dots, T$ conditional on all observations upto time T, that is, $p(\mathbf{x}_k \mid \mathbf{y}_{1:T})$ for $k = 0, \dots, T$.

As pointed out by Kitagawa (1987), the marginal smoothing distributions of the model defined in Equation (2.4) can be obtained from the filtering results based on the following equation

$$p(\mathbf{x}_k \mid \mathbf{y}_{1:T}) = p(\mathbf{x}_k \mid \mathbf{y}_{1:k}) \int \frac{p(\mathbf{x}_{k+1} \mid \mathbf{x}_k)\, p(\mathbf{x}_{k+1} \mid \mathbf{y}_{1:T})}{p(\mathbf{x}_{k+1} \mid \mathbf{y}_{1:k})} \, \mathrm{d}\mathbf{x}_{k+1}. \tag{2.28}$$

On the right hand side, $p(\mathbf{x}_{k+1} \mid \mathbf{x}_k)$ is the dynamic model density, $p(\mathbf{x}_k \mid \mathbf{y}_{1:k})$ and $p(\mathbf{x}_{k+1} \mid \mathbf{y}_{1:k})$ are the filtering and prediction densities computed during the filtering recursion, and $p(\mathbf{x}_{k+1} \mid \mathbf{y}_{1:T})$ is the marginal smoothing distribution of $\mathbf{x}_{k+1}$. Thus, the smoothing distributions may be computed using a forward-backward algorithm where first the filtering distributions are computed iteratively for times $k = 1, \ldots, T$ and then the smoothing distributions are computed in a backward iteration ($k = T-1, \ldots, 0$) using Equation (2.28).

However, similarly to the filtering problem, the integral in Equation (2.28) is intractable in the general case. In the linear-Gaussian case (Eq. 2.8), closed form expressions exist and thus the smoothing distributions can be obtained based on the Kalman filter results using the so-called Rauch–Tung–Striebel smoother (Rauch et al., 1965). In the remainder of this section, we present the Rauch–Tung–Striebel smoother equations in Subsection 2.5.2 and discuss sigma-point based approximative smoothing for nonlinear systems with additive Gaussian noise in Subsection 2.5.3.

2.5.2 Rauch–Tung–Striebel Smoothing

Consider the linear-Gaussian state-space model (Eq. 2.8). As discussed in Section 2.3, we have $p(\mathbf{x}_k \mid \mathbf{y}_{1:k}) = \mathcal{N}(\mathbf{x}_k \mid \mathbf{m}_k, \mathbf{P}_k)$ and $p(\mathbf{x}_{k+1} \mid \mathbf{y}_{1:k}) = \mathcal{N}(\mathbf{x}_{k+1} \mid \mathbf{m}_{k+1}^-, \mathbf{P}_{k+1}^-)$, where the means $(\mathbf{m}, \mathbf{m}^-)$ and covariances $(\mathbf{P}, \mathbf{P}^-)$ may be computed by the Kalman filter. If the smoothing distribution of $\mathbf{x}_{k+1}$ is Gaussian,

$$p(\mathbf{x}_{k+1} \mid \mathbf{y}_{1:T}) = \mathcal{N}(\mathbf{x}_{k+1} \mid \mathbf{m}_{k+1|T}, \mathbf{P}_{k+1|T}), \tag{2.29}$$

the smoothing equation (Eq. 2.28) implies that the smoothing distribution of $\mathbf{x}_k$ is also Gaussian, $p(\mathbf{x}_k \mid \mathbf{y}_{1:T}) = \mathcal{N}(\mathbf{x}_k \mid \mathbf{m}_{k|T}, \mathbf{P}_{k|T})$ with

$$\begin{aligned} \mathbf{G}_k &= \mathbf{P}_k \, \mathbf{A}_k^\mathsf{T} \, (\mathbf{P}_{k+1}^-)^{-1}, \\ \mathbf{m}_{k|T} &= \mathbf{m}_k + \mathbf{G}_k \, (\mathbf{m}_{k+1|T} - \mathbf{m}_{k+1}^-), \\ \mathbf{P}_{k|T} &= \mathbf{P}_k + \mathbf{G}_k \, (\mathbf{P}_{k+1|T} - \mathbf{P}_{k+1}^-) \, \mathbf{G}_k^\mathsf{T}. \end{aligned} \tag{2.30}$$

Since the smoothing distribution $p(\mathbf{x}_T \mid \mathbf{y}_{1:T})$ is by definition equal to the filtering distribution and thus Gaussian, all smoothing distributions are

Gaussian by induction. The Rauch–Tung–Striebel smoothing algorithm then consists of i) running the Kalman filter for $k = 1, \ldots, T$, ii) initalizing $\mathbf{m}_{T|T} = \mathbf{m}_T, \mathbf{P}_{T|T} = \mathbf{P}_T$ and iii) iterating Equation (2.30) backwards in time for $k = T-1, \ldots, 0$.

2.5.3 Nonlinear Kalman Smoothing

Consider the nonlinear system with additive Gaussian noise (Eq. 2.14). For this system, the smoothing equations (Eq. 2.28) are intractable, as even the filtering distributions are intractable. Similarly to the Gaussian density assumption for filtering, Särkkä and Hartikainen (2010) proposed the assumed Gaussian density framework for smoothing. The idea is to approximate the smoothing recursion step for $\mathbf{x}_k$ by assuming

$$p(\mathbf{x}_{k+1} \mid \mathbf{y}_{1:T}) \approx \mathcal{N}(\mathbf{x}_{k+1} \mid \mathbf{m}_{k+1|T}, \mathbf{P}_{k+1|T}) \tag{2.31}$$

and using a Gaussian approximation to the joint distribution $p(\mathbf{x}_k, \mathbf{x}_{k+1} \mid \mathbf{y}_{1:k})$. These Gaussianity assumptions and the Markov property of the model then imply a Gaussian smoothing distribution $p(\mathbf{x}_k \mid \mathbf{y}_{1:T})$. Furthermore, approximations $p(\mathbf{x}_k \mid \mathbf{y}_{1:k}) \approx \mathcal{N}(\mathbf{x}_k \mid \mathbf{m}_k, \mathbf{P}_k),\ p(\mathbf{x}_{k+1} \mid \mathbf{y}_{1:k}) \approx \mathcal{N}(\mathbf{x}_{k+1} \mid \mathbf{m}_{k+1}^-, \mathbf{P}_{k+1}^-)$, that are required for the joint Gaussian approximation to $p(\mathbf{x}_k, \mathbf{x}_{k+1} \mid \mathbf{y}_{1:k})$, are available from the assumed density Gaussian filter. To obtain the complete Gaussian approximation, the cross-covariance is computed from the assumed density Gaussian filtering results as

$$\mathbf{D}_{k+1} = \int (\mathbf{x}_k - \mathbf{m}_k)\,(\mathbf{f}_k(\mathbf{x}_k) - \mathbf{m}_{k+1}^-)^\mathsf{T}\,\mathcal{N}(\mathbf{x}_k \mid \mathbf{m}_k, \mathbf{P}_k)\,\mathrm{d}\mathbf{x}_k. \tag{2.32}$$

From the assumed Gaussian approximations to $p(\mathbf{x}_{k+1} \mid \mathbf{y}_{1:T})$ and $p(\mathbf{x}_k, \mathbf{x}_{k+1} \mid \mathbf{y}_{1:k})$ one then obtains the Gaussian smoothing distribution $p(\mathbf{x}_k \mid \mathbf{y}_{1:T}) = \mathcal{N}(\mathbf{x}_k \mid \mathbf{m}_{k|T}, \mathbf{P}_{k|T})$, where

$$\begin{aligned}
\mathbf{G}_k &= \mathbf{D}_{k+1}\,(\mathbf{P}_{k+1}^-)^{-1}, \\
\mathbf{m}_{k|T} &= \mathbf{m}_k + \mathbf{G}_k\,(\mathbf{m}_{k+1|T} - \mathbf{m}_{k+1}^-), \\
\mathbf{P}_{k|T} &= \mathbf{P}_k + \mathbf{G}_k\,(\mathbf{P}_{k+1|T} - \mathbf{P}_{k+1}^-)\,\mathbf{G}_k^\mathsf{T}.
\end{aligned} \tag{2.33}$$

Since the Gaussian integral in Equation (2.32) is generally intractable, it is evaluated using a sigma-point rule similarly to the Gaussian integrals arising in the assumed density Gaussian filter. The resulting sigma-point smoother then consists of i) running a sigma-point filter for $k = 1, \ldots, T$, ii) initalizing $\mathbf{m}_{T|T} = \mathbf{m}_T, \mathbf{P}_{T|T} = \mathbf{P}_T$ and iii) iterating Equation (2.33) backwards in time while evaluating the integral (Eq. 2.32) using a sigma-point rule.

2.6 Parameter Estimation

In practice, the state-space model may be defined conditional on some static parameters $\boldsymbol{\theta}$, so that Equation (2.4) becomes

$$\begin{aligned} \mathbf{x}_0 &\sim p(\mathbf{x}_0 \mid \boldsymbol{\theta}), \\ \mathbf{x}_k &\sim p(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \boldsymbol{\theta}), \; k = 1, 2, \ldots, \\ \mathbf{y}_k &\sim p(\mathbf{y}_k \mid \mathbf{x}_k, \boldsymbol{\theta}), \; k = 1, 2, \ldots. \end{aligned} \tag{2.34}$$

For example, in the linear-Gaussian dynamic system (Eq. 2.8), the transition matrices $\mathbf{A}_k$, the measurement matrices $\mathbf{H}_k$, the initial distribution moments $\mathbf{m}_0, \mathbf{P}_0$ and the noise covariances $\mathbf{Q}_k, \mathbf{R}_k$ may depend on the parameter vector $\boldsymbol{\theta}$. In such settings, estimation of the parameters $\boldsymbol{\theta}$ from the observations is of interest. In the following, we review the Bayesian posterior distribution approach and the Markov chain Monte Carlo algorithm for approximating the posterior distribution, as well as the maximum likelihood approach including the expectation–maximization algorithm for likelihood maximization.

2.6.1 Posterior Approximation by Markov Chain Monte Carlo

The Bayesian approach to parameter estimation is to model the parameter as a random variable with a prior distribution $p(\boldsymbol{\theta})$. Estimating the parameter from observations $\mathbf{y}_{1:T}$ then refers to computing the posterior distribution of the parameters conditional on the data, $p(\boldsymbol{\theta} \mid \mathbf{y}_{1:T})$.

Computing the posterior distribution is unfortunately typically intractable. A possible approximation method is the family of Markov chain Monte Carlo algorithms (MCMC, see, e.g., Gelman et al., 2013). The idea of MCMC algorithms is to construct a random process $\boldsymbol{\theta}^1, \boldsymbol{\theta}^2, \boldsymbol{\theta}^3, \ldots$ that moves in the support of parameter so that the process has the Markov property and its limiting distribution is the posterior distribution of interest. In the following, we review the Metropolis–Hastings algorithm (Hastings, 1970) (see, e.g., Gelman et al. (2013) for the use in the Bayesian context). The Metropolis–Hastings algorithm is based on selecting a proposal density $q(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}^i)$ that defines the probability distribution of a proposed new value for the parameter conditional on the parameter value at the previous step. Then, at each iteration of the algorithm one proposes a value from q and either accepts the proposal with a certain probability, or rejects it, in which case the old value is repeated. The acceptance probability is selected so that the Markov chain will have the correct limiting distribution. The algorithm is as follows. First, $\boldsymbol{\theta}^0$ is initialized. Then, for as many steps $i = 0, 1, \dots$ as desired:

- $\boldsymbol{\theta}^*$ is drawn from $q(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}^i)$.
- The acceptance probability $\alpha = \frac{p(\boldsymbol{\theta}^* | \mathbf{y})\, q(\boldsymbol{\theta}^i | \boldsymbol{\theta}^*)}{p(\boldsymbol{\theta}^i | \mathbf{y})\, q(\boldsymbol{\theta}^* | \boldsymbol{\theta}^i)}$ is computed.
- With probability α, $\boldsymbol{\theta}^{i+1} = \boldsymbol{\theta}^*$, else $\boldsymbol{\theta}^{i+1} = \boldsymbol{\theta}^i$.

