Bayesian Approaches to Tracking, Sensor Fusion and Intent Prediction

Jiaming Liang

Department of Engineering
University of Cambridge

This dissertation is submitted for the degree of
Doctor of Philosophy

Girton College October 2019

To my wife Congying
and my family

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Jiaming Liang
October 2019

Acknowledgements

First of all, I would like to thank my supervisor, Prof. Simon Godsill, for patiently nurturing and advising me over the past four years, and most importantly, for 'taking the bite' (accepting me as a PhD student) at the very beginning. It is impossible to overstate his influence on me, as it is well beyond the work presented here.

I would then like to thank my colleagues in the SigProC Lab at the Cambridge University Engineering Department - thank-you all for the knowledge, discussion and humor. In particular, I would like to thank Bashar, Chenhao, Herbert, Jichi, Qing and Runze, for their friendship, helpful suggestions and companion during those late nights in the department. I would also like to thank my friends - in China, Cambridge and elsewhere - for their advice and support.

This journey would not have been possible without the support and love from my family. Thank you, mom and dad, for everything you have done for me throughout my life and for showing me how to live a life full of integrity, acceptance and love.

Finally, to my beloved wife, Congying, who has been through this PhD with me - I cannot imagine what this journey will look like without your love, companion and constant support. This thesis is dedicated to *you*.

Abstract

This thesis presents work on the development of model-based Bayesian approaches to object tracking and intent prediction. Successful navigation/positioning applications rely fundamentally on the choice of appropriate dynamic model and the design of effective tracking algorithms capable of maximising the use of the structure of the dynamic system and the information from sensors. While the tracking problem with frequent and accurate position data has been well studied, we push back the frontiers of current technology where an object can undergo fast manoeuvres and position fixes are limited. On the other hand, intent prediction techniques which extract higher level information such as the intended destination of a moving object can be designed, given the ability to perform successful tracking. Such techniques can play important roles in various application areas, including traffic monitoring, intelligent human computer interaction systems and autonomous route planning.

In the first part of this thesis Bayesian tracking methods are designed based on a standard fix-rate setting in which the dynamic system is formulated into a Markovian state space form. We show that the combination of an intrinsic coordinate dynamic model and sensors in the object's body frame leads to novel state space models according to which efficient proposal kernels can be designed and implemented by the sequential Monte Carlo (SMC) methods. Also, sequential Markov chain Monte Carlo schemes are considered for the first time to tackle the sequential batch inference problems due to the presence of infrequent position data. Performance evaluation on both synthetic and real-world data shows that the proposed algorithms are superior to simpler particle filters, implying that they can be favourable alternatives to tracking problems with inertial sensors.

The modelling assumption that leads to Markovian state space models can be restrictive for real-world systems as it stipulates that the state sequence has to be synchronised with the observations. In the second major part of this thesis we relax this assumption and work with a more natural class of models, termed variable rate models. We generalise the existing variable rate intrinsic model to incorporate acceleration, speed, distance and position data and introduce new variable rate particle filtering methods tailored to the derived model to accommodate multi-sensor multi-rate tracking scenarios. The proposed algorithms can achieve substantial improvements in terms of tracking accuracy and robustness over a bootstrap variable rate particle filter. Moreover, full Bayesian inference schemes for the learning of both the hidden state and system parameters are presented, with numerical results illustrating their effectiveness.

The last part of the thesis is about designing efficient intent prediction algorithms within a Bayesian framework. A pseudo-observation based approach to the incorporation of destination knowledge is introduced, making the mathematics of the dynamical model and the observation process consistent with the Markov state process. Based on the new interpretation, two algorithms are proposed to sequentially estimate the probability of all possible endpoints. Whilst the synthetic maritime surveillance data demonstrate that the proposed methods can achieve comparable prediction performance with reduced computational cost in comparison to the existing bridging distribution based methods, the results on an extensive freehand pointing database, which contains 95 three-dimensional pointing trajectories, show that the new algorithms can outperform other state-of-the-art techniques. Some sensitivity tests are also performed, confirming the good robustness of the introduced methods against model mismatches.

Contents

List of figures

List of tables

Glossary

FFBS	Forward-filtering-backward-sampling
IS	Importance sampling
KF	Kalman filter
MAP	Maximum *a posteriori*
MCMC	Markov chain Monte Carlo
MH	Metropolis-Hastings
PED	Prediction error decomposition
PG	Particle Gibbs
RBPF	Rao-Blackwellised particle filter
RM	Resample-move
SDE	Stochastic differential equation
SIR	Sequential importance sampling and resampling
SIS	Sequential importance sampling
SMC	Sequential Monte Carlo
SMCMC	Sequential Markov chain Monte Carlo
SS	Simulation smoother
VRPF	Variable rate particle filter

Chapter 1

Introduction

For many problems in scientific fields such as statistical signal processing, time series analysis, biostatistics and econometrics, one of the most important tasks is to obtain an accurate estimate of some hidden quantity (state) based on observations from various sources. This is a nontrivial task since in addition to accounting for the randomness which the state of interest exhibits in its evolution process, we also need to deal with the data that is typically noisy, incomplete and/or limited in quantities.

In many cases, we may have already obtained some level of prior knowledge regarding the evolving pattern of the quantity from fields such as physics. For example, it is well known that the motion of a physical object can be described by certain mathematical models. This information, along with a probabilistic model stating the relationship between the data and the underlying quantity, may be represented using probabilistic state space models. While the statistical modelling of the system embodies our views about how the data is generated, performing reasoning about the properties of the underlying quantity is the next crucial step before making any decisions and conclusions given the data at hand.

Among a variety of statistical methods, the Bayesian approach provides a theoretically sound framework for us to perform statistical modelling and inference. It deals with the associated uncertainties in a quantitative way and thus allows informed assessments to be made. The beauty of Bayesian methods also lies in the fact that it provides a natural solution to the sequential update of our beliefs when new data arrives over time. This feature is well suited to the modeling of data that arrives sequentially and hence it underpins many sequential learning methodologies. In this thesis, we aim to tackle real-world problems, including object tracking and intent prediction, with model-based Bayesian approaches.

The ability to accurately track an object of interest is important in many application domains. A successful tracking algorithm relies heavily on the choice of a good model for object dynamics and an effective inference algorithm. This is because the former condition imposes an informed prior while the latter resolves the uncertainties in a better way. Therefore, these two essential ingredients are both sought when designing tracking algorithms in this thesis. Due to the presence of complicated mathematical models and/or indirect measurements, analytically tractable models are often unavailable in tracking applications, preventing the exact Bayesian inference. In such cases, we resort to methods which can provide reliable approximations of posterior distributions, such as those based on Monte Carlo theory. In particular, we are interested in developing Bayesian approaches for challenging tracking scenarios, based on Sequential Monte Carlo (SMC) methods which have proved to be very powerful when dealing with non-linear and/or non-Gaussian systems and are well adapted to sequential inference tasks. Furthermore, building upon the accurate tracking of a moving object, Bayesian approaches can also be applied to inference problems in which knowing the intended destination of an object is of concern. Many application areas, ranging from human computer interaction, smart assistance systems and surveillance, can benefit from correct and fast estimations of the object intent. Motivated by these potential applications, this thesis also addresses the problem of online intent prediction in object tracking.

1.1 Thesis Outline and Contributions

In the year of writing this thesis the bootstrap filter of [Gordon et al., 1993], i.e. the first working instance of modern SMC methods which will be used extensively in this thesis, has just celebrated its twenty-fifth anniversary. Chapter 2 starts with a brief introduction to the Bayesian inference framework, which is followed by a condensed review of stochastic simulation methods such as importance sampling (IS) and Markov chain Monte Carlo (MCMC). It is organised in this way because they are the building blocks of the SMC approaches. After that, we give a detailed review of the SMC methods, covering the sequential learning schemes based on the ideas of IS and MCMC, respectively and the relevant improvement strategies over the past 25 years.

Chapter 3 studies object tracking problems from a fixed rate perspective where the dynamic system is formulated using Markovian state space models. We adopt an intrinsic coordinate dynamic model for the motion, since it offers a promising solution to address the problem of fusing measurements from sensors that are physically in the intrinsic frame of an object. It is shown that the combination of inertial measurements with the intrinsic model enables the design of effective proposal kernels, which can be incorporated effectively into IS-based SMC filters and sequential MCMC. The performance of the proposed methods are assessed using synthetic data, track cycling data as well as real vehicle data, and the results indicate their strong potential for navigation/tracking problems where inertial sensors are involved and position data is limited.

Although the fixed rate assumption has been widely made in many practical problems, it is not necessarily a realistic description of the actual system dynamics in that hidden states are forced to arrive synchronously with measurements. The stringency of this assumption can be relaxed by allowing the hidden state to evolve continuously in time and independently from the observations. The resulting models are usually referred to as “variable rate models”.

Chapter 4 is devoted to the development of particle filtering methods tailored to variable rate manoeuvre models. The existing intrinsic coordinate model is generalised into a compatible state space form, allowing straightforward incorporation of inertial measurements. We then show how variable rate particle filters with improved performance in comparison to its bootstrap equivalent may be designed in the presence of asynchronous measurements, based on the ideas of sampling disturbances of the dynamic system and state rejuvenation. The proposed methods are tested on both the synthetic data and benchmark trajectory data, giving a comprehensive evaluation as to their performance. Furthermore, sampling-based schemes are proposed to perform full system inference for the considered model and numerical examples are provided to demonstrate their effectiveness in learning both the hidden state and system parameters.

While the previous chapters focus on building advanced Bayesian tracking algorithms from different modelling perspectives, Chapter 5 presents a Bayesian framework for predicting an object’s intended destination in the context of object tracking. The existing bridging distribution based intent prediction methods impose the assumption that the initial state is independent of the terminal state and thus is inconsistent with the Markov state process. To provide a consistent formulation, we propose to model the destination point as a ‘pseudo-observation’ instead of an augmented state. It will be shown that this new interpretation not only preserves the underlying Markovian structure but also leads to two new algorithms that are computationally more efficient than the original bridging distribution based approaches.

Chapter 6 summarises the arguments made and methods introduced in the previous chapters. Conclusions are drawn and some suggestions for future research are provided.

The main contributions of this thesis are summarised as follows:

- The presentation of a fixed rate intrinsic coordinate model which allows for the fusion of inertial (accelerometer, gyroscope, speedometer and distance meter) and asynchronous position measurements. (Chapter 3)
- The design of section-wise forward-filtering-backward-sampling based proposal kernels as well as sequential locally optimal proposal kernels, which leads to improved tracking performance when used with sequential Monte Carlo methods (including particle filters and sequential MCMC). (Chapter 3)
- A generalised variable rate intrinsic coordinate model and corresponding particle filtering algorithms for asynchronous multisensor object tracking, leading to better performance and robustness in some challenging tracking scenarios as compared with the standard variable rate particle filter. (Chapter 4)
- Effective proposal kernels based on simulation smoother are designed to accommodate conditionally deterministic models. (Chapter 4)
- A novel Bayesian approach to intent prediction based on Markov bridging distributions. Algorithms designed based on the new formulation can provide competitive performance with reduced computational complexity when compared to the existing bridging distribution based methods. (Chapter 5)

Some further contributions include:

- Two samplers are designed based on the particle MCMC methods to perform full system inference under a conditionally deterministic model. (Chapter 4)
- Improvement strategies based on MCMC sampling methods are presented for use with the fixed rate and variable rate SMC filters. (Chapter 3 and 4)

1.2 Publications

Works covered in this thesis

Parts of this thesis have been published in the following peer-reviewed journal and conferences:

- **J. Liang**, B. I. Ahmad, R. Gan, P. Langdon, R. Hardy, and S. Godsill. On destination prediction based on Markov bridging distributions, *IEEE Signal Processing Letters*, vol. 26, no. 11, November 2019.
- **J. Liang** and S. Godsill. Bayesian fusion of asynchronous inertial, speed and position data for object tracking, *in Proc. of the 44th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 5212-5216, 2019.
- **J. Liang** and S. Godsill. A particle filter localisation system for indoor track cycling using an intrinsic coordinate model, *in Proc. of the 21st International Conference on Information Fusion (FUSION)*, pp. 1896-1903, 2018.

The work in Chapter 4 is going to be submitted in the near future:

- **J. Liang** and S. Godsill. Particle filtering methods for variable rate models with asynchronous sensors, *In Preparation*.

Other peer-reviewed works

The below are relevant works that are not directly covered in this thesis:

- **J. Liang**, B. I. Ahmad, and S. Godsill. Simultaneous intent prediction and state estimation using an intent-driven intrinsic coordinate model, *2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP)*. In press.
- R. Gan, **J. Liang**[1], B. I. Ahmad, and S. Godsill. Modelling intent and destination prediction within a Bayesian framework: predictive touch as a usecase, *Data-Centric Engineering*, Cambridge University Press. In press.
- Q. Li, S. Godsill, **J. Liang**, and B. I. Ahmad. Inferring dynamic group leadership using sequential Bayesian methods, *in Proc. of the 45th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 8911-8915, 2020

[1] joint first author

- B. I. Ahmad, P. M. Langdon, **J. Liang**, S. Godsill, M. Delgado, and T. Popham. Driver and passenger identification from smartphone data, *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 4, pp. 1278-1288, April 2019.
- R. Gan, **J. Liang**, B. I. Ahmad, and S. Godsill. Bayesian intent prediction for fast maneuvering objects using variable rate particle filters, *in Proc. of the 29th IEEE International Workshop on Machine Learning For Signal Processing (MLSP)*, 2019.

Chapter 2

Bayesian Inference with Sequential Monte Carlo methods

2.1 Bayesian Inference

Being able to extract accurate and/or high-level information out of real-world data coming from various sources can be beneficial as better insights towards the problem of interest can be gained by knowing such hidden information. However, in the data collection process it is almost sure that random noises will be introduced owing to, for example, the use of imperfect instruments. In many cases such noises/randomness can be characterised by some known distributions and hence the likelihood function $p(D|x)$, with D the collected data and x some hidden quantities of interest, can be evaluated accordingly. Based on the computed likelihood one may identify the most probable value of x that leads to the observed data. Clearly, this approach implicitly assumes that the hidden quantities are fixed constants and thus it can not account for the uncertainty associated with the estimate of x. In order to quantify such uncertainty, the hidden quantities x need to be treated as random variables instead of constants. Consequently, the goal of the inference task is changed from obtaining a point estimate of x based on $p(D|x)$ to estimating the distribution $p(x|D)$. The way to obtain this inverse distribution is given by the Bayes' rule:

$$p(x|D) = \frac{p(D|x)p(x)}{p(D)}$$

where the probability density function $p(x|D)$ is known as the “posterior”, $p(x)$ is termed “prior” which governs the way how the hidden variables are generated and $p(D)$

is normally served as a normalising constant given a certain set of data. The philosophy behind this simple expression may be interpreted as follows: initial assumptions with respect to x, which may come from experts in the related fields, can be made at the beginning. These prior beliefs are embedded into the prior distributions which will be converted into posterior distributions in the light of observed data. In other words, the Bayes' rule offers a solution to the problem of inferring random variables by combining the expert knowledge (subjective) with observed data (objective). The hope is that a better estimation towards the variables of interest may be obtained by doing a trade-off between prior belief and likelihood. The inference principle given above can be regarded as an informal introduction to Bayesian inference, which is a general inference framework that plays an important role in the many scientific fields such as modern statistical signal processing and machine learning. A thorough introduction to the modern Bayesian inference methods is out of the scope of this thesis as there have been many exceptional materials covering this topic (see [Bishop, 2006; Godsill, 2019a; Liu, 2004; Murphy, 2012]).

The work presented in this thesis is devoted to the developments of sequential Bayesian inference methods. By sequential we refer to the scenarios where data arrive in order over time, that is $D = y_{0:n}$ where y_n is a measurement made at time instant t_n, with $t_{i-1} < t_i, \ i \in \mathbb{N}^+$. If the system is Markovian and observations y_n are conditionally independent of all other variables provided x_n, Bayes' formula may be re-written as

$$p(x_{0:n}|y_{0:n}) = \frac{p(y_{0:n}|x_{0:n})p(x_{0:n})}{p(y_{0:n})} \propto p(y_n|x_n)p(x_n|x_{n-1})p(x_{0:n-1}|y_{0:n-1})$$

where $x_{0:n}$ denotes the hidden variables linked to the measurements. This formulation offer a natural way to recursively update the posterior using the information contained in the incoming measurement and hence it forms the basis of many sequential Bayesian inference methods, including the Sequential Monte Carlo methods considered throughout this thesis. In the following sections we will give a detailed survey covering the progress of sequential Monte Carlo methods made over the last 25 years.

2.2 Sampling Methods

In many applications we want to estimate some important properties (e.g. moments) associated with some distribution p. This normally requires the evaluation of the following integral

$$\bar{\phi} = \int \phi(x)p(x)dx = \mathbb{E}_p\{\phi(x)\} \tag{2.1}$$

with $\phi(\cdot)$ being the function of interest and $\mathbb{E}_p$ the expectation with respect to p. The fact that the above integral can not be evaluated analytically in many realistic scenarios has led to the extensive use of Monte Carlo approximation methods. In the simplest case, when random samples can be drawn from a distribution p easily the above integral can be approximated in an unbiased manner as follows

$$\bar{\phi} \approx \frac{1}{N_p} \sum_{i=1}^{N_p} \phi(x^{(i)}), \quad x^{(i)} \sim p(x), \quad i = \{1, 2, \ldots, N_p\} \tag{2.2}$$

Note that this simple Monte Carlo approach is subject to the situation where direct simulation of random samples from p is possible (e.g. via the Inverse Transform method). More often, it is not possible to sample from complicated distributions directly and thus we have to resort to indirect sampling techniques such as Importance Sampling, Markov Chain Monte Carlo (MCMC) and Rejection Sampling.

2.2.1 Importance Sampling (IS)

In importance sampling, samples are proposed from another distribution q other than p of interest [Liu, 2004; Robert and Casella, 2004]. The proposal q should be a distribution where samples can be conveniently drawn and it must have the same (or greater) support as p in order to construct a valid importance sampler. More formally, p is called the target distribution, q is known as the importance distribution (or instrumental distribution) and the ratio $\omega(x) = p(x)/q(x)$ is termed importance weight. To see how this construction can lead to a valid Monte Carlo sampling scheme, rewrite (2.1) by adding $q(x)$ at the top and the bottom:

$$\bar{\phi}_{IS} = \int \frac{\phi(x)p(x)q(x)}{q(x)} dx = \int \phi(x)w(x)q(x)dx = \mathbb{E}_q\{\phi(x)w(x)\} \tag{2.3}$$

This familiar form suggests that we may approximate the integral of interest using draws from the importance distribution:

$$\bar{\phi}_{IS} \approx \frac{1}{N_p} \sum_{i=1}^{N_p} \phi(x^{(i)})w(x^{(i)}), \qquad x^{(i)} \sim q(x) \tag{2.4}$$

It can be shown that the resulting estimator is unbiased and consistent, which means that (2.4) converges almost surely to (2.1) as N_p goes to infinity [Geweke, 1989]. Note again that although the importance sampler has the same convergence rate as the simple Monte Carlo integration, it offers a solution to the simulation of random variables from some highly complex density p.

More importantly, importance sampler can be readily applied to target distributions that are only known up to an unknown normalising constant. This feature is very useful for Bayesian applications in which the objective is to generate variable x from the following target (posterior) distribution

$$p(x|y) = \frac{p(y|x)p(x)}{\int p(y|x)p(x)dx} = \frac{p(y|x)p(x)}{C} \tag{2.5}$$

where y is the observed data and the integral in the denominator serves as a normalising constant C which is typically unknown. It can be shown that a Monte Carlo approximation for C can be obtained straightforwardly using samples generated from the importance distribution:

$$\bar{C} = \int Cp(x|y)dx = \int \frac{p(y|x)p(x)}{q(x)} q(x)dx \approx \frac{1}{N_p} \sum_{i=1}^{N_p} \tilde{\omega}(x^{(i)}) \tag{2.6}$$

with $\tilde{\omega}(x) = \frac{p(y|x)p(x)}{q(x)}$ as the unnormalised importance weight. Consequently the expectation with respect to $p(x|y)$ is approximated as

$$\begin{aligned}\bar{\phi}_{IS} &= \int \phi(x) \frac{p(x|y)}{q(x)} q(x)dx \\ &= \int \phi(x) \frac{p(y|x)p(x)}{Cq(x)} q(x)dx \\ &\approx \frac{1}{N_p} \sum_{i}^{N_p} \phi(x^{(i)}) \frac{\tilde{\omega}(x^{(i)})}{\bar{C}} \\ &= \sum_{i=1}^{N_p} \phi(x^{(i)})\omega(x^{(i)}), \qquad \omega(x^{(i)}) = \frac{\tilde{\omega}(x^{(i)})}{\sum_{j=1}^{N} \tilde{\omega}(x^{(j)})} \end{aligned} \tag{2.7}$$

where $\omega(x)$ is the (self) normalised importance weight. Further, the target distribution $p(x|y)$ can be represented using the weighted point masses

$$p(x|y) \approx \sum_{i=1}^{N_p} \omega(x^{(i)}) \delta_{x^{(i)}}(x) \tag{2.8}$$

with $\delta(\cdot)$ the Dirac delta function. Importance sampling is the building block of the particle filter with which sequential Bayesian inference on non-linear non-Gaussian state space models can be performed. While the former is briefly reviewed here, the latter will be discussed in detail in the preceding sections.

2.2.2 Markov Chain Monte Carlo (MCMC)

An alternative class of indirect sampling techniques to importance sampling, which allows simulation from some high dimensional and complicated distribution by constructing a proper Markov chain with the distribution of interest as its stationary distribution, is Markov chain Monte Carlo (MCMC). As with the target distribution in importance sampling, it is often straightforward to identify the stationary distribution, say $p(x)$, in MCMC applications. However, knowing the target distribution is not equivalent to having it as the stationary distribution of the underlying chain. Hence, it is essential to ensure the constructed Markov chain will eventually leads to the desired stationary distribution.

Denoting $K(z|x)$ as the transition kernel associated with some MCMC algorithm, it turns out that a sufficient condition for p to be a stationary (or invariant) distribution of the underlying Markov chain is that the following detailed balance is satisfied:

$$K(z|x)p(x) = K(x|z)p(z) \tag{2.9}$$

Meeting the detailed balance condition implies that once we have obtained a sample $x \sim p$, subsequent samples generated according to the transition kernel K will be distributed marginally as p as well. While the detailed balance is simply a sufficient condition to check the stationary distribution, there are several other conditions under which one can also examine whether a distribution is guaranteed to be the stationary distribution of a Markov chain. However, a detailed review on all these aspects is out of the scope of this thesis and interested readers may find [Gilks et al., 1995] a decent and thorough introduction to MCMC.

Metropolis-Hastings

The Metropolis-Hastings (MH) algorithm, as one of the most representative MCMC algorithms, was introduced and generalised in [Hastings, 1970; Metropolis et al., 1953]. Let $p(x)$ be the target distribution, $q(\cdot|x)$ be the transition distribution from which samples are drawn and ρ be the acceptance ratio based on which an accept-reject decision can be made, a MH algorithm then proceeds as per Algorithm 1. Note that $a \wedge b = \min\{a, b\}$.

```
Set $x^{(0)}$ arbitrarily;
for $m = 1, 2, \dots, N_{iter}$ do
    Sample $x^* \sim q(x|x^{m-1})$;
    Compute:
        $\rho(x^*, x^{m-1}) = 1 \wedge \frac{p(x^*)}{p(x^{m-1})} \frac{q(x^{m-1}|x^*)}{q(x^*|x^{m-1})};$
    Sample $u \sim \text{Unif}(0, 1)$;
    Set $x^m = x^*$ if $u \leq \rho(x^*, x^{m-1})$; otherwise, set $x^m = x^{m-1}$;
end
```

Algorithm 1: Metropolis-Hastings (MH) Algorithm

Recall that each MCMC algorithm implicitly defines a transition kernel from which we simulate a new state, for MH the steps taken within a single iteration represent this kernel. More formally, the transition kernel of MH can be written as:

$$K(z|x) = \rho(z|x)q(z|x) + \Big(1 - \int \rho(s|x)q(s|x)ds\Big)\delta_x(z) \tag{2.10}$$

where δ_x is the Dirac mass at x. It is straightforward to show that this kernel satisfies the detailed balance condition by considering the two parts delimited by the plus symbol separately. An appealing feature of the MH algorithm is that it provides us with great flexibility in specifying proposal distributions.

Gibbs sampling

Suppose that we want to sample from a target distribution $p(\mathbf{x})$ with $\mathbf{x} = [x_1, x_2, \dots, x_k]^T$ and define $x_{-i} = \{x_1, \dots, x_{i-1}, x_{i+1}, \dots x_k\}$ the set of variables excluding x_i. In the simplest case, the Gibbs sampler [Brooks et al., 2011; Gelfand and Smith, 1990; Geman and Geman, 1984] produces samples by operating on one element x_i at a time at the n-th iteration, that is

(a) Pick an index $i \in \{1, 2, \dots, k\}$ either systematically or uniformly.

(b) Sample from the full conditional $x_i^* \sim p(x_i|x_{-i})$

(c) Set $x_i^{(n)} = x_i^*$ and return to (a), repeat until a sweep is finished

To understand why the above procedure leads to a valid sampling scheme, we may interpret the Gibbs sampler as a special variant of the MH algorithm whose acceptance ratio always equals to one. More precisely, consider a MH sampler with the full conditional as its proposal, the acceptance ratio can be shown as

$$\begin{aligned}\rho(x_i^*, x_i^{(n-1)}) &= 1 \wedge \frac{p(x_i^*, x_{-i}^{(n-1)})q(x_i^{(n-1)})}{p(x_i^{(n-1)}, x_{-i}^{(n-1)})q(x_i^*)}\\ &= 1 \wedge \frac{p(x_i^*|x_{-i}^{(n-1)})p(x_{-i}^{(n-1)})p(x_i^{(n-1)}|x_{-i}^{(n-1)})}{p(x_i^{(n-1)}|x_{-i}^{(n-1)})p(x_{-i}^{n-1})p(x_i^*|x_{-i}^{(n-1)})}\\ &= 1\end{aligned}$$

which suggests that properties of MH algorithms may be readily inherited by the Gibbs sampler.

In summary, MCMC methods have led to a huge body of successful applications in different domains since its rediscovery in 1990s and recent advances have revealed that 1). sequential Monte Carlo (SMC) methods can be designed based on MCMC and the resulting algorithms can be strong competitors to renowned importance sampling (IS) based SMC methods in challenging scenarios [Berzuini et al., 1997; Gilks and Berzuini, 2001; Golightly and Wilkinson, 2006; Li and Godsill, 2018; Liang and Godsill, 2019; Pang et al., 2011b; Pitt and Shephard, 1999; Septier and Peters, 2016]; 2). Leveraging the output of IS-based SMC methods, efficient MCMC algorithms can be designed to estimate unknown parameters of system models [Andrieu and Roberts, 2009; Beaumont, 2003; Fernández-Villaverde and Rubio-Ramírez, 2007; Riabiz et al., 2015] or to jointly infer both parameters and hidden states [Ahmad et al., 2016b; Andrieu et al., 2010; Pitt et al., 2012].

2.3 Particle Filters

2.3.1 State Space Models

State space representation of dynamic systems is widely considered when modelling time series data. Denoting $\mathbf{y}_{0:n} \triangleq \{\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_n\}$, $\mathbf{y}_n \in \mathbb{R}^{m\times 1}$, $m \geq 1$, as observations made at times $\{t_0, t_1, \dots, t_n\}$ and $\mathbf{x}_{0:n} \triangleq \{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_n\}$, $\mathbf{x}_n \in \mathbb{R}^{s\times 1}$, $s \geq 1$ as the hidden state sequence that is of interest, a generic dynamic system may be expressed in terms of the following state space form,

$$\begin{aligned}\mathbf{x}_n &= a(\mathbf{x}_{n-1}, \mathbf{u}_n)\\ \mathbf{y}_n &= b(\mathbf{x}_n, \mathbf{v}_n)\end{aligned} \tag{2.11}$$

where $a(\cdot)$ and $b(\cdot)$ are functions which specify the respective generation criterion of $\mathbf{x}_n$ and $\mathbf{y}_n$ given the state and noise disturbances $\mathbf{u}_n$ and $\mathbf{v}_n$. These functions can be either linear or non-linear, depending on different modelling choices. Moreover, it is necessary to specify the probability distributions of the noises so as to complete the model in a probabilistic setting. We assume here that the state process is Markovian, meaning that i) each state is generated according to its previous state; ii) the observation depends only upon current state; iii) $\mathbf{u}$ and $\mathbf{v}$ are white. When the noises are additive, (2.11) becomes

$$\begin{aligned} \mathbf{x}_n &= a(\mathbf{x}_{n-1}) + \mathbf{u}_n \\ \mathbf{y}_n &= b(\mathbf{x}_n) + \mathbf{v}_n \end{aligned} \tag{2.12}$$

Alternatively, a probabilistic state space model may be described in terms of a state evolution density and an observation density:

$$\begin{aligned} \mathbf{x}_n &\sim p(\mathbf{x}_n|\mathbf{x}_{0:n-1}, \mathbf{y}_{0:n-1}) \\ \mathbf{y}_n &\sim p(\mathbf{y}_n|\mathbf{y}_{0:n-1}, \mathbf{x}_{0:n}) \end{aligned} \tag{2.13}$$

This specification can be found in a huge body of literatures related to Bayesian modeling and inference [Cappé et al., 2007; Doucet et al., 2000b; Godsill et al., 2004; Liu and Chen, 1998; Särkkä, 2013; West and Harrison, 1997] and will be adopted throughout this thesis. If again a Markov system is considered, (2.13) can be re-written as

$$\begin{aligned} \mathbf{x}_n &\sim p(\mathbf{x}_n|\mathbf{x}_{n-1}) \\ \mathbf{y}_n &\sim p(\mathbf{y}_n|\mathbf{x}_n) \end{aligned} \tag{2.14}$$

Note that (2.14) and (2.11) are equivalent. More importantly, (2.14) allows the decomposition of the joint probability density $p(\mathbf{x}_{0:N}, \mathbf{y}_{0:N})$ at t_N to be simplified as

$$p(\mathbf{x}_{0:N}, \mathbf{y}_{0:N}) = p(\mathbf{x}_0) \prod_{n=1}^{N} p(\mathbf{x}_n|\mathbf{x}_{n-1}) p(\mathbf{y}_n|\mathbf{x}_n) \tag{2.15}$$

A graphical representation corresponding to this factorisation of the joint distribution is given in Fig. 2.1. Because the hidden state shares the same set of times as the observations and a Markov process is assumed, this model can be regarded as a hidden Markov (state space) Model (HMM) [Doucet et al., 2000b]. Throughout this thesis, we will frequently refer to the models formulated in a Markovian state space form as "fixed rate" models, in order to emphasise the difference between this class of models and the "variable rate" models. The assumption of a fixed rate system is ubiquitous due to its simplicity and inference methods based on this assumption have led to many successful applications. However, sometimes a relaxation of this restriction which allows the evolution of hidden state to be decoupled with observations may be more natural and attractive. Models based on such a variable-rate assumption have been shown effective in capturing different degrees of manoeuvrability of agile objects [Godsill et al., 2007] and in predicting sudden price changes in financial markets [Christensen et al., 2012]. Because of its usefulness in modelling real-world scenarios and the easy extension to non-Markovian cases, variable rate models have become a strong competitor to conventional fixed-rate models. More details regarding variable rate models can be found in Chapter 4.

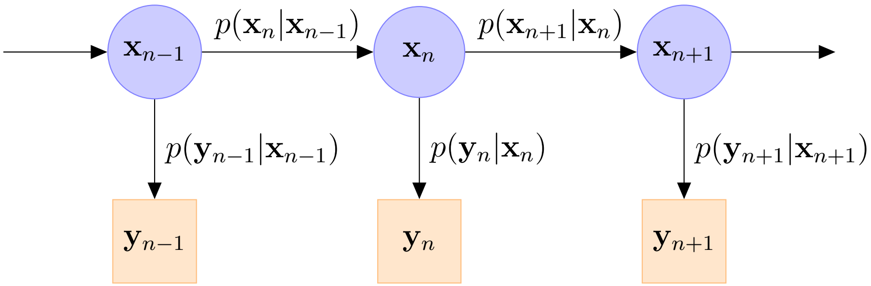

$\mathbf{x}_{n-1}$
$p(\mathbf{x}_n|\mathbf{x}_{n-1})$
$\mathbf{x}_n$
$p(\mathbf{x}_{n+1}|\mathbf{x}_n)$
$\mathbf{x}_{n+1}$
$p(\mathbf{y}_{n-1}|\mathbf{x}_{n-1})$
$p(\mathbf{y}_n|\mathbf{x}_n)$
$p(\mathbf{y}_{n+1}|\mathbf{x}_{n+1})$
$\mathbf{y}_{n-1}$
$\mathbf{y}_n$
$\mathbf{y}_{n+1}$

Fig. 2.1 Graphical model of the state space model in (2.14), showing the dependencies between the hidden state (circle) and the measurements (rectangle).

2.3.2 Bayesian Filtering

In the context of Bayesian filtering problems, our objective is to estimate the posterior density $p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})$ (or its marginal $p(\mathbf{x}_n|\mathbf{y}_{0:n})$) recursively, based on the state space models introduced before. In addition to the posterior distribution, the expectation of some function $\phi(\cdot)$ that takes the state as its input may be also of interest:

$$\bar{\phi} = \mathbb{E}\{\phi\} = \int \phi(\mathbf{x}_{0:n})p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})d\mathbf{x}_{0:n} \tag{2.16}$$

For instance, the estimator will give the posterior mean by setting $\phi(\mathbf{x}_{0:n}) = \mathbf{x}_{0:n}$. To illustrate the recursive procedure using which one can approach the desired posterior distribution over time, we start with assuming that $p(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1})$ from the last time instant is available. Consequently, the joint predictive distribution of the hidden state may be expressed as below,

$$\begin{aligned} p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n-1}) &= p(\mathbf{x}_n|\mathbf{x}_{0:n-1}, \mathbf{y}_{0:n-1})p(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1}) \\ &= p(\mathbf{x}_n|\mathbf{x}_{n-1})p(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1}) \end{aligned} \tag{2.17}$$

where the second line stems from the Markovian assumption. Alternatively, this prediction step can be formalised in terms of marginalisation, as in [Arulampalam et al., 2002] and [Doucet and Johansen, 2011],

$$p(\mathbf{x}_n|\mathbf{y}_{0:n-1}) = \int p(\mathbf{x}_n|\mathbf{x}_{n-1})p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1})d\mathbf{x}_{n-1} \tag{2.18}$$

This is partially due to the fact that many conventional filtering methods such as Kalman filters seek for closed form solutions for the above integral, whereas in intractable cases where simulation-based techniques are used the joint distribution (2.17), of which (2.18) admits a marginal, is believed to be a relatively more convenient representation.

According to Bayes' rule, information contained in the incoming measurement $\mathbf{y}_n$ can be incorporated into (2.17) via

$$\begin{aligned} p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n}) &= \frac{p(\mathbf{y}_n|\mathbf{x}_{0:n}, \mathbf{y}_{0:n-1})p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n-1})}{p(\mathbf{y}_n|\mathbf{y}_{0:n-1})} \\ &\propto p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n-1}) \end{aligned} \tag{2.19}$$

with the normalising constant given by

$$p(\mathbf{y}_n|\mathbf{y}_{0:n-1}) = \int p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{0:n-1})d\mathbf{x}_n \tag{2.20}$$

It is now clear that expressions (2.17) and (2.19) together offer a solution to the problem of recursive estimation of $p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})$ as well as any features associated with it. They are known as the "prediction-correction recursion" as per [Cappé et al., 2007; Ho and Lee, 1964; Särkkä, 2013].

If the initial distribution $p(\mathbf{x}_0)$ is Gaussian and the models in (2.14) are linear-Gaussian, it can be shown that $p(\mathbf{x}_n|\mathbf{y}_{0:n})$ is still Gaussian ($p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})$ is henceforth jointly Gaussian) owing to the properties of conjugate distributions. From this point onwards, a Bayesian interpretation of Kalman filters [Anderson and Moore, 1979] can be obtained [Ho and Lee, 1964]. However, although linear-Gaussian models or discrete HMMs can result in closed-form solutions the cases where they can be directly applied are limited. For most state space scenarios, the dynamic system is typically non-linear and/or non-Gaussian which means analytical forms of $p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n}) \propto p(\mathbf{x}_{0:n}, \mathbf{y}_{0:n})$

are usually unavailable. Henceforth, a lot of effort has been made to improve the conventional Kalman filtering methods such that they can accommodate non-linear systems. Among them, the extended Kalman filter (EKF) and its variants linearise the functions in (2.12) when computing covariance matrices so that analytical Kalman recursions can be applied again in inference. The approximation errors introduced by the linearisation as well as the uni-modal assumption on the posterior distribution can lead to poor performance when dealing with systems of high non-linearity and/or non-Gaussianity [Wan and Van Der Merwe, 2000]. Instead of relying on the lower-order linearisation as in EKFs, the unscented Kalman filter (UKF) [Julier and Uhlmann, 1997; Wan and Van Der Merwe, 2000] approximates the posterior distribution via a set of "sigma points" which are carefully chosen according to a deterministic sampling technique called unscented transformation (UT). As the distribution given by the UKF is parameterised by a mean and covariance (i.e. Gaussian), it still leads to a mismatch when the true posterior is multi-modal (which is typical for non-linear systems).

In fact, all these drawbacks/limitations associated with Kalman-type approaches have contributed to the great interest in using sampling-based methods for solving filtering and smoothing problems. The remaining question is whether it is feasible to adapt the aforementioned sampling strategies so that they can operate in a sequential manner to suit sequential Bayesian inference problems as per Kalman methods. The answer is YES thanks to the pioneering work by [Gordon et al., 1993] and many other parallel/following works as in [Andrieu et al., 2010; Carpenter et al., 1999; Doucet et al., 2000a,b; Gilks and Berzuini, 2001; Godsill and Clapp, 2001; Godsill et al., 2007; Hürzeler and Künsch, 1998; Kitagawa, 1996; Liu and Chen, 1998; Moral et al., 2006; Pang et al., 2011b; Pitt and Shephard, 1999; Septier and Peters, 2016] - just to name a few. In the sequel, we will start by introducing particle filtering methods. Later, an alternative SMC method termed "sequential MCMC" will be also discussed since together with the particle filter they serve as the main tools used in this thesis.

2.3.3 Sequential Importance Sampling

Importance Sampling, as discussed before, is not inherently recursive and thus not applicable for performing direct sequential inference. To see how a sequential version of IS may be designed, first consider factorising the posterior distribution (or target distribution) to be estimated in terms of the Bayesian prediction-correction recursion (i.e. (2.17) and (2.19)):

$$p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n}) = \frac{p(\mathbf{y}_n|\mathbf{x}_{0:n}, \mathbf{y}_{0:n-1})p(\mathbf{x}_n|\mathbf{x}_{0:n-1}, \mathbf{y}_{0:n-1})p(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1})}{p(\mathbf{y}_n|\mathbf{y}_{0:n-1})}$$

As in the non-sequential Importance Sampling scenario, we denote $q(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})$ as the importance distribution (or instrumental distribution) from which random samples of the state trajectory $\mathbf{x}_{0:n}$ can be drawn conveniently. The resulting equation for computing the unnormalised importance weights is then given by

$$\tilde{\omega}_n = \frac{p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})}{q(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})} = \frac{p(\mathbf{y}_n|\mathbf{x}_{0:n}, \mathbf{y}_{0:n-1})p(\mathbf{x}_n|\mathbf{x}_{0:n-1}, \mathbf{y}_{0:n-1})p(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1})}{q(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})p(\mathbf{y}_n|\mathbf{y}_{0:n-1})} \tag{2.21}$$

The key to a recursive version of (2.21) then lies in the following construction of the importance distribution:

$$q(\mathbf{x}_{0:n}|\mathbf{y}_{0:n}) = q(\mathbf{x}_n|\mathbf{x}_{0:n-1}, \mathbf{y}_{0:n})q(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1}) \tag{2.22}$$

Supposing that a finite set of weighted particles $\{\mathbf{x}_{0:n-1}^{(i)}, \omega_{n-1}^{(i)}\}_{i=1}^{N_p}$ has been generated to approximate the joint posterior distribution at the previous time instant:

$$p(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1}) \approx \sum_{i=1}^{N_p} \omega_{n-1}^{(i)} \delta_{\mathbf{x}_{0:n-1}^{(i)}}(\mathbf{x}_{0:n-1}), \qquad \omega_{n-1}^{(i)} = \frac{\tilde{\omega}_{n-1}^{(i)}}{\sum_{j=1}^{N_p} \tilde{\omega}_{n-1}^{(j)}}, \ \omega_{n-1}^{(i)} \geq 0$$

and plugging (2.22) into (2.21), we end up with

$$\begin{aligned} \tilde{\omega}_n^{(i)} &= \frac{p(\mathbf{x}_{0:n}^{(i)}|\mathbf{y}_{0:n})}{q(\mathbf{x}_{0:n}^{(i)}|\mathbf{y}_{0:n})} \\ &= \frac{p(\mathbf{x}_{0:n-1}^{(i)}|\mathbf{y}_{0:n-1})}{q(\mathbf{x}_{0:n-1}^{(i)}|\mathbf{y}_{0:n-1})} \frac{p(\mathbf{y}_n|\mathbf{x}_{0:n}^{(i)}, \mathbf{y}_{0:n-1})p(\mathbf{x}_n^{(i)}|\mathbf{x}_{0:n-1}^{(i)}, \mathbf{y}_{0:n-1})}{q(\mathbf{x}_n^{(i)}|\mathbf{x}_{0:n-1}^{(i)}, \mathbf{y}_{0:n})p(\mathbf{y}_n|\mathbf{y}_{0:n-1})} \\ &\propto \omega_{n-1}^{(i)} \frac{p(\mathbf{y}_n|\mathbf{x}_{0:n}^{(i)}, \mathbf{y}_{0:n-1})p(\mathbf{x}_n^{(i)}|\mathbf{x}_{0:n-1}^{(i)}, \mathbf{y}_{0:n-1})}{q(\mathbf{x}_n^{(i)}|\mathbf{x}_{0:n-1}^{(i)}, \mathbf{y}_{0:n})} \end{aligned} \tag{2.23}$$

with $\mathbf{x}_n^{(i)} \sim q(\mathbf{x}_n|\mathbf{x}_{0:n-1}^{(i)}, \mathbf{y}_{0:n})$. The proportionality can be achieved due to the self-normalised property of importance weights and the state-independent factor $p(\mathbf{y}_n|\mathbf{y}_{0:n-1})$. This equation allows the sequential update for importance weights and thus forms the basis of the Sequential Importance Sampling (SIS) algorithm [Arulampalam et al., 2002; Doucet et al., 2000b; Liu and Chen, 1998]. When the

Markovianity is assumed, (2.23) can be modified as

$$\tilde{\omega}_n^{(i)} \propto \omega_{n-1}^{(i)} \frac{p(\mathbf{y}_n|\mathbf{x}_n^{(i)})p(\mathbf{x}_n^{(i)}|\mathbf{x}_{n-1}^{(i)})}{q(\mathbf{x}_n^{(i)}|\mathbf{x}_{0:n-1}^{(i)}, \mathbf{y}_{0:n})} \tag{2.24}$$

This is the form that can be found in numerous tutorials and reviews about particle filters [Arulampalam et al., 2002; Cappé et al., 2007; Doucet et al., 2001, 2000b; Doucet and Johansen, 2011; Godsill, 2019b; Künsch, 2013]. Consequently, an approximation for the state posterior, after taking into account the newly received observation $\mathbf{y}_n$, is given by

$$p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n}) \approx \sum_{i=1}^{N_p} \omega_n^{(i)} \delta_{\mathbf{x}_{0:n}^{(i)}}(\mathbf{x}_{0:n}) \tag{2.25}$$

Based on (2.25), many quantities of interest may be obtained straightforwardly. For example, in most online applications, one may be particularly concerned with the marginal distribution

$$p(\mathbf{x}_n|\mathbf{y}_{0:n}) = \int p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})d\mathbf{x}_{0:n-1} \approx \sum_{i=1}^{N_p} \omega_n^{(i)} \delta_{\mathbf{x}_n^{(i)}}(\mathbf{x}_n) \tag{2.26}$$

as well as the associated mean and covariance:

$$\begin{aligned} \boldsymbol{\mu}_{n|0:n} &= \mathbb{E}\{\mathbf{x}_n|\mathbf{y}_{0:n}\} \approx \sum_{i=1}^{N_p} \omega_n^{(i)} \mathbf{x}_n^{(i)} \\ P_{n|0:n} &= \mathbb{E}\{(\mathbf{x}_n - \boldsymbol{\mu}_{n|0:n})(\mathbf{x}_n - \boldsymbol{\mu}_{n|0:n})^T|\mathbf{y}_{0:n}\} \\ &\approx \sum_{i=1}^{N_p} \omega_n^{(i)} (\mathbf{x}_n^{(i)} - \boldsymbol{\mu}_{n|0:n})(\mathbf{x}_n^{(i)} - \boldsymbol{\mu}_{n|0:n})^T \end{aligned} \tag{2.27}$$

For the forecasting purpose a m-step ahead predictive distribution may be of interest. This distribution can be also estimated using the same SIS framework since

$$\tilde{\omega}_{n+m}^{(i)} = \frac{p(\mathbf{x}_{0:n+m}^{(i)}|\mathbf{y}_{0:n})}{q(\mathbf{x}_{0:n+m}^{(i)}|\mathbf{y}_{0:n})} = \frac{\left(\prod_{j=n+1}^{n+m} p(\mathbf{x}_j^{(i)}|\mathbf{x}_{j-1}^{(i)})\right)p(\mathbf{x}_{0:n}^{(i)}|\mathbf{y}_{0:n}^{(i)})}{\left(\prod_{k=n+1}^{n+m} q(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})\right)q(\mathbf{x}_{0:n}^{(i)}|\mathbf{y}_{0:n}^{(i)})} \tag{2.28}$$

This suggests that if new states are drawn according to the state evolution density up to time t_{n+m} (i.e. setting $q(x_{n+1:n+m}|x_n) = p(x_{n+1:n+m}|x_n)$), the predictive weights using which we approximate $p(\mathbf{x}_{0:n+m}|\mathbf{y}_{0:n})$ will be identical to those at t_n. More specifically, in the light of the relationship $\tilde{\omega}_{n+m}^{(i)} \propto \omega_n^{(i)}$, an estimate of the predictive marginal distribution may be obtained as

$$p(\mathbf{x}_{n+m}|\mathbf{y}_{0:n}) \approx \sum_{i=1}^{N_p} \omega_n^{(i)} \delta_{\mathbf{x}_{n+m}^{(i)}}(\mathbf{x}_{n+m}) \tag{2.29}$$

Furthermore, one may have already realised that the joint distribution (2.25) can be regarded as a smoothing distribution. This property of having the smoothing density as a side product when performing sequential Monte Carlo filtering was first described in [Kitagawa, 1996].

Despite being adapted to a sequential setting, the IS-based framework presented by far is known for its poor performance when dealing with long data sequence, namely $n \in \mathbb{N}_0$ is large. This is due to the fact that we are trying to sample from a high dimensional state space with increasing dimensions using limited samples and a proposal which may be significantly distinctive from the target distribution. Imaging that at some time step many particles are far away from the peak of the likelihood function, they may never make it back to the right track in the subsequent steps. In the end, we only have very few particles that carries most of the probability mass. This is known as the degeneracy problem whose degree can be measured via the so-called “effective sample size” [Kong, 1992; Kong et al., 1994]. Anyhow, because of the unavoidable degeneracy the estimation error will accumulate over time, leading to completely useless results after several iterations.

2.3.4 Sequential Importance Sampling and Resampling

In order to use SIS in practice, the degeneracy problem has to be taken care of. There exists several favourable remedies for this problem. The first one is based on the idea of “Sampling Importance Resampling” introduced by [Rubin, 1987] where a resampling step is used following the weighting step of a non-sequential importance sampler. It turns out that this additional random selection scheme can be readily adopted in the case of SIS. The resulting recursive Monte Carlo algorithms are known as the “Sequential Importance Sampling and Resampling” (or SIR [Doucet et al., 2000b], which should not be confused with Rubin’s sampling importance resampling technique) methods. More generally, this class of methods is usually referred to as “particle filters” (PFs). In the following sections, these two terms will be used interchangeably.

In the simplest case, at time t_n the resampling procedure may be carried out by drawing M independent and identically distributed samples from a multinomial distribution parameterised by the normalised particle weights, that is

$$N_1, N_2, \ldots, N_{N_p} | M \sim \text{Mult}\Big(M, \{\omega_n^{(i)}\}_{i=1}^{N_p}\Big)$$

where $\sum_{i=1}^{N_p} N_i = M$. Following this procedure, the particle set $\{\mathbf{x}_{0:n}^{(i)}\}_{i=1}^{N_p}$ will be replaced by a new one $\{\mathbf{x}_{0:n}^{(j)}\}_{j=1}^{M}$ in which there are N_i replicas of the i-th particle in the original set and the importance weight associated with each particle will be reset to $\frac{1}{M}$. It is common to set $M = N_p$ for a fixed budget implementation, but theoretically M can be any number (mostly $M \geq N_p$). After such an intermediate random selection step, particles with large enough weights are split into multiple identical offsprings which can independently explore the state space at the next time step. As for particles with negligible weights, it is very likely that they will never be visited during the resampling sweep and thus be discarded before being propagated to the next iteration. In this way, the resulting algorithm can deliver better estimation results. Expect for this intuitive "sample with replacement" approach, there are many other more elaborate resampling methods that can lead to variance reduction, such as stratified resampling [Kitagawa, 1996], residual resampling [Liu and Chen, 1998], systematic resampling [Carpenter et al., 1999] and branching [Bain and Crisan, 2008]. For some theoretical analyses regarding common resampling methods, see [Douc et al., 2005; Gerber et al., 2019]. Nonetheless, in this thesis only the multinomial resampling method and a fully deterministic sampling method as in [Godsill et al., 2007] are considered.

It should be also pointed out that resampling actions do not have to be executed at each time step as in the debut of "bootstrap filter" [Gordon et al., 1993]. Resampling introduces additional Monte Carlo variance via random draws and thus we may only want to reasample particles when the degree of degeneracy reaches some certain thresholds. As mentioned before, effective sample size (ESS) is a good measure for degeneracy, meaning that it may be used as an indicator for making a resampling desicion. Specifically, ESS is defined as below

$$\begin{aligned} N_{ESS,n} &= \frac{N_p}{1 + var_{q(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})}(\bar{\omega}_n)} \\ &= \frac{N_p}{1 + \mathbb{E}_{q(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})}(\bar{\omega}_n^2) - \Big(\mathbb{E}_{q(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})}(\bar{\omega}_n)^2\Big)} \\ &= \frac{N_p}{\mathbb{E}_{q(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})}(\bar{\omega}_n^2)} \end{aligned} \tag{2.30}$$

where $var_{q(\cdot)}$ and $\mathbb{E}_{q(\cdot)}$ are the variance and the expectation with respect to distribution $q(\cdot)$, respectively. $\bar{\omega}_n = \frac{\tilde{\omega}_n^{(i)}}{\frac{1}{N_p}\sum_{i=1}^{N_p} \tilde{\omega}_n^{(i)}}$ are the standardised importance weights. Since (2.30) is not tractable, a Monte Carlo approximation to it is usually considered:

$$\widehat{N}_{ESS,n} = \frac{1}{\sum_{i=1}^{N_p}(\omega_n^{(i)})^2} = \frac{\left(\sum_{i=1}^{N_p}(\tilde{\omega}_n^{(i)})\right)^2}{\sum_{i=1}^{N_p}(\tilde{\omega}_n^{(i)})^2} \tag{2.31}$$

Small $\widehat{N}_{ESS,n}$ implies that only a few samples needs to be drawn from the target distribution in order to achieve the same variance as that of the IS estimator - a sign of degeneracy. Typically, the resampling step is triggered once $\widehat{N}_{ESS,n}$ is below some empirical threshold N_{thold}, e.g. $\frac{2}{3}N_p$.

For state space models, combining SIS with an optional resampling step gives rise to the well-known SIR algorithm as summarised in Algorithm 2. Note that the resampling method in the algorithm can be replaced by any other suitable schemes, as discussed above. Moreover, in practice owing to the limited machine precision, weight calculations can easily fail if a particle filter operates on a linear scale. To tackle this issue, one may instead store all weights on a log scale and bring them back to the linear scale when necessary. It is also helpful to subtract the largest log-weight from all computed log-weights in order to ensure large weights are within machine accuracy, as suggested by [Cappé et al., 2007]. The differences between the SIR filter and the bootstrap filter are that the latter uses the state transition density (prior) as the importance distribution (proposal) and resamples particles at each time step. Interestingly, 70 years ago, a SIS-like algorithm, which also used the same proposal as in bootstrap filter, was proposed in [Handschin and Mayne, 1969]. However, the resampling step, which turns out to be the key enabler for modern particle filters, was missing back then.

Better proposals

In addition to the use of resampling strategies, mitigation of degeneracy can be achieved by sampling from better proposals which lead to the reduction in the variance of importance weights. Considering that a SIR filter is designed using the prior kernel $p(\mathbf{x}_n|\mathbf{x}_{n-1})$ as its proposal, (2.24) can be simplified as

$$\tilde{\omega}_n^{(i)} \propto \omega_{n-1}^{(i)} p(\mathbf{y}_n|\mathbf{x}_n^{(i)}) \tag{2.32}$$

```
for $i = 1, 2, \ldots, N_p$ do
    Sample $\mathbf{x}_0^{(i)} \sim q(\mathbf{x}_0|\mathbf{y}_0)$;
    Compute initial particle weights: $\omega_0^{(i)} \propto \frac{p(\mathbf{y}_0|\mathbf{x}_0^{(i)})p(\mathbf{x}_0^{(i)})}{q(\mathbf{x}_0^{(i)}|\mathbf{y}_0)}$;
end
for $n = 1, 2, \ldots, N$ do
    if $\widehat{N}_{ESS,n} \leq N_{thold}$ then
        For all $i$:
        Select indexes $\{j^{(i)}\}$ with probability $\Pr(j^{(i)} = k) = \omega_{n-1}^{(k)}$;
        Reset weight $\omega_{n-1}^{(i)} = \frac{1}{N_p}$ ;
    end
    for $i = 1, 2, \ldots, N_p$ do
        Sample $\mathbf{x}_n^{(i)} \sim q(\mathbf{x}_n|\mathbf{x}_{0:n-1}^{(i)}, \mathbf{y}_{0:n-1})$;
        Weighting: $\tilde{\omega}_n^{(i)} = \omega_{n-1}^{(i)} \frac{p(\mathbf{y}_n|\mathbf{x}_{0:n}^{(i)},\mathbf{y}_{0:n-1})p(\mathbf{x}_n^{(i)}|\mathbf{x}_{0:n-1}^{(i)},\mathbf{y}_{0:n-1})}{q(\mathbf{x}_n^{(i)}|\mathbf{x}_{0:n-1}^{(i)},\mathbf{y}_{0:n})}$ ;
        Update state trajectories: $\mathbf{x}_{0:n}^{(i)} = \{\mathbf{x}_n^{(i)}, \mathbf{x}_{0:n-1}^{(i)}\}$ ;
    end
    for $i = 1, 2, \ldots, N_p$ do
        Normalise weights: $\omega_n^{(i)} = \frac{\tilde{\omega}_n^{(i)}}{\sum_{j=1}^{N_p} \tilde{\omega}_n^{(j)}}$;
    end
end
```

Algorithm 2: Sequential Importance Sampling and Resampling (SIR)

which shows that the update of importance weights only depends on the likelihood function. This construction is often associated with poor performance/severe degeneracy because proposing states blindly can result in a mismatch between the predictive distribution $p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n-1})$ and the posterior distribution $p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})$. Although this problem may be mitigated by significantly increasing the number of particles, it is not practical in most scenarios.

