JYU DISSERTATIONS 572

Santeri Karppinen

# Non-linear State-space Methods for Bayesian Time Series Modelling

UNIVERSITY OF JYVÄSKYLÄ
FACULTY OF MATHEMATICS
AND SCIENCE

JYU DISSERTATIONS 572

Santeri Karppinen

# Non-linear State-space Methods for Bayesian Time Series Modelling

Esitetään Jyväskylän yliopiston matemaattis-luonnontieteellisen tiedekunnan suostumuksella
julkisesti tarkastettavaksi yliopiston vanhassa juhlasalissa S212
joulukuun 10. päivänä 2022 kello 12.

Academic dissertation to be publicly discussed, by permission of
the Faculty of Mathematics and Science of the University of Jyväskylä,
in building Seminarium, Old Festival Hall S212, on December 10, 2022, at 12 o'clock.

JYVÄSKYLÄN YLIOPISTO
UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2022

Editors
Matti Vihola
Department of Mathematics and Statistics, University of Jyväskylä
Päivi Vuorio
Open Science Centre, University of Jyväskylä

ISBN 978-951-39-9226-2 (PDF)
URN:ISBN:978-951-39-9226-2
ISSN 2489-9003

Permanent link to this publication: http://urn.fi/URN:ISBN:978-951-39-9226-2

# ABSTRACT

Karppinen, Santeri
Non-linear state-space methods for Bayesian time series modelling
Jyväskylä: University of Jyväskylä, 2022, 51 p. (+ included articles)
(JYU Dissertations
ISSN 2489-9003; 572)
ISBN 978-951-39-9226-2 (PDF)

State-space methods are used in many fields of science to solve so called filtering, smoothing, prediction and parameter inference problems using multivariate time series data. Analytical solutions to these inference problems exist mainly for linear Gaussian state-space models and discrete state-space models. Outside these special cases, the inference is typically based on approximate methods, or simulation-based methods such as particle filters.

This thesis develops new methods for Bayesian inference of general state-space models and applies existing methods in challenging non-linear problems involving multivariate time series data. The new methods presented in this thesis are conditional particle filters that are relevant for the inference of models that involve uninformative initial state distributions and models that have slowly-mixing state dynamics and/or weakly informative observation processes. The applied problems develop new non-linear state-space models in order to solve a prediction problem related to childhood acute lymphoblastic leukaemia and a filtering problem related to the identification of wolf territories based on presence-only citizen science data.

Keywords: non-linear, state-space model, particle filter, Bayesian inference, Feynman–Kac model, Markov chain Monte Carlo, sequential Monte Carlo

# TIIVISTELMÄ (ABSTRACT IN FINNISH)

Karppinen, Santeri
Epälineaarisia tila-avaruusmenetelmiä Bayes-aikasarjamallinnukseen
Jyväskylä: Jyväskylän yliopisto, 2022, 51 s. (+ artikkelit)
(JYU Dissertations
ISSN 2489-9003; 572)
ISBN 978-951-39-9226-2 (PDF)

Tila-avaruusmenetelmiä ja moniulotteisia aikasarjoja käytetään useilla tieteenaloilla parametripäättelyyn sekä niin kutsuttujen suodatus-, tasoitus- ja ennustusongelmien ratkaisuun. Näihin tilastollisen päättelyn ongelmiin on olemassa analyyttiset ratkaisut pääasiassa lineaaristen sekä diskreettien tila-avaruusmallien tapauksessa. Näiden erikoistapausten ulkopuolella tila-avaruusmalleihin liittyvässä tilastollisessa päättelyssä käytetään yleensä likiarvoisia menetelmiä tai simulointimenetelmiä, kuten niin kutsuttuja hiukkassuodatusalgoritmeja.

Tässä väitöskirjassa kehitetään uusia Bayes-päättelyn menetelmiä yleisille tila-avaruusmalleille, ja sovelletaan jo olemassa olevia tila-avaruusmenetelmiä haastaviin moniulotteisiin aikasarjoihin ja epälineaarisiin mallinnusongelmiin. Väitöskirjassa kehitetyt uudet menetelmät ovat niin kutsuttuja ehdollisia hiukkassuodattimia. Menetelmät soveltuvat erityisesti tila-avaruusmalleille, joiden alkutilan todennäköisyysjakauma on epäinformatiivinen, ja tila-avaruusmalleille, joiden tiladynamiikka on hitaasti sekoittuva tai joiden havaintoprosessit ovat epäinformatiivisia. Soveltavissa mallinnusongelmissa tarkastellaan ennustusongelmaa, joka liittyy lääkityksen säätöön lasten akuutin lymfoblastileukemian hoidossa, sekä suodatusongelmaa, jossa suomalaisia susireviirejä paikannetaan kansalaishavaintojen perusteella.

Avainsanat: epälineaarinen, tila-avaruusmalli, hiukkassuodatin, Bayes-päättely, Feynman–Kac malli, Markovin ketju Monte Carlo, sekventiaalinen Monte Carlo

**Author**

Santeri Karppinen
Department of Mathematics and Statistics
University of Jyväskylä
santeri.j.karppinen@jyu.fi
ORCID: 0000-0002-4578-3147

**Supervisor**

Professor Matti Vihola
Department of Mathematics and Statistics
University of Jyväskylä

**Reviewers**

Professor Paul Fearnhead
Department of Mathematics and Statistics
Lancaster University

Dr. Kari Heine
Department of Mathematical Sciences
University of Bath

**Opponent**

Professor Nick Whiteley
School of Mathematics
University of Bristol

# ACKNOWLEDGEMENTS

I want to thank my supervisor, professor Matti Vihola for both encouraging me to pursue a PhD and patiently guiding me through it. Matti has taught me a great deal about statistics and mathematics and has profoundly affected the way I think about and approach statistical questions. In our collaboration, I have especially enjoyed the focus on the computational and algorithmic aspects of our research.

I must also thank Matti for extensive funding through the Academy of Finland research grant 315619 (Scalable methods for reliable Bayesian inference). Thanks also for the Department of Mathematics and Statistics for funding at the end of my PhD. It has been a privilege to not have to worry about funding and to be able to focus on research solely.

I wish to express my gratitude for Prof. Nick Whiteley for serving as the opponent, and to my examiners, Prof. Paul Fearnhead and Dr. Kari Heine. Thank you for taking the time to read and examine my thesis, and for your comments.

The research in this thesis would not have been done without the help of my co-authors: Prof. Sumeetpal Singh, Dr. Tuomas Rajala, Dr. (docent) Samu Mäntyniemi, Prof. Ilpo Kojola and Dr. (docent) Olli Lohi. Thank you for the collaboration, I have enjoyed working with you. I would also like to thank CSC, IT Center for Science, for computational resources.

My colleagues at the Department of Mathematics and Statistics also deserve a thanks. You have taught me a lot and we have had many interesting discussions over lunch and coffee that I have thoroughly enjoyed. Thanks also for setting up a friendly and supportive atmosphere at the department.

Finally, I wish to thank my friends and family for extensive support throughout my studies.

Santeri Karppinen
Jyväskylä, December 2022

# CONTENTS

## LIST OF INCLUDED ARTICLES

This thesis consists of an introductory part and the articles listed below.

I Karppinen, S., Lohi, O., and Vihola, M. Prediction of leukocyte counts during paediatric acute lymphoblastic leukaemia maintenance therapy. *Scientific Reports, 9.1, pp. 1–11,* DOI: https://doi.org/10.1038/s41598-019-54492-5, 2019.

II Karppinen, S., and Vihola, M. Conditional particle filters with diffuse initial distributions. *Statistics and Computing, 31.3, pp. 1–14,* DOI: https://doi.org/10.1007/s11222-020-09975-1, 2021.

III Karppinen, S., Rajala, T., Mäntyniemi, S., Kojola, I., and Vihola, M. Identifying territories using presence-only citizen science data: an application to the Finnish wolf population. *Ecological Modelling, 472*, DOI: https://doi.org/10.1016/j.ecolmodel.2022.110101, 2022.

IV Karppinen, S., Singh, S.S., and Vihola, M. Conditional particle filters with bridge backward sampling. *Preprint*, URL: https://arxiv.org/pdf/2205.13898.pdf, 2022.

The author of this thesis is the corresponding author of Articles I–IV. The author was solely responsible for the implementation of the models and methods, carrying out the experiments, and producing all visualisations and tables in Articles I–IV, with the exception of the intensity analysis in Article III. Together with coauthors, the author actively contributed to the writing of Articles I–IV and to the development of the methods, formulation of the research questions, experiments and models in Articles I–IV.

Article I is a follow-up to the master's thesis of the author [Karppinen 2018].

# 1 INTRODUCTION

The classical example of a time series is that of a financial series of daily stock prices. In fact, these kinds of univariate time series are so common that it is easy to treat them as definitions for time series data. In general, however, data observed in time may be much richer than this. For example, when the movement of an object is tracked in time using GPS, the resulting time series is two-dimensional with observations typically made at irregular time intervals. It is therefore more accurate to say that the defining feature of time series data is the dependence between consecutive observations, which manifests as serial correlation in time.

State-space models (SSMs) are statistical models commonly used for the analysis of multivariate time series data [cf. Durbin and Koopman 2012]. SSMs consist of an unobserved latent state process and an observation process that is assumed to generate the observed time series. The latent process and the observation process are linked such that the observation process depends on the latent process.

This structure of SSMs provides a convenient means of accounting for the dependence in time series data and allows for modelling the known (the data) conditional on the unknown (the latent states). Indeed, applications of SSMs often define the latent state process such that its interpretation and dynamics resemble a process of interest. It is then natural to construct the observation process given the values of the state variables, and to model the data as a realisation of the observation process.

Perhaps due to their convenient structure, SSMs find a plethora of applications in diverse fields such as ecology [Wood 2010; Johnson et al. 2008], environmental sciences [Helske et al. 2013], epidemiology [Rasmussen, Ratmann, and Koelle 2011], genetics [Mirauta, Nicolas, and Richard 2014], GPS positioning [Caron et al. 2007], and multi-target tracking [Vo, Singh, and Doucet 2003; Särkkä, Vehtari, and Lampinen 2007; Vihola 2007], to name a few.

The central inference problem related to SSMs is the computation of probability distributions of the latent states given the observed time series. In the context of Bayesian statistics, such distributions are called posterior distributions, and the probabilistic information contained in them describes what is known about the latent states based on the data.

Many time series arising in applied fields are generated by processes that are inherently non-linear, which motivates the need for state-space models and methods that account for such non-linearities. From the point of view of statistical inference of SSMs, non-linearities pose a problem, since analytical, closed-form solutions are mainly available for the special cases of linear Gaussian SSMs [cf. Durbin and Koopman 2012] and discrete state-space models (sometimes also called hidden Markov models) [Baum and Petrie 1966; see also Rabiner 1989]. Outside these special cases, the inference of SSMs typically involves some form of (Gaussian) approximation [cf. Särkkä 2013] or is fully based on simulation using methods such as Markov chain Monte Carlo (MCMC) [cf. Robert and Casella 2004].

Since the early 1990's, MCMC methods such as the Metropolis-Hastings algorithm [Hastings 1970] and the Gibbs sampler [S. Geman and D. Geman 1984; Gelfand and Smith 1990] have been successful in the inference of many Bayesian statistical models. Typically, these algorithms are used to update the unknown parameters of the model one at a time or in blocks of multiple parameters. However, it is well-known by practitioners of SSMs that the high dimension and dependence often present in the latent state process can render direct Metropolis-Hastings or Gibbs updates inefficient [cf. Fearnhead 2011]. Therefore, efficient inference methods that are tailored to the SSM inference problems are needed.

Fortunately, particle filters or more generally, 'sequential Monte Carlo' (SMC) methods [cf. Doucet, De Freitas, and Gordon 2001] have emerged as alternative methods of simulation-based inference. The history of particle filters dates back to the paper of Gordon, Salmond, and Smith [1993], who were the first to incorporate a crucial 'resampling step' to their inference algorithm, bridging the gap from (sequential) importance sampling to particle filtering. Since then, new particle filters have been developed, extensively applied and theoretically studied [cf. Godsill 2019; Doucet and Johansen 2011].

Recently, the interest in particle filtering and SMC methods has been further elevated by the seminal paper of Andrieu, Doucet, and Holenstein [2010] that introduced particle Markov chain Monte Carlo (PMCMC) methods that involve using particle filters within MCMC. In particular, the paper introduced a special kind of particle filter called the 'conditional particle filter', which forms the basis for the methodological developments in this thesis.

This thesis is a mix of methodology and application. Articles II and IV develop new particle MCMC methods for the statistical inference of general state-space models. These developments are both conditional particle filters that are suitable for models that have uninformative initial state distributions (Article II) and for models that have slowly-mixing dynamic models and uninformative observations (Article IV).

Articles I and III are applications of existing methodology that develop new non-linear state-space models in order to solve problems in applied fields. Article I focuses on a prediction problem where the white blood cell counts of children receiving chemotherapy for the treatment of acute lymphoblastic leukaemia are modelled using state-space models arising as approximations to non-linear stochastic differential equations. Article III applies a state-of-the-art particle filter and tracking model to the estimation of the number and locations of wolf territories in Finland, using presence-only citizen science observations such as tracks and sightings of wolves.

The rest of the introductory part of this thesis reviews the methodological background of Articles I–IV and summarises the research contribution of this thesis. Chapter 2 defines state-space models and describes inferential tasks that are of interest with them. Then, Chapters 3–5 review concrete inference methods for solving these inferential tasks under various assumptions on the state-space model. Chapter 3 covers linear Gaussian state-space models and approximate inference for Gaussian state-space models involving non-linearities using the extended Kalman filter. Chapter 4 discusses particle filters and the inference of general state-space models using them. Chapter 5 covers conditionally linear Gaussian models, the inference of which uses techniques discussed in Chapters 3 and 4. Finally, Chapter 6 summarises the research contribution of this thesis in relation to Chapters 3–5 and Chapter 7 concludes with a discussion.

# 2 STATE-SPACE MODELS

## 2.1 Definition

State-space models (SSMs) are a class of time series models for a $p$-dimensional time series $y_1, y_2, \dots, y_n$, assumed to arise as a realisation of a stochastic observation process $Y_1, Y_2, \dots, Y_n$. SSMs assume that each random variable $Y_i$ is conditionally independent given $X_i$, where each $X_i$ is a random variable from a $d$-dimensional latent Markov process $X_0, X_1, \dots, X_n$, which is commonly referred to as the state process. An SSM can be written in the following general form:

$$X_k \mid (X_{k-1} = x_{k-1}) \sim f_k(\cdot \mid x_{k-1}), \text{ for } k\ \geq 1 \text{ and } X_0 \sim f_0 \tag{1a}$$

$$Y_k \mid (X_k = x_k) \sim g_k(\cdot \mid x_k), \text{ for } k \geq 1, \tag{1b}$$

where $f_0$ is the prior distribution for the initial state $X_0$, $(f_k)_{k \geq 1}$ are Markov transitions of the state, and $g_k$ for $k \geq 1$ correspond to the conditional distributions of $Y_k$ given $X_k = x_k$. Equation (1a) is commonly referred to as the *state equation* or the *dynamic model* and (1b) as the *observation equation* or *observation model*. Figure 1 visualises the model as a directed acyclic graph.

Often, the distributions $f_k$ and $g_k$ may also depend on parameters $\theta$. When this dependence is of particular interest, we will denote the distributions by $f_k^{(\theta)}$ and $g_k^{(\theta)}$. Furthermore, we shall in general make the assumption that $f_k$ and $g_k$ admit densities, and reuse $f_k$ and $g_k$ to mean their densities instead of distributions.

## 2.2 Inference tasks

The most common inferential tasks related to SSMs of the form (1) are *filtering*, *prediction*, *smoothing* and *parameter inference* [cf. Särkkä 2013]. This section defines these inference tasks and discusses them on a high level. Hereafter, we denote by

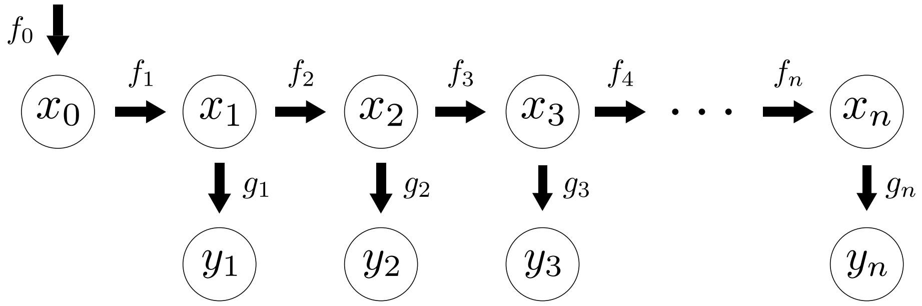


FIGURE 1 A directed acyclic graph highlighting the dependency structure of state-space models.

$z_{i:j} := (z_i, z_{i+1}, \ldots, z_j)$ for $i \leq j$ a sequence of consecutive variables, and by $z_{i:j}$ for $i > j$ an empty sequence.

*Filtering* or *state filtering* is concerned with the computation of the so called 'filtering distributions' $p(x_k \mid y_{1:k})$[1] for $k = 1, \ldots, n$. The $k$th filtering distribution corresponds to the distribution of the state $x_k$ given the first $k$ observations $y_{1:k}$, and may be expressed as

$$p(x_k \mid y_{1:k}) = \frac{g_k(y_k \mid x_k)p(x_k \mid y_{1:k-1})}{p(y_k \mid y_{1:k-1})}, \tag{2}$$

where $p(y_k \mid y_{1:k-1}) = \int_{\mathcal{X}} g_k(y_k \mid x_k)p(x_k \mid y_{1:k-1})\mathrm{d}x_k$, with $\mathcal{X}$ denoting the domain of a state variable.

*Prediction* means computing the *predictive distributions* of future states or observations, that is, the distributions $p(x_{n+1} \mid y_{1:n})$ or $p(y_{n+1} \mid y_{1:n})$:

$$\begin{aligned} p(x_{n+1} \mid y_{1:n}) &= \int_{\mathcal{X}} f_{n+1}(x_{n+1} \mid x_n)p(x_n \mid y_{1:n})\mathrm{d}x_n, \text{ and} \\ p(y_{n+1} \mid y_{1:n}) &= \int_{\mathcal{X}} g_{n+1}(y_{n+1} \mid x_{n+1})p(x_{n+1} \mid y_{1:n})\mathrm{d}x_{n+1}, \end{aligned} \tag{3}$$

which may also be recursively used to obtain $k > 1$ step ahead predictive distributions $p(x_{n+k} \mid y_{1:n})$ or $p(y_{n+k} \mid y_{1:n})$.

*Smoothing* or *state smoothing* refers to the computation of 'smoothing distributions': distributions of the states given the full time series. Typically, the interest is in the *marginal* smoothing distributions $p(x_k \mid y_{1:n})$ for $k = 0, \ldots, n$, or the *full* smoothing distribution $p(x_{0:n} \mid y_{1:n})$, which preserves the dependencies between consecutive state variables. In general, the difference between smoothing and filtering is that in smoothing, state $x_k$ is also conditioned on all observations after time $k$.

The transitions $(f_k)_{k \geq 0}$ and/or the observation densities $(g_k)_{k \geq 1}$ of the SSM (1) may also depend on unknown parameters $\theta$ that must — possibly in addition to the states — also be inferred based on the observed data $y_{1:n}$. This is known

[1] In general, we will use the common notation where $p$ stands for a generic joint, marginal or conditional distribution/density, and the arguments of $p$ indicate which distribution is in question. Especially with conditional distributions, when there is possibility of ambiguity with this notation, we use additional subscripts for $p$ to denote which random variables' distribution we mean.

as *parameter inference*. In the Bayesian setting, the parameters $\theta$ are given a prior distribution $p_\theta$ and modelled alongside the SSM (1).

Considering parameter inference as part of the inference problem complicates the solution of the filtering, smoothing and prediction problems, since then the parameters have to be integrated out from the filtering, smoothing and prediction equations. For example, the distribution $p(y_{n+1} \mid y_{1:n})$ above becomes

$$p(y_{n+1} \mid y_{1:n}) = \int_\Theta \int_\mathcal{X} g^{(\theta)}_{n+1}(y_{n+1} \mid x_{n+1}) p(x_{n+1} \mid \theta, y_{1:n}) p(\theta \mid y_{1:n}) \mathrm{d}x_{n+1} \mathrm{d}\theta,$$

involving an integral over the parameter space $\Theta$. In a fully Bayesian setting, this complication may be dealt with by inferring joint posteriors of the states and parameters, such as $p(x_k, \theta \mid y_{1:k})$ for filtering, $p(y_{n+1}, \theta \mid y_{1:n})$ for prediction or $p(x_{0:n}, \theta \mid y_{1:n})$ for smoothing.

It is also possible to obtain a point estimate of $\theta$, and then use the obtained estimate as the value of $\theta$. A point estimate can be obtained as a maximum a posteriori (MAP) estimate of $\theta$, by maximising the marginal posterior density

$$p(\theta \mid y_{1:n}) \propto p(y_{1:n} \mid \theta) p(\theta), \tag{4}$$

where

$$p(y_{1:n} \mid \theta) = \prod_{k=1}^{n} p(y_k \mid y_{1:k-1}, \theta), \tag{5}$$

with

$$p(y_k \mid y_{1:k-1}, \theta) = \int_\mathcal{X} p(y_k \mid x_k, y_{1:k-1}, \theta) p(x_k \mid y_{1:k-1}, \theta) \mathrm{d}x_k,$$

is the marginal likelihood. It is also possible to use maximum likelihood, in which solely the marginal likehood is maximised with respect to the parameters $\theta$. The use of point estimates might however result in a drastic underestimation of uncertainty in the inferred distributions, since uncertainty regarding the parameter value is disregarded.

# 3 GAUSSIAN STATE-SPACE MODELS WITH NON-LINEARITIES

This chapter discusses approximate inference for a special case of the SSM (1) where the distributions $f_k$ and $g_k$ are Gaussian but involve non-linear functions of the states. We will begin by reviewing the inference for a linear Gaussian state-space model, which provides a building block for the approximate inference.

## 3.1 Linear Gaussian state-space models

Linear Gaussian state-space models (LGSSMs) [cf. Durbin and Koopman 2012; Harvey 1990] are a special case of the SSM (1) where the distributions $f_k$ and $g_k$ are Gaussian with linear dependencies. An LGSSM has the general form:

$$\begin{aligned} X_k &= T_k X_{k-1} + R_k \eta_k, \text{ and } X_0 \sim N(\mu_0, \Sigma_0) \\ Y_k &= Z_k X_k + \epsilon_k, \end{aligned} \tag{6}$$

where $N(x, C)$ stands for the multivariate normal distribution with mean vector $x$ and covariance $C$, $\eta_k \sim N(0, Q_k)$ and $\epsilon_k \sim N(0, H_k)$ are independent, and $\mu_0$ and $\Sigma_0$ are the mean vector and covariance matrix of the initial distribution for $X_0$. The constant matrices $(T_k)_{k\geq 1}$, $(R_k)_{k\geq 1}$, $(Z_k)_{k\geq 1}$ and covariances $(Q_k)_{k\geq 1}$, $(H_k)_{k\geq 1}$ have appropriate dimensions. An LGSSM may be placed in to the general form of the SSM (1) by setting:

$$\begin{aligned} f_0 &= N(\mu_0, \Sigma_0) \\ f_k(\cdot \mid x_{k-1}) &= N(T_k x_{k-1}, R_k Q_k R_k^{'}) \text{ for } k \geq 1 \\ g_k(\cdot \mid x_k) &= N(Z_k x_k, H_k) \text{ for } k \geq 1, \end{aligned}$$

where $X^{'}$ stands for the transpose of the matrix $X$.

In the case of an LGSSM, the $k$th filtering distribution, $p(x_k \mid y_{1:k})$, is Gaussian with mean and covariance denoted by $\mu_{k|k}$ and $\Sigma_{k|k}$, respectively. These may be computed for $k = 1, 2, \ldots, n$ using the well-known recursive algorithm known as the Kalman filter [Kalman 1960; Durbin and Koopman 2012, Section 4.3]:

$$
\begin{aligned}
&\textbf{Prediction step:} \\
&\qquad \mu_{k|k-1} = T_k \mu_{k-1|k-1} \\
&\qquad \Sigma_{k|k-1} = T_k \Sigma_{k-1|k-1} T_k^{'} + R_k Q_k R_k^{'} \\
&\textbf{Update step:} \\
&\qquad v_k = y_k - Z_k \mu_{k|k-1} \\
&\qquad F_k = Z_k \Sigma_{k|k-1} Z_k^{'} + H_k \\
&\qquad K_k = \Sigma_{k|k-1} Z_k^{'} F_k^{-1} \\
&\qquad \mu_{k|k} = \mu_{k|k-1} + K_k v_k \\
&\qquad \Sigma_{k|k} = \Sigma_{k|k-1} - K_k Z_k \Sigma_{k|k-1},
\end{aligned}
\tag{7}
$$

where $\mu_{i|j} := \mathbb{E}[X_i \mid Y_{1:j} = y_{1:j}]$, $\Sigma_{i|j} := \mathrm{Cov}[X_i \mid Y_{1:j} = y_{1:j}]$, $\mu_{0|0} := \mu_0$, $\Sigma_{0|0} := \Sigma_0$ and $X^{-1}$ stands for the inverse of the matrix $X$. The Kalman filter proceeds by computing the predictive state distribution $p(x_k \mid y_{1:k-1}) = N(\mu_{k|k-1}, \Sigma_{k|k-1})$ in the 'prediction step', and then updating the predicted distribution to the $k$th filtering distribution $p(x_k \mid y_{1:k}) = N(\mu_{k|k}, \Sigma_{k|k})$ by conditioning on the latest observation $y_k$ in the 'update step'.

If at time $k$ the observation $y_k$ is missing (at random), the update step can be omitted, and $\mu_{k|k} = \mu_{k|k-1}$ and $\Sigma_{k|k} = \Sigma_{k|k-1}$ set instead. This provides a convenient mechanism for dealing with missing values in the time series [cf. Durbin and Koopman 2012, Section 4.10]. In a similar fashion, the predictive distribution of the states $p(x_{n+k} \mid y_{1:n})$ for some $k$ may be computed by considering the 'future values' of the series, $y_{n+1}, y_{n+2}, \ldots, y_{n+k}$, as missing, and then running the Kalman filter for the series $y_{1:n+k}$ [cf. Durbin and Koopman 2012, Section 4.11]. In effect, this computes the predictive state distribution

$$X_{n+k} \mid (Y_{1:n} = y_{1:n}) \sim N(\mu_{n+k|n}, \Sigma_{n+k|n}). \tag{8}$$

Since $Y_{n+k} = Z_{n+k} X_{n+k} + \epsilon_{n+k}$, the predictive distributions for observations are then given by

$$Y_{n+k} \mid (Y_{1:n} = y_{1:n}) \sim N(Z_{n+k}\mu_{n+k|n}, Z_{n+k}\Sigma_{n+k|n}Z_{n+k}^{'} + H_{n+k}). \tag{9}$$

The $k$th marginal smoothing distribution $p(x_k \mid y_{1:n})$ of an LGSSM is Gaussian with mean and covariance $\mu_{k|n}$ and $\Sigma_{k|n}$, respectively. These can be recursively obtained in a backward pass for $k = n-1, n-2, \ldots, 0$, which is known as the Kalman smoother (Rauch-Tung-Striebel smoother) [Rauch, Tung, and Striebel 1965; Särkkä 2013, p. 135–136]:

$$
\begin{aligned}
G_k &= \Sigma_{k|k} T_{k+1}^{'} \Sigma_{k+1|k}^{-1} \\
\mu_{k|n} &= \mu_{k|k} + G_k(\mu_{k+1|n} - \mu_{k+1|k}) \\
\Sigma_{k|n} &= \Sigma_{k|k} + G_k(\Sigma_{k+1|n} - \Sigma_{k+1|k})G_k^{'}.
\end{aligned}
\tag{10}
$$

Finally, the marginal likelihood of an LGSSM is available in closed form [cf. Durbin and Koopman 2012, p. 171], since a computation similar to (9) yields:

$$\begin{aligned}
&\log\big(p(y_{1:n})\big) \\
&= \log\Big(\prod_{k=1}^{n} p(y_k \mid y_{1:k-1})\Big) \\
&= \log\Big(\prod_{k=1}^{n} N(y_k; Z_k\mu_{k|k-1}, Z_k\Sigma_{k|k-1}Z_k' + H_k)\Big) \\
&= -\frac{np}{2}\log(2\pi) - \frac{1}{2}\sum_{k=1}^{n}\big[\log(|F_k|) + v_k' F_k^{-1} v_k\big],
\end{aligned} \tag{11}$$

where $p$ is the dimension of the observations, $N(y;\mu,\Sigma)$ stands for the density of $N(\mu,\Sigma)$ evaluated at $y$, and $v_k$ and $F_k$ are computed by the Kalman filter (7).

## 3.2 Introducing non-linearity and the extended Kalman filter

Let us now turn to study a non-linear generalisation of the LGSSM introduced in Section 3.1:

$$\begin{aligned}
f_0 &= N(\mu_0, \Sigma_0) \\
f_k(\cdot \mid x_{k-1}) &= N(T_k(x_{k-1}), R_k(x_{k-1})Q_k(x_{k-1})R_k(x_{k-1})') \\
g_k(\cdot \mid x_k) &= N(Z_k(x_k), H_k(x_k)),
\end{aligned} \tag{12}$$

that is

$$\begin{aligned}
X_k &= T_k(X_{k-1}) + R_k(X_{k-1})\eta_k \text{ and } X_0 \sim N(\mu_0, \Sigma_0) \\
Y_k &= Z_k(X_k) + \epsilon_k,
\end{aligned} \tag{13}$$

where $\eta_k \sim N\big(0, Q_k(X_{k-1})\big)$, $\epsilon_k \sim N\big(0, H_k(X_k)\big)$, and $T_k$, $R_k$, $Q_k$, $Z_k$ and $H_k$ are now differentiable functions.

There exists a vast literature regarding methods used for approximate inference of models similar to (13) [cf. Särkkä 2013, Sections 5 and 6], including unscented Kalman filters [Julier, Uhlmann, and Durrant-Whyte 1995; Julier and Uhlmann 2004], the Gauss-Hermite or quadrature Kalman filter [Ito and Xiong 2000; Arasaratnam, Haykin, and Elliott 2007], and the cubature Kalman filter [Arasaratnam and Haykin 2009], to name a few. This section reviews the earliest of them, called the extended Kalman filter (EKF) [Jazwinski 1970; Maybeck 1982], which may be seen as a generalisation of the Kalman filter (7).

The EKF may be derived by expanding the functions $T_k$, $R_k$, $Q_k$, $Z_k$ and $H_k$ in (13) using Taylor series around $\mu_{k-1|k-1}$ (for $T_k$, $R_k$ and $Q_k$) or $\mu_{k|k-1}$ (for $Z_k$ and $H_k$), and then applying the Kalman filter for the resulting linearised model [cf. Durbin and Koopman 2012, Section 10.2]. This yields the approximate prediction and update equations:

$$
\begin{aligned}
&\textbf{Prediction:}\\
&\qquad \mu_{k|k-1} = T_k(\mu_{k-1|k-1})\\
&\qquad \Sigma_{k|k-1} = \dot{T}_k \Sigma_{k-1|k-1} \dot{T}_k' + R_k(\mu_{k-1|k-1}) Q_k(\mu_{k-1|k-1}) R_k(\mu_{k-1|k-1})'\\
&\textbf{Update:}\\
&\qquad v_k = y_k - Z_k(\mu_{k|k-1})\\
&\qquad F_k = \dot{Z}_k \Sigma_{k|k-1} \dot{Z}_k' + H_k(\mu_{k|k-1})\\
&\qquad K_k = \Sigma_{k|k-1} \dot{Z}_k' F_k^{-1}\\
&\qquad \mu_{k|k} = \mu_{k|k-1} + K_k v_k\\
&\qquad \Sigma_{k|k} = \Sigma_{k|k-1} - K_k \dot{Z}_k P_{k|k-1},
\end{aligned}
\tag{14}
$$

where

$$\dot{T}_k := \left.\frac{\partial T_k(x)}{\partial x'}\right|_{x=\mu_{k-1|k-1}} \quad \text{and} \quad \dot{Z}_k := \left.\frac{\partial Z_k(x)}{\partial x'}\right|_{x=\mu_{k|k-1}}$$

are Jacobian matrices evaluated at the points $\mu_{k-1|k-1}$ and $\mu_{k|k-1}$, respectively.

Approximations to the predictive distributions, marginal smoothing distributions and the marginal log-likelihood may be obtained similarly as in the case of an LGSSM, using (8)–(11) with the required quantities obtained from (14) and $T_{k+1}$ replaced by $\dot{T}_{k+1}$ in (10). The resulting smoothing algorithm is known as the extended Rauch-Tung-Striebel smoother [Cox 1964; Särkkä 2013, p. 144].

# 4 INFERENCE OF GENERAL STATE-SPACE MODELS USING PARTICLE FILTERS

In this chapter, we discuss inference methods for SSMs (1) that do not place any further assumptions on the distributions $(f_k)_{k \geq 0}$ or $(g_k)_{k \geq 1}$ than those discussed in Section 2.1. This freedom in specifying 'any state-space model we want', rather generically, leads to inference algorithms that are fully based on Monte Carlo simulation. The methods discussed in this chapter are most relevant for the inference of non-linear state-space models (NSSMs), which are SSMs of the form (1) that do not have the structure of an LGSSM (6).

The main methods of this chapter will be based on so called particle filters that are Monte Carlo algorithms useful for approximating sequences of probability distributions [cf. Doucet, De Freitas, and Gordon 2001]. A particle filter approximates each distribution in sequence using $N$ weighted values called 'particles', which are propagated using simulation along the sequence of distributions. The weighted particles can then be used to estimate expected values of interest with respect to the distributions in the sequence. Particle filters find uses also outside the context of time series modelling where they are typically called sequential Monte Carlo samplers [Chopin and Papaspiliopoulos 2020, Section 17].

Next, Section 4.1 will introduce a basic particle filter. This method will then be slightly generalised using so called Feynman–Kac models that are the topic of Section 4.2. Section 4.3 discusses the conditional particle filter that is especially relevant for solving the smoothing problem. Finally, Section 4.4 introduces two methods that use particle filters for the joint inference of the states $x_{0:n}$ and parameters $\theta$ of an SSM.

## 4.1 A basic particle filter

Particle filters are extensions of importance sampling [cf. Geweke 1989], and in the context of SSMs, the sequence of distributions we are interested in approximating is the sequence $\pi_k(x_{0:k}) := p(x_{0:k} \mid y_{1:k})$ for $k = 1, 2, \ldots, n$. In particular, assume that we are interested in approximating the expected value

$$\mathbb{E}[h(X_{0:k}) \mid Y_{1:k} = y_{1:k}] = \int_{\mathcal{X}^{k+1}} h(x_{0:k}) p(x_{0:k} \mid y_{1:k}) \mathrm{d}x_{0:k},$$

for some $k \geq 1$. A naive, inefficient approximation may be obtained via ('self-normalised') importance sampling with an importance distribution $q^1$, using the estimator

$$\sum_{i=1}^{N} h(X_{0:k}^{(i)}) \tilde{W}_k^{(i)} \quad \text{with } X_{0:k}^{(i)} \sim q, \tag{15}$$

where $\tilde{W}_k^{(i)} = W_k^{(i)} / \sum_{j=1}^{N} W_k^{(j)}$ are the normalised importance weights, and $W_k^{(i)}$ are proportional to $\pi_k(X_{0:k}^{(i)}) / q(X_{0:k}^{(i)})$.

Notice that the distribution $\pi_k(x_{0:k}) = p(x_{0:k} \mid y_{1:k})$ can be related to the distribution $\pi_{k-1}(x_{0:k-1}) = p(x_{0:k-1} \mid y_{1:k-1})$ via the identity:

$$\begin{aligned} p(x_{0:k} \mid y_{1:k}) &= \frac{g_k(y_k \mid x_k) f_k(x_k \mid x_{k-1}) p(x_{0:k-1} \mid y_{1:k-1})}{p(y_k \mid y_{1:k-1})} \\ &\propto g_k(y_k \mid x_k) f_k(x_k \mid x_{k-1}) p(x_{0:k-1} \mid y_{1:k-1}), \end{aligned} \tag{16}$$

which may be derived using Bayes' rule and the conditional independencies arising from the structure of the model (1).

Furthermore, if the importance distribution $q = q_{0:k}$ is chosen such that

$$q_{0:k}(x_{0:k} \mid y_{1:k}) = q_0(x_0) \prod_{j=1}^{k} q_j(x_j \mid x_{j-1}, y_{1:j}), \tag{17}$$

the importance weights can be computed recursively, since

$$\begin{aligned} W_k^{(i)} &\propto \frac{\pi_k(X_{0:k}^{(i)})}{q_{0:k}(X_{0:k}^{(i)} \mid y_{1:k})} \\ &\propto \frac{g_k(y_k \mid X_k^{(i)}) f_k(X_k^{(i)} \mid X_{k-1}^{(i)})}{q_k(X_k^{(i)} \mid X_{k-1}^{(i)}, y_{1:k})} \frac{\pi_{k-1}(X_{0:k-1}^{(i)})}{q_{0:k-1}(X_{0:k-1}^{(i)} \mid y_{1:k-1})} \\ &\propto \frac{g_k(y_k \mid X_k^{(i)}) f_k(X_k^{(i)} \mid X_{k-1}^{(i)})}{q_k(X_k^{(i)} \mid X_{k-1}^{(i)}, y_{1:k})} W_{k-1}^{(i)}. \end{aligned} \tag{18}$$

This form of recursive weight computation applied to (15) is known as sequential importance sampling (SIS) [cf. Doucet, De Freitas, and Gordon 2001, Section 1.3.2]. Here, for simplicity (and consistency with Section 4.3), we have made the assumption that in (17) each $q_j$ for $1 \leq j \leq k$ only depends on the previous state $x_{j-1}$ and the observations $y_{1:j}$, although in general nothing prevents the $q_j$'s from depending on the full past 'trajectory' $x_{0:j-1}$ and/or on the observations $y_{1:k}$.

[1] satisfying the support condition $q(x_{0:k}) = 0 \Longrightarrow \pi_k(x_{0:k}) = 0$ for all $x_{0:k} \in \mathcal{X}^{k+1}$.