The proof that the limiting distribution of the resulting Markov chain is the intended posterior distribution is omitted in this thesis. Intuitively, the acceptance probability corrects for the discrepancy between the target posterior distribution and the proposal distribution.

The aforementioned acceptance probability requires the ratio of the posterior densities, while the posterior density is intractable. However, the ratio of the posterior densities can be computed, since the posterior distribution is proportional to $p(\boldsymbol{\theta})\, p(\mathbf{y} \mid \boldsymbol{\theta})$ and the proportionality constant cancels out in the ratio. Then, in the actual algorithm, the acceptance ratio is computed as

$$\alpha = \frac{p(\boldsymbol{\theta}^*)\, p(\mathbf{y} \mid \boldsymbol{\theta}^*)\, q(\boldsymbol{\theta}^i \mid \boldsymbol{\theta}^*)}{p(\boldsymbol{\theta}^i)\, p(\mathbf{y} \mid \boldsymbol{\theta}^i)\, q(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}^i)}. \tag{2.35}$$

In the state-space context, even evaluation of the term $p(\mathbf{y} \mid \boldsymbol{\theta})$ may be intractable as it requires integrating out the latent states $\mathbf{x}_{0:T}$. Combined with filtering algorithms, one may use the prediction error decomposition (Särkkä, 2013),

$$p(\mathbf{y}_{1:T} \mid \boldsymbol{\theta}) = \prod_{k=1}^{T} p(\mathbf{y}_k \mid \mathbf{y}_{1:k-1}, \boldsymbol{\theta}). \tag{2.36}$$

The factors on the right hand side may be evaluated by

$$p(\mathbf{y}_k \mid \mathbf{y}_{1:k-1}, \boldsymbol{\theta}) = \int p(\mathbf{y}_k \mid \mathbf{x}_k, \boldsymbol{\theta})\, p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}, \boldsymbol{\theta})\, \mathrm{d}\mathbf{x}_k, \tag{2.37}$$

where the second factor of the integral is the prediction distribution of the state $\mathbf{x}_k$. In the linear-Gaussian case, this decomposition gives a way to compute the likelihood $p(\mathbf{y}_{1:T} \mid \boldsymbol{\theta})$ along the Kalman filter recursion, when the Kalman filter is run with the parameters $\boldsymbol{\theta}$. This likelihood evaluation may then be plugged into, for example, the Metropolis–Hastings algorithm to obtain an MCMC algorithm sampling from the posterior distribution of the parameters of the linear-Gaussian state-space model.

2.6.2 Maximum Likelihood

If one does not want to define a prior probability distribution for the parameter, or if the posterior distribution would be tedious to compute, one may resort to the classical approach of obtaining a point estimate of the parameter by maximizing the likelihood, or equivalently the log-likelihood. That is, by selecting

$$\hat{\boldsymbol{\theta}}_{\mathrm{ML}} := \arg\max_{\boldsymbol{\theta}} \log p(\mathbf{y}_{1:T} \mid \boldsymbol{\theta}), \tag{2.38}$$

where the logarithm is typically used due to computational reasons. The prediction error decomposition (Eq. 2.36) may be used for evaluating the likelihood in the state-space model case. Furthermore, as some optimization methods employ information about the gradient of the likelihood function, one may be interested in computing

$$\frac{\partial}{\partial \boldsymbol{\theta}} \log p(\mathbf{y}_{1:T} \mid \boldsymbol{\theta}). \tag{2.39}$$

In the linear-Gaussian case, this gradient may be evaluated along the Kalman filter recursion by using the so-called sensitivity equations that are obtained by differentiating the Kalman filter recursion (Gupta and Mehra, 1974). An alternative approach based on smoother results is to use Fisher's identity (Segal and Weinstein, 1989).

Note that the likelihood maximization framework may also be used to find the mode of a posterior distribution, since the log-likelihood may be replaced by the log-posterior by adding the logarithm of the prior,

$$\hat{\boldsymbol{\theta}}_{\mathrm{MAP}} := \arg\max_{\boldsymbol{\theta}} \left(\log p(\mathbf{y}_{1:T} \mid \boldsymbol{\theta}) + \log p(\boldsymbol{\theta})\right). \tag{2.40}$$

However, the Bayesian interpretation of this maximum a posteriori estimation is questionable since the posterior mode is sensitive to reparametrizations of the model (see, e.g., Druilhet and Marin, 2007, and references therein).

2.6.3 Expectation–Maximization

The expectation–maximization (EM) algorithm (Dempster et al., 1977) is an algorithm for maximizing the marginal likelihood or posterior density in settings with some unobserved variables, that avoids the need to explicitly evaluate the marginal likelihood. In the state-space model case, as discussed by Shumway and Stoffer (1982), the EM algorithm may be applied by considering the latent states as the unobserved variables.

In the general setting, the log-likelihood has the following lower bound:

$$\log p(\mathbf{y}_{1:T} \mid \boldsymbol{\theta}) \geq \int q(\mathbf{x}_{0:T}) \log \frac{p(\mathbf{x}_{0:T}, \mathbf{y}_{1:T} \mid \boldsymbol{\theta})}{q(\mathbf{x}_{0:T})} \, \mathrm{d}\mathbf{x}_{0:T}, \tag{2.41}$$

where q is an arbitrary probability distribution over $\mathbf{x}_{0:T}$. The basic idea of EM is to iterate alternating between maximizing this bound i) over q and ii) over $\boldsymbol{\theta}$. When $\boldsymbol{\theta} = \boldsymbol{\theta}^{(n)}$, the maximum over q is obtained by

$$q(\mathbf{x}_{0:T}) = p(\mathbf{x}_{0:T} \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}^{(n)}), \tag{2.42}$$

in which case the bound, when maximizing over $\boldsymbol{\theta}$, equals

$$\log p(\mathbf{y}_{1:T} \mid \boldsymbol{\theta}) \geq \int p(\mathbf{x}_{0:T} \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}^{(n)}) \log \frac{p(\mathbf{x}_{0:T}, \mathbf{y}_{1:T} \mid \boldsymbol{\theta})}{p(\mathbf{x}_{0:T} \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}^{(n)})} \, \mathrm{d}\mathbf{x}_{0:T}, \tag{2.43}$$

which further decomposes as

$$\begin{aligned} =&\mathbb{E}[\log p(\mathbf{x}_{0:T}, \mathbf{y}_{1:T} \mid \boldsymbol{\theta}) \mid \boldsymbol{\theta}^{(n)}, \mathbf{y}_{1:T}] \\ &- \int p(\mathbf{x}_{0:T} \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}^{(n)}) \log p(\mathbf{x}_{0:T} \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}^{(n)}) \, \mathrm{d}\mathbf{x}_{0:T}. \end{aligned} \tag{2.44}$$

Since the latter term is independent of $\boldsymbol{\theta}$, it may be ignored when maximizing over $\boldsymbol{\theta}$. Then, the EM algorithm consists of two steps:

- E-step: compute the function $\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{n}) = \mathbb{E}[\log p(\mathbf{x}_{0:T}, \mathbf{y}_{1:T} \mid \boldsymbol{\theta}) \mid \boldsymbol{\theta}^{(n)}, \mathbf{y}_{1:T}]$ (corresponds to maximizing the bound over q).

- M-step: maximize over $\boldsymbol{\theta}$, i.e., set $\boldsymbol{\theta}^{(n+1)} := \arg\max_{\boldsymbol{\theta}} \mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{(n)})$.

In the state-space context, the $\mathcal{Q}$-function further factorizes as

$$\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{(n)}) = \mathrm{I}_1(\boldsymbol{\theta}, \boldsymbol{\theta}^{(n)}) + \mathrm{I}_2(\boldsymbol{\theta}, \boldsymbol{\theta}^{(n)}) + \mathrm{I}_3(\boldsymbol{\theta}, \boldsymbol{\theta}^{(n)}), \tag{2.45}$$

where the terms are

$$\mathrm{I}_1(\boldsymbol{\theta}, \boldsymbol{\theta}^{(n)}) = \mathbb{E}[\log p(\mathbf{x}_0 \mid \boldsymbol{\theta}) \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}^{(n)}], \tag{2.46}$$

$$\mathrm{I}_2(\boldsymbol{\theta}, \boldsymbol{\theta}^{(n)}) = \sum_{k=1}^{T} \mathbb{E}[\log p(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \boldsymbol{\theta}) \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}^{(n)}], \tag{2.47}$$

$$\mathrm{I}_3(\boldsymbol{\theta}, \boldsymbol{\theta}^{(n)}) = \sum_{k=1}^{T} \mathbb{E}[\log p(\mathbf{y}_k \mid \mathbf{x}_k, \boldsymbol{\theta}) \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}^{(n)}]. \tag{2.48}$$

Of these, the first and last term are expectations over the marginal fixed-interval smoothing distributions, and the middle term is an expectation over the joint smoothing distribution of two consecutive states. In the linear-Gaussian case, these are obtained by running the Rauch–Tung–Striebel smoother. The EM algorithm then consists of iteratively running the smoother in the E-step and optimizing the parameters in the M-step. In the more general case, computing the smoothing distributions is intractable, and thus the E-step cannot be evaluated in closed form.

3. Particle Filtering

3.1 Sequential Importance Sampling with Resampling

Following Doucet et al. (2001), we present the sequential importance sampling with resampling particle filter algorithm based on first considering sequential importance sampling as a special case of importance sampling and then introducing the resampling step.

3.1.1 Importance Sampling

Assume one desires to approximate a distribution $p(\mathbf{x})$ but direct sampling from p is not possible. In importance sampling, samples are drawn from another distribution $q(\mathbf{x})$, known as the importance distribution, and then weighted to correct for the discrepancy between the distributions p and q. To motivate the importance sampling approach, consider the expected value of a test function f, that is,

$$\int f(\mathbf{x})\, p(\mathbf{x})\, \mathrm{d}\mathbf{x}. \tag{3.1}$$

Defining the unnormalized importance weight function as $v(\mathbf{x}) = p(\mathbf{x})/q(\mathbf{x})$, the integral may be expressed as

$$= \frac{\int f(\mathbf{x})\, v(\mathbf{x})\, q(\mathbf{x})\, \mathrm{d}\mathbf{x}}{\int v(\mathbf{x})\, q(\mathbf{x})\, \mathrm{d}\mathbf{x}}, \tag{3.2}$$

where both integrals are expectations with respect to q. Let us approximate both integrals using the same Monte Carlo sample $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(N)}$ drawn from q so that the expression becomes

$$\approx \frac{\frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}^{(i)})\, v(\mathbf{x}^{(i)})}{\frac{1}{N}\sum_{i=1}^{N} v(\mathbf{x}^{(i)})}. \tag{3.3}$$

Defining the normalized importance weights as

$$w^{(i)} = \frac{v(\mathbf{x}^{(i)})}{\sum_{j=1}^{N} v(\mathbf{x}^{(j)})}, \tag{3.4}$$

the approximation may be expressed as

$$= \sum_{i=1}^{N} f(\mathbf{x}^{(i)})\, w^{(i)}. \tag{3.5}$$

Furthermore, since the same weighted Monte Carlo sample may be used with any test function f, the importance sample may be interpreted as an approximation to the density p by a linear combination of Dirac deltas,

$$p(\mathbf{x}) \approx \sum_{i=1}^{N} w^{(i)}\, \delta(\mathbf{x} - \mathbf{x}^{(i)}). \tag{3.6}$$