Instead, we may wish to propose states in the light of more useful information. The form of the general proposal $q(\mathbf{x}_n|\mathbf{x}_{0:n-1}, \mathbf{y}_{0:n})$ has already indicated that this proposal can be constructed by incorporating further information such as the incoming measurement $\mathbf{y}_n$. For a Markovian system if we can sample from $q(\mathbf{x}_n|\mathbf{x}_{0:n-1}, \mathbf{y}_n) = q(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)$, the variance of importance weights $var_{q(\mathbf{x}_n|\mathbf{x}_{n-1}^{(i)}, \mathbf{y}_n)}\{\tilde{\omega}_n^{(i)}\}$ will be minimised. This conditional state posterior is termed "optimal kernel" [Doucet et al., 2000b; Liu, 2004] and can be obtained as

$$q(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n) = \frac{p(\mathbf{x}_n|\mathbf{x}_{n-1})p(\mathbf{y}_n|\mathbf{x}_n)}{\int p(\mathbf{x}_n|\mathbf{x}_{n-1})p(\mathbf{y}_n|\mathbf{x}_n)d\mathbf{x}_n} = \frac{p(\mathbf{x}_n|\mathbf{x}_{n-1})p(\mathbf{y}_n|\mathbf{x}_n)}{p(\mathbf{y}_n|\mathbf{x}_{n-1})} \tag{2.33}$$

This proposal kernel should be used whenever it is possible and a discussion on models for which this distribution can be obtained in a closed form can be found in [Doucet et al., 2000b]. While it is in general difficult to obtain an optimal kernel analytically, approximation methods, such as the local linearisation and the progressive Gaussian approximation [Bunch and Godsill, 2013b], may be considered. Note also that in some cases there may be a subset of $\mathbf{x}_n$ whose optimal importance distribution is available [Godsill and Clapp, 2001; Liang and Godsill, 2018].

Auxiliary particle filter

Keeping the idea that proposals for $\mathbf{x}_n$ may be built in a smarter way by taking into account the new measurement $\mathbf{y}_n$ in mind and re-inspecting the importance distribution in (2.22), a natural thought would be whether it is plausible to modify the proposal for the past trajectories $\mathbf{x}_{0:n-1}$ such that

$$q(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1}) \xrightarrow{\mathbf{y}_n} q(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n})$$

This idea was firstly explored in [Pitt and Shephard, 1999] although here our presentation is based on [Cappé et al., 2007]. Suppose that a good empirical approximation to $p(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1})$ has been obtained at t_{n-1}, the sought density $q(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n})$ can be written as

$$\begin{aligned} q(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n}) &\propto \int p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{x}_{n-1})p(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1})d\mathbf{x}_n \\ &\approx \sum_{i=1}^{N_p} \omega_{n-1}^{(i)} \delta_{\mathbf{x}_{0:n-1}^{(i)}}(\mathbf{x}_{0:n-1}) \int p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{x}_{n-1}^{(i)})d\mathbf{x}_n \end{aligned} \tag{2.34}$$

When the integration in (2.34) is not tractable, we can resort to approximation. For instance, we may set $p(\mathbf{x}_n|\mathbf{x}_n^{(i)}) \approx \delta_{\boldsymbol{\mu}_n^{(i)}}(\mathbf{x}_n)$ with $\boldsymbol{\mu}_n^{(i)}$ being the mean, the mode or a draw associated with $p(\mathbf{x}_n|\mathbf{x}_n^{(i)})$. As a result, the following discrete distribution can be obtained

$$q(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n}) \propto \sum_{i=1}^{N_p} p(\mathbf{y}_n|\boldsymbol{\mu}_n^{(i)})\omega_{n-1}^{(i)} \delta_{\mathbf{x}_{0:n-1}^{(i)}}(\mathbf{x}_{0:n-1}) \tag{2.35}$$

from which we can easily sample $\mathbf{x}_{0:n-1}$.

A modification to the above approximation is to use other sampling techniques such as MCMC and rejection sampling with $q(\mathbf{x}_{0:n-1}, \mathbf{x}_n|\mathbf{y}_{0:n})$ as their target distributions, as also suggested by [Pitt and Shephard, 1999]. This idea is also closely related to the sequential MCMC framework [Pang et al., 2011b; Septier and Peters, 2016] in which a joint MH sampling step is adopted to draw samples from the state posterior $p(\mathbf{x}_{0:n-1}, \mathbf{x}_n|\mathbf{y}_{0:n})$, as will be discussed later in this chapter. For more details regarding different usages and developments of auxiliary particle filters, see [Whiteley and Johansen, 2010].

A more general SIR algorithm

According to (2.35), the weight for the past state sequence may be defined as

$$v_{n-1}^{(i)} \propto p(\mathbf{y}_n|\boldsymbol{\mu}_n^{(i)})\omega_{n-1}^{(i)}, \quad \sum_{i=1}^{N_p} v_{n-1}^{(i)} = 1$$

such that

$$q(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n}) = \sum_{i=1}^{N_p} v_{n-1}^{(i)} \delta_{\mathbf{x}_{0:n-1}^{(i)}}(\mathbf{x}_{0:n-1})$$

Although here we have used the auxiliary particle filter as an example, the implication is that we are free to design a proposal for $\mathbf{x}_{0:n-1}$ using a set of deliberately designed weights $\{v_{n-1}^{(i)}\}_{1\le i\le N_p}$. Accordingly, we can re-write the weight update equation in SIR (i.e. (2.23)) as follows

$$\begin{aligned}\tilde{\omega}_n^{(i)} &= \frac{p(\mathbf{x}_{0:n}^{(i)}|\mathbf{y}_{0:n})}{q(\mathbf{x}_{0:n}^{(i)}|\mathbf{y}_{0:n})} \\ &\propto \frac{p(\mathbf{x}_{0:n-1}^{(i)}|\mathbf{y}_{0:n-1})}{q(\mathbf{x}_{0:n-1}^{(i)}|\mathbf{y}_{0:n})} \frac{p(\mathbf{y}_n|\mathbf{x}_{0:n}^{(i)}, \mathbf{y}_{0:n-1})p(\mathbf{x}_n^{(i)}|\mathbf{x}_{0:n-1}^{(i)}, \mathbf{y}_{0:n-1})}{q(\mathbf{x}_n^{(i)}|\mathbf{x}_{0:n-1}^{(i)}, \mathbf{y}_{0:n})} \\ &= \frac{\omega_{n-1}^{(i)}}{v_{n-1}^{(i)}} \times \frac{p(\mathbf{y}_n|\mathbf{x}_{0:n}^{(i)}, \mathbf{y}_{0:n-1})p(\mathbf{x}_n^{(i)}|\mathbf{x}_{0:n-1}^{(i)}, \mathbf{y}_{0:n-1})}{q(\mathbf{x}_n^{(i)}|\mathbf{x}_{0:n-1}^{(i)}, \mathbf{y}_{0:n})}\end{aligned} \tag{2.36}$$

Expression (2.36) covers many important variants of particle filters, including the auxiliary particle filter discussed above, and hence it serves as a general weight update formula. This paradigm was first proposed in [Godsill and Clapp, 2001]. A SIR algorithm formalised in terms of this general weight expression is given in Algorithm 3.

Likelihood evaluation

Another key feature of particle filter is its capability of providing an unbiased estimate of the intractable likelihood $p(\mathbf{y}_{0:n})$ as a by-product. Specifically, the overall likelihood can be factorised as

$$p(\mathbf{y}_{0:n}) = p(\mathbf{y}_0) \prod_{k=1}^{n} p(\mathbf{y}_k|\mathbf{y}_{0:k-1}) \tag{2.37}$$

with

$$p(\mathbf{y}_k|\mathbf{y}_{0:k-1}) = \int\int p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{y}_{0:k-1})d\mathbf{x}_{k-1}d\mathbf{x}_k$$

which can be computed approximately using the predictive filtering distribution as in [Cappé et al., 2007; Doucet et al., 2000b]. Using (2.37), the overall likelihood can be evaluated sequentially over time. This quantity is of great importance when it comes to the estimation of system parameters θ because implicitly we have $p(\mathbf{y}_k|\mathbf{y}_{0:k-1}) = p(\mathbf{y}_k|\mathbf{y}_{0:k-1}, \theta)$.

```
for $i = 1, 2, \ldots, N_p$ do
    Sample $\mathbf{x}_0^{(i)} \sim q(\mathbf{x}_0|\mathbf{y}_0)$;
    Compute initial particle weights: $\omega_0^{(i)} \propto \frac{p(\mathbf{y}_0|\mathbf{x}_0^{(i)})p(\mathbf{x}_0^{(i)})}{q(\mathbf{x}_0^{(i)}|\mathbf{y}_0)}$;
end
for $n = 1, 2, \ldots, N$ do
    for $i = 1, 2, \ldots, N_p$ do
        Select $\mathbf{x}_{0:n-1}^{(i)}$ with probability $\{v_{n-1}^{(i)}\}$;
        Some common choices[a]:
            ○ $v_{n-1}^{(i)} = \omega_{n-1}^{(i)}$ leads to the standard resampling step as in Algo. 2;
            ○ $v_{n-1}^{(i)} = \frac{1}{N_p}$ leads to SIS;
            ○ $v_{n-1}^{(i)} \propto p(\mathbf{y}_n|\boldsymbol{\mu}_n^{(i)})\omega_{n-1}^{(i)}$ leads to the auxiliary particle filter;
    end
    for $i = 1, 2, \ldots, N_p$ do
        Sample $\mathbf{x}_n^{(i)} \sim q(\mathbf{x}_n|\mathbf{x}_{0:n-1}^{(i)}, \mathbf{y}_{0:n-1})$;
        Weighting: $\tilde{\omega}_n^{(i)} = \frac{\omega_{n-1}^{(i)}}{v_{n-1}^{(i)}} \frac{p(\mathbf{y}_n|\mathbf{x}_{0:n}^{(i)},\mathbf{y}_{0:n-1})p(\mathbf{x}_n^{(i)}|\mathbf{x}_{0:n-1}^{(i)},\mathbf{y}_{0:n-1})}{q(\mathbf{x}_n^{(i)}|\mathbf{x}_{0:n-1}^{(i)},\mathbf{y}_{0:n})}$ ;
        Update state trajectories: $\mathbf{x}_{0:n}^{(i)} = \{\mathbf{x}_n^{(i)}, \mathbf{x}_{0:n-1}^{(i)}\}$ ;
    end
    for $i = 1, 2, \ldots, N_p$ do
        Normalise weights: $\omega_n^{(i)} = \frac{\tilde{\omega}_n^{(i)}}{\sum_{j=1}^{N_p} \tilde{\omega}_n^{(j)}}$;
    end
end
```

Algorithm 3: A general SIR algorithm

[a] For algorithms that take future information into account when designing $\{v_{n-1}^{(i)}\}$, see [Lin et al., 2013]

2.3.5 Rao-Blackwellised Particle Filters

Standard particle filters (SIR filters) can be readily applied to non-linear and/or non-Gaussian dynamical systems, but it should be noted that for high dimensional systems a very large number of particles are usually required to deliver satisfactory estimation results owing to high computational complexity. For example, [Snyder et al., 2008] reported that there is an exponential relationship between the number of particles (computational effort) and the state dimension for a non-sequential importance sampling and resampling procedure. This necessitates many remedies for making the basic filter more efficient and effective, among which the Rao-Blackwellisation [Doucet et al., 2000b; Liu and Chen, 1998; MacEachern et al., 1999; Robert and Casella, 2004] method stands out as one popular choice. The idea is that if some subset of a state vector turns out to be conditionally tractable it can be marginalised out such that the particle filter will only need to operate on a state space with reduced dimensionality. It can be shown that the resulting importance weights will have smaller variances [MacEachern et al., 1999] and improvements in terms of mean squared error may be obtained [Casella and Robert, 1996]. Here the presentation will focus on an important class of models which is termed conditionally linear Gaussian state space model.

Suppose the original state vector $\mathbf{x} = [x_1, x_2, \ldots, x_d]^T$ can be partitioned into two subsets

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}^N \\ \hdashline \mathbf{x}^L \end{bmatrix} \tag{2.38}$$

where $\mathbf{x}^N$ denotes the collection of non-linear state variables while $\mathbf{x}^L$ is the linear subset conditioned on $\mathbf{x}^N$. Then we may factorise the joint posterior distribution in the following form

$$p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n}) = p(\mathbf{x}^L_{0:n}|\mathbf{x}^N_{0:n}, \mathbf{y}_{0:n})p(\mathbf{x}^N_{0:n}|\mathbf{y}_{0:n}) \tag{2.39}$$

From (2.39) one may readily recognise that if the conditionlly linear Gaussian component $p(\mathbf{x}^L_{0:n}|\mathbf{x}^N_{0:n}, \mathbf{y}_{0:n})$ can be computed by Kalman filters and smoothers the nonlinear component can be obtained as a result of marginalisation:

$$p(\mathbf{x}^N_{0:n}|\mathbf{y}_{0:n}) = \int p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})d\mathbf{x}^L_{0:n} \tag{2.40}$$

Consequently, (2.40) is the distribution which a particle filter targets. Because of the marginalisation, Rao-Blackwellised particle filter (RBPF) is frequently referred to as the marginalised particle filter. To see a concrete example of the method, consider a general model whose variants have been seen in [Hostettler and Särkkä, 2019; Lindsten et al., 2016; Schön et al., 2005]:

$$\mathbf{x}_n^N = a(\mathbf{x}_{n-1}^N) + A_n(\mathbf{x}_{n-1}^N)\mathbf{x}_n^L + \mathbf{u}_n \tag{2.41}$$

$$\mathbf{x}_n^L = b(\mathbf{x}_{0:n-1}^N) + B_n(\mathbf{x}_{0:n-1}^N)\mathbf{x}_{n-1}^L + \boldsymbol{\nu}_n \tag{2.42}$$

$$\mathbf{y}_n = c(\mathbf{x}_{0:n}^N) + C_n(\mathbf{x}_{0:n}^N)\mathbf{x}_n^L + \boldsymbol{\epsilon}_n \tag{2.43}$$

where $a(\cdot)$, $b(\cdot)$ and $c(\cdot)$ represent model-dependent functions that may take the non-linear state as their inputs while $A_n(\cdot)$, $B_n(\cdot)$, $C_n(\cdot)$ are matrices that may also depend on $\mathbf{x}^N$. $\mathbf{u}_n$, $\boldsymbol{\nu}_n$ and $\boldsymbol{\epsilon}_n$ are independent zero-mean Gaussian disturbances with covariances C_u, C_ν and C_ϵ.

In the sequel, the illustration of the Rao-Blackwellisation method will be in a probabilistic setting as per [Cappé et al., 2007]. Assuming that it is the marginal distribution $p(\mathbf{x}_{0:n}^N, \mathbf{x}_n^L|\mathbf{y}_{0:n})$ at t_n that is of interest and $p(\mathbf{x}_{0:n-1}^N, \mathbf{x}_{n-1}^L|\mathbf{y}_{0:n-1})$ is available from last time instant t_n. More precisely, suppose that we have, before receiving measurement $\mathbf{y}_n$,

$$p(\mathbf{x}_{0:n-1}^N|\mathbf{y}_{0:n-1}) \approx \sum_{i=1}^{N_p} \omega_{n-1}^{(i)} \delta_{\mathbf{x}_{0:n-1}^{N,(i)}}(\mathbf{x}_{0:n-1}^N) \tag{2.44}$$

estimated by a particle filter and

$$p(\mathbf{x}_{n-1}^L|\mathbf{x}_{0:n-1}^{N,(i)}, \mathbf{y}_{0:n-1}) = \mathcal{N}(\mathbf{x}_{n-1}^L|\mu_{n-1|n-1}^{(i)}, P_{n-1|n-1}^{(i)}) \tag{2.45}$$

obtained from Kalman filters running on each particle. As the idea is to use only the particle filter to target the posterior distribution of the non-linear state, it is intuitive to think that a prediction-correction recursion similar to (2.17) and (2.19) needs to be applied. However, because of the inclusion of $\mathbf{x}_n^L$ in the state-space equation of the non-linear state (2.41) a transition density for $\mathbf{x}^N$ independent of the linear part has to be devised. Thanks to the linear dependency on $\mathbf{x}_n^L$, this may be achieved analytically via

$$p(\mathbf{x}_n^N|\mathbf{x}_{0:n-1}^N, \mathbf{y}_{0:n-1}) = \int p(\mathbf{x}_n^N|\mathbf{x}_{n-1}^N, \mathbf{x}_n^L)p(\mathbf{x}_n^L|\mathbf{x}_{0:n-1}^N, \mathbf{y}_{0:n-1})d\mathbf{x}_n^L \tag{2.46}$$

where the second density in the integral can be thought of as a predictive distribution given by a Kalman filter. Note that the resulting transition density of $\mathbf{x}_n^N$ is no longer Markovian, as in the first line of (2.17). Similarly, to update the posterior of $\mathbf{x}_{n+1}^N$, one has to compute $p(\mathbf{y}_n|\mathbf{x}_{0:n}^N, \mathbf{y}_{0:n-1})$ instead of $p(\mathbf{y}_n|\mathbf{x}_n^N)$. Owing to the conditionally linear Gaussian structure in (2.43), the desired quantity can be evaluated via

$$p(\mathbf{y}_n|\mathbf{x}^N_{0:n}, \mathbf{y}_{0:n-1}) = \int p(\mathbf{y}_n|\mathbf{x}^N_{0:n}, \mathbf{x}^L_n)p(\mathbf{x}^L_n|\mathbf{x}^N_{0:n}, \mathbf{y}_{0:n-1})d\mathbf{x}^L_n \tag{2.47}$$

Again, the second component in the above integral turns out to be a Gaussian and can be obtained within the Kalman filtering framework. A detailed discussion regarding the constructions of this density as well as the one in (2.46) can be found in the subsequent derivations.

Taken together, the above results can be used to compute new unnormalised importance weights at time t_n

$$\begin{aligned}\tilde{\omega}^{(i)}_n &= \frac{p(\mathbf{x}^{N,(i)}_{0:n})|\mathbf{y}_{0:n})}{q(\mathbf{x}^{N,(i)}_{0:n}|\mathbf{y}_{0:n})}\\ &= \frac{p(\mathbf{x}^{N,(i)}_{0:n-1}|\mathbf{y}_{0:n-1})}{q(\mathbf{x}^{N,(i)}_{0:n-1}|\mathbf{y}_{0:n-1})}\frac{p(\mathbf{y}_n|\mathbf{x}^{N,(i)}_{0:n}, \mathbf{y}_{0:n-1})p(\mathbf{x}^{N,(i)}_n|\mathbf{x}^{N,(i)}_{0:n-1}, \mathbf{y}_{0:n-1})}{q(\mathbf{x}^{N,(i)}_n|\mathbf{x}^{N,(i)}_{0:n-1}, \mathbf{y}_{0:n})p(\mathbf{y}_n|\mathbf{y}_{0:n-1})}\\ &\propto \frac{\omega^{(i)}_{n-1}}{\upsilon^{(i)}_{n-1}} \times \frac{p(\mathbf{y}_n|\mathbf{x}^{N,(i)}_{0:n}, \mathbf{y}_{0:n-1})p(\mathbf{x}^{N,(i)}_n|\mathbf{x}^{N,(i)}_{0:n-1}, \mathbf{y}_{0:n-1})}{q(\mathbf{x}^{N,(i)}_n|\mathbf{x}^{N,(i)}_{0:n-1}, \mathbf{y}_{0:n})}\end{aligned} \tag{2.48}$$

with which the new posterior for the non-linear component, i.e. $p(\mathbf{x}^N_{0:n}|\mathbf{y}_{0:n})$, can be estimated. A bootstrap RBPF can thus be obtained by setting $\omega_{n-1} = \upsilon_{n-1}$ and $q(\mathbf{x}^N_n|\mathbf{x}^N_{0:n-1}, \mathbf{y}_{0:n}) = p(\mathbf{x}^N_n|\mathbf{x}^N_{0:n-1}, \mathbf{y}_{0:n-1})$.

Now that we have shown how the inference on non-linear subset of the state can be done using particle filters, an estimation scheme used to approximate the posterior distribution of $\mathbf{x}^L_n$ will be given in the sequel. Recall that we have already obtained $p(\mathbf{x}^L_{n-1}|\mathbf{x}^N_{0:n-1}, \mathbf{y}_{0:n-1}) = \mathcal{N}(\mathbf{x}^L_{n-1}|\mu_{n-1|n-1}, P_{n-1|n-1})$ (with superscript (i) omitted for the ease of presentation) for each particle, a predictive distribution can be readily computed according to (2.42):

$$\begin{aligned}p(\mathbf{x}^L_n|\mathbf{x}^N_{0:n-1}, \mathbf{y}_{0:n-1}) &= \int p(\mathbf{x}^L_n|\mathbf{x}^L_{n-1}, \mathbf{x}^N_{0:n-1})p(\mathbf{x}^L_{n-1}|\mathbf{x}^N_{0:n-1}, \mathbf{y}_{0:n-1})d\mathbf{x}^L_{n-1}\\ &= \mathcal{N}(\mathbf{x}^L_n|\mu_{n|n-1}, P_{n|n-1})\end{aligned} \tag{2.49}$$

where the mean and covariance $\mu_{n|n-1}$ and $P_{n|n-1}$ are given by the standard Kalman filter prediction step [Anderson and Moore, 1979]. This is also the distribution used to obtain the transition density of $\mathbf{x}^N$ in (2.46). However, when it comes to the correction for the posterior distribution

$$p(\mathbf{x}^L_n|\mathbf{x}^N_{0:n}, \mathbf{y}_{0:n}) \propto p(\mathbf{y}_n|\mathbf{x}^N_{0:n}, \mathbf{x}^L_n)p(\mathbf{x}^L_n|\mathbf{x}^N_{0:n}, \mathbf{y}_{0:n-1}) \tag{2.50}$$

and the evaluation of the likelihood function in (2.47), it should be noted that

$$p(\mathbf{x}_n^L|\mathbf{x}_{0:n}^N, \mathbf{y}_{0:n-1}) \neq p(\mathbf{x}_n^L|\mathbf{x}_{0:n-1}^N, \mathbf{y}_{0:n-1})$$

This implies that an additional update step is required in order to incorporate $\mathbf{x}_n^N$ into the condition. This can be achieved by using (2.41) since this transition equation can be regarded as a linear and Gaussian pseudo-measurement model for $\mathbf{x}_n^L$ conditioned on $\mathbf{x}_{0:n-1}^N$. Consequently, we have

$$\begin{aligned} p(\mathbf{x}_n^L|\mathbf{x}_{0:n}^N, \mathbf{y}_{0:n-1}) &\propto p(\mathbf{x}_n^N|\mathbf{x}_{n-1}^N, \mathbf{x}_n^L)p(\mathbf{x}_n^L|\mathbf{x}_{0:n-1}^N, \mathbf{y}_{0:n-1}) \\ &= \mathcal{N}(\mathbf{x}_n^L|\mu'_{n|n-1}, P'_{n|n-1}) \end{aligned} \tag{2.51}$$

with which the desired Kalman coorection step (2.50) can be performed:

$$\begin{aligned} p(\mathbf{x}_n^L|\mathbf{x}_{0:n}^N, \mathbf{y}_{0:n}) &\propto p(\mathbf{y}_n|\mathbf{x}_{0:n}^N, \mathbf{x}_n^L)p(\mathbf{x}_n^L|\mathbf{x}_{0:n}^N, \mathbf{y}_{0:n-1}) \\ &= \mathcal{N}(\mathbf{x}_n^L|\mu_{n|n}, P_{n|n}) \end{aligned} \tag{2.52}$$

Note again operations (2.49) - (2.52) have to be done for each particle. Now we are in a position to estimate the posterior distribution of the linear state according to the so-called Rao-Blackwellised estimation scheme [Robert and Casella, 2004]:

$$\begin{aligned} p(\mathbf{x}_n^L|\mathbf{y}_{0:n}) &= \int p(\mathbf{x}_n^L|\mathbf{x}_{0:n}^N, \mathbf{y}_{0:n})p(\mathbf{x}_{0:n}^N|\mathbf{y}_{0:n})d\mathbf{x}_{0:n}^N \\ &\approx \sum_{i=1}^{N_p} \omega_n^{(i)} p(\mathbf{x}_n^L|\mathbf{x}_{0:n}^{N,(i)}, \mathbf{y}_{0:n}) \\ &= \sum_{i=1}^{N_p} \omega_n^{(i)} \mathcal{N}(\mathbf{x}_n^L|\mu_{n|n}^{(i)}, P_{n|n}^{(i)}) \end{aligned} \tag{2.53}$$

Other than state estimation, RBPF is also useful for solving Bayesian identification (parameter estimation) problems. This normally requires either parameters or the hidden state to be linear and Gaussian such that a mixed linear/nonlinear state space model to which RBPF can be applied may be formulated [Lindsten, 2011]. Another usage of RBPFs can be found in inference problems on variable rate models [Christensen et al., 2012; Morelande and Gordon, 2009; Zhang and Godsill, 2016]. A Langevin dynamics based jump-diffusion model is considered in [Christensen et al., 2012] to model the behaviour of high frequency asset price data. The authors design a RBPF to estimate unknown jump times conditioned on which the model is linear and Gaussian. In a similar spirit, [Zhang and Godsill, 2016] uses a RBPF to detect pitch periods of varying lengths in speech signals while [Morelande and Gordon, 2009] demonstrate the effectiveness of a similar approach for target tracking applications where a target's motion is controlled by piecewise constant manoeuvres.

Furthermore, it is worth noting that despite of providing improved estimation results, RBPF may require much higher computational power than the standard particle filter since each particle is now associated with a Kalman filter. Therefore, in practice a trade-off between the computational cost and the performance has to be made. For a detailed complexity analysis about RBPFs, see [Karlsson et al., 2005].

2.3.6 Particle Smoothing

The introduction about particle filters has hitherto been focused on the joint filtering distribution $p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})$ as well as its marginal distribution $p(\mathbf{x}_n|\mathbf{y}_{0:n})$; indeed, the distribution $p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})$ can be thought of as a smoothing distribution [Kitagawa, 1996], but its efficacy is limited by the degeneracy problem. It is common that after several resampling steps most particles will share the same ancestor, resulting in an almost fixed smoothing trajectory. As a result, this smoothing distribution should be considered mainly for time-critical applications or when a smoothing distribution $p(\mathbf{x}_{n-L:n}|\mathbf{y}_{0:n})$ with small lag L is of interest.

Various particle smoothing algorithms have been proposed to better approximate either the marginal smoothing distribution $p(\mathbf{x}_n|\mathbf{y}_{0:N})$ or the joint smoothing distribution $p(\mathbf{x}_{0:N}|\mathbf{y}_{0:N})$.

When the marginal distribution is targeted, a backward recursive algorithm can be designed using the following Bayesian smoothing formula [Kitagawa, 1987]:

$$p(\mathbf{x}_n|\mathbf{y}_{0:N}) = p(\mathbf{x}_n|\mathbf{y}_{0:n}) \int \frac{p(\mathbf{x}_{n+1}|\mathbf{x}_n)p(\mathbf{x}_{n+1}|\mathbf{y}_{0:N})}{p(\mathbf{x}_{n+1}|\mathbf{y}_{0:n})} d\mathbf{x}_{n+1} \tag{2.54}$$

If the system is linear, the resulting algorithm will be the renowned Rauch-Tung-Striebel (RTS) smoother (or Kalman smoother) [Rauch et al., 1965; Särkkä, 2013]. Otherwise, a weighted approximation of the desired marginal can be obtained based on a forward particle filter. More specifically, by re-arranging (2.54) we have

$$p(\mathbf{x}_n|\mathbf{y}_{0:N}) = \int p(\mathbf{x}_n|\mathbf{x}_{n+1}, \mathbf{y}_{0:n})p(\mathbf{x}_{n+1}|\mathbf{y}_{0:N}) d\mathbf{x}_{n+1} \tag{2.55}$$

where the first component in the integral can be estimated using the filtering distribution $p(\mathbf{x}_n|\mathbf{y}_{0:n}) = \sum_{i=1}^{N_p} \omega_n^{(i)} \delta_{\mathbf{x}_n^{(i)}}(\mathbf{x}_n)$ obtained at t_n while the marginalisation can be approached in a similar manner as in the prediction step of particle filter, i.e.

(2.18). This forward-backward algorithm was firstly introduced in [Doucet et al., 2000b; Hürzeler and Künsch, 1998]. Another popular solution to the same smoothing problem is based on the "two filter" formula [Bresler, 1986; Briers et al., 2009; Fearnhead et al., 2010; Kitagawa, 1994] where the smoothing density is normally factorised as

$$p(\mathbf{x}_n|\mathbf{y}_{1:N}) \propto p(\mathbf{y}_{n:N}|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{0:n-1}) \tag{2.56}$$

where the predictive density $p(\mathbf{x}_n|\mathbf{y}_{0:n-1})$ can be again provided by a particle filter while the first term is found by an information filter running backwards in time.

As for the estimation of the joint smoothing distribution as a whole, consider the following construction of the joint distribution [Godsill et al., 2004; Hürzeler and Künsch, 2001]:

$$\begin{aligned} p(\mathbf{x}_{1:N}|\mathbf{y}_{1:N}) &= p(\mathbf{x}_N|\mathbf{y}_{0:N}) \prod_{n=0}^{N-1} p(\mathbf{x}_n|\mathbf{x}_{n+1:N}, \mathbf{y}_{0:N}) \\ &= p(\mathbf{x}_N|\mathbf{y}_{0:N}) \prod_{n=0}^{N-1} p(\mathbf{x}_n|\mathbf{x}_{n+1}, \mathbf{y}_{0:n}) \end{aligned} \tag{2.57}$$

where the second line is achieved by assuming a Markov system (see Fig. 2.1). This factorisation suggests that we may sample from the joint distribution using the Forward-Filtering-Backward-Sampling (FFBS) technique [Carter and Kohn, 1994; Frühwirth-Schnatter, 1994]. Sampling from the filtering distribution $p(\mathbf{x}_N|\mathbf{y}_{0:N})$ is straightforward given the particle filter output at the last time step. However, in order to draw the rest states we also need to sample from the conditional distributions. To this end, factor the distribution according to the Bayes' rule

$$p(\mathbf{x}_n|\mathbf{x}_{n+1}, \mathbf{y}_{0:n}) \propto p(\mathbf{x}_n|\mathbf{y}_{0:n})p(\mathbf{x}_{n+1}|\mathbf{x}_n)$$

This expression admits a similar form as that of the Bayesian filtering update step, with the likelihood function $p(\mathbf{y}_n|\mathbf{x}_n)$ replaced by the transition density. Consequently, a weighted approximation for the conditional distribution can be obtained with adjusted particle weights. The main issue with this FFBS based smoothing scheme is that N_s repetitions, each for one realisation of the state trajectories, are required, which implies a complexity of $\mathcal{O}(N \times N_s \times N_p)$. Improvements over this method may be achieved by using MCMC or rejection sampling in the backward pass, as in [Bunch and Godsill, 2013a; Douc et al., 2011]. Furthermore, Rao-Blackwellised smoother can be designed based on this generic simulation scheme when the system have analytically tractable substructures, see [Lindsten et al., 2016].

Finally, note that the particle smoothing problem can also be tackled using particle MCMC methods [Andrieu et al., 2010; Middleton et al., 2019] where the joint smoothing distribution $p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})$ is targeted by MCMC samplers.

2.3.7 Mixing MCMC with IS

It is very likely in practice that the chosen importance distribution $q(\mathbf{x}_{0:n})$ is significantly different from the target distribution $p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})$ and hence resampling procedures need to be invoked frequently. This scenario is known as sample impoverishment, which reflects the lack of diversity among the particle population (i.e. many particles will share the same trajectory after resampling). One popular way for solving this problem is to set $p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})$ as the stationary distribution of a Markov chain transition kernel $K(\mathbf{x}^*_{0:n}|\mathbf{x}_{0:n})$. In this way, identical particles can be moved to different positions in the state space and thus improvements in terms of estimation accuracy can be achieved. Based on this basic idea, many improvement schemes have been proposed, as in [Berzuini et al., 1997; Gilks and Berzuini, 2001; Godsill and Clapp, 2001; MacEachern et al., 1999; Marques et al., 2018]. Particularly, methods in which Markov chain moves are conducted following a resampling step are normally referred to as "resample-move" (RM), as introduced by [Gilks and Berzuini, 2001].

In fact, MCMC moves may be freely mixed with SIS at any interior stages. Suppose $\{\mathbf{x}^{(i)}_{0:n}, \omega^{(i)}_n\}_{i=1,\ldots,N_p}$ is the collection of particles and for each particle a new state trajectory $\mathbf{x}^*_{0:n}$ is drawn from a MCMC transition kernel $K(\mathbf{x}^*_{0:n}|\mathbf{x}_{0:n})$ with $p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})$ as its stationary distribution. It is also assumed that the kernel satisfies Detailed Balance condition. Consequently, we are working in an extended state space with $\bar{p}(\mathbf{x}^*_{0:n}, \mathbf{x}_{0:n}|\mathbf{y}_{0:n}) = p(\mathbf{x}^*_{0:n}|\mathbf{y}_{0:n})K(\mathbf{x}_{0:n}|\mathbf{x}^*_{0:n})$ [Moral et al., 2006] and the expectation of some function $\phi(\cdot)$ of interest can be shown to be

$$\begin{aligned}
\mathbb{E}\{\phi\} &= \int \phi(\mathbf{x}^*_{0:n})p(\mathbf{x}^*_{0:n}|\mathbf{y}_{0:n})d\mathbf{x}^*_{0:n} \\
&= \int\int \phi(\mathbf{x}^*_{0:n})\bar{p}(\mathbf{x}^*_{0:n}, \mathbf{x}_{0:n}|\mathbf{y}_{0:n})d\mathbf{x}_{0:n}d\mathbf{x}^*_{0:n} \\
&= \int\int \phi(\mathbf{x}^*_{0:n})\frac{p(\mathbf{x}^*_{0:n}|\mathbf{y}_{0:n})K(\mathbf{x}_{0:n}|\mathbf{x}^*_{0:n})}{q(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})K(\mathbf{x}^*_{0:n}|\mathbf{x}_{0:n})}q(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})K(\mathbf{x}^*_{0:n}|\mathbf{x}_{0:n})d\mathbf{x}_{0:n}d\mathbf{x}^*_{0:n} \\
&= \int\int \phi(\mathbf{x}^*_{0:n})\frac{p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})K(\mathbf{x}^*_{0:n}|\mathbf{x}_{0:n})}{q(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})K(\mathbf{x}^*_{0:n}|\mathbf{x}_{0:n})}q(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})K(\mathbf{x}^*_{0:n}|\mathbf{x}_{0:n})d\mathbf{x}_{0:n}d\mathbf{x}^*_{0:n} \\
&= \int\int \phi(\mathbf{x}^*_{0:n})p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})K(\mathbf{x}^*_{0:n}|\mathbf{x}_{0:n})d\mathbf{x}_{0:n}d\mathbf{x}^*_{0:n} \\
&\approx \sum_{i=1}^{N_p} \omega_n^{(i)}\phi(\mathbf{x}^{*,(i)}_{0:n}) \qquad (2.58)
\end{aligned}$$

by noticing that the following joint distribution can be obtained within the same IS framework

$$p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})K(\mathbf{x}^*_{0:n}|\mathbf{x}_{0:n}) \approx \sum_{i=1}^{N_p} \omega_n^{(i)}\delta_{\mathbf{x}^{(i)}_{0:n},\mathbf{x}^{*,(i)}_{0:n}}(\mathbf{x}_{0:n}, \mathbf{x}^*_{0:n}) \qquad (2.59)$$

Equation (2.58) indicates that we may retain the computed importance weights after moving particles around according to some MCMC kernels, as also reported by [Bunch, 2014; MacEachern et al., 1999]. Alternatively, a re-weighting procedure before the resampling stage may be carried out by introducing some artificial backward kernel $\alpha(\mathbf{x}_{0:n}|\mathbf{x}^*_{0:n})$. This is known as the "move-reweighting" scheme as in [Marques et al., 2018]. In the subsequent chapters, it will be shown that better inference schemes can be designed by exploiting the complementary relationship between MCMC sampling methods and particle filters.

2.4 MCMC-based Sequential Monte Carlo

Hitherto, the introduced SMC methods are mainly based on importance sampling (IS). Provided the fact that MCMC algorithms is very powerful and flexible in contrast to IS when dealing with systems of high dimensions and distributions of high complexity, we have a good reason to develop sequential algorithms that take advantage of MCMC for tracking evolving distributions, e.g. $p(\mathbf{x}_n|\mathbf{y}_{0:n})$ and $p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})$, as with particle filters.

One early attempt of this kind can be found in [Berzuini et al., 1997] where the state posterior of interest is $p(\mathbf{x}_{n-1}, \mathbf{x}_n|\mathbf{y}_{0:n})$. The authors split the joint posterior into two full conditionals, namely $p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_{0:n})$ and $p(\mathbf{x}_{n-1}|\mathbf{x}_n, \mathbf{y}_{0:n})$ (corresponding to

$\pi(\Phi_{t+1}|\epsilon_t, E_t, F_{t+1})$ and $\pi(\epsilon_t|\Phi_{t+1}, D_t, E_t, F_{t+1})$ in the original paper), each of which is targeted by a MH algorithm. The idea is to simulate both $\mathbf{x}_{n-1}$ and $\mathbf{x}_n$ under the guidance of the incoming observation $\mathbf{y}_n$. Because of the use of the MH algorithm for sampling from the full conditionals which do not admit closed-form representations, the resulting algorithm may be thought of as an example of "Metropolis-within-Gibbs" [Casella and Robert, 1996; Müller, 1991, 1993]. The principle used here is also closely related to that in the auxiliary particle filter (see Section. 2.3.4). In fact, the authors of [Pitt and Shephard, 1999] did mention that instead of using approximations in the SIR procedure, one may instead apply a MH kernel targeting $g(\alpha_{t+1}, k|Y_{t+1})$, where α_{t+1} corresponds to $\mathbf{x}_n$ here while the auxiliary variable k is related to the selection of $\mathbf{x}_{n-1}$ (or $\mathbf{x}_{0:n-1}$).

Another related approach is the aforementioned resample-move (RM) algorithm. In that case, a MCMC move is applied to each resampled particle i with the stationary distribution

$$p(\mathbf{x}_n, \mathbf{x}_{0:n-1}^{(i)}|\mathbf{y}_{0:n}) \propto p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{x}_{n-1}^{(i)})p(\mathbf{x}_{0:n-1}^{(i)}|\mathbf{y}_{0:n})$$

The resulting acceptance ratio can be shown as follows

$$\rho_{RM}(\mathbf{x}_n^*, \mathbf{x}_n^{m-1}) = 1 \wedge \frac{p(\mathbf{y}_n|\mathbf{x}_n^*)p(\mathbf{x}_n^*|\mathbf{x}_{n-1}^{(i)})q(\mathbf{x}_n^{m-1}|\mathbf{x}_{n-1}^{(i)})}{p(\mathbf{y}_n|\mathbf{x}_n^{m-1})p(\mathbf{x}_n^{m-1}|\mathbf{x}_{n-1}^{(i)})q(\mathbf{x}_n^*|\mathbf{x}_{n-1}^{(i)})} \tag{2.60}$$

with q the proposal and m the iteration index. Suggested by [Crisan and Doucet, 2002; Gilks and Berzuini, 2001], for RM it is not necessary to have a burn-in period as in most MCMC applications owing to the fact that the particle filter has already provided a reasonable approximation to the true posterior.

While the approaches discussed so far are more or less focusing on using MCMC moves to diversify particles within IS-based SMC methods, a MCMC-based particle filter is presented in [Khan et al., 2004; Khan et al., 2005; Septier et al., 2009] for tracking multiple targets. Specifically, by supposing that $X_n = \{\mathbf{x}_{n,1}^T, \mathbf{x}_{n,2}^T, \ldots, \mathbf{x}_{n,k}^T\}^T$ is a vector consisting of states of k targets, the joint distribution of interest at t_{n-1} can be represented by N_p unweighted samples as follows,

$$p(X_{n-1}|\mathbf{y}_{0:n-1}) \approx \frac{1}{N_p}\sum_{i=1}^{N_p} \delta_{X_{n-1}^{(i)}}(X_{n-1})$$

According to the Bayesian filtering recursion, the predictive and the posterior distribution at the new time step are given by

$$\begin{aligned} p(X_n|\mathbf{y}_{0:n-1}) &= \int p(X_n|X_{n-1})p(X_{n-1}|\mathbf{y}_{0:n-1})dX_{n-1} \approx \frac{1}{N_p}\sum_{i=1}^{N_p} p(X_n|X_{n-1}^{(i)}) \\ p(X_n|\mathbf{y}_{0:n}) &\propto p(\mathbf{y}_n|X_n)p(X_n|\mathbf{y}_{0:n-1}) \end{aligned}$$

Subsequently, a MH chain can be constructed readily with $p(X_n|\mathbf{y}_{0:n})$ as its stationary distribution and the associated acceptance ratio can be obtained as

$$\rho(X_n^*, X_n^{m-1}) = 1 \wedge \frac{p(X_n^*|\mathbf{y}_{0:n})q(X_n^{m-1}|X_n^*)}{p(X_n^{m-1}|\mathbf{y}_{0:n})q(X_n^*|X_n^{m-1})} \tag{2.61}$$

It was shown in the papers that this scheme outperformed the IS-based SMC counterpart in multiple target tracking scenarios, demonstrating the strong potential of the MCMC-based sequential Monte Carlo methods. However, in this MCMC algorithm the summation $\sum_{i=1}^{N_p} p(X_n^*|X_{n-1}^{(i)})$ needs to be computed every time a new proposal of X^* is made, implying a complexity of $\mathcal{O}(N_p)$ at each MCMC iteration. Hence, the authors suggested at each iteration one may only update the state $\mathbf{x}_{n,j}^*$ of a particular target j such that many terms in (2.61) remain unchanged and thus need not to be calculated. Similar ideas were also explored in [Golightly and Wilkinson, 2006] where the authors proposed to use a "simulation filter" based purely on MCMC to tackle the Bayesian missing data problem in a sequential fashion. In their work, the stationary distribution of the MH chain is set to

$$\begin{aligned} &p(\mathbf{x}_{n-1}, \mathbf{x}_n, \theta, X_{(t_{n-1},t_n)}|\mathbf{y}_{0:n}) \\ &\quad \propto p(\mathbf{x}_{n-1}, \theta|\mathbf{y}_{0:n-1})p(X_{(t_{n-1},t_n)}|\mathbf{x}_{n-1}, \theta)p(\mathbf{y}_n|\mathbf{x}_n, X_{(t_{n-1},t_n)}, \theta) \end{aligned}$$

with θ being the system parameters and $X_{(t_{n-1},t_n)}$ the interpolated (missing) hidden state values in the time interval (t_{n-1}, t_n). If the proposal for $\mathbf{x}_{n-1}$ and θ is made according to the posterior distribution estimated from the last time step, it can be shown that the resulting acceptance ratio is simplified to

$$\begin{aligned} \rho = 1 \wedge &\frac{p(X_{(t_{n-1},t_n)}^*|\mathbf{x}_{n-1}^*, \theta^*)p(\mathbf{y}_n|\mathbf{x}_n^*, X_{(t_{n-1},t_n)}^*, \theta^*)}{p(X_{(t_{n-1},t_n)}^{m-1}|\mathbf{x}_{n-1}^{m-1}, \theta^{m-1})p(\mathbf{y}_n|\mathbf{x}_n^{m-1}, X_{(t_{n-1},t_n)}^{m-1}, \theta^{m-1})} \\ &\times \frac{q(\mathbf{x}_n^{m-1}, X_{(t_{n-1},t_n)}^{m-1}|\mathbf{x}_n^*, X_{(t_{n-1},t_n)}^*)}{q(\mathbf{x}_n^*, X_{(t_{n-1},t_n)}^*|\mathbf{x}_n^{m-1}, X_{(t_{n-1},t_n)}^{m-1})} \end{aligned}$$

After making an enough number of draws, the posterior of interest, that is $p(\mathbf{x}_n, \theta|\mathbf{y}_{0:n})$, can be obtained easily by discarding $X_{[t_{n-1},t_n)}$ in the resulting empirical distribution.

2.4.1 Sequential MCMC

Here a MCMC-based sequential Monte Carlo framework based on which sequential Bayesian inference methods can be designed will be discussed, following the formulation in [Mihaylova et al., 2014; Pang et al., 2011a, 2008, 2011b; Septier and Peters, 2016]. This framework incorporates the many useful, but previously separated, features of the online MCMC schemes which are mentioned above. As a result, it is quite generic and can be preferable when dealing with certain complicated state space models in comparison to commonly used IS-based SMC algorithms. In the sequel, this framework will be abbreviated as SMCMC. As in most filtering applications, at t_n we would like to estimate the joint distribution $p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})$ (or its marginal $p(\mathbf{x}_n|\mathbf{y}_{0:n})$) for which direct sampling is not possible. In contrast to IS-based SMC methods, the distribution is approximated using a set of unweighted particles in SMCMC:

$$\begin{aligned} p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n}) &\approx \hat{p}(\mathbf{x}_{0:n}|\mathbf{y}_{0:n}) \\ &= \frac{1}{N_p}\sum_{i=1}^{N_p} \delta_{\mathbf{x}_{0:n}^{(i)}}(\mathbf{x}_{0:n}) \end{aligned} \tag{2.62}$$

In the most general case, SMCMC follows a mixture sampling based procedure that selects a joint update with probability P_J which updates all states simultaneously and refinement steps with probability $1 - P_J$ in which states are updated individually. For detailed justifications on the use of such hybrid MCMC strategies, see [Tierney, 1994].

Joint Update

In a similar spirit to that of [Golightly and Wilkinson, 2006; Pitt and Shephard, 1999], we may jointly update $\mathbf{x}_n$ and $\mathbf{x}_{0:n-1}$ via a MH kernel with the following acceptance ratio at the m-th iteration

$$\rho_{joint}(\mathbf{x}_{0:n}^{*}, \mathbf{x}_{0:n}^{m-1}) = 1 \wedge \frac{p(\mathbf{x}_{0:n}^{*}|\mathbf{y}_{0:n})q(\mathbf{x}_n^{m-1}, \mathbf{x}_{0:n-1}^{m-1}|\mathbf{x}_n^{*}, \mathbf{x}_{0:n-1}^{*})}{p(\mathbf{x}_{0:n}^{m-1}|\mathbf{y}_{0:n})q(\mathbf{x}_n^{*}, \mathbf{x}_{0:n-1}^{*}|\mathbf{x}_n^{m-1}, \mathbf{x}_{0:n-1}^{m-1})} \tag{2.63}$$

The joint update aims to move particles in the light of all observations up to t_n. In this way the probability mass of the posterior may be better approximated. Recall that in the MCMC-based particle filter [Khan et al., 2005] the computational inefficiency arises in the direct Monte Carlo computations at each iteration. This problem is avoided here by targeting the joint distribution, leading to an more efficient online inference algorithm. Moreover, this algorithm may be thought of as a MCMC implementation of the auxiliary particle filter [Pang et al., 2011a].

To give a concrete (simple) example, consider a HMM where the intractable joint posterior can be factorised as

$$p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n}) \propto p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{x}_{n-1})p(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1})$$

As with the classic bootstrap particle filter the state transition density $p(\mathbf{x}_n|\mathbf{x}_{n-1})$ can be used as the proposal for $\mathbf{x}_n$ while for the proposal of $\mathbf{x}_{0:n-1}$ the discrete empirical distribution from the last time step is considered, that is,

$$\begin{aligned} q(\mathbf{x}_n|\mathbf{x}_n^{m-1}) &= p(\mathbf{x}_n|\mathbf{x}_{n-1}^{m-1}) \\ q(\mathbf{x}_{0:n-1}|\mathbf{x}_{0:n-1}^{m-1}) &= \hat{p}(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1}) \end{aligned} \tag{2.64}$$

In this case, the acceptance ratio (2.63) depends only on the observation density function

$$\rho_{joint}(\mathbf{x}_{0:n}^*, \mathbf{x}_{0:n}^{m-1}) = 1 \wedge \frac{p(\mathbf{y}_n|\mathbf{x}_n^*)}{p(\mathbf{y}_n|\mathbf{x}^{m-1})}$$

Individual Refinements

Coupled with the joint update, a series of individual refinement steps can be carried out. The theoretical underpinnings are provided by the Gibbs sampling framework where draws from a target distribution $p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})$ are generated by cycling over its sub-conditionals. These refinement steps allow us to explore the structured state space, which may increase the effectiveness of the algorithm. For the Markov system discussed above, a simple configuration for such sub-conditionals can be:

$$\hat{p}(\mathbf{x}_{0:n-1}|\mathbf{x}_n^m, \mathbf{y}_{0:n}) \propto \frac{1}{N_p}\sum_{i=1}^{N_p} p(\mathbf{y}_n|\mathbf{x}_n^m)p(\mathbf{x}_n^m|\mathbf{x}_{n-1}^{(i)})\delta_{\mathbf{x}_{0:n-1}^{(i)}}(\mathbf{x}_{0:n-1}) \tag{2.65}$$

$$\hat{p}(\mathbf{x}_n|\mathbf{x}_{0:n-1}^m, \mathbf{y}_{0:n}) \propto \frac{1}{N_p} p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{x}_{n-1}^m) \tag{2.66}$$

as suggested in [Godsill, 2019b]. Whereas sampling from the discrete distribution (2.65) can be done exactly, we need to resort to Metropolis-within-Gibbs since direct sampling from the full conditional (2.66) is not possible. The acceptance ratio for the

Metropolis-within-Gibbs step can be expressed as

$$\rho(\mathbf{x}_n^*, \mathbf{x}_n^{m-1}) = 1 \wedge \frac{p(\mathbf{x}_n^*|\mathbf{x}_{0:n-1}^m, \mathbf{y}_{0:n})q(\mathbf{x}_n^{m-1}|\mathbf{x}_n^*)}{p(\mathbf{x}_n^{m-1}|\mathbf{x}_{0:n-1}^m, \mathbf{y}_{0:n})q(\mathbf{x}_n^*|\mathbf{x}_n^{m-1})} \tag{2.67}$$

Again, the use of the proposal in (2.64) can lead to a simplified expression for the acceptance ratio in the refinement step. Note that we may consider using a MH algorithm to draw samples from the conditional distribution $p(\mathbf{x}_{0:n-1}|\mathbf{x}_n, \mathbf{y}_{0:n})$ in cases where it is expensive to obtain (2.65) (see [Pang et al., 2011a; Septier et al., 2009] for examples). As mentioned before, this Gibbs-based refinement procedure is very similar to that of [Berzuini et al., 1997], though in that case the authors did not employ a mixture sampling scheme, i.e. the joint update step which can increase the mobility of the algorithm is missing. Furthermore, it is worth noting that in multi-target tracking applications the refinements may be conducted at a finer scale. More precisely, rather than updating the state vector as a whole at a time instant we can progressively refine subsets of the state vector that correspond to a single or multiple targets [Mihaylova et al., 2014; Pang et al., 2011b].

Algorithm 4 gives the implementation steps of a generic SMCMC algorithm. Similar to all MCMC algorithms, the first N_{burn} samples at each iteration will be discarded in order to allow the chain to mix adequately. Additionally, chain thinning can be used to reduce correlation between samples. This is achieved by discarding all but every N_{thin} samples. Also, as with most probabilistic numerical methods, the use of log scale is recommended when calculating acceptance ratios.

```
Initialise particle set $\{\mathbf{x}_{-1}^{(i)}\}_{i=1}^{N_p}$ ;
for $n = 1, 2, \ldots, N$ do
    for $m = 1, 2, \ldots, N_{iter}$ do
        Sample $u \sim \mathrm{Unif}(0, 1)$;
        if $u < P_J$ then
            // Joint Update
            Sample $\{\mathbf{x}_n^*, \mathbf{x}_{0:n-1}^*\} \sim q(\mathbf{x}_n, \mathbf{x}_{0:n-1}|\mathbf{x}_n^{m-1}, \mathbf{x}_{0:n-1}^{m-1})$ ;
            Compute the MH acceptance ratio:
                $$\rho_{joint} = 1 \wedge \frac{p(\mathbf{x}_{0:n}^*|\mathbf{y}_{0:n})q(\mathbf{x}_n^{m-1}, \mathbf{x}_{0:n-1}^{m-1}|\mathbf{x}_n^*, \mathbf{x}_{0:n-1}^*)}{p(\mathbf{x}_{0:n}^{m-1}|\mathbf{y}_{0:n})q(\mathbf{x}_n^*, \mathbf{x}_{0:n-1}^*|\mathbf{x}_n^{m-1}, \mathbf{x}_{0:n-1}^{m-1})};$$
            Accept $\{\mathbf{x}_n^m, \mathbf{x}_{0:n-1}^m\} = \{\mathbf{x}_n^*, \mathbf{x}_{0:n-1}^*\}$ with probability $\rho_{joint}$;
        else
            // Refinements
            Sample $\mathbf{x}_{0:n-1}^* \sim \hat{p}(\mathbf{x}_{0:n-1}|\mathbf{x}_n^{m-1}, \mathbf{y}_{0:n})$ as in (2.65) ;
            Accept $\mathbf{x}_{0:n-1}^m = \mathbf{x}_{0:n-1}^*$ ;
            Sample $\mathbf{x}_n^* \sim q(\mathbf{x}_n|\mathbf{x}_n^{m-1})$ ;
            Compute the MH acceptance ratio:
                $$\rho_R = 1 \wedge \frac{p(\mathbf{x}_n^*|\mathbf{x}_{0:n-1}^m, \mathbf{y}_{0:n})q(\mathbf{x}_n^{m-1}|\mathbf{x}_n^*)}{p(\mathbf{x}_n^{m-1}|\mathbf{x}_{0:n-1}^m, \mathbf{y}_{0:n})q(\mathbf{x}_n^*|\mathbf{x}_n^{m-1})};$$
            Accept $\mathbf{x}_n^m = \mathbf{x}_n^*$ with probability $\rho_R$ ;
        end
        Keep every $N_{thin}$ MCMC output $\mathbf{x}_{0:n}^{(i)} = \mathbf{x}_{0:n}^m$ to form the new particle set
         after a burn-in period $N_{burn}$ ;
        Approximate the posterior: $\hat{p}(\mathbf{x}_{0:n}|\mathbf{y}_{0:n}) = \frac{1}{N_p}\sum_{i=1}^{N_p}\delta_{\mathbf{x}_{0:n}^{(i)}}(\mathbf{x}_{0:n})$;
    end
end
```

Algorithm 4: A general sequential MCMC algorithm

2.5 Summary

This chapter presents the basic idea of Bayesian inference and gives a thorough review of SMC methods. In addition to a detailed introduction to IS-based SMC (or particle filters) methods, a sequential Bayesian inference framework based purely on MCMC methods is discussed. While particle filtering methods have proved to be very powerful in solving non-linear/non-Gaussian filtering problems, sequential MCMC provides a promising solution for state estimation in high-dimensional systems. It is worth emphasising again that particle filtering methods and MCMC sampling schemes are in many ways complementary. This important insight has led to many improvement techniques for SMC methods over the last 25 years.