In the context of SIS (and particle filters), it is useful to note that the estimator (15) is equal to the expected value of the function $h$ with respect to the empirical distribution placing probability $\tilde{W}_k^{(i)}$ for trajectory (particle) $X_{0:k}^{(i)}$. Thus, the normalised weights and particles $(\tilde{W}_k^{(i)}, X_{0:k}^{(i)})$, $i = 1, 2, \ldots, N$ can be interpreted as an empirical distribution approximating $\pi_k(x_{0:k})$.

It turns out that SIS suffers from a problem where most of the particles $X_{0:k}^{(i)} \sim q_{0:k}$ for $k$ large will have close to zero (importance) weights, thus providing a poor approximation of the distribution $\pi_k(x_{0:k})$ [cf. Cappé, Moulines, and Rydén 2005, Section 7.3.1]. To circumvent this problem, a particle filter adds a resampling step to SIS, which probabilistically eliminates and duplicates particles after weight computation such that particles with small weights are most likely to be eliminated.

This leads to a basic particle filter (Algorithm 1) targeting an SSM of the form (1) with transitions $(f_k)_{0 \le k \le n}$, observation densities $(g_k)_{1 \le k \le n}$, importance distribution $q_{0:n}$ and $N$ particles [cf. Särkkä 2013, Section 7.4; or Doucet, De Freitas, and Gordon 2001]. Here, we also introduce a notation where $z_k^{(i:j)} := (z_k^{(i)}, z_k^{(i+1)}, \ldots, z_k^{(j)})$ for $i \le j$ stands for a collection of values of state variables, and $z_k^{(i:j)}$ for $i > j$ stands for an empty collection.

Algorithm 1 progresses for $k \ge 0$ by simulating the particles $\mathbf{X}_k^{(1:N)}$ from the proposal $q_k$ (lines 1–2 and 6–7), computing their weights $W_k^{(1:N)}$ (lines 3 and 8), and resampling the particles. The resampling operation $r$ occurs on line 5 and outputs so called 'ancestor indices' $A_{k-1}^{(1:N)}$ that index the particles that were chosen (not eliminated) in the resampling. As the algorithm progresses, the full set of ancestor indices $A_{0:n-1}^{(1:N)}$ forms an 'ancestral lineage', where $A_{k-1}^{(i)} = j$ implies that the 'ancestor' of the particle $\mathbf{X}_k^{(i)}$ is the particle $\mathbf{X}_{k-1}^{(j)}$. Note that the particles $\mathbf{X}_k^{(1:N)}$ may be reconstructed from the 'particle system' $X_{0:n}^{(1:N)}$ and ancestor indices $A_{0:n-1}^{(1:N)}$.

The resampling operation can be implemented in many ways, the most common of which is multinomial resampling, which was the first resampling algorithm used with particle filters [Gordon, Salmond, and Smith 1993]. In multinomial resampling, the ancestor indices $A_{k-1}^{(1:N)}$ are drawn from the categorical distribution $\mathrm{Categ}(W_{k-1}^{(1:N)})$, that is, a discrete probability distribution that places probability $W_{k-1}^{(i)} / \sum_{j=1}^{N} W_{k-1}^{(j)}$ for outcome $i$. Another common resampling operation is systematic resampling [Whitley 1994; Carpenter, Clifford, and Fearnhead 1999] (Algorithm 2), which is popular since it typically works well in practice and is simple to implement efficiently. Other common resampling algorithms are stratified [Kitagawa 1996] and residual resampling [Baker 1985; Higuchi 1997].

In fact, Algorithm 1 can be used with any *unbiased resampling* [Crisan, Del

**Algorithm 1** BasicParticleFilter($(f_k)_{0\le k\le n}, (g_k)_{1\le k\le n}, q_{0:n}, N$)

```
Simulate $X_0^{(i)} \sim q_0$ for $i = 1, \dots, N$.
Set $\mathbf{X}_0^{(i)} = X_0^{(i)}$ for $i = 1, \dots, N$.
Compute $W_0^{(i)} = f_0(\mathbf{X}_0^{(i)}) / q_0(\mathbf{X}_0^{(i)})$ for $i = 1, \dots, N$.
for $k = 1, 2, \dots, n$ do
    Simulate $A_{k-1}^{(i)} \sim r(\cdot \mid W_{k-1}^{(1:N)})$ for $i = 1, \dots, N$.
    Simulate $X_k^{(i)} \sim q_k(\cdot \mid X_{k-1}^{(A_{k-1}^{(i)})}, y_{1:k})$ for $i = 1, \dots, N$.
    Set $\mathbf{X}_k^{(i)} = (\mathbf{X}_{k-1}^{(A_{k-1}^{(i)})}, X_k^{(i)})$ for $i = 1, \dots, N$.
    Compute $W_k^{(i)} = \dfrac{g_k(y_k \mid X_k^{(i)}) f_k(X_k^{(i)} \mid X_{k-1}^{(A_{k-1}^{(i)})})}{q_k(X_k^{(i)} \mid X_{k-1}^{(A_{k-1}^{(i)})}, y_{1:k})}$ for $i = 1, \dots, N$.
end for
return $(X_{0:n}^{(1:N)}, W_{0:n}^{(1:N)}, A_{0:n-1}^{(1:N)})$
```

**Algorithm 2** SystematicResampling($W^{(1:N)}$)

```
Set $\tilde{W}^{(i)} = W^{(i)} / \sum_{j=1}^{N} W^{(j)}$ for $i = 1, 2, \dots, N$.
Simulate $U \sim \mathrm{Unif}(0, 1)$.
Set $\tilde{U}^{(i)} = (U + i - 1)/N$ for $i = 1, 2, \dots, N$.
for $i = 1, 2, \dots, N$ do
    Set $A^{(i)} = j$ where $j$ is such that $\sum_{k=1}^{j-1} \tilde{W}^{(k)} < \tilde{U}^{(i)} \le \sum_{k=1}^{j} \tilde{W}^{(k)}$.
end for
return $A^{(1:N)}$
```

Moral, and Lyons 1999], that is, a resampling that satisfies:

$$\mathbb{E}\Big[\sum_{i=1}^{N} 1(A_{k-1}^{(i)} = j)\Big] = N \frac{W_{k-1}^{(j)}}{\sum_{i=1}^{N} W_{k-1}^{(i)}}. \tag{19}$$

With simple modifications to Algorithm 1, it is also possible to resample the particles 'adaptively' [cf. Chopin and Papaspiliopoulos 2020, Section 10.2], only when the effective sample size of the normalised weights [Liu 1996] reaches some threshold, but this direction is not studied further in this thesis.

Most importantly, the output of Algorithm 1 provides an approximation of $\pi_k(x_{0:k}) = p(x_{0:k} \mid y_{1:k})$ as the empirical distribution with support points $\mathbf{X}_k^{(i)}$ and associated probabilities $\tilde{W}_k^{(i)} = W_k^{(i)} / \sum_{j=1}^{N} W_k^{(j)}$, $i = 1, 2, \dots, N$. For $k \ge 0$, we may estimate

$$\int_{\mathcal{X}^{k+1}} h(x_{0:k}) p(x_{0:k} \mid y_{1:k}) \mathrm{d}x_{0:k} \ \text{ using } \ \sum_{i=1}^{N} h(\mathbf{X}_k^{(i)}) \tilde{W}_k^{(i)} \tag{20a}$$

$$p(y_{1:k}) \ \text{ using } \ \hat{p}(y_{1:k}) := \prod_{j=0}^{k} \left( \frac{1}{N} \sum_{i=1}^{N} W_j^{(i)} \right). \tag{20b}$$

The estimator (20a) is consistent as $N \to \infty$ under certain technical assumptions using multinomial [Del Moral 2004], and stratified as well as residual resampling

[Gerber, Chopin, and Whiteley 2019]. Gerber, Chopin, and Whiteley [2019] further discuss that for systematic resampling, a convergence criterion might fail depending on the order of the input particles, based on an example given by Douc and Cappé [2005]. The example constructed by Douc and Cappé [2005] however considers only a single isolated resampling operation, and is therefore inconclusive from the point of view of practical applications of the estimator (20a). The work of Gerber, Chopin, and Whiteley [2019] also features another resampling (Srinivasan sampling process), which is in a sense 'close' to systematic resampling, and leads to a consistent estimator.

In practice, the estimator (20a) typically converges quickly for functions $h$ that only depend on state variables later in the sequence, that is, $h$ depends on $x_{l:k}$, where $k - l$ is small. For estimating expectations involving early state variables (such as $x_1$) convergence may be slow, and therefore, the methods introduced later in Section 4.3 are preferable.

The estimator (20b) gives a means of estimating the normalising constant of $p(x_{0:k} \mid y_{1:k})$. It can be shown that the estimator is unbiased given an unbiased resampling (19) [Del Moral 2004; see also Vihola, Helske, and Franks 2017, Proposition 21 (i)].

Finally, the particle approximation $(\tilde{W}_n^{(i)}, X_n^{(i)}), i = 1, 2, \ldots, N$ of $p(x_n \mid y_{1:n})$ can also be used for prediction, since for $k \geq 1$, we may estimate

$$\begin{aligned} &p(x_{n+k} \mid y_{1:n}) \;\text{ using }\; \sum_{i=1}^{N} \tilde{W}_n^{(i)} p_{X_{n+k}|X_n}(x_{n+k} \mid X_n^{(i)}), \text{ and} \\ &p(y_{n+k} \mid y_{1:n}) \;\text{ using }\; \sum_{i=1}^{N} \tilde{W}_n^{(i)} p_{Y_{n+k}|X_n}(y_{n+k} \mid X_n^{(i)}), \end{aligned} \tag{21}$$

which follow by appropriate choices of $h$ in (20a). If the densities in (21) are not available analytically, their Monte Carlo approximations can be used instead.

## 4.2 The Feynman–Kac representation of a state-space model

Next, we will discuss an alternative representation of an SSM of the form (1) as a so called Feynman–Kac (FK) model [Del Moral 2004; see also Chopin and Papaspiliopoulos 2020, Section 5] that is used for the remainder of Chapter 4. FK models are attractive representations of SSMs from many points of view [see Chopin and Papaspiliopoulos 2020, Section 5.1.3 for a listing], but in particular they provide a convenient abstraction that allows for separating the statistical model of interest (the SSM) from its representation in particle filtering algorithms. As an example of this, we slightly generalise Algorithm 1 of Section 4.1 below.

The idea of FK models is to represent the SSM in terms of

- an (alternative) initial distribution $M_0$ for the state,
- (alternative) state Markov transitions $M_k(\cdot \mid x_{k-1})$ for $1 \leq k \leq n$, and

- so called 'potential functions' $G_0 : \mathcal{X} \to [0, \infty)$ and $G_k : \mathcal{X}^2 \to [0, \infty)$ for $1 \leq k \leq n$.

In this thesis, we make the additional assumption that $M_{0:n} = (M_k)_{0 \leq k \leq n}$ admit densities, which we will also denote using the symbols $M_k$.

It is assumed that an FK representation (a particular choice of $M_{0:n}$ and $G_{0:n}$ above) of an SSM has the same joint distribution as the joint posterior distribution of the states of the SSM. In our setting, this means that given that the observations $y_{1:n}$ are fixed, the joint density of all latent states $x_{0:n}$ and observations $y_{1:n}$ of the SSM,

$$p(x_{0:n}, y_{1:n}) = f_0(x_0) \prod_{k=1}^{n} f_k(x_k \mid x_{k-1}) g_k(y_k \mid x_k), \tag{22}$$

is equal to the joint density of the FK model

$$\kappa(x_{0:n}) := M_0(x_0) G_0(x_0) \prod_{k=1}^{n} M_k(x_k \mid x_{k-1}) G_k(x_{k-1}, x_k), \tag{23}$$

that is,

$$p(x_{0:n}, y_{1:n}) = \kappa(x_{0:n}), \text{ for any } x_{0:n} \in \mathcal{X}^{n+1} \text{ when } y_{1:n} \text{ are fixed.} \tag{24}$$

As an example, we can express an SSM of the form (1) using the FK model:

$$\begin{aligned} M_0(\cdot) &= q_0 \\ G_0(x_0) &= f_0(x_0)/q_0(x_0) \\ M_k(\cdot \mid x_{k-1}) &= q_k(\cdot \mid x_{k-1}), \text{ for } 1 \leq k \ \leq n \\ G_k(x_{k-1}, x_k) &= \frac{g_k(y_k \mid x_k) f_k(x_k \mid x_{k-1})}{q_k(x_k \mid x_{k-1})}, \text{ for } 1 \leq k \leq n, \end{aligned} \tag{25}$$

which clearly satisfies assumption (24), and where $q_k$ for $k \geq 0$ is a 'proposal distribution' as in Section 4.1.

Algorithm 3 details a generic particle filter for the SSM underlying the FK model $\mathcal{FK}_{0:n} := (M_{0:n}, G_{0:n})$ using $N$ particles and an unbiased resampling $r$ [Chopin and Papaspiliopoulos 2020, Section 10.1]. With the FK model (25), Algorithm 3 corresponds to the basic particle filter (Algorithm 1) of Section 4.1. However, we note that Algorithm 3 is less involved and allows us to interpret the distributions $M_{0:n}$ as distributions for propagating the particles forward ('proposal distributions'), and the potential functions $G_{0:n}$ as functions for computing the weights of the particles.

Algorithm 3 is also slightly more general than Algorithm 1. Algorithm 1 only considers models of the form (25), which satisfy

$$p(x_{0:k}, y_{1:k}) = \kappa(x_{0:k}), \text{ for } \textit{all } 1 \leq k \leq n. \tag{26}$$

In contrast, with Algorithm 3 we may target also models that do not satisfy (26). As an example, consider the models similar to the ones introduced in [Guarniero,

**Algorithm 3** PARTICLEFILTER($\mathcal{FK}_{0:n}, N$)

```
Simulate $X_0^{(i)} \sim M_0(\cdot)$ for $i = 1, 2, \ldots, N$.
Set $\mathbf{X}_0^{(i)} = X_0^{(i)}$ for $i = 1, 2, \ldots, N$.
Compute $W_0^{(i)} = G_0(\mathbf{X}_0^{(i)})$ for $i = 1, 2, \ldots, N$.
for $k = 1, \ldots, n$ do
    Simulate $A_{k-1}^{(1:N)} \sim r(\cdot \mid W_{k-1}^{(1:N)})$.
    Simulate $X_k^{(i)} \sim M_k(\cdot \mid X_{k-1}^{(A_{k-1}^{(i)})})$ for $i = 1, 2, \ldots, N$.
    Set $\mathbf{X}_k^{(i)} = (\mathbf{X}_{k-1}^{(A_{k-1}^{(i)})}, X_k^{(i)})$ for $i = 1, 2, \ldots, N$.
    Compute $W_k^{(i)} = G_k(X_{k-1}^{(A_{k-1}^{(i)})}, X_k^{(i)})$ for $i = 1, 2, \ldots, N$.
end for
return $(X_{0:n}^{(1:N)}, W_{0:n}^{(1:N)}, A_{0:n-1}^{(1:N)})$
```

Johansen, and Lee 2017], such as

$$\begin{aligned}
M_0(\cdot) &= q_0 \\
G_0(x_0) &= \frac{f_0(x_0)}{q_0(x_0)} \eta_0(x_0) \\
M_k(\cdot \mid x_{k-1}) &= q_k(\cdot \mid x_{k-1}), \text{ for } 1 \leq k \leq n \\
G_k(x_{k-1}, x_k) &= \frac{f_k(x_k \mid x_{k-1}) g_k(y_k \mid x_k)}{q_k(x_k \mid x_{k-1})} \frac{\eta_k(x_k)}{\eta_{k-1}(x_{k-1})}, \text{ for } 1 \leq k \leq n-1 \\
G_n(x_{n-1}, x_n) &= \frac{f_n(x_n \mid x_{n-1}) g_n(y_n \mid x_n)}{q_n(x_n \mid x_{n-1})} \frac{1}{\eta_{n-1}(x_{n-1})},
\end{aligned} \tag{27}$$

where $(\eta_k)_{0 \leq k \leq n-1}$ are suitably chosen 'twisting' functions. When (26) does not hold, however, (20) only holds for $k = n$ for the output of Algorithm 3.

## 4.3 Particle smoothing using the conditional particle filter

We will now move on to discuss the conditional particle filter (CPF) [Andrieu, Doucet, and Holenstein 2010], which is a PMCMC method for particle smoothing, that is, for the inference of the posterior $p(x_{0:n} \mid y_{1:n})$.

In a practical sense, the CPF together with a so called 'traceback' operation (discussed below) can be described as an MCMC method for simulating 'trajectories' $X_{0:n}^{(i)}, i = 1, 2, \ldots, M$ from $p(x_{0:n} \mid y_{1:n})$. The CPF is similar to the particle filter (Algorithm 3), but features a so called 'reference trajectory/particle' that is never mutated as the algorithm progresses.

Algorithm 4 below details the CPF targeting the FK model $\mathcal{FK}_{0:n} = (M_{0:n}, G_{0:n})$ using $N$ particles, given that the reference trajectory $X_{0:n}^*$ resides in the indices $B_{0:n}$ of the particle system $X_{0:n}^{(1:N)}$. In contrast to Algorithm 3, immutability of the reference trajectory is ensured on lines 2 and 7 — where $X_k^{(B_k)} = X_k^*$ for $k \geq 0$ is enforced — and by the resampling operation that occurs on line 5.

In fact, the resampling has been written in a slightly nonstandard way that differs from [Andrieu, Doucet, and Holenstein 2010], who focused on multinomial resampling. In contrast, Algorithm 4 works with a generic 'conditional resampling' $r^{(a,b)}$, which draws the ancestor indices $A_{k-1}^{(i)}$ for $i \neq b$ conditional on $A_{k-1}^{(b)} = a$, and therefore ensures that the reference indices $B_{0:n}$ are correctly recorded to the ancestor indices $A_{0:n-1}^{(1:N)}$. A sufficient condition which ensures that $r^{(a,b)}$ is valid for use with Algorithm 4 is given in Article IV (Definition 1), complementing the earlier work of Chopin and Singh [2015]. Article IV also provides two concrete resamplings that may be used with Algorithm 4: the conditional killing resampling and conditional systematic resampling with mean partition (Algorithms 5 and 6 in Article IV, respectively). Furthermore, the work of Chopin and Singh [2015] provides conditional variants of standard systematic resampling and residual resampling.[2]

**Algorithm 4** CPF($\mathcal{FK}_{0:n}, N, X_{0:n}^*, B_{0:n}$)

```
Simulate $X_0^{(i)} \sim M_0(\cdot)$ for $i = 1, 2, \ldots, N, i \neq B_0$.
Set $X_0^{(B_0)} = X_0^*$ and $\mathbf{X}_0^{(i)} = X_0^{(i)}$ for $i = 1, 2, \ldots, N$.
Compute $W_0^{(i)} = G_0(\mathbf{X}_0^{(i)})$ for $i = 1, 2, \ldots, N$.
for $k = 1, 2, \ldots, n$ do
    Simulate $A_{k-1}^{(1:N)} \sim r^{(B_{k-1}, B_k)}(\cdot \mid W_{k-1}^{(1:N)})$.
    Simulate $X_k^{(i)} \sim M_k(\cdot \mid X_{k-1}^{(A_{k-1}^{(i)})})$ for $i = 1, 2, \ldots, N, i \neq B_k$.
    Set $X_k^{(B_k)} = X_k^*$.
    Set $\mathbf{X}_k^{(i)} = (X_{k-1}^{(A_{k-1}^{(i)})}, X_k^{(i)})$ for $i = 1, 2, \ldots, N$.
    Compute $W_k^{(i)} = G_k(\mathbf{X}_k^{(i)})$ for $i = 1, 2, \ldots, N$.
end for
return $(X_{0:n}^{(1:N)}, W_{0:n}^{(1:N)}, A_{0:n-1}^{(1:N)})$
```

**Algorithm 5** TRACEBACK($X_{0:n}^{(1:N)}, W_n^{(1:N)}, A_{0:n-1}^{(1:N)}$, TRACEMETHOD)

```
Simulate $\tilde{B}_n \sim \mathrm{Categ}(W_n^{(1:N)})$.
$\tilde{B}_{0:n-1} \leftarrow$ TRACEMETHOD($X_{0:n}^{(1:N)}, A_{0:n-1}^{(1:N)}, \tilde{B}_n$)
return $(X_{0:n}^{(\tilde{B}_{0:n})}, \tilde{B}_{0:n})$
```

After running the CPF, a 'traceback' algorithm (Algorithm 5) may be run on the output $(X_{0:n}^{(1:N)}, W_{0:n}^{(1:N)}, A_{0:n-1}^{(1:N)})$ of the CPF. This yields new indices $\tilde{B}_{0:n}$ and reference $X_{0:n}^{(\tilde{B}_{0:n})}$. There are two ways to implement TRACEMETHOD in Algorithm 5: the original 'ancestor tracing' variant (Algorithm 6) introduced by Andrieu, Doucet, and Holenstein [2010] and the 'backward sampling' variant (Algorithm 7) [Whiteley 2010].

[2] With the additional assumption that the condition in the conditional resampling is always $A^{(1)} = 1$.

In ancestor tracing, the new ancestral path is chosen by backtracking along the ancestor indices $A_{0:n-1}^{(1:N)}$ sampled during the 'forward pass' (Algorithm 4). In contrast, backward sampling employs further sampling to generate the ancestral path, whose indices are drawn from $\mathrm{Categ}(\omega_k^{(1:N)})$. The 'backward sampling weights' $\omega_k^{(i)}$ are computed using the FK model, and their computation requires that the transition densities $(M_k)_{1\leq k\leq n}$ can be evaluated pointwise. Note that ancestor tracing does not require $X_{0:n}^{(1:N)}$; the first argument to Algorithm 6 is only included for consistency with Algorithm 7 as this allows us to write the generic Algorithm 5, which will be relevant for Section 4.4.

A single update $(X_{0:n}^*, B_{0:n}) \to (X_{0:n}^{(\tilde{B}_{0:n})}, \tilde{B}_{0:n})$ (that is, Algorithm 4 followed by Algorithm 5) detailed in Algorithm 8 constitutes a Markov transition that leaves $p(x_{0:n} \mid y_{1:n}) \times \mathrm{Unif}(\{1{:}N\}^{n+1})$ invariant. This result was shown first by Andrieu, Doucet, and Holenstein [2010] in the case of multinomial resampling and ancestor tracing. Then, Whiteley [2010] noted that the same holds with backward sampling. Chopin and Singh [2015] then showed that the above invariance holds in the cases of conditional systematic and residual resampling using ancestor tracing. Theorems 2 and 8 of Article IV further extend these results to the 'general case' of Algorithm 8 where the resampling (within Algorithm 4) is any valid conditional resampling (satisfies Definition 1 of Article IV) together with either ancestor tracing or backward sampling.

The invariance of the above Markov update together with mild irreducibility assumptions [cf. Roberts and Smith 1994] yield the estimator

$$\frac{1}{M}\sum_{j=1}^{M} h(\tilde{X}_{0:n}^{(j)}) \xrightarrow{M\to\infty} \int_{\mathcal{X}^{n+1}} h(x_{0:n})p(x_{0:n} \mid y_{1:n})\mathrm{d}x_{0:n}, \tag{28}$$

for any $N \geq 2$ in Algorithm 4. Here, $\tilde{X}_{0:n}^{(j)}, j = 1, 2, \ldots, M$ correspond to trajectories simulated by iterating Algorithm 8.

Chopin and Singh [2015] showed that (under multinomial resampling) the asymptotic variance of the estimator (28) is smaller with backward sampling than with ancestor tracing. This suggests that backward sampling should always be used with Algorithm 8, if the transition densities $(M_k)_{1\leq k\leq n}$ are tractable. Indeed, in practice the difference between the methods is typically strikingly obvious, and the Markov chains sampled using backward sampling often exhibit far better mixing than with ancestor tracing [cf. Chopin and Singh 2015, Section 6; Lindsten and Schön 2012, Section 4]. Despite this, ancestor tracing is still relevant when the densities $(M_k)_{1\leq k\leq n}$ are intractable or expensive to evaluate.

**Algorithm 6** AncestorTracing($X_{0:n}^{(1:N)}, A_{0:n-1}^{(1:N)}, \tilde{B}_n$)

```
for $k = n-1, n-2, \ldots, 0$ do
    Set $\tilde{B}_k = A_{k-1}^{(\tilde{B}_{k+1})}$.
end for
return $\tilde{B}_{0:n}$
```

**Algorithm 7** BACKWARDSAMPLING($X_{0:n}^{(1:N)}, A_{0:n-1}^{(1:N)}, \tilde{B}_n$)

```
for $k = n-1, n-2, \dots, 0$ do
    for $i = 1, 2, \dots, N$ do
        If $k \geq 1$, set $\mathbf{X}_k^{(i)} = (X_{k-1}^{(A_{k-1}^{(i)})}, X_k^{(i)})$; otherwise set $\mathbf{X}_k^{(i)} = X_k^{(i)}$.
        Compute $\omega_k^{(i)} = G_k(\mathbf{X}_k^{(i)}) G_{k+1}(X_k^{(i)}, X_{k+1}^{(\tilde{B}_{k+1})}) M_{k+1}(X_{k+1}^{(\tilde{B}_{k+1})} \mid X_k^{(i)})$.
    end for
    Simulate $\tilde{B}_k \sim \mathrm{Categ}(\omega_k^{(1:N)})$.
end for
return $\tilde{B}_{0:n}$
```

**Algorithm 8** CPF-UPDATE($\mathcal{FK}_{0:n}, N, X_{0:n}^*, B_{0:n}$, TRACEMETHOD)

```
$(X_{0:n}^{(1:N)}, W_{0:n}^{(1:N)}, A_{0:n-1}^{(1:N)}) \leftarrow$ CPF($\mathcal{FK}_{0:n}, N, X_{0:n}^*, B_{0:n}$)
$(X_{0:n}^{(\tilde{B}_{0:n})}, \tilde{B}_{0:n}) \leftarrow$ TRACEBACK($X_{0:n}^{(1:N)}, W_n^{(1:N)}, A_{0:n-1}^{(1:N)}$, TRACEMETHOD)
return $(X_{0:n}^{(\tilde{B}_{0:n})}, \tilde{B}_{0:n})$
```

## 4.4 Joint parameter and state inference with particle Markov chain Monte Carlo methods

We conclude this chapter by reviewing the particle marginal Metropolis-Hastings (PMMH) and particle Gibbs (PG) [Andrieu, Doucet, and Holenstein 2010], which are PMCMC methods for joint parameter and state inference, that is, the inference of the joint posterior $p(\theta, x_{0:n} \mid y_{1:n})$. In other words, we now assume that the SSM depends on parameters $\theta \sim p_\theta(\cdot)$ and we are interested in inferring them together with the states $x_{0:n}$.

The joint posterior of the parameters and the states has the form

$$p(\theta, x_{0:n} \mid y_{1:n}) \propto \kappa^{(\theta)}(x_{0:n}, y_{1:n}) p_\theta(\theta), \tag{29}$$

where the FK model now depends on the parameters $\theta$ and satisfies (24) for all $\theta$, that is,

$$\begin{aligned} \kappa^{(\theta)}(x_{0:n}, y_{1:n}) &= p(x_{0:n}, y_{1:n} \mid \theta) \\ &= M_0^{(\theta)}(x_0) G_0^{(\theta)}(x_0) \prod_{k=1}^{n} M_k^{(\theta)}(x_k \mid x_{k-1}) G_k^{(\theta)}(x_{k-1}, x_k). \end{aligned}$$

The superscripts by $\theta$ in the transitions and potential functions of the FK model signify that they may now depend on the parameters.

Algorithm 9 details the PMMH targeting the FK model $\mathcal{FK}_{0:n}^{(\theta)} := (M_{0:n}^{(\theta)}, G_{0:n}^{(\theta)})$ with $N$ particles and $M$ iterations, with starting value $\theta^{(0)}$ and a proposal distribution $q$ for $\theta$. Lines 1–3 constitute the initialisation of the algorithm, where the initial trajectory $\tilde{X}_{0:n}^{(0)}$ and initial normalisation constant $\hat{Z}^{(0)}$ are obtained by running the particle filter (Algorithm 3) together with ancestor tracing (Algorithm 5 with TRACEMETHOD = ANCESTORTRACING). The computation of $\hat{Z}^{(0)}$

uses (20b), where the estimate $\hat{p}(y_{1:n})$ now implicitly depends on the value of $\theta^{(0)}$ through the weights $W_{0:n}^{(1:N)}$. Each iteration (lines 5–15) then corresponds to a Metropolis-Hastings step where first a proposal $\theta^*$ for $\theta$ is simulated from $q$. Then, the proposed normalising constant estimate $\hat{Z}^*$ and trajectory $\tilde{X}_{0:n}^*$ are obtained from the output of the particle filter (Algorithm 3) targeting $\mathcal{FK}_{0:n}^{(\theta^*)}$. Finally, the joint proposal $(\theta^*, \hat{Z}^*, \tilde{X}_{0:n}^*)$ is accepted or rejected based on the acceptance rate computed on line 10.

**Algorithm 9** PMMH($\mathcal{FK}_{0:n}^{(\theta)}, N, M, \theta^{(0)}, q$)

```
1: $(X_{0:n}^{(1:N)}, W_{0:n}^{(1:N)}, A_{0:n-1}^{(1:N)}) \leftarrow$ ParticleFilter($\mathcal{FK}_{0:n}^{(\theta^{(0)})}, N$)
2: $(\tilde{X}_{0:n}^{(0)}, \tilde{B}_{0:n}) \leftarrow$ Traceback($X_{0:n}^{(1:N)}, W_n^{(1:N)}, A_{0:n-1}^{(1:N)}$, AncestorTracing)
3: Compute $\hat{Z}^{(0)} = \hat{p}(y_{1:n})$ using $W_{0:n}^{(1:N)}$ in (20b).
4: for $k = 1, 2, \ldots, M$ do
5:     Simulate $\theta^* \sim q(\cdot \mid \theta^{(k-1)})$.
6:     $(X_{0:n}^{(1:N)}, W_{0:n}^{(1:N)}, A_{0:n-1}^{(1:N)}) \leftarrow$ ParticleFilter($\mathcal{FK}_{0:n}^{(\theta^*)}, N$)
7:     $(\tilde{X}_{0:n}^*, \tilde{B}_{0:n}) \leftarrow$ Traceback($X_{0:n}^{(1:N)}, W_n^{(1:N)}, A_{0:n-1}^{(1:N)}$, AncestorTracing)
8:     Compute $\hat{Z}^* = \hat{p}(y_{1:n})$ using $W_{0:n}^{(1:N)}$ in (20b).
9:     Simulate $U \sim \mathrm{Unif}(0,1)$.
10:    Compute $p = \dfrac{\hat{Z}^* p(\theta^*) q(\theta^{(k-1)} \mid \theta^*)}{\hat{Z}^{(k-1)} p(\theta^{(k-1)}) q(\theta^* \mid \theta^{(k-1)})}$.
11:    if $U < \min(1, p)$ then
12:        Set $\theta^{(k)} = \theta^*$, $\hat{Z}^{(k)} = \hat{Z}^*$, $\tilde{X}_{0:n}^{(k)} = \tilde{X}_{0:n}^*$.
13:    else
14:        Set $\theta^{(k)} = \theta^{(k-1)}$, $\hat{Z}^{(k)} = \hat{Z}^{(k-1)}$, $\tilde{X}_{0:n}^{(k)} = \tilde{X}_{0:n}^{(k-1)}$.
15:    end if
16: end for
17: return $(\theta^{(1:M)}, \tilde{X}_{0:n}^{(1:M)}, \hat{Z}^{(1:M)})$
```

The PMMH constructs proposals whose components are jointly either accepted or rejected on each iteration of the algorithm. In contrast, the PG algorithm proposes changes to the trajectory and parameter individually by approximating a Gibbs sampler. One iteration of a perfect Gibbs sampler would simulate from $p(\theta, x_{0:n} \mid y_{1:n})$ by simulating from the full conditionals of $\theta$ and $x_{0:n}$, as follows:

1. $\theta^* \sim p_{\theta \mid X_{0:n}, Y_{1:n}}(\theta \mid x_{0:n}, y_{1:n})$
2. $x_{0:n}^* \sim p_{X_{0:n} \mid \theta, Y_{1:n}}(x_{0:n} \mid \theta^*, y_{1:n})$.

However, simulating from the full conditional of $x_{0:n}$ is typically difficult. The PG algorithm works around this by approximating the draw from full conditional of $x_{0:n}$ using the CPF (Algorithm 8). The resulting algorithm is detailed in Algorithm 10. The inputs consists of the FK model, amount of particles $N$, amount of iterations $M$, a trace method and the initial value $\theta^{(0)}$.

The first two lines of the algorithm correspond to an initialisation phase where the initial trajectory $\tilde{X}_{0:n}^{(0)}$ is obtained by running the particle filter, although

**Algorithm 10** $\textsc{ParticleGibbs}(\mathcal{FK}^{(\theta)}_{0:n}, N, M, \textsc{TraceMethod}, \theta^{(0)})$

```
1: $(X^{(1:N)}_{0:n}, W^{(1:N)}_{0:n}, A^{(1:N)}_{0:n-1}) \leftarrow \textsc{ParticleFilter}(\mathcal{FK}^{(\theta^{(0)})}_{0:n}, N)$
2: $(\tilde{X}^{(0)}_{0:n}, \tilde{B}^{(0)}_{0:n}) \leftarrow \textsc{Traceback}(X^{(1:N)}_{0:n}, W^{(1:N)}_{n}, A^{(1:N)}_{0:n-1}, \textsc{TraceMethod})$
3: for $k = 1, 2, \ldots, M$ do
4:     Simulate $\theta^{(k)} \sim p_{\theta \mid x_{0:n}, y_{1:n}}(\cdot \mid \tilde{X}^{(k-1)}_{0:n}, y_{1:n})$.
5:     $(\tilde{X}^{(k)}_{0:n}, \tilde{B}^{(k)}_{0:n}) \leftarrow \textsc{CPF-Update}(\mathcal{FK}^{(\theta^{(k)})}_{0:n}, N, \tilde{X}^{(k-1)}_{0:n}, \tilde{B}^{(k-1)}_{0:n}, \textsc{TraceMethod})$
6: end for
7: return $(\theta^{(1:M)}, \tilde{X}^{(1:M)}_{0:n})$
```

in general $\tilde{X}^{(0)}_{0:n}$ could also be provided as input. Each iteration (lines 4 and 5) then corresponds to a draw from the full conditional of $\theta$, and the approximate draw from the full conditional of $x_{0:n}$. If direct sampling from the full conditional of $\theta$ is infeasible, line 4 of the algorithm can also be changed to an MCMC move for $\theta$ (such as a Metropolis-Hastings step) that leaves the full conditional of $\theta$ invariant.

Under unbiasedness of the resampling (19), the PMMH and particle Gibbs updates (single iterations of the loops in Algorithms 9 and 10) leave $p(\theta, x_{0:n} \mid y_{1:n})$ invariant. This, together with additional mild irreducibility assumptions [cf. Roberts and Smith 1994], yields the estimator

$$\frac{1}{M}\sum_{i=1}^{M} h(\theta^{(i)}, \tilde{X}^{(i)}_{0:n}) \xrightarrow{M\to\infty} \mathbb{E}[h(\theta, X_{0:n}) \mid Y_{1:n} = y_{1:n}], \tag{30}$$

for any $N \geq 2$ for both algorithms, where $h$ is a function that depends on both $\theta$ and $X_{0:n}$.

# 5 FILTERING CONDITIONALLY LINEAR GAUSSIAN STATE-SPACE MODELS

This chapter discusses state filtering for conditionally linear Gaussian state-space models (CLGSSMs) [cf. Särkkä 2013, Section 7.5; Doucet, De Freitas, and Gordon 2001, Section 24]. CLGSSMs are SSMs with latent states $L_{0:n}$ and observations $Y_{1:n}$, such that the state $L_k$ consists of components $X_k$ and $U_k$, that is $L_k = (X_k, U_k)$. The latent states and observations are assumed to be related such that

$$\begin{aligned} X_k &= T_k^{(U_k)} X_{k-1} + R_k^{(U_k)} \eta_k^{(U_k)}, \text{ and } X_0 \sim N(\mu_0^{(U_0)}, \Sigma_0^{(U_0)}) \\ Y_k &= Z_k^{(U_k)} X_k + \epsilon_k^{(U_k)}, \end{aligned} \tag{31}$$

where $\eta_k^{(U_k)} \sim N(0, Q_k^{(U_k)})$ and $\epsilon_k^{(U_k)} \sim N(0, H_k^{(U_k)})$. In other words, conditioning on the sequence $U_{0:n}$ yields an LGSSM as in Section 3.1. The dynamics of the process $U_{0:n}$ are assumed to be characterised by an initial distribution $p_{U_0}$ and transitions $(p_{U_k|U_{k-1}}(\cdot \mid u_{k-1}))_{1 \leq k \leq n}$, that is:

$$\begin{aligned} U_0 &\sim p_{U_0} \\ U_k \mid (U_{k-1} = u_{k-1}) &\sim p_{U_k|U_{k-1}}(\cdot \mid u_{k-1}). \end{aligned} \tag{32}$$

Here, we focus on the scenario where the variables $U_k$ have a finite state-space $\mathcal{U}$.