3.1.2 Sequential Importance Sampling

In principle, the importance sampling approach discussed in the previous section could be used for state-space models by sampling from $p(\mathbf{x}_{0:T} \mid \mathbf{y}_{1:T})$ using some importance distribution $q(\mathbf{x}_{0:T} \mid \mathbf{y}_{1:T})$. However, as such it cannot be used sequentially. Furthermore, it may be difficult to directly construct a reasonable importance distribution for the complete state sequence. Sequential importance sampling is based on considering a sequence of importance distributions $q(\mathbf{x}_{0:k} \mid \mathbf{y}_{1:k})$ $(k = 1, 2, \ldots)$ for target distributions $p(\mathbf{x}_{0:k} \mid \mathbf{y}_{1:k})$ so that the sequence has the property

$$q(\mathbf{x}_{0:k+1} \mid \mathbf{y}_{1:k+1}) = q(\mathbf{x}_{k+1} \mid \mathbf{x}_{0:k}, \mathbf{y}_{1:k+1})\, q(\mathbf{x}_{0:k} \mid \mathbf{y}_{1:k}), \tag{3.7}$$

where $q(\mathbf{x}_{k+1} \mid \mathbf{x}_{1:k}, \mathbf{y}_{1:k+1})$ is a conditional importance distribution for the latent state at the $k + 1$th step. Then, a sample from $q(\mathbf{x}_{0:k+1} \mid \mathbf{y}_{1:k+1})$ may be constructed based on a sample from $q(\mathbf{x}_{0:k} \mid \mathbf{y}_{1:k})$ by augmenting $\mathbf{x}_{0:k}^{(i)}$ by $\mathbf{x}_{k+1}$ drawn from $q(\mathbf{x}_{k+1} \mid \mathbf{x}_{0:k}^{(i)}, \mathbf{y}_{1:k+1})$. The unnormalized importance weights may be simplified as follows:

$$v(\mathbf{x}_{0:k+1}^{(i)}) = \frac{p(\mathbf{x}_{0:k+1}^{(i)} \mid \mathbf{y}_{1:k+1})}{q(\mathbf{x}_{0:k+1}^{(i)} \mid \mathbf{y}_{1:k+1})} = \frac{p(\mathbf{x}_{k+1}^{(i)} \mid \mathbf{x}_{k}^{(i)})\, p(\mathbf{y}_{k+1} \mid \mathbf{x}_{k+1}^{(i)})}{q(\mathbf{x}_{k+1}^{(i)} \mid \mathbf{y}_{1:k}, \mathbf{x}_{0:k}^{(i)})\, p(\mathbf{y}_{k+1} \mid \mathbf{y}_{1:k})}\, v(\mathbf{x}_{0:k}^{(i)}). \tag{3.8}$$

Furthermore, since the weights are normalized, the constant factor $p(\mathbf{y}_{k+1} \mid \mathbf{y}_{1:k})$ can be omitted. The sequential importance sampling algorithm then operates by initially constructing an importance sampling based sample for $p(\mathbf{x}_0)$ and then iterating for $k = 1, 2, \ldots$:

1. For $i = 1, \ldots, N$: draw $\mathbf{x}_k^{(i)} \sim q(\mathbf{x}_k^{(i)} \mid \mathbf{x}_{0:k-1}^{(i)}, \mathbf{y}_{1:k})$.

2. For $i = 1, \ldots, N$: compute updated weights $v_k^{(i)} := w_{k-1}^{(i)} \frac{p(\mathbf{x}_k^{(i)} | \mathbf{x}_{k-1}^{(i)})\, p(\mathbf{y}_k | \mathbf{x}_k^{(i)})}{q(\mathbf{x}_k^{(i)} | \mathbf{y}_{1:k}, \mathbf{x}_{0:k-1}^{(i)})}$.

3. For $i = 1, \ldots, N$: normalize the weights: $w_k^{(i)} = \frac{v_k^{(i)}}{\sum_j v_k^{(j)}}$.

3.1.3 Resampling

A drawback of the aforementioned sequential importance sampling algorithm is that eventually one sample sequence $\mathbf{x}_{0:k}^{(i)}$ will get almost all weight and thus the sample becomes degenerate. To fix this problem, Gordon et al. (1993) introduced the following resampling step: At each time step, after the sampled values have been drawn and the weights computed, the weighted sample is replaced by a new sample that is randomly drawn from the original sample, so that the expected number of times a particle gets selected is proportional to the importance weight of that particle. After this resampling step is performed, the weights are reset to $1/N$. This sequential importance sampling/resampling algorithm then consists of first drawing a sample $\mathbf{x}_0^{(1)}, \ldots, \mathbf{x}_0^{(N)}$ from $p(\mathbf{x}_0)$ and then iterating

1. For $i = 1, \ldots, N$: draw $\tilde{\mathbf{x}}_k^{(i)} \sim q(\mathbf{x}_k^{(i)} \mid \mathbf{x}_{0:k-1}^{(i)}, \mathbf{y}_{1:k})$.

2. For $i = 1, \ldots, N$: compute updated weights $v_k^{(i)} := w_{k-1}^{(i)} \frac{p(\tilde{\mathbf{x}}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})\, p(\mathbf{y}_k|\tilde{\mathbf{x}}_k^{(i)})}{q(\tilde{\mathbf{x}}_k^{(i)}|\mathbf{y}_{1:k-1}, \mathbf{x}_{0:k-1}^{(i)})}$.

3. For $i = 1, \ldots, N$: normalize the weights: $\tilde{w}_k^{(i)} = \frac{v_k^{(i)}}{\sum_j v_k^{(j)}}$.

4. Resample: draw new particles $(\mathbf{x}_{0:k}^{(i)} \mid i = 1, \ldots, N)$ and reset the weights to $w_k^{(i)} = 1/N$.

This algorithm and its variants are known as particle filters, and the weighted sample values are typically called particles.

For the resampling algorithm, a typical requirement is that the expected number of times the particle $\tilde{\mathbf{x}}_{0:k}^{(i)}$ gets replicated equals $\tilde{w}_k^{(i)} N$, or alternatively a stronger condition that for each i, j, the probability that $\mathbf{x}_{0:k}^{(i)}$ is selected to be a copy of $\tilde{\mathbf{x}}_{0:k}^{(j)}$ equals the weight $\tilde{w}_k^{(j)}$. A simple resampling algorithm is the so-called multinomial resampling algorithm where each new particle is drawn independently. This was used by Gordon et al. (1993). Various other alternatives have been proposed, see Li et al. (2015) for a recent review. Note that it is not necessary to perform the resampling at every step. Possible alternatives are resampling at fixed intervals, or when the so-called effective sample size (Liu and Chen, 1995) is below a threshold. When resampling is not performed, $w_k^{(i)}$ is set equal to $\tilde{w}_k^{(i)}$ in the above algorithm.

For the purposes of solving the filtering problem, one may simply discard the old states, that is, consider samples $\mathbf{x}_k^{(1)}, \ldots, \mathbf{x}_k^{(N)}$ where $\mathbf{x}_k^{(i)}$ is the kth state in $\mathbf{x}_{0:k}^{(i)}$. Furthermore, as the importance distribution $q(\tilde{\mathbf{x}}_k^{(i)} \mid \mathbf{x}_{0:k-1}^{(i)}, \mathbf{y}_{1:k})$ is typically selected so that the conditioning is only on $(\mathbf{x}_{k-1}^{(i)}, \mathbf{y}_k)$, the state values from previous time steps need not be stored in the particle filter algorithm.

3.2 Importance Distributions

The choice of the importance distribution $q(\tilde{\mathbf{x}}_k^{(i)} \mid \mathbf{x}_{k-1}^{(i)}, \mathbf{y}_k)$ impacts the performance of the algorithm. Natural requirements for implementability of the algorithm are that evaluating the density pointwise and sampling from it need to be feasible. Gordon et al. (1993) proposed the use of the dynamic model density $p(\mathbf{x}_k \mid \mathbf{x}_{k-1}^{(i)})$. In this case, the dynamic model density and the importance distribution cancel in the weight expression, and thus the weight factors are proportional to the observation model density. The resulting algorithm is known as the bootstrap filter.

While the bootstrap filter is simple to implement, the performance of the filter may be improved by selecting an importance distribution that matches the target better by taking into account the new observation. The optimal importance distribution (in terms of minimizing the variance of the weights) is the conditional distribution of the state conditioned on the previous state and the new observation, $p(\mathbf{x}_k \mid \mathbf{x}_{k-1}^{(i)}, \mathbf{y}_k)$ (Doucet et al., 2000, and references therein). However, this optimal importance density is not generally available in closed form. Doucet et al. (2000) discuss Gaussian approximations such as the Laplace approximation or a Gaussian distribution formed by using an extended Kalman filter type update step obtained by linearizing the observation model. Similarly, sigma-point filters may be used. For example, van der Merwe et al. (2000) proposed the unscented particle filter that uses the unscented transform based sigma-point filter to generate a Gaussian importance distribution.

For convergence of the particle filter algorithm (discussed in Section 3.6), it is beneficial to obtain bounded importance weights. Boundedness depends on the tail behavior of the importance density. To ensure boundedness of the weights, scaling the variance of a Gaussian approximation based importance distribution or using a t-distribution instead of a Gaussian has been proposed (Cappé et al., 2005).

3.3 Rao–Blackwellization

The Rao–Blackwellized, or marginalized, particle filter (Chen and Liu, 2000; Doucet et al., 2000) may be used in situations where part of the state can be integrated out analytically. That is, when the state can be decomposed as $\mathbf{x}_k = (\mathbf{x}_{k,1}, \mathbf{x}_{k,2})$ so that the conditional filtering problem for one component conditional on the other component, $p(\mathbf{x}_{k,2} \mid \mathbf{y}_{1:k}, \mathbf{x}_{0:k,1})$, is solvable in closed form. The Rao–Blackwellized particle filter then constructs the particle approximation only for the component $\mathbf{x}_{k,1}$ and uses the analytic approach to the component $\mathbf{x}_{k,2}$.

For example, consider the following conditional linear-Gaussian state-space model

$$\begin{aligned} \mathbf{x}_{k,1} \mid \mathbf{x}_{k-1} &\sim p(\mathbf{x}_{k,1} \mid \mathbf{x}_{k-1,1}), \\ \mathbf{x}_{k,2} \mid \mathbf{x}_{k-1} &\sim \mathcal{N}(\mathbf{A}_{k-1}(\mathbf{x}_{k,1})\, \mathbf{x}_{k-1,2}, \mathbf{Q}_{k-1}(\mathbf{x}_{k,1})), \\ \mathbf{y}_k \mid \mathbf{x}_k &\sim \mathcal{N}(\mathbf{H}_k(\mathbf{x}_{k,1})\, \mathbf{x}_{k,2}, \mathbf{R}_k(\mathbf{x}_{k,1})), \end{aligned} \tag{3.9}$$

with the usual conditional independence properties for the states $\mathbf{x}_k$ and observations $\mathbf{y}_k$. Conditional on the sequence $\mathbf{x}_{0:k,1}$, this is a linear-Gaussian state-space model with the state $\mathbf{x}_{k,2}$ and thus the Kalman filter can be applied. The Rao–Blackwellized particle filter algorithm then is a particle filter for $\mathbf{x}_{k,1}$ where the Kalman filter is used to evaluate $p(\mathbf{x}_{k,1} \mid \mathbf{x}_{0:k-1,1})$ and $p(\mathbf{y}_k \mid \mathbf{x}_{0:k,1})$. Note that the conditional independence properties do not hold when $\mathbf{x}_{k,2}$ is marginalized out and thus the conditioning in the aforementioned marginal dynamic model and observation model densities is on the entire history of $\mathbf{x}_{\cdot,1}$.

Schön et al. (2005) apply the Rao–Blackwellized particle filter for various generalizations of the conditional linear-Gaussian model. Another case where the Rao–Blackwellized particle filter is applicable is when one can marginalize over static parameters analytically (e.g. Storvik, 2002).

3.3.1 Rao–Blackwellized Monte Carlo Data Association

Särkkä et al. (2007) suggested a Rao–Blackwellized particle filter for multiple target tracking. Multiple target tracking (see, e.g., Challa et al., 2011) refers to the problem of estimating the locations of multiple moving targets based on noisy measurements. Generally, a measurement does not contain information about which target it comes from. Therefore, the multiple target tracking problem comprises the subproblems of estimating the locations of the objects, determining which target each measurement comes from (data-association), as well as estimating the number of targets.