Chapter 3

Bayesian Object Tracking: a Fixed Rate Perspective

In a tracking problem we are typically concerned with the accurate extraction of useful information, such as position, velocity and orientation, of a moving object over time provided a set of noisy and possibly limited measurements. These measurements, being the observed phenomena of hidden kinematic states in the physical world, may be provided by sensors such as inertial measurement units (IMUs), odometers, wireless signal receivers, radars and GPS units. Since most objects being tracked are physical representations whose motion can be described by mathematical models, the efficacy of a tracking algorithm is closely coupled with the choice of dynamical models. Hence, an appropriately chosen motion prior which fits the target's moving pattern can be vital when designing a tracking algorithm, especially in challenging scenarios like indoor positioning. Also, in practice not only are measurements made continuously over time but they can also arrive asynchronously. This implies that a continuous-time model is more suitable since it accommodates the continuity and asynchronicity in a natural way. Moreover, the fixed rate setting where the state process is assumed synchronised with the observation process has been widely adopted in tracking applications over decades as it simplifies the modelling and implementation procedure. While this prevalent assumption is also adopted throughout this chapter, in the subsequent chapter a relaxation which allows the state process to be separated from the observation process will be made. On the other hand, a tracking algorithm that can effectively fuse information coming from multiple sources is essential and it can be found that Kalman filter-based methods and SMC methods are possible (perhaps the most popular) candidates. The intimate tie between the mathematical model and the data fusion algorithm indicates that to design a successful integrated tracking system particular attention has to be paid to these two main ingredients.

In this chapter, a fixed-rate intrinsic frame state space model, based on curvilinear motion models from kinematics, is considered. This model has appeared in a number of publications since the late 90s [Best and Norton, 1997; Bunch and Godsill, 2013c; Godsill and Vermaak, 2004, 2005; Godsill et al., 2007; Hostettler and Särkkä, 2016; Li and Jilkov, 2003] and it is a natural model for representing turn motion. However, due to the non-linearity in the state equations the use of this class of models is not as widespread as other simpler models. Given the fact that in many applications such as track cycling and vehicle monitoring inertial sensors are physically in the body frame of the object, we show here that the combination of inertial measurements[1] with the intrinsic dynamic model leads to linear equations, which can be incorporated effectively into particle filtering as well as MCMC-based sequential estimation schemes. The asynchronicity in measurements is accounted for by treating the state process in continuous time and it will also be shown that the proposed Bayesian inference methods offer a promising solution to multi-rate sensor fusion problems in which inertial and occasional position measurements are available.

The tracking algorithms, without taking into account distance measurements, have been published elsewhere [Liang and Godsill, 2018, 2019].

3.1 A Review of Models and Algorithms for Object Tracking

3.1.1 Tracking Models

As stated, research in object tracking has been active over many decades and there is a vast amount of literature focusing on the modelling of object motion. In general, these existing mathematical models can be classified into two main categories, depending on whether or not the model is coupled across the chosen spatial coordinates. It can be shown that many target models can be conveniently formulated in continuous time through stochastic differential equations (SDEs) [Särkkä, 2013; Øksendal, 2003]. Let $[x, y]^T$ be the spatial positions of a point object in a 2-dimensional Cartesian coordinate

[1] not to be confused with the sensors in an IMU. By inertial measurements here we refer to all possible sensors that are capable of giving direct (but noisy) information about the states on an intrinsic coordinate, which is not limited to the IMU sensors.

system, a SDE for the object motion along x-axis may be written as

$$d\dot{x}_t = \sigma_x dB_t^x$$

where $\dot{x} = dx/dt$ and B_t^x is a Brownian motion. When σ_x is set to a reasonably small value, which means that only a small acceleration is allowed along x (or y) direction, the object will move as if it has nearly constant velocities. Therefore, this tracking model is known as the "nearly constant velocity" (CV) model. Note that the product $\sigma_x dB_t^x$ is normally referred to as the "diffusion" part of the SDE. If we prepare to extend the state vector to include the accelerations, the basic SDE for one direction can be revised as

$$d\ddot{x}_t = \sigma_x dB_t^x$$

where the acceleration is modelled as a Wiener process [Bar-Shalom and Li, 2001]. The resulting model is termed "constant acceleration" (CA) model. These two models have been widely adopted in tracking applications because of their simplicity and mathematical tractability. As higher dimensional motions in these models are obtained by cascading several 1-D processes, this class of models is referred to as "coordinate-uncoupled" models [Li and Jilkov, 2003]. It is worth noting that such random process based models can be easily modified to include further knowledge regarding an object's behaviour and the corresponding modifications may lead to improved tracking/inference performance. For example, the acceleration may be modelled as a zero-mean Ornstein–Uhlenbeck (OU) process:

$$d\ddot{x}_t = -\lambda \ddot{x}_t + \sigma_x dB_t^x \tag{3.1}$$

where λ is some positive damping factor controlling the strength of pull towards zero acceleration. The usefulness of this model has been demonstrated in [Godsill, 2007] in application domains such as object tracking and financial market; see also [Pang et al., 2011b] where a similar idea is explored to model coordinating behaviours of group targets and [Ahmad et al., 2016a] in which OU-type tracking models are used for predicting the indented destination of an object.

As actual manoeuvring behaviours rarely have constant velocity or acceleration all the time, standard random process based models may not accurately model physical manoeuvres of an object. Also, we know from kinematics that the state of an object is normally structured, meaning that it does not move with independent forces applied on each coordinate. Consequently, there exists a necessity for the developments of models suitable for describing spatial trajectories of objects. Motion models based on target kinematics are usually coordinate-coupled. As surveyed in [Li and Jilkov, 2003], most kinematic models, including the intrinsic coordinate model we consider, follow the basic equations of curvilinear motion:

$$\dot{x}_t = v_t \cos\psi_t, \quad \dot{y}_t = v_t \sin\psi_t, \quad \dot{v}_t = a_{T,t}, \quad \dot{\psi}_t = a_{P,t}/v_t$$

Here x, y are positions in a 2-D Cartesian coordinate system, v stands for speed and ψ represents the heading angle with respect to x-axis, respectively. a_T and a_P are the accelerations along the tangential direction and the perpendicular (normal) direction on the motion plane, respectively. The majority of the existing work based on the above equations relies heavily on a rough assumption of having known a constant turn rate (i.e. $\dot{\psi}_t = const.$) as it leads to closed-form expressions for the state transition matrix for both the corresponding continuous time and discrete time models. Therefore, a model relying on this assumption is usually referred to as the "constant turn" (CT) model (or "coordinated turn model" as per [Blackman and Popoli, 1999; Roth et al., 2014]). The restriction of a known turn rate may be relaxed by modeling $\dot{\psi}_t$ as a random process as in [Gustafsson, 2000; Kastella and Biscuso, 1995; Maskell, 2004], but discretisation and/or linearisation needs to be performed due to the lack of a tractable solution for the resulting model.

The class of models that we are concerned with in this chapter, though still based on the same curvilinear equations given above, differs from the CT models in the way that it is constructed. It is a continuous time turn model which is based on an intrinsic coordinate frame as in the particle filtering methods of [Bunch and Godsill, 2012; Gilholm et al., 2005; Godsill and Vermaak, 2005; Godsill et al., 2007]. It should be noted that in the previous work the intrinsic model was mainly used in variable dimensional state inference problems while here we preserve the standard fixed-rate assumption. Another very similar model can be found in [Best and Norton, 1997] where an interacting multiple model tracker is implemented for an intrinsic coordinate frame. But the model relies on the assumption of zero perpendicular (cross track) acceleration; see also a 3-D variation to the intrinsic coordinate model in [Pilté et al., 2017], in which the dynamics of an aircraft is described based on Frenet-Serret formulas, and [Hostettler and Särkkä, 2016] where a simplified intrinsic model for tracking of a pedestrian is obtained by assuming zero tangential (along track) acceleration. Unlike [Best and Norton, 1997] and [Hostettler and Särkkä, 2016], here we do not impose any assumption of zero acceleration and in contrast to [Pilté et al., 2017] where the fusion of inertial measurement is difficult we focus on the case of planar motion.

In the following sections, it will be shown that while being powerful in terms of modelling target manoeuvres accurately, the intrinsic model turns out to be also very well adapted to the case of inertial measurements since it leads to a set of linear, Gaussian equations for turn rate, ground speed, travelled distance and acceleration measurements based upon which novel state space models can be constructed.

3.1.2 Tracking Algorithms

Closely coupled with the choice of a good model for target dynamics, inference algorithms are of particular importance when developing successful tracking applications. However, it is very common to see that state space models in tracking applications comprise of varied degrees of non-linearity and/or non-Gaussianity. For instance, the aforementioned CT model (with unknown turn rate) is a typical non-linear dynamic model whereas many measurement models, including the radar models [Bar-Shalom et al., 2011; Gustafsson et al., 2002] and the received signal strength index (RSSI) based measurement models [Harle, 2013; Liu et al., 2007; Nurminen et al., 2015b], are non-linear. Also, non-Gaussian noises are often employed for measurement models affected by non-line-of-sight (NLOS) issues, see [Ahmad et al., 2017; Nurminen et al., 2015a] for examples.

The inference of hidden state in systems exhibiting non-linearity and non-Gaussianity can be accomplished either by using nonlinear Gaussian filtering techniques (e.g. extended Kalman filter or unscented Kalman filter) or by using simulation-based approaches such as the IS-based SMC approaches and the sequential MCMC (SMCMC) reviewed in the previous chapter. Compared with the classical Kalman filter and its variants, particle filtering methods have proved to be a powerful methodology in dealing with non-linear and non-Gaussian problems [Ristic et al., 2003]. On the other hand, SMCMC is becoming known as a strong competitor for the IS-based particle methods in challenging scenarios. In contrast to IS-based SMC methods that can perform poorly in high dimensions, SMCMC has shown stronger potential [Mihaylova et al., 2014; Septier and Peters, 2016]. In this Chapter, novel tracking algorithms based on IS-based filters and SMCMC will be presented and the effectiveness of them will be demonstrated via a series of experiments.

3.2 Models

In this section we describe the dynamic model, defined in terms of an intrinsic coordinate system, to express the motion of manoeuvring object, as well as the corresponding measurements processes for the inertial, speed, distance and position data.

3.2.1 Dynamic Model in an Intrinsic Coordinate System

The intrinsic coordinate model considered here is a continuous-time manoeuvring model based on [Gilholm et al., 2005; Godsill and Vermaak, 2005; Godsill et al., 2007] and it is similar to the curvilinear model as in [Li and Jilkov, 2003] and [Best and Norton, 1997]. An applied force acting upon an object can be decomposed into a tangential component T_T and a component T_P perpendicular to the tangential vector. Let s denote the distance travelled by an object represented as a point mass m along a curved trajectory, and ψ the heading angle relative to a reference axis (e.g. x-axis in Fig. 3.1). The tangential and perpendicular equations of motion are then given by:

$$T_T = \lambda \frac{ds}{dt} + m\frac{d^2s}{dt^2} \tag{3.2}$$

$$T_P = m\frac{ds}{dt}\frac{d\psi}{dt} \tag{3.3}$$

where λ is a positive damping factor. Assume that T_T and T_P are piecewise constant between time τ and $\tau + \Delta\tau$. A 2-dimensional motion may be described by the evolution of the tangential and perpendicular unit vectors for the object over time, see for example Fig. 3.1, which illustrates their motion around a circular track.

Equation (3.2) may be solved directly as in [Godsill and Vermaak, 2005; Godsill et al., 2007] to obtain an expression for speed $v = ds/dt$:

$$v_n = v_{n-1}e^{-\frac{\Delta\tau_n\lambda}{m}} + \frac{T_T}{\lambda}\left(1 - e^{-\frac{\Delta\tau_n\lambda}{m}}\right) \tag{3.4}$$

with $\Delta\tau_n$ being the time difference between t_n and t_{n-1} (this may in practice be variable for some sensor types), and the distance s can then be routinely obtained by further integrating this expression:

$$s_n = s_{n-1} + \frac{\Delta\tau_n}{\lambda}T_T + \frac{m}{\lambda^2}\left(T_T - \lambda v_{n-1}\right)\left(e^{-\frac{\Delta\tau_n\lambda}{m}} - 1\right) \tag{3.5}$$

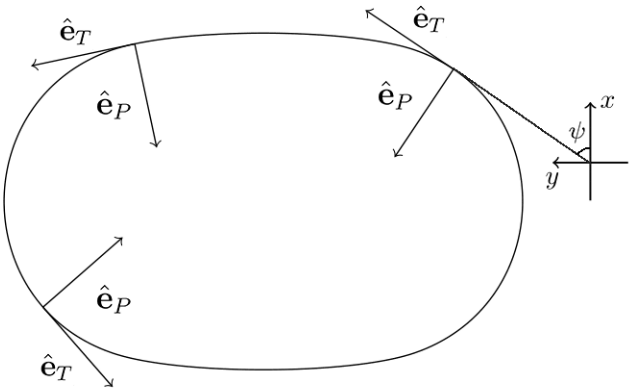


Fig. 3.1 2D curvilinear motion in an intrinsic coordinate system. $\hat{\mathbf{e}}_T$ and $\hat{\mathbf{e}}_P$ are unit vectors on which T_T and T_P are applied respectively.

Now, using the result for speed, (3.3) can be solved for the heading angle:

$$\psi_n = \psi_{n-1} + \frac{T_P}{T_T}\left(\frac{\Delta\tau_n\lambda}{m} - \log\left|\frac{v_{n-1}}{v_n}\right|\right) \tag{3.6}$$

Finally, the change in Cartesian position can be computed in closed form only for the case $\lambda = 0$ (no resistance to motion) [Bunch and Godsill, 2012], so in general the 2-D Cartesian position $\mathbf{p}_n = [x_n,\ y_n]^T$ can be obtained for example by Euler approximation:

$$\mathbf{p}(t+\delta t) \approx \mathbf{p}(t) + \delta t \cdot v(t) \cdot \begin{bmatrix}\cos\psi(t)\\ \sin\psi(t)\end{bmatrix} \tag{3.7}$$

Speed transition density

If the tangential thrust is assumed to be drawn from a simple Gaussian distribution, say $T_T \sim \mathcal{N}(\mu_T, \sigma_T^2)$, (3.4) becomes a linear Gaussian state space model and the transition density of speed is as follows:

$$p(v_n|v_{n-1}) = \mathcal{N}\left(v_n\middle|e^{-\frac{\Delta\tau_n\lambda}{m}}v_{n-1} + \frac{\mu_T}{\lambda}\left(1 - e^{-\frac{\Delta\tau_n\lambda}{m}}\right),\ \frac{\sigma_T^2}{\lambda^2}\left(1 - e^{-\frac{\Delta\tau_n\lambda}{m}}\right)^2\right) \tag{3.8}$$

Modelling the speed in this particular way allows the use of linear Kalman-style updating when direct speed measurements are available. This is in contrast to the commonly used white-noise driven models [Li and Jilkov, 2003] that usually leads to non-linear speed measurement models.

Heading (turn) rate transition density

Conditioned on the speed of the object, (3.3) can also be rearranged to give the dynamic equation of heading rate $\dot{\psi} = d\psi/dt$:

$$\dot{\psi}_n = \frac{T_P}{mv_n}$$

with $v_n \neq 0$. This is again a linear Gaussian model conditioning on v_n. Now, under the assumption that $T_P \sim \mathcal{N}(0, \sigma_P^2)$ we have

$$p(\dot{\psi}_n|v_n) = \mathcal{N}\Big(\dot{\psi}_n\Big|0, \frac{\sigma_P^2}{m^2v_n^2}\Big) \tag{3.9}$$

Moreover, we propose to use a state variable $\boldsymbol{\alpha}_n = [\dot{\psi}_n,\ b_n]^T$ in which b is a gyroscope bias term that is modelled as

$$db_t = \sigma_b dB_t \tag{3.10}$$

with B_t a Brownian motion. The state transition density for $\boldsymbol{\alpha}$ is thus given by

$$p(\boldsymbol{\alpha}_n|\boldsymbol{\alpha}_{n-1}, v_n) = \mathcal{N}\Big(\boldsymbol{\alpha}_n|A\boldsymbol{\alpha}_{n-1}, C\Big) \tag{3.11}$$

with

$$A = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

$$C = \begin{bmatrix} \sigma_P^2/m^2v_n^2 & 0 \\ 0 & \sigma_b^2\Delta\tau_n \end{bmatrix}$$

As a final remark on the system model, note that sampling from (3.8) and (3.11) is all we need to construct a trajectory of an object as it is clear from (3.2) - (3.4) that $T_T(v_n, v_{n-1})$, $T_P(\dot{\psi}_n, v_n)$ and the other states (i.e. headings, travelled distances and positions) are deterministic in the interval $(t_{n-1}, t_n]$ given v_n, v_{n-1} and $\dot{\psi}_n$.

3.2.2 Measurement Models

One of the main reasons here for choosing an intrinsic frame dynamic model is that the body frames of inertial sensors (e.g. accelerometers, gyroscopes and speedometers) are often aligned with the object's intrinsic coordinate (body) frame. This results in linear models for the kinematic states in the body frame connecting the dynamics to the inertial measurements, which later allows us to propose states effectively using Kalman filtering-based methods. Below are the details of each considered sensor measurement model.

Speed measurements Direct speed measurements are modelled by

$$\hat{v}_n = v_n + \epsilon_{\hat{v},n} \tag{3.12}$$

where the measurement noise $\epsilon_{v,n}$ is additive, white Gaussian noise of the form $\epsilon_{\hat{v},n} \sim \mathcal{N}(0, \sigma_{\hat{v}}^2)$.

Turn rate measurements Provided direct noisy measurements of the turn rate of the object on a 2D plane, $\hat{\dot{\psi}}_n$, and the random additive bias term b_n, the observation model for turn rate is given by:

$$\hat{\dot{\psi}}_n = \dot{\psi}_n + b_n + \epsilon_{\hat{\dot{\psi}},n}, \quad \epsilon_{\hat{\dot{\psi}},n} \sim \mathcal{N}\left(0, \sigma_{\hat{\dot{\psi}}}^2\right) \tag{3.13}$$

Note that the orientation of the IMU relative to an object can be arbitrary, depending on the way how it is placed/mounted. Consequently, direct measurements of turn rate may be unavailable as there can be a misalignment between the vertical axis of the gyroscope and the normal vector of the plane of motion. In such cases, we could obtain angular velocity on the plane of motion by rotating the gyroscope's frame onto the intrinsic frame, which is a future direction to extend the model. In this chapter however we treat the problem as a purely 2D tracking model. As a result, angular speed from the gyroscope's vertical axis (i.e. the axis vertical to the ground) is considered as a direct measurement of the heading rate. The bias term b_n in our observation model will then be considered to include a component both from the modelling error and the true instrumentation bias of the device.

Acceleration measurements Similar to the treatment for turn rate measurements, we here consider the case where direct measurements for tangential and perpendicular accelerations are also available. Specifically, we use $\hat{a}_T$ and $\hat{a}_P$ to represent measurements on the x-axis (forward) and the y-axis (leftward) of an accelerometer. To present the models, re-arrange first the equation of tangential thrust with $a_T = \frac{d^2s}{dt^2}$:

$$T_T = \lambda \frac{ds}{dt} + m \frac{d^2s}{dt^2} \implies a_T = -\frac{\lambda}{m} v + \frac{T_T}{m} \tag{3.14}$$

As for a_P, the dynamic equation is simply

$$a_P(t) = \frac{T_P}{m} = \dot{\psi}(t)v(t), \quad \tau < t \leq \tau + \Delta\tau \tag{3.15}$$

and from (3.4) we have

$$T_T(v_n, v_{n-1}) = \frac{\lambda}{1 - e^{-\frac{\Delta\tau_n\lambda}{m}}}\Big(v_n - e^{-\frac{\Delta\tau_n\lambda}{m}}v_{n-1}\Big) \tag{3.16}$$

Note that combining (3.11) and (3.14) - (3.16), the joint state transition density can be simplified to

$$\begin{aligned} &p(v_n, \boldsymbol{\alpha}_v, a_{T,n}, a_{P_n}|v_{n-1}, \alpha_{n-1}, a_{T,n-1}, a_{P,n-1}) \\ &= p(v_n|v_{n-1})p(\boldsymbol{\alpha}_n|v_n, \alpha_{n-1})p(a_{T,n}|v_{n-1:n})p(a_{P,n}|v_n, \alpha_n) \\ &= p(v_n, \boldsymbol{\alpha}_n|v_{n-1}, \boldsymbol{\alpha}_{n-1}) \end{aligned}$$

owing to the deterministic nature of the model. Now, we are in a position to give the measurement models for the acceleration components at time t_n:

$$\hat{a}_{T,n} = \frac{\lambda}{m}\frac{e^{-\frac{\Delta\tau_n\lambda}{m}}}{1 - e^{-\frac{\Delta\tau_n\lambda}{m}}}(v_n - v_{n-1}) + \epsilon_{\hat{a}_{T,n}} \tag{3.17}$$

$$\hat{a}_{P,n} = \dot{\psi}_n v_n + \epsilon_{\hat{a}_{P,n}} \tag{3.18}$$

with $\epsilon_{\hat{a}_{T,n}} \sim \mathcal{N}(0, \sigma^2_{\hat{a}_T})$ and $\epsilon_{\hat{a}_{P,n}} \sim \mathcal{N}(0, \sigma^2_{\hat{a}_P})$. Note that the right hand side of (3.17), excluding the noise term, is obtained by substituting (3.16) into (3.14).

Distance measurements According to (3.5), distance travelled along the curved path in the time interval $(t_{n-1}, t_n]$ can be written as

$$\begin{aligned} d_n &= s_n - s_{n-1} \\ &= \frac{\Delta\tau_n}{\lambda}T_T(v_{n-1}, v_n) + \frac{m}{\lambda^2}\Big(T_T(v_{n-1}, v_n) - \lambda v_{n-1}\Big)\Big(e^{-\frac{\Delta\tau_n\lambda}{m}} - 1\Big) \end{aligned} \tag{3.19}$$

and the corresponding measurement model is given by

$$\hat{d}_n = \Bigg(\frac{\Delta\tau_n}{1 - e^{-\frac{\Delta\tau_n\lambda}{m}}} - \frac{m}{\lambda}\Bigg)v_n + \Bigg(\frac{m}{\lambda} - \frac{\Delta\tau_n e^{-\frac{\Delta\tau_n\lambda}{m}}}{1 - e^{-\frac{\Delta\tau_n\lambda}{m}}}\Bigg)v_{n-1} + \epsilon_{\hat{d},n} \tag{3.20}$$

with $\epsilon_{\hat{d},n} \sim \mathcal{N}(0, \sigma^2_{\hat{d}})$.

Position measurements In addition to inertial, speed and distance measurements, the model for intermittent (not necessarily) position information is given by

$$p(\hat{\mathbf{p}}_n|\mathbf{p}_n) = \mathcal{N}(\hat{\mathbf{p}}_n|g(\mathbf{p}_n), \Sigma_{\hat{\mathbf{p}}}) \tag{3.21}$$

with $\hat{\mathbf{p}}_n$ given by (3.7), $g(\cdot)$ some linear or nonlinear mapping function and $\Sigma_{\mathbf{p}} = \mathrm{diag}([\sigma_x^2, \sigma_y^2])$.

3.3 State Estimation

Since we are dealing with a tracking problem with non-linear dynamic models, there is no closed-form analytic solution for the whole problem. Therefore, we consider using sequential Monte Carlo (SMC) methods, including IS-based SMC (i.e. particle filters) as well as MCMC-based SMC (i.e. SMCMC), to sequentially estimate the target states under a Bayesian framework.

More specifically, in this section two approaches for improving performance are presented: the first cascades locally optimal proposal kernels together, seeking to reduce the mismatch between prior predictive distribution and the posterior distribution conditioned on new measurements, and the second provides a sequential batch inference procedure based on the idea of forward-filtering-backward-sampling (FFBS) [Carter and Kohn, 1994; Frühwirth-Schnatter, 1994]. In particular, for the latter we will show that efficient section-wise FFBS-based kernels can be constructed and implemented by IS-based SMC filters and SMCMC and the resulting tracking algorithms provide effective alternative solutions for multi-rate sensor fusion problems.

3.3.1 IS-based SMC Filter with Locally Optimal Proposals

Owing to the particular design of our models, we can sample states from efficiently constructed locally optimal proposal kernels. Specifically, for speed we suggest an importance distribution $q(\cdot)$ in the following form, using (3.8), (3.12), (3.17) and (3.20):

$$\begin{aligned} q(v_n|v_{n-1}, \hat{v}_n, \hat{a}_{T,n}, \hat{d}_n) &= p(v_n|v_{n-1}, \hat{v}_n, \hat{a}_{T,n}, \hat{d}_n) \\ &= \mathcal{N}(v_n|\tilde{m}_v, \tilde{\sigma}_v^2) \end{aligned} \tag{3.22}$$

The values of $\tilde{m}_v$, $\tilde{\sigma}_v^2$ and some others which are corresponding to different combination of sensors are given in Appendix 3.A.1. This distribution is the "optimal kernel" for the speed in the sense of [Doucet et al., 2000b] conditioned upon the observations directly linked to it. Similarly, we suggest an importance distribution for $\boldsymbol{\alpha}_n$ as follows:

$$\begin{aligned} q(\boldsymbol{\alpha}_n|\boldsymbol{\alpha}_{n-1}, \hat{\dot{\psi}}_n, \hat{a}_{P,n}, v_n) &= p(\boldsymbol{\alpha}_n|\boldsymbol{\alpha}_{n-1}, \hat{\dot{\psi}}_n, \hat{a}_{P,n}, v_n) \\ &= \mathcal{N}(\boldsymbol{\alpha}_t|\tilde{m}_{\boldsymbol{\alpha}}, \tilde{P}_{\boldsymbol{\alpha}}) \end{aligned} \tag{3.23}$$

This is still a locally optimal kernel conditioned upon the speed $s_t^{(i)}$, the heading rate measurement and the acceleration measurement (see details in Appendix 3.A.2).

By construction, sequential importance sampling and resampling (SIR) can be applied to obtain samples from the target distribution

$$p(v_{0:n}, \boldsymbol{\alpha}_{0:n}|\hat{v}_{0:n}, \hat{\dot{\psi}}_{0:n}, \hat{a}_{T,0:n}, \hat{a}_{P,0:n}, \hat{d}_{0:n}, \hat{\mathbf{p}}_{0:n}) \tag{3.24}$$

and the update equation for the unnormalised importance weight $\tilde{w}_n^{(i)}$ is given by

$$\begin{aligned} \tilde{\omega}_n^{(i)} &= \frac{p(v_{0:n}^{(i)}, \boldsymbol{\alpha}_{0:n}^{(i)}|\hat{v}_{0:n}, \hat{\dot{\psi}}_{0:n}, \hat{a}_{T,0:n}, \hat{a}_{P,0:n}, \hat{d}_{0:n}, \hat{\mathbf{p}}_{0:n})}{q(v_{0:n}^{(i)}, \boldsymbol{\alpha}_{0:n}^{(i)}|\hat{v}_{0:n}, \hat{\dot{\psi}}_{0:n}, \hat{a}_{T,0:n}, \hat{a}_{P,0:n}, \hat{d}_{0:n}, \hat{\mathbf{p}}_{0:n})} \\ &\propto \tilde{\omega}_{n-1}^{(i)} \times \frac{p(v_n^{(i)}, \boldsymbol{\alpha}_n^{(i)}|v_{n-1}^{(i)}, \boldsymbol{\alpha}_{n-1}^{(i)})p(\hat{v}_n, \hat{\dot{\psi}}_n, \hat{a}_{T,n}, \hat{a}_{P,n}, \hat{d}_n|v_{n-1:n}^{(i)}, \boldsymbol{\alpha}_n^{(i)})}{q(v_n^{(i)}|v_{n-1}^{(i)}, \hat{v}_t, \hat{a}_{T,n}, \hat{d}_n)q(\boldsymbol{\alpha}_n^{(i)}|\boldsymbol{\alpha}_{n-1}^{(i)}, \hat{\dot{\psi}}_n, \hat{a}_{P,n}, v_n^{(i)})} p(\hat{\mathbf{p}}_n|v_{0:n}^{(i)}, \boldsymbol{\alpha}_{0:n}^{(i)}) \\ &\propto \tilde{\omega}_{n-1}^{(i)} \times p(\hat{v}_n|v_{n-1}^{(i)})p(\hat{\dot{\psi}}_n|\alpha_{n-1}^{(i)}, v_n^{(i)})p(\hat{a}_{T,n}|v_{n-1}^{(i)}, \hat{v}_n)p(\hat{a}_{P,n}|\hat{\dot{\psi}}_n, \alpha_{n-1}^{(i)}, v_n^{(i)}) \\ &\qquad \times p(\hat{d}_n|v_{n-1}^{(i)}, \hat{v}_n, \hat{a}_{T,n})p(\hat{\mathbf{p}}_n|v_{0:n}^{(i)}, \boldsymbol{\alpha}_{0:n}^{(i)}) \end{aligned} \tag{3.25}$$

where the likelihood terms, except for the position related one, are computed as side products when constructing the optimal kernels (see Appendix 3.A). Note that in a more challenging scenario where position-related measurements are not available at every time instant (i.e. a multi-rate sensor fusion case where the position observations do not have the same sampling rate as that of inertial sensors) we can compute the likelihood $p(\hat{\mathbf{p}}_n|v_{0:n}^{(i)}, \boldsymbol{\alpha}_{0:n}^{(i)})$ only when the object receives a position fix - at all other times this term is omitted from the weight update. When position measurements are present, the likelihood is computed as $p(\hat{\mathbf{p}}_n|\boldsymbol{\alpha}_{0:n}^{(i)}, s_{0:t}^{(i)}) = p(\hat{\mathbf{p}}_n|\mathbf{p}_n^{(i)})$ as in (3.21) and $\mathbf{p}_n^{(i)}$ is the Cartesian location of the i-th particle, obtained using (3.7). More generally, as the intrinsic dynamic model we adopt is a continuous-time model the position measurement can arrive asynchronously with the inertial measurements. In such cases, we can propose states in terms of their priors and update the weights using the position likelihood.

3.3.2 IS-based SMC Filter with Section-wise Optimal Proposals

Here we introduce a scheme that performs section-wise backward smoothing, aimed at improving the retrospective performance of the scheme and enhancing the proposed state sequences. The scheme is operable over any interval desired and thus it is well adapted to any tracking problems with asynchronous position and inertial measurements. To simplify the presentation, we first re-define the measurement models at time t_n for v and $\boldsymbol{\alpha}$ as:

$$\mathbf{y}_{\alpha,n} = \begin{bmatrix} \hat{\psi}_n \\ \hat{a}_{P,n} \end{bmatrix} = B_{\alpha,n}\boldsymbol{\alpha}_n + \boldsymbol{\eta}_{\alpha,n} \tag{3.26}$$

$$\mathbf{y}_{v,n} = \begin{bmatrix} \hat{d}_n \\ \hat{a}_{T,n} \\ \hat{s}_n \end{bmatrix} = B_v \begin{bmatrix} v_n \\ v_{n-1} \end{bmatrix} + \boldsymbol{\eta}_{v,n} \tag{3.27}$$

with

$$\begin{aligned}
B_{\alpha,n} &= \begin{bmatrix} 1 & 1 \\ v_n & 0 \end{bmatrix} \\
B_v &= \begin{bmatrix} \frac{\Delta\tau_n}{1-e^{-\frac{\Delta\tau_n\lambda}{m}}} - \frac{m}{\lambda} & \frac{m}{\lambda} - \frac{\Delta\tau_n e^{-\frac{\Delta\tau_n\lambda}{m}}}{1-e^{-\frac{\Delta\tau_n\lambda}{m}}} \\ \frac{\lambda}{m}\frac{e^{-\frac{\Delta\tau_n\lambda}{m}}}{1-e^{-\frac{\Delta\tau_n\lambda}{m}}} & -\frac{\lambda}{m}\frac{e^{-\frac{\Delta\tau_n\lambda}{m}}}{1-e^{-\frac{\Delta\tau_n\lambda}{m}}} \\ 1 & 0 \end{bmatrix} \\
\boldsymbol{\eta}_{\alpha,n} &\sim \mathcal{N}(0, \Sigma_\alpha), \quad \Sigma_{\boldsymbol{\alpha}} = \mathrm{diag}([\sigma^2_{\hat{\psi}},\ \sigma^2_{\hat{a}_P}]) \\
\boldsymbol{\eta}_{v,n} &\sim \mathcal{N}(0, \Sigma_v), \quad \Sigma_v = \mathrm{diag}([\sigma^2_{\hat{d}},\ \sigma^2_{\hat{a}_T},\ \sigma^2_{\hat{v}}])
\end{aligned}$$

Subsequently, we may factorise the posterior distribution as follows, with batch size $k \geq 1$ defined as the section length between two consecutive position measurements,

$$\begin{aligned}
&p(v_{0:n}, \boldsymbol{\alpha}_{0:n}|\mathbf{y}_{\alpha,0:n}, \mathbf{y}_{v,0:n}, \hat{\mathbf{p}}_{0:n}) \\
&\propto p(v_{0:n}, \boldsymbol{\alpha}_{0:n}|\mathbf{y}_{\alpha,0:n}, \mathbf{y}_{v,0:n}, \hat{\mathbf{p}}_{0:n-k})p(\hat{\mathbf{p}}_n|\boldsymbol{\alpha}_{0:n}, v_{0:n}) \\
&\propto p(v_{0:n-k}, \boldsymbol{\alpha}_{0:n-k}|\mathbf{y}_{\alpha,0:n-k}, \mathbf{y}_{v,0:n-k}, \hat{\mathbf{p}}_{0:n-k}) \\
&\quad \times p(v_{n-k+1:n}|\mathbf{y}_{v,n-k+1:n}, v_{n-k})p(\mathbf{y}_{v,n-k+1:n}|v_{n-k}) \\
&\quad \times p(\boldsymbol{\alpha}_{n-k+1:n}|\mathbf{y}_{\alpha,n-k+1:n}, v_{n-k+1:n}, \boldsymbol{\alpha}_{n-k})p(\mathbf{y}_{\alpha,n-k+1:n}|v_{n-k+1:n}, \boldsymbol{\alpha}_{n-k}) \\
&\quad \times p(\hat{\mathbf{p}}_n|\boldsymbol{\alpha}_{0:n}, v_{0:n})
\end{aligned} \tag{3.28}$$

The dependency on states at time $n-k$ is a direct result from the Markovian assumption. This particular factorisation of the target distribution is based on the observation that the speed states are jointly Gaussian conditioned on the speed related measurements, i.e. $p(v_{n-k+1:n}|\mathbf{y}_{v,n-k+1:n}, v_{n-k})$ is a k-variate Gaussian that may be sampled using standard forward-filtering-backward-sampling (FFBS) methods, see [Carter and Kohn, 1994; Frühwirth-Schnatter, 1994]. This arises because of the linear Gaussian state-space structure of (3.8) and (3.27). Similarly, the linear Gaussian structures of (3.9), (3.10) and (3.26) means that the turn rate and bias have a jointly Gaussian distribution conditioned on sampled speeds and heading related measurements, $p(\boldsymbol{\alpha}_{n-k+1:n}|\mathbf{y}_{\alpha,n-k+1:n}, v_{n-k+1:n}, \boldsymbol{\alpha}_{n-k})$, which may also be drawn jointly using FFBS. These proposals can be considered as section-wise batch versions of the locally optimal proposals from the previous section. Therefore, the proposed method also applies to a strictly sequential case where $k=1$. In this case the proposals degenerate to locally optimal kernels conditioned on new measurements and previous states. Suppose it is now possible to draw a joint sequence of speeds from the local optimal distribution

$$q(v_{n-k+1:n}|v_{n-k}) = p(v_{n-k+1:n}|\mathbf{y}_{v,n-k+1:n}, v_{n-k})$$

using FFBS and then propose a sequence of $\boldsymbol{\alpha}$ states conditionally upon that proposed sequence of speeds according to

$$q(\boldsymbol{\alpha}_{n-k+1:n}|\boldsymbol{\alpha}_{n-k}) = p(\boldsymbol{\alpha}_{n-k+1:n}|\mathbf{y}_{\alpha,n-k+1:n}, v_{n-k+1:n}, \boldsymbol{\alpha}_{n-k})$$

the weight updating equation for the particle filter is given by

$$\begin{aligned}\tilde{\omega}_n^{(i)} &= \frac{p(v_{0:n}^{(i)}, \boldsymbol{\alpha}_{0:n}^{(i)}|\mathbf{y}_{\alpha,0:n}, \mathbf{y}_{v,0:n}, \hat{\mathbf{p}}_{0:n})}{q(v_{0:n}^{(i)}, \boldsymbol{\alpha}_{0:n}^{(i)}|\mathbf{y}_{\alpha,0:n}, \mathbf{y}_{v,0:n}, \hat{\mathbf{p}}_{0:n})} \\ &\propto \tilde{\omega}_{n-k}^{(i)} \times p(\mathbf{y}_{v,n-k+1:n}|v_{n-k}^{(i)}) p(\mathbf{y}_{\alpha,n-k+1:n}|v_{n-k+1:n}^{(i)}, \boldsymbol{\alpha}_{n-k}^{(i)}) p(\hat{\mathbf{p}}_n|\boldsymbol{\alpha}_{0:n}^{(i)}, v_{0:n}^{(i)}) \end{aligned} \tag{3.29}$$

Section-wise proposal for v

In the following, we show how to sample from the proposals as well as to compute the likelihood terms in the weight updating equation, based on the idea of FFBS. Specifically, for speed the forward filtering (FF) step updates $p(v_{n-1:n}|\mathbf{y}_{v,n-k+1:n}, v_{n-k})$ to $p(v_{n:n+1}|\mathbf{y}_{v,n-k+1:n+1}, v_{n-k})$ in the light of a new measurement $\mathbf{y}_{v,n+1}$ recursively. More precisely, as a starting point assume the distribution

$$p(v_n|\mathbf{y}_{v,n-k+1:n}, v_{n-k}) = \mathcal{N}(v_n|\mu_{v,n|n}, \sigma^2_{v,n|n})$$

which is a Gaussian, has been obtained. At time t_{n+1}, in the FF prediction stage a joint Gaussian distribution can be obtained as

$$\begin{aligned} p(v_{n:n+1}|\mathbf{y}_{v,n-k+1:n}, v_{n-k}) &= p(v_{n+1}|v_n)p(v_n|\mathbf{y}_{v,n-k+1:n}, v_{n-k}) \\ &= \mathcal{N}(v_n|a_v v_{n-1} + b_v, \sigma^2)\mathcal{N}(v_n|\mu_{v,n|n}, \sigma^2_{v,n|n}) \\ &= \mathcal{N}\Bigg(\begin{bmatrix} v_n \\ v_{n+1} \end{bmatrix} \Bigg| \tilde{\boldsymbol{\mu}}_{v,n+1}, \tilde{P}_{v,n+1} \Bigg) \end{aligned} \tag{3.30}$$

with

$$\begin{aligned} \tilde{\boldsymbol{\mu}}_{v,n+1} &= \begin{bmatrix} \mu_{v,n|n} \\ a_v\mu_{v,n|n} + b_v \end{bmatrix} \\ \tilde{P}_{v,n+1} &= \begin{bmatrix} \sigma^2_{v,n|n} & \sigma^2_{v,n|n}a_v \\ a_v\sigma^2_{v,n|n} & a_v^2\sigma^2_{v,n|n} + \sigma^2 \end{bmatrix} \end{aligned}$$

Note that a_v, b_v and σ^2 have been defined in (3.48). This predictive distribution can then be corrected when new measurements become available:

$$\begin{aligned} p(v_{n:n+1}|\mathbf{y}_{v,n-k+1:n+1}, v_{n-k}) &\propto p(\mathbf{y}_{v,n+1}|v_{n:n+1})p(v_{n:n+1}|\mathbf{y}_{v,n-k+1:n}, v_{n-k}) \\ &= \mathcal{N}\Bigg(\mathbf{y}_{v,n+1}\Bigg|B_v \begin{bmatrix} v_n \\ v_{n+1} \end{bmatrix}, \Sigma_v\Bigg)\mathcal{N}\Bigg(\begin{bmatrix} v_n \\ v_{n+1} \end{bmatrix} \Bigg| \tilde{\boldsymbol{\mu}}_{v,n+1}, \tilde{P}_{v,n+1} \Bigg) \\ &= \mathcal{N}\Bigg(\begin{bmatrix} v_n \\ v_{n+1} \end{bmatrix} \Bigg| \breve{\boldsymbol{\mu}}_{v,n+1}, \breve{P}_{v,n+1} \Bigg) \end{aligned} \tag{3.31}$$

where

$$\begin{aligned} \breve{\boldsymbol{\mu}}_{v,n+1} &= \tilde{\boldsymbol{\mu}}_v + K(\mathbf{y}_{v,n+1} - B_v\tilde{\boldsymbol{\mu}}_{v,n+1}) \\ \breve{P}_{v,n+1} &= (\mathbf{I} - KB_v)\tilde{P}_{v,n+1} \\ K &= \tilde{P}_{v,n+1}B_v^T(\Sigma_v + B_v\tilde{P}_{v,n+1}B_v^T)^{-1} \end{aligned}$$

Note also that the distribution needed for the next time instant can be obtained as a marginal of (3.31).

Backward sampling (BS) is applied to proposing samples from the joint Gaussian distribution conditioned on all observations recursively:

$$p(v_{n-k+1:n}|\mathbf{y}_{v,n-k+1:n}, v_{n-k}) = p(v_n, v_{n-1}|\mathbf{y}_{v,n-k+1:n}, v_{n-k}) \times \prod_{l=n-2}^{n-k+1} p(v_l|v_{l+1}, \mathbf{y}_{v,n-k+1:l+1}, v_{n-k}) \quad (3.32)$$

Here the component in the product is the conditional of a joint Gaussian (i.e. (3.31)):

$$p(v_l|v_{l+1}, \mathbf{y}_{v,n-k+1:l+1}, v_{n-k}) \propto p(v_l, v_{l+1}|\mathbf{y}_{v,n-k+1:l+1}, v_{n-k})$$

which is again a Gaussian and can be routinely obtained as per [Bishop, 2006]. At last, the likelihood term $p(\mathbf{y}_{v,n-k+1:n}|v_{n-k})$ can be calculated via Prediction Error Decomposition (PED) [Harvey, 1989], that is,

$$p(\mathbf{y}_{v,n-k+1:n}|v_{n-k}) = p(\mathbf{y}_{v,n-k+1}|v_{n-k}) \prod_{j=n-k+2}^{n} p(\mathbf{y}_{v,j}|\mathbf{y}_{v,n-k+1:j-1}, v_{n-k}) \quad (3.33)$$

with

$$\begin{aligned} p(\mathbf{y}_{v,j}|\mathbf{y}_{v,n-k+1:j-1}, v_{n-k}) &= \int p(\mathbf{y}_{v,j}|v_{j-1}, v_j)p(v_{j-1}, v_j|\mathbf{y}_{v,n-k+1:j-1}, v_{n-k})dv_{j-1}dv_j \\ &= \mathcal{N}(\mathbf{y}_{v,j}|B_v\tilde{\boldsymbol{\mu}}_{v,j+1}, B_v\tilde{P}_{v,j+1}B_v^T + \Sigma_v) \end{aligned}$$

Section-wise proposal for $\boldsymbol{\alpha}$

Owing to the linear and Gaussian state space models in (3.11) and (3.26), the FF stage for $\boldsymbol{\alpha}$ follows the standard Kalman filtering recursions [Ho and Lee, 1964; Kalman, 1960]. Supposing that $p(\boldsymbol{\alpha}_n|\mathbf{y}_{n-k+1:n}, v_{n-k+1:n}, \boldsymbol{\alpha}_{n-k}) = \mathcal{N}(\boldsymbol{\alpha}_n|\boldsymbol{\mu}_{\alpha,n|n}, P_{\alpha,n|n})$ is the output from a Kalman filter, we can compute the term $p(\boldsymbol{\alpha}_n|\boldsymbol{\alpha}_{n+1}, \mathbf{y}_{n-k+1:n}, v_{n-k+1:n}, , \boldsymbol{\alpha}_{n-k})$ as below

$$\begin{aligned} p(\boldsymbol{\alpha}_n|\boldsymbol{\alpha}_{n+1}, \mathbf{y}_{n-k+1:n}, v_{n-k+1:n}, , \boldsymbol{\alpha}_{n-k}) &= \frac{p(\boldsymbol{\alpha}_{n+1}|\boldsymbol{\alpha}_n)p(\boldsymbol{\alpha}_n|\mathbf{y}_{n-k+1:n}, v_{n-k+1:n}, \boldsymbol{\alpha}_{n-k})}{p(\boldsymbol{\alpha}_{n+1}|\mathbf{y}_{n-k+1:n}, v_{n-k+1:n}, \boldsymbol{\alpha}_{n-k})} \\ &\propto \mathcal{N}(\boldsymbol{\alpha}_{n+1}|A\boldsymbol{\alpha}_n, C)\mathcal{N}(\boldsymbol{\alpha}_n|\boldsymbol{\mu}_{\alpha,n|n}, P_{\alpha,n|n}) \\ &= \mathcal{N}(\boldsymbol{\alpha}_n|\mu_{\alpha,n|n+1}, P_{\alpha,n|n+1}) \quad (3.34) \end{aligned}$$

with

$$\begin{aligned}\mu_{\alpha,n|n+1} &= \boldsymbol{\mu}_{\alpha,n|n} + P_{\alpha,n|n}A^T(AP_{\alpha,n|n}A^T + C)^{-1}(\boldsymbol{\alpha}_{n+1} - A\boldsymbol{\mu}_{\alpha,n|n}) \\ P_{\alpha,n|n+1} &= P_{\alpha,n|n} - P_{\alpha,n|n}A^T(AP_{\alpha,n|n}A^T + C)^{-1}AP_{\alpha,n|n}\end{aligned}$$

Now, we can firstly draw $\boldsymbol{\alpha}_n$ given the speed sequence $v_{n-k+1:n}$ and the measurements $\mathbf{y}_{n-k+1:n}$ firstly and then work backwards in time drawing state using the backward smoothing recursions:

$$\begin{aligned}&p(\boldsymbol{\alpha}_{n-k+1:n}|\mathbf{y}_{\alpha,n-k+1:n}, v_{n-k+1:n}, \boldsymbol{\alpha}_{n-k}) \\ &\quad= p(\boldsymbol{\alpha}_n|\mathbf{y}_{\alpha,n-k+1:n}, v_{n-k+1:n}, \boldsymbol{\alpha}_{n-k}) \prod_{l=n-k+1}^{n-1} p(\boldsymbol{\alpha}_l|\boldsymbol{\alpha}_{l+1}, \mathbf{y}_{\alpha,n-k+1:l}, v_{n-k+1:l}, \boldsymbol{\alpha}_{n-k})\end{aligned} \tag{3.35}$$

Similarly, the second likelihood term in (3.29) can be also computed using PED:

$$\begin{aligned}&p(\mathbf{y}_{\alpha,n-k+1:n}|v_{n-k+1:n}, \boldsymbol{\alpha}_{n-k}) \\ &\quad= p(\mathbf{y}_{\alpha,n-k+1}|v_{n-k+1}, \boldsymbol{\alpha}_{n-k}) \prod_{j=n-k+2}^{n} p(\mathbf{y}_{\alpha,j}|\mathbf{y}_{\alpha,n-k+1:j-1}, v_{n-k+1:j}, \boldsymbol{\alpha}_{n-k})\end{aligned} \tag{3.36}$$

where

$$\begin{aligned}&p(\mathbf{y}_{\alpha,j}|\mathbf{y}_{\alpha,n-k+1:j-1}, v_{n-k+1:j}, \boldsymbol{\alpha}_{n-k}) \\ &= \int p(\mathbf{y}_{\alpha,j}|\boldsymbol{\alpha}_j, v_j)p(\boldsymbol{\alpha}_j|\mathbf{y}_{\alpha,n-k+1:j-1}, v_{n-k+1:j}, \boldsymbol{\alpha}_{n-k})d\boldsymbol{\alpha}_j \\ &= \int \mathcal{N}(\mathbf{y}_{\alpha,j}|B_{\alpha,j}\boldsymbol{\alpha}_j, \Sigma_\alpha)\mathcal{N}(\boldsymbol{\alpha}_j|\boldsymbol{\mu}_{\alpha,j|j-1}, P_{\alpha,j|j-1})d\boldsymbol{\alpha}_j \\ &= \mathcal{N}(\mathbf{y}_{\alpha,j}|B_{\alpha,j}\mu_{\alpha,j|j-1}, B_{\alpha,j}P_{\alpha,j|j-1}B_{\alpha,j}^T + \Sigma_\alpha)\end{aligned}$$

with the predictive distribution $\mathcal{N}(\boldsymbol{\alpha}_j|\boldsymbol{\mu}_{\alpha,j|j-1}, P_{\alpha,j|j-1})$ provided by a Kalman filter .

Resample-move (RM)

In the section-wise batch formulation, sampling highly correlated states from a high dimensional sample space can exacerbate the problem of sample degeneration and impoverishment. To mitigate this issue, we consider using Resample-move (RM) technique [Gilks and Berzuini, 2001] to rejuvenate degenerated particles after the resampling stage so as to improve the empirical approximation obtained by the particle filter. Conditioned on the states up to time instant t_{n-k} and all measurements, the invariant distribution of the MCMC move in the RM step is set to be

$$p(v_{n-k+1:n}, \boldsymbol{\alpha}_{n-k+1:n} | v_{0:n-k}, \boldsymbol{\alpha}_{0:n-k}, \mathbf{y}_{v,0:n}, \mathbf{y}_{\alpha,0:n}, \hat{\mathbf{p}}_{0:n}) \propto p(v_{0:n}, \boldsymbol{\alpha}_{0:n} | \mathbf{y}_{v,0:n}, \mathbf{y}_{\alpha,0:n}, \hat{\mathbf{p}}_{0:n}) \tag{3.37}$$

For each resampled particle, a Metropolis-Hastings (MH) [Robert and Casella, 2004] kernel can be designed to propose $\{v^*_{n-k+1:n}, \boldsymbol{\alpha}^*_{n-k+1:n}\}$ from the target distribution with proposals the same as those introduced above, that is,

$$\begin{aligned} q_{RM}(v_{n-k+1:n}) &= p(v_{n-k+1:n} | \mathbf{y}_{v,n-k+1:n}, v^{(i)}_{n-k}) \\ q_{RM}(\boldsymbol{\alpha}_{n-k+1:n}) &= p(\boldsymbol{\alpha}_{n-k+1:n} | \mathbf{y}_{\alpha,n-k+1:n}, v^{(i)}_{n-k+1:n}, \boldsymbol{\alpha}^{(i)}_{n-k}) \end{aligned}$$

It is then easy to show that the resulting acceptance ratio has the following form

$$\rho_{RM} = 1 \wedge \frac{p(\mathbf{y}_{\alpha,n-k+1:n} | v^*_{n-k+1:n}, \boldsymbol{\alpha}^{(i)}_{n-k})}{p(\mathbf{y}_{\alpha,n-k+1:n} | v^{(i)}_{n-k+1:n}, \boldsymbol{\alpha}^{(i)}_{n-k})} \frac{p(\hat{\mathbf{p}}_n | v^{(i)}_{0:n-k}, \boldsymbol{\alpha}^{(i)}_{0:n-k}, v^*_{n-k+1:n}, \boldsymbol{\alpha}^*_{n-k+1:n})}{p(\hat{\mathbf{p}}_n | v^{(i)}_{0:n}, \boldsymbol{\alpha}^{(i)}_{0:n})} \tag{3.38}$$

Since each selected particle, that is $\{v^{(i)}_{0:n}, \boldsymbol{\alpha}^{(i)}_{0:n}\}$, before being moved, is already approximately distributed according to $p_{0:n}$, the usual burn-in period for MCMC is not necessary, see [Gilks and Berzuini, 2001].

3.3.3 SMCMC with Section-wise Optimal Proposals

SMCMC has shown great potential in breaking the curse of dimensionality from which the IS-based SMC methods usually suffer. This can be a desirable property when developing algorithms for sequential batch inference. Hence, we propose to use SMCMC for state estimation, aiming at providing effective alternative approaches for the tracking of an object with inertial sensors. The adopted SMCMC algorithm, as detailed in Chapter 2, follows a mixture sampling based procedure that selects a joint draw action with probability P_J which updates all states simultaneously and refinement steps with probability $(1 - P_J)$ in which $\{v_{0:n-k}, \boldsymbol{\alpha}_{0:n-k}\}$ and $\{v_{n-k+1:n}, \boldsymbol{\alpha}_{n-k+1:n}\}$ are updated individually.