## 5.1 The Rao-Blackwellised particle filter

Suppose that we are interested in evaluating expectations of a function $f$ involving the variables $X_k$ and $U_k$ of the model defined by (31) and (32), with respect to the distribution $p(x_{0:k}, u_{0:k} \mid y_{1:k})$. We have

$$\begin{aligned} \mathbb{E}[f(X_k, U_k) \mid Y_{1:k} = y_{1:k}] &= \sum_{u_{0:k} \in \mathcal{U}^{k+1}} \int_{\mathcal{X}^{k+1}} f(x_k, u_k) p(x_{0:k}, u_{0:k} \mid y_{1:k}) \mathrm{d}x_{0:k} \\ &= \sum_{u_{0:k} \in \mathcal{U}^{k+1}} p(u_{0:k} \mid y_{1:k}) \int_{\mathcal{X}} f(x_k, u_k) p(x_k \mid u_{0:k}, y_{1:k}) \mathrm{d}x_k, \end{aligned} \tag{33}$$

where we have used the factorisation

$$p(x_{0:k}, u_{0:k} \mid y_{1:k}) = p(x_{0:k} \mid u_{0:k}, y_{1:k}) p(u_{0:k} \mid y_{1:k}).$$

Given an empirical approximation of $p(u_{0:k} \mid y_{1:k})$ with support points $U_{0:k}^{(i)}$ and (normalised) weights $\tilde{W}_k^{(i)}$, $i = 1, 2, \ldots, N$, we could therefore estimate (33) with

$$\sum_{i=1}^{N} \tilde{W}_k^{(i)} \int_{\mathcal{X}} f(x_k, U_k^{(i)}) p_{X_k \mid U_{0:k}, Y_{1:k}}(x_k \mid U_{0:k}^{(i)}, y_{1:k}) \mathrm{d}x_k. \tag{34}$$

Note that (34) involves integrating with respect to the Gaussian distributions $p_{X_k \mid U_{0:k}, Y_{1:k}}(x_k \mid U_{0:k}^{(i)}, y_{1:k})$ that can be computed analytically using the Kalman filter (7) of Section 3.1. The integral in (34) can therefore be analytically computed if the form of the function $f$ permits this. Otherwise, numerical integration methods such as Gauss-Hermite quadrature [cf. Särkkä 2013, Section 6.3] can be used.

A Rao-Blackwellised particle filter (RBPF) [Akashi and Kumamoto 1977[1]; see also Särkkä 2013, Section 7.5] — also known as a 'mixture Kalman filter' [Chen and Liu 2000] — uses a particle filter to approximate the distribution $p(u_{0:k} \mid y_{1:k})$, and the Kalman filter in the evaluation of the integrals related to the conditional distribution of $x_k$ (or even $x_{0:k}$) given $u_{0:k}$ and $y_{1:k}$. Similarly as in (18), the weights of a particle filter targeting $p(u_{0:k} \mid y_{1:k})$ satisfy

$$W_k^{(i)} \propto \frac{p_{Y_k \mid U_{0:k}, Y_{1:k-1}}(y_k \mid U_{0:k}^{(i)}, y_{1:k-1}) p_{U_k \mid U_{k-1}}(U_k^{(i)} \mid U_{k-1}^{(i)})}{q_k(U_k^{(i)} \mid U_{1:k-1}^{(i)}, y_{1:k})} W_{k-1}^{(i)}, \tag{35}$$

where we have used the fact that $U_k$ is conditionally independent from $Y_{1:k-1}$ given $U_{k-1}$, and $q_k$ is a proposal distribution for $U_k$ that depends on the full history $U_{1:k-1}$ and (possibly) $Y_{1:k}$.

Algorithm 11 details a basic RBPF targeting the model consisting of (31) and (32), using proposals $q_{0:n}$ and $N$ particles [cf. Särkkä 2013, Section 7.5]. The RBPF is similar to Algorithm 1, with the latent states $U_k$ sampled instead of $X_k$. In contrast, however, the RBPF makes use of the analytical formulas available for LGSSMs once $U_{0:k}$ is conditioned on. In particular, the weight computation on line 9 and the update step on line 10 make use of (9) and the Kalman filter (7). The update step computes the mean and covariance $S_k^{(i)}$ of the distributions $p_{X_k \mid U_{0:k}, Y_{1:k}}(x_k \mid U_{0:k}^{(i)}, y_{1:k})$ in (34) for each $i$. The output of the RBPF may be therefore readily used to compute (34) after normalisation of the weights $W_k^{(1:N)}$.

## 5.2 The one step optimal proposal distribution

The choice of the proposal distribution $q_{0:n}$ input to Algorithm 11 affects the performance of the estimator (34). When the state-space $\mathcal{U}$ of the states $U_k$ is finite,

[1] Although this early form of the 'RBPF' was published before the first particle filter of Gordon, Salmond, and Smith [1993], it did not feature a resampling operation.

**Algorithm 11** BASIC-RBPF($q_{0:n}, N$)

```
Simulate $U_0^{(i)} \sim q_0$ for $i = 1, 2, \ldots, N$.
Compute $W_0^{(i)} = p_{U_0}(U_0^{(i)}) / q_0(U_0^{(i)})$ for $i = 1, 2, \ldots, N$.
Set $\mathbf{U}_0^{(i)} = U_0^{(i)}$ for $i = 1, 2, \ldots, N$.
Set $S_0^{(i)} = (\mu_{0|0}^{(\mathbf{U_0}^{(i)})}, \Sigma_{0|0}^{(\mathbf{U_0}^{(i)})})$ for $i = 1, 2, \ldots, N$, where $\mu_{0|0}^{(\mathbf{U}_0^{(i)})} := \mu_0^{(U_0^{(i)})}$, and $\Sigma_{0|0}^{(\mathbf{U}_0^{(i)})} := \Sigma_0^{(U_0^{(i)})}$.
for $k = 1, 2, \ldots, n$ do
    Simulate $A_{k-1}^{(i)} \sim r(\cdot \mid W_{k-1}^{(1:N)})$ for $i = 1, 2, \ldots, N$.
    Simulate $U_k^{(i)} \sim q_k(\cdot \mid \mathbf{U}_{k-1}^{(A_{k-1}^{(i)})}, y_{1:k})$ for $i = 1, 2, \ldots, N$.
    Set $\mathbf{U}_k^{(i)} = (\mathbf{U}_{k-1}^{(A_{k-1}^{(i)})}, U_k^{(i)})$ for $i = 1, 2, \ldots, N$.
    Compute $W_k^{(i)} = \dfrac{p_{Y_k|U_{0:k},Y_{1:k-1}}(y_k \mid \mathbf{U}_k^{(i)}, y_{1:k-1}) p_{U_k|U_{k-1}}(U_k^{(i)} \mid U_{k-1}^{(A_{k-1}^{(i)})})}{q_k(U_k^{(i)} \mid \mathbf{U}_{k-1}^{(A_{k-1}^{(i)})}, y_{1:k})}$ for $i = 1, 2, \ldots, N$. Note that $p_{Y_k|U_{0:k},Y_{1:k-1}}(y_k \mid \mathbf{U}_k^{(i)}, y_{1:k-1}) = N(y_k; \mu^{(\mathbf{U}_k^{(i)})}, \Sigma^{(\mathbf{U}_k^{(i)})})$, where
        $$\mu^{(\mathbf{U}_k^{(i)})} := Z_k^{(U_k^{(i)})} \mu_{k|k-1}^{(\mathbf{U}_k^{(i)})},$$
        $$\Sigma^{(\mathbf{U}_k^{(i)})} := Z_k^{(U_k^{(i)})} \Sigma_{k|k-1}^{(\mathbf{U}_k^{(i)})} \big(Z_k^{(U_k^{(i)})}\big)' + H_k^{(U_k^{(i)})},$$
    with $\mu_{k|k-1}^{(\mathbf{U}_k^{(i)})}$ and $\Sigma_{k|k-1}^{(\mathbf{U}_k^{(i)})}$ computed using (7):
        $$\mu_{k|k-1}^{(\mathbf{U}_k^{(i)})} = T_k^{(U_k^{(i)})} \mu_{k-1|k-1}^{(\mathbf{U}_{k-1}^{(A_{k-1}^{(i)})})}$$
        $$\Sigma_{k|k-1}^{(\mathbf{U}_k^{(i)})} = T_k^{(U_k^{(i)})} \Sigma_{k-1|k-1}^{(\mathbf{U}_{k-1}^{(A_{k-1}^{(i)})})} \big(T_k^{(U_k^{(i)})}\big)' + R_k^{(U_k^{(i)})} Q_k^{(U_k^{(i)})} \big(R_k^{(U_k^{(i)})}\big)'.$$
    Compute $S_k^{(i)} = (\mu_{k|k}^{(\mathbf{U_k}^{(i)})}, \Sigma_{k|k}^{(\mathbf{U_k}^{(i)})})$ for $i = 1, 2, \ldots, N$ using $y_k$ and the update step in (7).
end for
Return $U_{0:n}^{(1:N)}, W_{0:n}^{(1:N)}, A_{0:n-1}^{(1:N)}, S_{0:n}^{(1:N)}$
```

an appealing candidate for the proposal distribution $q_{0:n}$ can be formed from the one step optimal proposals [Liu and Chen 1998; Akashi and Kumamoto 1977]

$$\begin{aligned} q_k(u_k \mid U^{(i)}_{0:k-1}, y_{1:k}) &:= p_{U_k \mid U_{0:k-1}, Y_{1:k}}(u_k \mid U^{(i)}_{0:k-1}, y_{1:k}) \\ &\propto p_{Y_k \mid U_{0:k}, Y_{1:k-1}}(y_k \mid u_k, U^{(i)}_{0:k-1}, y_{1:k-1}) p_{U_k \mid U_{k-1}}(u_k \mid U^{(i)}_{k-1}), \end{aligned} \tag{36}$$

the unnormalised probabilities of which can be evaluated for all $u_k \in \mathcal{U}$. Note that the term $p_{Y_k \mid U_{0:k}, Y_{1:k-1}}(y_k \mid u_k, U^{(i)}_{0:k-1}, y_{1:k-1})$ above can be evaluated as in line 9 of Algorithm 11. When the proposals (36) are used with Algorithm 11, the weights of the particles reduce to the normalising constant of (36) as can be seen from (35).

## 5.3 The discrete particle filter with optimal resampling

Line 7 of Algorithm 11 simulates a single outcome $U^{(i)}_k$ given the particle $\mathbf{U}^{(A^{(i)}_{k-1})}_{k-1}$ for each $i$. Each new particle $\mathbf{U}^{(i)}_k$ is then constructed by setting $\mathbf{U}^{(i)}_k = (\mathbf{U}^{(A^{(i)}_{k-1})}_{k-1}, U^{(i)}_k)$, that is, combining the outcome with the ancestor particle $\mathbf{U}^{(A^{(i)}_{k-1})}_{k-1}$. Therefore, if the proposal for $U^{(i)}_k$ spans $\mathcal{U}$, each $\mathbf{U}^{(i)}_k$ is chosen among $|\mathcal{U}|$ candidates.

To obtain a more efficient method, the proposal and resampling steps of Algorithm 11 may be modified. This may be done by performing an 'exhaustive one step lookahead' at each time step, which constructs all possible 'future particles' $\hat{\mathbf{U}}^{(1:|\mathcal{U}|N)}_k$ based on the particles at time $k-1$, $\mathbf{U}^{(1:N)}_{k-1}$, and then uses a resampling step to select again $N$ particles $\mathbf{U}^{(1:N)}_k$ among $\hat{\mathbf{U}}^{(1:|\mathcal{U}|N)}_k$. This explores the state-space $\mathcal{U}^{k+1}$ more effectively, since now each $\mathbf{U}^{(i)}_k$ is chosen among $|\mathcal{U}|N$ candidates.

A method fitting the description above is known as the discrete particle filter (DPF) [Fearnhead 1998; see also Whiteley, Andrieu, and Doucet 2010]. Algorithm 12 details the DPF using $N$ particles, targeting the model comprised of (31) and (32). The main difference to Algorithm 11 is that the DPF does not simulate particles from a proposal distribution, but exhaustively constructs all possible particles $u^{(i)}_{0:k}, i = 1, 2, \ldots, |D_k|$ given particles that were chosen in the previous resampling operation. The Kalman filter is then used to compute the filtered means and covariances given each possible particle on line 10).

Algorithm 12 uses a lowercase $u$ for the particles to signify that they are not simulated from a proposal; the only randomness in them comes from the resampling operation, which occurs on lines 7–8. The resampling corresponds to the optimal resampling algorithm of Fearnhead and Clifford [2003], which is unbiased and optimal in the sense that it minimises a squared error loss function. Furthermore, the algorithm guarantees that each resampled particle is unique.

Resampling occurs once the number of possible particles at time point $k-1$, $|D_{k-1}|$, grows larger than the number of particles $N$, and entails computing a con-

**Algorithm 12** DPF($N$)

```
Set $D_0 = \mathcal{U}$.
Set $\mu_{0|0}^{(u_0^{(i)})} = \mu_0^{(u_0^{(i)})}$ and $\Sigma_{0|0}^{(u_0^{(i)})} = \Sigma_0^{(u_0^{(i)})}$ for all $u_0^{(i)} \in D_0, i = 1, 2, \ldots, |D_0|$.
Set $\mathbf{S}_0 = (\mu_{0|0}^{(u_0^{(1:|D_0|)})}, \Sigma_{0|0}^{(u_0^{(1:|D_0|)})})$.
Compute $W_0^{(u_0^{(i)})} = p_{U_0}(u_0^{(i)})$ for $i = 1, 2, \ldots, |D_0|$.
Normalise $W_0^{(u_0^{(1:|D_0|)})}$ to obtain $\tilde{W}_0^{(u_0^{(1:|D_0|)})}$. Set $\tilde{\mathbf{W}}_0 = \tilde{W}_0^{(u_0^{(1:|D_0|)})}$.
for $k = 1, 2, \ldots, n$ do
    If $|D_{k-1}| \leq N$, set $c_{k-1} = \infty$. Otherwise find $c_{k-1}$ such that
        $\sum_{i=1}^{|D_{k-1}|} \min(1, c_{k-1}\tilde{W}_{k-1}^{(u_{0:k-1}^{(i)})}) = N.$   (37)
    Maintain the $L_{k-1}$ particles in $D_{k-1}$ that have weights strictly greater than $1/c_{k-1}$ ('maintained partition'). Use systematic resampling to select $\min(N, |D_{k-1}|) - L_{k-1}$ from the remaining $|D_{k-1}| - L_{k-1}$ ('resampling partition'). Denote by $D_{k-1}^{'}$ the total $\min(N, |D_{k-1}|)$ particles that remain.
    Set $D_k = D_{k-1}^{'} \times \mathcal{U}$.
    Compute $\mu_{k|k}^{(u_{0:k}^{(i)})}$ and $\Sigma_{k|k}^{(u_{0:k}^{(i)})}$ for all $u_{0:k}^{(i)} \in D_k$, $i = 1, 2, \ldots, |D_k|$ using the Kalman filter. Set $\mathbf{S}_k = (\mu_{k|k}^{(u_{0:k}^{(1:|D_k|)})}, \Sigma_{k|k}^{(u_{0:k}^{(1:|D_k|)})})$.
    Compute
        $W_k^{(u_{0:k}^{(i)})} = p_{Y_k|Y_{1:k-1},U_{0:k}}(y_k \mid y_{1:k-1}, u_{0:k}^{(i)}) p_{U_k|U_{k-1}}(u_k^{(i)} \mid u_{k-1}^{(i)}) \frac{\tilde{W}_{k-1}^{(u_{0:k-1}^{(i)})}}{\min(1, c_{k-1}\tilde{W}_{k-1}^{(u_{0:k-1}^{(i)})})},$
    for $i = 1, 2, \ldots, |D_k|$.
    Normalise $W_k^{(u_{0:k}^{(1:|D_k|)})}$ to obtain $\tilde{W}_k^{(u_{0:k}^{(1:|D_k|)})}$. Set $\tilde{\mathbf{W}}_k = \tilde{W}_k^{(u_{0:k}^{(1:|D_k|)})}$.
end for
Return $D_{0:n}, \tilde{\mathbf{W}}_{0:n}, \mathbf{S}_{0:n}$
```

stant, $c_{k-1}$, satisfying (37). The value of $c_{k-1}$ acts as a cutoff for the weights and divides the particles to two partitions: the maintained partition and the resampling partition. The particles in the maintained partition are kept with their original weights, and the particles in the resampling partition are resampled using systematic resampling (Algorithm 2). An algorithm for computing the constant $c_{k-1}$ is presented in Fearnhead and Clifford [2003, Appendix C].

If resampling does not occur (that is, $c_{k-1} = \infty$ is set) all particles are maintained, that is, the resampling partition has zero particles. Note that in theory, if $N$ were set large enough, resampling would never occur and $D_k$ would equal $\mathcal{U}^{k+1}$ at time $k$, that is, all possible particles would have been constructed and the results of the DPF would be exact.

The output of the DPF consists of the particles $D_{0:n}$, their weights $\tilde{\mathbf{W}}_{0:n}$, and the filtered means and covariances $\mathbf{S}_{0:n}$. For $k = 1, 2, \ldots, n$, the estimator

$$\sum_{i=1}^{|D_k|} \tilde{W}_k^{(u_{0:k}^{(i)})} \int_X f(x_k, u_k^{(i)}) p_{X_k|U_{0:k},Y_{1:k}}(x_k \mid u_{0:k}^{(i)}, y_{1:k}) \mathrm{d}x_k \tag{38}$$

estimates (33) and may be computed using the outputs $D_k$, $\tilde{\mathbf{W}}_k$ and $\mathbf{S}_k$.

Typically, for $k$ moderately large, $|D_k|$ in (38) will equal $|\mathcal{U}|N$. If $|\mathcal{U}|$ is large, an approximation with just $N$ particles may also be obtained by running the resampling of Fearnhead and Clifford [2003] once more, to obtain $N$ particles consisting of the particles in the maintained partition (with original weights), and the particles resampled from the resampling partition (with weights $1/c_k$ each). An estimate of the form (34) may be then used with the particles and filtered means and covariances corresponding to the particles output by the resampling.

# 6 RESEARCH CONTRIBUTION

This chapter summarises the research contribution of this thesis. We first discuss the methodological contributions of Articles II and IV that develop new particle filtering methods. Then, we discuss the applied contributions of Articles I and III that involve development of NSSMs, approximations and use of existing methodology for NSSMs.

## 6.1 Methodological contributions

### Article II: smoothing for general state-space models with 'diffuse' initial distributions

Article II focuses on the smoothing problem for general state-space models with 'diffuse' initial distributions, that is, models where the variability of the initial distribution $M_0$ in the Feynman–Kac model (23) is high in comparison to the first marginal smoothing distribution of the SSM.

For such models, direct use of the conditional particle filter (CPF) (Algorithm 4) – even with the traceback implemented using the often efficient backward sampling (Algorithm 7) – leads to suboptimal mixing of the early state variables in the Markov chain. Intuitively speaking, this occurs since the initial particles drawn from the diffuse $M_0$ will likely fall into unlikely (that is, low potential) regions of the state-space and are thus rarely selected in the traceback. Figure 2 (adapted from Article II) shows how this phenomenon occurs even for a simple LGSSM, the 'noisy AR(1)' model:

$$\begin{aligned} &x_k \sim N(\rho x_{k-1}, \sigma_x^2) \text{ for } 1 \leq k \leq n-1 \text{ and } x_0 \sim N(0, \sigma_0^2) \\ &y_k \sim N(x_k, \sigma_y^2) \text{ for } 0 \leq k \leq n-1, \end{aligned} \tag{39}$$

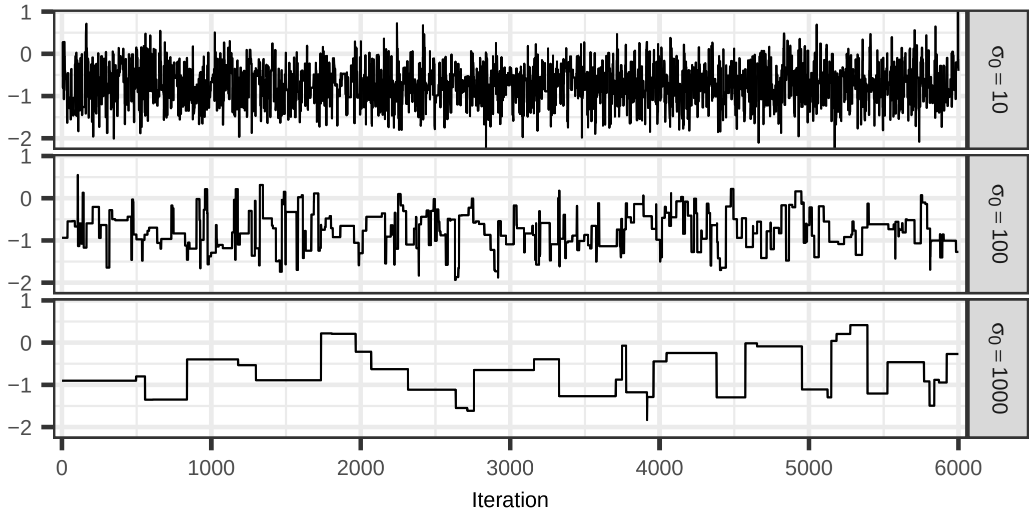


FIGURE 2 Figure adapted from Article II. Traceplots of the samples from the first marginal smoothing distribution $p(x_0 \mid y_{0:n-1})$ of the noisy AR(1) model (39) with $\sigma_0 \in \{10, 100, 1000\}$.

represented using the FK model

$$\begin{aligned} M_0(\cdot) &= N(0, \sigma_0^2) \\ M_k(\cdot \mid x_{k-1}) &= N(\cdot; \rho x_{k-1}, \sigma_x^2) \text{ for } 1 \le k \le n-1 \\ G_0(x_0) &= N(y_0; x_0, \sigma_y^2) \\ G_k(x_{k-1}, x_k) &= N(y_k; x_k, \sigma_y^2) \text{ for } 1 \le k \le n-1 \end{aligned}$$

where $\rho$, $\sigma_x$, $\sigma_y$ and $\sigma_0$ are parameters. To draw Figure 2, we set $\rho = 0.8$, $\sigma_x = \sigma_y = 0.5$ and iterated the conditional particle filter with backward sampling 6000 times with $N = 16$ for each $\sigma_0 \in \{10, 100, 1000\}$. A single data set with fifty observations simulated from the model (39) with the above parameters and $x_0 = 0$ was used in all simulations (see Article II for more details).

To avoid such problems, we introduce a new 'auxiliary initialisation conditional particle filter' (AI-CPF) that avoids direct sampling from the diffuse $M_0$. Instead, we employ a Markov transition $Q$ that is $M_0$-reversible. Such reversible transitions are easy to define (see Section 3 of Article II). In particular, we focus on the case where $M_0$ is an improper distribution, namely the uniform distribution on $\mathbb{R}^d$, in which case $Q$ can be a symmetric Gaussian random walk.

Furthermore, based on advances in the adaptive MCMC literature [Andrieu and Thoms 2008; Haario, Saksman, and Tamminen 2001], we refine the AI-CPF to allow for choosing the tuning parameters of $Q$ automatically to facilitate efficient mixing. In the case mentioned above, with $Q(\cdot \mid x) = N(\cdot; x, c\Sigma)$, our adaptive AI-CPF tunes (i) the scale $c$ such that a desired target acceptance rate is reached, and (ii) the shape $\Sigma$ such that it converges to the covariance of the first marginal smoothing distribution. We also demonstrate how the AI-CPF can be readily embedded within a particle Gibbs algorithm (Algorithm 10) and provide a target acceptance rate heuristic that eliminates the need for setting any tuning parameters to use our method.

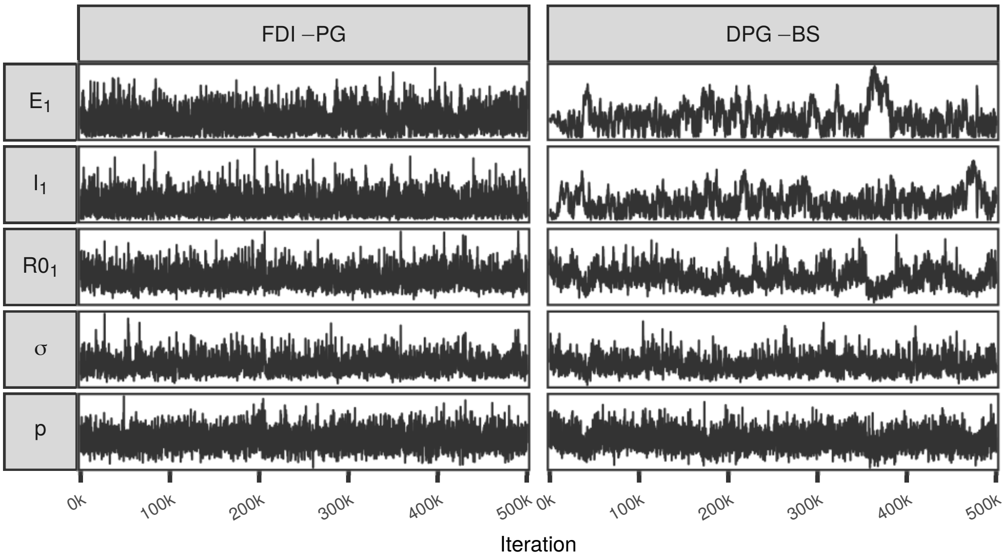


FIGURE 3 Figure from Article II. Traceplots for initial states $(E_1, I_1, R0_1)$ and parameters $(\sigma, p)$ of a stochastic SEIR model. Left: an adaptive variant of the developed AI-CPF. Right: a particle Gibbs algorithm that treats the initial state as a parameter. See Article II for more details.

In our concluding example with an epidemic model (a stochastic susceptible-exposed-infected-removed (SEIR) model) and a real data set, we observe substantially better mixing compared to a particle Gibbs method that treats the initial state as a parameter (see Figure 3).

**Article IV: smoothing for state-space models with weakly informative observations and/or slowly-mixing dynamic models**

Article IV considers the smoothing problem for state-space models with weakly informative observations and/or slowly-mixing dynamic models. Such models arise for example with fine discretisations of continuous-time FK path integral models [cf. Del Moral and Miclo 2000].

In this setting, the conditional particle filter with backward sampling (CPF-BS) (Algorithm 8 with TRACEMETHOD = BACKWARDSAMPLING) and multinomial resampling suffers from two problems. First, with weakly informative observations, multinomial resampling introduces excess variance. Second, in the case of a slowly-mixing dynamic model, the traceback using backward sampling essentially degenerates to ancestor tracing (Algorithm 6).

Inspired by the recent findings in [Chopin, Singh, et al. 2022], we avoid the issue with multinomial resampling by introducing two new 'conditional' resampling algorithms (briefly mentioned in Section 4.3) that are suitable for the CPF in the weakly informative context. We further provide a sufficient condition that guarantees the validity of the CPF with a 'generic' conditional resampling algorithm.

To address the degeneracy issue with backward sampling, we develop a new 'conditional particle filter with bridge backward sampling' (CPF-BBS) that

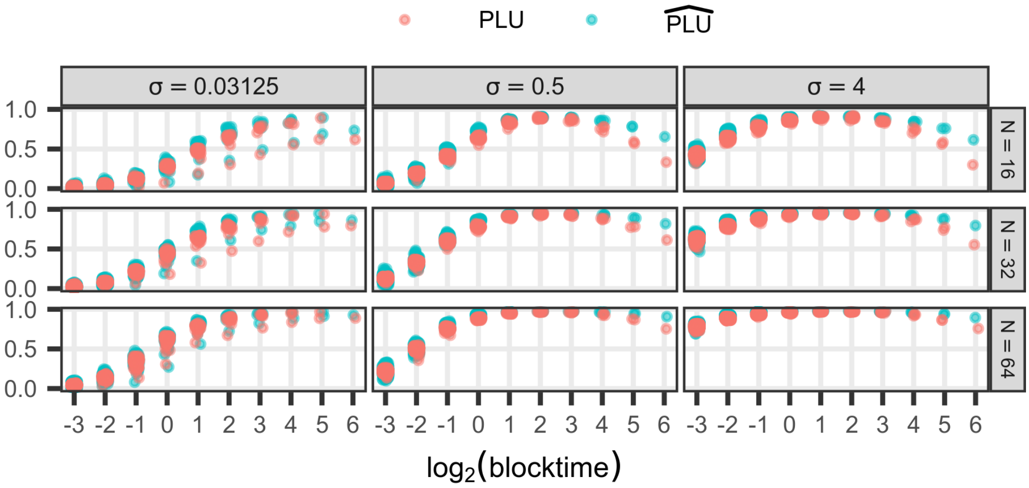


FIGURE 4 Figure adapted from Article IV. Empirical comparison of the agreement of the developed estimator $\widehat{\text{PLU}}$ and PLU computed by iterating the CPF-BBS for a particular SSM with varying parameters $\sigma$ and numbers of particles $N$. Each point depicts the values of $\widehat{\text{PLU}}$ and PLU in a particular block of the blocking sequence. The value on the horizontal axis parameterises the blocking sequence used.

updates the latent states $x_{0:n}$ in 'blocks' $x_{T_{i-1}:T_i}$ for $i = 1, 2, \ldots, K$ during backward sampling. The block bounds $[T_{i-1}, T_i]$ are parameterised by a 'blocking sequence' $0 = T_0 < T_1 < T_2 < \ldots < T_K = n$ specified by the user. The blocked updates require that conditional densities related to the dynamic model can be evaluated and sampled from. The CPF-BBS may be seen as a generalisation of the CPF-BS, since choosing the blocking sequence $T_{0:n} = 0{:}n$ yields the CPF-BS.

The blocking sequence is an important tuning parameter of the CPF-BBS that affects the mixing of the output Markov chain, and for efficient mixing, it should be chosen based on the model of interest. To make this feasible in practice, we develop and empirically verify a computationally inexpensive heuristical procedure for selecting the blocking sequence prior to iterating the CPF-BBS.

The procedure seeks for a blocking sequence that maximises the so called 'probability of lower boundary updates' (PLU), which is equal to the probability that the CPF-BBS update refreshes the value at a particular block lower boundary $T_{i-1}$. Based on our empirical results, PLU is inversely related to the integrated autocorrelation time of the Markov chain. We develop an approximate estimator for PLU, denoted by $\widehat{\text{PLU}}$, which may be evaluated without iterating the CPF-BBS. Figure 4 compares $\widehat{\text{PLU}}$ to PLU computed by iterating the CPF-BBS (see Article IV for more details).

Our concluding experiment applies the CPF-BBS to a smoothing problem related to movement modelling (see Figure 5). The experiment models the movement of an object that has a preference for moving in certain types of terrain. The object has been observed at the blue crosses. The preference for terrain is modelled using the potential functions of the FK model and is depicted in the background map. Here, there are a finite number of terrain types and the movement

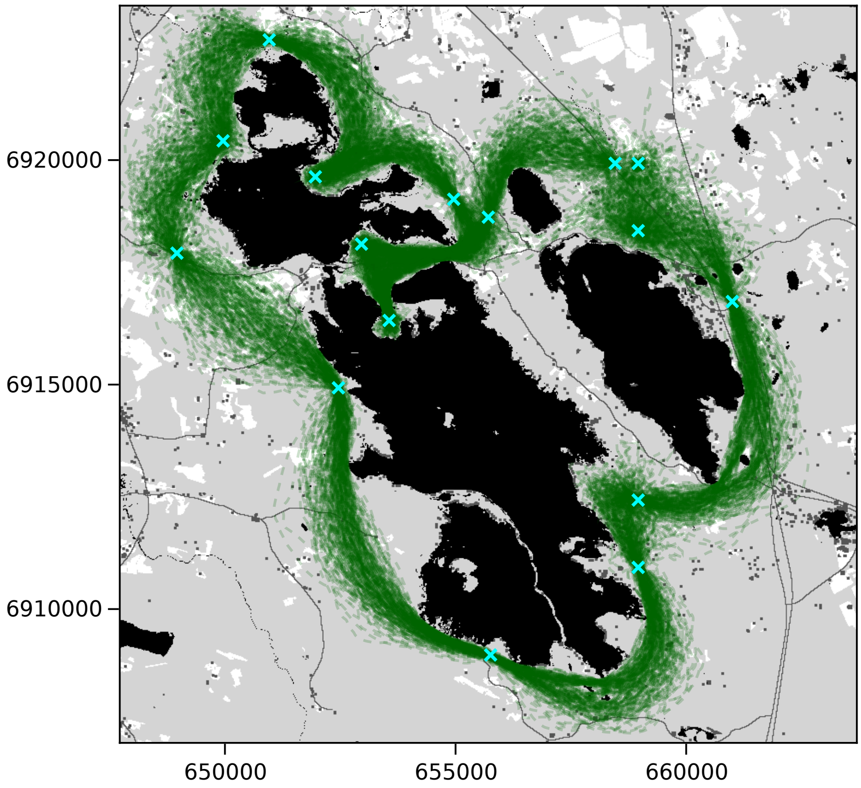


FIGURE 5 Figure adapted from Article IV. The green lines correspond to 250 trajectories simulated from the full smoothing distribution of the model that conditions on the observations and the terrain preference. Blue crosses are observed locations and the background map depicts the preference of terrain with lighter meaning higher preference.

is constrained on land. The experiment demonstrates a substantial efficiency gain over CPF-BS in the weakly informative regime (see Figure 6).

## 6.2 Applied contributions

### Article I: predicting leukocyte counts using non-linear state-space models

Article I develops new NSSMs for predicting leukocyte (white blood cell) counts of children diagnosed with acute lymphoblastic leukaemia (ALL). In the final treatment phase of ALL, patients receive so called ‘maintenance therapy’ for a period of up to two years [Schmiegelow et al. 2014]. Maintenance therapy consists of low-dose chemotherapy involving daily oral 6-mercaptopurine and weekly methotrexate. The doses of these chemotherapeutic drugs are adjusted weekly or biweekly based on a ‘target level’ for the leukocyte counts of the patients, which motivates the interest in the predictive modelling of the leukocyte counts.

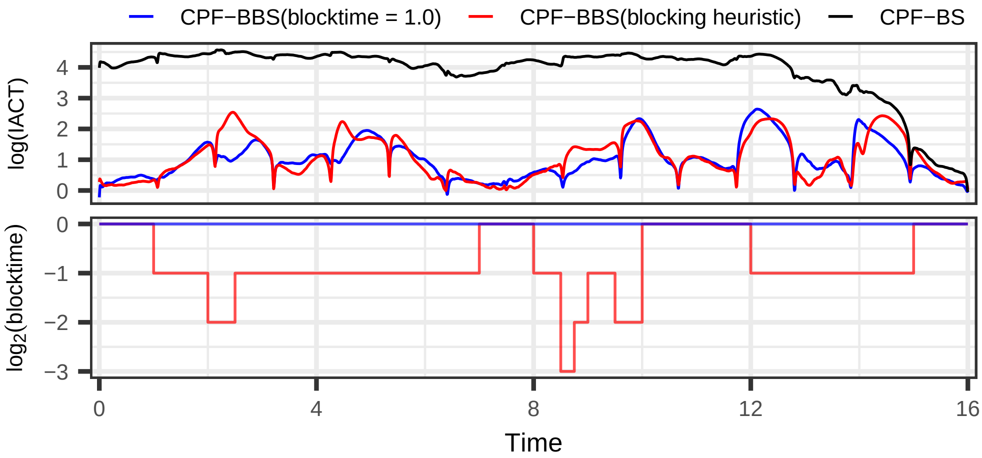


FIGURE 6 Figure adapted from Article IV. Top: Logarithm of the integrated autocorrelation time for the horizontal location of the object with respect to time when the smoothing distribution shown in Figure 5 was simulated using CPF-BS (black), hand-tuned CPF-BBS (blue) and CPF-BBS with blocking sequence obtained automatically using heuristical procedure based on PLU (red). Bottom: size of blocks with respect to time for the variants of CPF-BBS used.

The work of Jayachandran et al. [2014] developed a mechanistic model for the leukocyte counts based on a compartment model of Friberg et al. [2002], which is considered to be the ‘gold standard’ approach for modelling the production of neutrophils under chemotherapy [Craig 2017]. The models of Jayachandran et al. [2014] and Friberg et al. [2002] consists of a system of ordinary differential equations.

We simplify these models and introduce noise in their dynamics, which leads to state-space models that arise as approximations to non-linear stochastic differential equations. One of our models features a stochastic volatility [cf. Taylor 2007] type component that is used to incorporate the measured C-reactive protein (CRP) to the model (see Figure 7). The CRP is a surrogate for a common treatment adversity, infections, which may affect the leukocyte counts. We compare our models to the model of Jayachandran et al. [2014] using time series cross-validation and find that our simplified models appear more robust and are competitive in terms of the predictive performance.

The model estimation and prediction is an application of the extended Kalman filter (14) discussed in Section 3.2, with maximum a posteriori estimates obtained for the model parameters using numerical optimisation.

### Article III: using presence-only citizen science data to estimate the number and locations of animal territories

Article III is an ecological application that develops a modelling framework that can be used to estimate the number and locations of animal territories using presence-only citizen science data, under assumptions on the typical territory life-

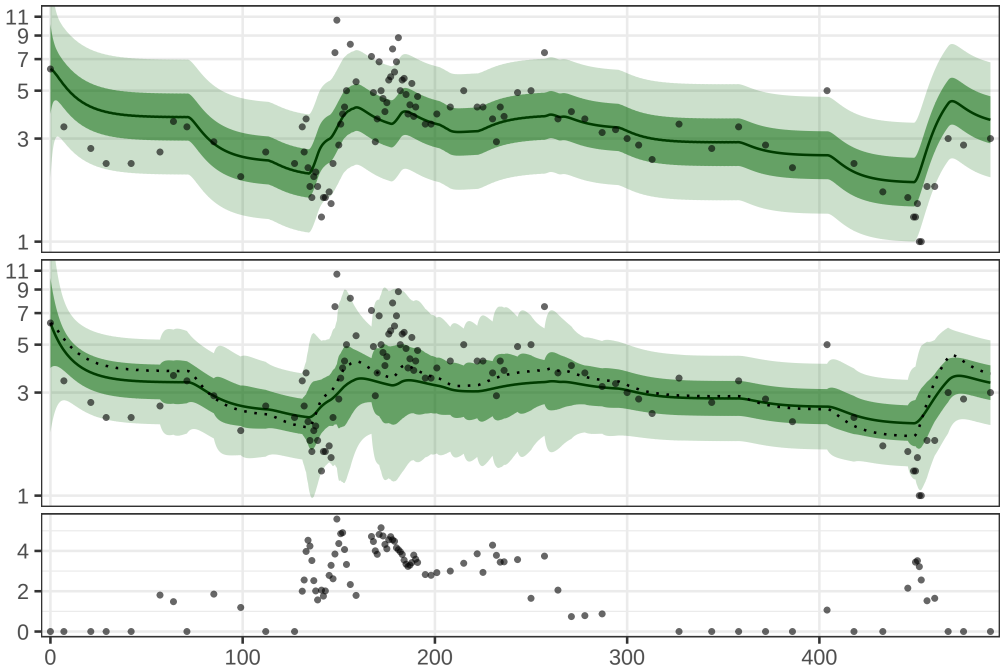


FIGURE 7 Figure from Article I. Mean leukocyte count and approximate prediction intervals (50%, 90%) based on the fits of two non-linear state-space models (top and middle) to the measured leukocyte counts (black points) of a single patient during maintenance therapy. The horizontal axis shows the time in days. The middle plot shows a model fit with the stochastic volatility component and the top plot without. The mean from the model in the top plot appears as a dotted line in the middle plot. The bottom plot shows the $\log(x + 1)$-transformed CRP measurements.

time and size. We apply the framework for identifying wolf territories in Finland.