In the Rao–Blackwellized Monte Carlo data association (RBMCDA) by Särkkä et al. (2007), each target is assumed to have a state following linear-Gaussian dynamics. Denote by $\mathbf{x}_{k,j}$ the state of the jth target at the kth time-step. Then, for each target, the state sequence is assumed Markovian with

$$\mathbf{x}_{k,j} \mid \mathbf{x}_{k-1,j} \sim \mathcal{N}(\mathbf{A}_{k-1}\,\mathbf{x}_{k-1,j},\ \mathbf{Q}_{k-1}), \tag{3.10}$$

and the states of different targets are assumed independent of each other. The measurement model is such that at every step k, a discrete random variable c_k tells which target the measurement comes from. Conditional on this data-association c_k, the measurement is then linear-Gaussian with

$$\mathbf{y}_k \mid c_k, \mathbf{x}_k \sim \mathcal{N}(\mathbf{H}_k\,\mathbf{x}_{k,c_k}, \mathbf{R}_k). \tag{3.11}$$

To complete the model, one then needs to define a probability distribution for the data-associations, in form of conditional distributions $p(c_k \mid c_{1:k-1})$. A varying number of targets is handled so that at each step k, c_k may be either some target existing in the state (observed during steps $1, \ldots, k-1$) or a new target. The state of a new target follows a Gaussian distribution $\mathcal{N}(\mathbf{m}_0, \mathbf{P}_0)$ at the time of its first observation.

Overall, the model described above is a state-space model where the state consists of the data-associations c_k and the states $\mathbf{x}_{k,j}$ of the targets, although the dynamic model is not necessary Markovian in that c_k depends on the entire history $c_{0:k-1}$. Furthermore, the model is a conditional linear-Gaussian state-space model, where the conditioning is on c, and thus the Rao–Blackwellized particle filter may be applied. In addition, since in each sampling step there is a finite number of possible data-associations, the optimal importance distribution can be used. In the resulting algorithm, each particle stores the data-association history, $c_{0:k}^{(i)}$, the number of targets, $T_k^{(i)}$, and the parameters of the Gaussian conditional distributions of the states of the targets, $\mathbf{m}_{k,j}^{(i)}, \mathbf{P}_{k,j}^{(i)}$. Initially, all particles are empty and have weight $w_0^{(i)} = 1/N$. Then, for $k = 1, \ldots$ the algorithm iterates the following steps

1. For each particle $i = 1, \ldots, N$:

   (a) For each target $j = 1, \ldots, T_k^{(i)}$, compute the Kalman filter prediction to form $\mathbf{m}_{k,j}^{-(i)}, \mathbf{P}_{k,j}^{-(i)}$.

   (b) For each target $j = 1, \ldots, T_k^{(i)}$, as well as the possible new target $j = T_k^{(i)} + 1$, evaluate the likelihood $p(\mathbf{y}_k \mid c_k = j, c_{0:k-1}^{(i)}, \mathbf{y}_{1:k})$ using the Kalman filter update step.

   (c) Evaluate the optimal importance distribution $p(c_k \mid c_{0:k-1}^{(i)}, \mathbf{y}_{1:k})$.

   (d) Draw the data association $c_k^{(i)}$ from the optimal importance distribution.

   (e) For the selected target, update $\mathbf{m}_{k,c_k^{(i)}}^{(i)}, \mathbf{P}_{k,c_k^{(i)}}^{(i)}$ using the Kalman filter update step. For other targets, let the updated parameters of state distributions be equal to the parameters of the predicted state distributions.

   (f) Update the weight.

2. Normalize weights.

3. Possible resampling step.

In addition to the simplified description above, Särkkä et al. (2007) consider also the possibility of clutter measurements, that is, measurements with data-association $c_k = 0$ that are independent of the target states, and target deletion, that is, the possibility of removing targets from the state if they are not observed for a long time. Furthermore, for target dynamic and measurement models that do not have the conditional linear-Gaussian structure, they propose an approximative algorithm based on using nonlinear Kalman filters in place of the Kalman filter within the Rao–Blackwellized particle filter.

3.4 Particle Smoothing

In principle, the particle filter may be used for approximating the smoothing distributions $p(\mathbf{x}_k \mid \mathbf{y}_{0:T})$ directly by using the samples $(w_T^{(i)}, \mathbf{x}_{0:T}^{(i)})$ and considering only the kth state in each sampled sequence $\mathbf{x}_{0:T}^{(i)}$. However, there is a degeneracy problem caused by the fact that at each resampling it is possible that some distinct sequences are removed, while no new values for the old states are generated during the filtering recursion. It is then likely that after multiple resamplings, there will be only few, or even only one, distinct values for the initial sequences $\mathbf{x}_{0:k}$ that have survived the resampling steps. Thus, the particle filter produces a poor approximation to $p(\mathbf{x}_k \mid \mathbf{y}_{0:T})$. Therefore, it is of interest to use smoothing algorithms that utilize also those values $\mathbf{x}_k^{(i)}$ generated during the filtering recursion that are discarded in further resampling steps.

Hürzeler and Künsch (1998) as well as Doucet et al. (2000) proposed a particle smoothing algorithm that is based on reweighting the samples $(w_k, \mathbf{x}_k^{(i)})$ generated during the filtering recursion to obtain weighted samples $(w_{k|T}, \mathbf{x}_k^{(i)})$ that approximate the smoothing distributions $p(\mathbf{x}_k \mid \mathbf{y}_{0:T})$. The reweighting is performed in a backward pass based on the smoothing equation introduced by Kitagawa (1987) (see Eq. 2.28). The algorithm consists of the following steps.

1. Run the particle filter for $k = 0, \ldots, T$, storing all intermediate filtering distributions $(w_k^{(i)}, \mathbf{x}_k^{(i)})$.

2. Initialize for $i = 1, \ldots, N$: $w_{T|T}^{(i)} := w_T^{(i)}$.

3. For $k = T-1, T-2, \ldots, 0$:

   - For $i = 1, \ldots, N$: set

$$w_{k|T}^{(i)} = \sum_{j=1}^{N} w_{k+1|T}^{(j)} \frac{w_k^{(i)} \, p(\mathbf{x}_{k+1}^{(j)} \mid \mathbf{x}_k^{(i)})}{\sum_{l=1}^{N} w_k^{(l)} \, p(\mathbf{x}_{k+1}^{(j)} \mid \mathbf{x}_k^{(l)})}.$$

A drawback of this reweighting particle smoother is that the computational complexity is quadratic in the number of particles N since the reweighting considers all possible pairs of particles. An alternative algorithm is the backward simulation suggested by Godsill et al. (2004), where smoothed trajectories are simulated by sampling from the particles generated in the filter instead of considering all particles.

3.5 Particle Markov Chain Monte Carlo

Particle Markov chain Monte Carlo methods (PMCMC, Andrieu et al., 2010) are algorithms for approximating the joint posterior of states and static parameters in state-space models of the form (Eq. 2.34), that is, the distribution $p(\mathbf{x}_{0:T}, \boldsymbol{\theta} \mid \mathbf{y}_{1:T})$. They are MCMC algorithms (cf. Section 2.6.1), that is, they produce a Markovian random sequence of pairs $(\mathbf{x}_{0:T}, \boldsymbol{\theta})$ such that the limiting distribution of the Markov chain is the intended posterior distribution. They use particle filtering type steps within the MCMC algorithm. However, they are not filtering algorithms in that they cannot be used recursively but are intended for batch estimation conditional on a fixed sequence of observations $\mathbf{y}_{1:T}$.

The PMCMC algorithms may also be used as particle smoothing algorithms simply by letting the parameters $\boldsymbol{\theta}$ be constant (Svensson et al., 2015).

In the remainder of this section, we review the particle marginal Metropolis–Hastings and particle Gibbs algorithms discussed by Andrieu et al. (2010).

3.5.1 Particle Marginal Metropolis–Hastings

The particle marginal Metropolis–Hastings algorithm is based on an estimate of the marginal likelihood $p(\mathbf{y}_{1:T} \mid \boldsymbol{\theta})$ obtained from the particle filter. This estimate is based on approximating the factors of the prediction error decomposition (Eq. 2.36), that is,

$$p(\mathbf{y}_{1:T} \mid \boldsymbol{\theta}) \approx \hat{p}(\mathbf{y}_{1:T} \mid \boldsymbol{\theta}) = \prod_{k=1}^{T} \hat{p}(\mathbf{y}_k \mid \boldsymbol{\theta}, \mathbf{y}_{1:k-1}). \tag{3.12}$$

The estimate of each factor is obtained as a weighted average of the unnormalized weights:

$$\hat{p}(\mathbf{y}_k \mid \boldsymbol{\theta}, \mathbf{y}_{1:k-1}) = \sum_{i=1}^{N} w_{k-1}^{(i)} v_k^{(i)}. \tag{3.13}$$

Recall that if resampling is performed, $w_{k-1}^{(i)}$ is equal to $1/N$.

The particle marginal Metropolis–Hastings algorithm is based on using the marginal likelihood estimate in place of the true likelihood in the Metropolis–Hastings algorithm. The algorithm is as follows:

1. Pick an initial $\boldsymbol{\theta}^0$.

2. Run the particle filter conditional on $\boldsymbol{\theta}^0$ to obtain a sample $(w_T^{(i)}, \mathrm{x}_{0:T}^{(i)}), i = 1, \ldots, N$ and the marginal likelihood estimate $\hat{p}^0$.

3. Draw sample state sequence $\mathrm{x}_{0:T,0}$ from $\mathrm{x}_{0:T}^{(1:N)}$ with probabilities $w_T^{(1:N)}$.

4. For $j = 1, 2, \ldots$:

   (a) Propose a new parameter value $\boldsymbol{\theta}^* \sim q(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}^{j-1})$.

   (b) Run the particle filter conditional on $\boldsymbol{\theta}^*$ to:

   - Evaluate $\hat{p}^* = \hat{p}(\mathbf{y}_{1:T} \mid \boldsymbol{\theta}^*)$.
   - Obtain the particles $(w_T^{(i)},\ \mathrm{x}_{0:T}^{(i)}),\ i = 1, \ldots, N$.

   (c) Select the proposed state sequence $\mathrm{x}_{0:T}^*$ from $\mathrm{x}_{0:T}^{(1:N)}$ with probabilities $w_T^{(1:N)}$.

   (d) Evaluate $\alpha = \frac{q(\boldsymbol{\theta}^{j-1}|\boldsymbol{\theta}^*)}{q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{j-1})} \frac{p(\boldsymbol{\theta}^*)\,\hat{p}^*}{p(\boldsymbol{\theta}^{j-1})\hat{p}^{j-1}}$.

   (e) With probability α, $(\boldsymbol{\theta}^j, \hat{p}^j, \mathrm{x}_{0:T,j}) = (\boldsymbol{\theta}^*, \hat{p}^*, \mathrm{x}_{0:T}^*)$, else $(\boldsymbol{\theta}^j, \hat{p}^j, \mathrm{x}_{0:T,j}) = (\boldsymbol{\theta}^{j-1}, \hat{p}^{j-1}, \mathrm{x}_{0:T,j-1})$.

Even though the marginal likelihood estimate is used in place of the exact marginal likelihood, this algorithm admits the correct posterior distribution $p(\mathrm{x}_{0:T}, \boldsymbol{\theta} \mid \mathbf{y}_{1:T})$ as the stationary distribution.

3.5.2 Particle Gibbs

The particle Gibbs algorithm is based on updating the static parameters and the latent state sequence in two separate steps, one generating new static parameters (holding the states fixed) and one generating new states (holding the parameters fixed). For the parameters, one may for example draw from the conditional posterior $p(\boldsymbol{\theta} \mid \mathrm{x}_{0:T}, \mathbf{y}_{1:T})$ as in a (blocked) Gibbs sampler (Geman and Geman, 1984; Gelfand and Smith, 1990). For the state sequence, sampling from the conditional posterior $p(\mathrm{x}_{0:T} \mid \boldsymbol{\theta}, \mathbf{y}_{1:T})$ is typically not feasible. The state update is performed using the conditional particle filter, which is a modification of the particle filter algorithm that takes a reference state sequence $\mathbf{x}^*_{0:T}$ as input. The conditional particle filter is the particle filter conditional on the first particle $\mathbf{x}^{(1)}_{0:T}$ obtaining the values in the reference path. The practical implementation of the algorithm is otherwise exactly like the particle filter, except that the first particle $\mathbf{x}^{(1)}_{0:T}$ is not sampled but is copied from the reference path and that in the resampling step, the first particle is guaranteed to survive. That is,

1. For $i = 2, \ldots, N$: draw $\tilde{\mathbf{x}}^{(i)}_k \sim q(\tilde{\mathbf{x}}^{(i)}_k \mid \mathbf{x}^{(i)}_{0:k-1}, \mathbf{y}_{1:k}, \boldsymbol{\theta})$. Set $\tilde{\mathbf{x}}^{(1)}_k = \mathbf{x}^*_k$.