Joint Update of $\{v_{0:n}, \boldsymbol{\alpha}_{0:n}\}$

To make a joint draw with a target distribution defined in (3.28) at the m-th MCMC iteration, two steps are required: first draw past state trajectories $\{v^*_{0:n-k}, \boldsymbol{\alpha}^*_{0:n-k}\}$

using the empirical particle representation obtained at time t_{n-k}:

$$q(v_{0:n-k}, \boldsymbol{\alpha}_{0:n-k} | v_{0:n-k}^{m-1}, \boldsymbol{\alpha}_{0:n-k}^{m-1}) = \hat{p}(v_{0:n-k}, \boldsymbol{\alpha}_{0:n-k} | \mathbf{y}_{\alpha,0:n-k}, \mathbf{y}_{v,0:n-k}, \hat{\mathbf{p}}_{0:n-k}) \tag{3.39}$$

This is then followed by the proposal of $\{v_{n-k+1:n}^*, \boldsymbol{\alpha}_{n-k+1:n}^*\}$ conditioned on the states drawn in the first step:

$$\begin{aligned} & q(v_{n-k+1:n}, \boldsymbol{\alpha}_{n-k+1:n} | \boldsymbol{\alpha}_{n-k+1:n}^{m-1}, v_{n-k+1:n}^{m-1}) \\ & \quad = p(\boldsymbol{\alpha}_{n-k+1:n} | \mathbf{y}_{\alpha,n-k+1:n}, v_{n-k+1:n}, \boldsymbol{\alpha}_{n-k}^*) p(v_{n-k+1:n} | \mathbf{y}_{v,n-k+1:n}, v_{n-k}^*) \end{aligned} \tag{3.40}$$

According to (3.39)-(3.40) the acceptance ratio for a MH kernel is obtained as follows,

$$\begin{aligned} \rho_1 &= 1 \wedge \frac{p(v_{0:n}^*, \boldsymbol{\alpha}_{0:n}^* | \mathbf{y}_{v,0:n}, \mathbf{y}_{\alpha,0:n}, \hat{\mathbf{p}}_{0:n})}{p(v_{0:n}^{m-1}, \boldsymbol{\alpha}_{0:n}^{m-1} | \mathbf{y}_{v,0:n}, \mathbf{y}_{\alpha,0:n}, \hat{\mathbf{p}}_{0:n})} \frac{q(v_{0:n}^{m-1}, \boldsymbol{\alpha}_{0:n}^{m-1} | v_{0:n-k}^*, \boldsymbol{\alpha}_{0:n-k}^*)}{q(v_{0:n}^*, \boldsymbol{\alpha}_{0:n}^* | v_{0:n-k}^{m-1}, \boldsymbol{\alpha}_{0:n-k}^{m-1})} \\ &= 1 \wedge \frac{p(\mathbf{y}_{\alpha,n-k+1:n} | v_{n-k+1:n}^*, \boldsymbol{\alpha}_{n-k}^*)}{p(\mathbf{y}_{\boldsymbol{\alpha},n-k+1:n} | v_{n-k+1:n}^{m-1}, \boldsymbol{\alpha}_{n-k}^{m-1})} \frac{p(\mathbf{y}_{v,n-k+1:n} | v_{n-k}^*)}{p(\mathbf{y}_{v,n-k+1:n} | v_{n-k}^{m-1})} \frac{p(\hat{\mathbf{p}} | \boldsymbol{\alpha}_{0:n}^*, v_{0:n}^*)}{p(\hat{\mathbf{p}}_n | \boldsymbol{\alpha}_{0:n}^{m-1}, v_{0:n}^{m-1})} \end{aligned} \tag{3.41}$$

Refinements of $\{v_{0:n-k}, \boldsymbol{\alpha}_{0:n-k}\}$ and $\{v_{n-k+1:n}, \boldsymbol{\alpha}_{n-k+1:n}\}$

Coupled with the joint proposal, a series of Metropolis-within-Gibbs steps can be carried out to refine states respectively. More specifically, we first refine the estimate of $\{v_{0:n-k}, \boldsymbol{\alpha}_{0:n-k}\}$ using a MH kernel with the following target distribution:

$$\begin{aligned} & p(v_{0:n-k}, \boldsymbol{\alpha}_{0:n-k} | v_{n-k+1:n}, \boldsymbol{\alpha}_{n-k+1:n}, \mathbf{y}_{v,0:n}, \mathbf{y}_{\alpha,0:n}, \hat{\mathbf{p}}_{0:n}) \\ & \qquad = \frac{p(v_{0:n}, \boldsymbol{\alpha}_{0:n} | \mathbf{y}_{v,0:n}, \mathbf{y}_{\alpha,0:n}, \hat{\mathbf{p}}_{0:n})}{p(v_{n-k+1:n}, \boldsymbol{\alpha}_{n-k+1:n} | \mathbf{y}_{v,0:n}, \mathbf{y}_{\alpha,0:n}, \hat{\mathbf{p}}_{0:n})} \end{aligned} \tag{3.42}$$

and a proposal identical to that given by (3.39). $\{v_{0:t-k}^*, \boldsymbol{\alpha}_{0:t-k}^*\}$ is then accepted according to the following ratio:

$$\begin{aligned} \rho_2 = 1 \wedge \Bigg(& \frac{p(\boldsymbol{\alpha}_{n-k+1}^{m-1} | \boldsymbol{\alpha}_{n-k}^*, v_{n-k+1}^{m-1})}{p(\boldsymbol{\alpha}_{n-k+1}^{m-1} | \boldsymbol{\alpha}_{n-k}^{m-1}, v_{n-k+1}^{m-1})} \frac{p(v_{n-k+1}^{m-1} | v_{n-k}^*)}{p(v_{n-k+1}^{m-1} | v_{n-k}^{m-1})} \\ & \times \frac{p(\hat{\mathbf{p}}_n | v_{0:n-k}^*, \boldsymbol{\alpha}_{0:n-k}^*, v_{n-k+1:n}^{m-1}, \boldsymbol{\alpha}_{n-k+1:n}^{m-1})}{p(\hat{\mathbf{p}}_n | v_{0:n-k}^{m-1}, \boldsymbol{\alpha}_{0:n-k}^{m-1}, v_{n-k+1:n}^{m-1}, \boldsymbol{\alpha}_{n-k+1:n}^{m-1})} \Bigg) \end{aligned} \tag{3.43}$$

where $p(\boldsymbol{\alpha}_{n-k+1} | \boldsymbol{\alpha}_{n-k}, v_{n-k+1})$ and $p(v_{n-k+1}^{m-1} | v_{n-k})$ are just the state transition densities given in (3.11) and (3.8). The next step is refining $\{v_{n-k+1:n}, \boldsymbol{\alpha}_{n-k+1:n}\}$. To this end, set the target distribution of the MH kernel to be

$$p(v_{n-k+1:n}, \boldsymbol{\alpha}_{n-k+1:n} | v_{0:n-k}, \boldsymbol{\alpha}_{0:n-k}, \mathbf{y}_{v,0:n}, \mathbf{y}_{\alpha,0:n}, \hat{\mathbf{p}}_{0:n}) \tag{3.44}$$

and propose states from the section-wise proposal kernels constructed above. As a result, we obtain an acceptance rate similar to that used in the RM step (note the change of indexing notation):

$$\rho_3 = 1 \wedge \frac{p(\mathbf{y}_{\alpha,n-k+1:n} | v^*_{n-k+1:n}, \boldsymbol{\alpha}^m_{n-k})}{p(\mathbf{y}_{\alpha,n-k+1:n} | v^{m-1}_{n-k+1:n}, \boldsymbol{\alpha}^m_{n-k})} \frac{p(\hat{\mathbf{p}}_n | v^m_{0:n-k}, \boldsymbol{\alpha}^m_{0:n-k}, v^*_{n-k+1:n}, \boldsymbol{\alpha}^*_{n-k+1:n})}{p(\hat{\mathbf{p}}_n | v^m_{0:n-k}, \boldsymbol{\alpha}^m_{0:n-k}, v^{m-1}_{n-k+1:n}, \boldsymbol{\alpha}^{m-1}_{n-k+1:n})} \tag{3.45}$$

Also note that after any acceptance event all likelihood terms need to be updated accordingly. The pseudo code of the above procedure, for a single batch, is given in Algorithm 5.

3.3.4 Parallel SMCMC

Generic SMCMC algorithms can still be inefficient when dealing with highly correlated variables in high dimensional systems. To account for this, we propose an intuitive but effective solution which runs multiple Markov chains in parallel. The idea is to provide greater diversity and to avoid the situation in which we only have one chain and it gets stuck at one ancestor trajectory. As a result, this algorithm introduces a new design parameter, which is the number of parallel chains N_{Chain}.

We would like to note that in the proposed parallel SMCMC scheme we do not combine multiple chains into one chain at the end of each section/batch. A simple mixing scheme, in which past state trajectories $\{v_{0:n-k}, \boldsymbol{\alpha}_{0:n-k}\}$ are drawn uniformly from a combined chain, has been experimented with and the results indicate that its performance is not comparable with the scheme running multiple independent chains. One possible reason for this outcome is that the negative effects from one or multiple poorly mixed chains may be exacerbated in such a simple combination scheme.

```
In a batch defined by consecutive position measurements $\hat{\mathbf{p}}_{n-k}$ and $\hat{\mathbf{p}}_n$:
for $m = 1, 2, \ldots, N_{iter}$ do
    Sample $u \sim \mathrm{Unif}(0, 1)$;
    if $u < P_J$ then
        // Joint Update
        Sample $\{v^*_{0:n-k}, \boldsymbol{\alpha}^*_{0:n-k}\}$ according to (3.39) ;
        Sample $\{v^*_{n-k+1:n}, \boldsymbol{\alpha}^*_{n-k+1:n}\}$ according to (3.40) ;
        Compute the MH acceptance ratio $\rho_1$ according to (3.41) ;
        Accept $\{v^m_{0:n}, \boldsymbol{\alpha}^m_{0:n}\} = \{v^*_{0:n}, \boldsymbol{\alpha}^*_{0:n}\}$ with probability $\rho_1$;
    else
        // Refinement step 1
        Sample $\{v^*_{0:n-k}, \boldsymbol{\alpha}^*_{0:n-k}\}$ according to (3.39) ;
        Compute the MH acceptance rate $\rho_2$ according to (3.43) ;
        Accept $\{v^m_{0:n-k}, \boldsymbol{\alpha}^m_{0:n-k}\} = \{v^*_{0:n-k}, \boldsymbol{\alpha}^*_{0:n-k}\}$ with probability $\rho_2$ ;
        // Refinement step 2
        Sample $\{v^*_{n-k+1:n}, \boldsymbol{\alpha}^*_{n-k+1:n}\}$ according to (3.40) ;
        Compute the MH acceptance rate $\rho_3$ according to (3.45) ;
        Accept $\{v^m_{n-k+1:n}, \boldsymbol{\alpha}^m_{n-k+1:n}\} = \{v^*_{n-k+1:n}, \boldsymbol{\alpha}^*_{n-k+1:n}\}$ with probability $\rho_3$ ;
    end
    Keep every $N_{thin}$ MCMC output $\{v^{(i)}_{0:n}, \boldsymbol{\alpha}^{(i)}_{0:n}\} = \{v^m_{0:n}, \boldsymbol{\alpha}^m_{0:n}\}$ to form the new
     particle set after a burn-in period $N_{burn}$ ;
    Approximate the posterior:
     $\hat{p}(v_{0:n}, \boldsymbol{\alpha}_{0:n} | \mathbf{y}_{\alpha,0:n}, \mathbf{y}_{v,0:n}, \hat{\mathbf{p}}_{0:n}) = \frac{1}{N_p} \sum_{i=1}^{N_p} \delta_{v^{(i)}_{0:n}, \boldsymbol{\alpha}^{(i)}_{0:n}} (v_{0:n}, \boldsymbol{\alpha}_{0:n})$;
end
```

Algorithm 5: SMCMC algorithm (single batch)

3.4 Experimental Results

3.4.1 Synthetic Data

The proposed inference methods are firstly tested on 100 Monte Carlo realisations (i.e. 100 trajectories generated according to the intrinsic model, each with $N = 300$ data points) of the problem. The parameters used in the simulation are given in Table 3.1 whilst the abbreviated names of the algorithms are defined in Table 3.2. Here f_s is the sampling rate for inertial, speed and distance measurements while the time of arrival of position signals is governed by a homogeneous Poisson process with intensity $\lambda_{\mathcal{POI}}$. The state estimation is based on the basic smoothing scheme as in [Kitagawa, 1996], which estimates the entire state trajectory $p(v_{0:N}, \boldsymbol{\alpha}_{0:N} | \mathbf{y}_{v,0:N}, \mathbf{y}_{\alpha,0:N}, \hat{\mathbf{p}}_{0:N})$ using the final filtering approximation.

In our first evaluation, we examine the performance of the algorithms in challenging tracking scenarios, where regular inertial measurements and occasional position measurements are available, with varied number of particles/MCMC iterations. A Poisson intensity $\lambda_{POI} = 1/6$ is used, which corresponds to the case where there are on average 50 position measurements arriving randomly during 300 seconds. The estimation performance is evaluated by computing the position RMSE, which for a single dataset is defined as

$$\text{RMSE}_{\mathbf{p}} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \|\mathbf{p}_n - \mathbf{p}_{\text{true},n}\|^2} \tag{3.46}$$

The total number of iterations (including burn-in period) used in MCMC-based algorithms is set to be the same as the particle number used in FFBS, FFBS-RM and OPT (see Table 3.3 for a detailed algorithm parameter specification). This is to obtain

Table 3.1 Model parameters (synthetic data)

Symbol	Value	Symbol	Value
μ_T	15	$\sigma_{\hat{v}}$	1
σ_T	30	$\sigma_{\hat{a}_T}$	0.5
σ_P	220	$\sigma_{\hat{d}}$	1
σ_b	0.5°	$\sigma_{\hat{a}_P}$	0.5
m	200	$\sigma_{\hat{\psi}}$	18°
λ	3	σ_x	5
f_s	1Hz	σ_y	5

Table 3.2 Tracking algorithms

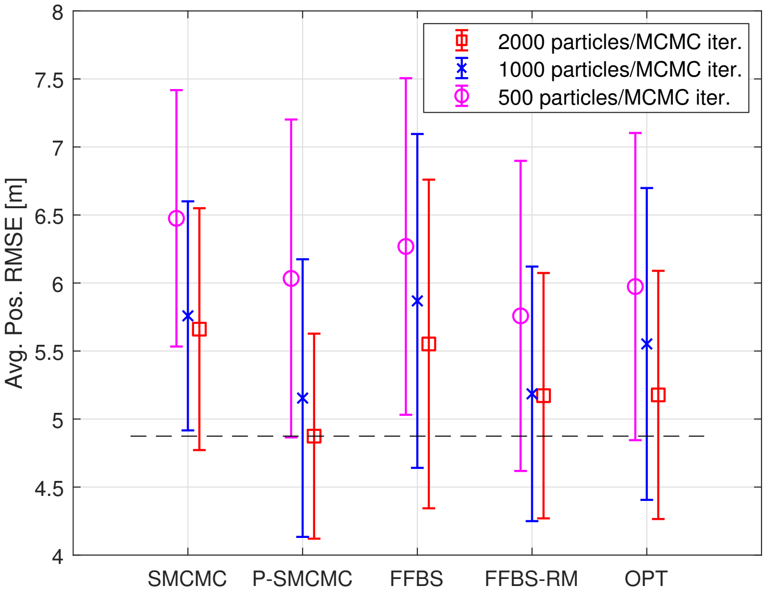
Tracking algorithm	Name
SIR w/ locally optimal proposals	OPT
SIR w/ section-wise optimal proposals	FFBS
SIR w/ section-wise optimal proposals and resample-move	FFBS-RM
SMCMC w/ section-wise optimal proposals	SMCMC
Parallel SMCMC w/ section-wise optimal proposals	P-SMCMC

Table 3.3 Algorithm Parameter Specification

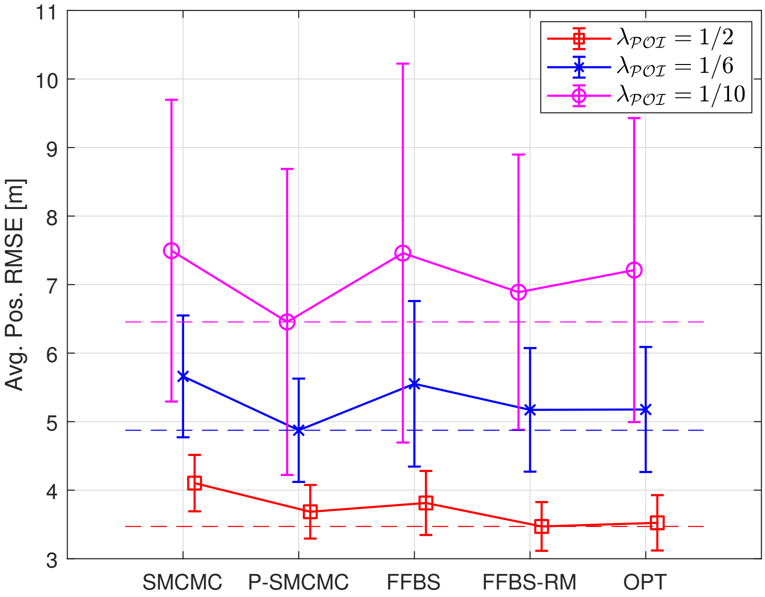
Algorithm Parameter	Symbol	case 1	case 2	case 3
# of particles/MCMC iterations	N_p/N_{iter}	500	1000	2000
# of RM steps	N_{RM}	1	1	1
Burn-in period	N_{Burn}	25	100	200
# of parallel chains	N_{Chain}	4	4	4

roughly equal running time for the considered algorithms. The results of the average position RMSEs and their associated standard deviations are shown in Fig. 3.2a, from which it is clear that the proposed P-SMCMC algorithm improves the tracking performance compared to the generic SMCMC. It also gives the best estimates with enough iterations. However, the performance of SMCMC based algorithms drops significantly when not running the chains for long enough time. As each parallel chain is run for just a quarter of the total number of iterations, P-SMCMC gives poor results with 500 iterations. Although FFBS-RM turns out to be more robust compared to the other IS-based methods, it actually requires more computational power as it has to do MCMC one or multiple times ($N_{RM} \geq 1$) for each resampled particle. A bootstrap SIR filter, in which states are proposed from their priors, is also implemented for comparison. However, the poor performance of the filter (mean of position RMSEs: 16.17m; standard deviation of position RMSEs: 13.64m) renders it unsuitable to be included in the figure.

A further experiment, in which we vary the number of received position measurements by adapting the values of $\lambda_{\mathcal{POI}}$, is conducted to assess the capabilities of the proposed algorithms. Position RMSE is again used to quantify the comparison. Fig. 3.2b shows that while the best performance is achieved by IS-based SMC filters (OPT and FFBS-RM) provided frequent position observations, P-SMCMC is able to accommodate more challenging scenarios. This can be attributed to the fact that MCMC-based scheme is more efficient when sampling from high-dimensional posterior



(a) Tracking performance of different algorithms on data synthesised according to the intrinsic coordinate model, with $\lambda_{\mathcal{POI}} = 1/6$; the errorbars represent average (and standard deviations) of position RMSEs over 100 datasets.



(b) Algorithm performance evaluated with respect to varied number of random position measurements; $N_p = N_{iter} = 2000$.

Fig. 3.2 Simulation results on 100 trajectories.

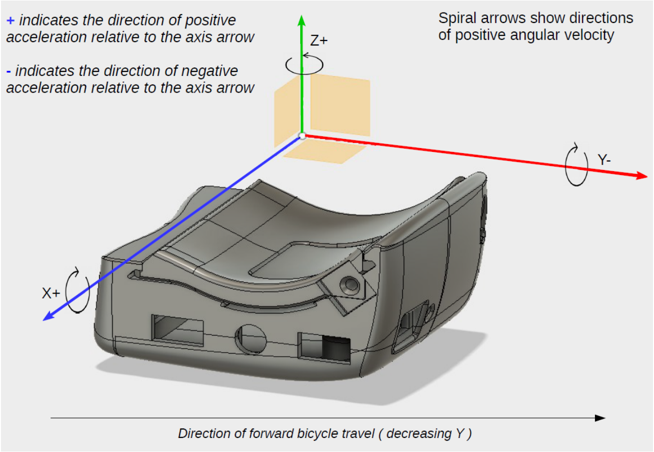

+ indicates the direction of positive acceleration relative to the axis arrow
- indicates the direction of negative acceleration relative to the axis arrow
Z+
Spiral arrows show directions of positive angular velocity
Y-
X+
Direction of forward bicycle travel (decreasing Y)

Fig. 3.3 Bike-mounted IMU orientation: $Z+$ is aligned to the surface normal with the bike being held still on the flat ground

(corresponding to a long waiting period between two consecutive position measurements). This result can serve as a guideline for algorithm selection in different tracking applications.

3.4.2 Manoeuvring Object Tracking: Track Cycling

The proposed methods are also evaluated in an on-line localisation/tracking scenario where the aim is to track a fast-moving bicycle in an indoor velodrome environment using inertial sensors and occasional position measurements provided by timing lines on the velodrome track. To be in accordance with the actual circumstance, it is necessary to introduce the sensor system used in our experiments.

Bike sensors Firstly, direct speed data are available so that the speed measurement model, as per (3.12), can be used. An IMU is mounted at the bottom bracket shell of the bike frame, with its orientation relative to the bike shown in Fig. 3.3. When cycling around a velodrome track, direct measurements of turn rate may be unavailable. This is because the geometry of the track and the bike sway prevent the vertical axis of gyroscope (i.e. $Z+$ in Fig. 3.3) from being aligned with the normal vector of the plane of motion. However, recall that this chapter is mainly concerned with 2D tracking model we still adopt (3.13) as our turn rate measurement model. Consequently, angular speed from the gyroscope's Z-axis is treated as a direct measurement of the rate of heading. We do not use acceleration measurements given by the accelerometer in this test since the sensor measures not only the acceleration caused by the cyclist but also the gravitational acceleration. Without an accurate modelling of the bike tilt angle, the extremely high readings from the accelerometer, caused by the gravity, are useless to tracking algorithms. Furthermore, direct distance measurement is not available, either.

Position measurements Unlike outdoor sports where it is possible to take advantage of global navigation systems to help localise an object, we need additional positioning systems for indoor track cycling. Here, two additional systems are available for providing position information:

(**a**). There are 9 lines (0m, 25m, 50m, 100m, 115m, 125m, 150m, 200m and 240m) across the velodrome track, see Fig. 3.4, and the crossing times for these lines can be measured. Such measurements are treated as a Gaussian measurement of the true position, $\hat{\mathbf{p}}_n$, at the measured crossing time t_n, truncated to lie within the track width:

$$p(\hat{\mathbf{p}}_n|\mathbf{p}_n) \propto \mathcal{N}(\hat{\mathbf{p}}_n|\mathbf{p}_n, \Sigma_{\text{TL}}) \tag{3.47}$$

where $\hat{\mathbf{p}}_n$ and Σ_{TL} are respectively centred on and aligned with each timing line (TL). When accurate position information is not available, Σ_{TL} would be set very wide along the timing line. Fig. 3.4 shows the positions of timing lines along the track and the corresponding Gaussian distributions for two of the timing lines.

(**b**). For certain timing lines along the straights (i.e. 0m, 100m, 115m, 125m and 240m) a camera system measuring lateral distance to the track's inner black line has been deployed around the velodrome, see [Carey et al., 2017]. In these cases the same truncated Gaussian is employed, but with a narrower covariance function centred on the measured lateral position.

On-track detection In addition to the above measurements, we also possess point cloud data acquired by laser scanning. This geometry information allows us to determine whether or not a particle stays physically within the track width. We can then assign zero weight to an out-of-track particle and resample particles when, for example, the effective sample size (ESS) is lower than a pre-defined threshold. Currently this detection mechanism is implemented only for OPT in which it is easy to detect whether a particle will go off-track at the next time epoch. For schemes with section-wise proposal kernels, we will not know whether a particle is moving off-track until the backward sampling step is completed. This more elaborate constraint enforcement is left as a topic of future exploration.

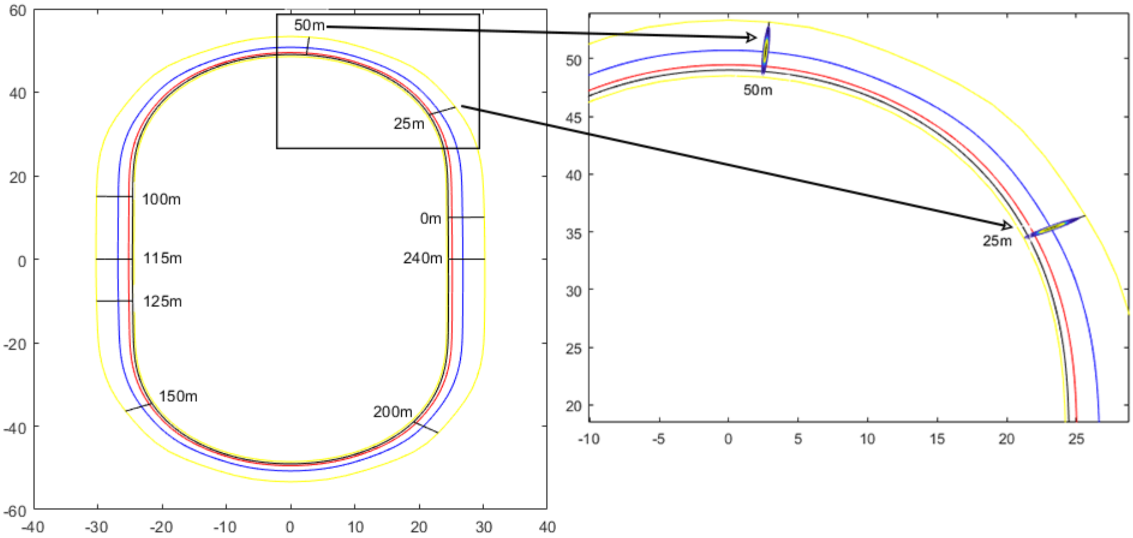


Fig. 3.4 Left: top view of the track with timing lines plotted in black. Yellow contours represent the edges of a velodrome track. Right: truncated Gaussian distributions on 25m and 50m timing lines.

Results on Synthetic Track Cycling Data

As the ground truth for the real data is not available, we synthesise several different datasets, each of which mimics a 2D cycling trajectory on the same velodrome track. By doing this, we could test and validate the tracking performance of our methods. Our idea of synthesising track cycling data is to run particle filters on the measurements acquired during the time when a cyclist cycled around a track for several laps. Then we adopt the output of the filters as our ground truth for further use. The parameters we use to generate the synthetic data are given in Table 3.4. Also note that the sensor measurements are logged every 0.1 seconds in order to be in line with the actual scenario. In total, there are 5 datasets, each corresponding to a 2- or 3-lap on-track trajectory.

As noted above, there is a difficulty in designing effective on-track detection mechanisms to accommodate the geometry constraint for the sequential batch algorithms. Although a simple solution, in which a point-by-point examination is conducted after the generation of a whole section, may be adopted, it is prone to be inefficient. For MCMC-based methods, this implies that a long period of time has to be waited until a satisfactory number of in-track paths are obtained. As a result, for this on-line application we decide to mainly focus on OPT as well as its equivalent bootstrap SIR filter (SIR). The section-wise scheme FFBS is still implemented for comparison while the other algorithms are left out.

Table 3.4 Parameters used for generating synthetic cycling data

λ	m	μ_T	σ_T	σ_P	σ_b	$\sigma_{\hat{v}}$	$\sigma_{\hat{\psi}}$	f_s
1	100	0	50	3000	0.5°	0.5	18°	10Hz

Table 3.5 Testing results across 5 datasets (10 runs for each algorithm). Particles used in OPT, FFBS and SIR are 2000, 5500 and 2000, respectively. Numbers given are means (and standard deviations) of position RMSEs.

Algorithms	w/ corner position?	Position RMSE [m]
OPT	no	0.681 (0.071)
	yes	0.526 (0.055)
FFBS	no	0.997 (0.274)
	yes	0.811 (0.097)
SIR	no	3.992 (0.781)
	yes	3.993 (0.568)

Now we are in a position to present performance comparison based on position RMSE values between different algorithms. Two testing scenarios are considered: one is in line with the practical situation where only timing information is available when a cyclist reaches one of the corner timing lines (i.e. 25m, 50m, 150m and 200m) whereas the other assumes that the relative distance between the bike and the black measurement line can be observed on these lines. In the first scenario, constrained Gaussian distributions with very large covariance are placed on the corner timing lines, showing no preference for particles that are able to get to one of the lines in time. Smoothed state estimate is used for computing the RMSEs as in the previous section.

The overall results are given in Table 3.5 from which it is clear that having lateral position measurement at corner lines helps reducing the tracking error. While being sufficiently robust in coping with less informative measurements, OPT outperforms both of the other two methods in the two testing scenarios. The lack of an on-track detection mechanism has rendered FFBS less effective than OPT. The bootstrap SIR filter has the worst performance as it uses state transition density as its proposal kernel. Drawing samples from the Gaussian prior results in more diverse heading rates and thus worse tracking performance. It is common that in the bootstrap SIR filter many particles fail to get to timing lines in time, which further degrades the performance of the filter. In contrast, the proposed methods perform better as the mismatch between prior predictive distribution and posterior distribution is mitigated.

A better visualisation with respect to the tracking performance for each algorithm can be found in Fig. 3.5 where the time evolution of the position errors and the corresponding empirical cumulative density functions with respect to one run on one of the datasets are plotted. The blue and red vertical lines on the left denote the timings of crossing corner and straight timing lines, respectively. It can be seen from the top left graph that the proposed schemes can keep the estimation errors at low levels when corner position measurements are provided. In this case, 85% of the position errors of the OPT are below $0.601m$ while the same percent of the errors in the FFBS lie below $0.845m$. As for the case where only timing information is provided around the corner, the OPT scheme still keeps 85% of the errors under $0.896m$. And it becomes more clear that without the on-track constraint and the corner position measurements, the performance of the FFBS scheme decreases.

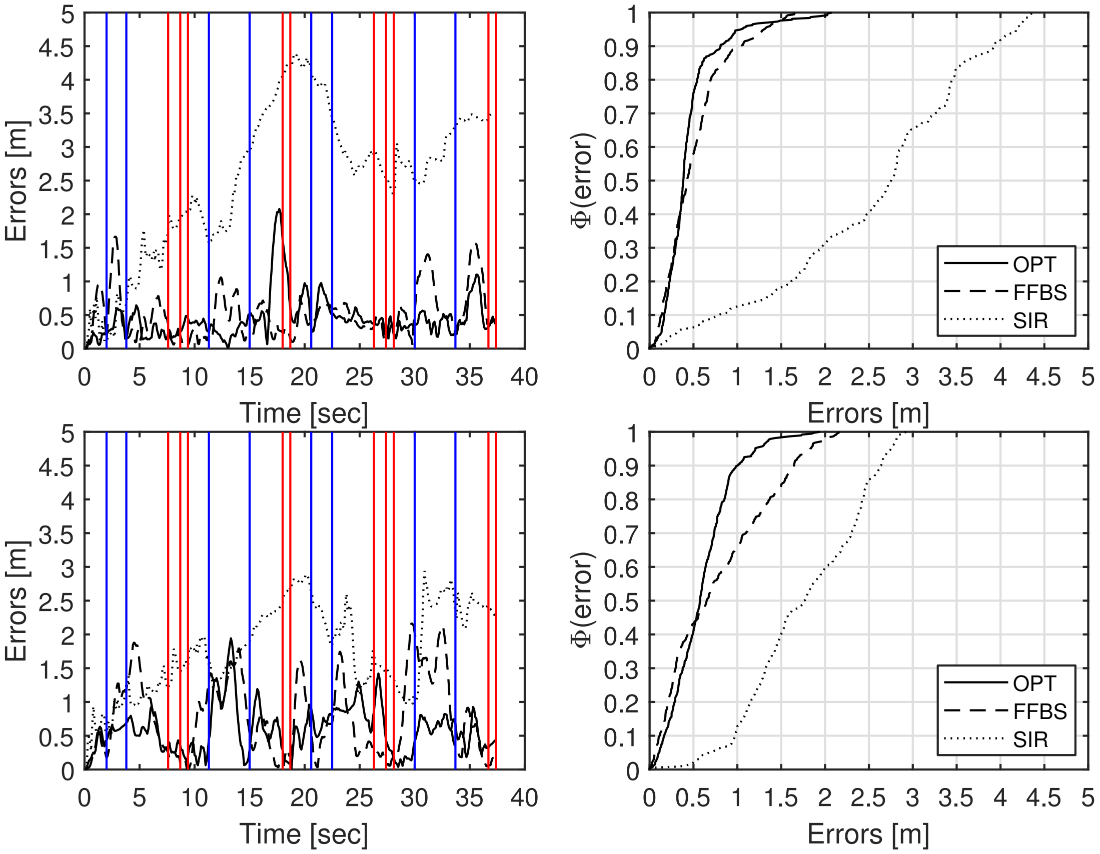


Fig. 3.5 Left: position errors over time, with(top left)/without(bottom left) corner position measurements. Blue vertical lines: timings at the corner timing lines; Red vertical lines: timings at the timing lines on the straights. Right: empirical CDFs of position errors, with(top right)/without(bottom right) corner position measurements.

Preliminary Results on Real Data

In practice, neither lateral position measurements at corner timing lines nor ground truth is available. To get a sense of the performance of the proposed algorithms under this circumstance, we ask a cyclist to follow a pre-defined trajectory on the track, which is defined as below:

1. from 115m timing line to 35m line, the cyclist stays on the black measurement line.
2. at 35m line, the cyclist starts a vector from the black measurement line and end up being around 220cm (slightly below the blue line which is 250cm away from the black measurement line) at 50m timing line.
3. from 50m timing line on, the cyclist dips down and cycles on the red line.
4. from 110m line (i.e. between 100m and 115m timing line) to 115m timing line, the cyclist drops again from the red line to the black line.
5. repeat for a few laps.

Lateral position measurements and timing measurements are synchronised with the on-bike sensor measurements whose sampling rate is 10Hz. Our methods is then tested on a collected dataset. As shown in Fig. 3.6, the smoothing trajectory obtained from OPT matches the defined trajectory well. We also plot 95% confidence ellipses along the filtered trajectory (not shown) occasionally. From the magnified area 2-3 (i.e. corresponding to the step 2 and 3), it can be seen that the ellipse becomes smaller because just a subset of the particles get to the line in time. As the particles are moving between the 50m line and the 100m line, the uncertainty becomes gradually larger as the model relies only on the IMU and speed measurements. However, owing to the optimal kernels and the on-track constraint the ellipses can still cover the red line around which the rider is positioned. This guarantees that the tracking model can use the accurate lateral position measurements to get back to the correct path when the rider arrives at the 100m timing line. As to FFBS, although there are more uncertainty due to the unconstrained particle motion it can still follow the pre-defined path. The example trajectory obtained by FFBS is not plotted here as it is visually very similar to that of OPT.

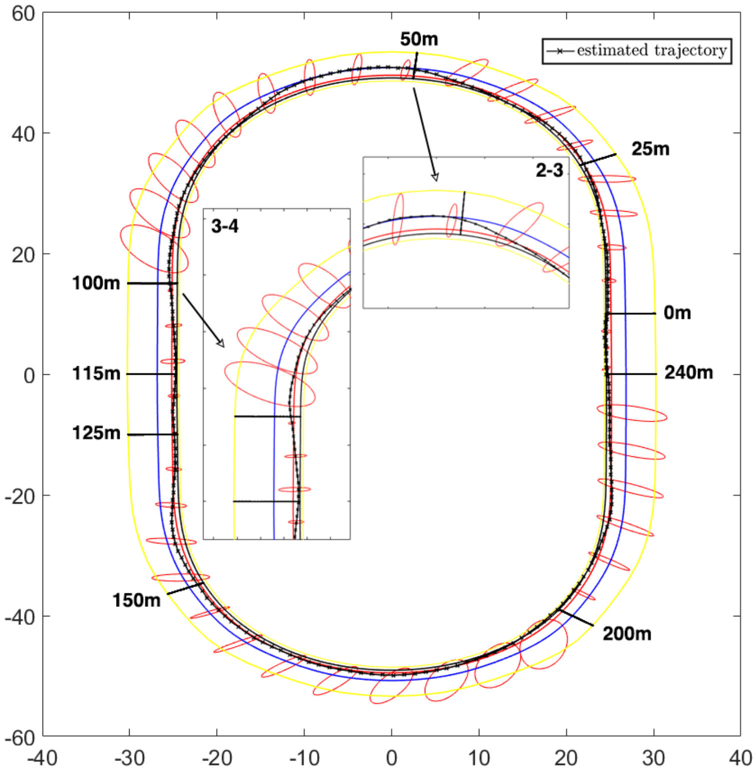


Fig. 3.6 A 1-lap smoothed trajectory estimated by OPT, with 95% confidence ellipses in red. Track (velodrome) description: yellow lines depict the edges of the velodrome; Blue line is the stayer's line; red line between the blue and the black line is the sprinter's line; black line is the measurement line.

Table 3.6 Parameters (ground vehicle tracking)

Symbols	Values
$\mu_T, \sigma_T, \sigma_P, \sigma_b$	$10, 30, 1000, 0.005°$
$\sigma_{\hat{v}}, \sigma_{\hat{a}_T}, \sigma_{\hat{a}_P}, \sigma_{\hat{\dot{\psi}}}, \sigma_x, \sigma_y$	$1, 2, 2, 15°, 2, 2$
m, λ, f_s	$200, 3, 10\mathrm{Hz}$
$N_p, N_{iter}, N_{RM}, N_{chain}, N_{burn}$	$2000, 2000, 1, 4, 2$

3.4.3 Manoeuvring Object Tracking: Vehicle Tracking

KITTI [Stiller and Urtasun, 2013] ground vehicle datasets are chosen to test our algorithms as they contain high precision RTK-GPS measurements which can be treated as ground truth. Because the IMU is fixed in the intrinsic frame of the ground vehicle, data from accelerometer can be used here. The idea is to test the algorithms in GPS-denied (very infrequent) scenarios and hence intermittent position data are again simulated according to a Poisson process. Relevant parameters are given in Table 3.6.

Fig. 3.7 shows tracking results of different methods on a 66-second dataset, given random initial heading angle (uniformly in $[0, 2\pi]$) and $\lambda_{\mathcal{POI}} = 1/5$. The red shaded area along the RTK trajectory represents the precision of the ground truth position measurements. The relatively poor performance of OPT, especially when it is around the area without position measurements, highlights the fact that it is a 'one-shot' sequential algorithm. It cannot 'look ahead' to use coming position measurements as a guidance, which is a main difference between it and the other algorithms. Although enhancement may be achieved by running a particle smoother backwards for an adequate number of repetitions, it is computationally expensive. On the contrary, FFBS-RM, SMCMC and P-SMCMC improve the proposed states and the retrospective performance while running in a sequential batch sense and maintaining relatively low computational cost. This characteristic becomes more prominent as in Fig. 3.8.

3.5 Summary and Conclusions

In this chapter, novel Bayesian inference methods are proposed for the fusion of inertial (accelerometer, gyroscope, speedometer and distance) and asynchronous position measurements, based on an intrinsic coordinate model. The proposed modelling allows an improved use of the model structure, leading to enhanced performance compared to the basic particle filtering scheme in challenging tracking scenarios. While the experimental results on the synthetic and real vehicle data reveal that a novel parallel chain based modification to the generic SMCMC has great potential in unconstrained, GPS-denied tracking scenarios, the results on cycling data show that a particle filter with the locally optimal kernels and an appropriate track edge detection mechanism can be desirable in a constrained tracking environment. Moreover, the good performance of the parallel SMCMC algorithm encourages further exploration on the effects of using different numbers of parallel chains and introducing more deliberate interaction among independent chains. As noted in Section 3.1.1, the use of continuous-time intrinsic coordinate dynamic model in a fixed rate tracking setting has been largely limited to approximations such as discretisation and linearisation, it is our hope that the introduction of the efficient proposal kernels as well as the corresponding inference algorithms will help shed more light on this class of models.

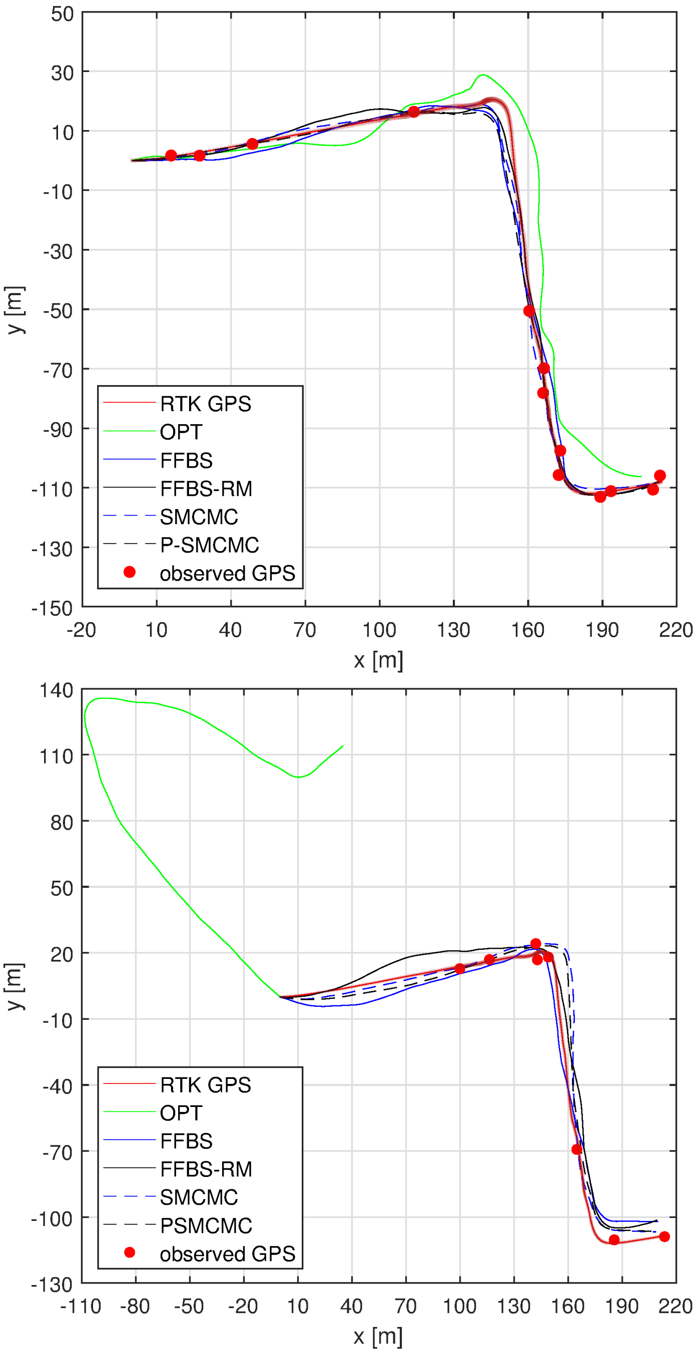


Fig. 3.7 Two independent runs (occasional GPS data are generated randomly in each run.) on the KITTI dataset (“2011_09_26_drive_0117_sync”).

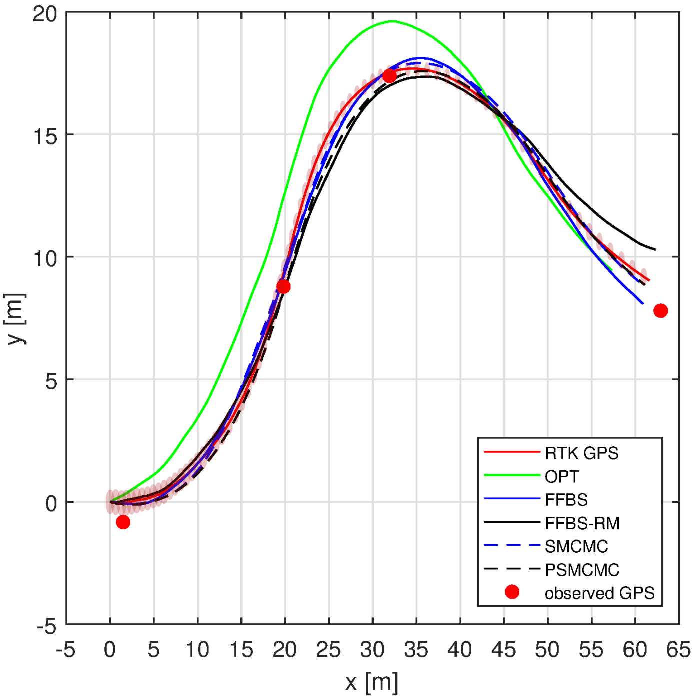


Fig. 3.8 Another KITTI dataset (“2011_09_26_drive_005_sync”), with merely 4 position observations.

Appendix

Appendix 3.A Locally Optimal Kernels

3.A.1 Speed Proposal Kernels

Here we show how to build the proposal kernel for v with the coming speed measurement, according to (3.8) and (3.12):

$$
\begin{aligned}
q(v_n|\hat{v}_n, v_{n-1}) &= p(v_n|\hat{v}_n, v_{n-1}) \\
&= \frac{p(\hat{v}_n|v_n)p(v_n|v_{n-1})}{p(\hat{v}_n|v_{n-1})} \\
&\propto \mathcal{N}(\hat{v}_n|v_n, \sigma_{\hat{v}}^2)\mathcal{N}(v_n|a_v v_{n-1} + b_v, \sigma^2) \\
&= \mathcal{N}(v_n|m_v, \sigma_v^2) \qquad (3.48)
\end{aligned}
$$

with

$$
\begin{aligned}
m_v &= av_{n-1} + b + \sigma_v^2(\hat{v}_n - a_v v_{n-1} - b_v)/(\sigma^2 + \sigma_{\hat{v}}^2) \\
\sigma_v^2 &= \sigma^2 - \sigma^4/(\sigma^2 + \sigma_{\hat{v}}^2) \\
a_v &= e^{-\Delta\tau_n\lambda/m} \\
b_v &= \mu_T(1 - e^{-\Delta\tau_n\lambda/m})/\lambda \\
\sigma^2 &= \sigma_T^2(1 - e^{-\Delta\tau_n\lambda/m})^2/\lambda^2
\end{aligned}
$$

and the normalising constant used for updating importance weights

$$
\begin{aligned}
p(\hat{v}_n|v_{n-1}) &= \int p(\hat{v}_n|v_n)p(v_n|v_{n-1})dv_n \\
&= \mathcal{N}(\hat{v}_n|a_v v_{n-1} + b_v, \sigma^2 + \sigma_{\hat{v}}^2) \qquad (3.49)
\end{aligned}
$$

Routinely, acceleration measurement $\hat{a}_{T,n}$ can be incorporated based on (3.48) and (3.17):

$$\begin{aligned} q(v_n|v_{n-1}, \hat{v}_n, \hat{a}_{T,n}) &= \frac{p(\hat{a}_{T,n}|v_n, v_{n-1})p(v_n|\hat{v}_t, v_{n-1})}{p(\hat{a}_{T,n}|\hat{v}_n, v_{n-1})} \\ &\propto p(\hat{a}_{T,n}|v_n, v_{n-1})p(v_n|\hat{v}_n, v_{n-1}) \\ &\quad = \mathcal{N}(\hat{a}_{T,n}|c(v_n - v_{n-1}), \sigma^2_{\hat{a}_T})\mathcal{N}(v_n|m_v, \sigma^2_v) \\ &= \mathcal{N}(v_n|m_v^\star, (\sigma_v^\star)^2) \end{aligned} \tag{3.50}$$

where

$$\begin{aligned} m_v^\star &= (\sigma_v^\star)^2\left(\frac{m_v}{\sigma_v^2} + \frac{c \cdot (\hat{a}_{T,n} + c \cdot v_{n-1})}{\sigma^2_{\hat{a}_T}}\right) \\ (\sigma_v^\star)^2 &= \left(\frac{1}{\sigma_v^2} + \frac{c^2}{\sigma^2_{\hat{a}_T}}\right)^{-1} \\ c &= \frac{\lambda}{m}\frac{e^{-\frac{\Delta\tau_n\lambda}{m}}}{1 - e^{-\frac{\Delta\tau_n\lambda}{m}}} \end{aligned}$$

and

$$p(\hat{a}_{T,n}|\hat{v}_n, v_{n-1}) = \mathcal{N}\Big(\hat{a}_{T,n}|c \cdot (m_v - v_{n-1}), \sigma^2_{\hat{a}_T} + c^2\sigma_v^2\Big) \tag{3.51}$$

Finally the speed proposal taking travelled distance information into consideration can be obtained via (3.50) and (3.20):

$$\begin{aligned} q(v_n|v_{n-1}, \hat{v}_n, \hat{a}_{T,n}, \hat{d}_n) &= \frac{p(\hat{d}_n|v_{n-1}, v_n)p(v_n|v_{n-1}, \hat{v}_n, \hat{a}_{T,n})}{p(\hat{d}_n|v_{n-1}, \hat{v}_n, \hat{a}_{T,n})} \\ &\propto p(\hat{d}_n|v_{n-1}, v_n)p(v_n|v_{n-1}, \hat{v}_n, \hat{a}_{T,n}) \\ &\quad = \mathcal{N}(\hat{d}_n|a_{d1}v_n + a_{d2}v_{n-1}, \sigma^2_{\hat{d}})\mathcal{N}(v_n|m_v^\star, (\sigma_v^\star)^2) \\ &= \mathcal{N}(v_n|\tilde{m}_v, \tilde{\sigma}_v^2) \end{aligned} \tag{3.52}$$

with

$$\begin{aligned}
\tilde{m}_v &= \tilde{\sigma}_v^2 \Big(\frac{m_v^\star}{(\sigma_v^\star)^2} + \frac{a_{d1}}{\sigma_{\hat{d}}^2} (\hat{d}_n - a_{d2} v_{n-1}) \Big) \\
\tilde{\sigma}_v^2 &= \Big(\frac{1}{(\sigma_v^\star)^2} + \frac{a_{d1}^2}{\sigma_{\hat{d}}^2} \Big)^{-1} \\
a_{d1} &= \frac{\Delta\tau_n}{1 - e^{-\frac{\Delta\tau_n \lambda}{m}}} - \frac{m}{\lambda} \\
a_{d2} &= \frac{m}{\lambda} - \frac{\Delta\tau_n}{1 - e^{-\frac{\Delta\tau_n \lambda}{m}}} e^{-\frac{\Delta\tau_n \lambda}{m}}
\end{aligned}$$

and

$$p(\hat{d}_n | v_{n-1}, \hat{s}_n, \hat{a}_{T,n}) = \mathcal{N}(\hat{d}_n | a_{d1} m_v^\star + a_{d2} v_{n-1}, \sigma_{\hat{d}}^2 + a_{d1}^2 (\sigma_v^\star)^2) \tag{3.53}$$

3.A.2 Turn Rate Proposal Kernels

In a similar fashion, in terms of (3.11) and (3.13) the proposal for $\boldsymbol{\alpha}$ conditioned on the speed and the turn rate measurement is given by

$$\begin{aligned}
q(\boldsymbol{\alpha}_n | \hat{\dot{\psi}}_n, \boldsymbol{\alpha}_{n-1}, v_n) &= p(\boldsymbol{\alpha}_n | \hat{\dot{\psi}}_n, \boldsymbol{\alpha}_{n-1}, v_n) \\
&\propto p(\hat{\dot{\psi}}_n | \boldsymbol{\alpha}_n, \boldsymbol{\alpha}_{n-1}, v_n) p(\boldsymbol{\alpha}_n | \boldsymbol{\alpha}_{n-1}, v_n) \\
&\quad = \mathcal{N}(\hat{\dot{\psi}}_n | B\boldsymbol{\alpha}_n, \sigma_{\hat{\dot{\psi}}}^2) \mathcal{N}(\boldsymbol{\alpha}_n | A\boldsymbol{\alpha}_{n-1}, C) \\
&= \mathcal{N}(\boldsymbol{\alpha}_n | m_{\boldsymbol{\alpha}}, P_{\boldsymbol{\alpha}})
\end{aligned} \tag{3.54}$$

where

$$\begin{aligned}
m_{\boldsymbol{\alpha}} &= A\boldsymbol{\alpha}_{t-1}^{(i)} + CB^T (BCB^T + \sigma_{\hat{\dot{\psi}}}^2)^{-1} (\hat{\dot{\psi}}_t - BA\boldsymbol{\alpha}_{t-1}^{(i)}) \\
P_{\boldsymbol{\alpha}} &= C - CB^T (BCB^T + \sigma_{\hat{\dot{\psi}}}^2)^{-1} BC \\
B &= \begin{bmatrix} 1 & 1 \end{bmatrix}
\end{aligned}$$

and

$$\begin{aligned}
p(\hat{\dot{\psi}}_n | \boldsymbol{\alpha}_{n-1}, v_n) &= \int p(\hat{\dot{\psi}}_n | \boldsymbol{\alpha}_n, \boldsymbol{\alpha}_{n-1}, v_n) p(\boldsymbol{\alpha}_n | \boldsymbol{\alpha}_{n-1}, v_n) d\boldsymbol{\alpha}_n \\
&= \mathcal{N}(\hat{\dot{\psi}}_n | BA\boldsymbol{\alpha}_{n-1}, BCB^T + \sigma_{\hat{\dot{\psi}}}^2)
\end{aligned} \tag{3.55}$$

When the measurement of perpendicular acceleration is available, the desired proposal kernel is obtained as follows, using (3.18) and (3.54):

$$\begin{aligned} q(\boldsymbol{\alpha}_n|\boldsymbol{\alpha}_{n-1}, v_n, \hat{\dot{\psi}}_n, \hat{a}_{P,n}) &= p(\boldsymbol{\alpha}_n|\hat{\dot{\psi}}_n, \hat{a}_{P,n}, \boldsymbol{\alpha}_{n-1}, v_n) \\ &\propto p(\hat{\dot{\psi}}_n|\boldsymbol{\alpha}_n) p(\hat{a}_{P,n}|v_n, \boldsymbol{\alpha}_n) p(\boldsymbol{\alpha}_n|\boldsymbol{\alpha}_{n-1}, v_n) \\ &\propto p(\hat{a}_{P,n}|v_n, \boldsymbol{\alpha}_n) p(\boldsymbol{\alpha}_n|\hat{\dot{\psi}}_n, \boldsymbol{\alpha}_{n-1}, v_n) \\ &= \mathcal{N}(\hat{a}_{P,n}|M\alpha_n, \sigma^2_{\hat{a}_P}) \mathcal{N}(\boldsymbol{\alpha}_n|m_\alpha, P_\alpha) \\ &= \mathcal{N}(\boldsymbol{\alpha}_n|\tilde{m}_\alpha, \tilde{P}_\alpha) \end{aligned} \tag{3.56}$$

with

$$\begin{aligned} \tilde{m}_\alpha &= m_\alpha + P_\alpha M^T (M P_\alpha M^T + \sigma^2_{\hat{a}_P})^{-1} (\hat{a}_{P,n} - M m_\alpha) \\ \tilde{P}_\alpha &= P_\alpha - P_\alpha M^T (M P_\alpha M^T + \sigma^2_{\hat{a}_P})^{-1} M P_\alpha \\ M &= [v_n, 0] \end{aligned}$$

and the normalising constant

$$p(\hat{a}_{P,n}|\hat{\dot{\psi}}_n, \boldsymbol{\alpha}_{n-1}, v_n) = \mathcal{N}(\hat{a}_{P,n}|M m_\alpha, M P_\alpha M^T + \sigma^2_{\hat{a}_P}) \tag{3.57}$$

Chapter 4

Bayesian Object Tracking: a Variable Rate Perspective

4.1 Introduction and Related Work

We may recall from Chapter 3 that in standard tracking applications a modelling assumption that state arrival time is synchronised with that of measurements is typically imposed. While being adequate to accommodate many tracking scenarios, this fixed rate premise can be a limiting factor for building more realistic models. For examples, when there exist abrupt changes (e.g. sharp manoeuvres of a fast moving target [Bunch and Godsill, 2012; Davey et al., 2016; Godsill et al., 2007; Whiteley et al., 2011] and price jumps in financial markets [Christensen et al., 2012; Johannes et al., 2009]) in the state process, it is not desirable to assume that their locations (in time) need to be in accordance with measurement time instants. Instead, if a state process whose state arrival time is typically unknown and not necessarily synchronised to the measurement process is considered, we will obtain a more natural class of models which is called "variable rate models" [Bunch and Godsill, 2012; Bunch and Godsill, 2013c; Christensen et al., 2012; Godsill and Vermaak, 2004; Godsill et al., 2007; Whiteley et al., 2011]. In this chapter, we turn our focus on the development of a variable rate manoeuvring model and the corresponding Bayesian inference methods.

Studies related to the inference problems on variable rate models can be traced back to the work by [Page, 1954] where it was referred to as a changepoint detection problem and the aim was to infer model parameters with variable dimensionality. Later, thanks to the rapid growth in computational power Markov chain Monte Carlo (MCMC) methods were extensively applied to similar model determination problems, as in the representative work by [Chib, 1995; Godsill, 2001; Green, 1995]. These works mostly focused on changepoint detection in which the main objective is to estimate changepoint locations as well as the corresponding parameters rather than to jointly infer both the changepoint and the hidden state. While it is not the case for model uncertainty problems, a large body of work dedicated to the joint estimation for both hidden state and changepoints can be found in communities such as target tracking [Godsill et al., 2007; Mazor et al., 1998], control [Costa et al., 2013] and economics [Blair Jr. and Sworder, 1975; Christensen et al., 2012]. For instance, in tracking applications of Interactive Multiple Models (IMMs) [Mazor et al., 1998; Xie et al., 2018] a target is allowed to switch its dynamics according to a finite set of candidate models capturing different types of motions (rectilinear, curvilinear and etc.). A known finite state Markovian transition matrix acting on dynamical models is usually assumed. As a result, an indicator variable that specifies which dynamical model is currently in use needs to be estimated jointly with the hidden state. There is also another standalone class of methods known as Markov Jump Systems (MJSs) [Costa et al., 2013; Shi and Li, 2015] which shares similar modelling principles with IMMs. However, most IMMs and MJSs are based on the assumption that changepoints must arrive at the same time instants as measurements. This is one of the essential differences between these switching hybrid systems and the variable rate models where changepoints are allowed to appear freely in their own time horizon.

It has been shown by [Bunch and Godsill, 2012; Godsill et al., 2007; Whiteley et al., 2011] that conditioning on an unknown sequence of changepoints, continuous-time dynamical models can be used to capture varied degrees of manoeuvrability of a target. Compared with [Bunch and Godsill, 2012; Godsill et al., 2007] where intrinsic coordinate dynamic models are considered, [Whiteley et al., 2011] adopts a coordinate-decoupled model assuming independent accelerations acting along each Cartesian coordinate. To solve the state and changepoint estimation problem associated with these variable rate models, general particle filtering methods which are referred to as variable rate particle filters (VRPFs) have been developed [Godsill and Vermaak, 2005; Godsill et al., 2007]. Apart from being useful in the field of object tracking, applications of variable rate models can be found in financial markets. In [Christensen et al., 2012], the jumps in market prices are treated as a sequence of changepoints that is estimated by a VRPF while the price itself is tackled by a Rao-Blackwellised estimation scheme because of the existence of a conditionally linear Gaussian structure. Other related work includes (but not restricted to) the Rao-Blackwellised VRPF developed for pitch estimation in audio applications [Zhang and Godsill, 2016] and the SMC samplers framework [Moral et al., 2006].