In particular, our framework features a tracking model [cf. Goodman, Mahler, and Nguyen 1997] that consists of a birth and death submodel for the appearance and removal of the animal territories, and an observation submodel that links the citizen science observations to the territories.

A key feature of our observation submodel is that it can account for temporally and spatially varying observation intensities that are common for citizen science observation processes. The model we develop extends the similar models of Särkkä, Vehtari, and Lampinen [2007] and Vihola [2007] for multiple target tracking by allowing for the spatial inhomogeneity in the observation model.

We apply the developed framework to analyse citizen-made observations of wolves from April 2019 to March 2020 in Finland. The data come from a digital large carnivore observation database named ‘Tassu’ (see [Natural Resources Institute Finland 2022] for some of the latest data).

We use the discrete particle filter (Algorithm 12) discussed in Section 5.3 with approximate Kalman filter updates to infer the filtering distribution of the number and locations of Finnish wolf territories in March 2020. The obtained results resemble those reported in the annual wolf population assessment by the Natural Resources Institute Finland in March 2020 (see Figure 8). This is promis-

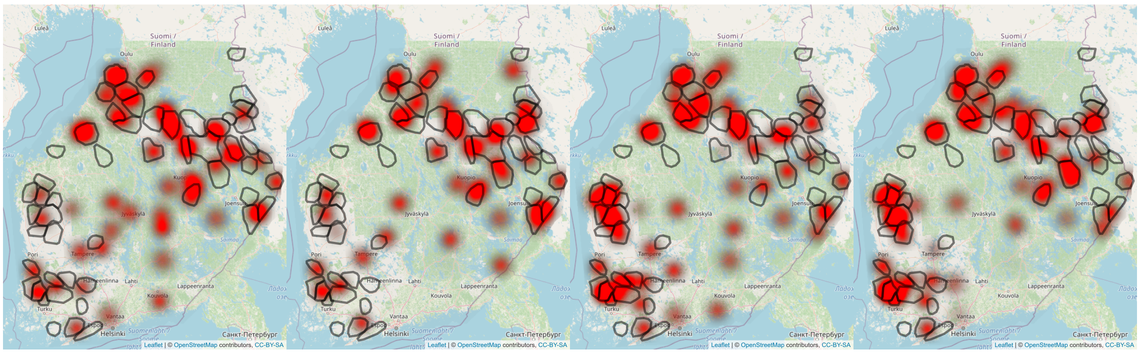


FIGURE 8 Figure from Article III. Wolf territories (red) found by four model variants (left to right) using a year of citizen-collected wolf observations from April 2019 to March 2020 in Finland. The polygons outlined in black are wolf territories found in the annual wolf population assessment of March 2020 by the Natural Resources Institute Finland.

ing, since the wolf population assessments are expected to be quite accurate as they are based on more data: citizen-collected observations, non-invasive genetic samples, tracks of GPS-collared wolves and known wolf mortality.

# 7 DISCUSSION

This thesis consists of the methodological Articles II and IV that develop new particle filters, and of Articles I and III that develop new NSSMs and apply them in challenging practical problems.

The applications of NSSMs in Articles I and III analyse data sets of the type that are increasingly common and important. In ecology and the environmental sciences, for example, citizen science data are collected by volunteers at a global scale that is unattainable by traditional research teams [Silvertown 2009]. In a similar vein, clinical data and its statistical modelling hold great promise in precision medicine, where treatments and dosage tailored for the individual patient are sought [Fröhlich et al. 2018]. A common theme with citizen science data and clinical data is that they both possess great future potential, but are generated by processes that are often complex and noisy, complicating the use of the data in practice. To unlock the potential in these data, the complexities need to be accounted for by careful modelling of the data generating process. Articles I and III demonstrate that NSSMs provide flexible tools for this: the developed NSSMs in Article I model the individual responses of patients to chemotherapy, and the model in Article III accounts for the spatially and temporally varying intensity of citizen-collected wolf observations.

The (adaptive) AI-CPF of Article II complements the literature on the inference of SSMs by providing an easy-to-use and efficient method for general SSMs with diffuse initial distributions. To the best of our knowledge, the diffuse initialisation of state-space models has been mostly used in the context of LGSSMs [cf. Durbin and Koopman 2012, Section 5]. The AI-CPF may be seen as an instance of a general 'pseudo-observation' augmentation scheme for particle MCMC [Fearnhead and Meligkotsidou 2016], which is based on conjugacy between the associated probability distributions. In the context of diffuse initialisation, the AI-CPF is simple to implement and use, since it is not constrained by conjugacy and does not require the specification of the (tuning) parameters related to the conjugate probability distributions. Instead, the AI-CPF relies on Markov transitions that are reversible with respect to the initial distribution of the FK model. Furthermore, the adaptive AI-CPF uses adaptive MCMC methods to reduce the number of tuning parameters to a single target acceptance rate, for which a heuristic is provided, as well.

The CPF-BBS introduced in Article IV is a conditional particle filter that is efficient with SSMs that have slowly-mixing dynamic models and/or uninformative observations. These kinds of settings arise in particular with FK models whose dynamic model corresponds to a fine discretisation of a linear SDE. Such situations arise for instance with a log-Gaussian Cox process [Møller, Syversveen, and Waagepetersen 1998] whose driving Gaussian process has Markov dynamics, or with path-integral models [cf. Del Moral and Miclo 2000].

The CPF-BBS might find further applications in modelling animal movement [cf. Johnson et al. 2008] or in so called 'step selection analyses' that are conducted to study for example animal resource selection [cf. Thurfjell, Ciuti, and Boyce 2014] based on telemetry data. In fact, the experiment in Figure 5 — where the CPF-BBS excelled — is closely related to these fields. We suspect that it is possible to elaborate the model behind the experiment (see Article IV for details) and use the CPF-BBS within a method that does full Bayesian inference for the location and velocity of the object (that is, the latent states) as well as the potential values for each terrain type, given the observed locations. In effect, this could provide a simultaneous solution to the animal movement and resource selection problems, and might lead to an appealing alternative to step-selection analyses.

From a high level perspective, both the AI-CPF and the CPF-BBS improve the simulation performance in scenarios where the celebrated conditional particle filter with backward sampling (CPF-BS) works suboptimally. Furthermore, a common theme in our developments has been the study of heuristics that can be used to choose their tuning parameters. In our experiments in Articles II and IV, the developed heuristics worked well 'out of the box', but it is important to keep in mind that further tuning might be necessary in other applications, since the heuristics are backed by empirical experiments rather than rigorous theory. We think that the developed heuristics still provide important starting points for tuning, which in general may be a non-trivial problem in the context of particle MCMC methods.

Interestingly, the heuristics for both the adaptive AI-CPF and the CPF-BBS are related to the probability of state updates in a conditional particle filter. In the adaptive AI-CPF (see for example Algorithm 7 of Article II), the target acceptance rate controls the desired probability of a change of the initial state, and in the CPF-BBS the heuristic procedure attempts to maximise the probability of updates at the block lower boundaries (PLU). Therefore, we suspect that investigation of the probability of state updates in future works could yield interesting results from the point of view of tuning a conditional particle filter. In particular, in the context of the CPF-BBS, the developed estimator for PLU could possibly be used for developing a criterion for choosing the number of particles such that a near-optimal trade-off between computational load and statistical efficiency is reached (see also the related discussion in Article IV).

# REFERENCES

Akashi, H. and H. Kumamoto (1977). Random sampling approach to state estimation in switching environments. *Automatica* 13.4, pp. 429–434. DOI: https://doi.org/10.1016/0005-1098(77)90028-0.

Andrieu, C., A. Doucet, and R. Holenstein (2010). Particle Markov chain Monte Carlo methods. *J. R. Stat. Soc. Ser. B Stat. Methodol.* 72.3, pp. 269–342. DOI: https://doi.org/10.1111/j.1467-9868.2009.00736.x.

Andrieu, C. and J. Thoms (2008). A tutorial on adaptive MCMC. *Statistics and Computing* 18.4, pp. 343–373. DOI: https://doi.org/10.1007/s11222-008-9110-y.

Arasaratnam, I. and S. Haykin (2009). Cubature Kalman filters. *IEEE Transactions on Automatic Control* 54.6, pp. 1254–1269. DOI: 10.1109/TAC.2009.2019800.

Arasaratnam, I., S. Haykin, and R. J. Elliott (2007). Discrete-time nonlinear filtering algorithms using Gauss–Hermite quadrature. *Proceedings of the IEEE* 95.5, pp. 953–977. DOI: 10.1109/JPROC.2007.894705.

Baker, J. E. (1985). Adaptive selection methods for genetic algorithms. *Proceedings of the First International Conference on Genetic Algorithms and their Applications* 1, pp. 101–111.

Baum, L. E. and T. Petrie (1966). Statistical inference for probabilistic functions of finite state Markov chains. *The annals of mathematical statistics* 37.6, pp. 1554–1563. URL: https://www.jstor.org/stable/2238772.

Cappé, O., E. Moulines, and T. Rydén (2005). *Inference in Hidden Markov models*. Springer. DOI: https://doi.org/10.1007/0-387-28982-8.

Caron, F., M. Davy, E. Duflos, and P. Vanheeghe (2007). Particle filtering for multisensor data fusion with switching observation models: Application to land vehicle positioning. *IEEE Transactions on Signal Processing* 55.6, pp. 2703–2719. DOI: 10.1109/TSP.2007.893914.

Carpenter, J., P. Clifford, and P. Fearnhead (1999). Improved particle filter for nonlinear problems. *IEE Proceedings - Radar, Sonar and Navigation* 146.1, pp. 2–7. DOI: 10.1049/ip-rsn:19990255.

Chen, R. and J. S. Liu (2000). Mixture Kalman filters. *J. R. Stat. Soc. Ser. B Stat. Methodol.* 62.3, pp. 493–508. DOI: https://doi.org/10.1111/1467-9868.00246.

Chopin, N. and O. Papaspiliopoulos (2020). *An introduction to sequential Monte Carlo*. Springer. DOI: https://doi.org/10.1007/978-3-030-47845-2.

Chopin, N. and S. S. Singh (2015). On particle Gibbs sampling. *Bernoulli* 21.3, pp. 1855–1883. DOI: https://doi.org/10.3150/14-BEJ629.

Chopin, N., S. S. Singh, T. Soto, and M. Vihola (2022). On resampling schemes for particle filters with weakly informative observations. *arXiv preprint 2203.10037*. DOI: https://doi.org/10.48550/arXiv.2203.10037.

Cox, H. (1964). On the estimation of state variables and parameters for noisy dynamic systems. *IEEE Transactions on Automatic Control* 9.1, pp. 5–12. DOI: 10.1109/TAC.1964.1105635.

Craig, M. (2017). Towards quantitative systems pharmacology models of chemotherapy-induced neutropenia. *CPT: Pharmacometrics & Systems Pharmacology* 6.5, pp. 293–304. DOI: https://doi.org/10.1002/psp4.12191.

Crisan, D., P. Del Moral, and T. Lyons (1999). Discrete Filtering Using Branching and Interacting Particle Systems. *Markov Processes and Related Fields* 5.3, pp. 293–318.

Del Moral, P. (2004). *Feynman-Kac formulae: genealogical and interacting particle systems with applications*. Springer. DOI: https://doi.org/10.1007/978-1-4684-9393-1.

Del Moral, P. and L. Miclo (2000). Branching and interacting particle systems. Approximations of Feynman-Kac formulae with applications to non-linear filtering. *Séminaire de probabilités de Strasbourg* 34, pp. 1–145. URL: http://eudml.org/doc/114038.

Douc, R. and O. Cappé (2005). Comparison of resampling schemes for particle filtering. *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis*, pp. 64–69. DOI: 10.1109/ISPA.2005.195385.

Doucet, A., N. De Freitas, and N. J. Gordon (2001). *Sequential Monte Carlo methods in practice*. Springer. DOI: https://doi.org/10.1007/978-1-4757-3437-9.

Doucet, A. and A. M. Johansen (2011). A tutorial on particle filtering and smoothing: fifteen years later. *The Oxford Handbook of Nonlinear Filtering*. Oxford University Press. Chap. 8.2. URL: https://warwick.ac.uk/fac/sci/statistics/staff/academic-research/johansen/publications/DJ11.pdf.

Durbin, J. and S. J. Koopman (2012). *Time series analysis by state space methods*. 2nd edition. Oxford University Press. DOI: https://doi.org/10.1093/acprof:oso/9780199641178.001.0001.

Fearnhead, P. (1998). Sequential Monte Carlo methods in filter theory. PhD thesis. University of Oxford.

Fearnhead, P. (2011). MCMC for state-space models. *Handbook of Markov Chain Monte Carlo*. Chapman & Hall/CRC. Chap. 21. URL: https://eprints.lancs.ac.uk/id/eprint/8846/1/StateSpaceModels.pdf.

Fearnhead, P. and P. Clifford (2003). On-line inference for hidden Markov models via particle filters. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 65.4, pp. 887–899. DOI: https://doi.org/10.1111/1467-9868.00421.

Fearnhead, P. and L. Meligkotsidou (2016). Augmentation schemes for particle MCMC. *Statistics and Computing* 26.6, pp. 1293–1306. DOI: https://doi.org/10.1007/s11222-015-9603-4.

Friberg, L. E. et al. (2002). Model of chemotherapy-induced myelosuppression with parameter consistency across drugs. *Journal of clinical oncology* 20.24, pp. 4713–4721. DOI: https://doi.org/10.1200/JCO.2002.02.140.

Fröhlich, H. et al. (2018). From hype to reality: data science enabling personalized medicine. *BMC Medicine* 16.150. DOI: https://doi.org/10.1186/s12916-018-1122-7.

Gelfand, A. E. and A. F. Smith (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association* 85.410, pp. 398–409. DOI: 10.1080/01621459.1990.10476213.

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

Gerber, M., N. Chopin, and N. Whiteley (2019). Negative association, ordering and convergence of resampling methods. *The Annals of Statistics* 47.4, pp. 2236–2260. DOI: https://doi.org/10.1214/18-AOS1746.

Geweke, J. (1989). Bayesian inference in econometric models using Monte Carlo integration. *Econometrica* 57.6, pp. 1317–1339. DOI: https://doi.org/10.2307/1913710.

Godsill, S. (2019). Particle filtering: the first 25 years and beyond. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7760–7764. DOI: 10.1109/ICASSP.2019.8683411.

Goodman, I. R., R. P. S. Mahler, and H. T. Nguyen (1997). *Mathematics of Data Fusion*. Springer. DOI: https://doi.org/10.1007/978-94-015-8929-1.

Gordon, N. J., D. J. Salmond, and A. F. Smith (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. 140.2, pp. 107–113. DOI: https://doi.org/10.1049/ip-f-2.1993.0015.

Guarniero, P., A. M. Johansen, and A. Lee (2017). The iterated auxiliary particle filter. *Journal of the American Statistical Association* 112.520, pp. 1636–1647. DOI: 10.1080/01621459.2016.1222291.

Haario, H., E. Saksman, and J. Tamminen (2001). An adaptive Metropolis algorithm. *Bernoulli* 7.2, pp. 223–242. DOI: https://doi.org/10.2307/3318737.

Harvey, A. C. (1990). *Forecasting, structural time series models and the Kalman filter*. Cambridge University Press. DOI: https://doi.org/10.1017/CBO9781107049994.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* 57.1, pp. 97–109. DOI: https://doi.org/10.2307/2334940.

Helske, J., J. Nyblom, P. Ekholm, and K. Meissner (2013). Estimating aggregated nutrient fluxes in four Finnish rivers via Gaussian state space models. *Environmetrics* 24.4, pp. 237–247. DOI: https://doi.org/10.1002/env.2204.

Higuchi, T. (1997). Monte Carlo filter using the genetic algorithm operators. *Journal of Statistical Computation and Simulation* 59.1, pp. 1–23. DOI: 10.1080/00949659708811843.

Ito, K. and K. Xiong (2000). Gaussian filters for nonlinear filtering problems. *IEEE Transactions on Automatic Control* 45.5, pp. 910–927. DOI: 10.1109/9.855552.

Jayachandran, D. et al. (2014). Optimal chemotherapy for leukemia: a model-based strategy for individualized treatment. *PLOS ONE* 9, pp. 1–18. DOI: https://doi.org/10.1371/journal.pone.0109623.

Jazwinski, A. H. (1970). *Stochastic Processes and Filtering Theory*. Academic Press.

Johnson, D. S., J. M. London, M. Lea, and J. W. Durban (2008). Continuous-time correlated random walk model for animal telemetry data. *Ecology* 89.5, pp. 1208–1215. DOI: https://doi.org/10.1890/07-1032.1.

Julier, S. J. and J. K. Uhlmann (2004). Unscented filtering and nonlinear estimation. *Proceedings of the IEEE* 92.3, pp. 401–422. DOI: 10.1109/JPROC.2003.823141.

Julier, S. J., J. K. Uhlmann, and H. F. Durrant-Whyte (1995). A new approach for filtering nonlinear systems. *Proceedings of 1995 American Control Conference - ACC'95*. Vol. 3, pp. 1628–1632. DOI: 10.1109/ACC.1995.529783.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering* 82.1, pp. 35–45. DOI: https://doi.org/10.1115/1.3662552.

Karppinen, S. (2018). Valkosolupitoisuuksien bayesilainen mallintaminen lasten leukemian ylläpitohoidossa. Master's thesis. University of Jyväskylä. URL: http://urn.fi/URN:NBN:fi:jyu-201809214197.

Kitagawa, G. (1996). Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of computational and graphical statistics* 5.1, pp. 1–25. DOI: https://doi.org/10.1080/10618600.1996.10474692.

Lindsten, F. and T. B. Schön (2012). On the use of backward simulation in the particle Gibbs sampler. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3845–3848. DOI: 10.1109/ICASSP.2012.6288756.

Liu, J. S. (1996). Metropolized independent sampling with comparisons to rejection sampling and importance sampling. *Statistics and Computing* 6.2, pp. 113–119. DOI: https://doi.org/10.1007/BF00162521.

Liu, J. S. and R. Chen (1998). Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association* 93.443, pp. 1032–1044. DOI: https://doi.org/10.2307/2669847.

Maybeck, P. S. (1982). *Stochastic models, estimation, and control*. Academic Press.

Mirauta, B., P. Nicolas, and H. Richard (2014). Parseq: reconstruction of microbial transcription landscape from RNA-Seq read counts using state-space models. *Bioinformatics* 30.10, pp. 1409–1416. DOI: https://doi.org/10.1093/bioinformatics/btu042.

Møller, J., A. R. Syversveen, and R. P. Waagepetersen (1998). Log Gaussian Cox processes. *Scandinavian Journal of Statistics* 25.3, pp. 451–482. DOI: https://doi.org/10.1111/1467-9469.00115.

Natural Resources Institute Finland (2022). *Luonnonvaratieto: suurpetohavainnot*. Accessed on September 13th, 2022. URL: https://luonnonvaratieto.luke.fi/kartat?panel=suurpedot.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE* 77.2, pp. 257–286. DOI: 10.1109/5.18626.

Rasmussen, D. A., O. Ratmann, and K. Koelle (2011). Inference for nonlinear epidemiological models using genealogies and time series. *PLOS Computational Biology* 7, pp. 1–11. DOI: https://doi.org/10.1371/journal.pcbi.1002136.

Rauch, H. E., F. Tung, and C. T. Striebel (1965). Maximum likelihood estimates of linear dynamic systems. *AIAA Journal* 3.8, pp. 1445–1450. DOI: https://doi.org/10.2514/3.3166.

Robert, C. P. and G. Casella (2004). *Monte Carlo statistical methods*. 2nd edition. Springer. DOI: https://doi.org/10.1007/978-1-4757-4145-2.

Roberts, G. O. and A. F. Smith (1994). Simple conditions for the convergence of the Gibbs sampler and Metropolis-Hastings algorithms. *Stochastic Processes and their Applications* 49.2, pp. 207–216. DOI: https://doi.org/10.1016/0304-4149(94)90134-1.

Särkkä, S. (2013). *Bayesian filtering and smoothing*. Cambridge University Press. DOI: https://doi.org/10.1017/CBO9781139344203.

Särkkä, S., A. Vehtari, and J. Lampinen (2007). Rao-Blackwellized particle filter for multiple target tracking. *Information Fusion* 8.1, pp. 2–15. DOI: https://doi.org/10.1016/j.inffus.2005.09.009.

Schmiegelow, K., S. N. Nielsen, T. L. Frandsen, and J. Nersting (2014). Mercaptopurine/methotrexate maintenance therapy of childhood acute lymphoblastic leukemia: clinical facts and fiction. *Journal of pediatric hematology/oncology* 36.7, pp. 503–517. DOI: https://doi.org/10.1097/mph.0000000000000206.

Silvertown, J. (2009). A new dawn for citizen science. *Trends in Ecology & Evolution* 24.9, pp. 467–471. DOI: https://doi.org/10.1016/j.tree.2009.03.017.

Taylor, S. J. (2007). *Modelling financial time series*. 2nd edition. World Scientific. DOI: https://doi.org/10.1142/6578.

Thurfjell, H., S. Ciuti, and M. S. Boyce (2014). Applications of step-selection functions in ecology and conservation. *Movement Ecology* 2.4. DOI: https://doi.org/10.1186/2051-3933-2-4.

Vihola, M. (2007). Rao-Blackwellised particle filtering in random set multitarget tracking. *IEEE Transactions on Aerospace and Electronic Systems* 43.2, pp. 689–705. DOI: 10.1109/TAES.2007.4285362.

Vihola, M., J. Helske, and J. Franks (2017). Importance sampling type estimators based on approximate marginal Markov chain Monte Carlo. *arXiv preprint 1609.02541v3*. DOI: https://doi.org/10.48550/arXiv.1609.02541.

Vo, B., S. S. Singh, and A. Doucet (2003). Sequential Monte Carlo implementation of the PHD filter for multi-target tracking. *Proceedings of the Sixth International Conference of Information Fusion*. Vol. 2, pp. 792–799. DOI: 10.1109/ICIF.2003.177320.

Whiteley, N. (2010). Discussion on "Particle Markov chain Monte Carlo methods". *J. R. Stat. Soc. Ser. B Stat. Methodol.* 72.3, pp. 306–307. DOI: https://doi.org/10.1111/j.1467-9868.2009.00736.x.

Whiteley, N., C. Andrieu, and A. Doucet (2010). Efficient Bayesian inference for switching state-space models using discrete particle Markov chain Monte Carlo methods. *arXiv preprint 1011.2437*. DOI: https://doi.org/10.48550/arXiv.1011.2437.

Whitley, D. (1994). A genetic algorithm tutorial. *Statistics and Computing* 4.2, pp. 65–85. DOI: https://doi.org/10.1007/BF00175354.

Wood, S. N. (2010). Statistical inference for noisy nonlinear ecological dynamic systems. *Nature* 466, pp. 1102–1104. DOI: https://doi.org/10.1038/nature09319.

# ORIGINAL PAPERS

# I

## PREDICTION OF LEUKOCYTE COUNTS DURING PAEDIATRIC ACUTE LYMPHOBLASTIC LEUKAEMIA MAINTENANCE THERAPY

by

Karppinen, S., Lohi, O., and Vihola, M. 2019

Scientific Reports, 9.1, 1–11

DOI: 10.1038/s41598-019-54492-5

SCIENTIFIC REPORTS
natureresearch

OPEN

# Prediction of leukocyte counts during paediatric acute lymphoblastic leukaemia maintenance therapy

Santeri Karppinen[1]*, Olli Lohi[2] & Matti Vihola[1]

**Maintenance chemotherapy with oral 6-mercaptopurine and methotrexate remains a cornerstone of modern therapy for acute lymphoblastic leukaemia. The dosage and intensity of therapy are based on surrogate markers such as peripheral blood leukocyte and neutrophil counts. Dosage based leukocyte count predictions could provide support for dosage decisions clinicians face trying to find and maintain an appropriate dosage for the individual patient. We present two Bayesian nonlinear state space models for predicting patient leukocyte counts during the maintenance therapy. The models simplify some aspects of previously proposed models but allow for some extra flexibility. Our second model is an extension which accounts for extra variation in the leukocyte count due to a treatment adversity, infections, using C-reactive protein as a surrogate. The predictive performances of our models are compared against a model from the literature using time series cross-validation with patient data. In our experiments, our simplified models appear more robust and deliver competitive results with the model from the literature.**

Acute lymphoblastic leukaemia (ALL) is the most common cancer in childhood. In the Nordic countries, approximately 210 children are diagnosed yearly and patients are treated with chemotherapeutic drugs according to the ALL protocols of the Nordic Society of Paediatric Haematology and Oncology (NOPHO)[1]. The last phase of the treatment, maintenance therapy (MT), continues until 2 to 3 years from diagnosis. During MT, patients are treated orally with daily 6-mercaptopurine (6 MP) and weekly methotrexate (MTX).

Conventional MT starts with a standard 6 MP/MTX dose defined in the protocol. After initialisation of treatment, the dosage of the cytotoxic drugs is adjusted to reach a degree of myelosuppression, reflected in the NOPHO ALL-2008 protocol by targeting a leukocyte count of $1.5–3.0 \times 10^9$/L, while keeping the neutrophil count above $0.5 \times 10^9$/L[2]. Individual adjustments of 6 MP/MTX doses are necessary due to substantial interindividual variability in 6 MP/MTX bioavailability and cellular pharmacokinetics, and a narrow therapeutic index.

Finding the right 6 MP/MTX dosage may be challenging because there is a substantial delay before steady-state response in the leukocyte count is reached. Furthermore, many other factors, such as infections, can cause leukocyte fluctuations, and the dosage decisions during MT may be made by clinicians who have limited prior experience with 6 MP/MTX chemotherapy. Making the right decisions is crucial, as excessive dosage is associated with acute toxicity[3] and the risk of second cancers[4], whereas insufficient dosage results in poor treatment outcomes[5–7].

In this work, we develop statistical models for predicting leukocyte counts based on the doses administered during MT. One motivation for our work is a potential future application, where predictive modelling would be a part of a dosage decision support system, which automatically fits the model with data accumulated for the patient so far. The system then provides the clinician with an interactive visualisation of the patient's data, and leukocyte count predictions under alternative future dosing scenarios. This offers the clinician an analytical look on the data, and reassurance on her dosage decision. Ideally, the system could provide reliable predictions for most of the patients, but the clinician's expertise would remain essential for decision-making under exceptional scenarios such as patients with rare genotypes that affect 6 MP metabolism or patients with an infection.

[1]University of Jyväskylä, Department of Mathematics and Statistics, Jyväskylä, FI-40014, Finland. [2]Tampere Center for Child Health Research, Faculty of Medicine and Health Technology, Tampere University and Tampere University Hospital, Tampere, FI-33521, Finland. *email: santeri.j.karppinen@jyu.fi

The scope of this work is in the development of the predictive models, and in the evaluation of their predictive accuracies. We do not consider the implementation of the models into the clinical practice, or suggest alternative dosing strategies. We focus on the mathematical modelling related to the prediction of leukocyte counts in the context of ALL, but our developments may also be relevant outside this context, for instance in computational personalised medicine regarding other myelosuppressive medication. Currently, there are two published works where leukocyte counts during ALL MT are predicted[8,9]. Here, we present two statistical models following a structure similar to the existing models, but instead of using ordinary differential equation models, we use nonlinear Gaussian state space models[10] that stem from analogous stochastic differential equations. Our models introduce two simplifications, on the pharmacokinetic model for 6 MP[11] and on the leukopoiesis model[8,12]. Our second model, an extension of the first, incorporates C-reactive protein (CRP) measurements as a surrogate for infections and models the effect of an infection as extra variation in (or discrepancy from) the leukopoiesis model.

## Methods

The patient data were collected from historical medical records and consist of 23 patients under the age of 18 who had received MT under the NOPHO ALL-2000 or ALL-2008 treatment protocols at the Tampere University Hospital in Finland. This registry study (R16527) was accepted by the director of the Science Center in the Tampere University Hospital according to the local practice, and the data were anonymized before further analysis. The treatment length per patient varies from 227 to 524 days, with most of the patients receiving MT for more than 400 days. For each patient, the data contain the daily 6 MP dosage prescribed, as well as the leukocyte count and the CRP measurements made typically during weekly or biweekly visits to the hospital or the laboratory. The height and weight of each patient is also available at the start of MT. We used the Mosteller formula[13] to calculate the body surface area (BSA) for all patients during the treatment. The height and weight gain of the patients during the treatment was estimated by interpolating median growth curves obtained from the Centers for Disease Control and Prevention[14]. Because the patients' genders are not available in the data, average growth curves over boys and girls aged under 20 years were used. For each patient, the interpolation was begun from the height and weight values recorded in the data. The patientwise time series of the leukocyte counts, 6 MP, CRP and BSA are in the Supplementary Dataset 1.

To compare the models, we use the root mean squared error (RMSE) and the mean absolute error (MAE). In addition, we compute $\alpha$% coverage probabilities, that is, $\mathrm{CP}_\alpha = \frac{1}{n}\sum_{i=1}^{n}\mathbf{1}\{y_i \in \mathrm{I}_i^\alpha\}$ for $\alpha \in \{50, 90\}$, where $y_i$ denotes observation number $i$ and $\mathrm{I}_i^\alpha$ denotes the $\alpha$% probability interval for observation $y_i$. This metric is used to evaluate the ability of the models to quantify the uncertainty related to the point predictions. All of the metrics are computed out-of-sample and in-sample.

The out-of-sample metrics are of most interest, as they are computed using data not used in the model fitting and are directly tied to the predictive performance of the models. In a time series context, a natural way to compute them is to use time series cross-validation[15] (TSC). In a single round of TSC, we partition the data to a training set with data up to time $t$, and a prediction horizon immediately following the training set. The model is fit using the training set and the observations that fall into the prediction horizon are predicted using the fitted model. The training dataset is then augmented with observations in the prediction horizon and the process repeats until the data have been exhausted. After TSC, we compute the metrics using the obtained predictions and the corresponding observations. In the in-sample case, the metrics are computed based on model fits to full datasets by predicting all of the observations that were also used in the model fitting.

In the following subsections, we discuss the predictive models and estimation methods. We denote model state variables with capital letters, and parameters and data values in lowercase. A glossary and details regarding symbols used in the model definitions are also given in the Supplementary Tables 1–5.

### Jayachandran *et al*. model (JM).

The model from the literature, which we refer to as JM, is a joint 8-compartment model based on the work of Jayachandran *et al.*[8,11] The model consists of two submodels, the first of which is the 3-compartment pharmacokinetic model[11] for the metabolisation of 6 MP to red blood cell 6-thioguanine (TGNRBC):

$$\begin{aligned} \mathrm{d}X_{gut}/\mathrm{d}t &= -k_{ab}X_{gut} + d(t) \\ \mathrm{d}X_{plasma}/\mathrm{d}t &= k_{ab}X_{gut} - k_{el}X_{plasma} - \frac{k_{cm}X_{plasma}}{k + X_{plasma}} \\ \mathrm{d}X_{tgn}/\mathrm{d}t &= \frac{v_{cm}k_{cm}X_{plasma}}{k + X_{plasma}} - k_{me}X_{tgn}. \end{aligned} \tag{1}$$

The authors assume TGNRBC to be associated with the toxicity in the bone marrow due to 6 MP and hence model the variable as a surrogate for the myelosuppressive effect of 6 MP. The dataset in the article contained the administered 6 MP doses and the measured TGNRBC concentrations. The compartments $X_{gut}$ and $X_{plasma}$ represent 6 MP in gut and plasma, $X_{tgn}$ is the TGNRBC compartment, $d(t)$ is the dose input at time $t$ and the remaining symbols are parameters. The functional form of $d(t)$ was not given[8,11]. Hence, we assume $d(t)$ equals zero unless a dose is given exactly at time $t$.

The second submodel is the leukopoiesis model by Jayachandran *et al.*[8], which is a modification of the widely used 5-compartment model introduced by Friberg *et al.*[12]. We detail the model for log-transformed state variables:

$$\begin{aligned} \mathrm{d}S/\mathrm{d}t &= k_{pl}^{max}\frac{\rho^{\gamma}}{\rho^{\gamma}+\exp(L)^{\gamma}} - \frac{e_{max}X_{tgn}}{e_{C50}+X_{tgn}} - k_{tr} \\ \mathrm{d}C^{(1)}/\mathrm{d}t &= k_{tr}\exp(S - C^{(1)}) - k_{tr} \\ \mathrm{d}C^{(2)}/\mathrm{d}t &= k_{tr}\exp(C^{(1)} - C^{(2)}) - k_{tr} \\ \mathrm{d}C^{(3)}/\mathrm{d}t &= k_{tr}\exp(C^{(2)} - C^{(3)}) - k_{tr} \\ \mathrm{d}L/\mathrm{d}t &= k_{tr}\exp(C^{(3)} - L) - k_{L}. \end{aligned} \tag{2}$$

Here, the state variables form a maturation chain from stem cells ($S$) to leukocytes in circulation ($L$) through three maturation phases denoted by the compartments $C^{(i)}$, $i = 1, 2, 3$. TGNRBC ($X_{tgn}$) is assumed to diminish the rate of stem cell production. The remaining symbols are parameters.

As no information regarding the initial values of (1) or (2) is given[8,11], we assume that the patient's system starts in a steady state where no change in the cell concentrations is occuring initially. The steady state initialisation is obtained by setting the time derivatives at the start of the treatment (time zero) to zero. This is achieved by:

$$\begin{aligned} L(0) &= \log(\rho) + \log\left(\frac{k_{pl}^{max}}{k_{tr}} - 1\right)/\gamma \\ C^{(3)}(0) &= \log(k_L) - \log(k_{tr}) + L(0) \\ S(0) &= C^{(1)}(0) = C^{(2)}(0) = C^{(3)}(0), \end{aligned} \tag{3}$$

whenever $k_{tr} < k_{pl}^{max}$. Furthermore, we assume no 6 MP or TGNRBC exists in the patient's system at the beginning of MT, i.e. $X_{gut}(0) = X_{plasma}(0) = X_{tgn}(0) = 0$.

The log-leukocyte count measurements of a patient, $(l_k)_{k\geq 1}$, observed at times $t_k$, are assumed i.i.d. with Gaussian errors:

$$l_k \sim N(\hat{L}(t_k, \theta, d_{1:k}), \sigma_{leuk}^2), \tag{4}$$

where $\hat{L}(t_k, \theta, d_{1:k})$ is the solution of the state variable $L$ at time $t_k$, dependent on patient specific parameters $\theta$ and administered doses up to time index $k$, $d_{1:k}$.

**2-compartment model (TCM).** Our first model, denoted TCM, can be seen as a K-PD model[16]. TCM has a structure similar to that of JM, which it simplifies in two ways.

First, the pharmacokinetic model (1) is replaced with the pharmacokinetic model

$$\frac{\mathrm{d}M(t)}{\mathrm{d}t} = k_{me}\frac{e_{tgn}d(t)}{d(t)+h} - k_{me}M(t). \tag{5}$$

The model reflects changes in the cytotoxicity induced by 6 MP, $M$, in response to the 6 MP dose administered to the patient. The value of $M$ models the direct effect of chemotherapy, and is the counterpart of the term $\frac{e_{max}X_{tgn}}{e_{C50}+X_{tgn}}$ in (2). The value of the drug input function at time $t$, $d(t)$, equals the last 6 MP dose administered during the last 24 ($T_{dur}$) hours normalised by the patient's BSA, and zero if no dose was given. While this leads to noticeably different behaviour compared to (1) in the hourly time scale, the average daily behaviour of $M(t)$ remains very similar to that of $X_{tgn}$. A similar observation is made by Le *et al.*[9], who note that varying $T_{dur}$ does not have a strong influence on the concentration of TGNRBC in a prior pharmacokinetic model introduced by Jayachandran *et al.*[8], which is very similar to (1). Like the pharmacokinetic model of JM, (5) concentrates on the cytotoxic effect of 6 MP, and does not include MTX. We return to this matter in the discussion.

The parameters $e_{tgn}$ and $h$ play roles similar to $k_{cm}$ and $k$ in (1) as is evident from the similar functional form of (5) and the differential equation for $X_{tgn}$. Furthermore, the parameter $k_{me}$ is equivalent in (1) and (5). Jayachandran *et al.* reported a very high posterior correlation between the parameters $k_{me}$ and $k_{cm}$ in (1) [11]. We incorporate $k_{me}$ into the first term of (5) as this reduces the correlation between $k_{me}$ and $e_{tgn}$. The simplified form of (5) is motivated by simulation and parameter estimation, which reveal that the functional form of (5) is flexible enough to match solutions of $X_{tgn}$ when most of the parameters in (1) are fixed as in the analysis of Jayachandran *et al.*

The second simplification concerns the leukopoiesis model (2), which is replaced with a stochastic differential equation analogue of the equation for $S$:

$$dL_t = \left(k_{pl}^{max}\frac{\rho^{\gamma}}{\rho^{\gamma}+\exp(L)^{\gamma}} - M_t - k_L\right)dt + \sigma_L dB_t^{(L)}, \tag{6}$$

where $B_t^{(L)}$ is the Brownian motion and the parameter $\sigma_L$ is the leukopoiesis standard deviation. The parameter $k_{tr}$ in the equation for $S$ is substituted by the leukocyte elimination rate $k_L$ in (6), as (6) is a model for leukocyte counts. The leukopoiesis model (6) eliminates the cell maturation chain in (2) and models the effect of chemotherapy directly on the leukocytes in circulation. Unlike in (2), the drug effect is linear.