2. For $i = 1, \ldots, N$: compute updated weights $v^{(i)}_k := w^{(i)}_{k-1} \frac{p(\tilde{\mathbf{x}}^{(i)}_k | \mathbf{x}^{(i)}_{k-1}, \boldsymbol{\theta})\, p(\mathbf{y}_k | \tilde{\mathbf{x}}^{(i)}_k, \boldsymbol{\theta})}{q(\tilde{\mathbf{x}}^{(i)}_k | \mathbf{y}_{1:k-1}, \mathbf{x}^{(i)}_{0:k-1}, \boldsymbol{\theta})}$.

3. For $i = 1, \ldots, N$: normalize the weights: $\tilde{w}^{(i)}_k = \frac{v^{(i)}_k}{\sum_j v^{(j)}_k}$.

4. Resample conditional on $\mathbf{x}^{(1)}_k$ being copied from $\tilde{\mathbf{x}}^{(1)}_k$ and reset weights.

When multinomial resampling is used, the conditional resampling (step 4) can be implemented simply by sampling the particles $2, \ldots, N$ just as in the normal multinomial resampling.

Then, the procedure of i) picking a reference path, ii) running the conditional particle filter, iii) sampling a path $\mathbf{x}^{(i)}_{0:T}$ with the probabilities $w^{(i)}_T$ has the property that the conditional posterior $p(\mathbf{x}_{0:T} \mid \boldsymbol{\theta}, \mathbf{y}_{0:T})$ is an invariant distribution of this procedure. Thus, it can be plugged into an MCMC algorithm. The resulting particle Gibbs algorithm then consists of selecting initial values $(\boldsymbol{\theta}^0, \mathbf{x}_{0:T,0})$ and iterating for $j = 1, 2 \ldots$:

1. Run the conditional particle filter with $\boldsymbol{\theta}^{j-1}$ and reference path $\mathbf{x}_{0:T,j-1}$ to obtain $(w^{(i)}_T, \mathbf{x}^{(i)}_{0:T})$, $i = 1, \ldots, N$.

2. Draw index $K \in \{1, \ldots, N\}$ with probabilities $w^{(i)}_T$.

3. Set $\mathbf{x}_{0:T,j} := \mathbf{x}^{(K)}_{0:T}$.

4. Draw $\boldsymbol{\theta}^j \sim p(\boldsymbol{\theta} \mid \mathbf{x}_{0:T,j}, \mathbf{y}_{1:T})$.

This algorithm admits $p(\boldsymbol{\theta}, \mathrm{x}_{0:T} \mid \mathbf{y}_{1:T})$ as the stationary distribution.

Note that Andrieu et al. (2010) present the conditional particle filter slightly differently using ancestor sequences of particles. See, for example, Lindsten et al. (2014) for the formulation where the fixed path has a fixed index.

3.6 Convergence

For motivating the use of the particle filter algorithms, an interesting question is whether one can obtain theoretical guarantees that the approximations (Eq. 3.6) to the filtering distributions in some sense become exact when the number of particles approaches infinity. Since the particle filter is a random algorithm, this becoming exact is to be interpreted as some kind of convergence of random variables. In this section, we review the convergence concepts used for random variables as well as some convergence results for particle filters.

3.6.1 Convergence of Random Variables

Consider a sequence of (scalar) random variables $x_1, x_2, \ldots$, and another random variable x. There exist various probabilistic concepts of convergence of the sequence $x_1, x_2, \ldots$ to x (see, e.g., Jacod and Protter, 2003). Here we present only the definitions of almost sure convergence and convergence in L^p.

- $x_1, x_2, \ldots$ *converges* to x *almost surely*, if
$$\Pr(\lim_{i\to\infty} x_i = x) = 1. \tag{3.14}$$
- $x_1, x_2, \ldots$ *converges* to x *in* L^p with $1 \leq p \leq \infty$, if $\mathbb{E}(|x_i|^p)$ and $\mathbb{E}(|x|^p)$ are finite and
$$\lim_{i\to\infty} \mathbb{E}(|x_i - x|^p) = 0. \tag{3.15}$$

3.6.2 Particle Filter Convergence Results

The survey by Crisan and Doucet (2002) and references therein present various convergence results for particle filters. In the convergence analysis of particle filters, the observation sequence $\mathbf{y}_{1:T}$ is assumed fixed, that is, the randomness considered is the randomness of the particle filter algorithm, rather than that of the state-space model. One concept of interest is the L^p convergence of estimated integrals of test functions to the true integral of the test function, that is, whether the sequence of approximations

$$\mathbb{E}_N(f(\mathbf{x}_k) \mid \mathbf{y}_{1:k}) = \sum_{i=1}^{N} w_k^{(i)} f(\mathbf{x}_k^{(i)}) \tag{3.16}$$

converges in L^p to the true expectation

$$\mathbb{E}(f(\mathbf{x}_k) \mid \mathbf{y}_{1:k}) = \int p(\mathbf{x}_k \mid \mathbf{y}_{1:k})\, f(\mathbf{x}_k)\, \mathrm{d}\mathbf{x}_k. \tag{3.17}$$

Note that each of the random variables $\mathbb{E}_N(f(\mathbf{x}_k) \mid \mathbf{y}_{1:T})$ for $N = 1, \ldots$ refers to a different realization of the particle filter algorithm.

The convergence results in, e.g., Crisan and Doucet (2002) assume bounded test functions f. Hu et al. (2011) (see also Hu et al., 2008) extended the results to show L^p convergence for unbounded test functions for the bootstrap filter, assuming a slight modification of the particle filter. More recently, this result has been extended to general importance distributions for L^4 convergence, assuming bounded importance weights (Mbalawata and Särkkä, 2014). The condition on bounded importance weights may be replaced by a condition on the moments of the importance weights. Various other convergence results exist in the literature, see, for example, Del Moral (1996); Crisan and Doucet (2000, 2002); Mbalawata (2014) and references therein. In the following, we present two theorems used in this thesis.

- Assuming that the observation model and dynamic model densities are bounded, and that the unnormalized importance weights $v_k^{(i)}$ are bounded from above as a function of $\mathbf{x}_{k-1}, \mathbf{x}_k$, as well as certain conditions for the resampling algorithm, for all $p \geq 2$ and bounded test functions f:

  $$\mathbb{E}\left[\left|\sum_{i=1}^{N} w_k^{(i)} f(\mathbf{x}_k^{(i)}) - \int p(\mathbf{x}_k \mid y_{1:k})\, f(\mathbf{x}_k)\, \mathrm{d}\mathbf{x}_k\right|^p\right] \leq c_k \frac{\|f\|^p}{N^{\frac{p}{2}}}, \tag{3.18}$$

  where c_k is a constant and $\|f\|$ is the supremum norm of f. Thus, the estimates $\mathbb{E}_N(f(\mathbf{x}_k) \mid \mathbf{y}_{1:T})$ converge in L^p to the correct value.

- (Mbalawata, 2014). The previous boundedness condition may be relaxed by assuming instead that $\mathbb{E}[|v_k^{(i)}|^p \mid \mathbf{x}_{k-1}] \leq c_k$ where the expectation is over the importance distribution q and the bound is uniform over $\mathbf{x}_{k-1}$. A particular value of p in the assumption then implies L^p-convergence with the same p.

4. Contributions of the Thesis

4.1 Overview

This thesis has three main research topics:

1. Static parameter estimation.

2. Importance distributions for particle filters.

3. Animal population size estimation.

Static parameter estimation is considered in Publication I, Publication II, and Publication V. Of these, Publication I and Publication V are concerned with applying the expectation–maximization (EM) algorithm to nonlinear state-space models using particle filters/smoothers or sigma-point filters/smoothers to evaluate the intermediate quantities required in the algorithm. Besides EM, we also consider direct likelihood maximization. In Publication II, particle Markov chain Monte Carlo methods are combined with the Rao–Blackwellized Monte Carlo data association algorithm to obtain a method for estimating static parameters in multiple target tracking problems.

Importance distributions of particle filters are considered in Publication III and Publication IV. In Publication IV, we consider the split-Gaussian importance distribution for sequential importance sampling/resampling particle filtering, fitting the split-Gaussian approximation to the optimal importance distributions in the manner proposed by Geweke (1989) for (non-sequential) importance sampling. In Publication III, we motivate the use of a split-Gaussian importance distribution in a certain class of models by convergence considerations. There, the parameters of the split-Gaussian distribution are selected based on the convergence consideration rather than by Geweke's fitting procedure.

Animal population size estimation is considered in Publication II. Abbas (2011) studied estimation of large carnivore populations in Finland based on databases of recorded observations. The problem was formulated as a multiple target tracking problem and RBMCDA was used to solve it. In Publication II we applied the suggested parameter estimation to bear observation data, thus extending the approach of Abbas (2011) to unknown static parameters.

4.2 Research Topic I: Parameter Estimation

When applied to state-space models, the auxiliary function computed in the E-step of the expectation–maximization algorithm contains expectations with respect to smoothing distributions (Eqs. 2.45–2.48). In the nonlinear case, the smoothing problem is not tractable in closed form. However, approximate EM algorithms may be obtained by using approximate smoothing algorithms. Roweis and Ghahramani (2001) used the extended Kalman filter and the corresponding smoother. More recently, Schön et al. (2011) (see also Schön et al., 2006) proposed the use of the particle filter and the reweighting particle smoother (cf. Section 3.4), and Gašperin and Juričić (2011) proposed the use of unscented transform based sigma-point filter and smoother. The use of sigma-point filters and smoothers within EM was also considered by Väänänen (2012) in his master's thesis. Chitralekha et al. (2009) also considered particle smoother, unscented Kalman smoother and extended Kalman smoother based EM. Their UKF-EM algorithm differs slightly from the one we considered in that they use Monte Carlo simulation for evaluating the expectations with respect to the smoothing distribution. In this thesis, we, following Väänänen (2012) as well as Gašperin and Juričić (2011), used the same sigma-point rules that are used in the smoother.

In Publication I, we presented and compared the EM algorithms for nonlinear dynamic systems with additive Gaussian noise (cf. Eq. 2.14, with unknown static parameters added) using either particle or sigma-point based smoothing approximations. We gave closed-form expressions for

Table 4.1. From Publication I. Correlations of the final EM estimates versus the direct ML estimates for all the trajectories in the UNGM example. Reprinted from Juho Kokkala, Arno Solin, and Simo Särkkä. Expectation Maximization Based Parameter Estimation by Sigma-Point and Particle Smoothing. In The 17th International Conference on Information Fusion (FUSION), with permission from ISIF.

Parameter	a	b	c	$\log Q$	$\log R$
Particle EM vs. ML	0.998	0.945	0.994	0.977	0.792
Sigma-point EM vs. ML	0.997	0.909	0.989	0.972	0.731

the solution of models that are linear-in-parameters, that is, of the form

$$\begin{aligned}\mathbf{x}_k &= \mathbf{f}(\mathbf{x}_{k-1}, \boldsymbol{\theta}) + \mathbf{q}_{k-1},\\ \mathbf{y}_k &= \mathbf{h}(\mathbf{y}_k, \boldsymbol{\theta}) + \mathbf{r}_k,\end{aligned} \tag{4.1}$$

with the nonlinear functions $\mathbf{f}$ and $\mathbf{h}$ decomposed as

$$\begin{aligned}\mathbf{f}(\mathbf{x}, \boldsymbol{\theta}) &= \mathbf{A}(\boldsymbol{\theta})\,\tilde{\mathbf{f}}(\mathbf{x}),\\ \mathbf{h}(\mathbf{x}, \boldsymbol{\theta}) &= \mathbf{H}(\boldsymbol{\theta})\,\tilde{\mathbf{h}}(\mathbf{x}),\end{aligned} \tag{4.2}$$

so that the parameters enter the model through the matrices $\mathbf{A}, \mathbf{H}$, the noise covariances, and the moments of the initial state distribution. We present the closed-form expressions for the optimal parameters (a detailed derivation is given in Publication V) and explain how to evaluate them using either particle smoother or sigma-point smoother results. Similar linear-in-parameters results were given by Schön et al. (2006). Their model formulation however differs from the one we considered.