In contrast to the fixed rate model introduced in Chapter 3, here we consider an intrinsic coordinate model formulated within a variable rate framework. The fact that the intrinsic model has a closed form solution when piecewise-constant driving forces are provided has made it a natural variable rate model in which the states evolve in a deterministic manner conditioned on a sequence of changepoint times and manoeuvre parameters. The variable rate intrinsic model is more realistic compared to its fixed rate counterpart as it is able to capture both short- and long-term dependencies in the state trajectory. More importantly, the model to be introduced can be regarded as a generalisation to those in [Gilholm et al., 2005; Godsill and Vermaak, 2004; Godsill et al., 2007] and it offers a convenient way to fuse information from various sources under a variable rate particle filtering framework.

The proposed model also leads to an alternative effective solution to the problem of fusing asynchronous data, which is a common issue associated with real world tracking applications owing to different characteristics of sensors. A typical example is that the sampling rate of inertial measurement units (IMUs) is usually different from that of GPS units. Data fusion algorithms designed to deal with the asynchronicity present in sensor measurements can be found in various fields ranging from vehicle tracking [Hostettler and Djurić, 2015], target monitoring in wireless sensor networks [Safari et al., 2014; Zhu et al., 2014] to general object tracking applications [Schön et al., 2007]. Whereas most of the work relies on either conducting discretization for the state space model or imposing constraints on the sampling rate ratios between sensor systems, it should be clear that a continuous-time state space model along with a particle filter offers a more natural way to solve asynchronisity when the system is non-linear and/or non-Gaussian. In this chapter we will show that better estimation results can be obtained for certain challenging tracking problems by using a continuous-time dynamic model and by exploiting the model structure related to high frequency data (e.g. those from an IMU). Note that a more relevant work can be found in [Schön et al., 2007] where a linear and Gaussian measurement model is assumed to be available for partial measurements whose sampling frequencies are higher than the others. Consequently, a standard particle filter is used to approximate the state posterior when a lower frequency measurement is received, whereas the state estimation is handled by a set of Kalman filters between the receiving times of two consecutive slow measurements. However, the estimation for intermediate states in the paper is based on point estimates, which is in contrast to the sampling-based strategies taken in our work. Moreover, the approach is still based on the premise that the state arrival time needs to be constrained by the observation interval (i.e. a fixed rate setting).

In the following sections, we introduce the generalised version of the intrinsic coordinate model according to which data from inertial and other sensors can be easily fused. Based on the variable rate formulation of the proposed model, we show how variable rate particle filters with better performance compared to its bootstrap counterparts may be designed in the presence of asynchronous measurements. A variant of the simulation smoother [de Jong and Shephard, 1995] tailored to variable rate models is designed in order to overcome the difficulty in sampling from degenerated linear Gaussian state transition models. Also, we present an improvement scheme based on Reversible Jump Markov chain Monte Carlo [Green, 1995] moves for the resulting filtering framework. Algorithm performance is then demonstrated using synthetic and benchmark data. Furthermore, in the second last section we present solutions based on particle Markov chain Monte Carlo methods [Andrieu et al., 2010] to the joint state and parameter estimation problem.

4.2 Models

4.2.1 Variable Rate Models

The state in a variable rate model is defined as $\{\mathbf{x}_k\}_{k\in\mathbb{N}} = \{\tau_k, \mathbf{u}_k\}_{k\in\mathbb{N}} \in \mathbb{R}^+ \times \mathrm{E}$ with τ_k the changepoint occurrence time ($\tau_0 < \tau_1 < \cdots < \tau_n \leq \infty,\ \tau_k < \tau_{k+1}$ if $\tau_k < \infty$), $\mathbf{u}_k$ the vector of variables offering a complete and parsimonious description for the state process and E the state space of $\mathbf{u}_k$. It can be also viewed as the element in a Marked Point Process (MPP) where τ_k represents the event time at which a certain random event occurs while $\mathbf{u}_k$ denotes the type or mark of the event [Jacobsen, 2006]. If a Markovian assumption (can be relaxed) is imposed, the variable state transition density may be of the following form:

$$\begin{aligned} \mathbf{x}_k &\sim p(\mathbf{x}_k|\mathbf{x}_{k-1}) \\ &= p(\mathbf{u}_k|\mathbf{u}_{k-1}, \tau_k, \tau_{k-1})p(\tau_k|\mathbf{u}_{k-1}, \tau_{k-1}) \end{aligned} \tag{4.1}$$

From a Bayesian inference perspective, it will be of particular importance to obtain the joint density $p(\mathbf{x}_{0:K_n})$ of a sequence of variable rate states in the time interval $[0, t_n)$, with $\{t_0 = 0, t_1, \ldots, t_n\}$ the times at which observations $\mathbf{y}_{0:n}$ are made and $K_n = |\{k : \forall \tau_k < t_n\}|$ the number of states arriving before t_n. It has been shown in

[Bunch and Godsill, 2013c; Whiteley et al., 2011] that this density is given by

$$p(\mathbf{x}_{0:K_n}) = S(\mathbf{x}_{K_n}, t_n) \times p(\mathbf{x}_0) \prod_{k=1}^{K_n} p(\mathbf{x}_k|\mathbf{x}_{k-1}) \tag{4.2}$$

where the survival function $S(\mathbf{x}_k, t) = 1 - \int_{\tau_k}^{t} p(\xi \mid \mathbf{u}_k, \tau_k) d\xi$ gives the probability of the next jump arriving after t conditioned on $\mathbf{x}_k$. It should be noted that K_n is a random variable itself whose value is deterministic given a sequence of variable rate states and time t_n. Furthermore, if the state process is piecewise-deterministic, an interpolated state $\breve{\mathbf{u}}_n = h(\mathbf{x}_{K_n:K_n+1})$ at t_n, with $h(\cdot)$ being some deterministic function, can be defined. This function then serves to connect variable rate states to observations that are regularly sampled. Accordingly, the observation density can be written as $p(\mathbf{y}_n \mid \widehat{\mathbf{u}}_n)$ if independent observations are assumed. See [Godsill et al., 2007; Jacobsen, 2006] for detailed justifications regarding the existence of such a density. As a consequence, the joint probability of variable rate states and measurements up to time t_n, denoted as $p(\mathbf{x}_{0:K_n}, \mathbf{y}_{0:n})$, is given by

$$p(\mathbf{x}_{0:K_n}, \mathbf{y}_{0:n}) = S(\mathbf{x}_{K_n}, t_n) \times p(\mathbf{x}_0) \prod_{k=1}^{K_n} p(\mathbf{x}_k|\mathbf{x}_{k-1}) \prod_{i=0}^{n} p(\mathbf{y}_n|\breve{\mathbf{u}}_n) \tag{4.3}$$

4.2.2 Variable Rate Models for an Intrinsic Coordinate System

Now we will focus on the development of a continuous-time state-space representation associated with the intrinsic coordinate model. Building upon this, different tracking algorithms, either operating on a fixed-rate basis or a variable-rate basis, can be applied with minimal work of adaptation. Without the loss of generality the formulation presented is done in terms of a variable-rate setting where state arrival times are not necessarily synchronised with those of observations. Nevertheless, it will be straightforward to transform the model into a fixed-rate manner by setting the variable timings of state process to be identical to those of measurement process. To begin with, first recapitulate the following basic equations of curvilinear motion [Greenwood, 2006]:

$$\frac{d\psi}{dt} = \frac{T_P}{mv}, \quad \frac{dv}{dt} = \frac{T_T}{m} \tag{4.4}$$

where ψ is the heading angle relative to the x-axis, v is the speed, m is the mass of the target and a_P is the centripetal acceleration perpendicular to the tangential acceleration a_T. Considering the fact that a target, such as a car, a vessel or a bike, does not manoeuvre continuously, it is reasonable to assume the applied forces tangential to (T_T) and perpendicular to (T_P) the motion to be piecewise constant over the time interval $(\tau_k, \tau_{k+1}]$. If a resistance term whose value is proportional to the speed of the object in the tangential direction is included into (4.4), as in [Gilholm et al., 2005; Godsill et al., 2007], (4.4) can be re-written as

$$T_{T,k} = \lambda \frac{ds}{dt} + m\frac{d^2s}{dt^2} \tag{4.5}$$

$$T_{P,k} = m\frac{ds}{dt}\frac{d\psi}{dt}, \qquad t \in (\tau_k, \tau_{k+1}] \tag{4.6}$$

with s being the travelled distance/arc length along the path and λ the coefficient of resistance. As accelerations are of interest in most tracking applications instead of manoeuvring forces, accelerations a_T and a_P are directly included in our state vector. The state process presented here also falls into the category of piecewise deterministic processes (PDPs) since the states, as will be seen later, will evolve deterministically in continuous time before a random jump in the forces occurs. Note also (4.5) and (4.6) are in the same forms as (3.2) and (3.3), except that we now allow the forces to be piecewise constant within intervals defined by changepoint arrival times.

In the sequel, the variable rate intrinsic coordinate model will be introduced in terms of its tangential and perpendicular components, respectively. The state vector is expanded to be $\mathbf{x} = \{\boldsymbol{\theta}, \boldsymbol{\alpha}, \tau\}$ where $\boldsymbol{\theta}$ and $\boldsymbol{\alpha}$ stands for tangential and perpendicular components. Furthermore, we define τ_k^+, τ_k^- to be the times right after/before a manoeuvre at τ_k respectively and will use

$$\{\mathbf{x}_k, \mathbf{x}_{\tau_k}, \mathbf{x}_{\tau_k^+}\}$$

interchangeably in the subsequent sections.

Tangential components

The following ordinary differential equation (ODE) can be constructed according to the tangential equation (4.5):

$$\begin{bmatrix} \dot{s} \\ \dot{v} \\ \dot{a}_T \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{\lambda}{m} & 0 \\ 0 & 0 & -\frac{\lambda}{m} \end{bmatrix} \begin{bmatrix} s \\ v \\ a_T \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{T_T}{m} \\ 0 \end{bmatrix}$$

As $T_{T,k}$ is assumed piecewise constant, integrating the ODE from τ_k to $\tau_k + \Delta\tau$ and noticing that $a_T = -\frac{\lambda}{m}v + \frac{1}{m}T_T$ lead to the following solution

$$\begin{bmatrix} s_{\tau_k+\Delta\tau} \\ v_{\tau_k+\Delta\tau} \\ a_{T,\tau_k+\Delta\tau} \end{bmatrix} = \begin{bmatrix} 1 & \frac{m}{\lambda}(1-e^{-\frac{\Delta\tau\lambda}{m}}) & 0 \\ 0 & e^{-\frac{\Delta\tau\lambda}{m}} & 0 \\ 0 & e^{-\frac{\Delta\tau\lambda}{m}}(-\frac{\lambda}{m}) & 0 \end{bmatrix} \begin{bmatrix} s_{\tau_k} \\ v_{\tau_k} \\ a_{T,\tau_k} \end{bmatrix} + \begin{bmatrix} \frac{\Delta\tau}{\lambda} - \frac{m}{\lambda^2}(1-e^{-\frac{\Delta\tau\lambda}{m}}) \\ \frac{1}{\lambda}(1-e^{-\frac{\Delta\tau\lambda}{m}}) \\ \frac{1}{m}e^{-\frac{\Delta\tau\lambda}{m}} \end{bmatrix} T_{T,k} \tag{4.7}$$

Now, define $\boldsymbol{\theta} = [s, v, a_T]^T$ such that (4.5) becomes

$$T_{T,k} = \begin{bmatrix} 0 & \lambda & m \end{bmatrix} \boldsymbol{\theta}_{\tau_k^+} \tag{4.8}$$

Substituting (4.8) into (4.7) we get

$$\boldsymbol{\theta}_{\tau_k+\Delta\tau} = L_\theta(\Delta\tau)\boldsymbol{\theta}_{\tau_k^+} \tag{4.9}$$

with

$$L_\theta(\Delta\tau) = \begin{bmatrix} 1 & \Delta\tau & \frac{m\Delta\tau}{\lambda} - \frac{m^2}{\lambda^2}(1-e^{-\frac{\Delta\tau\lambda}{m}}) \\ 0 & 1 & \frac{m}{\lambda}(1-e^{-\frac{\Delta\tau\lambda}{m}}) \\ 0 & 0 & e^{-\frac{\Delta\tau\lambda}{m}} \end{bmatrix}$$

This means that, right after a change of the applied force, $\boldsymbol{\theta}$ follows a deterministic transition before another changepoint arrives at a future time (beyond $\tau_k + \Delta\tau$). Note that (4.9) holds under the assumption that s and v are continuous at τ_k (i.e. $v_{\tau_k} = v_{\tau_k^-} = v_{\tau_k^+}$ and $s_{\tau_k} = s_{\tau_k^-} = s_{\tau_k^+}$). Accordingly, the transition equation from τ_k^- to τ_k^+ (i.e. at the changepoint) can be written as

$$\boldsymbol{\theta}_{\tau_k^+} = A_\theta \boldsymbol{\theta}_{\tau_k^-} + \mathbf{h}_\theta T_{T,k}$$

where

$$A_\theta = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{\lambda}{m} & 0 \end{bmatrix}, \quad \mathbf{h}_\theta = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{m} \end{bmatrix}$$

If it is further assumed that the underlying distribution of the tangential force is Gaussian (i.e. $T_T \sim \mathcal{N}(\mu_T, \sigma_T^2)$), one will obtain the transition density at τ_k as

$$p(\boldsymbol{\theta}_{\tau_k^+} | \boldsymbol{\theta}_{\tau_k^-}, \tau_k) = \mathcal{N}\Big(\boldsymbol{\theta}_{\tau_k^+} | A_\theta \boldsymbol{\theta}_{\tau_k^-} + \mathbf{m}_\theta, Q_\theta\Big) \tag{4.10}$$

with

$$\begin{aligned} \mathbf{m}_\theta &= [0, 0, \mu_T/m]^T \\ Q_\theta &= \mathrm{diag}([0, 0, \frac{\sigma_T^2}{m^2}]) \end{aligned}$$

We now have the following transition densities according to which $\boldsymbol{\theta}$ will evolve:

(a). From right after the changepoint (τ_k^+) to some future time $\tau_k + \Delta\tau$ before a new changepoint arriving:

$$p(\boldsymbol{\theta}_{\tau_k + \Delta\tau} | \boldsymbol{\theta}_{\tau_k^+}, \tau_k) = \delta(L_\theta(\Delta\tau)\boldsymbol{\theta}_{\tau_k^+})$$

with $\delta(\cdot)$ being the Kronecker delta function.

(b). At the changepoint: (4.10).

(c). From right after the previous changepoint (τ_k^-) to right before the current changepoint (τ_k^+):

$$p(\boldsymbol{\theta}_{\tau_k^-} | \boldsymbol{\theta}_{\tau_{k-1}^+}, \tau_k, \tau_{k-1}) = \delta(L_\theta(\tau_k - \tau_{k-1})\boldsymbol{\theta}_{\tau_{k-1}^+})$$

As a result, a Gaussian state transition density for $\boldsymbol{\theta}$ from τ_{k-1}^+ to τ_k^+ can be obtained:

$$\begin{aligned} & p(\boldsymbol{\theta}_{\tau_k^+} | \boldsymbol{\theta}_{\tau_{k-1}^+}, \tau_k, \tau_{k-1}) \\ =& \mathcal{N}(\boldsymbol{\theta}_{\tau_k^+} | A_\theta(L_\theta(\tau_k - \tau_{k-1})\boldsymbol{\theta}_{\tau_{k-1}^+}) + \mathbf{m}_\theta, Q_\theta) \\ =& \mathcal{N}(\boldsymbol{\theta}_{\tau_k^+} | T_\theta(\tau_k - \tau_{k-1})\boldsymbol{\theta}_{\tau_{k-1}^+} + \mathbf{m}_\theta, Q_\theta) \end{aligned} \tag{4.11}$$

where

$$T_\theta(\Delta\tau) = A_\theta L_\theta(\Delta\tau) = \begin{bmatrix} 1 & \Delta\tau & \frac{\Delta\tau m}{\lambda} - \frac{m^2}{\lambda^2}(1 - e^{\frac{-\Delta\tau\lambda}{m}}) \\ 0 & 1 & \frac{m}{\lambda}(1 - e^{\frac{-\Delta\tau\lambda}{m}}) \\ 0 & -\frac{\lambda}{m} & e^{\frac{-\Delta\tau\lambda}{m}} - 1 \end{bmatrix}$$

It should be noted that if $\Delta\tau = \tau_k - \tau_{k-1}$ equals the fixed time interval between successive observations (4.11) will become a fixed-rate state transition equation as in standard tracking models.

Perpendicular components

Recall that the second row of (4.7) depicts how speed changes in the time interval $t \in (\tau_k, \tau_{k+1}]$ given a constant applied force. Based on this result, (4.6) can be solved for the change in heading angle as in [Godsill et al., 2007]:

$$\psi_{\tau_k+\Delta\tau} = \psi_{\tau_k} + \frac{T_{P,k}}{T_{T,k}}\left(\frac{\Delta\tau\lambda}{m} - \log\left|\frac{v_{\tau_k}}{v_{\tau_k+\Delta\tau}}\right|\right)$$

with $\Delta\tau \leq \tau_k - \tau_{k-1}$. Accordingly, a transition equation for heading angle from τ_{k-1}^+ to τ_k^+ can be expressed as

$$\psi_{\tau_k^+} = \psi_{\tau_{k-1}^+} + f(\boldsymbol{\theta}_{\tau_{k-1:k}^+}, \tau_{k-1:k}) \times a_{P,\tau_{k-1}^+} \tag{4.12}$$

where

$$f(\boldsymbol{\theta}_{\tau_{k-1:k}^+}, \tau_{k-1:k}) = \frac{m}{\lambda v_{\tau_{k-1}^+} + m a_{T,\tau_{k-1}^+}}\left(\frac{(\tau_k - \tau_{k-1})\lambda}{m} - \log\left|\frac{v_{\tau_{k-1}^+}}{v_{\tau_k^+}}\right|\right)$$

In applications where gyroscope data is available a bias terms is often included to capture instrumental bias/modelling error associated with measured angular velocities. The fact that this bias term varies very slowly over time has led to tracking applications where this term is treated as a constant variable, such as those in [Huang et al., 2010; Solin et al., 2018]. However, for long-term robustness sometimes a time-varying setup for the bias is preferred [Klein and Drummond, 2004].

The bias term b in this chapter is also assumed to be piecewise constant. It has the same underlying "sojourn" times as those of applied forces. Alternatively, if the bias is modelled as a continuous-time conditionally linear Gaussian process, a *Rao-Blackwellised* estimation scheme [Robert and Casella, 2004] may be adopted. Since the bias term is closely related to the heading angle, the state vector for perpendicular components is designed as

$$\boldsymbol{\alpha} = [\psi, a_P, b]^T$$

with b being the bias term. The corresponding transition model is then given by

$$\boldsymbol{\alpha}_{\tau_k^+} = T_\alpha(\boldsymbol{\theta}, \tau)\boldsymbol{\alpha}_{\tau_{k-1}^+} + \mathbf{h}_\alpha T_{P,k} + \mathbf{h}_b J_k \tag{4.13}$$

where

$$\begin{aligned} T_\alpha(\boldsymbol{\theta}, \tau) &= \begin{bmatrix} 1 & f(\boldsymbol{\theta}_{\tau_{k-1:k}^+}, \tau_{k-1:k}) & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ \mathbf{h}_\alpha &= \begin{bmatrix} 0 & \frac{1}{m} & 0 \end{bmatrix}^T \\ \mathbf{h}_b &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \\ J_k &\sim \mathcal{N}(0, \sigma_b^2) \end{aligned}$$

In the same manner of designing T_T, we assume the perpendicular force is also a Gaussian $T_P \sim \mathcal{N}(0, \sigma_P^2)$ and thus a linear Gaussian state space model can be obtained by conditioning on $\boldsymbol{\theta}_{\tau_{k-1:k}^+}$ and $\tau_{k-1:k}$:

$$p(\boldsymbol{\alpha}_{\tau_k^+} | \boldsymbol{\alpha}_{\tau_{k-1}^+}, \boldsymbol{\theta}_{\tau_{k-1:k}^+}, \tau_{k-1:k}) = \mathcal{N}(\boldsymbol{\alpha}_{\tau_k^+} | T_\alpha(\boldsymbol{\theta}, \tau)\boldsymbol{\alpha}_{\tau_{k-1}^+}, Q_\alpha) \tag{4.14}$$

with $Q_\alpha = \text{diag}([0, \frac{\sigma_P^2}{m^2}, \sigma_b^2])$. When there is no changepoint between τ_k and $\tau_k + \Delta\tau$ the state transition equation is

$$p(\boldsymbol{\alpha}_{\tau_k + \Delta\tau} | \boldsymbol{\alpha}_{\tau_k^+}, \boldsymbol{\theta}_{\tau_k^+}, \boldsymbol{\theta}_{\tau_k + \Delta\tau}, \tau_k, \tau_k + \Delta\tau) = \delta(L_\alpha(\theta, \tau)\boldsymbol{\alpha}_{\tau_k^+}) \tag{4.15}$$

with

$$L_\alpha(\theta, \tau) = \begin{bmatrix} 1 & f(\boldsymbol{\theta}_{\tau_k^+}, \boldsymbol{\theta}_{\tau_k + \Delta\tau}, \tau_k, \tau_k + \Delta\tau) & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

In the sequel, subscripts of $\boldsymbol{\theta}$ and τ in the condition will be removed for notational brevity. Note further that another feasible formulation may include heading rate $\dot{\psi}$ instead of perpendicular acceleration a_P within $\boldsymbol{\alpha}$, as in the last chapter. These two formulations are actually interchangeable given the speed, according to (4.6).

Position

For the intrinsic model based upon (4.5) and (4.6), a closed form solution for the Cartesian position $\mathbf{p} = [p_x, p_y]^T$ will not be available unless the damping factor λ equals zero [Bunch and Godsill, 2013c]. However, numerical approximation, such as Euler approximation, may be used to obtain the position over time via

$$\mathbf{p}_{t+\delta t} \approx \mathbf{p}_t + [\delta p_x, \delta p_y]^T \tag{4.16}$$

where $\delta p_x = v_t \cos(\psi_t)\delta t$, $\delta p_y = v_t \sin(\psi_t)\delta t$ and δt controls the accuracy of the numerical approximation. Note that it is optional to include the position $\mathbf{p}$ in the state vector as it can be deterministically computed given the entire sequence of the variable states, namely $p(\mathbf{p}_{0:n}|\widehat{\mathbf{u}}_{0:n}) = \delta(h(\mathbf{x}_{0:K_n}))$ at time t_n.

4.2.3 Measurement Models

Tangential components

Noisy observations of travelled distance $\hat{s}$, speed $\hat{v}$ and tangential acceleration $\hat{a}_T$ can be considered to be in a linear relationship with $\boldsymbol{\theta}$ when the body frames of inertial sensors coincides with target's intrinsic coordinate frame. As $\boldsymbol{\theta}$ will evolve in a deterministic manner conditioned on the most recent changepoint at τ_k, a measurement model for an augmented observation $\mathbf{y}_\theta = [\hat{s}, \hat{v}, \hat{a}_T]^T$ is given by

$$\mathbf{y}_{\theta,\tau_k+\Delta t} = M_\theta(L_\theta(\Delta t)\boldsymbol{\theta}_{\tau_k^+}) + \boldsymbol{\nu}_{\theta,\tau_k+\Delta t}, \quad \boldsymbol{\nu}_{\theta,\tau_k+\Delta t} \sim \mathcal{N}(\mathbf{0}, R_\theta) \tag{4.17}$$

with

$$\begin{aligned} M_\theta &= \mathbf{I}_3 \\ R_\theta &= \operatorname{diag}([\sigma_{\hat{s}}^2, \sigma_{\hat{v}}^2, \sigma_{\hat{a}_T}^2]) \end{aligned}$$

and Δt_i the time difference between τ_k and the observation time. Note that depending on the available sensors the measurement matrix M_θ can vary. It is also useful to group multiple observations such that an aggregate model is obtained as

$$\begin{aligned}\mathbf{Y}_{\theta,k} = \begin{bmatrix}\mathbf{y}_{\theta,\tau_k+\Delta t_1}\\ \vdots \\ \mathbf{y}_{\theta,\tau_k+\Delta t_N}\end{bmatrix} &= \begin{bmatrix}B_\theta(\Delta t_1)\\ \vdots \\ B_\theta(\Delta t_N)\end{bmatrix}\boldsymbol{\theta}_{\tau_k^+} + \mathbf{w}_\theta \\ &= Z_\theta \boldsymbol{\theta}_{\tau_k^+} + \mathbf{w}_\theta \end{aligned} \tag{4.18}$$

with $B_\theta(\Delta t_i) = M_\theta L_\theta(\Delta t_i)$, $\mathbf{w}_\theta \in \mathbb{R}^{2N}$ a zero-mean additive white noise satisfying $\mathbf{w}_\theta \sim \mathcal{N}(\mathbf{0}, \text{blkdiag}(R_\theta, \cdots, R_\theta))$ and $\{\Delta t_i\}_{i=1}^N$ the time differences between changepoint τ_k and the observations associated to it. To conclude, (4.11) and (4.18) form the state-space models for the tangential component vector $\boldsymbol{\theta}$.

Perpendicular components

Similarly, noisy measurements related to $\boldsymbol{\alpha}$ can be denoted as $\mathbf{y}_\alpha = [\hat{\dot{\psi}}, \hat{a}_P]^T$ with $\hat{\dot{\psi}}$ the angular velocity measured with respect to the axis of gyroscope that is perpendicular to the moving plane and $\hat{a}_P$ the side-way linear acceleration measured by accelerometers. Again, given the speed a conditionally linear Gaussian measurement model can be obtained as:

$$\mathbf{y}_{\alpha,\tau_k+\Delta t} = B_\alpha(\boldsymbol{\theta},\tau)\boldsymbol{\alpha}_{\tau_k^+} + \boldsymbol{\nu}_{\alpha,\tau_k+\Delta t}, \quad \boldsymbol{\nu}_{\alpha,\tau_k+\Delta t} \sim \mathcal{N}(\mathbf{0}, R_\alpha) \tag{4.19}$$

where

$$\begin{aligned} B_\alpha(\boldsymbol{\theta},\tau) &= \begin{bmatrix}0 & \frac{1}{v_{\tau_k+\Delta t}} & 1\\ 0 & 0 & 1\end{bmatrix} \\ R_\alpha &= \text{diag}([\sigma^2_{\hat{\dot{\psi}}}, \sigma^2_{\hat{a}_P}]) \end{aligned}$$

and by stacking the observations we have,

$$\mathbf{Y}_{\alpha,k} = \begin{bmatrix}\mathbf{y}_{\alpha,\tau_k+\Delta t_1}\\ \vdots \\ \mathbf{y}_{\alpha,\tau_k+\Delta t_N}\end{bmatrix} = Z_\alpha(\boldsymbol{\theta})\boldsymbol{\alpha}_{\tau_k^+} + \mathbf{w}_\alpha \tag{4.20}$$

with

$$\begin{aligned} Z_\alpha &= [B_\alpha(\boldsymbol{\theta},\tau), \cdots, B_\alpha(\boldsymbol{\theta},\tau)]^T \\ \mathbf{w}_\alpha &= \text{blkdiag}(R_\alpha, \cdots, R_\alpha) \end{aligned}$$

Position Measurements

The generic form of position measurement density at time t_n can be denoted as

$$p(\mathbf{y}_{\mathbf{p},n} \mid \mathbf{x}_{0:K_n}) = p(\mathbf{y}_{\mathbf{p},n} \mid \mathbf{p}_n) \tag{4.21}$$

If the measurement noise is assumed to be additive and white, we will have

$$p(\mathbf{y}_{\mathbf{p},n} \mid \mathbf{p}_n) = \mathcal{N}(\mathbf{y}_{\mathbf{p},n} \mid g(\mathbf{p}_n), R_{\mathbf{p}}) \tag{4.22}$$

with $g(\cdot)$ a linear or non-linear mapping function and $R_{\mathbf{p}} = \text{diag}([\sigma_{p_x}^2, \sigma_{p_y}^2])$. While the inertial measurements and speed measurements can be made with an identical sampling rate it is common that the position-related measurements, which may be provided by a GPS unit or other measurement units such as Doppler Radar and Laser scanner, may have a different (normally smaller) arrival rate. This issue of asynchronicity will be accounted for in our proposed algorithms.

4.3 Inference

4.3.1 Variable Rate Particle Filters

Many models of practical interests, including the models considered in this chapter, are of non-linear and/or non-Gaussian nature, and thus state estimations need to be carried out via non-linear filtering techniques such as particle filters [Cappé et al., 2007; Doucet et al., 2001; Godsill, 2019b; Gordon et al., 1993]. As mentioned earlier, a variant of particle filters that is tailored for variable rate models, termed variable rate particle filter (VRPF) [Bunch and Godsill, 2013c; Godsill and Vermaak, 2004; Godsill et al., 2007; Morelande and Gordon, 2009], has been introduced in order to accommodate the variable dimensionality of the hidden state sequence which is not known *a priori* within a given time interval. Typically, based on Bayes' rule the intractable posterior distribution of the hidden state sequence up to time t_n can be expressed as

$$p(\mathbf{x}_{0:K_n}|\mathbf{y}_{0:n}) \propto p(\mathbf{y}_n|\check{\mathbf{u}}_n)p(\mathbf{x}_{K_{n-1}+1:K_n}|\mathbf{x}_{0:K_{n-1}})p(\mathbf{x}_{0:K_{n-1}}|\mathbf{y}_{0:n-1}) \tag{4.23}$$

where

$$\begin{aligned} p(\mathbf{x}_{K_{n-1}+1:K_n}|\mathbf{x}_{0:K_{n-1}}) &= \frac{p(\mathbf{x}_{0:K_n})}{p(\mathbf{x}_{0:K_{n-1}})} \\ &= \frac{S(\mathbf{x}_{K_n}, t_n)}{S(\mathbf{x}_{K_{n-1}}, t_{n-1})} \times \left(\prod_{\forall j: t_{n-1} \leq \tau_j < t_n} p(\mathbf{x}_j|\mathbf{x}_{j-1}) \right) \end{aligned} \tag{4.24}$$

is the joint distribution of changepoints that occur within $[t_{n-1}, t_n)$. Note if there is no new changepoints occurring between t_{n-1} and t_n, we have $K_{n-1} = K_n$ (i.e. the next changepoint will arrive after t_n). VRPF now approximates the distribution (4.23) at time t_n via a set of weighted particles $\{\mathbf{x}_{0:K_n}^{(i)}\}_{1 \leq i \leq N_p}$ whose normalised weights $(\sum_{i=1}^{N_p} \omega_n^{(i)} = 1)$ are determined by sequential importance sampling (SIS). More specifically, to compute the importance weights at t_n for each particle we start with constructing an importance distribution $q(\cdot)$ of the entire state sequence as

$$q(\mathbf{x}_{0:K_n} \mid \mathbf{y}_{0:n}) = q(\mathbf{x}_{0:K_{n-1}} \mid \mathbf{y}_{0:n}) q(\mathbf{x}_{K_{n-1}+1:K_n}|\mathbf{x}_{0:K_{n-1}}, \mathbf{y}_{0:n}) \tag{4.25}$$

Similar to Section 2.3.4 of Chapter 2, we also intentionally impose the dependency on $\mathbf{y}_n$ in the proposal of past path $\mathbf{x}_{0:K_{n-1}}$ to allow the design of better proposals that take future information into account, under the variable rate setting. This means that instead of drawing samples from the empirical distribution represented by normalised particle weights $\{\omega_{n-1}^{(i)}\}_{1 \leq i \leq N_p}$ from last time instant, we may propose particles according to weights $\{\nu_{n-1}^{(i)}\}_{1 \leq i \leq N_p}$ that are chosen in the light of a new measurement $\mathbf{y}_n$:

$$\mathbf{x}_{0:K_{n-1}}^{(i)} \sim \sum_{i=0}^{N_p} \nu_{n-1}^{(i)} \delta_{\mathbf{x}_{0:K_{n-1}}^{(i)}}(\mathbf{x}_{0:K_{n-1}}) \tag{4.26}$$

An auxiliary sampling scheme (or "auxiliary particle filter") [Pitt and Shephard, 1999] based on this rationale has been proven very successful. For a thourough review on different look-ahead strategies that explore similar ideas, we refer interested readers to

[Lin et al., 2013]. Now the weight update equation at t_n for VRPF can be shown as

$$\begin{aligned}\tilde{\omega}_n^{(i)} &= \frac{p(\mathbf{x}_{0:K_n}^{(i)}|\mathbf{y}_{0:n})}{q(\mathbf{x}_{0:K_n}^{(i)}|\mathbf{y}_{0:n})} \\ &\propto \frac{p(\mathbf{x}_{0:K_{n-1}}^{(i)}|\mathbf{y}_{0:n-1})p(\mathbf{y}_n|\widehat{\mathbf{u}}_n^{(i)})p(\mathbf{x}_{K_{n-1}+1:K_n}^{(i)}|\mathbf{x}_{0:K_{n-1}}^{(i)})}{q(\mathbf{x}_{0:K_{n-1}}^{(i)}|\mathbf{y}_{0:n})q(\mathbf{x}_{K_{n-1}+1:K_n}^{(i)}|\mathbf{x}_{0:K_{n-1}}^{(i)},\mathbf{y}_{0:n})} \\ &= \frac{\omega_{n-1}^{(i)}}{\nu_{n-1}^{(i)}} \times \frac{p(\mathbf{y}_n|\breve{\mathbf{u}}_n^{(i)})p(\mathbf{x}_{K_{n-1}+1:K_n}^{(i)}|\mathbf{x}_{0:K_{n-1}}^{(i)})}{q(\mathbf{x}_{K_{n-1}+1:K_n}^{(i)}|\mathbf{x}_{0:K_{n-1}}^{(i)},\mathbf{y}_{0:n})} \end{aligned} \tag{4.27}$$

with $\tilde{\omega}_n^{(i)}$ being the unnormalised importance weight of the i-th particle. Note that (4.27) is akin to the general SIR formulation discussed in Chapter 2.3.4. As a result, different choices of selection weights ν_{n-1} will result in different versions of particle filters. For instance, by setting $\nu_{n-1} = \omega_{n-1}$ at each iteration the standard resampling scheme used in the original bootstrap filter [Gordon et al., 1993] is obtained. More often, one may choose $\nu_{n-1} = \omega_{n-1}$ when an effective sample size (ESS) based resampling criterion is met and $\nu_{n-1} = 1/N_p$ when it is not.

Note that for the rest of this chapter, except for the full system inference section, a biased resampling scheme as in [Godsill et al., 2007] will be adopted. The main idea behind this scheme is to preserve low weight particles which may turn out good in the future. More specifically, at the beginning of the selection stage N_i replicas of the i-th particle are firstly produced, with N_i determined by

$$N_i = \max(1, \lfloor N_p \varrho_{n-1}^{(i)} \rfloor)$$

where $\lfloor \cdot \rfloor$ is the floor function while $\varrho_{n-1}^{(i)}$ is the selection weight. Note that in the following experiments where this selection scheme is used, we set $\varrho_{n-1}^{(i)} = \omega_{n-1}^{(i)}$. Now replicas (offsprings) of the same particle will be assigned a modified selection weight $\nu_{n-1}^{(ij)} \propto N_i$, based on which the first stage weight for the j-th offspring of particle i can be computed as

$$\frac{\omega_{n-1}^{(ij)}}{\nu_{n-1}^{(ij)}} \propto \frac{\omega_{n-1}^{(ij)}}{N_i} \tag{4.28}$$

with $j = \{1, \ldots, N_i\}$. This scheme will result in a varied number of particles after the selection stage and hence in a fixed budget implementation a pruning step is introduced to keep N_p particles with the highest weights after the weighting stage.

4.3.2 Bayesian Fusion of Asynchronous Sensor Data

Denote all measurements made at times $\{t_0, \ldots, t_n\}$ as $\mathbf{Y}_{0:n} = \{\mathbf{Y}_\theta, \mathbf{Y}_\alpha, \mathbf{y}_\mathbf{p}\}_{0:n}$, the state vector as $\mathbf{x}_{0:K_n} = \{\boldsymbol{\theta}, \boldsymbol{\alpha}, \tau\}_{0:K_n}$ and $L \geq 1$ as the time lag between two consecutive slow (position in our case) measurements arriving at t_{n-l} and t_n. Essentially, L indicates the asynchronicity among position and the inertial measurements. Now we are in a position to factorise the posterior distribution at t_n, which is the timing when a position observation arrives, as

$$\begin{aligned}
p(\mathbf{x}_{0:K_n}|\mathbf{Y}_{0:n}) \propto\ & p(\mathbf{x}_{0:K_{n-L}}|\mathbf{Y}_{0:n-L}) \\
& \times p(\mathbf{Y}_{\{\alpha,\theta\},n-L+1:n-L+r}|\mathbf{x}_{0:K_{n-L}})p(\mathbf{y}_{\mathbf{p},n}|\mathbf{x}_{0:K_n}) \\
& \times p(\boldsymbol{\alpha}_{K_{n-L}+1:K_n}|\mathbf{Y}_{\alpha,n-L+r+1:n}, \{\boldsymbol{\alpha}, \tau\}_{K_{n-L}}, \{\boldsymbol{\theta}, \tau\}_{K_{n-L}+1:K_n}) \\
& \times p(\boldsymbol{\theta}_{K_{n-L}+1:K_n}|\mathbf{Y}_{\theta,n-L+r+1:n}, \{\boldsymbol{\theta}, \tau\}_{K_{n-L}}, \tau_{K_{n-L}+1:K_n}) \\
& \times p(\mathbf{Y}_{\alpha,n-L+r+1:n}|\{\boldsymbol{\alpha}, \tau\}_{K_{n-L}}, \{\boldsymbol{\theta}, \tau\}_{K_{n-L}+1:K_n}) \\
& \times p(\mathbf{Y}_{\theta,n-L+r+1:n}|\boldsymbol{\theta}_{K_{n-L}}, \tau_{K_{n-L}:K_n}) \\
& \times \frac{S(\mathbf{x}_{K_n}, t_n)}{S(\mathbf{x}_{K_{n-L}}, t_{n-L})} \prod_{\forall j: t_{n-L} \leq \tau_j < t_n} p(\tau_j|\tau_{j-1})
\end{aligned} \tag{4.29}$$

with r being the number of inertial and speed measurements observed within the time interval $(t_{n-L}, \tau_{K_{n-L}+1})$. This can be seen as a variable rate equivalent to the fixed rate sequential batch formulation given in Chapter 3. It should be noted that position measurements $\{\mathbf{y}_{\mathbf{p},0:n}\}$ can have equal or less number of components $(\leq n)$ compared with those of $\mathbf{Y}_{\theta,0:n}$ and $\mathbf{Y}_{\alpha,0:n}$ $(= n)$, although for the sake of notational brevity here we use the same subscripts "$_{0:n}$". We have also assumed that the latest position measurement $\mathbf{y}_{\mathbf{p},n}$ arrives synchronously with inertial measurements at t_n. Again, it does not have to be the case in practice. The above formulation in general is operable given any interval defined by $[t_{n-L}, t_n]$. Furthermore, while any other reasonable densities may be considered, in the sequel we choose a shifted gamma distribution for the interarrival times between changepoints, as per [Godsill et al., 2007]:

$$\tau_j \sim \tau_{j-1} + \tau_{\min} + \mathcal{G}(\alpha_\tau, \beta_\tau), \quad \tau_{\min} \geq 0 \tag{4.30}$$

with α_τ and β_τ being the shape and the rate parameters, respectively.

Under this particular factorisation, the problem of fusing inertial measurements with other asynchronous measurements can be effectively tackled by exploiting the linear/conditionally linear Gaussian models on which Kalman-type methods can be used. More precisely, instead of proposing state sequences $\boldsymbol{\theta}_{K_{n-L}+1:K_n}$ and $\boldsymbol{\alpha}_{K_{n-L}+1:K_n}$ from the prior distribution given in (4.24), we would like to draw these sequences from more informative proposals

$$q(\boldsymbol{\theta}_{K_{n-L}+1:K_n}|\mathbf{Y}_{\theta,n-L+r+1:n}, \{\boldsymbol{\theta}, \tau\}_{K_{n-L}}, \tau_{K_{n-L}+1:K_n}) \tag{4.31}$$

and

$$q(\boldsymbol{\alpha}_{K_{n-L}+1:K_n}|\mathbf{Y}_{\alpha,n-l+r+1:n}, \{\boldsymbol{\alpha}, \tau\}_{K_{n-L}}, \{\boldsymbol{\theta}, \tau\}_{K_{n-L}+1:K_n}) \tag{4.32}$$

where information in observed data are used. Simulating from these smoothing densities may be done by using the forward-filtering-backward-sampling (FFBS) technique [Carter and Kohn, 1994; Frühwirth-Schnatter, 1994] in which a joint draw from a smoothing density is carried out recursively given multiple subdraws from a bunch of conditional densities. However, here the components in state $\boldsymbol{\theta}$ (also applies to $\boldsymbol{\alpha}$) cannot be varied independently of each other due to their deterministic relationships enforced by the piecewise constant force. Therefore, transition models defined by (4.11) and (4.14) are degenerated ones because their process noise covariance matrices Q_θ and Q_α are not positive definite. This is not uncommon in tracking applications where people deal with degenerated dynamical models, see [Ahmad et al., 2018; Gustafsson et al., 2002; Li and Jilkov, 2003; Maskell, 2004; Zhang et al., 2018] for examples. While the degeneracy will not cause any problems when considering only sequential forward filtering, it renders backward simulation approaches such as FFBS not applicable unless mechanisms capable of tackling degeneracies are introduced. One of such mechanisms can be found in [Bunch and Godsill, 2013c] where the authors manage to do backward samplings on piecewise deterministic trajectories by introducing extra degrees of freedom in their dynamical model.

To address the issue of sampling from a degenerated model backwards, here we adopt an alternative multi-state sampler termed Simulation Smoother (SS) [de Jong and Shephard, 1995; Koopman, 1993] in which it is the disturbances η instead of the states that are sampled:

$$p(\eta_{0:n}|\mathbf{Y}_{1:n}) = p(\eta_n|\mathbf{Y}_{1:n}) \prod_{j=n-1}^{0} p(\eta_j|\eta_{j+1:n}, \mathbf{Y}_{1:n}) \tag{4.33}$$

As a complete state trajectory can be constructed forwardly using sampled disturbances $\eta_{0:n}$ given an initial state, problems associated with singular matrix inversions and disjoint deterministic state trajectories can be avoided.

Although it will be relatively straightforward to design a SS for fixed-rate models by supposing that the occurrence of changepoints coincides with that of observations (i.e. $\tau \triangleq t_n - t_{n-1}$ and the disturbances will be piecewise constant in the time interval $(t_{n-1}, t_n]$), it will be necessary to adapt SS such that it can work on variable-dimensional cases. To this end, consider first the following state space model:

$$\mathbf{Y}_k = Z_k \mathbf{a}_k + G_k \mathbf{n}_k \qquad (k = 1, 2, \ldots, K) \tag{4.34}$$

$$\mathbf{a}_{k+1} = T_k \mathbf{a}_k + \mathbf{m}_k + H_k \mathbf{n}_k \qquad (k = 0, 1, \ldots, K) \tag{4.35}$$

with known $\mathbf{m}_k$ and independent variable $\mathbf{n}_k \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_d) \in \mathbb{R}^d$ whose dimension d is of variable length at different time τ_k. It can be readily recognised that models represented by (4.11), (4.18) and (4.14), (4.20) fall into the above general model. Note again $\mathbf{a}_k = \mathbf{a}_{\tau_k} = \mathbf{a}_{\tau_k^+}$. Subsequently, disturbance η_k can be obtained as

$$\eta_k = F_k \mathbf{n}_k \tag{4.36}$$

where F_k is an arbitrary design matrix whose choice depends on the dimension of $\mathbf{n}_k$, namely d, and positions of disturbances in a dynamical model. Now, forward filtering can be done by running a Kalman filter forwardly. During the run, quantities such as innovation $\mathbf{e}_k$, scaled innovation covariance D_k and Kalman gain $K_{gain,k}$ will be stored over time $(k = 1, 2, \ldots, K)$:

$$\begin{aligned}
\mathbf{e}_k &= \mathbf{Y}_k - Z_k \mu_{k|k-1} - \mathbf{m}_k \\
D_k &= Z_k P_{k|k-1} Z_k^T + G_k G_k^T \\
K_{gain,k} &= (T_k P_{k|k-1} Z_k^T + H_k G_k^T) D_k^{-1}
\end{aligned} \tag{4.37}$$

where the mean and the covariance of predictive distribution $p(\mathbf{a}_{k+1}|\mathbf{Y}_{0:k}) = \mathcal{N}(\mu_{k+1|k}, P_{k+1|k})$ are given by [Anderson and Moore, 1979]:

$$\begin{aligned}
\mu_{k+1|k} &= T_k \mu_{k|k-1} + \mathbf{m}_k + K_{gain,k} \mathbf{e}_k \\
P_{k+1|k} &= T_k P_{k|k-1} L_k^T + H_k J_k^T
\end{aligned} \tag{4.38}$$

with

$$L_k = T_k - K_{gain,k} Z_k, \quad J_k = H_k - K_{gain,k} G_k$$

When there is no observation between τ_k and τ_{k+1} we set $\mathbf{e}_k = 0$, $D_k = \infty$ and $K_{gain,k} = 0$.

A backward sampling run, starting with $\mathbf{r}_K = \mathbf{0}, U_K = \mathbf{0}$, can then be conducted from τ_K to τ_1:

$$\begin{aligned} C_k &= F_k(I - G_k^T D_k^{-1} G_k - J_n^T U_k J_k) F_k^T \\ V_k &= F_k(G_k^T D_k^{-1} Z_k + J_k^T U_k L_k) \\ \eta_k &\sim \mathcal{N}\Big(F_k(G_k^T D_k^{-1} \mathbf{e}_k + J_k^T \mathbf{r}_k),\ \sigma^2 C_k\Big) \\ \mathbf{r}_{k-1} &= Z_k^T D_k^{-1} \mathbf{e}_k + L_k^T \mathbf{r}_k - V_k^T C_k^{-1} \epsilon_k \\ U_{k-1} &= Z_k^T D_k^{-1} Z_k + L_k^T U_k L_k + V_k^T C_k^{-1} V_k \end{aligned} \tag{4.39}$$

with $\epsilon_k \sim \mathcal{N}(\mathbf{0}, \sigma^2 C_k)$. If η_0 is needed to be sampled, it can be obtained by setting $G_0 = \mathbf{0}$. As a result, $\eta_{0:K}$ is a joint draw from (4.33) building upon which the sequence of $\mathbf{a}_{1:K}$, with an initial state $\mathbf{a}_0$ provided, can be constructed recursively via

$$\mathbf{a}_{k+1} = T_k \mathbf{a}_k + \mathbf{m}_k + [0, 0, \ldots, \eta_k^T]^T \tag{4.40}$$

Also note that (4.37)-(4.39) can be simplified by assuming uncorrelated process and measurement noises (i.e. $H_k G_k = 0$). More specifically, when $\mathbf{a}_k = \boldsymbol{\theta}_k \in \mathbb{R}^3$ we have $\sigma = \sigma_\theta = \frac{\sigma_T}{m}$, $\mathbf{m}_k = \mathbf{m}_\theta$ and the following variable dimensional matrices

$$H_{\theta,k} = \begin{bmatrix} 0 & & 0 \\ 0 & \cdots & 0 \\ 0 & & 1 \end{bmatrix} \in \mathbb{R}^{3\times d}$$

$$G_{\theta,k} = \begin{bmatrix} G_{1,\theta} & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & G_{1,\theta} & 0 \end{bmatrix} \in \mathbb{R}^{(d-1)\times d}$$

$$G_{1,\theta} = \begin{bmatrix} \frac{\sigma_{\hat{s}}}{\sigma_\theta} & 0 & 0 \\ 0 & \frac{\sigma_{\hat{v}}}{\sigma_\theta} & 0 \\ 0 & 0 & \frac{\sigma_{\hat{a}_T}}{\sigma_\theta} \end{bmatrix}$$

$$F_{\theta,k} = \begin{bmatrix} 0 & 0 & \ldots & 1 \end{bmatrix} \in \mathbb{R}^{1\times d}$$

The dimension of $\mathbf{n}_{\theta,k}$ in this case is $d = 1 + N_k$ with N_k being the dimension of vector $\mathbf{Y}_{\theta,k}$. The disturbance of interest in this case is $\eta_\theta = T_T$. It should be noted that $G_{1,\theta}$, along with $G_{2,\alpha}$ introduced below, are measurement-specific matrices. As for the perpendicular state component, $\mathbf{a}_k = \boldsymbol{\alpha}_k \in \mathbb{R}^3$, $\sigma = \sigma_\alpha = \frac{\sigma_P}{m}$, $\mathbf{m}_k = 0$ and the corresponding matrices are as below

$$
\begin{aligned}
H_{\alpha,k} &= \begin{bmatrix} 0 & & 0 & 0 & 0 \\ 0 & \cdots & 0 & 1 & 0 \\ 0 & & 0 & 0 & \frac{\sigma_b m}{\sigma_P} \end{bmatrix} \in \mathbb{R}^{3\times d} \\
G_{\alpha,k} &= \begin{bmatrix} G_{1,\alpha} & \cdots & 0 & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & G_{1,\alpha} & 0 & 0 \end{bmatrix} \in \mathbb{R}^{(d-2)\times d} \\
G_{1,\alpha} &= \begin{bmatrix} \frac{\sigma_{\hat{\dot{\psi}}}}{\sigma_\alpha} & 0 \\ 0 & \frac{\sigma_{\hat{a}_P}}{\sigma_\alpha} \end{bmatrix} \\
F_{\alpha,k} &= \begin{bmatrix} 0 & \dots & 0 & 1 & 0 \\ 0 & \dots & 0 & 0 & \frac{\sigma_b m}{\sigma_P} \end{bmatrix} \in \mathbb{R}^{2\times d}
\end{aligned}
$$

with $d = 2 + N_k$ and $\eta_\alpha = [T_P, J]^T$ owing to the fact that we have an additional disturbance introduced by the bias term.

We can now run SS from the beginning of each batch, with initial states $\mathbf{a}_0 = \boldsymbol{\theta}_{K_{n-L}}$ and $\mathbf{a}_0 = \boldsymbol{\alpha}_{K_{n-L}}$ respectively, to propose sequences $\boldsymbol{\theta}_{K_{n-L}+1:K_n}$ and $\boldsymbol{\alpha}_{K_{n-L}+1:K_n}$ from the aforementioned proposal kernels. Together with the changepoint timings drawn from the prior and $\mathbf{x}_{0:K_{n-L}}$ from the weighted particle representation of the posterior distribution at time t_{n-L}, the weight update equation for target distribution (4.29) can be shown as

$$
\begin{aligned}
\tilde{\omega}_n^{(i)} &= \frac{p(\mathbf{x}_{0:K_n}|\mathbf{Y}_{0:n})}{q(\mathbf{x}_{0:K_n}|\mathbf{Y}_{0:n})} \\
&\propto \frac{\omega_{n-L}^{(i)}}{\nu_{n-L}^{(i)}} \times p(\mathbf{Y}_{\{\alpha,\theta\},n-L+1:n-L+r}|\mathbf{x}_{K_{n-L}})p(\mathbf{y}_{\mathbf{p},n}|\mathbf{x}_{0:K_n}) \\
&\quad \times p(\mathbf{Y}_{\alpha,n-L+r+1:n}|\{\boldsymbol{\alpha},\tau\}_{K_{n-L}},\{\boldsymbol{\theta},\tau\}_{K_{n-L}+1:K_n}) \\
&\quad \times p(\mathbf{Y}_{\theta,n-L+r+1:n}|\boldsymbol{\theta}_{K_{n-L}},\tau_{K_{n-L}:K_n})
\end{aligned} \tag{4.41}
$$

The first likelihood in (4.41), which consists of inertial and speed observations related to $\mathbf{x}_{K_{n-L}}$ after time t_{n-L}, can be easily computed via (4.18) and (4.20) assumed independence between tangential and perpendicular observations while the position-related likelihood is evaluated by using (4.21). Attributed to the conditionally linear Gaussian models of $\boldsymbol{\theta}$ and $\boldsymbol{\alpha}$, the last two likelihood terms can be approached via Prediction Error Decomposition (PED) [Harvey, 1989] which allows us to sequentially evaluate the likelihood function within the Kalman filtering framework. Specifically, the tangential component of the likelihood function, with the changepoint timings $\tau_{K_{n-L}:K_n}$ omitted in the condition for simplicity, can be computed as

$$p(\mathbf{Y}_{\theta,n-L+r+1:n}|\boldsymbol{\theta}_{K_{n-L}}) = p(\mathbf{Y}_{\theta,K_{n-L}+1}|\boldsymbol{\theta}_{K_{n-L}}) \times \prod_{k=K_{n-L}+2}^{K_n} p(\mathbf{Y}_{\theta,k}|\mathbf{Y}_{\theta,K_{n-L}+1:k-1},\boldsymbol{\theta}_{K_{n-L}}) \tag{4.42}$$

with

$$p(\mathbf{Y}_{\theta,k}|\mathbf{Y}_{\theta,K_{n-l}+1:k-1},\boldsymbol{\theta}_{K_{n-l}}) = \int p(\mathbf{Y}_{\theta,k}|\boldsymbol{\theta}_k) \times p(\boldsymbol{\theta}_k|\mathbf{Y}_{\theta,K_{n-l}+1:k-1},\boldsymbol{\theta}_{K_{n-l}})d\boldsymbol{\theta}_k$$

Note the first component in the integral is given by (4.18) while the second component is the predictive distribution provided by the Kalman filtering step (4.38) in SS. The same procedure can be applied to computing the likelihood for $\boldsymbol{\alpha}$, after which the unnormalised particle weight $\tilde{\omega}_n^{(i)}$ in (4.41) can be obtained accordingly.