To obtain the state equation of TCM, (7), we solve the piecewise linear differential Eq. (5) at each interval $[t_{k-1}, t_k)$ with the initial condition $M(t_{k-1}) = M_{k-1}$, and apply the Euler-Maruyama discretisation[17] to (6), which results in

$$\begin{aligned} M_k &= \left(M_{k-1} - \frac{e_{tgn} d_{k-1}^{BSA}}{d_{k-1}^{BSA} + h}\right)\exp(-k_{me}\Delta t_k) + \frac{e_{tgn} d_{k-1}^{BSA}}{d_{k-1}^{BSA} + h} \\ L_k &= L_{k-1} + \Delta t_k\left(k_{pl}^{max}\frac{\rho^{\gamma}}{\rho^{\gamma} + \exp(L_{k-1})^{\gamma}} - M_{k-1} - k_L\right) + \sigma_L\sqrt{\Delta t_k}\zeta_k, \end{aligned} \tag{7}$$

where $d_{k-1}^{BSA} = d(t_{k-1})$, $\Delta t_k = t_k - t_{k-1}$, and $\zeta_k$ are standard normal random variables for all $k$. Initial distributions $M_1 \sim N(0, 0)$ and $L_1 \sim N(l_1, 0.5)$ are assumed for the state variables.

The log-leukocyte counts are related to the state variable $L$ with the observation equation

$$l_k = L_k + \varepsilon_k^{leuk}, \;\; \varepsilon_k^{leuk} \sim N(0, \sigma_{leuk}^2). \tag{8}$$

**2-compartment model with incorporated CRP (TCM-CRP).** Our second model, denoted TCM-CRP, is an extension of TCM, where the leukopoiesis standard deviation $\sigma_L$ is inflated in case of infection, for which the patient CRP measurements are taken as a surrogate.

TCM-CRP appends the state Eq. (7) with a third equation concerning an additional state variable, $V$, the level of infection. We model $V$ using an Ornstein-Uhlenbeck process:

$$dV_t = [\theta_{ou}V_t]dt + \sigma_{ou}dB_t^{(V)}, \;\; V_0 = v_0, \tag{9}$$

where $t$ denotes time, $B_t^{(V)}$ is the Brownian motion, and $\theta_{ou}$ and $\sigma_{ou}$ are parameters. Conditional on the previous value in the series, $V$ in (9) is Gaussian[18], which leads to the following state equation:

$$V_k = V_{k-1}e^{-\theta_{ou}\Delta t_k} + \sigma_{ou}(2\theta_{ou})^{-1/2}e^{-\theta_{ou}\Delta t_k}\sqrt{e^{2\theta_{ou}\Delta t_k} - 1}\,\eta_k, \tag{10}$$

where $k$ denotes the index of the time point and $\eta_k$ are standard normal random variables for all $k$. The only modification to (7) in TCM-CRP is that $\sigma_L$ is set to depend on $V_k$ and parameters $\sigma_L^0$ and $\beta_{crp}$ by

$$\sigma_L(V_k) = \sigma_L^0\exp(\beta_{crp}V_k), \tag{11}$$

making TCM-CRP a stochastic volatility type model. The state equation for TCM-CRP then consists of (7) modified with (11), and (10). The distribution of $V_1$ is set to the stationary distribution of (9),

$$N\left(0, \left(\frac{\sigma_{ou}}{\sqrt{2\theta_{ou}}}\right)^2\right),$$

and the distributions for $M_1$ and $L_1$ remain as in TCM.

Finally, TCM-CRP incorporates the $\log(x + 1)$-transformed CRP measurements, $v_k$, into the observation Eq. (8) by setting

$$v_k = V_k + \varepsilon_k^{crp}, \;\; \varepsilon_k^{crp} \sim N(0, \sigma_{crp}^2). \tag{12}$$

**Naive mean model (NM).** The fourth model we consider is a naive mean model (NM), which assumes that the leukocyte counts are i.i.d. and follow the normal distribution $N(\mu_{nm}, \sigma_{nm}^2)$. This model is an oversimplification, as it does not take into account the dosage given to the patient. Hence, we consider NM as a baseline for the models TCM, TCM-CRP and JM, and not as a realistic model candidate for predicting leukocyte counts.

**Estimation methods.** To estimate the parameters of the models TCM, TCM-CRP and JM, we use maximum a posteriori (MAP) estimation, where the posterior density

$$p(\theta|y) \propto p(y|\theta)p(\theta), \tag{13}$$

is maximised with respect to the logarithm of the free parameters, $\theta$, in the model. In (13), $y$ denotes the dataset for a single patient.

The value of $p(y|\theta)$ in (13) for JM and a given $\theta$ stems from (4). We use the Rosenbrock23 method[19] of the DifferentialEquations.jl package[20] in the Julia programming language[21] to solve the systems of differential equations. The predictions for JM are obtained by estimating the free parameters with data up to time index $k$, $y_{1:k}$, and solving the resulting system of differential equations on the interval $[t_{k+1}, t_{k+h_{pred}}]$, where $h_{pred}$ denotes the length of the prediction horizon.

To compute $p(y|\theta)$ and the predictions for TCM and TCM-CRP, we use the extended Kalman filter (EKF) which is an approximate method for computing the filtered state distributions for state space models with nonlinear dynamics in the state and observation equations[10,22].

| | 2 weeks | | | | 4 weeks | | | |
|---|---|---|---|---|---|---|---|---|
| | TCM | TCM-CRP | JM | NM | TCM | TCM-CRP | JM | NM |
| $\overline{\text{CP}}_{50}$ | 0.457 (0.13) | 0.442 (0.13) | 0.421 (0.13) | 0.514 (0.15) | 0.443 (0.15) | 0.428 (0.14) | 0.402 (0.15) | 0.507 (0.15) |
| $\overline{\text{CP}}_{90}$ | 0.795 (0.11) | 0.787 (0.11) | 0.761 (0.14) | 0.873 (0.09) | 0.767 (0.13) | 0.759 (0.13) | 0.722 (0.17) | 0.863 (0.09) |
| $\overline{\text{MAE}}$ | 0.860 (0.33) | 0.870 (0.35) | 0.964 (0.42) | 0.986 (0.43) | 0.896 (0.34) | 0.914 (0.38) | 1.016 (0.45) | 1.001 (0.44) |
| $\overline{\text{RMSE}}$ | 1.232 (0.51) | 1.244 (0.53) | 1.387 (0.66) | 1.308 (0.57) | 1.278 (0.55) | 1.309 (0.64) | 1.430 (0.68) | 1.326 (0.59) |

**Table 1.** The out-of-sample metrics for the models (TCM, TCM-CRP and JM) and the baseline model NM with both of the prediction horizons: means of coverage probability (CP), mean absolute error (MAE) and root mean squared error (RMSE). Standard deviations are in parentheses. Similar means are obtained if the metrics are computed modelwise without considering the patients separately.

In all maximisation problems, we assume the joint prior distribution $p(\theta)$ in (13) consists of vague independent $N(0,\ 10)$ distributions for each free parameter. The Nelder-Mead method[23,24] in the Optim.jl package[25] is used for the computation.

To estimate the parameters of the model NM, we compute the sample mean and variance of the leukocyte counts.

## Results

With JM, we attempted to reproduce the analysis of Jayachandran *et al.*[8,11] as accurately as possible and hence estimated parameters $k_{cm}$ in (1) and $k_{tr}$, $k_{pl}^{max}$, $k_L$, $\gamma$ and $e_{max}$ in (2). These parameters were found to have the greatest influence on the fitted values of JM's submodels in sensitivity analyses conducted in both articles[8,11]. In addition, the parameter $\sigma_{leuk}$ was estimated. The remaining parameters were fixed to the values reported by Jayachandran *et al.*

With TCM, the parameters $e_{tgn}$, $h$, $k_{pl}^{max}$, $k_L$ and $\sigma_L$ were estimated. The common parameters with JM, $k_{pl}^{max}$ and $k_L$, were estimated, but we fixed $\gamma$ to a value reported by Jayachandran *et al.*[8], because estimating it resulted in fits with oscillating behaviour not visible in the datasets. Furthermore, we estimated the leukopoiesis standard deviation $\sigma_L$, but fixed the measurement standard deviation $\sigma_{leuk}$ to a literature value of 0.057 for the accuracy of measuring neutrophil counts[26]. The remaining parameters, $k_{me}$ and $\rho$, were fixed to the same values as in JM. The discretisation $\Delta t_k$ was set to 0.25.

TCM-CRP was treated similarly to TCM, with the parameter $\sigma_L^0$ as the equivalent of $\sigma_L$. However, to maintain the same amount of free parameters as in TCM, we fixed the additional parameters $\sigma_{crp}$, $\sigma_{ou}$, $\theta_{ou}$ and $\beta_{crp}$. As the coefficient of variation for measuring CRP at 3.5 mg/l is close to 10%[27] and $\log(x + 1) \approx \log(x)$ when $x \geq 3.5$, we fixed $\sigma_{crp} = 0.1$ (note that if $X \sim N(\mu,\ \sigma^2)$, $\mu > 0$ and $\sigma^2$ sufficiently small, then $\log(X) \sim N(\log(\mu),\ (\sigma/\mu)^2)$ approximately). The remaining parameters, $\theta_V = (\sigma_{ou},\ \theta_{ou},\ \beta_{crp})$, were fixed to estimates obtained by maximising the objective

$$\prod_i p(y_i|\theta_i,\ \theta_V)p(\theta_i,\ \theta_V) \tag{14}$$

with respect to $(\theta_1,\ \theta_2,\ \dots,\ \theta_{23},\ \theta_V)$. In (14) each patient is indexed with *i*; $y_i$ and $\theta_i$ denote the dataset and the parameter vector of the free parameters in TCM for patient *i*. The joint approach for obtaining an estimate of $\theta_V$ was motivated by the fact that if $\theta_V$ were estimated individually for each patient, inadequate estimates of $\beta_{crp}$ were obtained for patients with mild or no infections during their treatment.

For all models, TSC was carried out such that the first training dataset for each patient was set to contain the first 8 weeks of the patient's data. In one case however, the first 8 weeks contained only one measured leukocyte count, and hence the first training set was extended to include two observations. For all models, TSC was run twice, with a prediction horizon of two and four weeks. The TSC schemes were completed successfully for the models TCM and TCM-CRP. With the two and four week schemes of JM, there were 31 and 19 TSC rounds where optimisation did not converge or prediction failed with a solver error. The patients who had at least one convergence or prediction failure during TSC with any horizon were 1, 4, 6, 7, 8, 11, 12, 20 and 22. Furthermore, when the models were fit to the full datasets, the optimisation of the parameters of JM did not converge for patient 6. In the summary tables that follow, the problematic TSC rounds and fits have been removed prior to computing the metrics. In the patientwise listings, these have not been removed.

The out-of-sample metrics with both of the prediction horizons are given in Table 1. The tabulated values are means over the metrics computed for each patient (underlying data available in the Supplementary Dataset 2). To compute the values, the logarithmic scale predictions of the models TCM, TCM-CRP and JM have been transformed to the linear scale. In the table, the means of RMSE and MAE suggest that the point predictive accuracies of TCM and TCM-CRP are slightly greater than the predictive accuracy of JM regardless of the prediction horizon. The baseline model NM performs surprisingly well and is roughly as accurate as JM.

The widths of the predictive probability intervals are closer to their target values for TCM and TCM-CRP than JM: in the case of the two week horizon, we observe discrepancies of 4–6% vs. 8% for $\text{CP}_{50}$ and discrepancies of 10–11% vs. 14% for $\text{CP}_{90}$. The respective discrepancies increase to about 6–7% vs. 10% for $\text{CP}_{50}$ and 13–14% vs. 18% for $\text{CP}_{90}$, when the horizon is extended to four weeks. The models TCM, TCM-CRP and JM underestimate the width of the intervals. This is likely a consequence of using MAP estimation, which does not account for

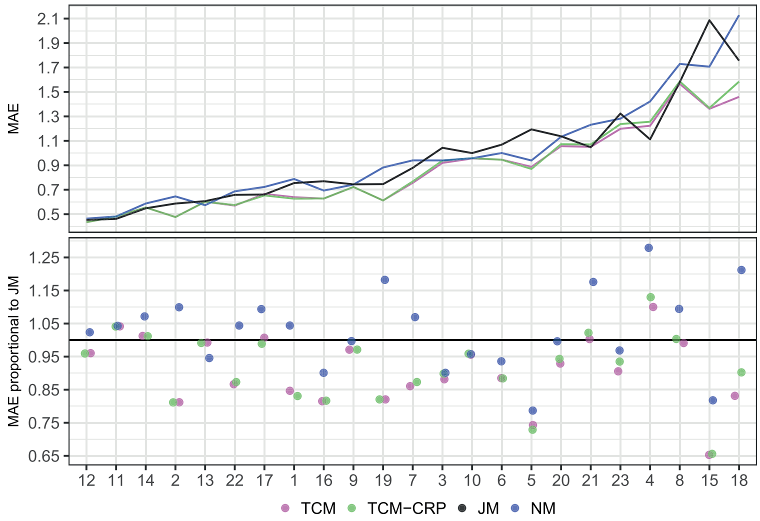


**Figure 1.** The patientwise out-of-sample mean absolute error for the models (TCM, TCM-CRP, JM) and the baseline model NM (top), and the relative error with respect to JM (bottom). The black line in the bottom plot depicts the line of equal predictive accuracy with JM. The out-of-sample values are from time series cross-validation with the two week prediction horizon. Each model is represented by a color. The patients have been ordered with increasing mean MAE over the models.

| | TCM | TCM-CRP | JM | NM |
|---|---|---|---|---|
| $\overline{\mathrm{CP}}_{50}$ | 0.571 (0.08) | 0.572 (0.08) | 0.545 (0.07) | 0.583 (0.11) |
| $\overline{\mathrm{CP}}_{90}$ | 0.904 (0.03) | 0.906 (0.03) | 0.904 (0.03) | 0.921 (0.03) |
| $\overline{\mathrm{MAE}}$ | 0.795 (0.31) | 0.800 (0.31) | 0.812 (0.32) | 0.924 (0.36) |
| $\overline{\mathrm{RMSE}}$ | 1.195 (0.56) | 1.203 (0.57) | 1.199 (0.56) | 1.304 (0.58) |

**Table 2.** The in-sample metrics per model: means of coverage probability (CP), mean absolute error (MAE) and root mean squared error (RMSE). Standard deviations are in parentheses.

the uncertainty in the model parameters. A more accurate representation of the uncertainty in the predictions could be obtained for example by using Markov chain Monte Carlo methods[28] that produce samples from the full posterior.

The predictive metrics are examined further in Fig. 1, which plots the patientwise MAE of the models in case of the two week prediction horizon. The plot shows that for most of the patients, TCM and TCM-CRP deliver predictions that are 5–20% more accurate than those of JM. For patients 7 and 15, however, the prediction accuracy is 25% and 35% better, respectively. Compared to TCM and TCM-CRP, JM performs slightly better for patients 11, 14, 21 and 4, who favour JM by 5–12%. In 13 cases out of 23, the predictive performance of JM appears better than that of NM. The figure with the four week prediction horizon, with similar findings, is given in the Supplementary Fig. 1.

Table 2 (underlying data available in the Supplementary Dataset 3) shows the in-sample metrics. The in-sample RMSE and MAE are computed between the ‘fitted mean‘ and the observed leukocyte counts. For TCM and TCM-CRP, we refer to the fitted mean as the exponentiated filtered mean of the state variable L obtained by first estimating the model parameters from the patient’s full dataset and then running EKF with all leukocyte counts set to missing, conditional on the estimated parameter values. For JM, the fitted mean is simply the exponentiated solution of L conditional on the parameter vector estimated from the full patient dataset, and for NM, the fitted mean is the estimate of $\mu_{nm}$. The in-sample means of RMSE, MAE and the coverage probabilities are very similar for JM, TCM and TCM-CRP, with JM reaching a slightly better value for $\mathrm{CP}_{50}$. As expected, the point predictions of TCM, TCM-CRP and JM are better than those of NM.

For many patients, the in-sample fits of JM exhibit oscillating behaviour, which by visual inspection is not present in the datasets. An example is shown in Fig. 2, which plots the fit of JM with TCM. In contrast, the fit of TCM is smoother and only captures the average behaviour of the leukocyte counts. See Supplementary Figs. 2–24 for graphical comparisons for all of the patients. The fits of the models TCM, TCM-CRP and JM to the full patient datasets are also given in the Supplementary Dataset 4.

Inspecting the predictions made during TSC in a similar manner, we found that the weaker out-of-sample metrics for JM are partly explained by the fact that for many patients, the model produces unstable predictions

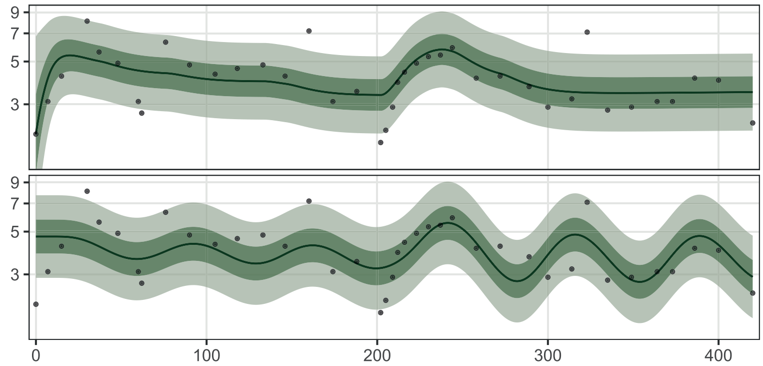


**Figure 2.** The models TCM (top) and JM (bottom) fit to the full dataset of patient 20 with time in days on the x-axis and leukocyte count on the y-axis. The fitted mean is the black line and probability intervals (50%, 90%) are plotted in green. The 6 MP dosage for the depicted patient was intensified incrementally to 50 mg during the first 200 days of treatment. After this, no dose was given for approximately 20 days. The dosage was then incrementally intensified back to 50 mg until treatment day 275 and kept constant until the end of the treatment. Further dose intensification was not possible due to low neutrophil counts.

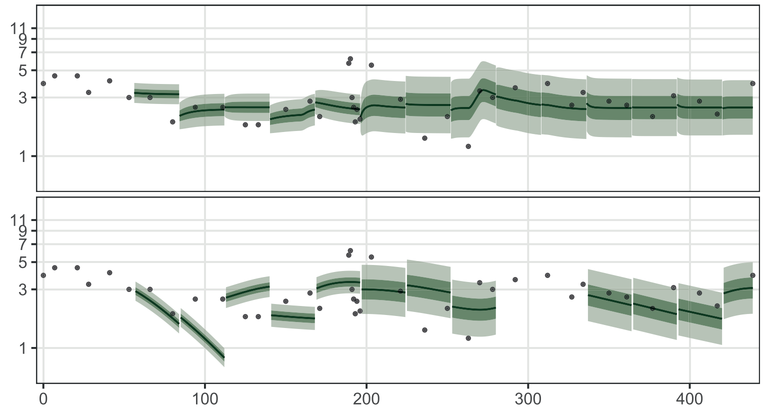


**Figure 3.** Predictions for patient 7 at each round of time series cross-validation with the four week prediction horizon for the models TCM (top) and JM (bottom). The plot for JM lacks predictions for treatment days 275–350, since the differential equation solver could not solve JM conditional on the parameter estimates found during optimisation. The 6 MP dosage for the depicted patient was intensified incrementally to 62.5 mg during the treatment. The dosage was not intensified further due to low neutrophil counts.

especially in the beginning of the treatment when only a few measurements are available for parameter estimation. Figure 3 shows an example of this by plotting the predictions of JM and TCM from cross-validation with the four week prediction horizon. Here, the predictions of JM appear unstable until treatment day 175 while the predictions of TCM appear more consistent. The unstability is unfortunate, since in the beginning of the treatment there is a lot of uncertainty in how the treatment will affect the patient. Hence, good predictions in this period of treatment are particularly important. The figure also shows some of the estimation problems we faced with JM, since the differential equation solver was unable to make a prediction for treatment days 275–350. The similar figures for all the patients are shown in the Supplementary Figs. 25–47.

Based on Table 1 and Fig. 1, there appears to be little difference between the out-of-sample metrics of TCM and TCM-CRP, with TCM reaching slightly better values than TCM-CRP. However, TCM-CRP has an interesting property that is not visible in the predictive metrics. This is showcased in Fig. 4 where the fit of TCM is compared to that of TCM-CRP in the case of a patient with infections during the treatment. Here, accounting for the infection induced variability in the leukocyte count results in narrower probability intervals for TCM-CRP, when infection is not present. Furthermore, when compared to TCM, the fitted mean of TCM-CRP is slightly shifted away from leukocyte counts measured during infection, indicating that the model is downweighting observations that occur during infection.

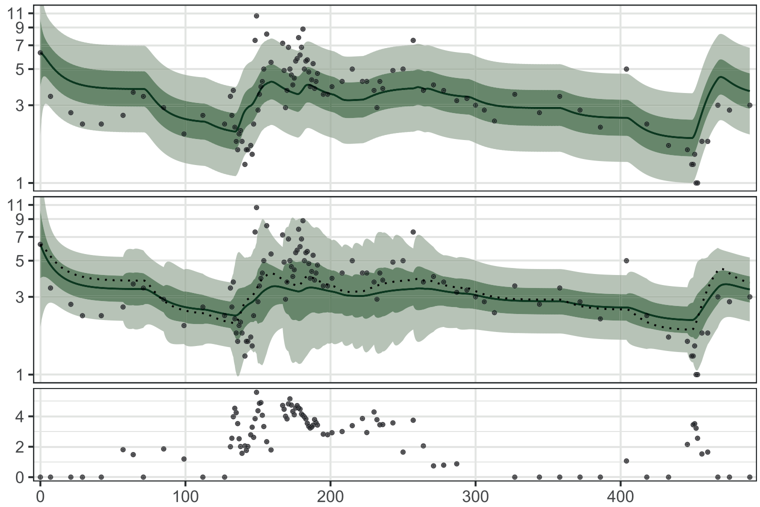


**Figure 4.** The fit of the models TCM (top) and TCM-CRP (middle) to the dataset of patient 4. The $\log(x + 1)$-transformed CRP measurements are shown at the bottom. The dotted line in the plot for TCM-CRP is the fitted mean of TCM.

## Discussion

In this work, we present two Bayesian nonlinear state space models, TCM and TCM-CRP, for predicting leukocyte counts during ALL MT. A predictive comparison between the models, and the model from the literature, JM, is then carried out. In prior works, predictive models for leukocyte counts during ALL MT have not been compared against each other according to their out-of-sample predictive performance. We argue that the development of predictive models should be guided by model comparison using out-of-sample metrics. Whenever possible, predictive models can also be validated by relating properties of the models to values available in the clinical literature. An approach like this was recently undertaken in a similar work[29] related to acute myeloid leukaemia, where leukocyte count recovery times were used to discriminate between model candidates with similar predictive power.

The best-performing model according to our results, TCM, simplifies the model from the literature, JM, in the pharmacokinetic and the leukopoiesis model, and delivers a prediction accuracy competitive with JM. The simplification in the pharmacokinetic model results in a focus on the daily behaviour of the cytotoxicity induced by 6 MP, which is in contrast with the pharmacokinetic model of JM that models the pharmacokinetics in the hourly granularity. We believe that such a fine time scale is unnecessary, when predictions are required on a daily or weekly basis, as in the present application. Similarly, the simplification of the leukopoiesis model changes the focus from the daily granularity to the weekly, which is justified since the leukocyte counts are typically measured at this rate. Despite these simplifications, we argue that TCM still captures the most important features of the phenomenon: the effect of 6 MP on the level of cytotoxicity, and the effect of the cytotoxicity on the leukocyte counts. The simplifications also reduce the number of parameters to be estimated, which allows for robust estimation of the model with sparse clinical datasets.

In our experiments, we found that JM was difficult to estimate reliably with our heterogeneous dataset and we had issues with optimisation and prediction. In absence of better initial values for the parameters, we used the estimates reported by Jayachandran *et al.*[8,11]. If these estimates are far from adequate for the patients in our dataset, they can play a role in the estimation problems. However, in general almost any variation of the model we attempted to fit during the process of preparing this work had estimation problems for at least some patients. Perhaps related to the estimation problems, the computation time to produce the cross-validation results with the two week prediction horizon, for example, was roughly hundredfold for JM compared to that of TCM (16.55 hours vs. 0.15 hours).

A comment by a reviewer led us to realise that the initial values of the state variables of the JM leukopoiesis model seem to play a significant role on how the model performs. When we initialised them by estimating a common value for every state variable, there were less TSC rounds with convergence or prediction issues. However, this initialisation resulted in a lower predictive accuracy than the model presented, and hence we chose the steady state initialisation. Lately, the impact of the initialisation has also been noted in a similar work[29], where models similar to the JM leukopoiesis model were investigated. It is possible that the alternative initialisation strategies found in the work might further improve the performance of JM.

Another noteworthy point regarding JM is that Jayachandran *et al.*[8,11] had additional TGNRBC measurements in their dataset, which our dataset does not contain. Fitting the model without these measurements might have implications for the identifiability of the model, and hence the observed predictive performance. Furthermore, the dataset of Jayachandran *et al.*[11] contains adults, and the pharmacokinetic profiles of adults and children differ. Allometric scaling[30] could improve the model, and allow for more immediate interpretation.

The baseline model NM was found to perform on par with JM and have point predictive metrics not far from those of TCM. This is surprising, as the model does not account for the dosage administered to the patient. We suspect that the success of this over-simplified model might be explained by our data, where for many patients, the treatment was successful and the leukocyte counts were centered around a common value, which makes their mean a relatively good prediction. With data having little variation in the leukocyte counts and/or dosage, it is difficult to improve the predictive performance.

The model TCM-CRP extends TCM by incorporating C-reactive protein (CRP) measurements into the model as a surrogate for infections. To our knowledge, TCM-CRP is the first model to attempt the inclusion of infection information to a leukopoiesis model. In our dataset, 65% of the patients have at least one infection during their treatment (by counting patients who have at least one CRP measurement greater than or equal to 10 mg/L), highlighting the prevalence of infections in MT. Furthermore, as there is an evident relationship between CRP and the leukocyte count (see Fig. 4), we argue that infections should be accounted for in ALL MT predictive modelling. In previous works, patients with infections during the treatment have been excluded from analysis[8,9].

Figure 4 shows a promising fit of TCM-CRP, but in general the parameter estimates computed for the model during TSC were similar to those of TCM, leading to similar predictions. This is likely because the state variable $V$ in TCM-CRP does not directly influence the mean of the state variable $L$, but controls its variability instead. We modelled infection this way, because the relationship between CRP and the leukocyte count appears hard to predict: at least in our data, elevated CRP seems to be associated with both increased and decreased leukocyte counts, with no apparent pattern. This is not surprising, since it is well known that CRP is nonspecific and can exhibit variable behaviour in different kinds of inflammatory states. Hence, our modelling strategy for infections did not aim to utilise CRP as a regressor (or predictor) for leukocyte counts, but rather to improve the robustness of the model against infections by downweighting the outlying leukocyte counts when CRP is elevated. Our hope was that this would result in a model that better predicts data measured when no infection is present. Based on the obtained results, this was not entirely successful, perhaps due to the proposed Ornstein-Uhlenbeck model and possibly the functional form of $\sigma_L$ being inadequately specified. The approximate nature of EKF can also play a role here, and better results for TCM-CRP could possibly be obtained by using more accurate estimation methods, such as particle Markov chain Monte Carlo[31].

The existing models predicting leukocyte counts during ALL MT use ordinary differential equation models[8,9]. In contrast to that approach, the nonlinear state space models we use allow for additional stochasticity in the state equation of the model, which we believe helps account for unmodelled variations in the data more accurately.

The leukopoiesis models of Jayachandran *et al.*[8] and Le *et al.*[9] extend the well-known 5-compartment structure introduced by Friberg *et al.*[12], for 6 MP (and MTX). It is worth mentioning that the chemotherapy drugs considered by Friberg *et al.* do not include 6 MP (or MTX), and are given in pulses, which is in contrast with the continuous low-dose administration of 6 MP in ALL MT. This may explain why our simpler one-compartment leukopoiesis model provided an improved predictive model in our experiments, and suggests that the commonly used 5-compartment model might not be optimal for all applications.

Little is known about the adequacy of pharmacokinetic models of 6 MP too, as datasets with recorded 6 MP doses and metabolites are rare and sparse, making model validation difficult. We are only aware of the works of Jayachandran *et al.*[8,11] and Hawwa *et al.*[32] where the dataset contained data on both administered 6 MP doses and TGNRBC. Furthermore, out-of-sample model comparison was only performed by Hawwa *et al.* Although TGNRBC was previously found to be associated with myelosuppression[33], later research has shown TGNRBC to be only weakly related to levels of DNA-thioguanine (TGNDNA), the main mediator of the cytotoxicity of 6 MP[34]. Hence, modelling TGNRBC as the end point of the pharmacokinetic model might not provide optimal predictions when the model is used in conjunction with a leukopoiesis model.

Recently, there has been increased interest in TGNDNA, as a study has found higher TGNDNA concentrations associated with improved relapse-free survival[2] and dosage could potentially be guided better by monitoring TGNDNA concentrations, as factors such as age, ethnicity and time of year confound the leukocyte counts[35,36]. However, to our knowledge, pharmacokinetic models for 6 MP with TGNDNA as the end point have not yet emerged and present an interesting prospect for future research regarding predictive modelling in the context of ALL MT. Moreover, if data with TGNDNA concentrations and leukocyte counts were available, the modelling framework of nonlinear state space models used in this work could readily incorporate the metabolite measurements into the model, and would in theory allow for the simultaneous prediction of the leukocyte count and the TGNDNA concentration, providing the clinician with extra information for decision-making.

In this work, we considered modelling the leukocyte counts based on 6 MP dosage only. We did not attempt to include MTX into our models, because the 6 MP and MTX dosages are strongly linked in our data, making reliable estimation of a joint model difficult. The concurrent work of Le *et al.*[9] incorporated patient MTX doses into their leukopoiesis model. While a comparison of the model to a model without MTX was not shown, incorporating MTX is likely an important step forward in ALL MT predictive modelling. However, we note that the rationale of MTX dosage in ALL MT is mainly that the drug increases the bioavailability of 6 MP[37,38], and only partially the cytotoxic effects of the drug itself. Hence, rather than incorporating the metabolites of MTX into the function $e_{drug}$ as was done by Le *et al.*, our intuition is that the MTX metabolites should rather be a covariate in the pharmacokinetic model for 6 MP, perhaps related to the value of the parameter $e_{tgn}$ in (7) or similar in another model.

Another interesting work in the literature is the work of Hawwa *et al.*[32], who investigated population pharmacokinetic models for 6 MP. The authors incorporated patient thiopurine methyltransferase (TPMT) genotype and BSA as covariates into their model and found that both variables reduced the interindividual variability in the model parameters significantly. We did not include the TPMT genotype to our pharmacokinetic model as the data were missing for 13 of the 23 patients, and of the remaining patients, 9 were of TPMT wildtype, only one was

TPMT heterozygous and there were no TPMT homozygotes. Hence, with the current data available, our model is representative of patients who are of TPMT wildtype, the genotype that covers 86–97%[39] of the population.

Combining our work with the work of Le *et al.*[9] and Hawwa *et al.*[32], it is possible to envision a model with the important covariates taken into account, improving the leukocyte count predictions. However, availability and sparseness of datasets remains a problem. Further improvements to the predictive performance could likely be obtained with hierarchical models linking the parameter vectors of the individual patients with hyperparameters. Such joint modelling has, to our knowledge, only been conducted in the context of ALL MT by Hawwa *et al*. with their pharmacokinetic model. In the course of preparing this work, we attempted to fit such models, but faced unresolvable computational problems likely due to the lack of 6 MP metabolite measurements in the dataset. Simpler joint models assuming the same values for a subset of parameters across patients were estimatable, but did not produce better predictive results than fitting the models to each dataset individually, likely due to the high interindividual variability in the parameters.

## Data availability

All data generated or analysed during this study are included in this published article (and its Supplementary Information files).

Received: 22 February 2019; Accepted: 12 November 2019;
Published online: 02 December 2019

## References

1. Toft, N. *et al*. Results of NOPHO ALL2008 treatment for patients aged 1–45 years with acute lymphoblastic leukemia. *Leukemia* **32**, 606–615 (2018).
2. Nielsen, S. N. *et al*. DNA-thioguanine nucleotide concentration and relapse-free survival during maintenance therapy of childhood acute lymphoblastic leukaemia (NOPHO ALL2008): a prospective substudy of a phase 3 trial. *The Lancet Oncology* **18**, 515–524 (2017).
3. Ebbesen, M. S. *et al*. Hepatotoxicity during maintenance therapy and prognosis in children with acute lymphoblastic leukemia. *Journal of pediatric hematology/oncology* **39**, 161–166 (2017).
4. Schmiegelow, K. *et al*. Methotrexate/6-mercaptopurine maintenance therapy influences the risk of a second malignant neoplasm after childhood acute lymphoblastic leukemia: results from the NOPHO ALL-92 study. *Blood* **113**, 6077–6084 (2009).
5. Schmiegelow, K. Prognostic significance of methotrexate and 6-mercaptopurine dosage during maintenance chemotherapy for childhood acute lymphoblastic leukemia. *Pediatric hematology and oncology* **8**, 301–312 (1991).
6. Relling, M. V., Hancock, M. L., Boyett, J. M., Pui, C.-H. & Evans, W. E. Prognostic importance of 6-mercaptopurine dose intensity in acute lymphoblastic leukemia. *Blood* **93**, 2817–2823 (1999).
7. Peeters, M., Koren, G., Jakubovicz, D. & Zipursky, A. Physician compliance and relapse rates of acute lymphoblastic leukemia in children. *Clinical Pharmacology & Therapeutics* **43**, 228–232 (1988).
8. Jayachandran, D., Rundell, A. E., Hannemann, R. E., Vik, T. A. & Ramkrishna, D. Optimal chemotherapy for leukemia: a model-based strategy for individualized treatment. *PloS one* **9**, e109623 (2014).
9. Le, T. T. *et al*. A mathematical model of white blood cell dynamics during maintenance therapy of childhood acute lymphoblastic leukemia. *Mathematical Medicine and Biology: A Journal of the IMA* (2018).
10. Durbin, J. & Koopman, S. J. *Time series analysis by state space methods*, vol. 2 (Oxford University Press, 2012).
11. Jayachandran, D. *et al*. Model-based individualized treatment of chemotherapeutics: Bayesian population modeling and dose optimization. *PloS one* **10**, e0133244 (2015).
12. Friberg, L. E., Henningsson, A., Maas, H., Nguyen, L. & Karlsson, M. O. Model of chemotherapy-induced myelosuppression with parameter consistency across drugs. *Journal of clinical oncology* **20**, 4713–4721 (2002).
13. Mosteller. Simplified calculation of body-surface area. *New England Journal of Medicine* **317**, 1098–1098, https://doi.org/10.1056/NEJM198710223171717, PMID: 3657876 (1987).
14. Centers for Disease Control and Prevention. Clinical growth charts. Accessed on 27.8.2018. https://www.cdc.gov/growthcharts/clinical_charts.htm (2017).
15. Hyndman, R. & Athanasopoulos, G. *Forecasting: principles and practice*, https://otexts.com/fpp2/ (OTexts, 2018).
16. Jacqmin, P. *et al*. Modelling response time profiles in the absence of drug concentrations: definition and performance evaluation of the K–PD model. *Journal of pharmacokinetics and pharmacodynamics* **34**, 57–85 (2007).
17. Kloeden, P. E. & Platen, E. *Numerical solution of stochastic differential equations*. Stochastic Modelling and Applied Probability, corrected edn. (Springer, 1995).
18. Vasicek, O. An equilibrium characterization of the term structure. *Journal of financial economics* **5**, 185 (1977).
19. Shampine, L. F. & Reichelt, M. W. The Matlab ODE suite. *SIAM journal on scientific computing* **18**, 1–22 (1997).
20. Rackauckas, C. & Nie, Q. DifferentialEquations.jl – a performant and feature-rich ecosystem for solving differential equations in Julia. *Journal of Open Research Software* **5** (2017).
21. Bezanson, J., Edelman, A., Karpinski, S. & Shah, V. B. Julia: A fresh approach to numerical computing. *SIAM review* **59**, 65–98 (2017).
22. Särkkä, S. *Bayesian filtering and smoothing*, vol. 3 (Cambridge University Press, 2013).
23. Nelder, J. A. & Mead, R. A simplex method for function minimization. *The computer journal* **7**, 308–313 (1965).
24. Gao, F. & Han, L. Implementing the Nelder-Mead simplex algorithm with adaptive parameters. *Computational Optimization and Applications* **51**, 259–277 (2012).
25. Mogensen, P. K. & Riseth, A. N. Optim: A mathematical optimization package for Julia. *Journal of Open Source Software* **3**, 615, https://doi.org/10.21105/joss.00615 (2018).
26. Amundsen, E. K., Urdal, P., Hagve, T.-A., Holthe, M. R. & Henriksson, C. E. Absolute neutrophil counts from automated hematology instruments are accurate and precise even at very low levels. *American journal of clinical pathology* **137**, 862–869 (2012).
27. Roberts, W. L., Sedrick, R., Moulton, L., Spencer, A. & Rifai, N. Evaluation of four automated high-sensitivity C-reactive protein methods: implications for clinical and epidemiological applications. *Clinical Chemistry* **46**, 461–468 (2000).
28. Gamerman, D. & Lopes, H. F. *Markov chain Monte Carlo: stochastic simulation for Bayesian inference* (Chapman and Hall/CRC, 2006).
29. Jost, F., Schalk, E., Rinke, K., Fischer, T. & Sager, S. Mathematical models for cytarabine-derived myelosuppression in acute myeloid leukaemia. *PloS one* **14**, e0204540 (2019).
30. Anderson, B. J. & Holford, N. H. Understanding dosing: children are small adults, neonates are immature children. *Archives of disease in childhood* **98**, 737–744 (2013).
31. Andrieu, C., Doucet, A. & Holenstein, R. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **72**, 269–342 (2010).