In the numeric experiment section of Publication I, we compared the particle EM and sigma-point EM approaches in estimating the parameters of a univariate nonstationary growth model of the form

$$\begin{aligned}x_k &\sim \mathcal{N}(a\,x_{k-1} + b\frac{x_{k-1}}{1+x_{k-1}^2} + c\,\cos(1.2\,(k-1)), Q),\\ y_k &\sim \mathcal{N}(x_k/\sqrt{20}, R),\\ \boldsymbol{\theta} &= (a, b, c, Q, R)\end{aligned} \tag{4.3}$$

Table 4.1 shows the correlations of the EM estimates versus direct maximum likelihood estimates obtained by a sigma-point filter, along different datasets with different generated parameter values. The particle EM approach seems to match the maximum likelihood estimates better, even though the maximum likelihood estimates are computed with the sigma-point filter. In addition, we compared the parameter estimates to posterior distributions computed by particle MCMC, and considered noise covariance estimation in a multidimensional tracking example.

In Publication V, we switched attention to comparing different sigma-point rules, as well as comparing EM and direct (gradient-based) likelihood optimization. We compared the third order spherical-radial rule and higher-order rules based on McNamee and Stenger (1967), as well as Gauss–Hermite rules. In the numeric experiment section of Publication V, we considered a five-dimensional target tracking example with simulated data.

In Publication II, we considered full Bayesian parameter estimation, that is, posterior distribution computation, for static parameters of multiple target tracking models of the type used in RBMCDA (cf. Section 3.3.1). The key idea is to apply the particle MCMC methods with the Rao–Blackwellized particle filter that samples the data association sequences $c_{1:T}$. The result is an MCMC algorithm targeting the joint posterior of parameters and data-association sequences, $p(c_{1:T}, \boldsymbol{\theta} \mid \mathbf{y}_{1:T})$. We consider both particle marginal Metropolis–Hastings and particle Gibbs algorithms.

As an example, we consider targets moving in two dimensions according to the Ornstein–Uhlenbeck model

$$\mathrm{d}\mathbf{x} = \lambda(\mathbf{x}_0 - \mathbf{x})dt + \sqrt{q}\,\mathrm{d}\mathbf{W}, \tag{4.4}$$

where $\mathbf{x}_0$ is a fixed point around which the target moves. As this is a linear stochastic differential equation, the discrete-time dynamics are of the linear-Gaussian form (e.g. Jazwinski, 1970). The target states are then 4-

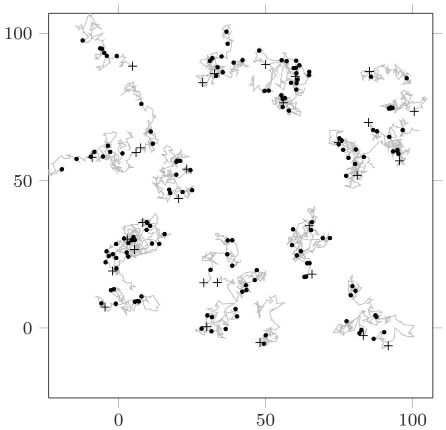

Figure 4.1. From Publication II. Simulated trajectories for the numeric experiment. Trajectories of targets are shown as gray lines, measurements as black dots and final target locations as black pluses.
Reprinted from Digital Signal Processing, Vol 47, J. Kokkala and S. Särkkä, Combining particle MCMC with Rao-Blackwellized Monte Carlo data association for parameter estimation in multiple target tracking, p. 89, Copyright (2015), with permission from Elsevier.

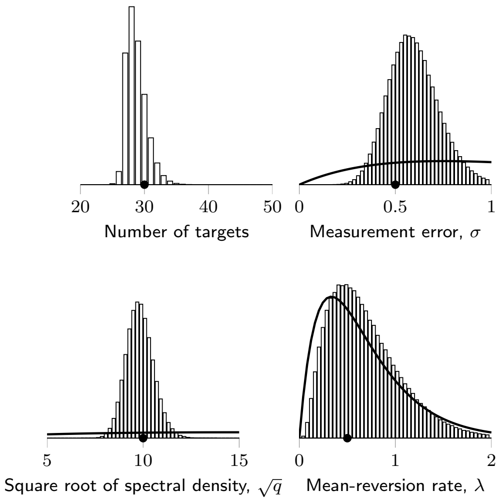

Figure 4.2. From Publication II. Posterior distributions of the parameters ($\sqrt{q}$,λ,σ) and the number of targets in the simulated scenario. Corresponding prior densities for parameters are shown as solid lines. Ground-truth values are shown as black dots on the axis.
Reprinted from Digital Signal Processing, Vol 47, J. Kokkala and S. Särkkä, Combining particle MCMC with Rao-Blackwellized Monte Carlo data association for parameter estimation in multiple target tracking, p. 89, Copyright (2015), with permission from Elsevier.

dimensional (both the location and the mean location in two-dimensional coordinates). The discrete-time transition matrix $\mathbf{A}_k$ and process noise covariance $\mathbf{Q}_k$ depend on the parameters λ and $\sqrt{q}$ and on the time difference of two consecutive observations. The observations were assumed to be simply the target location with additive Gaussian noise with variance σ^2 in both coordinates. The unknown parameters estimated were $\boldsymbol{\theta} = (\lambda, \sqrt{q}, \sigma^2)$.

In the numeric experiment with simulated data, we considered 30 targets. The fixed mean locations were drawn uniformly in the square $[0, 100] \times [0, 100]$, and target movements were simulated for a time period $[0, 1]$ from the Ornstein–Uhlenbeck model. Observations were generated at 150 random times during the interval. Parameters used were $\lambda = 0.5, \sqrt{q} = 10, \sigma = 0.5$. A visualization of the simulated target movements is shown in Figure 4.1. Posterior distributions obtained with our proposed algorithm are shown in Figure 4.2. In addition, we compared with results obtained by running the algorithm with fixed parameter values. The parameter estimation improved estimation performance in terms of mean OSPA (optimal subpattern assignment metric, Schuhmacher et al., 2008) of final target locations as well as posterior probability of the true number of targets. This mean OSPA was computed by computing the OSPA metric for each posterior sample of $(c_{1:T}, \boldsymbol{\theta})$ using the mean locations of the targets conditional on $(c_{1:T}, \boldsymbol{\theta})$, and then taking the average.

4.3 Research Topic II: Importance Distributions and Convergence for Particle Filters

Geweke (1989) proposed the split-Gaussian importance distribution for importance sampling. The idea is to take the Laplace approximation to the target distribution, and then scale the distribution differently along different directions from the mode. The distribution is parametrized by mode $\boldsymbol{\mu}$, a matrix $\mathbf{T}$ and scaling factors $\mathbf{q}, \mathbf{r}$. A draw from the split-Gaussian random variable $\mathbf{x}$ is obtained by drawing a standard multivariate Gaussian $\boldsymbol{\varepsilon}$, obtaining a split-Gaussian variable $\boldsymbol{\eta}$ by scaling $\boldsymbol{\varepsilon}$ along each component by $\mathbf{q}_i$ if the value is positive and by $\mathbf{r}_i$ if the value is negative, and finally performing a transformation $\mathbf{x} = \boldsymbol{\mu} + \mathbf{T}\,\boldsymbol{\eta}$. The density (expressed in terms of $\boldsymbol{\varepsilon}$) is

$$\mathrm{SN}(\mathbf{x} \mid \boldsymbol{\mu}, \mathbf{T}, \mathbf{q}, \mathbf{r}) = (2\,\pi)^{-n/2}\,\frac{1}{|\mathbf{T}|}\,\frac{1}{\prod_{i=1}^{n}(\mathbf{q}_i\,\mathbb{I}_{\boldsymbol{\varepsilon}_i \geq 0} + \mathbf{r}_i\,\mathbb{I}_{\boldsymbol{\varepsilon}_i < 0})}\,e^{-\frac{1}{2}\boldsymbol{\varepsilon}^\mathsf{T}\,\boldsymbol{\varepsilon}}. \tag{4.5}$$

In Publication IV, we used a continuous-density version of the split-Gaussian distribution, where the absolute value of ϵ_i is used in constructing $\boldsymbol{\eta}_i$, and the probabilities of the sign are selected so that the density is continuous at the mode (see, e.g., Villani and Larsson, 2006). The density of this continuous version is (expressed in terms of $\boldsymbol{\varepsilon}$)

$$\mathrm{SN}(\mathbf{x} \mid \boldsymbol{\mu}, \mathbf{T}, \mathbf{q}, \mathbf{r}) = \sqrt{\frac{2^n}{\pi^n}}\,\frac{1}{|\mathbf{T}|\,\prod(\mathbf{q}_i + \mathbf{r}_i)}\,e^{-\frac{1}{2}\,\boldsymbol{\varepsilon}^\mathsf{T}\,\boldsymbol{\varepsilon}}. \tag{4.6}$$

The scaling prodecure suggested by Geweke (1989) is the following. First, form the Laplace approximation $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Then, $\boldsymbol{\mu}$ is used as the mode of the split-Gaussian distribution. A decomposition of $\boldsymbol{\Sigma} = \mathbf{T}\,\mathbf{T}^\mathsf{T}$ is selected, and then the scaling factors $\mathbf{q}, \mathbf{r}$ are selected to match the decay of the target distribution along each column of $\mathbf{T}$. In one dimension, μ,σ are set as set as the Laplace approximation of the target density $p(x)$ and the target distribution is evaluated in a grid. For each grid point, a scaling factor is computed to match the rate of decline of the target distribution to the rate of decline of the Laplace approximation:

$$f(\delta) = |\delta|(2(\log p(\mu) - \log p(\mu + \delta\,\sigma)))^{-1/2}, \tag{4.7}$$

where $\mu + \delta\,\sigma$ are the grid points for various values of δ. The scaling factors corresponding to the widest tail are selected:

$$q = \sup_{\delta > 0} f(\delta), \qquad r = \sup_{\delta < 0} f(\delta). \tag{4.8}$$

In multiple dimensions, the scaling procedure is similar, but performed along each direction defined by columns of $\mathbf{T}$.

In Publication IV, we proposed the use of the split-Gaussian distribution in particle filters, using the aforementioned fitting procedure to form an approximation to the optimal importance distribution. We compared the split-Gaussian importance distribution against the plain Laplace appoximation as well as other Gaussian importance distributions using a one-dimensional test model. Based on the average number of resamplings required when using adaptive resampling with an effective sample size threshold, the split-Gaussian importance distribution had the best performance in our experiment. Furthermore, we also proposed a variant of the fitting procedure for cases where the Laplace approximation is not available. There, the scaling is performed using the mode of another Gaussian approximation. If a higher value of the target density is found in the grid, the mode of the importance distribution is switched. We tested this in a multi-dimensional tracking example.

In Publication III, we also used a split-Gaussian importance distribution. In contrast to Publication IV, in Publication III the split-Gaussian importance distribution was selected based on convergence considerations, rather than using the fitting procedure. We considered the following state-space model

$$\begin{aligned} \mathrm{x}_k &\sim \mathcal{N}(\mathbf{A}_{k-1}\,\mathrm{x}_{k-1},\ \mathbf{Q}_{k-1}), \\ y_k &\sim \mathrm{Poisson}(\exp(\boldsymbol{\beta}_k^\mathsf{T}\,\mathrm{x}_k)), \end{aligned} \tag{4.9}$$

which may be interpreted as a time-varying Poisson regression, with time-varying coefficients x_k and covariates β_k. Since the observation contains information about the state only via the scalar $s_k = \boldsymbol{\beta}_k^\mathsf{T}\,\mathrm{x}_k$, we considered importance distributions of such form that s_k is drawn from an importance distribution and then x_k is drawn from the dynamic model density conditional on s_k. We first showed that the Laplace approximation based importance distribution leads to unbounded importance weights, as well as infinite moments. Therefore, the criteria of the convergence theorems are not satisfied. Then, we considered a split-Gaussian importance distribution where the Laplace approximation was used in the $s_k \geq m_k$ half, where m_k is the mode of the Laplace approximation, while a widened density was used in the $s_k < m_k$ half. This guarantees bounded importance weights and thus convergence of the particle filter, while still utilizing information contained in the observation $\mathbf{y}_k$.