4.3.3 State Rejuvenation

In a similar spirit to the resample-move algorithm as in [Gilks and Berzuini, 2001] and Chapter 3, we would also like to mix MCMC moves with variable rate particle filters so as to mitigate the issue of particle impoverishment. The idea is to move particles from their current positions to some new positions which are more diverged after the particle selection stage. In this way, resampled particles whose weights are identical can now have different trajectories and thus improved estimation results may be obtained. Further, instead of moving the whole trajectory as in [Gilks and Berzuini, 2001], for algorithm efficiency here we only move partial (recent) trajectories, i.e. $\mathbf{x}_{K_{n-L+1}:K_n}$, using the following conditional distribution as target distribution:

$$p(\mathbf{x}_{K_{n-L+1}:K_n}|\mathbf{x}_{0:K_{n-L}}\mathbf{Y}_{0:n}) = \frac{p(\mathbf{x}_{0:K_n}|\mathbf{Y}_{0:n})}{p(\mathbf{x}_{0:K_{n-L}}|\mathbf{Y}_{0:n})} \tag{4.43}$$

Due to the intractability, a reversible jump Metropolis-Hastings (MH) kernel is utilised with (4.43) as its invariant distribution. The validity of such a MH setp for variable rate models has been established by [Green, 1995]. Note again that MCMC steps may be blended into a particle filter at any stage legitimately, as per [MacEachern et al., 1999]. If samples $\mathbf{x}^*_{K_{n-L+1}:K_n}$ are drawn according to 1) the prior distribution for changepoint times as per (4.30), 2) the weighted distribution $\hat{p}(\mathbf{x}_{0:K_{n-L}}|\mathbf{Y}_{0:n})$, 3) the section-wise proposal of tangential components (4.31) and 4) the proposal for perpendicular components (4.32), the acceptance ratio of a reversible jump MH kernel for particle i can be shown to be

$$\rho_{RM} = 1 \wedge \left(\frac{p(\mathbf{Y}_{\alpha,n-L+r+1:n} | \{\boldsymbol{\alpha}, \tau\}^{(i)}_{K_{n-L}}, \{\boldsymbol{\theta}, \tau\}^{*}_{K_{n-L}+1:K_n})}{p(\mathbf{Y}_{\alpha,n-L+r+1:n} | \{\boldsymbol{\alpha}, \tau\}^{(i)}_{K_{n-L}}, \{\boldsymbol{\theta}, \tau\}^{(i)}_{K_{n-L}+1:K_n})} \frac{p(\mathbf{y}_{\mathbf{p},n} | \mathbf{x}^{(i)}_{0:K_{n-L}}, \mathbf{x}^{*}_{K_{n-L+1}:K_n})}{p(\mathbf{y}_{\mathbf{p},n} | \mathbf{x}^{(i)}_{0:K_n})} \right. \\ \left. \times \frac{p(\mathbf{Y}_{\theta,n-L+r+1:n} | \{\boldsymbol{\theta}, \tau\}^{(i)}_{K_{n-L}}, \tau^{*}_{K_{n-L}+1:K_n})}{p(\mathbf{Y}_{\theta,n-L+r+1:n} | \{\boldsymbol{\theta}, \tau\}^{(i)}_{K_{n-L}}, \tau^{(i)}_{K_{n-L}+1:K_n})} \right) \quad (4.44)$$

After the selection step N_{RM} reversible jump MCMC moves can be applied to each selected particle for increasing the diversity within the particle collection. Particle weights will remain unchanged after these moves, as noted in Chapter 2.

4.4 Simulations

4.4.1 Manoeuvring Object Trajectories from Intrinsic Model

In our first experiment, the performance of the methods are assessed by 50 random realisations of the problem, each with 500 observations. Full set of sensors $(\hat{s}, \hat{v}, \hat{a}_T, \hat{\dot{\psi}}, \hat{a}_P, \mathbf{y_p})$ are assumed to be available, where the sampling period for inertial, speed and travelled distance measurements is set to 1 second while the arrival time of position measurements is Poisson distributed with intensity $\lambda_{\mathcal{POI}} = 0.1$ (i.e. average number: 50). Moreover, direct Cartesian position observations with additive white noises are considered, i.e. setting $g(\mathbf{p}_n) = \mathbf{p}_n$ in (4.22). All related parameters used in the experiment are summarised in Table 4.1 and an example trajectory is shown in Fig. 4.1.

The tracking performance of the proposed VRPFs, examined in terms of position RMSE, are given in Table 4.2, where the results obtained by running a standard

Table 4.1 VRPF parameters for computer simulations

Symbols	Values
$\mu_T, \sigma_T, \sigma_P, \sigma_b$	$3, 3, 100, 0.5°$
$\tau_{min}, \alpha_\tau, \beta_\tau$	$0, 5, 1$
$\sigma_{\hat{s}}, \sigma_{\hat{v}}, \sigma_{\hat{a}_T}, \sigma_{\hat{a}_P}, \sigma_{\hat{\dot{\psi}}}, \sigma_{p_x}, \sigma_{p_y}$	$3, 0.5, 1, 1, 8°, 5, 5$
$m, \lambda, f_s, \lambda_{\mathcal{POI}}$	$100, 0.3, 1\text{Hz}, 0.1$
N, N_{RM}	$500, 1$

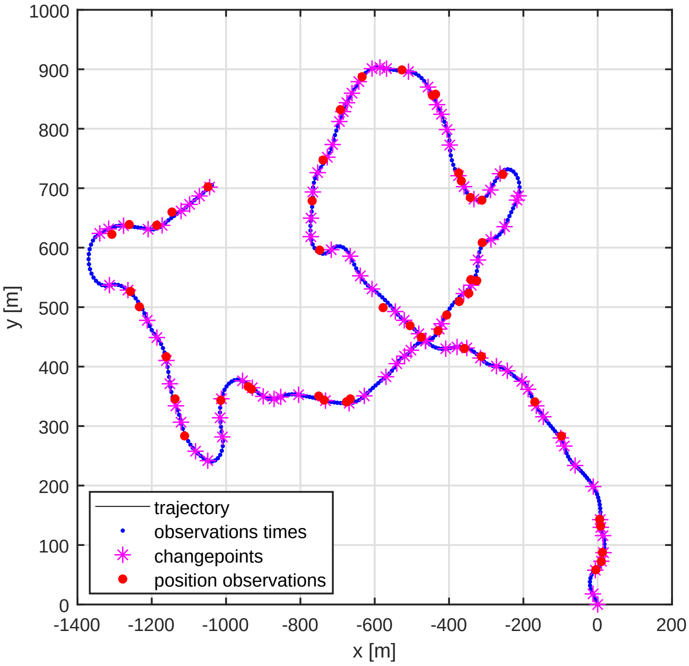


Fig. 4.1 One of the 50 realisations, showing the scale of the tracking problem, the locations of changepoints (magenta stars), the arrival times of occasional position data (red circles) and the arrival times of measurements from other sensors (500 blue dots).

bootstrap VRPF are also included for comparison. Also note that we use the smoothing state trajectory that is estimated from the final filter approximation to compute the position RMSE values. The difference in performance between the VRPF with simulation smoother (VRPF-SS) and the standard bootstrap VRPF (VRPF) confirms that the use of more effective proposals allows the former to draw samples in the light of measurements and hence reduces the mismatch between the predictive distribution and the posterior distribution. The best estimate is given by VRPF-SS with a state rejuvenation procedure (VRPF-SS-RM), from which we can see that the regeneration of both the changepoint timings as well as the kinematic states leads to a significant improvement in tracking performance over the other algorithms. Further enhancement can be achieved by using a larger value for N_{RM}, but apparently a trade-off between computational complexity and performance has to be made.

Table 4.2 Mean and standard deviations of position RMSEs (in meters) across 50 realisations

Algorithms	100 Particles		300 Particles		500 Particles	
	Mean	SD	Mean	SD	Mean	SD
VRPF	294.5	123.3	317.1	160.5	258.4	145.3
VRPF-SS	126.1	71.1	87.6	46.4	71.2	39.6
VRPF-SS-RM	59.2	32.8	45.6	33.0	42.6	33.6

4.4.2 Benchmark Object Trajectory

In the second experiment we adopt one of the benchmark trajectories presented in [Blair et al., 1998] to evaluate the proposed methods. This trajectory has been extensively used for benchmarking purpose in a large number of publications, see [Bunch and Godsill, 2012; Godsill et al., 2007; Lindsten et al., 2016; Masoumi-Ganjgah et al., 2017; Roth et al., 2014; Seah and Hwang, 2011]. Instead of the Cartesian position measurement model, a 2D radar model is used in which the observation vector and the corresponding function are

$$\mathbf{y}_{\mathbf{p},n} = \begin{bmatrix} r_n \\ \xi_n \end{bmatrix}$$

$$g(\mathbf{p}_n) = \begin{bmatrix} \|\mathbf{p}_n - \mathbf{p}_{\text{radar}}\| \\ \arctan\left(\frac{p_{y,n}}{p_{x,n}}\right) \end{bmatrix}$$

with r and ξ being the range and bearing measurements, respectively. $\mathbf{p}_{\text{radar}} = [0, \quad 0]^T$ is the radar location. Again, independent measurements are assumed and hence the measurement noise covariance is $R_{\mathbf{p}} = \text{diag}([\sigma_r^2, \sigma_\xi^2])$. Similar to the first experiment, inertial, speed and distance measurements are simulated based on the models described in Section 4.2.3. The tracking performance of the proposed algorithms are firstly tested on two challenging scenarios: one assuming radar measurements arriving occasionally while the other using the same radar measurement model but with a regular (but slower than that of inertial measurements) sampling rate. Moreover, we vary the Poisson intensity and the sampling rate in order to give a more comprehensive comparison of the algorithms. For all filters, 500 particles are used and their headings are initialised arbitrary (i.e. uniformly in $[0, 2\pi]$). Common parameters used across both test cases are summarised in Table. 4.3. Again, the simple smoothing state estimate as per [Kitagawa, 1996] will be used in the calculation of position RMSE values.

As shown in Table 4.4, while the performance of a standard bootstrap VRPF drops significantly when only a small quantity of radar observations is made (see the results corresponding to $\lambda_{\mathcal{POI}} = 0.1$ and $L = 15$, respectively), VRPF-SS and VRPF-SS-RM are robust to the deterioration of position measurements. This is because these two algorithms not only make a better use of local information (on-board sensors) but also take "future" global information (position) into account. It can be also seen from the table that although the number of radar measurements in the case $\lambda = 0.1$ (average number: $185 \cdot 0.1 = 18.5$) is roughly the same as that of the case $L = 10$, there is a obvious difference in algorithm performance. This can be attributed to the Poisson observation process: sometimes the gap between two radar measurements may be extremely large, thus causing severe degeneration for all methods; at other times (more frequent) the interarrival times of radar measurements can be very close to each other, rendering the proposed method ineffective. In Fig. 4.2, the effect of having a large gap between radar measurement can be seen clearly. In contrast to the standard VRPF which easily fail to stay on the right track, our proposed VRPFs offer greater potential to combat such problems. When the position-related measurements arrive regularly and frequently, the performance of the standard VRPF can be comparable with VRPF-SS; see Fig. 4.3 for an example. In all the tests VRPF-SS-RM gives the best performance in terms of the averaged position RMSE values. Recall that similar performance characteristics of algorithms have been presented in Chapter 3 under a fixed-rate modelling assumption. Indeed, the methods proposed here are essentially the variable rate counterparts to those introduced previously.

Finally, position measurements with decreased accuracy are used to examine the robustness of the proposed algorithms. Whilst the same scenarios with regular and occasional position observations are considered again, the radar measurement model is now replaced by a Cartesian position measurement model in order to illustrate the change of uncertainty in a linear scale. Specifically, in each test case the value of the measurement noise parameter, $\sigma_{pos} = \sigma_{p_x} = \sigma_{p_y}$, is raised from 100 m to 1000 m, corresponding to gradually increasing uncertainty in position measurements. All algorithms were run using 500 particles and the RMSE is computed for 10 independent runs for each σ_{pos}. The box plots given in Figure 4.4 show that the performance across the proposed algorithms and the bootstrap VRPF degrades as the position measurements become less informative when position measurements are received regularly. It is also obvious that VRPF-SS-RM is more robust compared to the others. Note that the results from the bootstrap filter comprise several outliers whose locations are distant from the boxes and hence they are not displayed for better visualisation.

When it comes to the test case with occasional position measurements, a similar conclusion regarding the performance of the proposed methods can be drawn based on Figure 4.5. Also, the observation that the effectiveness of all algorithms drops to some degree is consistent with the results shown in Table 4.4. Overall, the results from the above experiments demonstrate that compared to an off-the-shelf (bootstrap) VRPF the inference algorithms introduced not only offer improved tracking performance but also show great robustness in certain challenging tracking problems.

Table 4.3 VRPF parameters for the benchmark trajectory data

Symbols	Values
$\mu_T, \sigma_T, \sigma_P, \sigma_b$	$1800, 1800, 3000, 0.5°$
$\tau_{min}, \alpha_\tau, \beta_\tau$	$0, 1, 0.4$
$\sigma_{\hat{s}}, \sigma_{\hat{v}}, \sigma_{\hat{a}_T}, \sigma_{\hat{a}_P}, \sigma_{\hat{\psi}}, \sigma_r, \sigma_\xi$	$100, 100, 15, 15, 20°, 150, 0.05°$
m, λ, f_s	$100, 3, 1\text{Hz}$
$N, N_{particle}, N_{RM}$	$185, 500, 1$

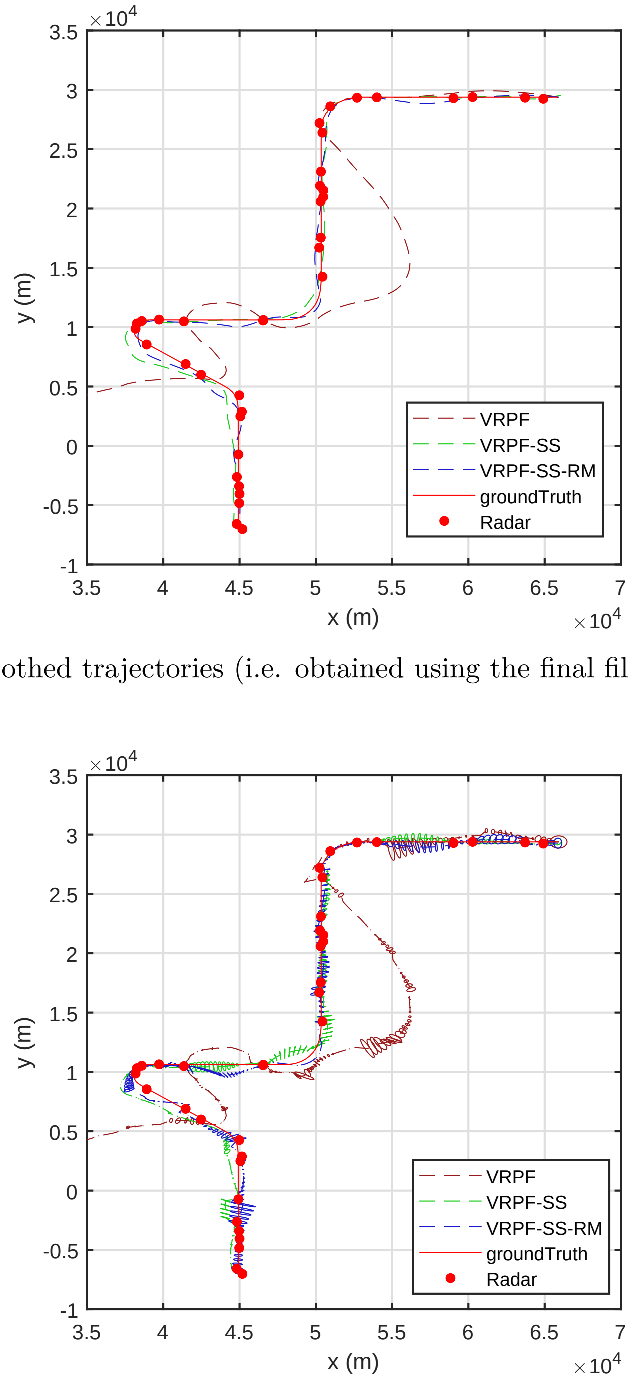
Table 4.4 Tracking performance on the benchmark trajectory. Numbers given are means (and standard deviations) of position RMSE (km), each of which is computed for 10 independent runs.

Algorithms	Poisson Intensity ($\lambda_{\mathcal{POI}}$)		
	0.1	0.2	0.3
VRPF	5.95 (8.18)	3.24 (2.38)	1.82 (1.73)
VRPF-SS	1.73 (0.55)	1.89 (1.14)	2.92 (2.31)
VRPF-SS-RM	1.34 (0.74)	0.61 (0.23)	0.60 (0.25)

(a) With occasional radar measurements.

Algorithms	Regular Lag (L)		
	15	10	5
VRPF	11.20 (9.33)	3.49 (4.24)	1.04 (0.81)
VRPF-SS	2.03 (1.18)	0.94 (0.54)	0.91 (0.46)
VRPF-SS-RM	1.23 (0.48)	0.84 (0.29)	0.37 (0.12)

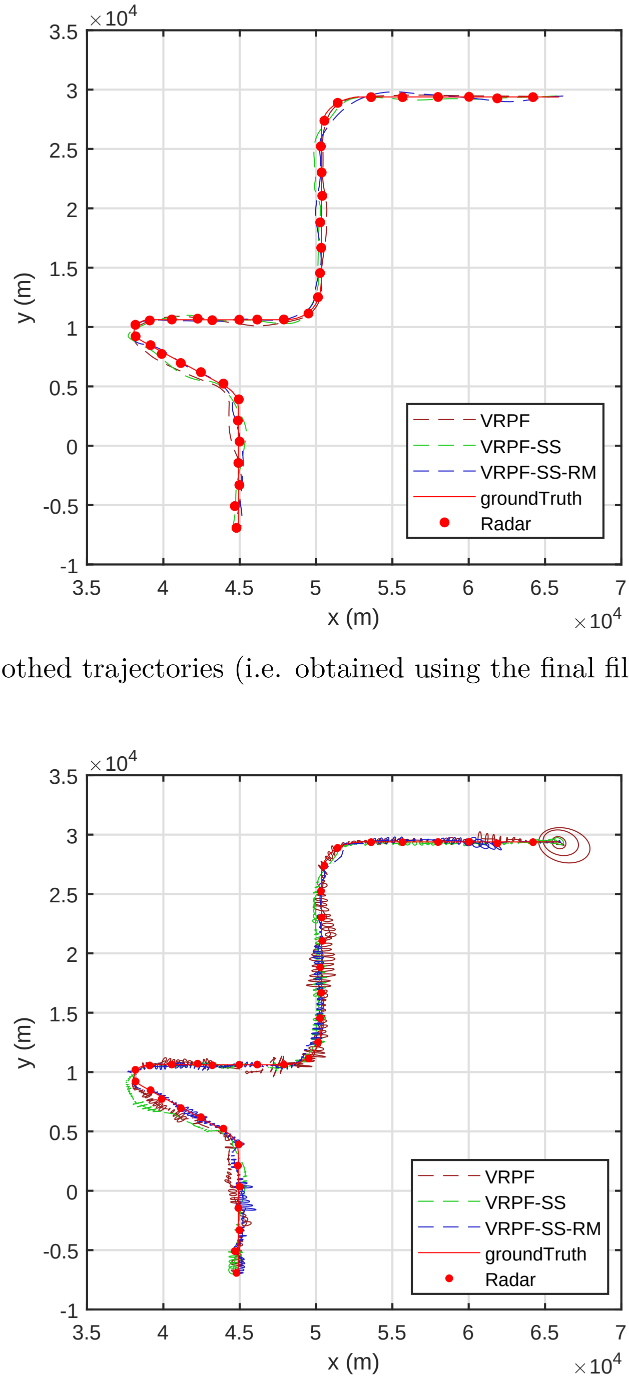
(b) With a fixed time lag (L) between consecutive radar measurements.



(a) Smoothed trajectories (i.e. obtained using the final filter approximation)

(b) filtered trajectories with 95% confidence ellipsoids

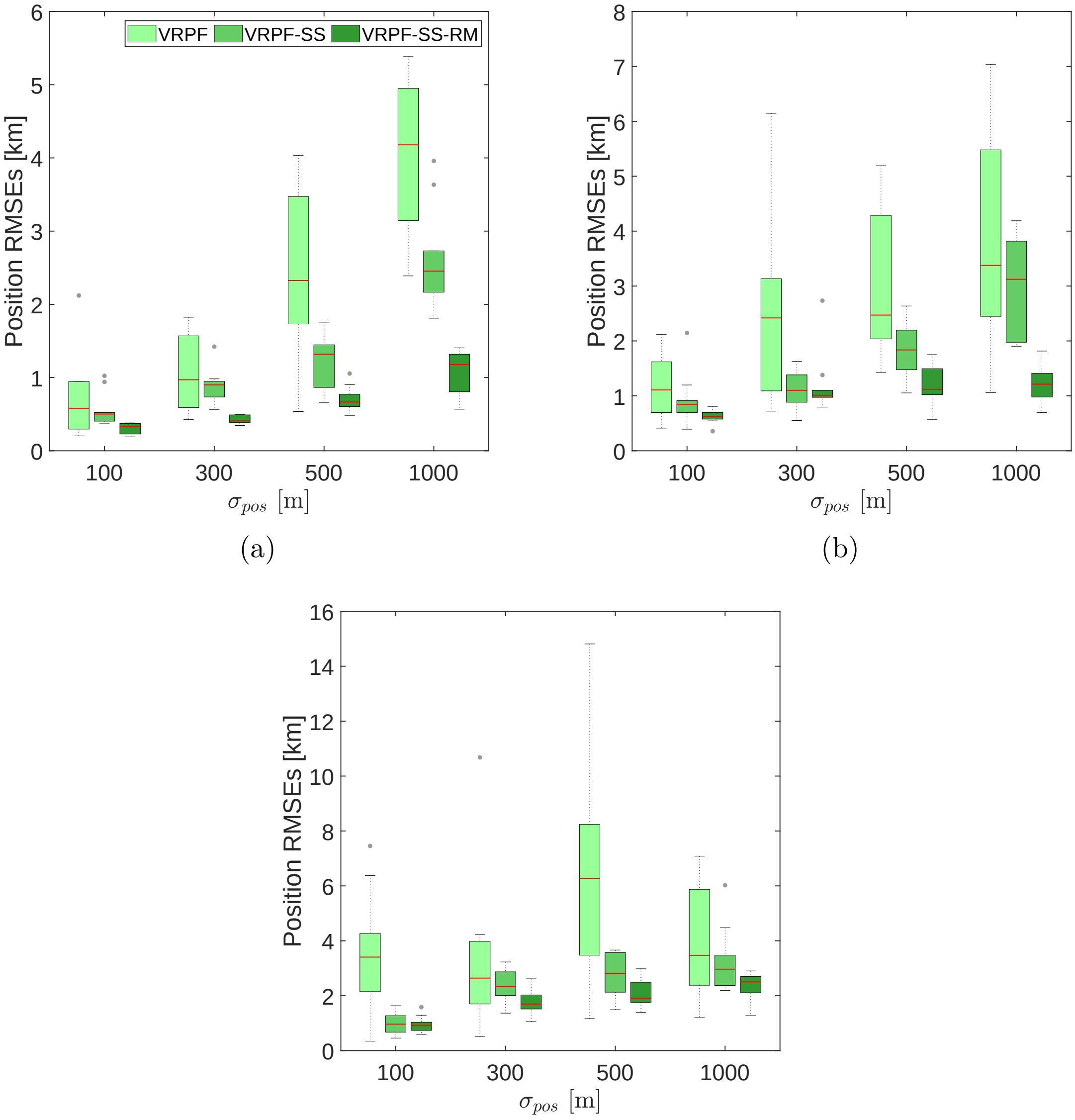
Fig. 4.2 Tracking with occasional radar measurements ($\lambda_{\mathcal{POI}} = 0.2$).



(a) Smoothed trajectories (i.e. obtained using the final filter approximation)

(b) Filtered trajectories with 95% confidence ellipsoids

Fig. 4.3 An example of tracking results with occasional radar measurements ($L = 5$).

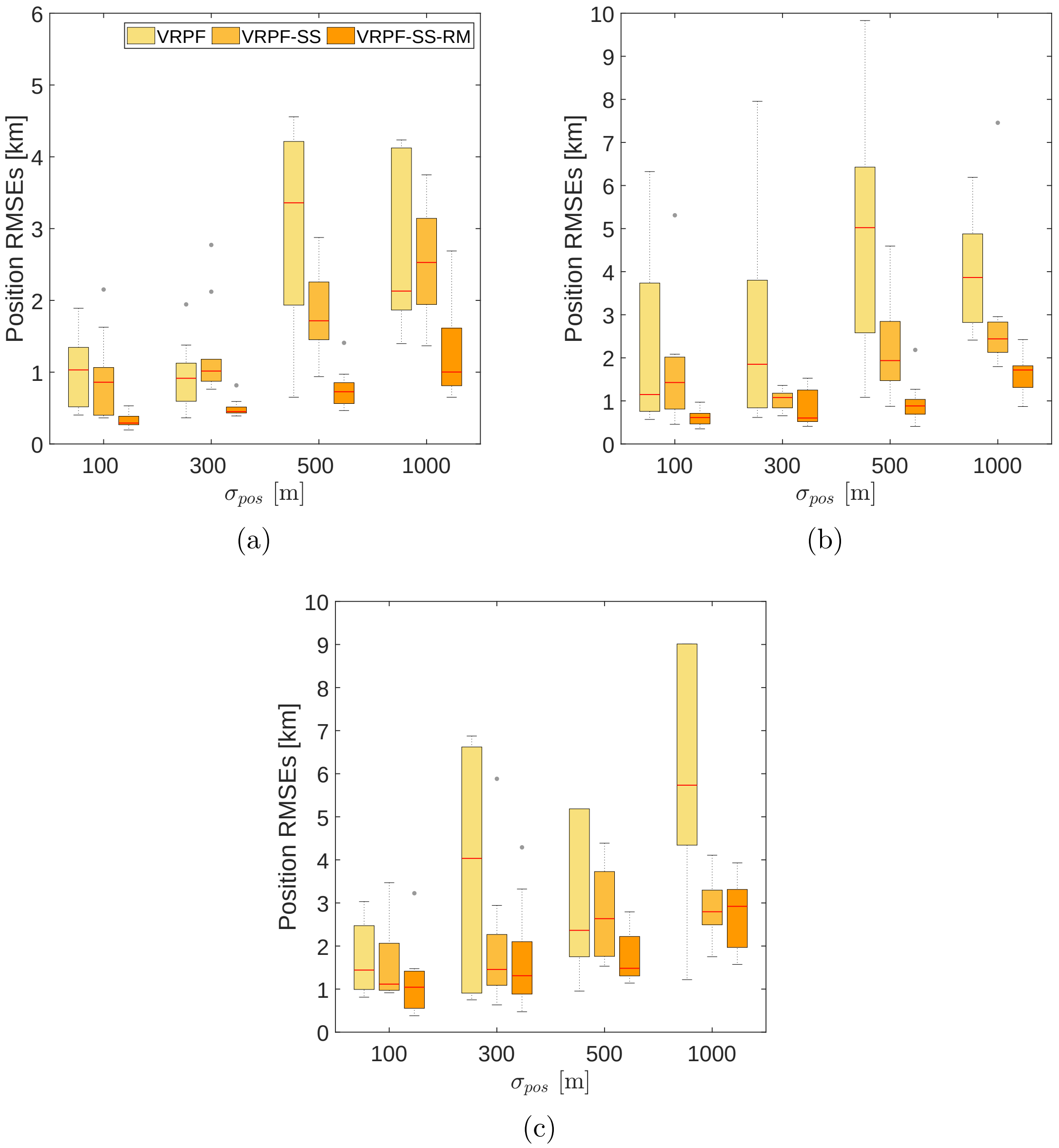


(a)

(b)

(c)

Fig. 4.4 Tracking results with regularly spaced Cartesian position measurements and varied noise parameter $\sigma_{pos} = \{100, 300, 500, 1000\}$. Fixed time lags used: (a) $L = 5$ (b) $L = 10$ (c) $L = 15$. Outliers (grey dots), medians (red horizontal lines).



(a)

(b)

(c)

Fig. 4.5 Tracking results with occasional Cartesian position measurements and varied noise parameter $\sigma_{pos} = \{100, 300, 500, 1000\}$. (a) $\lambda_{\mathcal{POI}} = 0.3$ (b) $\lambda_{\mathcal{POI}} = 0.2$ (c) $\lambda_{\mathcal{POI}} = 0.1$. Outliers (grey dots), medians (red horizontal lines).

4.5 Particle Gibbs Samplers for Full System Inference

In addition to state trajectories, we seek to design an effective sampling scheme for the inference of the system parameters. Denote $\boldsymbol{\beta}_I = [\mu_T, \sigma_T, \sigma_P, \sigma_b]^T$ and $\boldsymbol{\beta}_M = [\sigma_{\hat{s}}, \sigma_{\hat{v}}, \sigma_{\hat{a}_T}, \sigma_{\hat{a}_P}, \sigma_{\hat{\psi}}, \sigma_{p_x}, \sigma_{p_y}]^T$ as parameter vectors for the intrinsic model and measurement models, respectively. The objective is to simulate hidden states and constant parameters from the full joint posterior distribution $p(\mathbf{x}_{0:K_N}, \boldsymbol{\beta}_I, \boldsymbol{\beta}_M | \mathbf{Y}_{0:N})$ which can be decomposed as

$$p(\mathbf{x}_{0:K_N}, \boldsymbol{\beta}_I, \boldsymbol{\beta}_M | \mathbf{Y}_{0:N}) = p(\mathbf{x}_{0:K_N} | \boldsymbol{\beta}_I, \boldsymbol{\beta}_M, \mathbf{Y}_{0:N}) p(\boldsymbol{\beta}_I, \boldsymbol{\beta}_M | \mathbf{Y}_{0:N}) \tag{4.45}$$

Intuitively, one may consider drawing samples from this distribution using a Metropolis-Hastings (MH) sampler with its invariant distribution set to be (4.45). When a proposal distribution of the form

$$q(\mathbf{x}^*_{0:K_N}, \boldsymbol{\beta}^*_I, \boldsymbol{\beta}^*_M) = p(\mathbf{x}^*_{0:K_N} | \boldsymbol{\beta}^*_I, \boldsymbol{\beta}^*_M, \mathbf{Y}_{0:N}) q(\boldsymbol{\beta}^*_I, \boldsymbol{\beta}^*_M | \boldsymbol{\beta}^{(m-1)}_I, \boldsymbol{\beta}^{(m-1)}_M) \tag{4.46}$$

is used, the acceptance ratio of the MH algorithm can be shown as follows

$$\rho_{MH} = 1 \wedge \frac{p(\mathbf{Y}_{0:N} | \boldsymbol{\beta}^*_I, \boldsymbol{\beta}^*_M)}{p(\mathbf{Y}_{0:N} | \boldsymbol{\beta}^{(m-1)}_I, \boldsymbol{\beta}^{(m-1)}_M)} \frac{p(\boldsymbol{\beta}^*_I, \boldsymbol{\beta}^*_M)}{p(\boldsymbol{\beta}^{(m-1)}_I, \boldsymbol{\beta}^{(m-1)}_M)} \frac{q(\boldsymbol{\beta}^*_I, \boldsymbol{\beta}^*_M | \boldsymbol{\beta}^{(m-1)}_I, \boldsymbol{\beta}^{(m-1)}_M)}{q(\boldsymbol{\beta}^{(m-1)}_I, \boldsymbol{\beta}^{(m-1)}_M | \boldsymbol{\beta}^*_I, \boldsymbol{\beta}^*_M)}$$

with m being the iteration index. Note that to obtain the ratio we have utilised the fact that $p(\mathbf{x}_{0:K_N}, \boldsymbol{\beta}_I, \boldsymbol{\beta}_M | \mathbf{Y}_{0:N}) \propto p(\mathbf{Y}_{0:N} | \boldsymbol{\beta}_I, \boldsymbol{\beta}_M) p(\boldsymbol{\beta}_I, \boldsymbol{\beta}_M) p(\mathbf{x}_{0:K_N} | \boldsymbol{\beta}_I, \boldsymbol{\beta}_M, \mathbf{Y}_{0:N})$. Despite its elegance, it is impossible to compute the above acceptance ratio in our case since the likelihood function $p(\mathbf{Y}_{0:N} | \boldsymbol{\beta}_I, \boldsymbol{\beta}_M)$ is not available in closed form. Nevertheless, it has been shown by [Andrieu et al., 2010] that an exact MH sampler, termed particle marginal Metropolis-Hastings (PMMH) sampler (also known as pseudo-marginal sampler as in [Andrieu and Roberts, 2009]), can still be constructed provided a non-negative and unbiased estimate of the likelihood term, denoted as $\hat{p}(\mathbf{Y}_{0:N} | \boldsymbol{\beta}_I, \boldsymbol{\beta}_M)$. The key to the construction of such an sampler lies in working with a target distribution which is defined on an extended state space including all the random variables associated with a generic particle filter. It can be shown that the acceptance ratio for a standard PMMH update is given by (we refer readers to Section 4 and Appendix B of [Andrieu et al., 2010] for a very detailed derivation):

$$\rho_{PMMH} = 1 \wedge \frac{\hat{p}(\mathbf{Y}_{0:N}|\boldsymbol{\beta}_I^*, \boldsymbol{\beta}_M^*)}{\hat{p}(\mathbf{Y}_{0:N}|\boldsymbol{\beta}_I^{(m-1)}, \boldsymbol{\beta}_M^{(m-1)})} \frac{p(\boldsymbol{\beta}_I^*, \boldsymbol{\beta}_M^*)}{p(\boldsymbol{\beta}_I^{(m-1)}, \boldsymbol{\beta}_M^{(m-1)})} \frac{q(\boldsymbol{\beta}_I^*, \boldsymbol{\beta}_M^*|\boldsymbol{\beta}_I^{(m-1)}, \boldsymbol{\beta}_M^{(m-1)})}{q(\boldsymbol{\beta}_I^{(m-1)}, \boldsymbol{\beta}_M^{(m-1)}|\boldsymbol{\beta}_I^*, \boldsymbol{\beta}_M^*)}$$

where the likelihood term $\hat{p}(\mathbf{Y}_{0:N}|\boldsymbol{\beta}_I, \boldsymbol{\beta}_M)$ can be obtained as a side product of particle filter. Furthermore, as the intrinsic model is driven by Gaussian disturbances parametrized by $\boldsymbol{\beta}_I$ (see (4.11) and (4.14)), it will be straightforward to sample from $p(\boldsymbol{\beta}_I|\mathbf{x}_{0:K_N}, \mathbf{Y}_{0:N}, \boldsymbol{\beta}_M)$ using appropriately chosen conjugate priors. A similar principle can be adopted for proposing $\boldsymbol{\beta}_M$ from $p(\boldsymbol{\beta}_M|\mathbf{x}_{0:K_N}, \mathbf{Y}_{0:N}, \boldsymbol{\beta}_I)$, assuming the observation process is also disturbed by additive white Gaussian noises. If we replace the parameter proposal $q(\boldsymbol{\beta}_I^*, \boldsymbol{\beta}_M^*|\boldsymbol{\beta}_I^{(m-1)}, \boldsymbol{\beta}_M^{(m-1)})$ with the full conditionals of parameters, the PMMH acceptance ratio becomes

$$\rho'_{PMMH}$$
$$= 1 \wedge \frac{\hat{p}(\mathbf{Y}_{0:N}|\boldsymbol{\beta}_I^*, \boldsymbol{\beta}_M^*)}{\hat{p}(\mathbf{Y}_{0:N}|\boldsymbol{\beta}_I^{(m-1)}, \boldsymbol{\beta}_M^{(m-1)})} \frac{p(\mathbf{Y}_{0:N}, \mathbf{x}_{0:K_N}(m-1)|\boldsymbol{\beta}_M^{(m-1)})}{p(\mathbf{Y}_{0:N}, \mathbf{x}_{0:K_N}(m-1)|\boldsymbol{\beta}_M^*)} \frac{p(\mathbf{x}_{0:K_N}(m-1)|\boldsymbol{\beta}_M^{(m-1)})}{p(\mathbf{x}_{0:K_N}(m-1)|\boldsymbol{\beta}_M^*)}$$

However, the fact that ρ'_{PMMH} depends on the state trajectory $\mathbf{x}_{0:K_N}(m-1)$ obtained from the last iteration implies that the resulting chain can be extremely sticky in practice. In fact, our preliminary results also showed that the PMMH samplers described above gave very poor performance (i.e. the resulting chain retained the same value for very long periods).

While PMMH samplers cannot work efficiently here, we resort to the particle Gibbs (PG) sampler since sampling exactly from $p(\boldsymbol{\beta}_I, \boldsymbol{\beta}_M|\mathbf{x}_{0:K_N}, \mathbf{Y}_{0:N})$ is possible in our scenario. The PG sampler, as in [Andrieu et al., 2010], provides a valid particle approximation to a Gibbs sampler targeting $p(\mathbf{x}_{0:K_N}, \boldsymbol{\beta}_I, \boldsymbol{\beta}_M|\mathbf{Y}_{0:N})$. The key element of a PG sampler is the so-called conditional sequential Monte Carlo (SMC) update which maintain a pre-specified state trajectory $\mathbf{x}_{0:K_N}$ throughout the standard particle filtering procedure. In order to present a conditional SMC update, it is necessary to introduce $B_{0:N} = \{B_0, B_1, \ldots, B_N\}$ which specifies the indices of the ancestor particles of $\mathbf{x}_{0:K_N}$ at each time instant, i.e. $\{\mathbf{x}_0^{(B_0)}, \mathbf{x}_{K_0+1:K_1}^{(B_1)}, \ldots, \mathbf{x}_{K_{N-1}:K_N}^{(B_N)}\}$. With the introduction of $B_{0:N}$, a standard conditional SMC algorithm tailored to the variable rate setting is summarised in algorithm 6 (see Andrieu's paper for a fixed rate equivalent of this algorithm). Note also that to establish a valid PG sampler with $p(\mathbf{x}_{0:K_N}, \boldsymbol{\beta}_I, \boldsymbol{\beta}_M|\mathbf{Y}_{0:N})$ as its invariant distribution the biased selection/resampling scheme used in the previous section will be replaced by a multinomial resampling scheme in the conditional SMC

```
// Initialisation
Specify a state trajectory $\mathbf{x}_{0:K_N} = \{\mathbf{x}_0^{(B_0)}, \mathbf{x}_{K_0+1:K_1}^{(B_1)}, \ldots, \mathbf{x}_{K_{N-1}:K_N}^{(B_N)}\}$ ;
$\forall i, i \neq B_0$, sample $\mathbf{x}_0^{(i)} \sim q_0(\mathbf{x}_0|\mathbf{y}_0)$ ;
Compute particle weights $\omega_0^{(i)}$ according to (4.27) ;
for $n = 1, 2, \ldots, N$ do
    // Multinomial Resampling
    $\forall i, i \neq B_{n-1}$, select indexes $j^{(i)}$ with $\Pr(j^{(i)} = k) = \omega_{n-1}^{(k)}$;
    Reset $\omega_{n-1}^{(i)} = 1/N_p$ ;
    // Propagation
    $\forall i, i \neq B_{n-1}$, sample $\mathbf{x}_{K_{n-1}+1:K_n}^{(i)} \sim q(\mathbf{x}_{K_{n-1}+1:K_n}|\mathbf{x}_{0:K_{n-1}}, \mathbf{y}_{0:n})$ ;
    Update state trajectories: $\mathbf{x}_{0:N_n}^{(i)} = \{\mathbf{x}_{0:N_{n-1}}^{(i)}, \mathbf{x}_{N_{n-1}+1:N_n}^{(i)}\}$ ;
    // Weighting
    Compute particle weights $\omega_n^{(i)}$ according to (4.27) ;
end
Output $\hat{p}(\mathbf{x}_{0:K_N}|\mathbf{y}_{0:N}) \approx \sum_{i=1}^{N_p} \omega_N^{(i)} \delta_{\mathbf{x}_{0:K_N}^{(i)}}(d\mathbf{x}_{0:K_N})$ ;
```

Algorithm 6: Conditional SMC Update (Variable Rate)

update (though more sophisticated procedures such as the systematic and residual resampling are also qualified [Chopin and Singh, 2015]).

The conditional variable rate SMC update targets $p(\mathbf{x}_{0:K_N}|\boldsymbol{\beta}_I(m-1), \boldsymbol{\beta}_M(m-1), \mathbf{Y}_{0:N})$ conditional on $\{\mathbf{x}_{0:K_N}(m-1),\ B_{0:N}(m-1)\}$ drawn at the $(m-1)$-th PG iteration. For completing the PG sampling algorithm, we also need to be able to sample from $p(\boldsymbol{\beta}_I, \boldsymbol{\beta}_M|\mathbf{x}_{0:K_N}\mathbf{Y}_{0:N}\}$. Provided the state trajectory $\mathbf{x}_{0:K_N}$ and the measurements $\mathbf{Y}_{0:n}$, now we show how to obtain the conditional distributions for parameters of the intrinsic model. First, from the tangential state transition density (4.11) we have

$$a_{T,k} = -\frac{\lambda}{m} v_{k-1} + \left(e^{\frac{-\Delta\tau\lambda}{m}} - 1\right) a_{T,k-1} + \epsilon_{T,k}, \quad \epsilon_{T,k} \sim \mathcal{N}\left(\frac{\mu_T}{m}, \frac{\sigma_T^2}{m^2}\right)$$

which leads to the following likelihood function:

$$p(\mathbf{z}_T|\mu_T, \gamma_T) = \prod_{k=1}^{K_N} p(z_{T,k}|\mu_T, \gamma_T) = \prod_{k=1}^{K_N} \mathcal{N}\left(z_{T,k}\Big|\frac{\mu_T}{m}, \frac{\sigma_T^2}{m^2}\right) \tag{4.47}$$

where

$$\begin{aligned} z_{T,k} &= a_{T,k} + \frac{\lambda}{m} v_{k-1} - \left(e^{\frac{-\Delta\tau\lambda}{m}} - 1\right) a_{T,k-1} \\ \gamma_T &= \frac{1}{\sigma_T^2} \end{aligned}$$

and K_N is the number of changepoints arrving before t_n, as defined in Section 4.2.1. Note again that here we have formalised the distribution in terms of the precision γ_T. By choosing a conjugate normal-gamma prior, defined as

$$\begin{aligned} p(\mu_T, \gamma_T) &= \mathcal{NG}(\mu_T, \gamma_T | \mu_{T0}, \kappa_{T0}, \alpha_{T0}, \beta_{T0}) \\ &= \mathcal{N}(\mu_T | \mu_{T0}, (\kappa_{T0}\gamma_T)^{-1}) \mathcal{G}(\gamma_T | \alpha_{T0}, \beta_{T0}) \end{aligned} \tag{4.48}$$

the full conditional for the mean and the precision can be shown to be a normal-gamma distribution of the form

$$\begin{aligned} p(\mu_T, \gamma_T | \mathbf{x}_{0:K_n}, \mathbf{Y}_{0:n}, \boldsymbol{\beta}_M, \boldsymbol{\beta}_{I\setminus T}) &= p(\mu_T, \gamma_T | \mathbf{z}_T) \qquad (4.49) \\ &\propto p(\mathbf{z}_T | \mu_T, \gamma_T) p(\mu_T, \gamma_T) \\ &\propto \mathcal{N}\Big(\mu_T | \mu_{K_N}, \big((\kappa_{T0} + K_N)\gamma_T\big)^{-1}\Big) \mathcal{G}\Big(\gamma_T | \alpha_{K_N}, \beta_{K_N}\Big) \end{aligned}$$

where

$$\begin{aligned} \mu_{K_N} &= \frac{\kappa_{T0}\mu_{T0} + K_N \varpi}{\kappa_{T0} + K_N} \\ \alpha_{K_N} &= \alpha_{T0} + \frac{K_N}{2} \\ \beta_{K_N} &= \beta_{T0} + \frac{1}{2} \sum_{k=1}^{K_N} (m z_{T,k} - \varpi) + \frac{\kappa_{T0} K_N (\varpi - \mu_{T0})^2}{2(\kappa_{T0} + K_N)} \\ \varpi &= \frac{m}{K_N} \sum_{k=1}^{K_N} z_{T,k} \end{aligned}$$

and $\boldsymbol{\beta}_{I\setminus T}$ denotes the set of intrinsic model parameters excluding μ_T and γ_T. Conveniently, sampling from this joint distribution can be done using the method of composition. With regard to the parameter related to the perpendicular force, i.e. σ_P, we note from (4.14) that

$$a_{P,k} = \epsilon_{P,k}, \quad \epsilon_{P,k} \sim \mathcal{N}\Big(0, (m^2\gamma_P)^{-1}\Big) \tag{4.50}$$

with $\gamma_P = 1/\sigma_P^2$ being the precision. As this is a Gaussian distribution with unknown precision, a Gamma prior $p(\gamma_P) = \mathcal{G}(\gamma_P|\alpha_{P0}, \beta_{P0})$ is adopted. Consequently, the full conditional distribution of the precision is given by

$$\begin{aligned} p(\gamma_P|\mathbf{x}_{0:K_N}, \mathbf{Y}_{0:n}, \boldsymbol{\beta}_M, \boldsymbol{\beta}_{I\setminus P}) &= p(\gamma_P|\mathbf{z}_P) \\ &\propto p(\gamma_P) \prod_{k=0}^{K_N} p(z_{P,k}|\gamma_P) \\ &= \mathcal{G}\Big(\gamma_P|\alpha_{P0} + \frac{K_N+1}{2}, \beta_{P0} + \frac{m^2}{2}\sum_{k=0}^{K_N} z_{P,k}\Big) \end{aligned} \tag{4.51}$$

with $z_{P,k} = a_{P,k}$. Similarly, the system model for the bias term can be also extracted from (4.14):

$$b_k = b_{k-1} + \epsilon_{b,k}, \quad \epsilon_{b,k} \sim \mathcal{N}(0, \gamma_b^{-1}) \tag{4.52}$$

Equipped with a gamma prior $p(\gamma_b) = \mathcal{G}(\alpha_{b0}, \beta_{b0})$, the conditional posterior for γ_b is then obtained as

$$\begin{aligned} p(\gamma_b|\mathbf{x}_{0:K_N}, \mathbf{Y}_{0:n}, \boldsymbol{\beta}_M, \boldsymbol{\beta}_{I\setminus b}) &= p(\gamma_b|\mathbf{z}_b) \\ &\propto p(\gamma_b) \prod_{k=1}^{K_N} p(z_{b,k}|\gamma_b) \\ &= \mathcal{G}\Big(\gamma_P|\alpha_{b0} + \frac{K_N}{2}, \beta_{b0} + \frac{1}{2}\sum_{k=0}^{K_N} z_{b,k}\Big) \end{aligned} \tag{4.53}$$

with $z_{b,k} = b_k - b_{k-1}$.

As for the update of measurement model parameters $\boldsymbol{\beta}_M$, the assumption of additive white Gaussian noises and independence among different sensor systems has also led to straightforward constructions of corresponding full conditionals. More specifically, recall that in a general case, where we observe all possible data at some t_n (i.e. $\mathbf{Y}_n = \{\hat{s}_n, \hat{v}_n, \hat{a}_{T,n}, \hat{a}_{P,n}, \hat{\psi}_n, \mathbf{y}_{\mathbf{p},n}\}$) and the measurement models are those defined in Section 4.2.3, the joint observation density is given by

$$\begin{aligned} p(\mathbf{Y}_n|\breve{\mathbf{u}}_n) &= p(\hat{s}_n|s_n)p(\hat{v}_n|v_n)p(\hat{a}_{T,n}|a_{T,n})p(\hat{a}_{P,n}|a_{P,n})p(\hat{\psi}_n|a_{P,n}, v_n, b_n)p(\mathbf{y}_{\mathbf{p},n}|\mathbf{p}_n) \\ &= \mathcal{N}(\hat{s}_n|s_n, \gamma_{\hat{s}}^{-1})\mathcal{N}(\hat{v}_n|v_n, \gamma_{\hat{v}}^{-1})\mathcal{N}(\hat{a}_{T,n}|a_{T,n}, \gamma_{\hat{a}_T}^{-1})\mathcal{N}(\hat{a}_{P,n}|a_{P,n}, \gamma_{\hat{a}_P}^{-1}) \\ &\quad \times \mathcal{N}(\hat{\psi}_n|\frac{a_{P,n}}{v_n} + b_n, \gamma_{\hat{\psi}}^{-1})\mathcal{N}(\hat{p}_{x,n}|g(p_{x,n}), \gamma_{p_x}^{-1})\mathcal{N}(\hat{p}_{y,n}|g(p_{y,n}), \gamma_{p_y}^{-1}) \end{aligned}$$

with $\breve{\mathbf{u}}_n = h(\mathbf{x}_{K_n:K_{n+1}}) = \{s_n, v_n, a_{T,n}, \psi_n, a_{P,n}, b_n, \mathbf{p}_n\}$ depicting the state values deterministically interpolated at the time and $g(\cdot)$ being some known mapping function.

Here variances are again converted into precisions, i.e. $\gamma = \sigma^{-2}$, in order to use Gamma distributions as conjugate priors. Also note that in cases where there are correlations between measurements, a Wishart distribution may be placed over the precision matrix. Since the observation densities are all Gaussian, a general form of the measurement model of a sensor, denoted as $p(y_n|x_n, \gamma) = \mathcal{N}(y_n|x_n, \gamma^{-1})$, is considered here. As a result, the full conditional distribution of γ is given by

$$p(\gamma|x_{0:N}, y_{0:N}) \propto p(\gamma) \times \prod_{n=1}^{N} p(y_n|x_n, \gamma)$$

which is again a gamma distribution when a gamma prior $p(\gamma) = \mathcal{G}(\alpha_0, \beta_0)$ is considered:

$$\begin{aligned} p(\gamma|x_{0:N}, y_{0:N}) &= \mathcal{G}\Big(\gamma\Big|\ \alpha_0 + \frac{n+1}{2},\ \ \beta_0 + \frac{1}{2}\sum_{n=0}^{N}\big(y_n - g(x_n)\big)^2\Big) \\ &= p(\gamma|\mathbf{x}_{0:K_N}, \mathbf{Y}_{0:N}, \boldsymbol{\beta}_I, \boldsymbol{\beta}_{M\setminus\gamma}) \end{aligned} \tag{4.54}$$

While being akin to the procedure of deriving the full conditionals for γ_P and γ_b, the change of the upper limit (from the number of changepoints K_N to that of observation timestamps N) in the summation should be noted.

Having obtained all the necessary ingredients, Algorithm 7 summarises the PG sampler which is suitable for use with the considered conditionally deterministic variable rate model.

```
// Initialisation
Set $\boldsymbol{\beta}_I(0)$ and $\boldsymbol{\beta}_M(0)$ arbitrarily ;
Run a standard VRPF from $t_0$ to $t_N$:
Sample $\mathbf{x}_{0:K_N}(0) \sim \hat{p}(\mathbf{x}_{0:K_N}|\boldsymbol{\beta}_I(0), \boldsymbol{\beta}_M(0), \mathbf{y}_{0:N})$ ;
// Particle Gibbs Sweeps
for $m = 2, \ldots, N_{iter}$ do
    Sample $\boldsymbol{\beta}_I(m)$ according to (4.49), (4.51) and (4.53) ;
    Sample $\boldsymbol{\beta}_M(m)$ according to (4.54) ;
    // Conditional SMC update
    Run Algo. 6 (conditional on $\{\mathbf{x}_{0:K_N}(m-1)\}$) from $t_0$ to $t_N$:
    Sample $\mathbf{x}_{0:K_N}(m) \sim \hat{p}(\mathbf{x}_{0:K_N}|\boldsymbol{\beta}_I(m), \boldsymbol{\beta}_M(m), \mathbf{y}_{0:N})$ ;
    Keep every MCMC output after a specified burn-in period $N_{burn}$ ;
end
```

Algorithm 7: Particle Gibbs (PG) Sampler

Table 4.5 Experiment setup for the full system inference

Hyperparameters	Values
m, λ, f_s	$100, 0.3, 1\text{Hz}$
$\tau_{min}, \alpha_\tau, \beta_\tau$	$0, 3, 1$
$N, N_{particle}, N_{iter}, N_{burn}$	$250, 2000, 150, 15$
Parameters to be learnt	True Values
$\boldsymbol{\beta}_I : \mu_T, \sigma_T, \sigma_P, \sigma_b$	$5, 15, 150, 0.2618$ rad
$\boldsymbol{\beta}_M : \sigma_{\hat{s}}, \sigma_{\hat{v}}, \sigma_{\hat{a}_T}, \sigma_{\hat{a}_P}, \sigma_{\hat{\psi}}, \sigma_{p_x}, \sigma_{p_y}$	$10, 0.5, 5, 5, 0.2618$ rad$, 5, 5$

(a) System and filter configuration

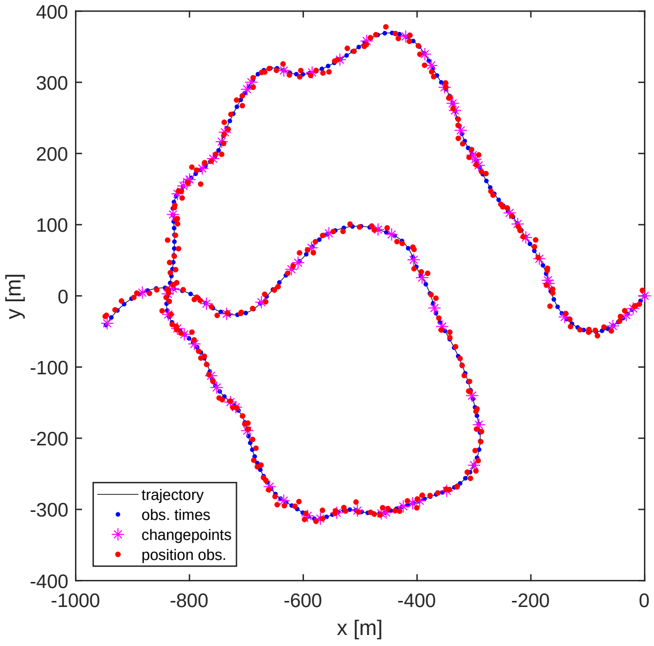
Parameters	Conjugate piror	Hyperparameters
μ_T, γ_T	Normal-gamma $\mathcal{NG}(\cdot)$	$\mu_{T0} = 0, \kappa_{T0} = 0.01, \alpha_{T0} = 1, \beta_{T0} = 0.5$
γ_P / γ_b	Gamma $\mathcal{G}(\cdot)$	$\alpha_{P0} = 1, \beta_{P0} = 5 \times 10^4$ / $\alpha_{b0} = 1, \beta_{b0} = 0.5$
γ in (4.54)	Gamma $\mathcal{G}(\cdot)$	$\alpha_0 = 1, \beta_0 = 0.5$

(b) Prior hyperparameters, with precisions $\gamma = \sigma^{-2}$.

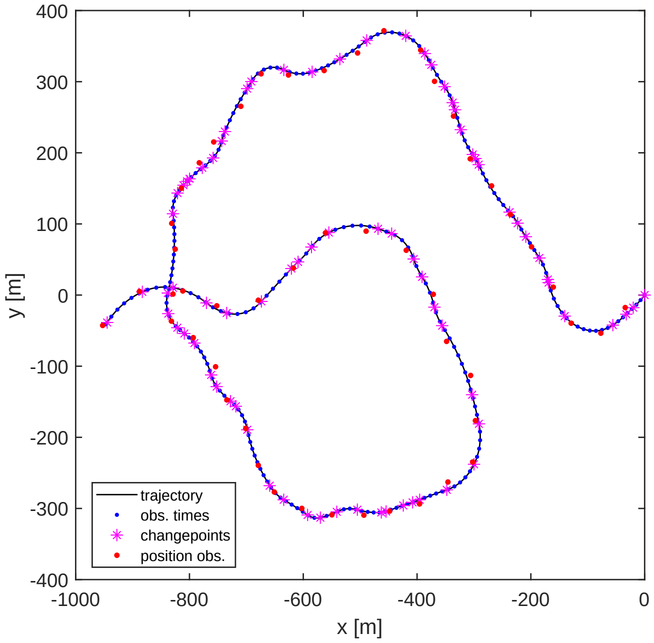
4.5.1 Numerical Results

The proposed sampler is now tested on a trajectory simulated from the variable rate intrinsic model. Two test cases are considered: one having regular (frequent and synchronised) inertial and position measurements and the other a reduced number of position measurements (obtained by downsampling the position measurements in the first case with a fixed lag of 5). Figure 4.6 shows the example trajectory under these two different experimental setups. Table 4.5a gives details about the true system parameters as well as the configuration of the sampler while Table 4.5b shows prior hyperparameters used in the experiments. All the parameters to be learnt are initialised arbitrarily from Unif$(1, 1000)$.