32. Hawwa, A. F. *et al*. Population pharmacokinetic and pharmacogenetic analysis of 6-mercaptopurine in paediatric patients with acute lymphoblastic leukaemia. *British journal of clinical pharmacology* **66**, 826–837 (2008).
33. Schmiegelow, K. *et al*. Risk of relapse in childhood acute lymphoblastic leukemia is related to RBC methotrexate and mercaptopurine metabolites during maintenance chemotherapy. Nordic Society for Pediatric Hematology and Oncology. *Journal of Clinical Oncology* **13**, 345–351 (1995).
34. Schmiegelow, K., Nielsen, S. N., Frandsen, T. L. & Nersting, J. Mercaptopurine/methotrexate maintenance therapy of childhood acute lymphoblastic leukemia: clinical facts and fiction. *Journal of pediatric hematology/oncology* **36**, 503 (2014).
35. Haddy, T. B., Rana, S. R. & Castro, O. Benign ethnic neutropenia: what is a normal absolute neutrophil count? *The Journal of laboratory and clinical medicine* **133**, 15–22 (1999).
36. Haus, E. & Smolensky, M. H. Biologic rhythms in the immune system. *Chronobiology international* **16**, 581–622 (1999).
37. Balis, F. M. *et al*. The effect of methotrexate on the bioavailability of oral 6-mercaptopurine. *Clinical Pharmacology & Therapeutics* **41**, 384–387, https://doi.org/10.1038/clpt.1987.45 (1987).
38. Innocenti, F. *et al*. Clinical and experimental pharmacokinetic interaction between 6-mercaptopurine and methotrexate. *Cancer chemotherapy and pharmacology* **37**, 409–414 (1996).
39. Nguyen, C. M., Mendes, M. A. & Ma, J. D. Thiopurine methyltransferase (TPMT) genotyping to predict myelosuppression risk. *PLoS currents* **3** (2011).

## Acknowledgements

We thank research nurse Satu Ranta for data preparation. S.K. and M.V. were supported by Academy of Finland grants 274740, 312605 and 315619. This research is related to the thematic research area DEMO (Decision Analytics utilising Causal Models and Multiobjective Optimisation) of the University of Jyväskylä.

## Author contributions

S.K. and M.V. did the statistical modelling and devised the experiments. S.K prepared the manuscript and implemented the experiments. O.L. provided the data and the medical expertise. All authors reviewed the manuscript.

## Competing interests

The authors declare no competing interests.

## Additional information

**Supplementary information** is available for this paper at https://doi.org/10.1038/s41598-019-54492-5.

**Correspondence** and requests for materials should be addressed to S.K.

**Reprints and permissions information** is available at www.nature.com/reprints.

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

# II

# CONDITIONAL PARTICLE FILTERS WITH DIFFUSE INITIAL DISTRIBUTIONS

by

Karppinen, S., and Vihola, M. 2021

Statistics and Computing, 31.3, 1–14

DOI: https://doi.org/10.1007/s11222-020-09975-1

Statistics and Computing (2021) 31:24
https://doi.org/10.1007/s11222-020-09975-1

# Conditional particle filters with diffuse initial distributions

Santeri Karppinen[1] · Matti Vihola[1]

Received: 30 June 2020 / Accepted: 4 December 2020

## Abstract

Conditional particle filters (CPFs) are powerful smoothing algorithms for general nonlinear/non-Gaussian hidden Markov models. However, CPFs can be inefficient or difficult to apply with diffuse initial distributions, which are common in statistical applications. We propose a simple but generally applicable auxiliary variable method, which can be used together with the CPF in order to perform efficient inference with diffuse initial distributions. The method only requires simulatable Markov transitions that are reversible with respect to the initial distribution, which can be improper. We focus in particular on random walk type transitions which are reversible with respect to a uniform initial distribution (on some domain), and autoregressive kernels for Gaussian initial distributions. We propose to use online adaptations within the methods. In the case of random walk transition, our adaptations use the estimated covariance and acceptance rate adaptation, and we detail their theoretical validity. We tested our methods with a linear Gaussian random walk model, a stochastic volatility model, and a stochastic epidemic compartment model with time-varying transmission rate. The experimental findings demonstrate that our method works reliably with little user specification and can be substantially better mixing than a direct particle Gibbs algorithm that treats initial states as parameters.

**Keywords** Adaptive Markov chain Monte Carlo · Bayesian inference · Compartment model · Conditional particle filter · Diffuse initialisation · Hidden Markov model · Smoothing · State space model

## 1 Introduction

In statistical applications of general state space hidden Markov models (HMMs), commonly known also as state space models, it is often desirable to initialise the latent state of the model with a diffuse (uninformative) initial distribution (cf. Durbin and Koopman 2012). We mean by 'diffuse' the general scenario, where the first marginal of the smoothing distribution is highly concentrated relative to the prior of the latent Markov chain, which may also be improper.

The conditional particle filter (CPF) (Andrieu et al. 2010), and in particular its backward sampling variants (Whiteley 2010; Lindsten et al. 2014), has been found to provide efficient smoothing even with long data records, both empirically (e.g. Fearnhead and Künsch 2018) and theoretically (Lee et al. 2020). However, a direct application of the CPF to a model with a diffuse initial distribution will lead to poor performance, because most of the initial particles will ultimately be redundant, as they become drawn from highly unlikely regions of the state space.

There are a number of existing methods which can be used to mitigate this inefficiency. For simpler settings, it is often relatively straightforward to design proposal distributions that lead to an equivalent model, which no longer has a diffuse initial distribution. Indeed, if the first filtering distribution is already informative, its analytical approximation may be used directly as the first proposal distribution. The iteratively refined look-ahead approach suggested by Guarniero et al. (2017) extends to more complicated settings, but can require careful tuning for each class of problems.

We aim here for a general approach, which does not rely on any problem-specific constructions. Such a general approach which allows for diffuse initial conditions with particle Markov chain Monte Carlo (MCMC) is to include the initial latent state of the HMM as a 'parameter'. This was suggested by Murray et al. (2013) with the particle marginal Metropolis–Hastings (PMMH). The same

**Supplementary Information** The online version contains supplementary material available at https://doi.org/10.1007/s11222-020-09975-1.

✉ Santeri Karppinen
santeri.j.karppinen@jyu.fi

[1] Department of Mathematics and Statistics, University of Jyväskylä, 40014 Jyväskylä, Finland

Published online: 03 March 2021

approach is directly applicable also with the CPF (using particle Gibbs); see Fearnhead and Meligkotsidou (2016), who discuss general approaches based on augmentation schemes.

Our approach may be seen as an instance of the general 'pseudo-observation' framework of Fearnhead and Meligkotsidou (2016), but we are unaware of earlier works about the specific class of methods we focus on here. Indeed, instead of building the auxiliary variable from the conjugacy perspective as Fearnhead and Meligkotsidou (2016), our approach is based on Markov transitions that are reversible with respect to the initial measure of the HMM. This approach may be simpler to understand and implement in practice, and is very generally applicable. We focus here on two concrete cases: the 'diffuse Gaussian' case, where the initial distribution is Gaussian with a relatively uninformative covariance matrix, and the 'fully diffuse' case, where the initial distribution is uniform. We suggest online adaptation mechanisms for the parameters, which make the methods easy to apply in practice.

We start in Sect. 2 by describing the family of models we are concerned with, and the general auxiliary variable initialisation CPF that underlies all of our developments. We present the practical methods in Sect. 3. Section 4 reports experiments of the methods with three academic models and concludes with a realistic inference task related to modelling the COVID-19 epidemic in Finland. We conclude with a discussion in Sect. 5.

## 2 The model and auxiliary variables

Our main interest is with HMMs having a joint smoothing distribution $\pi$ of the following form:

$$\pi(x_{1:T}) \propto p(x_1)p(y_1 \mid x_1) \prod_{k=2}^{T} p(x_k \mid x_{k-1})p(y_k \mid x_k), \quad (1)$$

where $\ell{:}u$ denotes the sequence of integers from $\ell$ to $u$ (inclusive), $x_{1:T}$ denotes the latent state variables, and $y_{1:T}$ the observations. Additionally, $\pi$ may depend on (hyper)parameters $\theta$, the dependence on which we omit for now, but return to later, in Sect. 3.4.

For the convenience of notation, and to allow for some generalisations, we focus on the Feynman–Kac form of the HMM smoothing problem (cf. Del Moral 2004), where the distribution of interest $\pi$ is represented in terms of a $\sigma$-finite measure $M_1(\mathrm{d}x_1)$ on the state space $\mathsf{X}$, Markov transitions $M_2, \ldots, M_T$ on $\mathsf{X}$ and potential functions $G_k : \mathsf{X}^k \to [0, \infty)$ so that

$$\pi(\mathrm{d}x_{1:T}) \propto M_1(\mathrm{d}x_1)G_1(x_1) \prod_{k=2}^{T} M_k(x_{k-1}, \mathrm{d}x_k)G_k(x_{1:k}). \quad (2)$$

The classical choice, the so-called 'bootstrap filter' (Gordon et al. 1993), corresponds to $M_1(\mathrm{d}x_1) = p(x_1)\mathrm{d}x_1$ and $M_k(x_{k-1}, \mathrm{d}x_k) = p(x_k \mid x_{k-1})\mathrm{d}x_k$, where '$\mathrm{d}x$' stands for the Lebesgue measure on $\mathsf{X} = \mathbb{R}^d$, and $G_k(x_{1:k}) = p(y_k \mid x_k)$, but other choices with other 'proposal distributions' $M_k$ are also possible. Our main focus is when $M_1$ is diffuse with respect to the first marginal of $\pi$. We stress that our method accomodates also improper $M_1$, such as the uniform distribution on $\mathbb{R}^d$, as long as (2) defines a probability.

The key ingredient of our method is an auxiliary Markov transition, $Q$, which we can simulate from, and which satisfies the following:

**Assumption 1** ($M_1$-reversibility) The Markov transition probability $Q$ is reversible with respect to the $\sigma$-finite measure $M_1$, or $M_1$-reversible, if

$$\int M_1(\mathrm{d}x_0)Q(x_0, \mathrm{d}x_1)\mathbf{1}(x_0 \in A, x_1 \in B) \\ = \int M_1(\mathrm{d}x_1)Q(x_1, \mathrm{d}x_0)\mathbf{1}(x_0 \in A, x_1 \in B), \quad (3)$$

for all measurable $A, B \subset \mathsf{X}$.

We discuss practical ways to choose $Q$ in Sect. 3. Assuming an $M_1$-reversible $Q$, we define an augmented target distribution, involving a new 'pseudo-state' $x_0$ which is connected to $x_1$ by $Q$:

$$\tilde{\pi}(\mathrm{d}x_{0:T}) = \pi(\mathrm{d}x_{1:T})Q(x_1, \mathrm{d}x_0) \\ \propto M_1(\mathrm{d}x_0)Q(x_0, \mathrm{d}x_1)G_1(x_1) \prod_{k=2}^{T} M_k(x_{k-1}, \mathrm{d}x_k)G_k(x_{1:k}).$$

It is clear by construction that $\tilde{\pi}$ admits $\pi$ as its marginal, and therefore, if we can sample $x_{0:T}$ from $\tilde{\pi}$, then $x_{1:T} \sim \pi$.

Our method may be viewed as a particle Gibbs (Andrieu et al. 2010) which targets $\tilde{\pi}$, regarding $x_0$ as the 'parameter', and $x_{1:T}$ the 'latent state', which are updated using the CPF. Algorithm 1 summarises the method, which we call the 'auxiliary initialisation' CPF (AI-CPF). Algorithm 1 determines a $\pi$-invariant Markov transition $\dot{x}_{1:T} \to \tilde{X}_{1:T}^{(B_{1:T})}$; the latter output of the algorithm will be relevant later, when we discuss adaptation.

```
**Algorithm 1** AI-CPF($\dot{x}_{1:T}; Q, M_{2:T}, G_{1:T}, N$)
Simulate $X_0 \sim Q(\dot{x}_1, \cdot)$.
Simulate $\tilde{X}_1^{(2:N)} \sim Q(X_0, \cdot)$ and set $\tilde{X}_1^{(1)} = \dot{x}_1$.
$(\tilde{X}_{1:T}^{(1:N)}, W_{1:T}^{(1:N)}, A_{1:T-1}^{(1:N)}) \leftarrow$ F-CPF($\dot{x}_{2:T}, \tilde{X}_1^{(1:N)}; M_{2:T}, G_{1:T}, N$).
$(B_{1:T}, V^{(1:N)}) \leftarrow$ PickPath- x($\tilde{X}_{1:T}^{(1:N)}, W_{1:T}^{(1:N)}, A_{1:T-1}^{(1:N)}, M_{2:T}, G_{2:T}$).
Set $\tilde{x}_{1:T} = \big(\tilde{X}_1^{(B_1)}, \tilde{X}_2^{(B_2)}, \ldots, \tilde{X}_T^{(B_T)}\big)$.
**output** $\big(\tilde{x}_{1:T}, (B_1, V^{(1:N)}, \tilde{X}_1^{(1:N)})\big)$.
```

Line 1 of Algorithm 1 implements a Gibbs step sampling $X_0$ conditional on $X_{1:T} = \dot{x}_{1:T}$, and lines 2–4 implement together a CPF targeting the conditional of $X_{1:T}$ given $X_0$. Line 3 runs what we call a 'forward' CPF, which is just a standard CPF conditional on the first state particles $X_1^{(1:N)}$, detailed in Algorithm 2. Line 4 refers to a call of PickPath- AT (Algorithm 3) for ancestor tracing as in the original work of Andrieu et al. (2010), or PickPath- BS (Algorithm 4) for backward sampling (Whiteley 2010). Categ$(w^{(1:N)})$ stands for the categorical distribution, that is, $A \sim \text{Categ}(w^{(1:N)})$ if $\Pr(A = i) = w^{(i)}$.

**Algorithm 2** F-CPF($\dot{x}_{2:T}, X_1^{(1:N)}; M_{2:T}, G_{1:T}, N$)

```
Set $\mathbf{X}_1^{(1:N)} \leftarrow X_1^{(1:N)}$.
for $k = 1, \ldots, T-1$ do
    $\tilde{W}_k^{(i)} \leftarrow G_k(\mathbf{X}_k^{(i)})$ and $W_k^{(i)} \leftarrow \tilde{W}_k^{(i)} / \sum_{j=1}^N \tilde{W}_k^{(j)}$ for $i \in \{1{:}N\}$.
    $A_k^{(2:N)} \sim \text{Categ}\big(W_k^{(1:N)}\big)$ and set $A_k^{(1)} \leftarrow 1$.
    Draw $X_{k+1}^{(i)} \sim M_{k+1}(\,\cdot \mid X_k^{(A_k^{(i)})})$ for $i \in \{2{:}N\}$.
    Set $X_{k+1}^{(1)} = \dot{x}_{k+1}$.
    Set $\mathbf{X}_{k+1}^{(i)} = (\mathbf{X}_k^{(A_k^{(i)})}, X_{k+1}^{(i)})$ for $i \in \{1{:}N\}$.
end for
$\tilde{W}_T^{(1:N)} \leftarrow G_T(\mathbf{X}_T^{(1:N)})$ and $W_T^{(i)} \leftarrow \tilde{W}_T^{(i)} / \sum_{j=1}^N \tilde{W}_T^{(j)}$ for $i = \{1{:}N\}$.
output $(X_{1:T}^{(1:N)}, W_{1:T}^{(1:N)}, A_{1:T-1}^{(1:N)})$.
```

The ancestor tracing variant can be used when the transition densities are unavailable. However, our main interest here is with backward sampling, summarised in Algorithm 4 in the common case where the potentials only depend on two consecutive states, that is, $G_k(x_{1:k}) = G_k(x_{k-1:k})$, and the transitions admit densities $M_k(x_{k-1}, \mathrm{d}x_k) = M_k(x_{k-1}, x_k)\,\mathrm{d}x_k$ with respect to some dominating $\sigma$-finite measure '$\mathrm{d}x_k$'.

**Algorithm 3** PickPath- AT($\tilde{X}_{1:T}^{(1:N)}, W_{1:T}^{(1:N)}, A_{1:T-1}^{(1:N)}, M_{2:T}, G_{2:T}$)

```
Draw $B_K \sim \text{Categ}(W_T^{(1:N)})$.
output $(B_{1:T}, W_1^{(1:N)})$ where $B_k = A_k^{(B_{k+1})}$ for $k = T-1, \ldots, 1$.
```

**Algorithm 4** PickPath- BS($\tilde{X}_{1:T}^{(1:N)}, W_{1:T}^{(1:N)}, A_{1:T-1}^{(1:N)}, M_{2:T}, G_{2:T}$)

```
Draw $B_K \sim \text{Categ}(W_T^{(1:N)})$.
for $k = T-1, \ldots, 1$ do
    $\tilde{V}_k^{(i)} \leftarrow W_k^{(i)} M_{k+1}(\tilde{X}_k^{(i)}, \tilde{X}_{k+1}^{(B_{k+1})}) G_{k+1}(\tilde{X}_k^{(i)}, \tilde{X}_{k+1}^{(B_{k+1})})$ for $i \in \{1{:}N\}$.
    Simulate $B_k \sim \text{Categ}(V_k^{(1:N)})$, where $V_k^{(i)} = \tilde{V}_k^{(i)} / \sum_{j=1}^N \tilde{V}_k^{(j)}$.
end for
output $(B_{1:T}, V_1^{(1:N)})$.
```

We conclude with a brief discussion on the general method of Algorithm 1.

(i) We recognise that Algorithm 1 is not new per se, in that it may be viewed just as a particle Gibbs applied for a specific auxiliary variable model. However, we are unaware of Algorithm 1 being presented with the present focus: with an $M_1$-reversible $Q$, and allowing for an improper $M_1$.

(ii) Algorithm 1 may be viewed as a generalisation of the standard CPF. Indeed, taking $Q(x_0, \mathrm{d}x_1) = M_1(\mathrm{d}x_1)$ in Algorithm 1 leads to the standard CPF. Note that Line 1 is redundant in this case, but is necessary in the general case.

(iii) In the case $T = 1$, Line 3 of Algorithm 1 is redundant, and the algorithm resembles certain multiple-try Metropolis methods (cf. Martino 2018) and has been suggested earlier by Mendes et al. (2015).

(iv) Algorithm 2 is formulated using multinomial resampling, for simplicity. We note that any other unbiased resampling may be used, as long as the conditional resampling is designed appropriately; see Chopin and Singh (2015).

The 'CPF generalisation' perspective of Algorithm 1 may lead to other useful developments; for instance, one could imagine the approach to be useful with the CPF applied for static (non-HMM) targets, as in sequential Monte Carlo samplers (Del Moral et al. 2006). The aim of the present paper is, however, to use Algorithm 1 with diffuse initial distributions.

## 3 Methods for diffuse initialisation of conditional particle filters

To illustrate the typical problem that arises with a diffuse initial distribution $M_1$, we examine a simple noisy AR(1) model:

$$\begin{aligned} x_{k+1} &= \rho x_k + \eta_k,\, \eta_k \sim N(0, \sigma_x^2) \\ y_k &= x_k + \epsilon_k,\, \epsilon_k \sim N(0, \sigma_y^2), \end{aligned} \tag{4}$$

for $k \geq 1$, $x_1 \sim N(0, \sigma_1^2)$, $M_1(\mathrm{d}x_1) = p(x_1)\mathrm{d}x_1$, $M_k(x_{k-1}, \mathrm{d}x_k) = p(x_k \mid x_{k-1})\mathrm{d}x_k$ and $G_k(x_{1:k}) = p(y_k \mid x_k)$.

We simulated a dataset of length $T = 50$ from this model with $x_1 = 0$, $\rho = 0.8$ and $\sigma_x = \sigma_y = 0.5$. We then ran 6000 iterations of the CPF with backward sampling (CPF-BS) with $\sigma_1 \in \{10, 100, 1000\}$; that is, Algorithm 1 with $Q(x_0, \cdot\,) = M_1(\,\cdot\,)$ together with Algorithm 4, and discarded the first 1000 iterations as burn-in. For each value of $\sigma_1$, we monitored the efficiency of sampling $x_1$. Figure 1 displays the resulting traceplots. The estimated integrated autocorrelation times (IACT) were approximately 3.75, 28.92 and 136.64, leading to effective sample sizes ($\mathrm{n_{eff}}$) of 1600, 207 and 44,

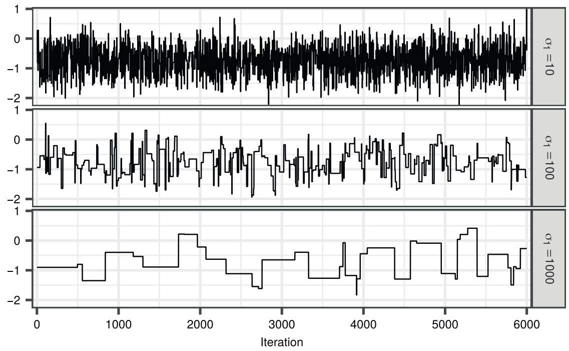


**Fig. 1** Traceplot of the initial state of the noisy AR(1) model, using the CPF with 16 particles and backward sampling with $\sigma_1 = 10$ (top), 100 (middle) and 1000 (bottom)

respectively. This demonstrates how the performance of the CPF-BS deteriorates as the initial distribution of the latent state becomes more diffuse.

### 3.1 Diffuse Gaussian initialisation

In the case that $M_1$ in (2) is Gaussian with mean $\mu$ and covariance $\Sigma$, we can construct a Markov transition function that satisfies (3) using an autoregressive proposal similar to 'preconditioning' in the Crank-Nicolson algorithm (cf. Cotter et al. 2013). This proposal comes with a parameter $\beta \in (0, 1]$, so we denote this kernel by $Q_\beta^{\mathrm{AR}}$. A variate $Z \sim Q_\beta^{\mathrm{AR}}(x, \cdot)$ can be drawn simply by setting

$$Z = \sqrt{1 - \beta^2}(x - \mu) + \beta W + \mu, \tag{5}$$

where $W \sim N(0, \Sigma)$. We refer to Algorithm 1 with $Q = Q_\beta^{\mathrm{AR}}$ as the diffuse Gaussian initialisation CPF (DGI-CPF). In the special case $\beta = 1$, we have $Q_1^{\mathrm{AR}} = M_1$, and so the DGI-CPF is equivalent with the standard CPF.

### 3.2 Fully diffuse initialisation

Suppose that $M_1(\mathrm{d}x) = M_1(x)\mathrm{d}x$ where $M_1(x) \equiv 1$ is a uniform density on $\mathsf{X} = \mathbb{R}^d$. Then, any symmetric transition $Q$ satisfies $M_1$-reversibility. In this case, we suggest to use $Q_C^{\mathrm{RW}}(x, \mathrm{d}y) = q_C^{\mathrm{RW}}(x, y)\mathrm{d}y$ with a multivariate normal density $q_C^{\mathrm{RW}}(x, y) = N(y; x, C)$, with covariance $C \in \mathbb{R}^{d \times d}$. In case of constraints, that is, a non-trivial domain $D \subset \mathbb{R}^d$, we have $M_1 = 1(x \in D)$. Then, we suggest to use a Metropolis–Hastings type transition probability:

$$Q_C^{\mathrm{RW}}(x, \mathrm{d}y) = q_C^{\mathrm{RW}}(x, y) \min\left\{1, \frac{M_1(y)}{M_1(x)}\right\}\mathrm{d}y \\ + \delta_x(\mathrm{d}y) r(x),$$

where $r(x) \in [0, 1]$ is the rejection probability. This method works, of course, with arbitrary $M_1$, but our focus is with a diffuse case, where the domain $D$ is regular and large enough, so that rejections are rare. We stress that also in this case, $M_1(x) = 1(x \in D)$ may be improper. We refer to Algorithm 1 with $Q_C^{\mathrm{RW}}$ as the 'fully diffuse initialisation' CPF (FDI-CPF).

We note that whenever $M_1$ can be evaluated pointwise, the FDI-CPF can always be applied, by considering the modified Feynman–Kac model $\tilde{M}_1 \equiv 1$ and $\tilde{G}_1(x) = M_1(x)G_1(x)$. However, when $M_1$ is Gaussian, the DGI-CPF can often lead to a more efficient method. As with standard random walk Metropolis algorithms, choosing the covariance $C \in \mathbb{R}^{d \times d}$ is important for the efficiency of the FDI-CPF.

### 3.3 Adaptive proposals

Finding a good autoregressive parameter of $Q_\beta^{\mathrm{AR}}$ or the covariance parameter of $Q_C^{\mathrm{RW}}$ may be time-consuming in practice. Inspired by the recent advances in adaptive MCMC (cf. Andrieu and Thoms 2008; Vihola 2020), it is natural to apply adaptation also with the (iterated) AI-CPF. Algorithm 5 summarises a generic adaptive AI-CPF (AAI-CPF) using a parameterised family $\{Q_\zeta\}_{\zeta \in Z}$ of $M_1$-reversible proposals, with parameter $\zeta$.

**Algorithm 5** AAI-CPF($\dot{x}_{1:T}^{(0)}$; $Q_{\zeta^{(0)}}$, $M_{2:T}$, $G_{1:T}$, $N$)

```
for $j = 1, \ldots, n$ do
    $(\dot{x}_{1:T}^{(j)}, \xi^{(j)}) \leftarrow$ AI-CPF($\dot{x}_{1:T}^{(j-1)}$; $Q_{\zeta^{(j-1)}}$, $M_{2:T}$, $G_{1:T}$, $N$).
    $\zeta^{(j)} \leftarrow$ ADAPT($\zeta^{(j-1)}$, $\xi^{(j)}$, $j$).
end for
output $(\dot{x}_{1:T}^{(1)}, \ldots, \dot{x}_{1:T}^{(n)})$.
```

The function ADAPT implements the adaptation, which typically leads to $\zeta^{(j)} \to \zeta^*$, corresponding to a well-mixing configuration. We refer to the instances of the AAI-CPF with the AI-CPF step corresponding to the DGI-CPF and the FDI-CPF as the adaptive DGI-CPF and FDI-CPF, respectively.

We next focus on concrete adaptations which may be used within our framework. In the case of the FDI-CPF, Algorithm 6 implements a stochastic approximation variant (Andrieu and Moulines 2006) of the adaptive Metropolis covariance adaptation of Haario et al. (2001).

Here, $\eta_j$ are step sizes that decay to zero, $\zeta_j = (\mu_j, \Sigma_j)$ the estimated mean and covariance of the smoothing distribution, respectively, and $Q_\zeta = Q_{c\Sigma}^{\mathrm{RW}}$ where $c > 0$ is a scaling factor of the covariance $\Sigma$. In the case of random walk

**Algorithm 6** $\textsc{Adapt}_{FDI,AM}\big((\mu, \Sigma), (B_1, W_1^{(1:N)}, X_1^{(1:N)}), j\big)$

```
$\mu_* \leftarrow (1-\eta_j)\mu + \eta_j X_1^{(B_1)}$.
$\Sigma_* \leftarrow (1-\eta_j)\Sigma + \eta_j (X_1^{(B_1)} - \mu)(X_1^{(B_1)} - \mu)^{\mathrm{T}}$.
output $(\mu_*, \Sigma_*)$.
```

Metropolis, this scaling factor is usually taken as $2.38^2/d$ (Gelman et al. 1996), where $d$ is the state dimension of the model. In the present context, however, the optimal value of $c > 0$ appears to depend on the model and on the number of particles $N$. This adaptation mechanism can be used both with PickPath- AT and with PickPath- BS, but may require some manual tuning to find a suitable $c > 0$.

Algorithm 7 details another adaptation for the FDI-CPF, which is intended to be used together with PickPath- BS only. Here, $\zeta_j = (\mu_j, \Sigma_j, \delta_j)$ contains the estimated mean, covariance and the scaling factor, and $Q_\zeta = Q^{\mathrm{RW}}_{C(\zeta)}$, where $C(\zeta) = e^\delta \Sigma$.

**Algorithm 7** $\textsc{Adapt}_{FDI,ASWAM}\big((\mu, \Sigma, \delta), (B_1, W_1^{(1:N)}, X_1^{(1:N)}), j\big)$

```
$\mu_* \leftarrow (1-\eta_j)\mu + \eta_j \sum_{i=1}^N W_1^{(i)} X_1^{(i)}$.
$\Sigma_* \leftarrow (1-\eta_j)\Sigma + \eta_j \sum_{i=1}^N W_1^{(i)} (X_1^{(i)} - \mu)(X_1^{(i)} - \mu)^{\mathrm{T}}$.
$\delta_* \leftarrow \delta + \eta_j(\alpha - \alpha_*)$ where $\alpha = 1 - W_1^{(1)}$.
output $(\mu_*, \Sigma_*, \delta_*)$.
```

**Algorithm 8** $\textsc{Adapt}_{DGI,AS}\big(\zeta, (B_1, W_1^{(1:N)}, X_1^{(1:N)}), j\big)$

```
$\zeta_* \leftarrow \zeta + \eta_j(\alpha - \alpha_*)$ where $\alpha = 1 - W_1^{(1)}$.
output $\zeta_*$.
```

This algorithm is inspired by a Rao–Blackwellised variant of the adaptive Metropolis within adaptive scaling method (cf. Andrieu and Thoms 2008), which is applied with standard random walk Metropolis. We use all particles with their backward sampling weights to update the mean $\mu$ and covariance $\Sigma$, and an 'acceptance rate' $\alpha$, that is, the probability that the first coordinate of the reference trajectory is not chosen. Recall that after the AI–CPF in Algorithm 5 has been run, the first coordinate of the reference trajectory and its associated weight reside in the first index of the particle and weight vectors contained in $\xi^{(j)}$.

The optimal value of the acceptance rate parameter $\alpha_*$ is typically close to one, in contrast with random walk Metropolis, where $\alpha_* \in [0.234, 0.44]$ are common (Gelman et al. 1996). Even though the optimal value appears to be problem-dependent, we have found empirically that $0.7 \le \alpha_* \le 0.9$ often leads to reasonable mixing. We will show empirical evidence for this finding in Sect. 4.

Algorithm 8 describes a similar adaptive scaling type mechanism for tuning $\beta = \mathrm{logit}^{-1}(\zeta)$ in the DGI-CPF, with $Q_\zeta = Q^{\mathrm{AR}}_\beta$. The algorithm is most practical with PickPath-BS.

We conclude this section with a consistency result for Algorithm 5, using the adaptation mechanisms in Algorithms 6 and 7. In Theorem 1, we denote $(\mu_j, \Sigma_j) = \zeta_j$ in the case of Algorithm 6, and $(\mu_j, \Sigma_j, \delta_j) = \zeta_j$ with Algorithm 7.

**Theorem 1** *Suppose $D$ is a compact set, a uniform mixing condition (Assumption 2 in Appendix A) holds, and there exists an $\epsilon > 0$ such that for all $j \ge 1$, the smallest eigenvalue $\lambda_{\min}(\Sigma_j) \ge \epsilon$, and with Algorithm 7 also $\delta_j \in [\epsilon, \epsilon^{-1}]$. Then, for any bounded function $f : \mathsf{X} \to \infty$,*

$$\frac{1}{n}\sum_{k=1}^n f(\dot{x}_{1:T}^{(k)}) \xrightarrow{n\to\infty} \pi(f). \qquad \textit{almost surely.}$$

The proof of Theorem 1 is given in Appendix A. The proof is slightly more general, and accomodates for instance $t$-distributed instead of Gaussian proposals for the FDI-CPF. We note that the latter stability condition, that is, existence of the constant $\epsilon > 0$, may be enforced by introducing a 'rejection' mechanism in the adaptation; see the end of Appendix A. However, we have found empirically that the adaptation is stable also without such a stabilisation mechanism.

### 3.4 Use within particle Gibbs

Typical application of HMMs in statistics involves not only smoothing, but also inference of a number of 'hyperparameters' $\theta$, with prior density $\mathrm{pr}(\theta)$, and with

$$\begin{aligned} \gamma_\theta(x_{1:T}) &= p(y_{1:T}, x_{1:T} \mid \theta) \qquad (6)\\ &= M_1(x_1) G_1^{(\theta)}(x_1) \prod_{k=2}^T M_k^{(\theta)}(x_{k-1}, x_k) G_k^{(\theta)}(x_{k-1}, x_k). \end{aligned}$$

The full posterior, $\check{\pi}(\theta, x_{1:T}) \propto \mathrm{pr}(\theta)\gamma_\theta(x_{1:T})$ may be inferred with the particle Gibbs (PG) algorithm of Andrieu et al. (2010). (We assume here that $M_1$ is diffuse, and thereby independent of $\theta$.)

The PG alternates between (Metropolis-within-)Gibbs updates for $\theta$ conditional on $x_{1:T}$, and CPF updates for $x_{1:T}$ conditional on $\theta$. The (A)AI-CPF applied with $M_{2:T}^{(\theta)}$ and $G_{1:T}^{(\theta)}$ may be used as a replacement of the CPF steps in a PG. Another adaptation, independent of the AAI-CPF, may be used for the hyperparameter updates (cf. Vihola 2020).

Algorithm 9 summarises a generic adaptive PG with the AAI-CPF. Line 2 involves an update of $\theta^{(j-1)}$ to $\theta^{(j)}$ using transition probabilities $K_{\zeta_\theta}(\,\cdot\,,\,\cdot \mid x_{1:T})$ which leave

$\check{\pi}(\theta \mid x_{1:T})$ invariant, and Line 3 is (optional) adaptation. This could, for instance, correspond to the robust adaptive Metropolis algorithm (RAM) (Vihola 2012). Lines 4 and 5 implement the AAI-CPF. Note that without Lines 3 and 5, Algorithm 9 determines a $\check{\pi}$-invariant transition rule.

**Algorithm 9** AAI-PG$(\theta^{(0)}, \dot{x}_{1:T}^{(0)}; Q_{C(\zeta^{(0)})}, M_{2:T}, G_{1:T}, N)$

```
1: for $j = 1, \ldots, n$ do
2:     $(\theta^{(j)}, \xi_\theta^{(j)}) \sim K_{\zeta_\theta^{(j-1)}}(\theta^{(j-1)}, \cdot \mid \dot{x}_{1:T}^{(j-1)})$.
3:     $\zeta_\theta^{(j)} \leftarrow \textsc{Adapt}_\theta(\zeta_\theta^{(j-1)}, \theta^{(j)}, \xi_\theta^{(j)})$.
4:     $(\dot{x}_{1:T}^{(j)}, \xi^{(j)}) \leftarrow \text{AI-CPF}(\dot{x}_{1:T}^{(j-1)}; Q_{\zeta^{(j-1)}}, M_{2:T}^{(\theta^{(j)})}, G_{1:T}^{(\theta^{(j)})}, N)$.
5:     $\zeta^{(j)} \leftarrow \textsc{Adapt}(\zeta^{(j-1)}, \xi^{(j)}, j)$.
6: end for
7: output $\big((\theta^{(1)}, \dot{x}_{1:T}^{(1)}), \ldots, (\theta^{(n)}, \dot{x}_{1:T}^{(n)})\big)$.
```

## 4 Experiments

In this section, we study the application of the methods presented in Sect. 3 in practice. Our focus will be on the case of the bootstrap filter, that is, $M_1(\mathrm{d}x_1) = p(x_1)\mathrm{d}x_1$, $M_k(x_{k-1}, \mathrm{d}x_k) = p(x_k \mid x_{k-1})\mathrm{d}x_k$ and $G_k(x_{1:k}) = p(y_k \mid x_k)$.

We start by investigating two simple HMMs: the noisy random walk model (RW), that is, (4) with $\rho = 1$, and the following stochastic volatility (SV) model:

$$\begin{aligned} x_{k+1} &= x_k + \eta_k, \\ y_k &= e^{x_k}\epsilon_k, \end{aligned} \tag{7}$$

with $x_1 \sim N(0, \sigma_1^2)$, $\eta_k \sim N(0, \sigma_x^2)$ and $\epsilon_k \sim N(0, \sigma_y^2)$. In Sect. 4.3, we study the dependence of the method with varying dimension, with a static multivariate normal model. We conclude in Sect. 4.4 by applying our methods in a realistic inference problem related to modelling the COVID-19 epidemic in Finland.

### 4.1 Comparing DGI-CPF and CPF-BS

We first studied how the DGI-CPF performs in comparison to the CPF-BS when the initial distributions of the RW and SV model are diffuse. Since the efficiency of sampling is affected by both the values of the model parameters (cf. Fig. 1) and the number of particles $N$, we experimented with a range of values $N \in \{8, 16, 32, 64, 128, 256, 512\}$ for which we applied both methods with $n = 10000$ iterations plus 500 burn-in. We simulated data from both the RW and SV models with $T = 50$, $x_1 = 0$, $\sigma_y = 1$ and varying $\sigma_x \in \{0.01, 0.05, 0.1, 0.5, 1, 2, 5, 10, 20, 50, 100, 200\}$. We then applied both methods for each dataset with the corresponding $\sigma_x$, but with varying $\sigma_1 \in \{10, 50, 100, 200, 500, 1000\}$, to study the sampling efficiency under different parameter configurations ($\sigma_x$ and $\sigma_1$). For the DGI-CPF, we varied the parameter $\beta \in \{0.01, 0.02, \ldots, 0.99\}$. We computed the estimated integrated autocorrelation time (IACT) of the simulated values of $x_1$ and scaled this by the number of particles $N$. The resulting quantity, the inverse relative efficiency (IRE), measures the asymptotic efficiencies of estimators with varying computational costs (Glynn and Whitt 1992).

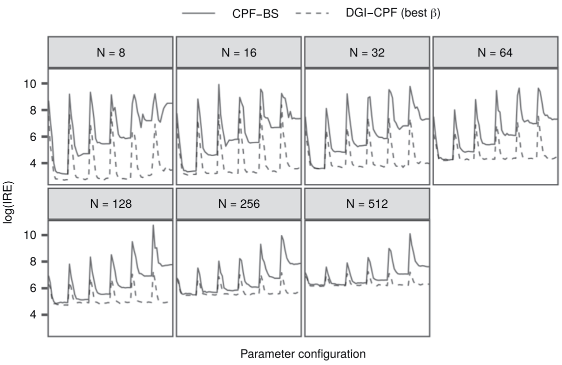


**Fig. 2** The log (IRE) resulting from the application of the CPF-BS and the best case DGI-CPF to the RW model. The horizontal axis depicts different configurations of $\sigma_1$ and $\sigma_x$, and in each panel $N$ varies

Figure 2 shows the comparison of the CPF-BS with the best DGI-CPF, that is, the DGI-CPF with the $\beta$ that resulted in the lowest IACT for each parameter configuration and $N$.

The results indicate that with $N$ fixed, a successful tuning of $\beta$ can result in greatly improved mixing in comparison with the CPF-BS. While the performance of the CPF-BS approaches that of the best DGI-CPF with increasing $N$, the difference in performance remains substantial with parameter configurations that are challenging for the CPF-BS.

The optimal $N$ which minimizes the IRE depends on the parameter configuration. For ‘easy’ configurations (where IRE is small), even $N = 8$ can be enough, but more ‘difficult’ configurations (where IRE is large), higher values of $N$ can be optimal. Similar results for the SV model are shown in Online Resource 1 (Fig. 1), and lead to similar conclusions.