4.4 Research Topic III: Animal Population Size Estimation

Abbas (2011) considered animal population size estimation using the RBM-CDA algorithm. The Natural Resources Institute Finland (formerly the Finnish Game and Fisheries Research Institute) estimates large carnivore populations based on a database of field sightings and direct observations. To obtain an estimate of the size of the population, or the number of families, manual analysis by experts is required to determine which observations are reliable and which observations are from the same family of animals. Abbas (2011) reformulates the problem as a multiple target tracking problem, the byproduct of which is a posterior probability distribution on the number of targets.

In Publication II, we applied the developed parameter estimation algorithm to estimating bear population based on the observation database. In this experiment, we used data from the year 2013 from one game management district (Kaakkois-Suomen riistanhoitopiiri). The dynamic model and measurement model were as in the simulated experiment. Posterior distributions on parameters and number of targets are shown in Figure 4.3. The method converges according to the potential scale reduction factor criterion (e.g. Gelman et al., 2013) and produces reasonable looking posterior distributions. However, compared to the expert estimates, our analysis overestimates the number of targets. The experts may have better information about which observations are unreliable. Furthermore, our prior distributions were possibly too noninformative compared to what can be inferred from data within only one game management district.

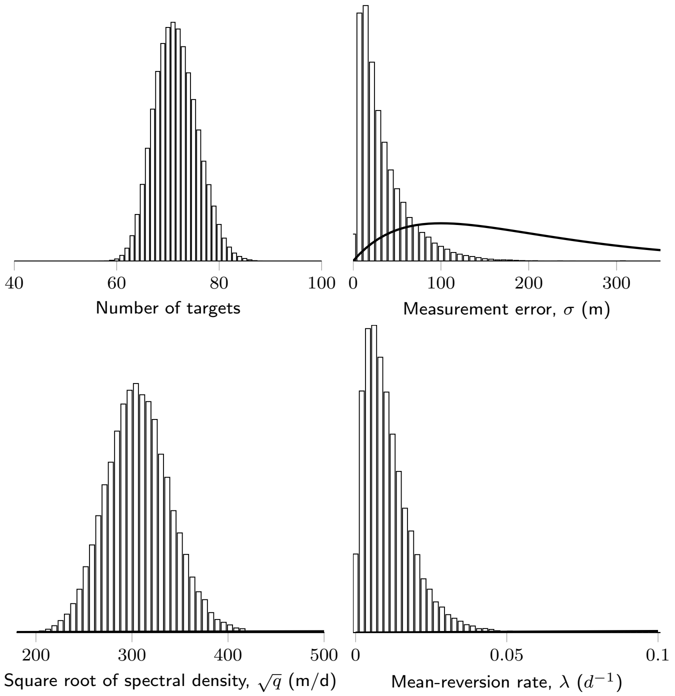

Figure 4.3. From Publication II. Posterior distributions of the parameters ($\sqrt{q}$,λ,σ) and the number of brown bear families in the Kaakkois-Suomi district (year 2013). Corresponding prior densities for parameters are shown as solid lines. Reprinted from Digital Signal Processing, Vol 47, J. Kokkala and S. Särkkä, Combining particle MCMC with Rao-Blackwellized Monte Carlo data association for parameter estimation in multiple target tracking, p. 89, Copyright (2015), with permission from Elsevier.

5. Discussion

The aim of this thesis was to study and improve inference algorithms for dynamic systems, by designing importance distributions for particle filters as well as by developing and comparing methods for static parameter estimation. Furthermore, the research was applied to animal population size estimation.

The animal population size estimation was based on the multiple target tracking formulation originally presented by Abbas (2011) in his Master's thesis. Abbas (2011) used Rao-Blackwellized Monte Carlo data association (RBMCDA) to track the animals, and subsequently also to obtain a probability distribution of the population size, based on a database of recorded observations. In this thesis, the aim was to extend the method to learn static parameters of the tracking models based on the observations. For this purpose, in Publication II, we developed a full Bayesian inference algorithm for multiple target tracking models with unknown static parameters.

The aforementioned algorithm is based on combining particle MCMC with RBMCDA. Based on the experiments documented in Publication II, the method in principle works for this type of data. However, the resulting population estimates with real data were quite different from the expert estimates. This may be due to, for example, misspecified dynamic models or the fact that observations from only one game management district were used, while in reality the targets may move away from the area of the district.

Ideally, improving the animal population estimation methodology, either by introducing more realistic dynamic models or by taking into account more expert information in the prior distributions, would be performed by iterative model development and checking in close collaboration with subject matter experts. To make this kind of interactive work feasible, as well as to be able to use observations from the entire country over multiple years, computational improvements into the algorithm are required. One possibility would be to use particle Gibbs with ancestor sampling (Lindsten et al., 2014) instead of the original particle Gibbs algorithm. As the model used in our algorithm is conditionally linear-Gaussian, one would then use the ancestor sampling step presented for Rao–Blackwellized particle filters by Svensson et al. (2014).

When full Bayesian posterior computation, for example using particle MCMC, is not feasible due to computational constraints, point estimates may be sought instead. Another reason for using point estimates is that one may simply desire to use some reasonable values for static parameters, rather than completely represent the uncertain knowledge about the parameters. One reasonable point estimate is the maximum likelihood estimate, that is, the parameter values that maximize the likelihood of the observed data. In the state-space model context, the conditional independence structure enables evaluating both the likelihood and its gradient by a recursion computed along the filtering recursion. These may then be used in any gradient-based optimization algorithm. Besides directly maximizing the likelihood, expectation–maximization (EM) may be used.

In Publication I and Publication V we discussed and compared static parameter estimation in nonlinear systems with additive Gaussian noise, using either particle filters and smoothers or sigma-point filters and smoothers, the latter with various different cubature rules. We discussed both direct gradient-based optimization and the EM algorithm. EM is especially useful in a certain type of linear-in-parameters models, for which we gave closed-form expressions for the maximization-step of the EM algorithm.

Recently, Lindsten (2013) suggested a further improvement to the particle smoother EM, based on the conditional particle filter originally used in particle Gibbs. The idea is to combine stochastic approximation EM (Delyon et al., 1999) and the conditional particle filter. This enables using state estimates obtained in one EM iteration to improve performance of the smoother in the next EM iteration, in contrast to the basic particle EM where the only information carried to the next EM iteration is the new estimate for parameters. An interesting future research topic would be to investigate whether the sigma-point EM algorithm could also be developed to, in some sense, incorporate information learned in previous EM iterations into the smoother. For example, this could be based on ongoing adaptation of the sigma-point sets (see Duník et al., 2012, for a discussion of adaptive selection of the sigma-points).

Development of importance distributions for particle filtering was considered in Publication III and Publication IV. In Publication IV, we considered the split-Gaussian importance distribution fit to the optimal importance distribution using the procedure suggested by Geweke (1989). In Publication III, we motivated the use of the split-Gaussian importance distribution for a Poisson regression model based on requirements set by the convergence theory of particle filters. A possible future research topic would be to generalize this idea to other state-space models beyond the Poisson regression model.

References

M. Abbas. Statistical estimation of wild animal population in Finland: a multiple target tracking approach. Master's thesis, School of Science, Aalto University, Finland, 2011.

B. D. Anderson and J. B. Moore. *Optimal Filtering*. Prentice-Hall, Inc., 1979. Reprinted by Dover, 2005.

C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010. doi: 10.1111/j.1467-9868.2009.00736.x.

I. Arasaratnam and S. Haykin. Cubature Kalman filters. *IEEE Transactions on Automatic Control*, 54(6):1254–1269, 2009. doi: 10.1109/TAC.2009.2019800.

Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, 2004.

O. Cappé, E. Moulines, and T. Ryden. *Inference in Hidden Markov Models*. Springer Series in Statistics. Springer Science+Business Media, Inc., 2005.

S. Challa, M. Morelande, D. Musicki, and R. Evans. *Fundamentals of Object Tracking*. Cambridge University Press, 2011.

R. Chen and J. S. Liu. Mixture Kalman filters. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(3):493–508, 2000. doi: 10.1111/1467-9868.00246.

S. Chitralekha, J. Prakash, H. Raghavan, R. B. Gopaluni, and S. Shah. Comparison of expectation-maximization based parameter estimation using particle filter, unscented and extended Kalman filtering techniques. In *15th IFAC Symposium on System Identification (SYSID)*, pages 804–809, 2009. doi: 10.3182/20090706-3-FR-2004.00133.

D. Crisan and A. Doucet. Convergence of sequential Monte Carlo methods. Technical Report Technical Report CUEDIF-INFENGrrR38, Signal Processing Group, Department of Engineering, University of Cambridge, 2000.

D. Crisan and A. Doucet. A survey of convergence results on particle filtering methods for practitioners. *IEEE Transactions on Signal Processing*, 50(3): 736–746, 2002. doi: 10.1109/78.984773.

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

B. Delyon, M. Lavielle, and E. Moulines. Convergence of a stochastic approximation version of the EM algorithm. *The Annals of Statistics*, 27(1):94–128, 1999. doi: 10.1214/aos/1018031103.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–38, 1977.

A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000. doi: 10.1023/A:1008935410038.

A. Doucet, N. de Freitas, and N. Gordon. An introduction to sequential Monte Carlo methods. In A. Doucet, N. de Freitas, and N. Gordon, editors, *Sequential Monte Carlo Methods in Practice*, Statistics for Engineering and Information Science, pages 3–14. Springer New York, 2001.

P. Druilhet and J.-M. Marin. Invariant {HPD} credible sets and {MAP} estimators. *Bayesian Analysis*, 2(4):681–691, 2007. doi: 10.1214/07-BA227.

J. Duník, M. Šimandl, and O. Straka. Unscented Kalman filter: aspects and adaptive setting of scaling parameter. *IEEE Transactions on Automatic Control*, 57(9):2411–2416, 2012. doi: 10.1109/TAC.2012.2188424.

M. Gašperin and Đ. Juričić. Application of unscented transformation in nonlinear system identification. In *Proceedings of the 18th IFAC World Congress*, pages 4428–4433, 2011. doi: 10.3182/20110828-6-IT-1002.03024.

A. Gelb, editor. *Applied Optimal Estimation*. The MIT Press, 1974.

A. E. Gelfand and A. F. Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association*, 85(410): 398–409, 1990. doi: 10.1080/01621459.1990.10476213.

A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin. *Bayesian Data Analysis*. Chapman & Hall/CRC Press, third edition, 2013.

S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):721–741, 1984. doi: 10.1109/TPAMI.1984.4767596.

J. Geweke. Bayesian inference in econometric models using Monte Carlo integration. *Econometrica: Journal of the Econometric Society*, 57(6):1317–1339, 1989. doi: 10.2307/1913710.

T. Glad and L. Ljung. *Control Theory: Multivariable and Nonlinear Methods*. Taylor & Francis, 2000.

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, 2004. doi: 10.1198/016214504000000151O.

N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F (Radar and Signal Processing)*, 140(2):107–113, 1993. doi: 10.1049/ip-f-2.1993.0015.

N. K. Gupta and R. K. Mehra. Computational aspects of maximum likelihood estimation and reduction in sensitivity function calculations. *IEEE Transactions on Automatic Control*, 19(6):774–783, 1974. doi: 10.1109/TAC.1974.1100714.

W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970. doi: 10.1093/biomet/57.1.97.

Y. C. Ho and R. C. K. Lee. A Bayesian approach to problems in stochastic estimation and control. *IEEE Transactions on Automatic Control*, 9(4):333–339, 1964. doi: 10.1109/TAC.1964.1105763.

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. doi: 10.1109/TSP.2007.911295.