For the first experiment, a standard bootstrap VRPF modified according to Algorithm 6 is used within the PG sampler (PG-VRPF) since all the measurements arrive regularly and synchronously. To facilitate the resulting conditional VRPF, at the end of each time step changepoint times will be regenerated according to $p(\tau_k | \tau_{k-1}^{(i)}, \tau_k > t_n)$, as in [Godsill et al., 2007]. The idea is to increase the particle diversity while avoiding the re-evaluation of likelihood terms. In the second evaluation where position data is limited, the inference performance of a PG sampler equipped with the section-wise inference algorithm introduced in Section 4.3 (PG-VRPF-SS-RM) is examined. Note



(a)



(b)

Fig. 4.6 (a) The simulated trajectory, with *regular* position data. (b) The same trajectory with *occasional* (L=5) position data.

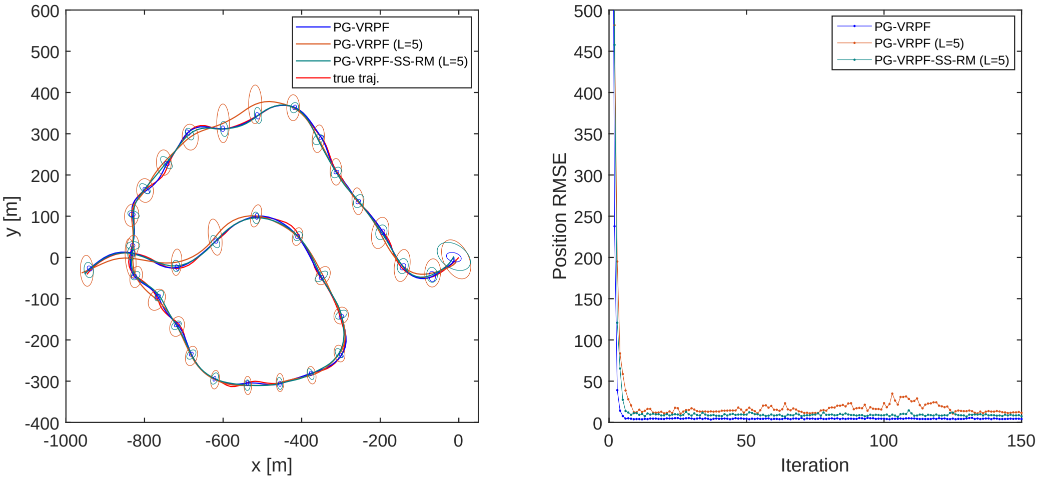


Fig. 4.7 Inference results over 150 MCMC iterations. Left: mean trajectories and error ellipses estimated by the introduced PG samplers, with burn-in samples discarded. Right: position RMSEs over MCMC iterations (including the burn-in period).

that due to the use of a multinomial resampling scheme at each timestamp the state rejuvenation step in the filter has now become the resample-move method as in [Gilks and Berzuini, 2001], although the formula of the acceptance ratio remains unchanged. We also apply the PG sampler with the conditional bootstrap VRPF in this testing scenario for comparison purposes. Moreover, as the only difference between the two testing cases is the number of position measurements, the results of the two experiments will be presented together.

The mean trajectories ($\mathbb{E}\{\mathbf{p}_{0:N}|\mathbf{y}_{0:N}\}$) estimated by the PG samplers, along with the associated 95% confidence ellipses, are given in Figure 4.7. The ellipses are intentionally spaced for better visualisation. From the figure it can be seen that state trajectories obtained from the samplers under different testing scenarios can fit the ground truth well after the burn-in period. It is obvious that PG-VRPF sampler delivers the best tracking performance when it receives position measurements regularly. This is not surprising since the sampler is provided with the full information in this case. However, when it comes to the scenario in which the number of position measurements is reduced, the PG-VRPF-SS-RM sampler outperforms the PG-VRPF sampler in terms of the position estimation accuracy. A similar conclusion can be drawn based on the right-hand side of the figure, where the position RMSEs over all MCMC iterations are plotted. The rapid drop in position RMSE in the right figure also indicates how quickly the samplers explore the state space.

While Figure 4.7 illustrates the efficacy of the PG samplers in inferring the latent state, Figure 4.8 and 4.9 shows the inference results for the system parameters. We first look at the case where enough position measurements are provided. It can be found from the figures that all the marginal posterior distributions approximated by the PG-VRPF sampler are able to concentrate around the true values. The sample autocorrelation functions (ACFs) decrease to zero very fast, indicating the Markov chain mixes well. The ACFs also show that low correlation among samples can be attained by the introduced PG-VRPF sampler. As for the second testing case with downsampled position data, the performance of the PG-VRPF degrades as expected. Notably, the PG-VRPF sampler find it difficult to estimate the parameters related to the position measurements (i.e. σ_x and σ_y), albeit being capable of inferring the rest of parameters fairly accurately. This is directly linked to its relatively poorer performance in terms of the trajectory estimation. Conversely, the PG-VRPF-SS-RM sampler appears to be able to better capture the true position measurement parameters. This leads to a better tracking performance as shown before. It is noted that the PG-VRPF-SS-RM sampler tends to become trapped in some local modes for $\sigma_{\hat{s}}$ and $\sigma_{\hat{v}}$. This can be attributed to the fact that the contribution from the position measurements overtakes that from the distance and speed measurements. The parameter estimation results from the PG-VRPF-SS-RM sampler regarding the bias term and the turn rate measurement also differ slightly from their true values. The differences are reasonable because the bias term can be regarded as a relaxation term and we have only limited observations. Nonetheless, combining the parameter estimation results with the state inference results shown in Figure 4.7, a conclusion with respect to the PG-VRPF-SS-RM sampler may be drawn for the moment: this sampler is capable of delivering accurate tracking results with inertial measurement data, limited position data and unknown parameters. This feature can be very useful in practice since it implies that the filter (VRPF-SS-RM) is more robust to model mismatch (i.e. not requiring the parameters to be set exactly). On the other hand, when accurate parameter estimation is sought and sensory information is rich, the PG-VRPF sampler should be considered instead.

Finally, there is still scope for further improvements. For instance, the gamma parameters related to changpoint arrival times, i.e. α_τ and β_τ, may be also learnt within the same framework. However, a MCMC or rejection sampling step needs to be implemented as there is no conjugate prior in this case, see [Miller, 2019; Son and Oh, 2006] for examples. It is worth stressing the inference schemes are also applicable to the fixed rate setting, since a fixed rate model can be obtained readily by enforcing $\tau = t_n - t_{n-1}$.

4.6 Summary and Conclusions

In this chapter, the problem of object tracking is investigated from a more natural modelling perspective. By decoupling the timings of manoeuvres from those of observations, a realistic model is obtained. The intrinsic coordinate dynamic model is formalised under the variable setting and we show the modelling leads to a convenient incorporation of different sensory data. The resulting data fusion problem is then tackled within a Bayesian inference framework. Effective and efficient particle filtering schemes are introduced and the experimental results on challenging tracking scenarios show significantly improved tracking performance over a bootstrap variable rate particle filter. Furthermore, two particle Gibbs samplers are presented to perform full system inference under the conditionally deterministic model and their effectiveness has been illustrated using synthesised data.

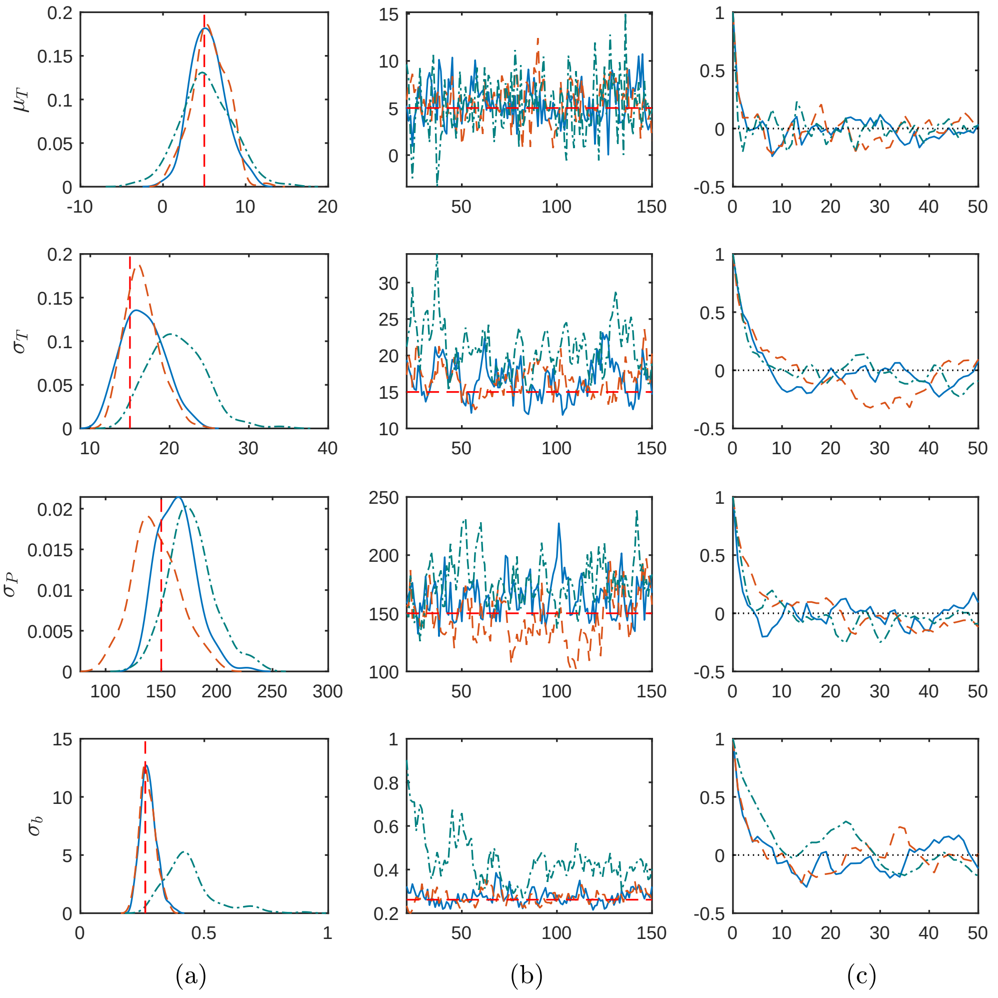
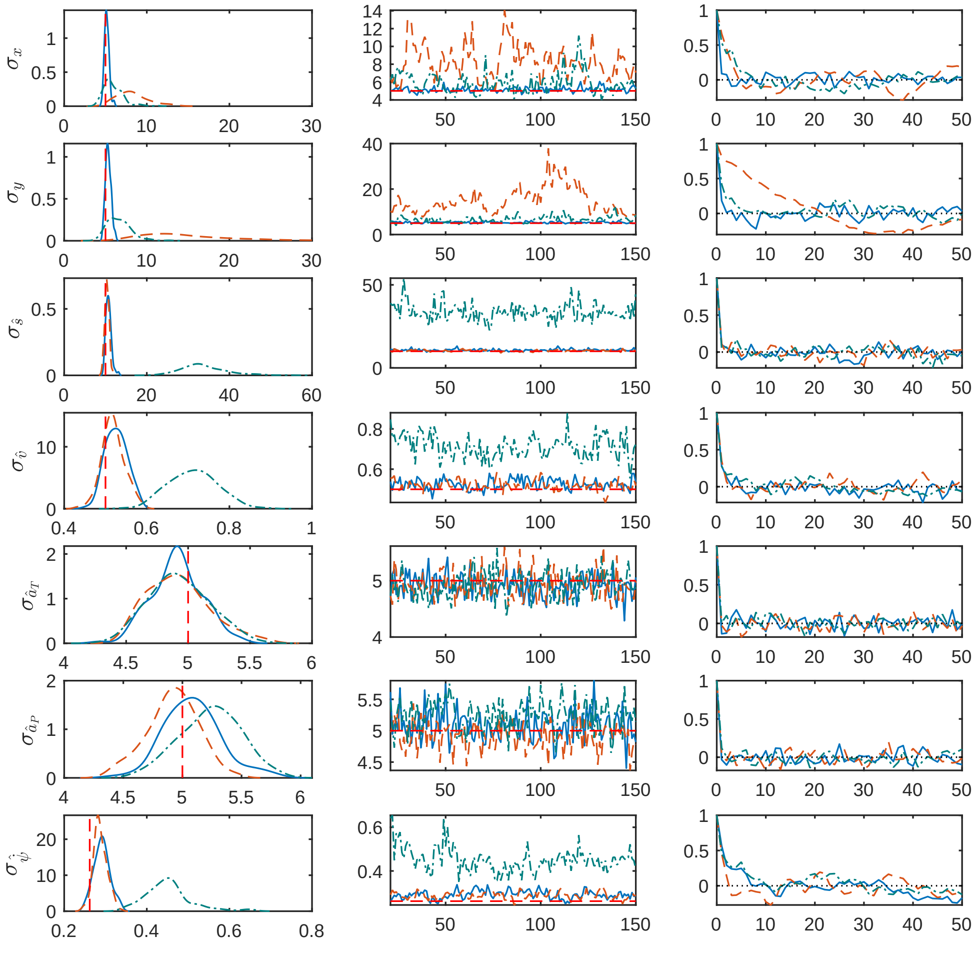


Fig. 4.8 Inference results for intrinsic dynamic model parameters $\boldsymbol{\beta}_I$ over 150 MCMC iterations, with (a) showing density curves fitted based on histograms (after the burn-in period), (b) showing the trace plots (after the burn-in period) and (c) the ACFs. ¦ true values on the histograms; - - - true values on the trace plots; —— PG-VRPF; - - - - - - PG-VRPF ($L = 5$); —·—·—·PG-VRPF-SS-RM ($L = 5$).



(a) (b) (c)

Fig. 4.9 Inference results for measurement-related model parameters $\boldsymbol{\beta}_M$ over 150 MCMC iterations, with (a) showing density curves fitted based on histograms (after the burn-in period), (b) showing the trace plots (after the burn-in period) and (c) the ACFs. ¦ true values on the histograms; - - - true values on the trace plots; —— PG-VRPF; - - - - - PG-VRPF ($L = 5$); —·—·—PG-VRPF-SS-RM ($L = 5$).

Chapter 5

Bayesian Intent Prediction

5.1 Introduction

While the previous chapters are primarily concerned with the accurate tracking of an moving object, this chapter addresses the problem of predicting the intended destination of an tracked object within a Bayesian framework. Since the destination may have a long term influence on the object's motion, knowing this information at an early stage can not only offer vital information on intent, enabling smart predictive functionalities and automation, but also facilitate more accurate state estimations, i.e. destination-aware tracking [Baccarelli and Cusani, 1998; Castanon et al., 1985; Duan and Li, 2013; Zhou et al., 2020]. It has various application areas such as smart navigation and trajectory planning for robots in the presence of other agents [Best and Fitch, 2015; Chiang et al., 2015; Kitani et al., 2012], intelligent interactive displays [Ahmad et al., 2017], revealing potential conflicts, patterns or anomalies in surveillance [d'Afflisio et al., 2018; Krishnamurthy and Gao, 2018; Millefiori et al., 2016; Uney et al., 2018], driver assistance systems [Bando et al., 2013; Völz et al., 2018], to name but a few.

5.1.1 Bayesian Intent Prediction in Object Tracking

Different from the standard tracking applications where it is usually the hidden state $\mathbf{x}_n \in \mathbb{R}^s$, which may comprise of an object's position, velocity and higher order kinematics, that is of interest, the main objective here is to carry out destination inference. Let $\mathbb{D} = \{\mathcal{D}_i : i = 1, 2, ..., N_{\mathcal{D}}\}$ be the set of $N_{\mathcal{D}}$ nominal endpoints (e.g. harbours where a vessel can dock or selectable on-display icons) of a tracked object (e.g. vessel or pointing apparatus); each can be an extended region. The objective is to sequentially calculate the probability of each of $\mathcal{D}_i \in \mathbb{D}$ being the intended destination, thus $p(\mathcal{D} = \mathcal{D}_i \mid \mathbf{y}_{0:n})$, $i = 1, ..., N_{\mathcal{D}}$. The noisy sensor measurements at the time instant t_n are $\mathbf{y}_{0:n} = \{\mathbf{y}_0, \mathbf{y}_1, ..., \mathbf{y}_n\}$ pertaining to the consecutive instants $\{t_0, t_1, ..., t_n\}$.

A Gaussian Linear Time Invariant (LTI) formulation is adopted below since provided relatively accurate measurements an approximate motion model that enables inferring the object's destination, rather than exact motion modelling, suffices for our purpose here. The state space model of $\mathbf{x}_n$ is given by

$$\mathbf{x}_n = F(\tau)\mathbf{x}_{n-1} + M(\tau) + \boldsymbol{\epsilon}_n \tag{5.1}$$

with $\boldsymbol{\epsilon}_n \sim \mathcal{N}(0, Q(\tau))$ a Gaussian process noise. Matrices F and Q as well as vector M, which define the system model, are functions of the time difference between consecutive observations $\tau = t_n - t_{n-1}$. Equation (5.1) encompasses any Gaussian LTI model, including those widely used in object tracking, e.g. the (nearly) Constant Velocity (CV), as well as mean reverting ones based on an Ornstein-Uhlenbeck (OU) process. Recall that the latter can be described by the following linear SDE:

$$d\mathbf{x}_t = \Lambda\left(\boldsymbol{\mu}_i - \mathbf{x}_t\right)dt + H(\sigma)d\mathbf{w}_t$$

which can be solved exactly, as per [Särkkä and Solin, 2019; Øksendal, 2003]. Its integration over time interval $[t_{n-1}, t_n]$ produces (5.1) where vector M is a function of the process mean $\boldsymbol{\mu}_i$. Matrix Λ contains parameters controlling the reversion strength while $\mathbf{w}_t$ is a s-dimensioned Brownian motion. The process noise scaling factor σ in matrix $H(\sigma)$ is assumed identical for brevity, though it should be clear that its value can be varied across different coordinates. Observation $\mathbf{y}_n \in \mathbb{R}^m$ is modelled by

$$\mathbf{y}_n = G\mathbf{x}_n + \boldsymbol{\nu}_n \tag{5.2}$$

where G is the observation matrix linking the hidden state to the observed measurement and noise component $\boldsymbol{\nu}_n \sim \mathcal{N}(0, V_n)$. A Gaussian distribution $\mathcal{D}_i \backsim \mathcal{N}(\mathbf{a}_i, \Sigma_i)$ is assumed here to model an endpoint. The mean $\mathbf{a}_i$ and covariance Σ_i represent the centre and orientation-extent of $\mathcal{D}_i$, respectively. This is to maintain the linear Gaussian structure of the overall system.

5.1.2 Related Work

Whilst the bridging distributions (BD) approach was introduced in [Ahmad et al., 2015; Ahmad et al., 2018], a concise overview of its key results, including schemes for estimating future state and arrival time, was presented in [Ahmad et al., 2019]. BD in [Ahmad et al., 2019, 2015; Ahmad et al., 2018] captures the influence of endpoint $\mathcal{D}_i$ on the object motion by prescribing that the motion model in (5.1) has a terminal state $\mathbf{x}_N$ at arrival time $t_N = T$ equal to that of $\mathcal{D}_i$, i.e. $\mathbf{x}_N \sim \mathcal{N}(a_i, \Sigma_i)$. Thereby, it constructs N bridged models, each with an extended state vector $\mathbf{z}_n = [\mathbf{x}_n, \mathbf{x}_N]'$. This implies that $\mathbf{x}_N$ and the initial state $\mathbf{x}_0$ at t_0 are independent, that is,

$$p(\mathbf{x}_0, \mathbf{x}_N) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_0 \\ \boldsymbol{\mu}_N \end{bmatrix}, \begin{bmatrix} \Sigma_0 & 0_s \\ 0_s & \Sigma_i \end{bmatrix} \right)$$

where 0_s is a s-by-s zero matrix. Although this can be approximately true in many scenarios, especially for $t_0 \ll T$, it is inconsistent with the Markov nature of (5.1) which dictates the transition density $p(\mathbf{x}_N|\mathbf{x}_0)$ and $p(\mathbf{x}_N) = \int p(\mathbf{x}_0)p(\mathbf{x}_N|\mathbf{x}_0)d\mathbf{x}_0$. Here, we introduce a different approach to [Ahmad et al., 2019, 2015; Ahmad et al., 2018] in which the destination point is considered to be a ‘pseudo-observation’ rather than a terminal state $\mathbf{x}_N$ of the system. Consequently, the mathematics of the dynamical model and observation process are made consistent with the Markov state process, in contrast with [Ahmad et al., 2019, 2015; Ahmad et al., 2018]. This new interpretation leads to two new destination prediction algorithms that can substantially reduce the computational complexity of the inference routine for all Gaussian LTI motion models in (5.1), e.g. by over 65% with Algorithm 1 in some cases (see Section 5.2.3).

Early studies in object tracking area considering the incorporation of, often known, destination information for improving the state estimation accuracy can be found in [Baccarelli and Cusani, 1998; Castanon et al., 1985], where destination-aware trackers are designed based on the premise that the arrival time at destination is known. This particular assumption allows the construction of a state transition distribution dependent on the future destination information and hence enables constrained state estimation. While these methods require additional timing information to build the condition on destination, OU processes provide an alternative solution in which the destination information is modelled inherently. Provided *a priori* learnt means (destination-related information), tracking algorithms based upon an OU model have been shown to reliably model and estimate vessels motion in maritime surveillance [d’Afflisio et al., 2018; Millefiori et al., 2016; Uney et al., 2018] and to effectively extract object's intended destination as in [Ahmad et al., 2016a; Gan et al., 2019]. It is worth noting that mean-reverting processes impose a prior which dictates that the farther away the object is from the process mean value (e.g. Cartesian location of the destination) the stronger restoring force it will experience. This feature can result in the problem of model mismatch in some scenarios and hence special attention should be paid when using OU model to describe the motion of an object. As for the BD approach considered here, it incorporates the desired destination information via the use of pseudo-observation and hence can be combined with any suitably chosen dynamic model. Note also that pseudo-observations (or 'pseudo-measurements') have been widely used for constrained state estimation owing to its simplicity, as in [Doran, 1992; Duan and Li, 2013; Zhou et al., 2020]; such measurements are usually obtained from linear or nonlinear equality constraints associated with the problem. For instance, in [Zhou et al., 2020] the pseudo-observations arise from the linear constraint that the object motion is subject to an unknown straight line. Notably, in the paper an augmented state vector is also considered, enabling the tracking of the intended destination on the line. This is in a similar spirit to that of [Ahmad et al., 2018] and thus the inconsistency issue still exists in the resulting model.

Discretised state-space models based on reciprocal processes or other models from natural language processing are proposed in [Fanaswala and Krishnamurthy, 2015; Krishnamurthy and Gao, 2018] to recognise the object intent. They stipulate that the target should pass through a finite number of predefined spatial grid cells to reach its endpoint. The proposed approach in this chapter aims to predict the unknown destination $\mathcal{D}$, not estimate $\mathbf{x}_n$, and has notably lower complexity compared with those in [Fanaswala and Krishnamurthy, 2015; Krishnamurthy and Gao, 2018]. It utilises continuous-time state space models to treat asynchronous measurements and capture, via a Markov bridge, the long term underlying dependencies in the object trajectory as dictated by the intended endpoint.

Finally, various data driven prediction-classification methods rely on a dynamical model and/or pattern of life learnt from previously recorded data, e.g. [Bando et al., 2013; Chiang et al., 2015; Kitani et al., 2012; Völz et al., 2018]. Whilst such techniques typically require substantial parameters training from extensive data sets (not always available), a probabilistic model-based framework is adopted here. It uses known dynamical and measurements models, with a few unknown parameters [Bar-Shalom et al., 2011; Haug, 2012]. Subsequently, an efficient inference approach, which requires minimal training, is introduced.

The inference approach has been published elsewhere [Liang et al., 2019].

5.2 Bayesian destination Inference

Within a Bayesian framework, for each candidate endpoint $\mathcal{D}_i \in \mathbb{D}$, the following posterior distribution is sought:

$$\begin{aligned} p(\mathcal{D} = \mathcal{D}_i | \mathbf{y}_{0:n}) &\propto p(\mathbf{y}_{0:n} | \mathcal{D} = \mathcal{D}_i) p(\mathcal{D} = \mathcal{D}_i) \\ &= \Big(\int p(\mathbf{y}_{0:n} | \mathcal{D} = \mathcal{D}_i, T) p(T | \mathcal{D} = \mathcal{D}_i) dT \Big) p(\mathcal{D} = \mathcal{D}_i) \end{aligned} \tag{5.3}$$

where $p(\mathcal{D} = \mathcal{D}_i)$ is the prior on the i-th destination while T depicts the arrival time at the location. This prior can normally be attained from relevant contextual information or learnt pattern-of-life; in cases where such information is not available, a non-informative prior may be considered. The objective of the inference module at t_n is hence to estimate the likelihoods $p(\mathbf{y}_{0:n} | \mathcal{D} = \mathcal{D}_i)$, $i = 1, 2, ..., N_{\mathcal{D}}$. The key challenge here is to capture the influence of endpoint $\mathcal{D}_i$ on the object behaviour while utilising the dynamic models (5.1) and measurement models (5.2). To this end, in the sequel a novel bridging formulation that facilitates introducing the conditioning on $\mathcal{D}_i$ in (5.3) will be described.

5.2.1 Pseudo-observation Formulation

The trajectory of the tracked object must end at the intended destination at arrival time T, albeit the exact path being random as per (5.1). A Markov bridge from t_n to T for $\mathcal{D}_i \in \mathbb{D}$ can be built to capture the influence of intent on the object motion by defining the pseudo-observation $\tilde{\mathbf{y}}_N^i$ at $t_N = T$,

$$p(\tilde{\mathbf{y}}_N^i = \mathbf{a}_i | \mathbf{x}_N, \mathcal{D} = \mathcal{D}_i) = \mathcal{N}(\tilde{\mathbf{y}}_N^i = \mathbf{a}_i \mid \tilde{G} \mathbf{x}_N, \Sigma_i) \tag{5.4}$$

Pseudo-observation matrix $\tilde{G}$ depends on the information available on the i^{th} endpoint $\mathcal{D}_i \backsim \mathcal{N}(\mathbf{a}_i, \Sigma_i)$. The design of pseudo-observations depends on what elements the state vector contains and what information is available at a given destination. For instance, in a CV model $\tilde{\mathbf{y}}_K^i$ may contain both position and velocity information at the endpoint. For brevity of notation $\mathcal{D} = \mathcal{D}_i$ is replaced by $\mathcal{D}_i$ henceforth. Based on (5.4), we can express the likelihood conditioned on the arrival time T at time instant t_n by

$$p(\mathbf{y}_{0:n} | \mathcal{D}_i, T) = p(\mathbf{y}_0 | \mathcal{D}_i, T) \prod_{l=1}^{n} p(\mathbf{y}_l | \mathbf{y}_{0:l-1}, \mathcal{D}_i, T) \tag{5.5}$$

such that $\tilde{\mathbf{y}}_N^i$ introduces the conditioning on the endpoint $\mathcal{D}_i$ since $p(\mathbf{y}_{0:n}|\mathcal{D}_i, T) = p(\mathbf{y}_{0:n}|\tilde{\mathbf{y}}_N^i = \mathbf{a}_i, T)$. While the likelihood $p(\mathbf{y}_{0:n-1}|\mathcal{D}_i, T)$ pertaining to the previous time instant t_{n-1} is available at t_n, estimating the arrival-time-conditioned Prediction Error Decompositions (PEDs) [Harvey, 1989], i.e. $p(\mathbf{y}_n|\mathbf{y}_{0:n-1}, \mathcal{D}_i, T)$, suffices to sequentially calculate the likelihood in (5.5).

Since T is unknown, it is treated as a random variable in our formulation. In practice, a prior distribution on T can be assumed based on the context, e.g. uniform where $p(T \mid \mathcal{D}_i) = \mathcal{U}(t_a, t_b)$ within the time window $\mathcal{T} = [t_a, t_b]$. The arrival time can then be integrated out via

$$p(\mathbf{y}_{0:n} \mid \mathcal{D}_i) = \int_{T \in \mathcal{T}} p(\mathbf{y}_{0:n} \mid \mathcal{D}_i, T) p(T \mid \mathcal{D}_i) dT \tag{5.6}$$

to obtain the likelihood in (5.3). Since it is a one dimensional integral, a numerical approximation can be efficiently applied [Davis and Rabinowitz, 2007], e.g. Simpson's rule. This requires q evaluations of the arrival-time-conditioned PEDs $p(\mathbf{y}_{0:n}|\mathcal{D}_i, T_n)$, for all $T_n \in \{T_1, T_2, ..., T_q\}$ which are drawn from the prior $p(T|\mathcal{D}_i)$.

5.2.2 Proposed Predictors

In this section, we present two algorithms for estimating the desired destination-dependent likelihood. The condition of the unknown arrival time T will be made implicit to simplify the notation in the derivations when confusion is not possible. It will be shown that the proposed methods lead to an efficient Kalman-filtering-type routine for the prediction of $\mathcal{D}$.

Algorithm 1

At time instant t_n, the PED conditioned on the i-th endpoint and a given arrival time T can be written as

$$\begin{aligned} p(\mathbf{y}_n|\mathbf{y}_{0:n-1}, \mathcal{D}_i) &= p(\mathbf{y}_n|\mathbf{y}_{0:n-1}, \tilde{\mathbf{y}}_N^i = \mathbf{a}_i) \\ &= \int p(\mathbf{y}_n|\mathbf{x}_n) p(\mathbf{x}_n|\mathbf{y}_{0:n-1}, \tilde{\mathbf{y}}_N^i = \mathbf{a}_i) d\mathbf{x}_n \\ &= \int p(\mathbf{y}_n|\mathbf{x}_n) \frac{p(\mathbf{x}_n|\mathbf{y}_{0:n-1}) p(\tilde{\mathbf{y}}_N^i = \mathbf{a}_i|\mathbf{x}_n)}{p(\tilde{\mathbf{y}}_N^i = \mathbf{a}_i|\mathbf{y}_{0:n-1})} d\mathbf{x}_n \end{aligned} \tag{5.7}$$

which can be computed by parts and is dubbed Algorithm 1. Given the linear Gaussian nature of (5.1) and (5.2), the state posterior at the previous time instant t_{n-1} is given by

$$p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1}) = \mathcal{N}(\mathbf{x}_{n-1}|\boldsymbol{\mu}_{n-1|n-1}, \Sigma_{n-1|n-1}) \tag{5.8}$$

where the mean $\boldsymbol{\mu}_{n-1|n-1}$ and covariance $\Sigma_{n-1|n-1}$ are the filtering outputs of a Kalman Filter (KF), which are optimal in the mean squared error sense [Haug, 2012]. Under the standard rules for marginalisation of a joint Gaussian [Bishop, 2006], the predictive distribution in the KF can be obtained as follows:

$$\begin{aligned} p(\mathbf{x}_n|\mathbf{y}_{0:n-1}) &= \int p(\mathbf{x}_n|\mathbf{x}_{n-1})p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1})d\mathbf{x}_{n-1} \\ &= \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_{n|n-1}, \Sigma_{n|n-1}) \end{aligned} \tag{5.9}$$

with

$$\begin{aligned} \boldsymbol{\mu}_{n|n-1} &= F(\tau)\boldsymbol{\mu}_{n-1|n-1} + M(\tau) \\ \Sigma_{n|n-1} &= F(\tau)\Sigma_{n-1|n-1}F(\tau)^T + Q(\tau) \end{aligned}$$

and $\tau = t_n - t_{n-1}$. Based on the pseudo-observation density (5.4) and state transition density (5.1), the pseudo-observation conditioned on the state at time t_n is given by

$$\begin{aligned} p(\tilde{\mathbf{y}}_N^i = \mathbf{a}_i|\mathbf{x}_n) &= \int p(\tilde{\mathbf{y}}_N^i = \mathbf{a}_i|\mathbf{x}_N)p(\mathbf{x}_N|\mathbf{x}_n)d\mathbf{x}_N \\ &= \int \mathcal{N}(\tilde{\mathbf{y}}_N^i = \mathbf{a}_i|\tilde{G}\mathbf{x}_N, \Sigma_i) \\ &\quad\times \mathcal{N}\Big(\mathbf{x}_N|F(T-t_n)\mathbf{x}_n + M(T-t_n),\ Q(T-t_n)\Big)d\mathbf{x}_N \\ &= \mathcal{N}(\tilde{\mathbf{y}}_N^i = \mathbf{a}_i|\boldsymbol{\mu}_{\tilde{\mathbf{y}}}, \Sigma_{\tilde{\mathbf{y}}}) \end{aligned} \tag{5.10}$$

where

$$\begin{aligned} \boldsymbol{\mu}_{\tilde{\mathbf{y}}} &= \tilde{G}\Big[F(T-t_n)\mathbf{x}_n + M(T-t_n)\Big] \\ \Sigma_{\tilde{\mathbf{y}}} &= \tilde{G}Q(T-t_n)\tilde{G}^T + \Sigma_i \end{aligned}$$

By utilising the following Gaussian identity

$$\mathcal{N}(x;\mu_1,\Sigma_1)\mathcal{N}(\mu_2;Lx,\Sigma_2) \propto \mathcal{N}(x;\mu_3,\Sigma_3)$$

with $\Sigma_3^{-1} = \Sigma_1^{-1} + L^T\Sigma_2^{-1}L$, $\mu_3 = \Sigma_3(\Sigma_1^{-1}\mu_1 + L^T\Sigma_2^{-1}\mu_2)$ and the matrix inversion lemma (Woodbury formula), the second component in the integral in (5.7) is given by

$$\begin{aligned} p(\mathbf{x}_n|\mathbf{y}_{0:n-1}, \tilde{\mathbf{y}}_N^i = \mathbf{a}_i) &\propto p(\mathbf{x}_n|\mathbf{y}_{0:n-1})p(\tilde{\mathbf{y}}_N^i = \mathbf{a}_i|\mathbf{x}_n) \\ &= \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_*, \Sigma_*) \end{aligned} \tag{5.11}$$

with

$$\begin{aligned} \boldsymbol{\mu}_* &= \boldsymbol{\mu}_{n|n-1} + L_*\Big[\mathbf{a}_i - B_*\boldsymbol{\mu}_{n|n-1} - \tilde{G}M(T-t_n)\Big] \\ L_* &= \Sigma_{n|n-1}B_*^T\Big[B_*\Sigma_{n|n-1}B_*^T + \tilde{G}Q(T-t_n)\tilde{G}^T + \Sigma_i\Big]^{-1} \\ \Sigma_* &= \Big(I - L_*B_*\Big)\Sigma_{n|n-1} \\ B_* &= \tilde{G}F(T-t_n) \end{aligned}$$

Now the individual PED in (5.7) can be readily obtained as

$$\begin{aligned} p(\mathbf{y}_n|\mathbf{y}_{0:n-1}, \mathcal{D}_i) &= \int p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{0:n-1}, \tilde{\mathbf{y}}_N^i = \mathbf{a}_i)d\mathbf{x}_n \\ &= \int \mathcal{N}(\mathbf{y}_n|G\mathbf{x}_n, V_n)\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_*, \Sigma_*)d\mathbf{x}_n \\ &= \mathcal{N}(\mathbf{y}_n|\boldsymbol{\mu}_{\mathbf{y}}, \Sigma_{\mathbf{y}}) \end{aligned} \tag{5.12}$$

with $\boldsymbol{\mu}_{\mathbf{y}} = G\boldsymbol{\mu}_*$ and $\Sigma_{\mathbf{y}} = G\Sigma_*G^T + V_n$. Since the condition of the arrival time has thus far been imposed implicitly in our derivations, the likelihood computed at this stage is actually

$$p(\mathbf{y}_{0:n}|\mathcal{D}_i, T) = p(\mathbf{y}_n|\mathbf{y}_{0:n-1}, \mathcal{D}_i, T)p(\mathbf{y}_{0:n-1}|\mathcal{D}_i, T) \tag{5.13}$$

where the first term in the right hand side is obtained by plugging the arrival time back into the condition while the second term is obtained at the last time instant. Hence, the condition on the arrival time must be integrated out based on (5.6) in order to obtain the desired likelihood $p(\mathbf{y}_{0:n}|\mathcal{D}_i)$. As already noted in Section 5.2.1, this may be accomplished via any valid numerical approximation schemes. Apart from this, the standard Kalman correction step needs to be performed to calculate the new posterior $p(\mathbf{x}_n|\mathbf{y}_{0:n}) = \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_{n|n}, \Sigma_{n|n})$ ready for the PED estimation at the next step t_{n+1}.

Algorithm 2

An alternative interpretation of endpoint inference using pseudo-observations can be achieved by re-writing the Bayesian smoothing (fixed-interval) equation, termed Rauch-Tung-Striebel smoother in linear Gaussian cases [Rauch et al., 1965],

$$\begin{aligned}
&p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1}, \tilde{\mathbf{y}}_N^i = \mathbf{a}_i) \\
&= \int \frac{p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1})p(\mathbf{x}_N|\mathbf{x}_{n-1})}{p(\mathbf{x}_N|\mathbf{y}_{0:n-1})} p(\mathbf{x}_N|\mathbf{y}_{0:n-1}, \tilde{\mathbf{y}}_N^i = \mathbf{a}_i) d\mathbf{x}_N \\
&= \int \frac{p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1})p(\mathbf{x}_N|\mathbf{x}_{n-1})}{p(\mathbf{x}_N|\mathbf{y}_{0:n-1})} \frac{p(\mathbf{x}_N|\mathbf{y}_{0:n-1})p(\tilde{\mathbf{y}}_N^i = \mathbf{a}_i|\mathbf{x}_N)}{p(\tilde{\mathbf{y}}_N^i = \mathbf{a}_i|\mathbf{y}_{0:n-1})} d\mathbf{x}_N \\
&\quad \propto p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1}) \int p(\tilde{\mathbf{y}}_N^i = \mathbf{a}_i|\mathbf{x}_N) p(\mathbf{x}_N|\mathbf{x}_{n-1}) d\mathbf{x}_N \\
&\quad \propto \mathcal{N}(\mathbf{x}_{n-1}|\boldsymbol{\mu}_{n-1|n-1}, \Sigma_{n-1|n-1}) \mathcal{N}(\tilde{\mathbf{y}}_N^i = \mathbf{a}_i|\boldsymbol{\mu}_{\tilde{\mathbf{y}}}, \Sigma_{\tilde{\mathbf{y}}}) \\
&= \mathcal{N}(\mathbf{x}_{n-1}|\tilde{\boldsymbol{\mu}}, \tilde{\Sigma})
\end{aligned} \tag{5.14}$$

with $\{\boldsymbol{\mu}_{\tilde{\mathbf{y}}}, \Sigma_{\tilde{\mathbf{y}}}\}$ and $\{\tilde{\boldsymbol{\mu}}, \tilde{\Sigma}\}$ obtained similarly to those in (5.10) and (5.11). The PED can then be easily shown to reduce to

$$\begin{aligned}
&p(\mathbf{y}_n|\mathbf{y}_{0:n-1}, \mathcal{D}_i) \\
&= p(\mathbf{y}_n|\mathbf{y}_{0:n-1}, \tilde{\mathbf{y}}_N^i = \mathbf{a}_i) \\
&= \int \Big(\int p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1}, \tilde{\mathbf{y}}_N^i = \mathbf{a}_i) p(\mathbf{x}_n|\mathbf{x}_{n-1}, \tilde{\mathbf{y}}_N^i = \mathbf{a}_i) d\mathbf{x}_{n-1} \Big) p(\mathbf{y}_n|\mathbf{x}_n) d\mathbf{x}_n \\
&= \mathcal{N}(\mathbf{y}_n|\mu_{\mathbf{y}}, \Sigma_{\mathbf{y}})
\end{aligned} \tag{5.15}$$

with

$$\begin{aligned}
\mu_{\mathbf{y}} &= G\Big[F(\tau)\tilde{\boldsymbol{\mu}} + M(\tau) + L_{\mathbf{y}}\Big[\mathbf{a}_i - \tilde{G}\Big(F(T-t_n)\Big(F(\tau)\tilde{\mu} \\
&\qquad - M(\tau)\Big) + M(T-t_n)\Big)\Big]\Big] \\
\Sigma_{\mathbf{y}} &= G\Big[A_{\mathbf{y}}\tilde{\Sigma}A_{\mathbf{y}}^T + (I - L_{\mathbf{y}}B_{\mathbf{y}})Q(\tau)\Big]G^T + V_n \\
L_{\mathbf{y}} &= Q(\tau)B_{\mathbf{y}}^T\Big[B_{\mathbf{y}}Q(\tau)B_{\mathbf{y}}^T + \tilde{G}Q(T-t_n)\tilde{G}^T + \Sigma_i\Big]^{-1} \\
A_{\mathbf{y}} &= \Big(I - L_{\mathbf{y}}B_{\mathbf{y}}\Big)F(\tau) \\
B_{\mathbf{y}} &= \tilde{G}F(T-t_n)
\end{aligned}$$

It should be noted that the state transition density in the inner integral of (5.15) needs to be modified to incorporate the information from destination since

$$p(\mathbf{x}_n|\mathbf{x}_{n-1}, \tilde{\mathbf{y}}_N^i = \mathbf{a}_i) \neq p(\mathbf{x}_n|\mathbf{x}_{n-1})$$

The pseudo-code for the proposed algorithms, which use (5.11)-(5.12) and (5.14)-(5.15), respectively, has been summarised as below. The arrival-time-conditioned likelihoods are computed recursively as in (5.5) and marginalisation of T in (5.6) is approximated by numerical quadrature. We also define $L_n^{i,q} = p(\mathbf{y}_{0:n}|\mathcal{D}_i, T_q)$ for the ease of presentation.

```
// Initialisation
Set Kalman mean $\boldsymbol{\mu}_{0|0}$ and covariance $\Sigma_{0|0}$ according to a prior;
Set likelihoods $L_0^{i,q} = 1$ for all $\mathcal{D}_i \in \mathbb{D}$ and $q = 1, \ldots, M$;
Set $Flag$ = true/false (Algorithm 1 / Algorithm 2);
// Sequential inference
for $n = 1, 2, \ldots, N$ do
    // at each time instant
    Standard Kalman prediction: $p(\mathbf{x}_n|\mathbf{y}_{0:n-1}) = \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_{n|n-1}, \Sigma_{n|n-1})$;
    Standard Kalman correction: $p(\mathbf{x}_n|\mathbf{y}_{0:n}) = \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_{n|n}, \Sigma_{n|n})$;
    for  $i = 1, 2, \ldots, N_{\mathcal{D}}$ do
        // for each candidate destination
        for  $q = 1, 2, \ldots, N_{\mathcal{D}}$ do
            // for each quadrature point
            if $Flag$ then
                // Algorithm 1
                Compute $\boldsymbol{\mu}_*$ and $\Sigma_*$ via (5.11);
                Compute $l_n^{i,q} = p(\mathbf{y}_n|\mathbf{y}_{0:n-1}, \mathcal{D}_i, T_q)$ via (5.12);
            else
                // Algorithm 2
                Compute $\tilde{\boldsymbol{\mu}}$ and $\tilde{\Sigma}$ via (5.14);
                Compute $l_n^{i,q} = p(\mathbf{y}_n|\mathbf{y}_{0:n-1}, \mathcal{D}_i, T_q)$ via (5.15);
            end
            Update $T_q$ likelihood $L_n^{i,q} = L_{n-1}^{i,q} \times l_n^{i,q}$;
        end
        Approximate $P_n^i \approx p(\mathbf{y}_{0:n}|\mathcal{D}_i)$ via numerical quadrature;
    end
    Obtain at $t_n$: $p(\mathcal{D} = \mathcal{D}_i|\mathbf{y}_{0:n}) \approx \frac{P_n^i \times p(\mathcal{D}=\mathcal{D}_i)}{\sum_{j \in N_{\mathcal{D}}} P_n^j \times p(\mathcal{D}=\mathcal{D}_i)}$
end
```

Algorithm 8: Sequential Intent Inference with Pseudo-observations

5.2.3 Computational Complexity Analysis

The computational complexity is analysed here by counting each floating-point multiplication followed by one addition, i.e. a "flop" as in [Axelsson, 1996]. Let l be the dimension of the pseudo-observation vector, whilst s and m are the dimensions of the state and measurement vectors, respectively. The computational cost of a Kalman filter at t_n is given by [Willner et al., 1976]:

$$\mathcal{C}_{\mathrm{KF}}(m,s)=\frac{3}{2}(s^3+s^2)+ms\Big(\frac{3}{2}(s+m)+3\Big)+\frac{2}{3}(m^3-m) \tag{5.16}$$

Since the original BD in [Ahmad et al., 2018] runs one KF per endpoint with an augmented state of dimension $2\times s$ (i.e. endpoint state $\mathbf{x}_N$ is also included in the state vector) and q points to approximate (5.6), its computational cost at a one time step, assuming $l=s$, is

$$\mathcal{C}_{\mathrm{BD}}(m,s)=qN\Gamma_{\mathrm{BD}} \tag{5.17}$$

where

$$\Gamma_{\mathrm{BD}}=\frac{41}{2}s^3+\frac{19}{2}s^2-2s+3ms(2s+m+2)+\frac{2}{3}(m^3-m)$$

Following a similar counting procedure, the complexities of the proposed Algorithms 1 and 2 are

$$\mathcal{C}_1(m,s)=qN\Big(\Gamma_1(m,s)+\mathcal{C}_{\mathrm{KF}}(m,s)\Big) \tag{5.18}$$

$$\mathcal{C}_2(m,s)=qN\Big(\Gamma_2(m,s)+\mathcal{C}_{\mathrm{KF}}(m,s)\Big) \tag{5.19}$$

respectively, where

$$\begin{aligned}\Gamma_1(m,s)&=\frac{43}{6}s^3+\frac{9}{2}s^2-\frac{2}{3}s+ms(s+\frac{1}{2}m+\frac{3}{2})\\ \Gamma_2(m,s)&=\frac{52}{3}s^3+12s^2-\frac{4}{3}s+ms(s+\frac{1}{2}m+\frac{3}{2})\end{aligned}$$

This is for all models in (5.1), including those dependent on $\mathcal{D}_i$ such as OU-type models in [Ahmad et al., 2018]. For models independent of $\mathcal{D}_i$, e.g. CV, the introduced algorithms run one KF for all nominal destinations unlike the original BD. Their costs thereby reduce to:

$$\begin{aligned}\mathcal{C}_1(m,s)&=qN\Gamma_1(m,s)+\mathcal{C}_{\mathrm{KF}}(m,s)\\ \mathcal{C}_2(m,s)&=qN\Gamma_2(m,s)+\mathcal{C}_{\mathrm{KF}}(m,s)\end{aligned}$$

Table 5.1 Complexity of BD methods as function of s; $c = 1.5s^3$.

Model (5.1)	Algorithm 1	Algorithm 2	Original BD
With $\mathcal{D}_i$	$\mathcal{O}(8.7qNs^3)$	$\mathcal{O}(18.8qNs^3)$	$\mathcal{O}(20.5qNs^3)$
Without $\mathcal{D}_i$	$\mathcal{O}(7.2qNs^3 + c)$	$\mathcal{O}(17.3qNs^3 + c)$	$\mathcal{O}(20.5qNs^3)$

Table 5.1 shows the complexity order $\mathcal{O}(\cdot)$ of all methods as a function of the state dimension s; complexity order for m or any other parameter can be similarly attained from (5.18) and (5.19). As well as (5.17)-(5.19), the table clearly illustrates that the proposed formulation is significantly more efficient compared with the original BD. Algorithm 1 can reduce the computational complexity of the destination inference routine by approximately 57.5%; 65% for motion models independent of $\mathcal{D}_i$. As for $l < s$, i.e. only partial knowledge of the destination state is available, the computational complexity is further reduced. In [Ardeshiri et al., 2017], a bridging-based method was also proposed to reduce the inference complexity. However, it: a) uses the same formulation as in [Ahmad et al., 2018] which leads to inconsistencies, b) utilises crude approximations with unknown impact on the predictions quality, and c) only applies to models that are independent of $\mathcal{D}_i$. Conversely, the methods presented here offer a consistent and efficient solution to the prediction problem without any approximations and it is applicable to any model in (5.1).

5.3 Experimental Results

5.3.1 Maritime Surveillance

First, the performance of the proposed algorithms is compared with that of the original BD in [Ahmad et al., 2018] using a maritime surveillance example. The aim is to predict a vessel endpoint, out of $N_{\mathcal{D}} = 6$ possible harbours in a bay, from noisy observations of its 2-D position, e.g. AIS-based. Vessel trajectories are generated synthetically from a bridged CV model as per [Ahmad et al., 2018], each starting from a rendezvous area off the coast. For the three predictors, we employ a continuous-time CV model whose state vector is 2-D per coordinate $\mathbf{x}_n = [\xi, \dot{\xi}]'$. The corresponding $F(\tau)$, $M(\tau)$ and $Q(\tau)$ in (5.1) are given by

$$F(\tau) = \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix}, \quad M(\tau) = [0, 0]^T, \quad Q(\tau) = \sigma^2 \begin{bmatrix} \frac{\tau^3}{3} & \frac{\tau^2}{2} \\ \frac{\tau^2}{2} & \tau \end{bmatrix}$$

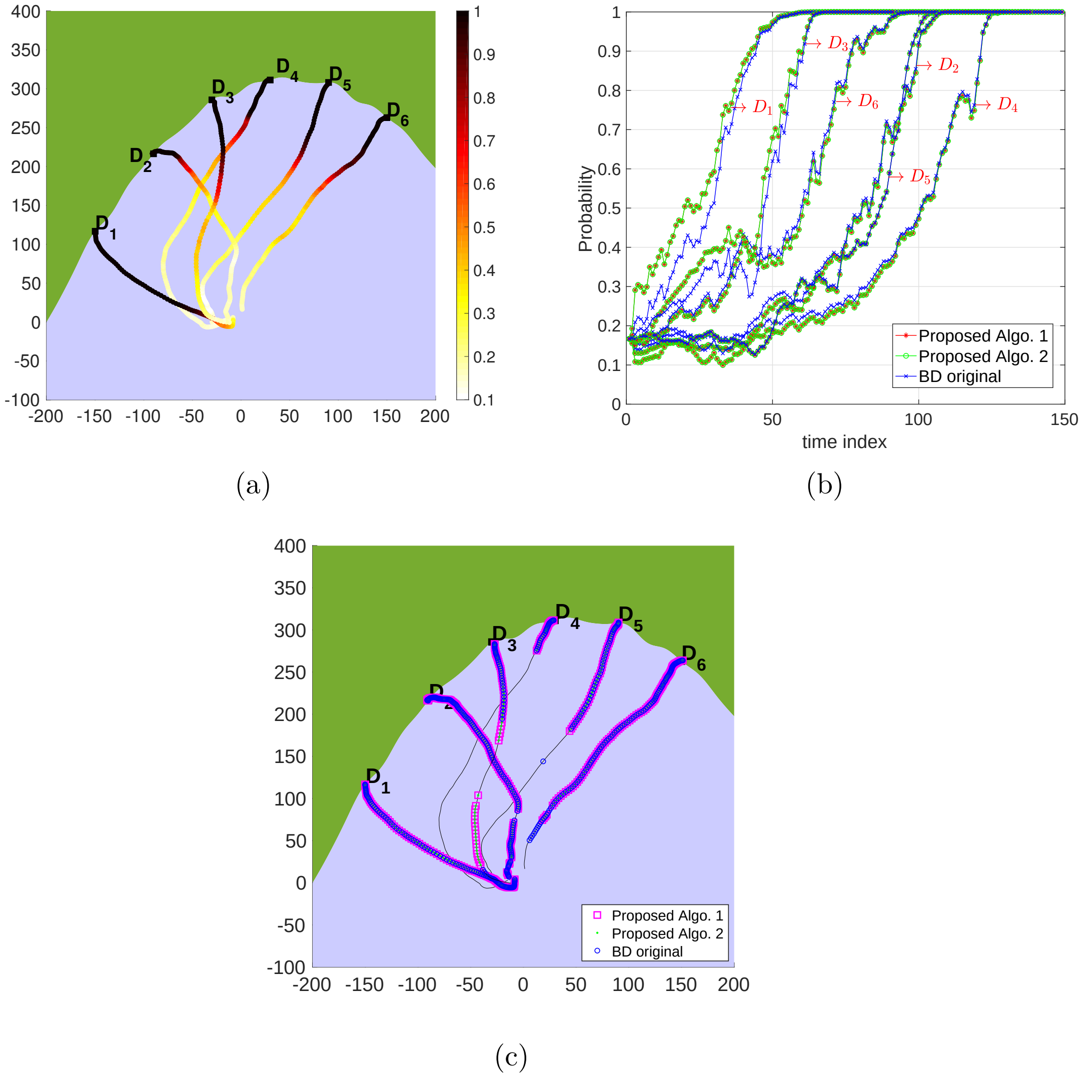
with σ being the process noise parameter. The 2-D motion of the vessel is then obtained by cascading two independent 1-D processes. For the measurement model, the observation matrix in (5.2) is given by $G = [I_2, 0_2]$. The measurement noises are assumed to be uncorrelated and hence $V_n = \sigma_m^2 I_2$. Simpson's quadrature scheme with $q = 15$ from a uniform prior $p(T|\mathcal{D}_i) = \mathcal{U}(80\text{mins}, 150\text{mins})$ is adopted to numerically marginalise over the possible arrival times. A non-informative prior is placed on the harbours, i.e. $p(\mathcal{D} = \mathcal{D}_i) = 1/6$ for $\mathcal{D}_i \in \mathbb{D}$. Note that for each $\mathcal{D}_i \in \mathbb{D}$, $\mathbf{a}_i$ is the location of the centre of the i^{th} harbour while $\Sigma_i = \text{diag}[5I_2, 0.1I_2]$ describes its region. We adopt maximum *a posteriori* (MAP) estimation for inferring the intended destination of a vessel

$$\widehat{\mathcal{D}}_n = \text{argmax}_{\mathcal{D}_i \in \mathbb{D}}\, p(\mathcal{D} = \mathcal{D}_i|\mathbf{y}_{0:n}) \tag{5.20}$$

based on which the prediction success is defined by $\mathcal{S}_n = 1$ if $\widehat{\mathcal{D}}_n = \mathcal{D}_{\text{true}}$ and $\mathcal{S}_n = 0$ otherwise (for all $t_n \in \{t_0, t_1, \ldots, t_N = T\}$). Consequently, the portion of each vessel trajectory (in time) for which correct predictions are made can be computed as $\frac{1}{N}\sum_{k=n}^{N} \mathcal{S}_k$.

The top panel of Figure 5.1 shows six synthetic tracks with the prediction results for the true destination, i.e. $p(\mathcal{D} = \mathcal{D}_{\text{true}}|\mathbf{y}_{0:n})$, while the bottom panel gives the proportion of each trajectory for which the true destination is correctly inferred using MAP estimate. Specifically, in Figure 5.1a the uncertainty diminishes as the target moves towards its endpoint and vice versa, showing the effectiveness of the proposed BD approach in predicting $\mathcal{D}$. This is closely related to the intent prediction results shown in Figure 5.1b. It should be noted that there is no limitation on the total travelling time T, the speed of identifying the most likely destination depends more on the followed trajectory, as can be seen in Figure 5.1a and 5.1c. For instance, the intent of the tracked object heading to D_1 is revealed significantly earlier than that heading to D_4. Moreover, from Figure 5.1b and 5.1c it can be seen that the difference between the outcome of the proposed formulation and that of the original BD is small whereas the results of the two introduced algorithms are nearly identical.