The varying ‘difficulty’ of the parameter configurations is further illustrated in Fig. 3, which shows the log (IACT) for the SV model with $N = 256$ particles. The CPF-BS performed the worst when the initial distribution was very diffuse with respect to the state noise $\sigma_x$, as expected. In contrast, the well-tuned DGI-CPF appears rather robust with respect to changing parameter configuration. The observations were similar with other $N$, and for the RW model; see Online Resource 1 (Fig. 2).

The results in Figs. 2 and 3 illustrate the potential of the DGI-CPF, but are overly optimistic because in practice, the

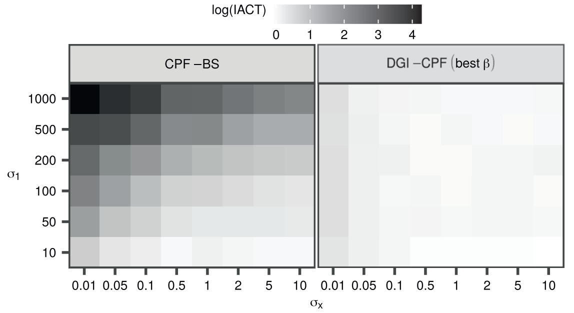


**Fig. 3** The log (IACT) of the CPF-BS (left) and the best case DGI-CPF (right) with respect to $\sigma_1$ and $\sigma_x$ in the case of the SV model and $N = 256$

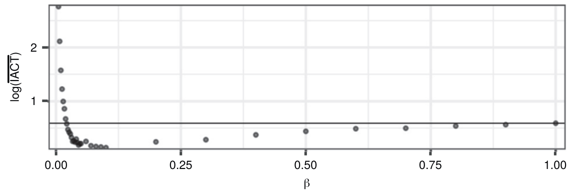


**Fig. 4** The logarithm of the mean IACT over 5 replicate runs of the DGI-CPF with respect to varying $\beta$. The dataset was simulated from the SV model with parameters $\sigma_x = 1$ and $\sigma_1 = 50$ and fixed in each replicate run of the algorithm. $N$ was set to 128. The horizontal line depicts the performance of the CPF-BS

$\beta$ parameter of the DGI-CPF cannot be chosen optimally. Indeed, the choice of $\beta$ can have a substantial effect on the mixing. Figure 4 illustrates this in the case of the SV model by showing the logarithm of the mean IACT over replicate runs of the DGI-CPF, for a range of $\beta$. Here, a $\beta$ of approximately 0.125 seems to yield close to optimal performance, but if the $\beta$ is chosen too low, the sampling efficiency is greatly reduced, rendering the CPF-BS more effective.

This highlights the importance of choosing an appropriate value for $\beta$, and motivates our adaptive DGI-CPF, that is, Algorithm 5 together with Algorithm 8. We explored the effect of the target acceptance rate $\alpha_* \in \{0.01, 0.02, \ldots, 1\}$, with the same datasets and parameter configurations as before. Figure 5 summarises the results for both the SV and RW models, in comparison with the CPF-BS. The results indicate that with a wide range of target acceptance rates, the adaptive DGI-CPF exhibits improved mixing over the CPF-BS. When $N$ increases, the optimal values for $\alpha_*$ appear to tend to one. However, in practice, we are interested in a moderate $N$, for which the results suggest that the best candidates for values of $\alpha_*$ might often be found in the range from 0.7 to 0.9.

For the CPF-BS, the mean IRE is approximately constant, which might suggest that the optimal number of particles is more than 512. In contrast, for an appropriately tuned DGI-CPF, the mean IRE is optimised by $N = 32$ in this experiment.

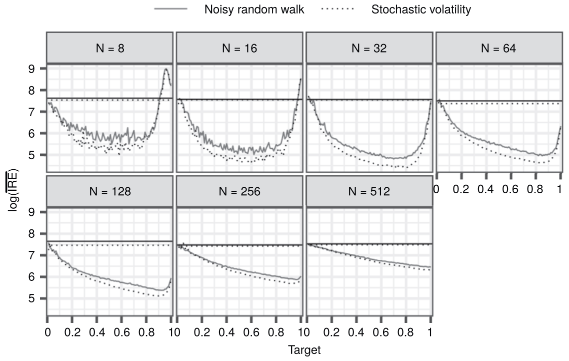


**Fig. 5** The logarithm of the mean IRE over the parameter configurations with the adaptive DGI-CPF and varying target acceptance rates. The horizontal lines depict the mean performance of the CPF-BS

## 4.2 Comparing FDI-CPF and particle Gibbs

Next, we turn to study a fully diffuse initialisation. In this case, $M_1$ is improper, and we cannot use the CPF directly. Instead, we compare the performance of the adaptive FDI-CPF with what we call the diffuse particle Gibbs (DPG-BS) algorithm. The DPG-BS is a standard particle Gibbs algorithm, where the first latent state $x_1$ is regarded as a 'parameter', that is, the algorithm alternates between the update of $x_1$ conditional on $x_{2:T}$ using a random walk Metropolis-within-Gibbs step, and the update of the latent state variables $x_{2:T}$ conditional on $x_1$ using the CPF-BS. We also adapt the Metropolis-within-Gibbs proposal distribution $Q_{\text{DPG}}$ of the DPG-BS, using the RAM algorithm (cf. Vihola 2020). For further details regarding our implementation of the DPG-BS, see Appendix B.

We used a similar simulation experiment as with the adaptive DGI-CPF in Sect. 4.1, but excluding $\sigma_1$, since the initial distribution was now fully diffuse. The target acceptance rates in the FDI-CPF with the ASWAM adaptation were again varied in $\alpha_* \in \{0.01, 0.02, \ldots, 1\}$ and the scaling factor in the AM adaptation was set to $c = 2.38^2$. In the DPG-BS, the target acceptance rate for updates of the initial state using the RAM algorithm was fixed to 0.441 following Gelman et al. (1996).

Figure 6 shows results with the RW model for the DPG-BS, the FDI-CPF with the AM adaptation, and the FDI-CPF with the ASWAM adaptation using the best value for $\alpha_*$. The FDI-CPF variants appear to perform better and improve upon the performance of the DPG-BS especially with small $\sigma_x$. Similar to Figs. 2 and 3, the optimal $N$ minimizing the IRE depends on the value of $\sigma_x$: smaller values of $\sigma_x$ call for higher number of particles.

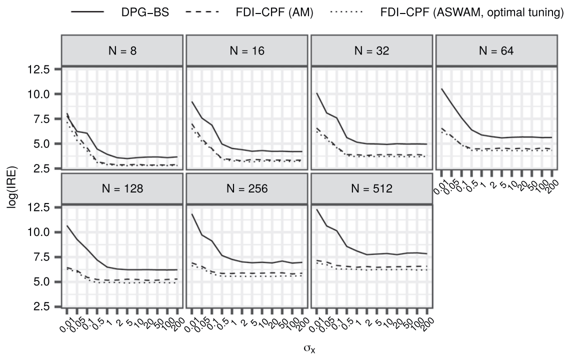


**Fig. 6** The log (IRE) for the DPG-BS, FDI-CPF with the AM adaptation and the best case FDI-CPF with the ASWAM adaptation to the datasets generated with varying $\sigma_x$ from the RW model

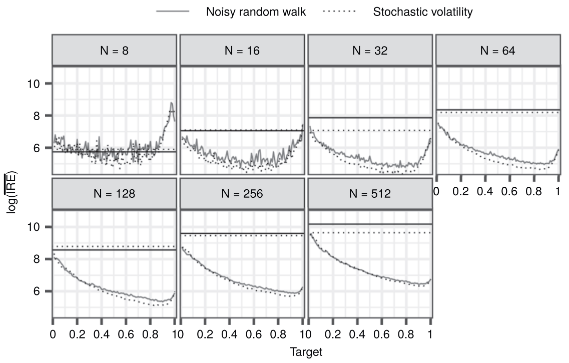


**Fig. 7** A comparison of the FDI-CPF with the ASWAM adaptation against the DPG-BS. The horizontal axis shows the target acceptance rate $\alpha_*$ used in the adaptive FDI-CPF. The logarithm of the mean IRE on the vertical axis is computed over the different $\sigma_x$ values. The black horizontal lines show the performance with the DPG-BS

The performance of the adaptive FDI-CPF appears similar regardless of the adaptation used, because the chosen scaling factor $c = 2.38^2$ for a univariate model was close to the optimal value found by the ASWAM variant in this example. We experimented also with $c = 1$, which led to less efficient AM, in the middle ground between the ASWAM and the DPG-BS.

The IACT for the DPG-BS stays approximately constant with increasing $N$, which results in a log (IRE) that increases roughly by a constant as $N$ increases. This is understandable, because in the limit as $N \to \infty$, the CPF-BS (within the DPG-BS) will correspond to a Gibbs step, that is, a perfect sample of $x_{2:T}$ conditional on $x_1$. Because of the strong correlation between $x_1$ and $x_2$, even an 'ideal' Gibbs sampler remains inefficient, and the small variation seen in the panels for the DPG-BS is due to sampling variability. The results for the SV model, with similar findings, are shown in Online Resource 1 (Fig. 3).

Figure 7 shows the logarithm of the mean IRE of the FDI-CPF with the ASWAM adaptation with respect to varying target acceptance rate $\alpha_*$. The results are reminiscent of Fig. 5 and show that with a moderate fixed $N$, the FDI-CPF with the ASWAM adaptation outperforms the DPG-BS with a wide range of values for $\alpha_*$. The optimal value of $\alpha_*$ seems to tend to one as $N$ increases, but again, we are mostly concerned with moderate $N$. For a well-tuned FDI-CPF the minimum mean IRE is found when $N$ is roughly between 32 and 64.

### 4.3 The relationship between state dimension, number of particles and optimal target acceptance rate

A well chosen value for the target acceptance rate $\alpha_*$ appears to be key for obtaining good performance with the adaptive DGI-CPF and the FDI-CPF with the ASWAM adaptation. In Sects. 4.1–4.2, we observed a relationship between $N$ and the optimal target acceptance rate, denoted here by $\alpha_{\text{opt}}$, with two univariate HMMs. It is expected that $\alpha_{\text{opt}}$ is generally somewhat model-dependent, but in particular, we suspected that the methods might behave differently with models of different state dimension $d$.

In order to study the relationship between $N$, $d$ and $\alpha_{\text{opt}}$ in more detail, we considered a simple multivariate normal model with $T = 1$, $M_1(x) \propto 1$, and $G_1(x_1) = N(x_1; 0, \sigma I_d)$, the density of $d$ independent normals. We conducted a simulation experiment with 6000 iterations plus 500 burn-in. We applied the FDI-CPF with the ASWAM adaptation with all combinations of $N \in \{2^4, 2^5, \ldots, 2^{11}\}$, $\alpha_* \in \{0.01, 0.02, \ldots, 1\}$, $\sigma \in \{1, 5, 10, 50, 100\}$, and with dimension $d \in \{1, 2, \ldots, 10\}$. Unlike before, we monitor the IACT over the samples of $x_1$ as an efficiency measure.

Figure 8 summarises the results of this experiment. With a fixed state dimension, $\alpha_{\text{opt}}$ tended towards 1 with increasing numbers of particles $N$, as observed with the RW and SV models above. With a fixed number of particles $N$, $\alpha_{\text{opt}}$ appears to get smaller with increasing state dimension $d$, but the change rate appears slower with higher $d$. Again, with moderate values for $N$ and $d$, the values in the range 0.7–0.9 seem to yield good performance.

Figure 9 shows a different view of the same data: $\text{logit}(\alpha_{\text{opt}})$ is plotted with respect to $\log(N)$ and $d$. Here, we computed $\alpha_{\text{opt}}$ by taking the target acceptance rate that produced the lowest IACT in the simulation experiment, for each value of $\sigma$, $N$ and $d$. At least with moderate $\alpha_{\text{opt}}$ and $N$, there appears to be a roughly linear relationship between $\text{logit}(\alpha_{\text{opt}})$ and $\log(N)$, when $d$ is fixed. However, because of the lack of theoretical backing, we do not suggest to use such a simple model for choosing $\alpha_{\text{opt}}$ in practice.

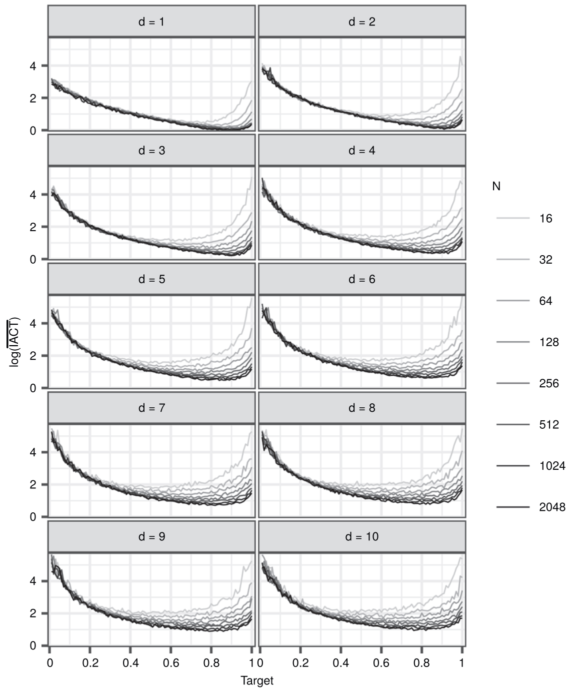


**Fig. 8** The effect of state dimension $d$, number of particles $N$ and target acceptance rate $\alpha_*$ on the logarithm of the mean IACT in the multivariate normal model. The means are computed over the different $\sigma$ in the simulation experiment

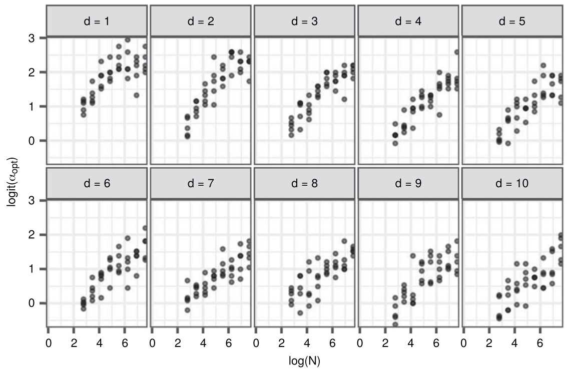


**Fig. 9** The best target acceptance rate $\alpha_{\text{opt}}$ with respect to the number of particles $N$ and state dimension $d$ on the multivariate normal model

### 4.4 Modelling the COVID-19 epidemic in Finland

Our final experiment is a realistic inference problem arising from the modelling of the progress of the COVID-19 epidemic in Uusimaa, the capital region of Finland. Our main interest is in estimating the time-varying transmission rate, or the basic reproduction number $\mathscr{R}_0$, which is expected to change over time, because of a number of mitigation actions and social distancing. The model consists of a discrete-time 'SEIR' stochastic compartment model, and a dynamic model for $\mathscr{R}_0$; such epidemic models have been used earlier in different contexts (e.g. Shubin et al. 2016).

We use a simple SEIR without age/regional stratification. That is, we divide the whole population $N_{\text{pop}}$ to four separate states: susceptible ($S$), exposed ($E$), infected ($I$) and removed ($R$), so that $N_{\text{pop}} = S + E + I + R$, and assume that $N_{\text{pop}}$ is constant. We model the transformed $\mathscr{R}_0$, denoted by $\rho$, such that $\mathscr{R}_0 = \mathscr{R}_0{}^{\max}\text{logit}^{-1}(\rho)$, where $\mathscr{R}_0{}^{\max}$ is the maximal value for $\mathscr{R}_0$. The state vector of the model at time $k$ is, therefore, $X_k = (S_k, E_k, I_k, R_k, \rho_k)$. One step of the SEIR is:

$$
\begin{aligned}
S_{k+1} &= S_k - \Delta E_{k+1}, \\
E_{k+1} &= E_k + \Delta E_{k+1} - \Delta I_{k+1}, \\
I_{k+1} &= I_k + \Delta I_{k+1} - \Delta R_{k+1}, \\
R_{k+1} &= R_k + \Delta R_{k+1}, \\
\rho_{k+1} &= \rho_k + \Delta \rho_{k+1},
\end{aligned}
$$

where the increments are as distributed as follows:

$$
\begin{aligned}
\Delta E_{k+1} &\sim \text{Binomial}(S_k, p_\beta), \;\; p_\beta = 1 - \exp\left(-\beta_k (I_k / N_{\text{pop}})\right), \\
\Delta I_{k+1} &\sim \text{Binomial}(E_k, p_a), \;\; p_a = 1 - \exp\left(-a\right), \\
\Delta R_{k+1} &\sim \text{Binomial}(I_k, p_\gamma), \;\; p_\gamma = 1 - \exp\left(-\gamma\right), \\
\Delta \rho_{k+1} &\sim \text{Normal}(0, \sigma^2).
\end{aligned}
$$

Here, $\beta_k = \mathscr{R}_0{}^{\max}\text{logit}^{-1}(\rho_k) p_\gamma$ is the time-varying infection rate, and $a^{-1}$ and $\gamma^{-1}$ are the mean incubation period and recovery time, respectively. Finally, the random walk parameter $\sigma$ controls how fast $(\rho_k)_{k \geq 2}$ can change.

The data we use in the modelling consist of the daily number of individuals tested positive for COVID-19 in Uusimaa (Finnish Institute for Health and Welfare 2020). We model the counts with a negative binomial distribution dependent on the number of infected individuals:

$$
Y_k \sim \text{NegativeBinomial}\left( e p_\gamma \frac{p}{1-p} I_k, \, p \right). \tag{8}
$$

Here, the parameter $e$ denotes sampling effort, that is, the average proportion of infected individuals that are observed, and $p$ is the failure probability of the negative binomial distribution, which controls the variability of the distribution.

In the beginning of the epidemic, there is little information available regarding the initial states, rendering the diffuse initialisation a convenient strategy. We set

$$
\begin{aligned}
&M_1(S_1, E_1, I_1, R_1, \rho_1) \\
&\quad = 1(S_1 + E_1 + I_1 = N_{\text{pop}}) 1(S_1, E_1, I_1 \geq 0) 1(R_1 = 0), \quad (9)
\end{aligned}
$$

where the number of removed $R_1 = 0$ is justified because we assume all were susceptible to COVID-19, and that the epidemic has started very recently.

In addition to the state estimation, we are interested in estimating the parameters $\sigma$ and $p$. We assign the prior $N(-2.0, (0.3)^2)$ to $\log(\sigma)$ to promote gradual changes in $\mathscr{R}_0$, and an uninformative prior, $N(0, 10^2)$, for $\text{logit}(p)$. The remaining parameters are fixed to $N_{\text{pop}} = 1638469$, $\mathscr{R}_0{}^{\max} = 10$, $a = 1/3$, $\gamma = 1/7$ and $e = 0.15$, which are in part inspired by the values reported by the Finnish Institute for Health and Welfare.

We used the AAI-PG (Algorithm 9) with the FDI-CPF with the ASWAM adaptation, and a RAM adaptation (Vihola 2012) for $\sigma$ and $p$, (i.e. in the Lines 2–3 of Algorithm 9). The form of (9) leads to the version of the FDI-CPF discussed in Sect. 3.2 where the initial distribution is uniform with constraints. We use a random walk proposal to generate proposals $(\rho_1, E_1, I_1) \to (\rho_1^*, E_1^*, I_1^*)$, round $E_1^*$ and $I_1^*$ to the nearest integer, and then set $R_1^* = 0$ and $S_1^* = N_{\text{pop}} - E_1^* - I_1^* - R_1^*$. We refer to this variant of the AAI-PG as the FDI-PG algorithm. Motivated by our findings in Sects. 4.1–4.3, we set the target acceptance rate $\alpha_*$ in the FDI-CPF (within the FDI-PG) to 0.8.

As an alternative to the FDI-PG we also used a particle Gibbs algorithm that treats $\sigma$, $p$ as well as the initial states $E_1$, $I_1$ and $\rho_1$ as parameters, using the RAM to adapt the random walk proposal (Vihola 2012). This algorithm is the DPG-BS detailed in Appendix B with the difference that the parameters $\sigma$ and $p$ are updated together with the initial state, and $p^{\text{DPG}}$ additionally contains all terms of (6) which depend on $\sigma$ and $p$.

We ran both the FDI-PG and the DPG-BS with $N = 64$ a total of $n = 500,000$ iterations plus $10,000$ burn-in, and thinning of 10. Figures 10 and 11 show the first 50 autocorrelations and traceplots of $E_1$, $I_1$, $(\mathscr{R}_0)_1$, $\sigma$ and $p$, for both methods, respectively. The corresponding IACT and $\text{n}_{\text{eff}}$ as well as credible intervals for the means of these variables are shown in Table 1. The FDI-PG outperformed the DPG-BS with each variable. However, as is seen from Online Resource 1 (Fig. 4), the difference is most notable with the initial states, and the relative performance of the DPG-BS approaches that of the FDI-PG with increasing state index. The slow improvement in the mixing of the state variable $R$ occurs because of the cumulative nature of the variable in the model, and the slow mixing of early values of $I$. We note that even though the mixing with the DPG-BS was worse, the inference with $500,000$ iterations leads in practice to similar findings. However, the FDI-PG could provide reliable inference with much less iterations than the DPG-BS. The marginal density estimates of the initial states and parameters are shown in Online Resource 1 (Fig. 5). The slight discrepancies in the density estimates of $E_1$ and $I_1$ between the methods are likely because of the poor mixing of these variables with the DPG-BS.

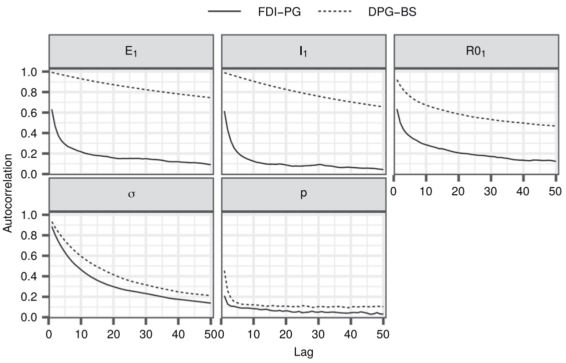


**Fig. 10** The first 50 autocorrelations for the model parameters and initial states with the FDI-PG and the DPG-BS computed after thinning the total 500000 samples to every 10th sample

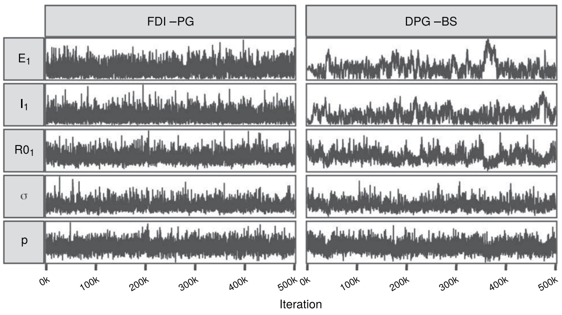


**Fig. 11** Traceplots for the initial states and model parameters for the SEIR model with the FDI-PG and the DPG-BS. The 5000 samples shown per method and parameter correspond to every 100th sample of the total 500000 samples simulated

We conclude with a few words about our findings regarding the changing transmission rate, which may be of some independent interest. Figure 12 displays the data and a posterior predictive simulation, and the estimated distribution of $\mathscr{R}_0$ computed by the FDI-PG with respect to time, with annotations about events that may have had an effect on the spread of the epidemic, and/or the data. The initial $\mathscr{R}_0$ is likely somewhat overestimated, because of the influx of infections from abroad, which were not explicitly modelled. There is an overall decreasing trend since the beginning of 'lockdown', that is, when the government introduced the first mitigation actions, including school closures. Changes in the testing criteria likely cause some bias soon after the change, but no single action or event stands out.

Interestingly, if we look at our analysis, but restrict our focus up to the end of April, we might be tempted to quantify how much certain mitigation actions contribute to the suppression of the transmission rate in order to build projections using scenario models (cf. Anderson et al. 2020).

**Table 1** The integrated autocorrelation time, effective sample size and credible intervals of the mean for the initial states and parameters in the SEIR model

| Variable | IACT | | $n_{\text{eff}}$ | | 95% mean CI | |
|---|---|---|---|---|---|---|
| | FDI-PG | DPG-BS | FDI-PG | DPG-BS | FDI-PG | DPG-BS |
| $E_1$ | 30.087 | 882.583 | 1661.838 | 56.652 | (353.888, 374.054) | (301.379, 423.106) |
| $I_1$ | 14.296 | 626.963 | 3497.603 | 79.75 | (165.697, 172.388) | (155.374, 203.458) |
| $(\mathscr{R}_0)_1$ | 32.168 | 436.755 | 1554.331 | 114.481 | (3.41, 3.513) | (3.266, 3.636) |
| $\sigma$ | 41.261 | 114.919 | 1211.796 | 435.088 | (0.15, 0.154) | (0.147, 0.154) |
| $p$ | 5.18 | 38.178 | 9652.794 | 1309.647 | (0.134, 0.135) | (0.133, 0.135) |

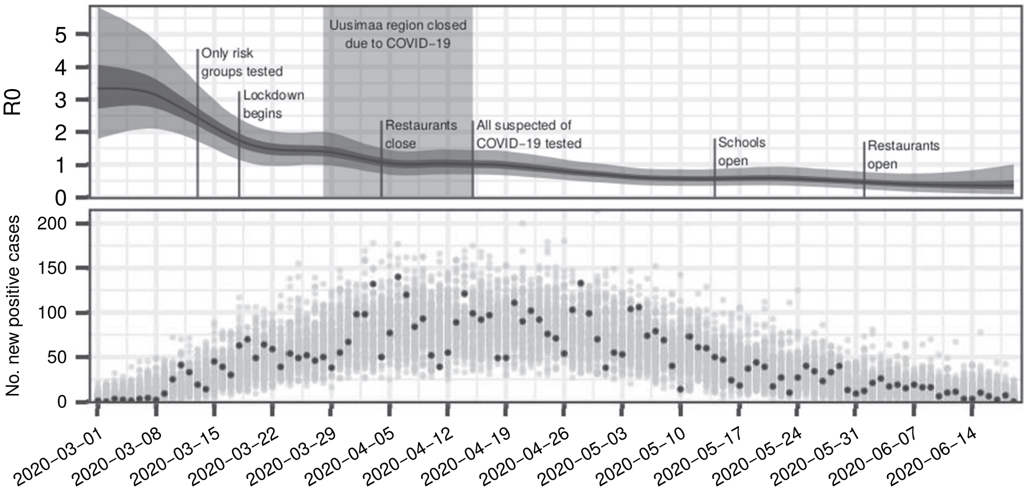


**Fig. 12** The distribution of the basic reproduction number $\mathscr{R}_0$ (top) and a posterior predictive simulation (bottom) based on the posterior distribution computed with the FDI-PG. The plot for $\mathscr{R}_0$ shows the median in black and probability intervals (75% and 95%) in shades of grey. The black points in the bottom plot represent the data used. The grey points represent observations simulated conditional on the posterior distribution of the model parameters and states

However, when the mitigation measures have been gradually lifted by opening the schools and restaurants, the openings do not appear to have had notable consequences, at least until now. It is possible that at this point, the number of infections was already so low, that it has been possible to test all suspected cases and trace contacts so efficiently, and that nearly all transmission chains have been contained. Also, the public may have changed their behaviour, and are now following the hygiene and social distancing recommendations voluntarily. Such a behaviour is, however, subject to change over time.

## 5 Discussion

We presented a simple general auxiliary variable method for the CPF for HMMs with diffuse initial distributions and focused on two concrete instances of it: the FDI-CPF for a uniform initial density $M_1$ and the DGI-CPF for a Gaussian $M_1$. We introduced two mechanisms to adapt the FDI-CPF automatically: the adaptive Metropolis (AM) of Haario et al. (2001) and a method similar to a Rao–Blackwellised adaptive scaling within adaptive Metropolis (ASWAM) (cf. Andrieu and Thoms 2008), and provided a proof of their consistency. We also suggested an adaptation for the DGI-CPF, based on an acceptance rate optimisation. The FDI-CPF or the DGI-CPF, including their adaptive variants, may be used directly within a particle Gibbs as a replacement for the standard CPF.

Our experiments with a noisy random walk model and a stochastic volatility model demonstrated that the DGI-CPF and the FDI-CPF can provide orders of magnitude speed-ups relative to a direct application of the CPF and to diffuse initialisation using particle Gibbs, respectively. Improvement was substantial also in our motivating practical example, where we applied the adaptive FDI-CPF (within particle Gibbs) in the analysis of the COVID-19 epidemic in Finland, using a stochastic 'SEIR' compartment model with changing transmission rate. Latent compartment models are, more generally, a good example where our approach can be useful: there is substantial uncertainty in the initial states, and it is difficult to design directly a modified model that leads to efficient inference.

Our adaptation schemes are based on the estimated covariance matrix and a scaling factor which can be adapted using acceptance rate optimisation. For the latter, we found empirically that with a moderate number of particles, good performance was often reached with a target acceptance rate ranging in 0.7–0.9. We emphasise that even though we found this '0.8 rule' to work well in practice, it is only a heuristic, and the optimal target acceptance rate may depend on the model of interest. Related to this, we investigated how the optimal target acceptance rate varied as a function of the number of particles and state dimension in a multivariate normal model, but did not find a clear pattern. Theoretical verification of the acceptance rate heuristic, and/or development of more refined adaptation rules, is left for future research. We note that while the AM adaptation performed well in our limited experiments, the ASWAM may be more appropriate when used within particle Gibbs (cf. Vihola 2020). The scaling of the AM remains similarly challenging, due to the lack of theory for tuning.

**Acknowledgements** We wish to acknowledge CSC, IT Center for Science, Finland, for computational resources, and thank Arto Luoma for inspiring discussions that led to the COVID-19 example.

**Funding** This work was supported by Academy of Finland Grant 315619.

**Data Availability Statement** All data analysed in this work are either freely available or available at https://nextcloud.jyu.fi/index.php/s/zjeiwDoxaegGcRe.

## Compliance with ethical standards

**Conflicts of interest** The authors declare that they have no conflict of interest.

**Code availability** The code used to produce the results in this work is freely available at https://github.com/skarppinen/cpf-diff-init.

# Appendix

## A Proof of Theorem 1

For a finite signed measure $\xi$, the total variation of $\xi$ is defined as $\|\xi\|_{\mathrm{tv}} = \sup_{\|f\|_\infty \le 1} \xi(f)$, where $\|f\|_\infty = \sup_x |f(x)|$, and the supremum is over measurable real-valued functions $f$, and $\xi(f) = \int f \mathrm{d}\xi$. For Markov transitions $P$ and $P'$, define $d(P, P') = \sup_x \|P(x, \cdot) - P'(x, \cdot)\|_{\mathrm{tv}}$.

In what follows, we adopt the following definitions:

**Definition 1** Consider Lines 3 and 4 of Algorithm 1 with $\tilde{X}_1^{(1:N)} = \tilde{x}_1^{(1:N)}$ and $\dot{x}_{2:T}$, and define:

(i) $P_{\mathrm{CPF}}(\tilde{x}_1^{(1:N)}, \dot{x}_{2:T}; \cdot)$ as the law of $\tilde{X}_{1:T}^{(B_{1:T})}$, and
(ii) (In case PickPath- BS is used:) $\tilde{P}_{\mathrm{CPF}}(\tilde{x}_1^{(1:N)}, \dot{x}_{2:T}; \cdot)$ as the law of $\big(\tilde{X}_{1:T}^{(B_{1:T})}, (B_1, V^{(1:N)}, \tilde{X}_1^{(1:N)})\big)$.

Consider then Algorithm 1 with parameterised $Q = Q_\zeta$, and define, analogously:

(iii) $P_\zeta$ is the Markov transition from $\dot{x}_{1:T}$ to $\tilde{X}_{1:T}^{(B_{1:T})}$.
(iv) $\tilde{P}_\zeta$ is the Markov transition from from $(\dot{x}_{1:T}, \cdot)$ to $\big(\tilde{X}_{1:T}^{(B_{1:T})}, (B_1, V^{(1:N)}, \tilde{X}_1^{(1:N)})\big)$.

**Lemma 1** *We have $d(P_\zeta, P_{\zeta'}) \le N d(Q_\zeta, Q_{\zeta'})$ and $d(\tilde{P}_\zeta, \tilde{P}_{\zeta'}) \le N d(Q_\zeta, Q_{\zeta'})$.*

***Proof*** Let $(\hat{P}_{\mathrm{CPF}}, \hat{P}_\zeta) \in \{(P_{\mathrm{CPF}}, P_\zeta), (\tilde{P}_{\mathrm{CPF}}, \tilde{P}_\zeta)\}$ and take measurable real-valued function $f$ on the state space of $\hat{P}_\zeta$ with $\|f\|_\infty = 1$.

We may write

$$\begin{aligned} &\hat{P}_\zeta(\dot{x}_{1:T}, f) \\ &= \int Q_\zeta(\dot{x}_1, \mathrm{d}x_0)\Big[\int \delta_{\dot{x}_1}(\mathrm{d}\tilde{x}_1^{(1)}) \\ &\quad \prod_{k=2}^{N} Q_\zeta(x_0, \mathrm{d}\tilde{x}_1^{(k)}) \hat{P}_{\mathrm{CPF}}(\tilde{x}_1^{(1:N)}, \dot{x}_{2:T}, f)\Big], \end{aligned} \tag{10}$$

and therefore, upper bound

$$\begin{aligned} &|\hat{P}_\zeta(\dot{x}_{1:T}, f) - \hat{P}_{\zeta'}(\dot{x}_{1:T}, f)| \\ &\quad \le |Q_\zeta(\dot{x}_1, g_0^{(\dot{x}_{1:T})}) - Q_{\zeta'}(\dot{x}_1, g_0^{(\dot{x}_{1:T})})| \\ &\qquad + \sum_{i=2}^{N} \int Q_{\zeta'}(\dot{x}_1, \mathrm{d}x_0)|Q_\zeta(x_0, g_i^{(\dot{x}_{1:T}, x_0)}) \\ &\qquad -Q_{\zeta'}(x_0, g_i^{(\dot{x}_{1:T}, x_0)})| \end{aligned}$$

with functions defined below, which satisfy $\|g_0^{(\dot{x}_{1:T})}\|_\infty \le 1$ and $\|g_i^{(\dot{x}_{1:T},x_0)}\|_\infty \le 1$:

$$
\begin{aligned}
g_0^{(\dot{x}_{1:T})}(x_0) &= \int \delta_{\dot{x}_1}(\mathrm{d}\tilde{x}_1^{(1)}) \prod_{k=2}^{N} Q_\zeta(x_0, \mathrm{d}\tilde{x}_1^{(k)}) \\
&\qquad P_{\mathrm{CPF}}(\tilde{x}_1^{(1:N)}, \dot{x}_{2:T}, f), \\
g_i^{(\dot{x}_{1:T},x_0)}(\tilde{x}_1^{(i)}) &= \delta_{\dot{x}_1}(\mathrm{d}\tilde{x}_1^{(1)}) \prod_{k=2}^{i-1} Q_{\zeta'}(x_0, \mathrm{d}\tilde{x}_1^{(k)}) \\
&\qquad \prod_{k=i+1}^{N} Q_\zeta(x_0, \mathrm{d}\tilde{x}_1^{(k)}) P_{\mathrm{CPF}}(\tilde{x}_1^{(1:N)}, \dot{x}_{2:T}, f).
\end{aligned}
$$

□

The following result is direct:

**Lemma 2** *Let $Q_\Sigma$ stand for the random walk Metropolis type kernel with increment proposal distribution $q_\Sigma$, and with target function $M_1 \ge 0$, that is, a transition probability of the form:*

$$
\begin{aligned}
Q_\Sigma(x, A) &= \int_A q_\Sigma(\mathrm{d}z) \min\left\{1, \frac{M_1(x+z)}{M_1(x)}\right\} \\
&+ 1(x \in A)\left(1 - \int q_\Sigma(\mathrm{d}z) \min\left\{1, \frac{M_1(x+z)}{M_1(x)}\right\}\right).
\end{aligned}
$$

*Then,* $\|Q_\Sigma(x, \cdot\,) - Q_{\Sigma'}(x, \cdot\,)\|_{\mathrm{tv}} \le 2\|q_\Sigma - q_{\Sigma'}\|_{\mathrm{tv}}$.

The following result is from (Vihola 2011, proof of Proposition 26):

**Lemma 3** *Let $q_\Sigma(x, \mathrm{d}y)$ stand for the centred Gaussian distribution with covariance $\Sigma$, or the centred multivariate $t$-distribution with shape $\Sigma$ and some constant degrees of freedom $\nu > 0$. Then, for any $0 < b_\ell < b_u < \infty$ there exists a constant $c = c(b_\ell, b_u) < \infty$ such that for all $\Sigma, \Sigma'$ with all eigenvalues within $[b_\ell, b_u]$,*

$$
\|q_\Sigma - q_{\Sigma'}\|_{\mathrm{tv}} \le c\|\Sigma - \Sigma'\|,
$$

*where the latter stands for the Frobenius norm in $\mathbb{R}^d$.*

**Assumption 2** (Mixing) The potentials are bounded:

(i) $\|G_k\|_\infty < \infty$ for all $k = 1, \ldots, T$.

Furthermore, there exists $\epsilon > 0$ and probability measures $\nu_\zeta$ such that for all $\zeta \in \mathsf{Z}$:

(ii) $Q_\zeta(x_0, A) \ge \epsilon\nu_\zeta(A)$ for all $x_0 \in \mathsf{X}$ and measurable $A \subset \mathsf{X}$.
(iii) $\int \nu_\zeta(\mathrm{d}x_0) Q_\zeta(x_0, \mathrm{d}x_1)$ $G_1(x_1) \prod_{k=2}^{T} M_k(x_{k-1}, \mathrm{d}x_k) G_k(x_{k-1}, x_k) \mathrm{d}x_{1:T} \ge \epsilon$.