X.-L. Hu, T. B. Schön, and L. Ljung. A general convergence result for particle filtering. *IEEE Transactions on Signal Processing*, 59(7):3424–3429, 2011. doi: 10.1109/TSP.2011.2135349.

M. Hürzeler and H. R. Künsch. Monte Carlo approximations for general state-space models. *Journal of Computational and Graphical Statistics*, 7(2):175–193, 1998. doi: 10.1080/10618600.1998.10474769.

K. Ito and K. Xiong. Gaussian filters for nonlinear filtering problems. *IEEE Transactions on Automatic Control*, 45(5):910–927, 2000. doi: 10.1109/9.855552.

J. Jacod and P. E. Protter. *Probability Essentials*. Springer Science & Business Media, 2003.

A. H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, 1970. Reprinted by Dover, 2007.

S. Julier, J. Uhlmann, and H. F. Durrant-Whyte. A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Transactions on Automatic Control*, 45(3):477, 2000. doi: 10.1109/9.847726.

S. J. Julier. The scaled unscented transformation. In *Proceedings of the 2002 American Control Conference*, volume 6, pages 4555–4559, 2002. doi: 10.1109/ACC.2002.1025369.

S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte. A new approach for filtering nonlinear systems. In *Proceedings of the 1995 American Control Conference*, volume 3, pages 1682–1632, 1995. doi: 10.1109/ACC.1995.529783.

R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME, Journal of Basic Engineering*, 82(1):35–45, 1960. doi: doi:10.1115/1.3662552.

D. E. Kirk. *Optimal Control Theory*. Prentice-Hall, Inc., 1970. Reprinted by Dover, 2004.

G. Kitagawa. Non-Gaussian state—space modeling of nonstationary time series. *Journal of the American Statistical Association*, 82(400):1032–1041, 1987. doi: 10.1080/01621459.1987.10478534.

H. J. Kushner. Approximations to optimal nonlinear filters. *IEEE Transactions on Automatic Control*, 12(5):546–556, 1967. doi: 10.1109/TAC.1967.1098671.

T. Li, M. Bolic, and P. M. Djuric. Resampling methods for particle filtering: Classification, implementation, and strategies. *IEEE Signal Processing Magazine*, 32(3):70–86, 2015. doi: 10.1109/MSP.2014.2330626.

F. Lindsten. An efficient stochastic approximation EM algorithm using conditional particle filters. In *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6274–6278, 2013. doi: 10.1109/ICASSP.2013.6638872.

F. Lindsten, M. I. Jordan, and T. B. Schön. Particle Gibbs with ancestor sampling. *Journal of Machine Learning Research*, 15(1):2145–2184, 2014.

J. S. Liu and R. Chen. Blind deconvolution via sequential imputations. *Journal of the American Statistical Association*, 90(430):567–576, 1995. doi: 10.1080/01621459.1995.10476549.

I. S. Mbalawata. *Adaptive Markov chain Monte Carlo and Bayesian filtering for state space models*. PhD thesis, Lappeenranta University of Technology, 2014.

I. S. Mbalawata and S. Särkkä. On the L^4 convergence of particle filters with general importance distributions. In *Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8048–8052, 2014. doi: 10.1109/ICASSP.2014.6855168.

J. McNamee and F. Stenger. Construction of fully symmetric numerical integration formulas. *Numerische Mathematik*, 10(4):327–344, 1967. doi: 10.1007/BF02162032.

H. E. Rauch, F. Tung, and C. T. Striebel. Maximum likelihood estimates of linear dynamic systems. *AIAA Journal*, 3(8):1445–1450, 1965. doi: 10.2514/3.3166.

B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman filter: Particle filters for tracking applications*. Artech House, 2004.

S. Roweis and Z. Ghahramani. Learning nonlinear dynamical systems using the expectation–maximization algorithm. In S. Haykin, editor, *Kalman Filtering and Neural Networks*, chapter 6, pages 175–220. Wiley-Interscience, 2001.

S. Särkkä. *Bayesian Filtering and Smoothing*, volume 3 of *Institute of Mathematical Statistics Textbooks*. Cambridge University Press, 2013.

S. Särkkä and J. Hartikainen. On Gaussian optimal smoothing of non-linear state space models. *IEEE Transactions on Automatic Control*, 55(8):1938–1941, 2010. doi: 10.1109/TAC.2010.2050017.

S. Särkkä, A. Vehtari, and J. Lampinen. Rao-Blackwellized particle filter for multiple target tracking. *Information Fusion*, 8(1):2–15, 2007. doi: doi:10.1016/j.inffus.2005.09.009.

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, 2005. doi: 10.1109/TSP.2005.849151.

T. B. Schön, A. Wills, and B. Ninness. Maximum likelihood nonlinear system estimation. In *14th IFAC Symposium on System Identification (SYSID)*, pages 1003–1008, 2006. doi: 10.3182/20060329-3-AU-2901.00160.

T. B. Schön, A. Wills, and B. Ninness. System identification of nonlinear state-space models. *Automatica*, 47(1):39–49, 2011. doi: 10.1016/j.automatica.2010.10.013.

D. Schuhmacher, B.-T. Vo, and B.-N. Vo. A consistent metric for performance evaluation of multi-object filters. *IEEE Transactions on Signal Processing,*, 56 (8):3447–3457, 2008. doi: 10.1109/TSP.2008.920469.

M. Segal and E. Weinstein. A new method for evaluating the log-likelihood gradient, the Hessian, and the Fisher information matrix for linear dynamic systems. *IEEE Transactions on Information Theory*, 35(3):682–687, 1989. doi: 10.1109/18.30995.

R. H. Shumway and D. S. Stoffer. An approach to time series smoothing and forecasting using the EM algorithm. *Journal of Time Series Analysis*, 3(4): 253–264, 1982. doi: 10.1111/j.1467-9892.1982.tb00349.x.

A. Solin. Cubature integration methods in non-linear Kalman filtering and smoothing. Bachelor's thesis, Faculty of Information and Natural Sciences, Aalto University, Finland, 2010.

R. F. Stengel. *Stochastic Optimal Control; Theory and Application*. John Wiley and Sons, 1986. Reprinted by Dover as 'Optimal Control and Estimation', 1994.

G. Storvik. Particle filters for state-space models with the presence of unknown static parameters. *IEEE Transactions on Signal Processing*, 50(2):281–289, 2002. doi: 10.1109/78.978383.

A. Svensson, T. Schön, and F. Lindsten. Identification of jump Markov linear models using particle filters. In *IEEE 53rd Annual Conference on Decision and Control (CDC)*, pages 6504–6509, 2014. doi: 10.1109/CDC.2014.7040409.

A. Svensson, T. B. Schön, and M. Kok. Nonlinear state space smoothing using the conditional particle filter. In *The 17th IFAC Symposium on System Identification (SYSID)*, IFAC-PapersOnline 48-28, pages 975–980, 2015. doi: doi:10.1016/j.ifacol.2015.12.257.

V. Väänänen. Gaussian filtering and smoothing based parameter estimation in nonlinear models for sequential data. Master's thesis, School of Electrical Engineering, Aalto University, Finland, 2012.

R. van der Merwe and E. Wan. Sigma-point Kalman filters for probabilistic inference in dynamic state-space models. In *Proceedings of the Workshop on Advances in Machine Learning*, 2003.

R. van der Merwe, A. Doucet, N. de Freitas, and E. Wan. The unscented particle filter. In *Advances in Neural Information Processing Systems 13 (NIPS)*, pages 584–590, 2000.

M. Villani and R. Larsson. The multivariate split normal distribution and asymmetric principal components analysis. *Communications in Statistics - Theory and Methods*, 35(6):1123–1140, 2006. doi: 10.1080/03610920600672252.

Y. Wu, D. Hu, M. Wu, and X. Hu. A numerical-integration perspective on Gaussian filters. *IEEE Transactions on Signal Processing*, 54(8):2910–2921, 2006. doi: 10.1109/TSP.2006.875389.

Errata

Publication I

On page 2, Eq. (5), the 4th equation should be

$$\mathbf{G}_k = \mathbf{D}_{k+1} \left[\mathbf{P}_{k+1|k}^{-1}\right]$$

and the 6th equation should be

$$\mathbf{P}_{k|T} = \mathbf{P}_{k|k} + \mathbf{G}_k \left(\mathbf{P}_{k+1|T} - \mathbf{P}_{k+1|k}\right) \mathbf{G}_k^{\mathsf{T}}.$$

These errors occur only in the text, not in the implementation of the experiments.

Publication III

There was a programming mistake in Newton's method for fitting the Laplace approximation. Fig. 1 on p. 798 should be replaced with Figure E.1 of this Errata.

In addition, the number of particles was varied from 100 to $20,000$ instead of from $1,000$ to $20,000$.

The conclusion mostly stands, that is, the Laplace approximation based particle filter tends to underestimate the test quantity compared to the batch MCMC estimate, althought there is an outlier in the split-Gaussian results at around $16,000$ particles.

Publication IV

In Section 3, p. 485, the sentence *Then, a standard multivariate Gaussian ε is drawn and each component of ε is scaled by the scaling factors* $\mathrm{q}_i, \mathrm{r}_i$ *to*

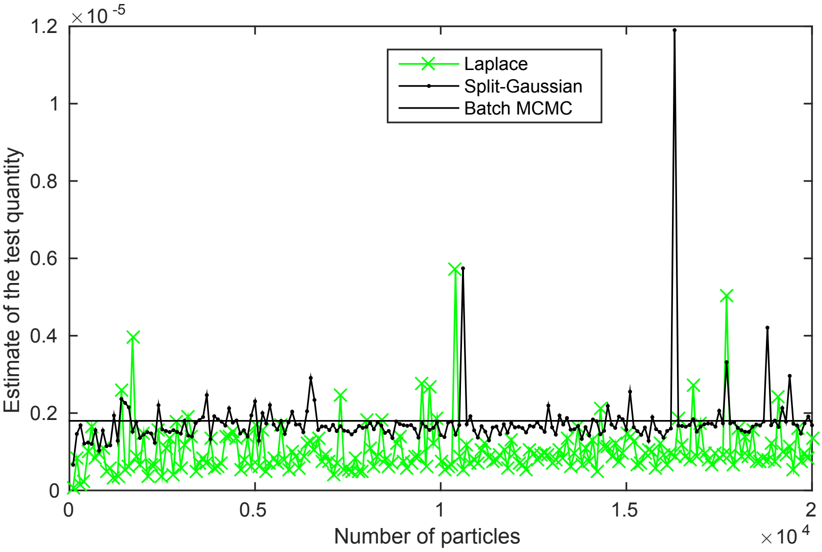

Figure E.1. Fourth central moment of $\exp(-\beta_2^\mathsf{T} x_2)$ estimated with Laplace and split-Gaussian particle filters, varying the number of particles from 100 to 20, 000. The black horizontal line is the MCMC estimate of the same quantity. Note that the MCMC estimate is shown only for comparison and it is obtained from a single run, that is, it is not a function of the number of particles.

form a split-Gaussian random variable η is unclear. For each component, one randomly decides whether i) the component is positive and $\mathbf{q}_i$ is used or ii) the component is negative and $\mathbf{r}_i$ is used. This is correctly explained in the pseudocode in Algorithm 1 (p. 486).

Publication V

On page 13, Eq. (58), the last term on the second line is missing a factor $\mathbf{Q}^{-1}$ inside the trace, i.e., it should read

$$\frac{\mathrm{d}\mathcal{Q}}{\mathrm{d}\mathbf{A}} \operatorname{tr}(\mathbf{Q}^{-1}\, \mathbf{A}\, \mathbf{\Phi}\, \mathbf{A}^\mathsf{T})$$

instead of

$$\frac{\mathrm{d}\mathcal{Q}}{\mathrm{d}\mathbf{A}} \operatorname{tr}(\mathbf{A}\, \mathbf{\Phi}\, \mathbf{A}^\mathsf{T}).$$

Aalto-DD 30/2016

ISBN 978-952-60-6663-9 (printed)
ISBN 978-952-60-6664-6 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)

Aalto University
School of Science
Department of Neuroscience and Biomedical Engineering /
Department of Computer Science
www.aalto.fi