The inference performance of the algorithms is also evaluated based on 100 synthetic vessel tracks. The process noise parameter of the bridged CV for the simulations is set to 0.2. Figure 5.2 depicts the averaged portions of correct predictions output by all algorithms, with different process and measurement noise parameters (i.e. σ and σ_m). More specifically, Figure 5.2a shows that provided accurate measurements the prediction performance of all predictors is robust against the mismatch between the simulation model and the actual model while Figure 5.2b illustrates that provided

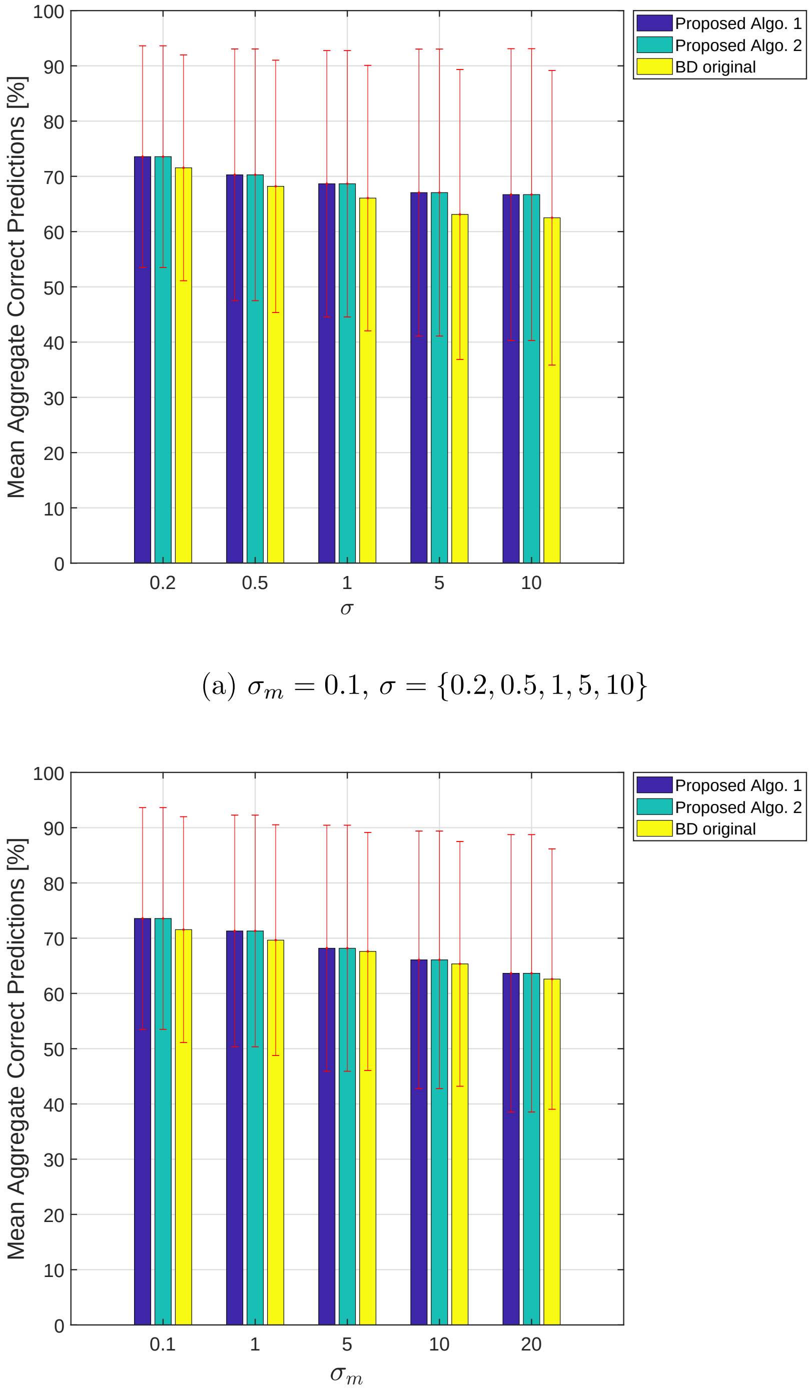


(a)

(b)

(c)

Fig. 5.1 Six synthetic vessel trajectories showing the intend prediction results. ($\sigma = 0.1, \sigma_m = 0.1$) (a) Shows $\mathbf{y}_{0:N}$ coloured by the prediction probability against true destination $p(\mathcal{D} = \mathcal{D}_{\text{true}}|\mathbf{y}_{0:n})$ from Algorithm 1. (b) Shows $p(\mathcal{D} = \mathcal{D}_{\text{true}}|\mathbf{y}_{0:n})$ for all algorithms; D_k ($k = 1, \ldots, 6$) indicates the corresponding track. (c) gives the portion of each vessel path for which correct destination predictions are made via MAP estimate.



(a) $\sigma_m = 0.1$, $\sigma = \{0.2, 0.5, 1, 5, 10\}$

(b) $\sigma = 0.2$, $\sigma_m = \{0.1, 1, 5, 10, 20\}$

Fig. 5.2 Portions of correct predictions averaged over 100 synthetic tracks. (error bars are one standard deviation)

correct modelling all methods maintain their performance at a satisfactory level ($> 60\%$) with increasingly noisy measurements. It is also visible that in the figures that the proposed algorithms yield better results than the original BD with regard to the portion of correct prediction. Apart from the ability to deliver as many correct predictions as possible in each dataset, it is crucial that the predictors can identify the true intent of an object at an early stage. To examine the algorithms in terms of their ability to achieve early intent recognition, the times at which the predictor successfully assigns the highest probability to the intended destination (i.e. $\{t_k \mid t_k \in \{t_0, t_1, \ldots, t_N = T\}, \widehat{\mathcal{D}}_k = \mathcal{D}_{\text{true}}$ $\}$) are recorded for each trajectory. Based on these timings, the average classification success against the percentage of the total travelling time ($100 \times t_n/t_N$) is plotted in Figure 5.3. According to this figure, all algorithms can make successful predictions in the first 30% of the voyage time in 60% of the cases, and this rate increases to about 90% when the vessel travels into 60% of the total time.

Furthermore, the run-times of MATLAB implementations of the three predictors are recorded for the above 100 tracks (System: *Intel(R) Core(TM) i7-4790 CPU@3.60GHz, 8GB RAM*). Algorithm 1 (mean run-time is $2.62ms$ at each t_n) shows a reduction of around 65% compared to the original BD (mean $7.42ms$); Algorithm 2 has mean run-time of $5.36ms$. This confirms the complexity analysis in (5.17)-(5.19) with parameters $\{m, s, q, N\} = \{2, 4, 15, 6\}$ and demonstrates the potential of the proposed efficient methods for real-time implementations.

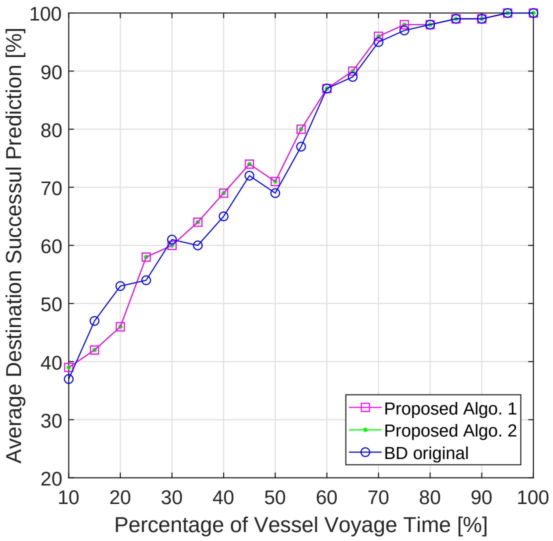


Fig. 5.3 Percentage of destination successful prediction, averaged over 100 tracks. ($\sigma = 0.2$, $\sigma_m = 0.1$)

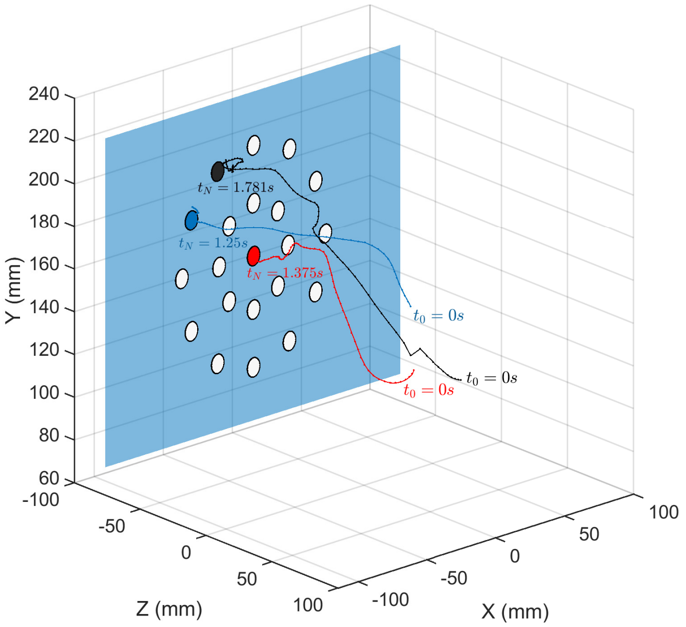


Fig. 5.4 Three examples of the recorded pointing trajectories; coloured circles are the intended destinations.

5.3.2 Freehand Pointing Data

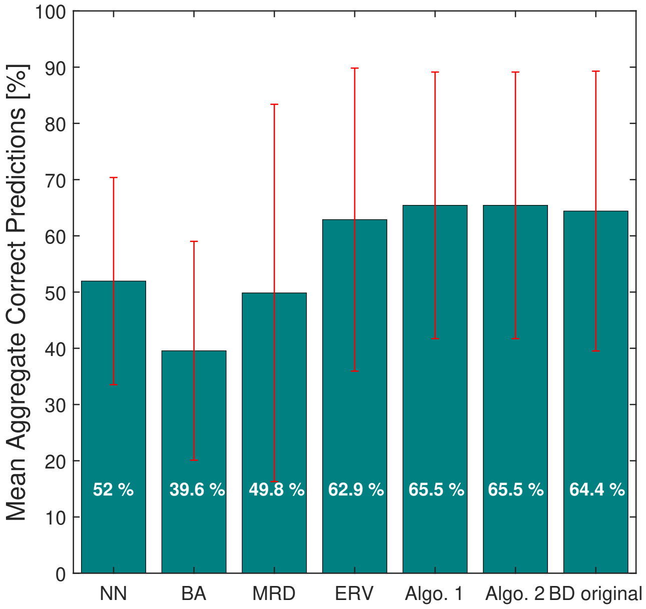
Now, we apply the three examined predictors to real data, namely 95 freehand pointing trajectories in 3-D. They were collected in an instrumented car with a vision-based gesture tracker (identical to the prototype used in [Ahmad et al., 2018; Ahmad et al., 2016a]) during driver/passenger interactions with the in-car touchscreen, at an average rate of $\approx$ 50 Hz. There are 21 selectable circular icons on the screen which are approximately 2 cm apart from each other, see Figure 5.4 for the touchscreen layout as well as three typical finger pointing trajectories captured by the gesture tracker. The objective is to predict, early in the pointing task, the intended on-display icon.

The inference performance is again assessed by the percentage of pointing time during which the true destination assigned the highest estimated probability to the intended icon $\mathcal{D}_{\text{true}}$ using MAP decision criterion; and the averaged proportion of successful predictions against that of completed pointing gesture (in time). Moreover, in order to demonstrate the performance of the proposed BD methods, their inference results are compared with those obtained from the original BD (with CV) algorithm, the OU-based intent prediction algorithms, i.e. Equilibrium Reverting Velocity (ERV) model and Mean Reverting Diffusion (MRD) model as in [Ahmad et al., 2016a], and the benchmark methods, i.e. Nearest Neighbor (NN) and Bearing Angle (BA) (see [Ahmad et al., 2016a] again for more details).

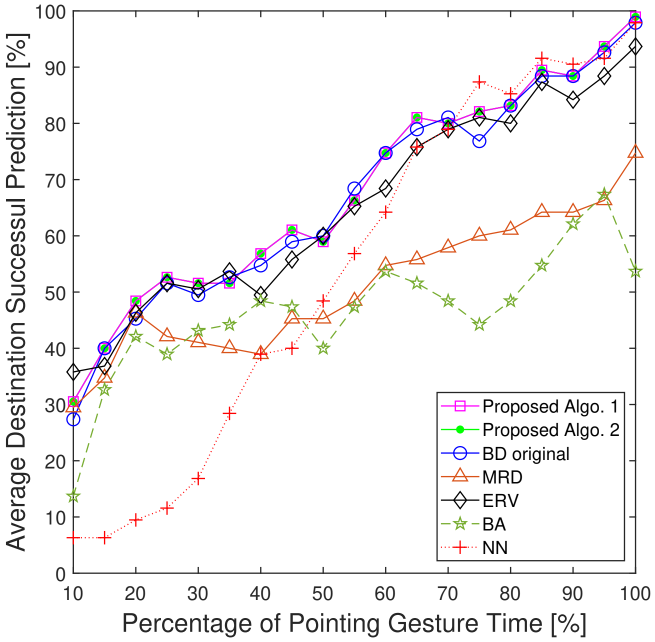
A CV model is used across all three BD algorithms, with the process noise parameter $\sigma = 0.65$ m s$^{-2/3}$ and the measurement noise parameter $\sigma_m = 3$ mm. As for the pseudo-observation on the i-th icon, $\mathbf{a}_i$ is the 3-D Cartesian position of the icon centre whereas in the covariance matrix $\Sigma_i = \text{diag}(\sigma_\xi^2 I_3,\ \sigma_{\dot{\xi}}^2 I_3)$ we set $\sigma_\xi = 1.5$ mm and $\sigma_{\dot{\xi}} = 0.1$ m s^{-1}. $q = 30$ quadrature points are placed uniformly within the time interval [0.1sec, 2sec]. Note that design parameters in the comparison algorithms have also been adjusted manually to produce the best possible results.

As shown in Figure 5.5a, the proposed predictors give the best estimates compared to the other methods in terms of the overall correct predictions. Though having a slightly lower average correction rate than the pseudo-observation based BD methods, the original BD algorithm still exceeds the OU-based and the benchmark methods. On the other hand, the average successful prediction rate against the percentage of pointing gesture time is shown in Figure 5.5b, where the bridging-distribution based methods achieve the earliest successful intent predictions most of the time. Although the NN method delivers high success rate as the finger tip is close to the screen, its performance is extremely poor at the early stage. This is not desired in time-critical applications such as the intelligent interactive display considered here. Excluding the NN method, BD-based approaches consistently outperform the others (except at 10% and around 40% the results of ERV are slightly better) and there is no significant performance difference between them. It should be emphasised that the superiority of the proposed BD formulation not only lies in its ability to achieve the highest accuracy and early predictions, but also in the fact that it significantly reduces the complexity and leads to a more computationally efficient inference procedure in comparison to the original BD.

Finally, the sensitivity of the BD methods to model mismatch is analysed. As the mismatch can arise from various sources such as dynamical model parameters, measurement model parameters and destination prior, two experiments are conducted in which the first test addresses the sensitivity to the dynamical and measurement models is examined with fixed destination prior parameters (i.e. σ_ξ and $\sigma_{\dot{\xi}}$) while the second one is concerned with the sensitivity to the destination prior (with fixed σ and σ_m). It can be seen from Figure 5.6a that reasonable changes to the CV dynamic noise, e.g. $\sigma \in [0.3\ \ 1.5]$, maintains a correct prediction rates of $60 - 65\%$ for all BD approaches when accurate measurements are available; values outside these ranges degrade the prediction accuracy since they represent more extreme deviations



(a)



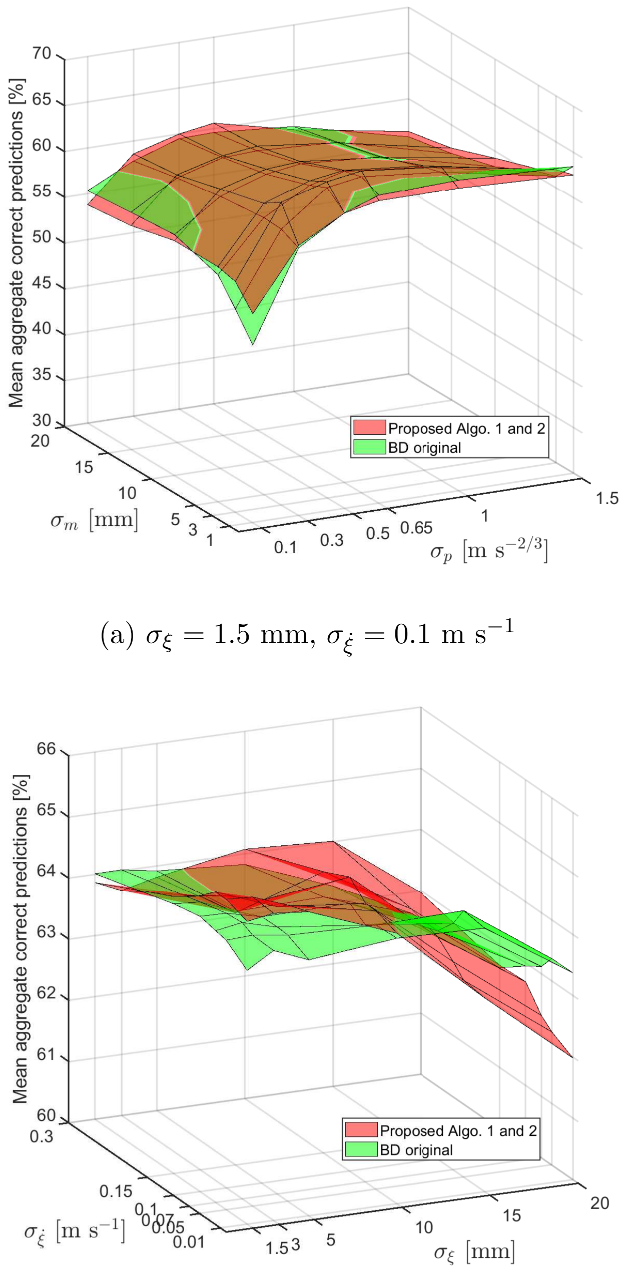
(b)

Fig. 5.5 Inference performance over 95 freehand pointing trajectories. (a) depicts the average aggregate correct predictions (error bars are one standard deviation). (b) shows the average proportion of successful predictions against the percentage of completed pointing task.

in pointing velocities. Although the performance of the algorithms decreases with increasing measurement noise level, it is observed that good prediction accuracies ($\geq 60\%$) can still be maintained when the system model matches the underlying dynamics of the finger movement. It should be noted that changing the destination information Σ_i has a very limited impact on the prediction performance, except for extreme values where the regions of all destination as captured by Σ_i overlap heavily (e.g. setting $\sigma_\xi = 20$ mm while the icons are approximately 2 cm apart.). This is depicted in Figure 5.6b. Overall, the testing results suggest that the inference framework is relatively robust against reasonable/moderate changes in the modelling parameters.

5.4 Summary and Conclusions

The proposed approach not only resolves the consistency issue with the previous bridging distributions construct but also substantially reduce the computational complexity of predicting the target destination. Similar to BD in [Ahmad et al., 2019, 2015; Ahmad et al., 2018], it however requires prior knowledge of the location of all nominal endpoints, considers Gaussian linear set-ups and treats targets singly, even if they are a group. These can be addressed by extending the BD framework in future work.



(a) $\sigma_\xi = 1.5$ mm, $\sigma_{\dot{\xi}} = 0.1$ m s^{-1}

(b) $\sigma = 0.65$ m s$^{-2/3}$, $\sigma_m = 3$ mm

Fig. 5.6 Average correct predictions over 95 pointing tracks as (a) the dynamical model (σ) and measurement model (σ_m) noise parameters are varied or (b) the destination prior parameters σ_ξ and $\sigma_{\dot{\xi}}$ are varied.

Chapter 6

Conclusions

Object tracking and intent prediction are important research topics as they meet the needs of various application domains. The ultimate goal of the former topic is to estimate kinematic states as accurately as possible under noisy and limited measurements obtained from multiple sensors, while for the latter it is to extract the intent information of a moving object efficiently and accurately. These goals were pursued in this thesis. In this chapter, we summarise and discuss the approaches and results presented earlier and provide recommendation for future work.

Chapter 3 The fixed rate modelling assumption, which leads to dynamic systems in a Markovian state space form, has been widely adopted when designing tracking algorithms. In this chapter, we stuck to this assumption and developed analytic structure for the inertial, speed and distance measurements based on a continuous-time intrinsic coordinate model. Consequently, the state space models are (conditionally) linear and Gaussian and thus allow for the construction of efficient sequential and section-wise proposal kernels which can be effectively incorporated into sequential Monte Carlo (SMC) methods. While the well-established Importance Sampling (IS) based particle filtering methods were considered for use with both the sequential and section-wise batch proposals, sequential Markov Chain Monte Carlo (SMCMC) was adopted when designing sequential batch inference algorithms since it has a strong potential to deal with high-dimensional systems. We also showed that the performance of the SMC methods can be further improved by blending MCMC moves with the IS-based SMC filters and using parallel independent chains in the SMCMC methods. The experimental results on synthetic and real data suggest that the proposed algorithms provide effective alternative solutions for tracking problems with inertial sensors and infrequent position/timing measurements.

The performance gains we obtained can be attributed to the use of a motion model capable of accurately modelling the object dynamics, the design of proposal kernels which take advantages of inertial measurements in the recent and/or more distant past and the use of the powerful SMC framework. The model, as well as the inference methods tailored to it, can serve as a guideline for others to develop their own algorithms for tracking applications with inertial sensors and degraded position signal. Also, the proposed model can be useful in describing the movement of an autonomous flight vehicle which normally travels in a horizontal plane. In order to obtain more accurate information regarding the turn rate, we could track the 3-D orientation of the object. Moreover, the promising results obtained via the use of parallel independent chains in the generic SMCMC encourage further exploration on effects of using different numbers of chains and introducing interaction between independent chains.

Chapter 4 In this chapter we investigated the use of a more realistic class of models, which removes the restriction that the state process is synchronised with the observation process, for the tracking of a manoeuvring object. We saw how the existing variable rate intrinsic model can be generalised to accommodate multi-sensor multi-rate tracking problems. The difficulty of sampling from the resulting degenerated models was overcome by adapting the simulation smoother to a variable dimensional case, allowing us to propose states effectively using Kalman-filtering-type methods. When tested on the synthetic data and a benchmark trajectory, our variable rate particle filtering algorithms achieve better tracking performance in terms of accuracy and robustness over a standard bootstrap variable rate particle filter. Furthermore, two particle Gibbs samplers were introduced to perform full Bayesian inference for the variable rate intrinsic model. Numerical results were presented to demonstrate the ability of the developed methods to infer both the latent state and system parameters.

A 3-D model for piece-wise planar motion can be readily obtained from the current model by introducing a binormal vector vertical to the plane formed by the tangential and perpendicular vector, as in [Bunch and Godsill, 2012]. As the linearity is preserved piecewisely, the efficient simulation smoother based proposal kernels can be used again and it would be interesting to see experiment results in such cases. Note that the same idea of modelling the change of motion plane using a jump process can also be applied to the Markovian state space model presented in Chapter 3. Moreover, it is worthwhile to extend the full Bayesian sampling scheme such that it can accommodate unknown jump time parameters.

Chapter 5 We turned our attention to the problem of inferring the intended destination of a moving object in a timely and accurate manner in this chapter. Instead of extending the latent state vector to incorporate the terminal state, the influence of the candidate destinations on the object state is modelled via pseudo-measurements. We have shown that this construction made the mathematics of the dynamical model and observation process consistent with the Markov state process and results in novel intent prediction algorithms which are capable of giving similar prediction performance to the state-of-art methods but with reduced computational effort. Notably, for general Gaussian linear time-invariant motion models, the proposed Algorithm 1 can at least reduce the computational complexity by approximately 57.5%, and by 65% for the non-destination-reverting models. Results on maritime surveillance and real finger pointing data revealed that the proposed methods are not only able to provide fast and accurate prediction of the object intent, they are also robust to model mismatch.

The methods can be readily extended to non-linear and/or non-Gaussian set-ups, allowing the modelling of more complicated dynamic systems and terminal points of non-ellipsoidal shapes. In such cases, particle filters can be used to approximate the sequential likelihood terms $p(\mathbf{y}_n|\mathbf{y}_{0:n-1}, \mathcal{D}_i, T)$ required for the intent prediction task. The use of numerical approximation methods will inevitably increase the computational cost and hence it can be useful to analyse different bridging distribution based methods in terms of their computation complexity in more general settings. The proposed approaches can also be applied to multiple target tracking. For example, the inference of group structure may be improved by incorporating intent information.

References

Ahmad, B. I., Ardeshiri, T., Langdon, P., Godsill, S. J., and Popham, T. (2017). Modelling received signal strength from on-vehicle BLE beacons using skewed distributions: A preliminary study. In *2017 20th International Conference on Information Fusion (FUSION)*, pages 1–7. 47

Ahmad, B. I., Langdon, P. M., and Godsill, S. J. (2019). A Bayesian framework for intent prediction in object tracking. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8439–8443. 129, 146

Ahmad, B. I., Murphy, J., Langdon, P. M., Hardy, R., and Godsill, S. J. (2015). Destination inference using bridging distributions. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5585–5589. 129, 146

Ahmad, B. I., Murphy, J. K., Godsill, S., Langdon, P., and Hardy, R. (2017). Intelligent interactive displays in vehicles with intent prediction: A Bayesian framework. *IEEE Signal Processing Magazine*, 34(2):82–94. 127

Ahmad, B. I., Murphy, J. K., Langdon, P. M., and Godsill, S. J. (2018). Bayesian intent prediction in object tracking using bridging distributions. *IEEE Transactions on Cybernetics*, 48(1):215–227. 99, 129, 130, 137, 138, 143, 146

Ahmad, B. I., Murphy, J. K., Langdon, P. M., Godsill, S. J., Hardy, R., and Skrypchuk, L. (2016a). Intent inference for hand pointing gesture-based interactions in vehicles. *IEEE Transactions on Cybernetics*, 46(4):878–889. 45, 130, 143, 144

Ahmad, M. F., Murphy, J., Vatansever, D., Stamatakis, E. A., and Godsill, S. J. (2016b). Bayesian inference of task-based functional brain connectivity using Markov chain Monte Carlo methods. *IEEE Journal of Selected Topics in Signal Processing*, 10(7):1150–1159. 13

Anderson, B. D. O. and Moore, J. B. (1979). *Optimal Filtering*. Prentice-Hall, Englewood Cliffs, NJ. 16, 30, 100

Andrieu, C., Doucet, A., and Holenstein, R. (2010). Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342. 13, 17, 34, 86, 113, 114

Andrieu, C. and Roberts, G. O. (2009). The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics*, 37(2):697–725. 13, 113

References

Ardeshiri, T., Ahmad, B. I., Langdon, P. M., and Godsill, S. J. (2017). Efficient bridging-based destination inference in object tracking. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4406–4410. IEEE. 138

Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188. 16, 18, 19

Axelsson, O. (1996). *Iterative solution methods*. Cambridge University Press. 137

Baccarelli, E. and Cusani, R. (1998). Recursive filtering and smoothing for reciprocal Gaussian processes with Dirichlet boundary conditions. *IEEE Transactions on Signal Processing*, 46(3):790–795. 127, 129

Bain, A. and Crisan, D. (2008). *Fundamentals of stochastic filtering*. Springer, New York, NY. 21

Bando, T., Takenaka, K., Nagasaka, S., and Taniguchi, T. (2013). Unsupervised drive topic finding from driving behavioral data. In *2013 IEEE Intelligent Vehicles Symposium (IV)*, pages 177–182. 127, 130

Bar-Shalom, Y. and Li, X.-R. (2001). *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc., New York, NY, USA. 45

Bar-Shalom, Y., Willett, P., and Tian, X. (2011). *Tracking and Data Fusion: A Handbook of Algorithms*. YBS Publishing. 47, 130

Beaumont, M. A. (2003). Estimation of population growth or decline in genetically monitored populations. *Genetics*, 164(3):1139–1160. 13

Berzuini, C., Best, N. G., Gilks, W. R., and Larizza, C. (1997). Dynamic conditional independence models and Markov chain Monte Carlo methods. *Journal of the American Statistical Association*, 92(440):1403–1412. 13, 34, 35, 40

Best, G. and Fitch, R. (2015). Bayesian intention inference for trajectory prediction with an unknown goal destination. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5817–5823. 127

Best, R. and Norton, J. (1997). A new model and efficient tracker for a target with curvilinear motion. *IEEE Transactions on Aerospace and Electronic Systems*, 33(3):1030–1037. 44, 46, 48

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg. 8, 58, 133

Blackman, S. and Popoli, R. (1999). *Design and analysis of modern tracking systems*. Norwood, MA: Artech House. 46

Blair, W. D., Watson, G. A., Kirubarajan, T., and Bar-Shalom, Y. (1998). Benchmark for radar allocation and tracking in ECM. *IEEE Transactions on Aerospace and Electronic Systems*, 34(4):1097–1114. 106

Blair Jr., W. P. and Sworder, D. D. (1975). Feedback control of a class of linear discrete systems with jump parameters and quadratic cost criteria. *International Journal of Control*, 21(5):833–841. 84

Bresler, Y. (1986). Two-filter formulae for discrete-time non-linear Bayesian smoothing. *International Journal of Control*, 43(2):629–641. 33

Briers, M., Doucet, A., and Maskell, S. (2009). Smoothing algorithms for state–space models. *Annals of the Institute of Statistical Mathematics*, 62(1):61. 33

Brooks, S., Gelman, A., Jones, G., and Meng, X.-L. (2011). *Handbook of Markov chain Monte Carlo*. Chapman & Hall/CRC. 12

Bunch, P. and Godsill, S. (2012). Dynamical models for tracking with the variable rate particle filter. In *2012 15th International Conference on Information Fusion*, pages 1769–1775. 46, 49, 83, 84, 106, 150

Bunch, P. and Godsill, S. (2013a). Improved particle approximations to the joint smoothing distribution using Markov chain Monte Carlo. *IEEE Transactions on Signal Processing*, 61(4):956–963. 33

Bunch, P. and Godsill, S. (2013b). Particle filtering with progressive gaussian approximations to the optimal importance density. In *2013 5th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 360–363. 24

Bunch, P. and Godsill, S. (2013c). Particle smoothing algorithms for variable rate models. *IEEE Transactions on Signal Processing*, 61(7):1663–1675. 44, 83, 87, 93, 95, 99

Bunch, P. J. (2014). Particle filtering and smoothing for challenging time series models. *PhD Thesis*. 35

Cappé, O., Godsill, S., and Moulines, E. (2007). An overview of existing methods and recent advances in sequential Monte Carlo. *Proceedings of the IEEE*, 95(5):899–924. 14, 16, 19, 22, 24, 26, 29, 95

Carey, P., Bennett, S., Lasenby, J., and Purnell, T. (2017). Aerodynamic analysis via foreground segmentation. *Electronic Imaging, Computer Vision Applications in Sports*, pages 10–14. 68

Carpenter, J., Clifford, P., and Fearnhead, P. (1999). Improved particle filter for nonlinear problems. *IEE Proceedings - Radar, Sonar and Navigation*, 146(1):2–7. 17, 21

Carter, C. and Kohn, R. (1994). On Gibbs sampling for state space models. *Biometrika*, 81(3):541–553. 33, 53, 56, 99

Casella, G. and Robert, C. P. (1996). Rao-blackwellisation of sampling schemes. *Biometrika*, 83(1):81–94. 28, 36

Castanon, D. A., Levy, B. C., and Willsky, A. S. (1985). Algorithms for the incorporation of predictive information in surveillance theory. *International Journal of Systems Science*, 16(3):367–382. 127, 129

Chiang, H., Rackley, N., and Tapia, L. (2015). Stochastic ensemble simulation motion planning in stochastic dynamic environments. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3836–3843. 127, 130

Chib, S. (1995). Marginal Likelihood from the Gibbs Output. *Journal of the American Statistical Association*, 90(432):1313–1321. 83

Chopin, N. and Singh, S. S. (2015). On particle gibbs sampling. *Bernoulli*, 21(3):1855–1883. 115

Christensen, H. L., Murphy, J., and Godsill, S. J. (2012). Forecasting high-frequency futures returns using online Langevin dynamics. *IEEE Journal of Selected Topics in Signal Processing*, 6(4):366–380. 15, 31, 83, 84

Costa, O. L., Fragoso, M. D., and Todorov, M. G. (2013). *Continuous-Time Markov Jump Linear Systems*. Springer Berlin Heidelberg, Berlin, Heidelberg. 84

Crisan, D. and Doucet, A. (2002). A survey of convergence results on particle filtering methods for practitioners. *IEEE Transactions on Signal Processing*, 50(3):736–746. 36

d'Afflisio, E., Braca, P., Millefiori, L. M., and Willett, P. (2018). Maritime anomaly detection based on mean-reverting stochastic processes applied to a real-world scenario. In *2018 21st International Conference on Information Fusion (FUSION)*, pages 1171–1177. 127, 129

Davey, S., Gordon, N., Holland, I., Rutten, M., and Williams, J. (2016). *Bayesian Methods in the Search for MH370*. Springer Singerpore. 83

Davis, P. J. and Rabinowitz, P. (2007). *Methods of numerical integration*. Courier Corporation. 132

de Jong, P. and Shephard, N. (1995). The simulation smoother for time series models. *Biometrika*, 82(2):339–350. 86, 99

Doran, H. E. (1992). Constraining Kalman Filter and Smoothing Estimates to Satisfy Time-Varying Restrictions. *The Review of Economics and Statistics*, 74(3):568–572. 130

Douc, R., Cappé, O., and Moulines, E. (2005). Comparison of resampling schemes for particle filtering. In *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005.*, pages 64–69. 21

Douc, R., Garivier, A., Moulines, E., and Olsson, J. (2011). Sequential Monte Carlo smoothing for general state space hidden Markov models. *The Annals of Applied Probability*, 21(6):2109–2145. 33

Doucet, A., de Freitas, N., and Gordon, N. (2001). *Sequential Monte Carlo Methods in Practice*. Springer-Verlag New York. 19, 95

Doucet, A., De Freitas, N., Murphy, K., and Russell, S. (2000a). Rao-Blackwellised particle filtering for dynamic Bayesian networks. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pages 176–183. Morgan Kaufmann Publishers Inc. 17

Doucet, A., Godsill, S., and Andrieu, C. (2000b). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208. 14, 17, 18, 19, 20, 23, 24, 26, 28, 33, 53

Doucet, A. and Johansen, A. M. (2011). A tutorial on particle filtering and smoothing: Fifteen years later. 16, 19

Duan, Z. and Li, X. R. (2013). The role of pseudo measurements in equality-constrained state estimation. *IEEE Transactions on Aerospace and Electronic Systems*, 49(3):1654–1666. 127, 130

Fanaswala, M. and Krishnamurthy, V. (2015). Spatiotemporal trajectory models for metalevel target tracking. *IEEE Aerospace and Electronic Systems Magazine*, 30(1):16–31. 130

Fearnhead, P., Wyncoll, D., and Tawn, J. (2010). A sequential smoothing algorithm with linear computational cost. *Biometrika*, 97(2):447–464. 33

Fernández-Villaverde, J. and Rubio-Ramírez, J. F. (2007). Estimating macroeconomic models: A likelihood approach. *The Review of Economic Studies*, 74(4):1059–1087. 13

Frühwirth-Schnatter, S. (1994). Data augmentation and dynamic linear models. *Journal of Time Series Analysis*, 15(2):183–202. 33, 53, 56, 99

Gan, R., Liang, J., Ahmad, B. I., and Godsill, S. (2019). Bayesian intent prediction for fast maneuvering objects using variable rate particle filters. In *2019 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*. 130

Gelfand, A. E. and Smith, A. F. M. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85(410):398–409. 12

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741. 12

Gerber, M., Chopin, N., and Whiteley, N. (2019). Negative association, ordering and convergence of resampling methods. *The Annals of Statistics*, 47(4):2236–2260. 21

Geweke, J. (1989). Bayesian inference in econometric models using Monte Carlo integration. *Econometrica*, 57(6):1317–39. 9

Gilholm, K., Godsill, S., Maskell, S., and Salmond, D. (2005). Poisson models for extended target and group tracking. In *Signal and Data Processing of Small Targets 2005*, volume 5913, page 59130R. International Society for Optics and Photonics. 46, 48, 85, 88

Gilks, W. and Berzuini, C. (2001). Following a moving target — Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 63(1):127–146. 13, 17, 34, 36, 59, 60, 103, 121

Gilks, W. R., Richardson, S., and Spiegelhalter, D. (1995). *Markov chain Monte Carlo in practice*. Chapman and Hall/CRC. 11

Godsill, S. (2007). Particle filters for continuous-time jump models in tracking applications. *ESAIM: Proc.*, 19:39–52. 45

Godsill, S. (2019a). *Bayesian Computational Methods for Statistical Signal Processing*. Unpublished Research Notes. 8

Godsill, S. (2019b). Particle filtering: the First 25 years and beyond. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7760–7764. 19, 39, 95

Godsill, S. and Clapp, T. (2001). Improvement strategies for Monte Carlo particle filters. In *Sequential Monte Carlo methods in practice*, pages 139–158. Springer. 17, 24, 25, 34

Godsill, S. and Vermaak, J. (2004). Models and algorithms for tracking using trans-dimensional sequential Monte Carlo. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages iii–976. 44, 83, 85, 95

Godsill, S. and Vermaak, J. (2005). Variable rate particle filters for tracking applications. In *IEEE/SP 13th Workshop on Statistical Signal Processing, 2005*, pages 1280–1285. 44, 46, 48, 84

Godsill, S. J. (2001). On the relationship between markov chain monte carlo methods for model uncertainty. *Journal of Computational and Graphical Statistics*, 10(2):230–248. 83

Godsill, S. J., Doucet, A., and West, M. (2004). Monte carlo smoothing for nonlinear time series. *Journal of the American Statistical Association*, 99(465):156–168. 14, 33

Godsill, S. J., Vermaak, J., Ng, W., and Li, J. F. (2007). Models and algorithms for tracking of maneuvering objects using variable rate particle filters. *Proceedings of the IEEE*, 95(5):925–952. 15, 17, 21, 44, 46, 48, 83, 84, 85, 87, 88, 91, 95, 97, 98, 106, 119

Golightly, A. and Wilkinson, D. J. (2006). Bayesian sequential inference for nonlinear multivariate diffusions. *Statistics and Computing*, 16(4):323–338. 13, 37, 38

Gordon, N., Salmond, D., and Smith, A. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings F - Radar and Signal Processing*, pages 107–113. IET. 2, 17, 21, 95, 97

Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732. 83, 86, 103

Greenwood, D. T. (2006). *Advanced dynamics*. Cambridge University Press. 87

Gustafsson, F. (2000). *Adaptive filtering and change detection*. John Wiley & Sons, Ltd. 46

Gustafsson, F., Gunnarsson, F., Bergman, N., Forssell, U., Jansson, J., Karlsson, R., and Nordlund, P. . (2002). Particle filters for positioning, navigation, and tracking. *IEEE Transactions on Signal Processing*, 50(2):425–437. 47, 99

Handschin, J. E. and Mayne, D. Q. (1969). Monte Carlo techniques to estimate the conditional expectation in multi-stage non-linear filtering. *International Journal of Control*, 9(5):547–559. 22

Harle, R. (2013). A survey of indoor inertial positioning systems for pedestrians. *IEEE Communications Surveys Tutorials*, 15(3):1281–1293. 47

Harvey, A. C. (1989). *Forecasting, structural time series models and the Kalman filter*. Cambridge University Press. 58, 102, 132

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109. 11

Haug, A. J. (2012). *Bayesian Estimation and Tracking: A Practical Guide*. John Wiley & Sons. 130, 133

Ho, Y. and Lee, R. (1964). A Bayesian approach to problems in stochastic estimation and control. *IEEE Transactions on Automatic Control*, 9(4):333–339. 16, 58

Hostettler, R. and Djurić, P. M. (2015). Vehicle tracking based on fusion of magnetometer and accelerometer sensor measurements with particle filtering. *IEEE Transactions on Vehicular Technology*, 64(11):4917–4928. 85

Hostettler, R. and Särkkä, S. (2016). IMU and magnetometer modeling for smartphone-based PDR. In *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8. 44, 46

Hostettler, R. and Särkkä, S. (2019). Rao–Blackwellized Gaussian smoothing. *IEEE Transactions on Automatic Control*, 64(1):305–312. 29

Huang, C., Liao, Z., and Zhao, L. (2010). Synergism of INS and PDR in self-contained pedestrian tracking with a miniature sensor module. *IEEE Sensors Journal*, 10(8):1349–1359. 91

Hürzeler, M. and Künsch, H. R. (1998). Monte Carlo approximations for general state-space models. *Journal of Computational and Graphical Statistics*, 7(2):175–193. 17, 33

Hürzeler, M. and Künsch, H. R. (2001). Approximating and maximising the likelihood for a general state-space model. In *Sequential Monte Carlo methods in practice*, pages 159–175. Springer. 33

Jacobsen, M. (2006). *Point process theory and applications: Marked point and piecewise deterministic processes*. Boston: Birkhäuser, MA. 86, 87

Johannes, M. S., Polson, N. G., and Stroud, J. R. (2009). Optimal filtering of jump diffusions: Extracting latent states from asset prices. *The Review of Financial Studies*, 22(7):2759–2799. 83

Julier, S. J. and Uhlmann, J. K. (1997). New extension of the Kalman filter to nonlinear systems. In *Signal processing, sensor fusion, and target recognition VI*, volume 3068, pages 182–193. International Society for Optics and Photonics. 17

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, 82(1):35–45. 58

Karlsson, R., Schon, T., and Gustafsson, F. (2005). Complexity analysis of the marginalized particle filter. *IEEE Transactions on Signal Processing*, 53(11):4408–4411. 32

Kastella, K. and Biscuso, M. (1995). Tracking algorithms for air traffic control applications. *Air Traffic Control Quarterly*, 3(1):19–43. 46

Khan, Z., Balch, T., and Dellaert, F. (2004). An MCMC-based particle filter for tracking multiple interacting targets. In Pajdla, T. and Matas, J., editors, *Computer Vision - ECCV 2004*, pages 279–290, Berlin, Heidelberg. Springer Berlin Heidelberg. 36

Khan, Z., Balch, T., and Dellaert, F. (2005). MCMC-based particle filtering for tracking a variable number of interacting targets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1805–1819. 36, 38

Kitagawa, G. (1987). Non-Gaussian state-space modeling of nonstationary time series. *Journal of the American Statistical Association*, 82(400):1032–1041. 32

Kitagawa, G. (1994). The two-filter formula for smoothing and an implementation of the Gaussian-sum smoother. *Annals of the Institute of Statistical Mathematics*, 46(4):605–623. 33

Kitagawa, G. (1996). Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5(1):1–25. 17, 20, 21, 32, 64, 106

Kitani, K. M., Ziebart, B. D., Bagnell, J. A., and Hebert, M. (2012). Activity forecasting. In Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., and Schmid, C., editors, *Computer Vision – ECCV 2012*, pages 201–214, Berlin, Heidelberg. Springer Berlin Heidelberg. 127, 130

Klein, G. and Drummond, T. (2004). Tightly integrated sensor fusion for robust visual tracking. *Image and Vision Comput.*, 22(10):769 – 776. 91

Kong, A. (1992). A note on importance sampling using standardized weights *Technical Report 348*, Department of Statistics, University of Chicago. 20

Kong, A., Liu, J. S., and Wong, W. H. (1994). Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association*, 89(425):278–288. 20

Koopman, S. J. (1993). Disturbance smoother for state space models. *Biometrika*, 80(1):117–126. 99

Krishnamurthy, V. and Gao, S. (2018). Syntactic enhancement to VSIMM for roadmap based anomalous trajectory detection: A natural language processing approach. *IEEE Transactions on Signal Processing*, 66(20):5212–5227. 127, 130

Künsch, H. R. (2013). Particle filters. *Bernoulli*, 19(4):1391–1403. 19

Li, C. and Godsill, S. J. (2018). Sequential inference methods for non-homogeneous poisson processes with state-space prior. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2856–2860. 13

Li, X. R. and Jilkov, V. (2003). Survey of maneuvering target tracking. Part I. Dynamic models. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1333–1364. 44, 45, 46, 48, 49, 99

Liang, J., Ahmad, B. I., Gan, R., Langdon, P., Hardy, R., and Godsill, S. (2019). On destination prediction based on Markov bridging distributions. *IEEE Signal Processing Letters*, 26(11):1663–1667. 130

Liang, J. and Godsill, S. (2018). A particle filter localisation system for indoor track cycling using an intrinsic coordinate model. In *2018 21st International Conference on Information Fusion (FUSION)*, pages 1896–1903. 24, 44

Liang, J. and Godsill, S. (2019). Bayesian fusion of asynchronous inertial, speed and position data for object tracking. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6016–6020. 13, 44

Lin, M., Chen, R., and Liu, J. S. (2013). Lookahead strategies for sequential Monte Carlo. *Statistical Science*, 28(1):69–94. 27, 97

Lindsten, F. (2011). *Rao-Blackwellised particle methods for inference and identification.* PhD thesis, Linköping University Electronic Press. 31

Lindsten, F., Bunch, P., Särkkä, S., Schön, T. B., and Godsill, S. J. (2016). Rao-Blackwellized particle smoothers for conditionally linear Gaussian models. *IEEE Journal of Selected Topics in Signal Processing*, 10(2):353–365. 29, 33, 106

Liu, H., Darabi, H., Banerjee, P., and Liu, J. (2007). Survey of wireless indoor positioning techniques and systems. *IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews*, 37(6):1067–1080. 47

Liu, J. S. (2004). *Monte Carlo strategies in scientific computing.* Springer-Verlag New York. 8, 9, 24

Liu, J. S. and Chen, R. (1998). Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93(443):1032–1044. 14, 17, 18, 21, 28

MacEachern, S. N., Clyde, M., and Liu, J. S. (1999). Sequential importance sampling for nonparametric Bayes models: The next generation. *The Canadian Journal of Statistics / La Revue Canadienne de Statistique*, 27(2):251–267. 28, 34, 35, 103

Marques, R., Silva, W. d., Hoffmann, R., Dutra, J., and Coral, F. (2018). Sequential state inference of engineering systems through the particle move-reweighting algorithm. *Computational and Applied Mathematics*, 37(1):220–236. 34, 35

Maskell, S. (2004). *Sequentially Structured Bayesian Solutions*. University of Cambridge. 46, 99

Masoumi-Ganjgah, F., Fatemi-Mofrad, R., and Ghadimi, N. (2017). Target tracking with fast adaptive revisit time based on steady state IMM filter. *Digital Signal Processing*, 69:154 – 161. 106

Mazor, E., Averbuch, A., Bar-Shalom, Y., and Dayan, J. (1998). Interacting multiple model methods in target tracking: A survey. *IEEE Transactions on Aerospace and Electronic Systems*, 34(1):103–123. 84

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092. 11

Middleton, L., Deligiannidis, G., Doucet, A., and Jacob, P. E. (2019). Unbiased smoothing using particle independent Metropolis-Hastings. In Chaudhuri, K. and Sugiyama, M., editors, *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*, pages 2378–2387. PMLR. 34

Mihaylova, L., Carmi, A. Y., Septier, F., Gning, A., Pang, S. K., and Godsill, S. (2014). Overview of Bayesian sequential Monte Carlo methods for group and extended object tracking. *Digital Signal Processing*, 25:1 – 16. 38, 40, 47

Millefiori, L. M., Braca, P., Bryan, K., and Willett, P. (2016). Modeling vessel kinematics using a stochastic mean-reverting process for long-term prediction. *IEEE Transactions on Aerospace and Electronic Systems*, 52(5):2313–2330. 127, 129

Miller, J. W. (2019). Fast and accurate approximation of the full conditional for gamma shape parameters. *Journal of Computational and Graphical Statistics*, 28(2):476–480. 122

Moral, P. D., Doucet, A., and Jasra, A. (2006). Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 68(3):411–436. 17, 34, 84

Morelande, M. R. and Gordon, N. (2009). Rao-Blackwellised variable rate particle filters. In *2009 12th International Conference on Information Fusion*, pages 1–8. 31, 32, 95

Müller, P. (1991). A generic approach to posterior integration and Gibbs sampling. Technical report, Department of Statistics, Purdue University. 36

Müller, P. (1993). Alternatives to the Gibbs sampling scheme. Technical report, Institute of Statistics and Decision Science, Duke University. 36

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. The MIT Press. 8

Nurminen, H., Ardeshiri, T., Piché, R., and Gustafsson, F. (2015a). A NLOS-robust TOA positioning filter based on a skew-t measurement noise model. In *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–7. 47

Nurminen, H., Ardeshiri, T., Piché, R., and Gustafsson, F. (2015b). Robust inference for state-space models with skewed measurement noise. *IEEE Signal Processing Letters*, 22(11):1898–1902. 47

Page, E. S. (1954). Continuous inspection schemes. *Biometrika*, 41(1/2):100–115. 83

Pang, S. K., Godsill, S. J., Li, J., Septier, F., and Hill, S. (2011a). Sequential inference for evolving groups of objects. In *Bayesian Time Series Models*, pages 245–276. Cambridge University Press. 38, 39, 40

Pang, S. K., Li, J., and Godsill, S. J. (2008). Models and algorithms for detection and tracking of coordinated groups. In *2008 IEEE Aerospace Conference*, pages 1–17. 38

Pang, S. K., Li, J., and Godsill, S. J. (2011b). Detection and tracking of coordinated groups. *IEEE Transactions on Aerospace and Electronic Systems*, 47(1):472–502. 13, 17, 25, 38, 40, 45

Pilté, M., Bonnabel, S., and Barbaresco, F. (2017). Tracking the Frenet-Serret frame associated to a highly maneuvering target in 3D. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 1969–1974. 46

Pitt, M. K., dos Santos Silva, R., Giordani, P., and Kohn, R. (2012). On some properties of Markov chain Monte Carlo simulation methods based on the particle filter. *Journal of Econometrics*, 171(2):134 – 151. Bayesian Models, Methods and Applications. 13

Pitt, M. K. and Shephard, N. (1999). Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599. 13, 17, 24, 25, 36, 38, 96

Rauch, H. E., Striebel, C. T., and Tung, F. (1965). Maximum likelihood estimates of linear dynamic systems. *AIAA Journal*, 3(8):1445–1450. 32, 135

Riabiz, M., Lindsten, F., and Godsill, S. (2015). Pseudo-marginal MCMC for parameter estimation in α-stable distributions. *IFAC-PapersOnLine*, 48(28):472 – 477. 17th IFAC Symposium on System Identification SYSID 2015. 13

Ristic, B., Arulampalam, S., and Gordon, N. (2003). *Beyond the Kalman filter: Particle filters for tracking applications*. Artech house. 47

Robert, C. P. and Casella, G. (2004). *Monte Carlo statistical methods*. New York: Springer. 9, 28, 31, 60, 91

Roth, M., Hendeby, G., and Gustafsson, F. (2014). EKF/UKF maneuvering target tracking using coordinated turn models with polar/Cartesian velocity. In *17th International Conference on Information Fusion (FUSION)*, pages 1–8. 46, 106

Rubin, D. B. (1987). Comment. *Journal of the American Statistical Association*, 82(398):543–546. 20

Safari, S., Shabani, F., and Simon, D. (2014). Multirate multisensor data fusion for linear systems using Kalman filters and a neural network. *Aerospace Science and Technology*, 39:465 – 471. 85

Särkkä, S. (2013). *Bayesian filtering and smoothing*. Cambridge University Press. 14, 16, 32, 44

Schön, T., Gustafsson, F., and Nordlund, P.-J. (2005). Marginalized particle filters for mixed linear/nonlinear state-space models. *IEEE Transactions on Signal Processing*, 53(7):2279–2289. 29

Schön, T. B., Törnqvist, D., and Gustafsson, F. (2007). Fast particle filters for multirate sensors. In *2007 15th European Signal Processing Conference*, pages 876–880. 85

Seah, C. E. and Hwang, I. (2011). Algorithm for performance analysis of the IMM algorithm. *IEEE Transactions on Aerospace and Electronic Systems*, 47(2):1114–1124. 106

Septier, F., Pang, S. K., Carmi, A., and Godsill, S. (2009). On MCMC-based particle methods for Bayesian filtering: Application to multitarget tracking. In *2009 3rd IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 360–363. 36, 40

Septier, F. and Peters, G. W. (2016). Langevin and Hamiltonian based sequential MCMC for efficient Bayesian filtering in high-dimensional spaces. *IEEE Journal of Selected Topics in Signal Processing*, 10(2):312–327. 13, 17, 25, 38, 47

Shi, P. and Li, F. (2015). A survey on Markovian jump systems: Modeling and design. *International Journal of Control, Automation and Systems*, 13(1):1–16. 84

Snyder, C., Bengtsson, T., Bickel, P., and Anderson, J. (2008). Obstacles to high-dimensional particle filtering. *Monthly Weather Review*, 136(12):4629–4640. 28

Solin, A., Cortes, S., Rahtu, E., and Kannala, J. (2018). Inertial odometry on handheld smartphones. In *2018 21st International Conference on Information Fusion (FUSION)*, pages 1–5. 91

Son, Y. S. and Oh, M. (2006). Bayesian estimation of the two-parameter gamma distribution. *Communications in Statistics - Simulation and Computation*, 35(2):285–293. 122

Stiller, A. G. P. L. C. and Urtasun, R. (2013). Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11):1231–1237. 74

Särkkä, S. and Solin, A. (2019). *Applied Stochastic Differential Equations*. Institute of Mathematical Statistics Textbooks. Cambridge University Press. 128

Tierney, L. (1994). Markov Chains for exploring posterior distributions. *The Annals of Statistics*, 22(4):1701–1728. 38

Uney, M., Millefiori, L. M., and Braca, P. (2018). Prediction of rendezvous in maritime situational awareness. In *2018 21st International Conference on Information Fusion (FUSION)*, pages 622–628. IEEE. 127, 129

Völz, B., Mielenz, H., Gilitschenski, I., Siegwart, R., and Nieto, J. (2018). Inferring pedestrian motions at urban crosswalks. *IEEE Transactions on Intelligent Transportation Systems*. 127, 130

Wan, E. A. and Van Der Merwe, R. (2000). The unscented Kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, pages 153–158. 17

West, M. and Harrison, J. (1997). *Bayesian forecasting and dynamic models*. New York: Springer-Verlag, 2nd edition. 14

Whiteley, N. and Johansen, A. M. (2010). Recent developments in auxiliary particle filtering. *Inference and learning in dynamic models*, 38:39–47. 25

Whiteley, N., Johansen, A. M., and Godsill, S. (2011). Monte Carlo filtering of piecewise deterministic processes. *Journal of Computational and Graphical Statistics*, 20(1):119–139. 83, 84, 87

Willner, D., Chang, C.-B., and Dunn, K.-P. (1976). Kalman filter configurations for multiple radar systems. Technical report, Massachusetts Inst. of Tech. Lexington Lincoln Lab. 137

Xie, G., Gao, H., Qian, L., Huang, B., Li, K., and Wang, J. (2018). Vehicle trajectory prediction by integrating physics- and maneuver-based approaches using interactive multiple models. *IEEE Transactions on Industrial Electronics*, 65(7):5999–6008. 84

Zhang, G. and Godsill, S. (2016). Fundamental frequency estimation in speech signals with variable rate particle filters. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(5):890–900. 31, 84

Zhang, H., Li, L., and Xie, W. (2018). Constrained multiple model particle filtering for bearings-only maneuvering target tracking. *IEEE Access*, 6:51721–51734. 99

Zhou, G., Li, K., and Kirubarajan, T. (2020). Constrained state estimation using noisy destination information. *Signal Processing*, 166:107226. 127, 130

Zhu, G., Zhou, F., Xie, L., Jiang, R., and Chen, Y. (2014). Sequential asynchronous filters for target tracking in wireless sensor networks. *IEEE Sensors Journal*, 14(9):3174–3182. 85

Øksendal, B. (2003). *Stochastic Differential Equations (6th Ed.): An Introduction with Applications*. Springer-Verlag, Berlin, Heidelberg. 44, 128