**Lemma 4** *Suppose that Assumption 2 holds, then the kernels $P_\zeta$ and $\tilde{P}_\zeta$ satisfy simultaneous minorisation conditions, that is, there exists $\delta > 0$ and probability measures $\nu_\zeta, \tilde{\nu}_\zeta$, such that*

$$
P_\zeta^k(x_{1:T}, \cdot\,) \ge \delta\nu_\zeta(\cdot\,) \quad \textit{and} \quad \tilde{P}_\zeta^k(\tilde{x}_{1:T}, \cdot\,) \ge \delta\tilde{\nu}_\zeta(\cdot\,),
$$

*for all $x_{1:T} \in \mathsf{X}$, $\tilde{x}_1^{(1:N)} \in \mathsf{X}^N$, and $\zeta \in \mathsf{Z}$.*

***Proof*** For $\hat{P}_\zeta \in \{P_\zeta, \tilde{P}_\zeta\}$, we may write as in the proof of Lemma1

$$
\hat{P}_\zeta(x_{1:T}, \cdot\,) = \int Q_\zeta(x_1, \mathrm{d}x_0) \hat{P}^*_{\mathrm{CPF},\zeta,x_0}(x_{1:T}, \cdot\,),
$$

where the latter term refers to the term in brackets in (10) — the transition probability of a conditional particle filter, with reference $x_{1:T}$, and the Feynman–Kac model $\check{M}_1^{(\zeta,x_0)}(\mathrm{d}x_1) = Q_\zeta(x_0, \mathrm{d}x_1)$, $M_{2:T}$ and $G_{1:T}$, whose normalised probability we call $\pi^*_{\zeta,x_0}$. Assumption 2, 2 and 2 guarantee that $P^*_{\mathrm{CPF},\zeta,x_0}(x_{1:T}, \mathrm{d}x'_{1:T}) \ge \varepsilon\pi^*_{\zeta,x_0}(\mathrm{d}x'_{1:T})$, where $\hat{\epsilon} > 0$ is independent of $x_0$ and $\zeta$ (Andrieu et al. 2018, Corollary 12). Note that the same conclusion holds also with backward sampling, because it is only a further Gibbs step to the standard CPF. Likewise, in case of $\tilde{P}_\zeta$, the result holds because we may regard $\tilde{P}_\zeta$ as an augmented version of $P_\zeta$ (e.g. Franks and Vihola 2020). We conclude that

$$
\hat{P}_\zeta(x_{1:T}, \cdot\,) \ge \epsilon\hat{\epsilon} \int \nu_\zeta(\mathrm{d}x_0) \pi^*_{\zeta,x_0}(\cdot\,),
$$

where the integral defines a probability measure independent of $x_{1:T}$. □

We may write the $k$:th step of Algorithm 5 as:

(i) $(X_k, \xi_k) \sim \tilde{P}_{\zeta_{k-1}}(X_{k-1}, \cdot\,)$,
(ii) $\zeta_k^* = \zeta_{k-1} + \eta_k H(\zeta_{k-1}, X_k, \xi_k)$,

where $H$ correspond to Algorithm 6 or 7, respectively. The stability may be enforced by introducing the following optional step:

(iii) $\zeta_k = \zeta_k^* 1(\zeta_k \in \mathsf{Z}) + \zeta_{k-1} 1(\zeta_k^* \notin \mathsf{Z})$,

which ensures that $\zeta \in \mathsf{Z}$, the feasible set for adaptation.

***Proof*** (Proof of Theorem 1) The result follows by (Saksman and Vihola 2010, Theorem 2), as (A1) is direct, Lemma 4 implies (A2) with $V \equiv 1$, $\lambda_n = 0$, $b_n = 1$, $\delta_n = \delta$ and $\epsilon = 0$, Lemmas 2 and 3 imply (A3), and (A4) holds trivially, as $\|H(\cdot\,)\|_\infty < \infty$, thanks to the compactness of $D$. □

## B Details of the DPG-BS algorithm

The diffuse particle Gibbs algorithm targets (2) by alternating the sampling of $x_{2:T}$ given $x_1$, and $x_1$ given $x_{2:T}$. Hence, the algorithm is simply particle Gibbs where the initial state is treated as a parameter. Define

$$p^{\mathrm{DPG}}(x_1 \mid x_{2:T}) \propto M_1(x_1)G_1(x_1)M_2(x_2 \mid x_1)G_2(x_1, x_2).$$

With this definition, the DPG-BS algorithm can be written as in Algorithm 10. Lines 3–5 constitute a CPF-BS update for $x_{2:T}$, and Line 6 updates $x_1$. A version of the RAM algorithm (Vihola 2012) (Algorithm 11) is used for adapting the normal proposal used in sampling $x_1$ from $p^{\mathrm{DPG}}$.

**Algorithm 10** DPG-BS($X_1^{(0)}, \dot{x}_{2:T}^{(0)}; \pi, N$)

```
1: Set $S_0 = I, \alpha_* = 0.441, \eta^{\max} = 0.5, \gamma = 0.66$.
2: for $j$ in $1, \ldots, n$ do
3:     Simulate $\tilde{X}_2^{(2:N)} \sim M_2(\cdot \mid X_1^{(j-1)})$ and set $\tilde{X}_2^{(1)} = \dot{x}_2^{(0)}$.
4:     $(\tilde{X}_{2:T}^{(1:N)}, W_{2:T}^{(1:N)}, A_{2:T-1}^{(1:N)}) \leftarrow$ F-CPF$(\dot{x}_{3:T}, \tilde{X}_2^{(1:N)}; M_{3:T}, G_{2:T}, N)$.
5:     $(B_{2:T}, \xi) \leftarrow$ PickPath- BS$(\tilde{X}_{2:T}^{(1:N)}, W_{2:T}^{(1:N)}, A_{2:T-1}^{(1:N)}, M_{3:T}, G_{3:T})$
6:     $(X_1^{(j)}, S_j) \leftarrow$ RAM$(p^{\mathrm{DPG}}(\cdot \mid \tilde{X}_{2:T}^{(B_{2:T})}), X_1^{(j-1)}, S_{j-1}, \alpha_*, \eta^{\max}, \gamma)$.
7:     Set $\mathbf{X}^{(j)} = (X_1^{(j)}, \tilde{X}_2^{(B_2)}, \tilde{X}_3^{(B_3)}, \ldots, \tilde{X}_T^{(B_T)})$.
8: end for
9: output $\mathbf{X}^{(1:n)}$
```

**Algorithm 11** RAM($p, \theta^{(n-1)}, S_{n-1}, \alpha_*, \eta^{\max}, \gamma$) (iteration $n$)

```
Simulate $U_n \sim N(0, I_d)$.
Propose $\theta^* = \theta^{(n-1)} + S_{n-1}U_n$.
Compute $\alpha_n = \min\left\{1, \frac{p(\theta^*)}{p(\theta^{(n-1)})}\right\}$.
With probability $\alpha_n$, set $\theta^{(n)} = \theta^*$; otherwise set $\theta^{(n)} = \theta^{(n-1)}$.
Set $\eta_n = \min\{\eta^{\max}, dn^{-\gamma}\}$.
Compute $S_n$ such that $S_n S_n' = S_{n-1}\left(I + \eta_n(\alpha_n - \alpha_*)\frac{U_n U_n'}{\|U_n\|^2}\right)S_{n-1}'$.
output $\theta^{(n)}, S_n$.
```

## References

Anderson, R.M., Heesterbeek, H., Klinkenberg, D., Hollingsworth, T.D.: How will country-based mitigation measures influence the course of the COVID-19 epidemic? Lancet **395**(10228), 931–934 (2020)

Andrieu, C., Moulines, É.: On the ergodicity properties of some adaptive MCMC algorithms. Ann. Appl. Probab. **16**(3), 1462–1505 (2006)

Andrieu, C., Thoms, J.: A tutorial on adaptive MCMC. Statist. Comput. **18**(4), 343–373 (2008)

Andrieu, C., Doucet, A., Holenstein, R.: Particle Markov chain Monte Carlo methods. J. R. Stat. Soc. Ser. B Stat. Methodol. **72**(3), 269–342 (2010)

Andrieu, C., Lee, A., Vihola, M.: Uniform ergodicity of the iterated conditional SMC and geometric ergodicity of particle Gibbs samplers. Bernoulli **24**(2), 842–872 (2018)

Chopin, N., Singh, S.S.: On particle Gibbs sampling. Bernoulli **21**(3), 1855–1883 (2015)

Cotter, S.L., Roberts, G.O., Stuart, A.M., White, D.: MCMC methods for functions: modifying old algorithms to make them faster. Statist. Sci. **28**(3), 424–446 (2013)

Del Moral, P.: Feynman-Kac Formulae. Springer, New York (2004)

Del Moral, P., Doucet, A., Jasra, A.: Sequential Monte Carlo samplers. J. R. Stat. Soc. Ser. B Stat. Methodol. **68**(3), 411–436 (2006)

Durbin, J., Koopman, S.J.: Time Series Analysis by State Space Methods, 2nd edn. Oxford University Press, New York (2012)

Fearnhead, P., Künsch, H.R.: Particle filters and data assimilation. Ann. Rev. Stat. Its Appl. **5**, 421–449 (2018)

Fearnhead, P., Meligkotsidou, L.: Augmentation schemes for particle MCMC. Stat. Comput. **26**(6), 1293–1306 (2016)

Finnish Institute for Health and Welfare (2020) Confirmed corona cases in Finland (COVID-19). https://thl.fi/en/web/thlfi-en/statistics/statistical-databases/open-data/confirmed-corona-cases-in-finland-covid-19-, accessed on 2020-06-22

Franks, J., Vihola, M.: Importance sampling correction versus standard averages of reversible MCMCs in terms of the asymptotic variance. Stochastic Process Appl. **130**(10), 6157–6183 (2020)

Gelman, A., Roberts, G.O., Gilks, W.R.: Efficient metropolis jumping rules. Bayesian Stat. **5**, 599–607 (1996)

Glynn, P.W., Whitt, W.: The asymptotic efficiency of simulation estimators. Oper. Res. **40**(3), 505–520 (1992)

Gordon, N.J., Salmond, D.J., Smith, A.F.M.: Novel approach to nonlinear/non-Gaussian Bayesian state estimation. IEE Proceedings-F **140**(2), 107–113 (1993)

Guarniero, P., Johansen, A.M., Lee, A.: The iterated auxiliary particle filter. J. Am. Stat. Assoc. **112**(520), 1636–1647 (2017)

Haario, H., Saksman, E., Tamminen, J.: An adaptive Metropolis algorithm. Bernoulli **7**(2), 223–242 (2001)

Lee, A., Singh, S.S., Vihola, M.: Coupled conditional backward sampling particle filter. Ann. Stat. **48**(5), 3066–3089 (2020)

Lindsten, F., Jordan, M.I., Schön, T.B.: Particle Gibbs with ancestor sampling. J. Mach. Learn. Res. **15**(1), 2145–2184 (2014)

Martino, L.: A review of multiple try MCMC algorithms for signal processing. Digit. Signal Proc. **75**, 134–152 (2018)

Mendes, EF., Scharth, M., Kohn, R.: Markov interacting importance samplers. (2015) Preprint arXiv:1502.07039

Murray, L.M., Jones, E.M., Parslow, J.: On disturbance state-space models and the particle marginal Metropolis-Hastings sampler. SIAM/ASA J. Uncertain. Quantification **1**(1), 494–521 (2013)

Saksman, E., Vihola, M.: On the ergodicity of the adaptive Metropolis algorithm on unbounded domains. Ann. Appl. Probab. **20**(6), 2178–2203 (2010)

Shubin, M., Lebedev, A., Lyytikäinen, O., Auranen, K.: Revealing the true incidence of pandemic A(H1N1) pdm09 influenza in Finland during the first two seasons - an analysis based on a dynamic transmission model. PLoS Comput. Biol. **12**(3), 1–19 (2016). https://doi.org/10.1371/journal.pcbi.1004803

Vihola, M.: On the stability and ergodicity of adaptive scaling Metropolis algorithms. Stochastic Process. Appl. **121**(12), 2839–2860 (2011)

Vihola, M.: Robust adaptive Metropolis algorithm with coerced acceptance rate. Stat. Comput. **22**(5), 997–1008 (2012)

Vihola, M.: Ergonomic and reliable bayesian inference with adaptive markov chain monte carlo. In: Balakrishnan, N., Colton, T., Everitt, B., Piegorsch, W., Ruggeri, F., Teugels, J. L. (eds.) Wiley statsRef : statistics reference online, pp. 1–12. Wiley (2020). https://doi.org/10.1002/9781118445112.stat08286

Whiteley, N.: Discussion on "Particle Markov chain Monte Carlo methods". J. R. Stat. Soc. Ser. B Stat. Methodol. **72**(3), 306–307 (2010)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

# III

# IDENTIFYING TERRITORIES USING PRESENCE-ONLY CITIZEN SCIENCE DATA: AN APPLICATION TO THE FINNISH WOLF POPULATION

by

Karppinen, S., Rajala, T., Mäntyniemi, S., Kojola, I., and Vihola, M. 2022

Ecological Modelling, 472

DOI: https://doi.org/10.1016/j.ecolmodel.2022.110101

Ecological Modelling 472 (2022) 110101

Contents lists available at ScienceDirect

Ecological Modelling

journal homepage: www.elsevier.com/locate/ecolmodel

# Identifying territories using presence-only citizen science data: An application to the Finnish wolf population

Santeri Karppinen [a,*], Tuomas Rajala [b], Samu Mäntyniemi [b], Ilpo Kojola [c], Matti Vihola [a]

[a] Department of Mathematics and Statistics, University of Jyväskylä, P.O. Box 35, FI-40014, Jyväskylä, Finland
[b] Natural Resources Institute Finland (Luke), Latokartanonkaari 9, FI-00790, Helsinki, Finland
[c] Natural Resources Institute Finland (Luke), Ounasjoentie 6, FI-96200, Rovaniemi, Finland

## ARTICLE INFO

Dataset link: https://github.com/antiphon/tassu-intensity, https://github.com/skarppinen/tassu-filtering

*Keywords:*
Citizen science data
Bayesian statistics
Sequential Monte Carlo
Spatio-temporal model
Territory identification
Presence-only data

## ABSTRACT

Citizens, community groups and local institutions participate in voluntary biological monitoring of population status and trends by providing species data e.g. for regulations and conservation. Sophisticated statistical methods are required to unlock the potential of such data in the assessment of wildlife populations.

We develop a statistical modelling framework for identifying territories based on presence-only citizen science data. The framework can be used to jointly estimate the number of active animal territories and their locations in time. Our approach is based on a data generating model which consists of a dynamic submodel for the appearance/removal of territories and an observation submodel that accounts for the varying observation intensity and links the data to the territories. We first estimate the observation intensity using past presence-only observations made by citizens, conditioning on previously known territories. We then infer the territories using a state-of-the-art sequential Monte Carlo method, which extends earlier approaches by allowing for spatial inhomogeneity in the observation process.

We verify our data generating model and inference method successfully in synthetic scenarios. We apply our framework for estimating the locations and number of wolf territories in March 2020 in Finland using one year of confirmed citizen-made wolf observations. The observation intensity is estimated using wolf observation data collected in 2011–2019, conditioning on official territory estimates and data from GPS-collared wolves.

Our experiments with synthetic data suggest that the estimation of territories can be feasible with presence-only data. Our location and territory count inferences for March 2020 based on past data are comparable to the official wolf population assessment of March 2020 by the Natural Resources Institute Finland. The results suggest that the framework can provide useful information for assessing populations of territorial animals. Furthermore, our methods and findings, such as the developed data generating model and the estimation of the spatio-temporal observation intensity can be relevant also beyond the strictly territorial setting.

## 1. Introduction

Volunteers contribute to many wildlife monitoring programs but standardised monitoring schemes are available for only a small number of taxa in a few countries (Gregory et al., 2005; Isaac, 2014). Citizens, community groups and local institutions participate in biological monitoring of population status and trends by providing species data e.g. for regulations and conservation (Conrad and Hichley, 2011; Lawrence, 2006). The involvement of citizens as data collectors has demonstrated its ability to gather massive amounts of data at a spatio-temporal scale unattainable by research teams and state authorities active in biodiversity monitoring (Silvertown, 2009). For instance, in many European countries, hunters are integrated as data-providers in wildlife management structures that are intended to support sustainable harvest (Bragina et al., 2015; Cretois et al., 2020; Linnell et al., 2015).

Statistical developments in data integration as well as more rigorous protocols for data collection are needed to unlock further the potential that volunteers' data holds (Cretois et al., 2020; Isaac, 2014). The statistical interpretation of citizen-collected data faces problems less frequently encountered in traditional scientific research. For example, the spatio-temporal sampling effort of citizens is usually not known nor controllable. Sophisticated methods that model the data collection process offer the greatest potential to estimate e.g. timely trends (Isaac, 2014).

In this paper, we propose a statistical modelling framework that can be used to make inferences about animal populations with territorial

[*] Corresponding author.
*E-mail address:* santeri.j.karppinen@jyu.fi (S. Karppinen).

https://doi.org/10.1016/j.ecolmodel.2022.110101
Received 16 December 2021; Received in revised form 3 August 2022; Accepted 9 August 2022
Available online 29 August 2022

behaviour, using observations reported voluntarily by citizens. More specifically, our focus is on the following scenario:

- Citizens report presence-only observations of territorial animals. Each observation consists of a GPS coordinate and an approximate time stamp.
- We wish to estimate the number and locations of the animal territories within some area and time interval using the data collected by the citizens.
- Prior knowledge on the territorial behaviour of the species is assumed to exist in the form of a typical territory size and on the rate of appearance and disappearance of territories.

Because of the high spatio-temporal variability common in citizen science observation processes, modelling of the varying sampling effort, that is, the observation intensity, is a crucial first part of our framework. We take this variation into account by modelling the intensity based on past data. This yields intensity functions that capture the spatio-temporal variation in the observations reported by citizens. These functions are then fed into a data generating model consisting of two submodels that model the appearance and removal of territories, and the generation of citizen science observations from the territories, respectively. To estimate the number of territories and their locations, we use the latest available citizen science data and perform Bayesian inference for the data generating model.

The data generating model jointly approximates the evolution of the number of active territories and their locations in time, characterised by a sequence of posterior distributions conditioned on observation sets of increasing size. In the engineering literature, similar models are called 'tracking models' (cf. Goodman et al., 1997). Indeed, the inference algorithm we develop is a Rao-Blackwellised particle filter similar to those developed for tracking (Särkkä et al., 2007; Vihola, 2007). We further elaborate these methods by employing a state-of-the-art optimal resampling of Fearnhead and Clifford (2003), and further refine the inference algorithm so that it can incorporate the spatial inhomogeneity arising from our observation model.

Our data generating model is similar to dynamic occupancy models (Royle and Kéry, 2007) and open N-mixture models (Zhao et al., 2017) in the sense that it has a latent process model for the appearance and disappearance ("occupancy") of animal territories, and a variable observation intensity. However, unlike in the work of Zhao et al. (2017), the principal objects of analysis in our model are animal territories rather than individual animals. In addition, our model does not assume a fixed set of potentially occupied sites but operates in continuous space, where territories are delineated without a pre-defined grid. Finally, our model is formulated in continuous-time, which allows the estimation of the state of the population at any time points within the interval of interest. For example, our model can be used to track the state of the population at daily or weekly time steps. In contrast, the methods of Royle and Kéry (2007) and Zhao et al. (2017) operate in discrete time, and are typically used for annual data with a considerably smaller number of time steps.

The motivation for the development of our modelling framework has been to aid in the task of assessing the Finnish wolf (*Canis lupus*) population, although the framework can be relevant for other territorial species as well. Currently, the Finnish wolf population is assessed annually in March by the Natural Resources Institute Finland (Luke). In the assessments, wolf observations provided by citizens from the beginning of August to the end of February are combined with non-invasive genetic samples, tracks of GPS collared wolves and records of known mortality (Kojola et al., 2018). The assessments are carried out in two phases. In the first phase a panel of experts conducts a systematic review of all the data and judges territory boundaries that are potentially occupied by wolf packs or pairs in March. In the second phase, a Bayesian state–space model is used to infer the number of wolves living in each territory by combining wolf observations, DNA -recaptures and known mortality (Heikkinen et al., 2020). In particular, we envision that the developed framework can work as a useful tool in the first phase, providing a statistical look at the citizen science data and an aid in judging the territory boundaries.

We examine the performance of the developed particle filter with a sequence of simulation experiments, where we start from simple simulated conditions and work towards conditions that resemble more closely our concluding experiment, which is a realistic situation that could be faced in the assessment of the Finnish wolf population. Here, we use previous estimates of territory locations and citizen-provided observations to estimate the spatio-temporal variation of the conditional probability of wolf observations given known existence of wolf territories. Even though similar approaches have been used for species abundance estimations (e.g. Renner et al., 2015; Ver Hoef et al., 2021; Tang et al., 2021), the conditioning requirement provides a novel challenge. Using the results of said intensity modelling, we apply our data generating model to a real data set consisting of wolf observations made by Finnish citizens between April 2019 and March 2020. We estimate the number and locations of wolf territories and compare the result to the official estimates by the Natural Resources Institute Finland (Luke) which are based on the method discussed above.

The main contributions of this paper are as follows. First, we believe that the developed framework is of interest in assessing populations of territorial species using presence-only citizen science data. We focus on the application to wolves, but our methods are readily adaptable for other territorial species. Second, we believe that the observation intensity estimation is of its own independent interest, because it addresses the problem of estimating the conditional spatio-temporal intensity of presence-only citizen science observations. Third, from a methodological point of view, the developed data generating model and particle filter might be relevant also in the context of 'general purpose' target tracking (e.g Vihola, 2007; Särkkä et al., 2007) applications where a spatially varying observation process is needed.

## 2. Materials and methods

The general modelling framework proposed in this paper can be summarised into four successive analysis steps numbered from one to four. The flowchart in Fig. 1 depicts their dependencies and relation with each other, highlighting the inputs, outputs and datasets associated with each step. The following subsections will explain how we apply the framework in the context of wolf territory estimation. Section 2.1 discusses the Datasets A, B and C. Section 2.2 then describes the data generating model, which motivates the intensity estimation consisting of steps 1 and 2, which in turn are discussed in Sections 2.3 and 2.4, respectively. Section 2.5 describes step 3 of the analysis, the statistical inference based on the data generating model, using a particle filter we have developed for the problem. Finally, in Section 2.6 we conclude with a description of step 4 where we extract the number and locations of the territories from the output of the particle filter.

### 2.1. Data

In this Section, we will discuss the Datasets A, B and C seen in the framework of Fig. 1. In summary, the Datasets A and B contain past data used in the construction of the data generating model, and Dataset C contains the latest data to be processed by the particle filter. Each datum in Datasets A and C is a spatio-temporal point, that is, it has the form $(t, y)$, where $t$ is the time of observation, and $y$ is a two-dimensional point on a domain we denote by $D_y$. The difference between Datasets A and C is that Dataset A is past data, and Dataset C corresponds to the latest data we wish to infer the territories with. In contrast, Dataset B is more heterogeneous and contains all additional data such as covariates and expert knowledge required in the construction of the data generating model.

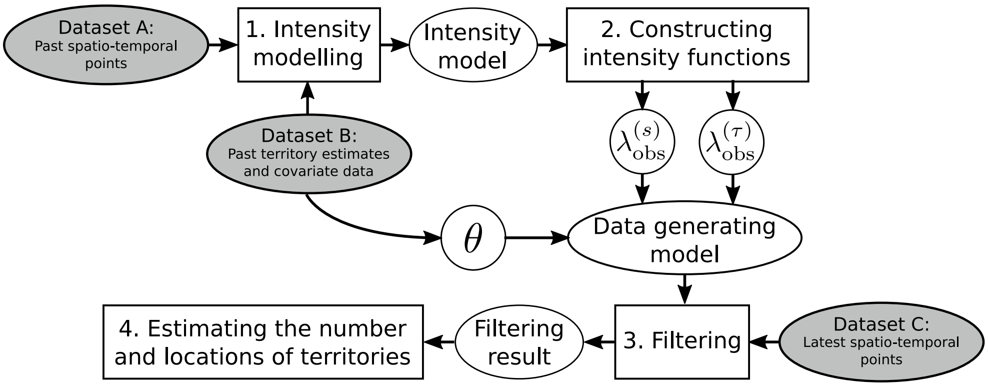

Dataset A:
Past spatio-temporal points
1. Intensity modelling
Intensity model
2. Constructing intensity functions
Dataset B:
Past territory estimates and covariate data
$\lambda_{\text{obs}}^{(s)}$
$\lambda_{\text{obs}}^{(\tau)}$
$\theta$
Data generating model
4. Estimating the number and locations of territories
Filtering result
3. Filtering
Dataset C:
Latest spatio-temporal points

**Fig. 1.** A flowchart of the proposed general modelling framework. The rectangles depict the analysis steps 1–4. The elliptical shapes denote inputs and outputs of the analysis steps. Input datasets are marked with the gray fill. The symbols $\lambda_{\text{obs}}^{(\tau)}$ and $\lambda_{\text{obs}}^{(s)}$ denote the intensity functions and $\theta$ stands for the other parameters of the data generating model.

### *2.1.1. Datasets A and C*

We extract the spatio-temporal points in Datasets A and C from a digital large carnivore observation database named "Tassu" (meaning a "paw" in Finnish) (Kojola et al., 2018). The observations enter the database through a network of approximately 2000 large carnivore contact persons (LCCPs), who are nominated by management associations and educated by the Finnish Wildlife Agency and Luke in the biology, ecology and movement behaviour of wolves as well as footprint identification. There are, however, no formal exams used in the nomination process.

The LCCPs have their own local trusted network of people who report their observations of wolves to the LCCP. These networks consist mostly of hunters that are proficient in identifying wolves based on sightings, tracks, prey kills and camera-trap documents. In addition, it is in principle possible for any citizen to report their observations, since the contact details of the LCCPs are publicly available and known in local rural societies (Pellikka and Hiedanpää, 2017). However, the networks of the LCCP are particularly relevant for wolf sightings in snow-free conditions because such observations usually cannot be verified afterwards.

Wolf observations found to be valid by the LCCPs are saved into the Tassu database. Each saved datum includes information about the time and location of the observation, the type of observation (such as wolf track, sighting, droppings, game camera photograph, prey kill site or livestock predation) and the estimated count of wolves observed simultaneously. The count estimate is based on the judgement of the LCCP based on the information available. Since the observations saved to the database are subject to the confirmation of the LCCP (possibly days after the initial report), we consider the observation times to be accurate on a daily granularity. In total, Datasets A and C contain all observations from the Tassu database that reported two or more wolves between January 2011 and March 2020. Since the purpose of our framework is to infer the number and locations of wolf territories, we only focus on observations that report more than one wolf, since this indicates that the observed wolves form a wolf pack and very likely exhibit territorial behaviour. In contrast, observations of single wolves can originate from lone, vagrant wolves, that do not yet maintain a territory. Furthermore, for simplicity, we make no distinction for data points with different observation types; we regard each observation simply as a spatio-temporal point. We return to this matter in the discussion.

We split the data such that Dataset C contains the observations made between April 1st 2019 and March 31st 2020, and Dataset A the observations before this. The locations in Dataset C are illustrated in Fig. 2 (top left). The domain of the locations, $D_y$, is mainland Finland south of the reindeer husbandry region in the north. The wolf territories in the reindeer husbandry region are few and short-term owing to lethal control that is justified by the prevention of damages to reindeer husbandry.

We organise the Datasets A and C according to so-called "wolf years". A wolf year starts April 1st and ends in March 31st of the next year. Hence, Dataset C consists of the observations made during the wolf year 2019–2020. The organisation of the data to wolf years has two reasons. First, as described in the introduction, the annual Finnish wolf population assessments describe the state of the wolf population in March. Second, the data indicate that the highest observation intensity is reached during the winter season and declines towards the spring. Year-by-year changes in the observation activity are expected to occur between the winter seasons, rather than between calendar years. We will also use the term "wolf month" to refer to the months within a wolf year such that the first wolf month corresponds to April, the second to May, and so on.

### *2.1.2. Dataset B*

Dataset B contains two kinds of information. Most importantly, Dataset B contains information about past known wolf territories until March 2019, that is, before the wolf year associated with Dataset C. In addition, Dataset B also contains covariates.

Dataset B is primarily used in the intensity modelling described in Section 2.3, but also for setting certain parameters in the data generating model. The details on how the data sources described below are used in the intensity modelling are given in Section 2.3. The relation of Dataset B to the parameters of the data generating model is discussed in the results of Section 3.3.

The information about past wolf territories was constructed from two sources of data, independently of Datasets A and C. We call the resulting territories 'auxiliary territories'. The first source consists of the space–time trajectories of 34 GPS-collared wolves that were tracked between 2011 and 2019. The transmitters in the collars stored the wolf's position at one- or four-hour intervals, depending on the season. The capture, handling and immobilisation protocols of these wolves are described in Kojola et al. (2016).

We assumed that each collared wolf was part of a wolf territory. The trajectories contain outlier recordings, such as test measurements at a lab or 'glitch' jumps of hundreds of kilometres occurring due to device malfunction or other reason. Some trajectories also cover two clearly separate territories. We therefore preprocessed the data as follows. First, the recorded GPS trajectories were divided into separate, contiguous trajectories at temporal jumps of more than two weeks or spatial jumps of more than 100 km. Trajectories less than 24 h were rejected. Second, each contiguous trajectory was processed by assigning to each trajectory point a probability of being an outlier. The probability was given by the velocity density $v \sim Exp(28.8)$ with median at 20 km/h, multiplied by the function $w(d_c = x) = 1(x <$

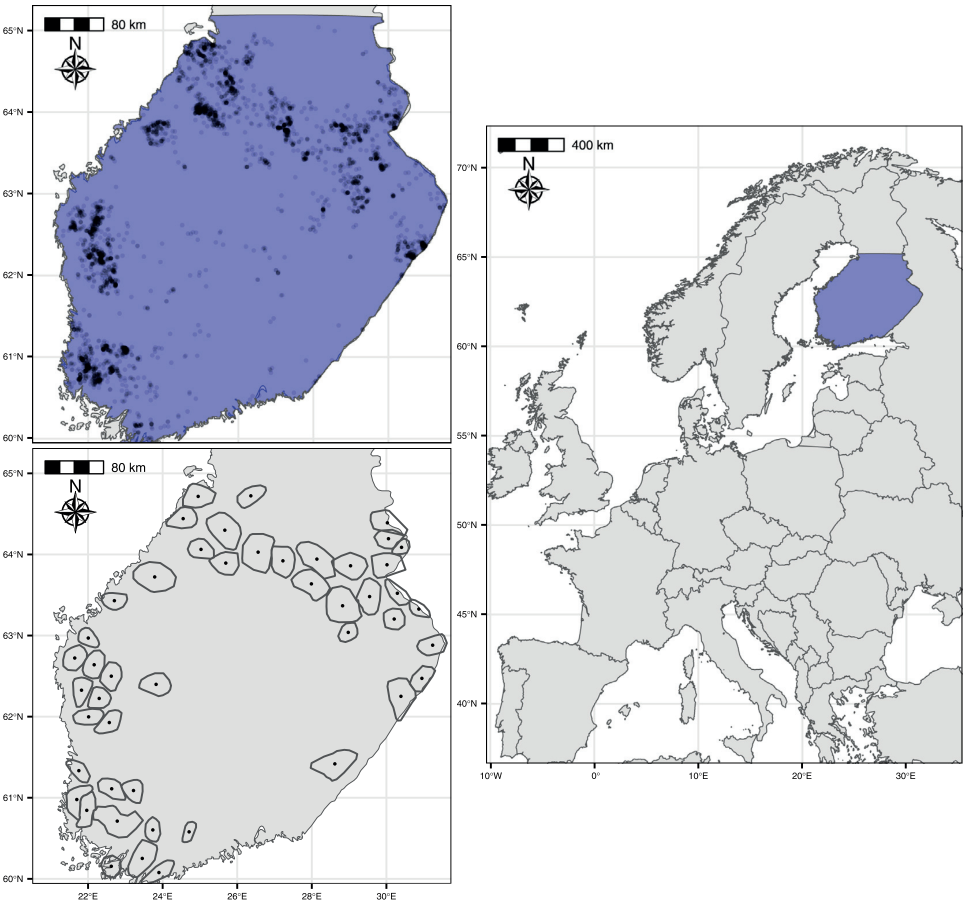


**Fig. 2.** Top left: The observed locations in Dataset C, that is, the locations of the Tassu observations (black points) that reported two or more wolves from April 2019 to March 2020. The blue overlay shows the domain of interest (the study area). Bottom left: Wolf territories found in the wolf population assessment of April 2019 by the Natural Resources Institute Finland. The point within each territory represents the centroid of the territory polygon. Right: The study area highlighted on a map of Europe. The distance scales are approximate due to coordinate transformations applied in drawing the maps.

$(5\sigma)) + 1(x > (5\sigma)) \exp[-0.5(x - 5\sigma)^2/40000^2]$, where $d_c$ denotes a point's distance from the trajectory's centre of mass and $\sigma$ is the 90%-truncated standard deviation of the centre of mass-distances. Points with probability less than 50% were excluded from the trajectory. Finally, the first and second steps were repeated to account for significant gaps after the outlier detection second step. From each remaining trajectory, an auxiliary territory was constructed as a polytope in $D_{\mathrm{y}} \times T_B$, where $T_B$ denotes the time span 2003-03-04–2019-03-31, by taking the Cartesian product of the convex hull of the spatial locations and the time interval of the trajectory. In total, 59 auxiliary territories were constructed from the GPS trajectories, covering approximately 74,000 km$^2$ and with time spans that add up to approximately 36.5 years.

The second type of auxiliary territories were constructed based on expert knowledge using the official population assessments of Luke from 2017 onwards. The assessments include estimates of active wolf pack territories as polygons in $D_y$, during March of the corresponding years. Fig. 2 (bottom left) shows the active pack territory location estimates of experts in the assessment of March 2019. We assumed that these territories were active also during January and February. We then constructed polytopes in $D_{\mathrm{y}} \times T_B$ as the Cartesian products of the polygons and January-March-intervals of each year between 2017–2019. The resulting 203 expert judgement auxiliary territories covered approximately 180,000 km$^2$ with time spans that add up to approximately 35.3 years.

The additional covariates in Dataset B consist of two datasets. The first of these is the CORINE land cover data for 2018 (Finnish Environment Institute SYKE, 2018). The dataset comes as a raster covering Finland and contains an approximate land use class (e.g. river, small road) for each of its 20 by 20 metre cells. The original 49 classes were first reclassed down to 8: Residential areas, other build areas, roads, cultivated fields, lakes and rivers, swamps and other wetlands, closed forests, and open forests. For each of the 8 classes, we aggregated their frequency in 1 km$^2$ cells, and to slightly reduce the amount of zeros, applied smoothing with a Gaussian blur with standard deviation 3 km. Each cell of the resulting 8-layer raster stack then contained a vector giving the (smoothed) frequencies of each land use class in (and near) the 1 km$^2$ cell. Since the resulting vector for each cell $k$, $[\mathrm{Corine}_{k,1}, \dots, \mathrm{Corine}_{k,8}]$, is (nearly) a simplex, we dropped the first class, residential areas, and kept the remaining 7 as frequencies. A log-ratio transformation, which is a popular approach in compositional data analysis, might have been more suitable here but was not done due to numerous zeros in all classes.

The CORINE road information capture larger streets and highways, and to describe accessible forest areas we computed an additional forest road frequency variable to Dataset B. This variable is derived from the national road and street database Digiroad (Finnish Transport Infrastructure Agency, 2021). From the database we extracted the polyline feature class '12', roads and paths traversable by offroad vehicle. We binarised the polylines on to the 20 by 20 metre cells of the CORINE land cover data, and then computed the frequencies of those cells on a $1 \times 1$ km raster. We denote the value of this variable in the $k$th 1 km$^2$ cell by $forestroad_k$.

### 2.2. Data generating model

In this section, we discuss the data generating model we have developed for citizen science observations of a territorial species. The model we have developed is more general than the instance of it that we use for modelling the wolf data. Therefore, this section will also highlight certain modelling decisions we make in the present application. Furthermore, the data generating model we describe here is 'ideal' in the sense that it must be approximated further to be tractable for our inference method. We will discuss this in more detail in Section 2.5 that is devoted to the filtering algorithm. For the interested reader, the mathematical details of the general data generating model are given in Sections 1.1 and 1.2 of the supplementary material.

The data generating model consists of two submodels, the birth and death process and the observation model, which we will discuss in Sections 2.2.1 and 2.2.2, respectively. In summary, the birth and death process models how new territories emerge and disappear, and the observation model describes how each existing territory produces citizen science observations (spatio-temporal points as in Dataset C).

#### 2.2.1. Birth and death process

Our model assumes that the territories of interest exist and emerge within a domain denoted by $D_\mu \subset \mathbb{R}^2$, with $D_\mu \subseteq D_\mathrm{y}$. The location of the territory $i$ is represented by its centroid, $\mu_i \in D_\mu$, which is assumed to be constant in time. New territories emerge within $D_\mu$ with the instantaneous birth intensity $\lambda_\mathrm{b}(u)N_u + \lambda_{\mathrm{b}0}$ where $\lambda_\mathrm{b}(u)$ is the (known) birth intensity function and $N_u$ stands for the number of existing territories at time $u$. The function $\lambda_\mathrm{b}(u)$ can be interpreted as birth intensity per each existing territory. The baseline birth intensity parameter $\lambda_{\mathrm{b}0}$, on the other hand, models additional birth intensity due to external factors such as inflow from outside $D_\mu$. For modelling of the wolf data, we simplify $\lambda_\mathrm{b}(u)$ to a constant, denoted by $\lambda_\mathrm{b}$, and set $\lambda_{\mathrm{b}0} = 0$. As a new territory emerges, its centroid follows the uniform distribution on $D_\mu$.

Similarly, each existing territory disappears with the instantaneous death intensity $\lambda_\mathrm{d}(u)$, that is, the total instantaneous death intensity induced by all territories equals $\lambda_\mathrm{d}(u)N_u$. In case of the wolf data, we fix $\lambda_\mathrm{d}(u)$ to a constant that we denote by $\lambda_\mathrm{d}$. Therefore, the lifetime of a single territory follows an exponential distribution with mean $\lambda_\mathrm{d}^{-1}$. Furthermore, in a similar fashion as was done in Vihola (2007), we 'symmetrise' the birth and death process by setting $\lambda_\mathrm{b} = \lambda_\mathrm{d} = \lambda_\mathrm{bd}$, where $\lambda_\mathrm{bd}$ then remains the only birth/death intensity parameter. This minimises the bias in the birth and death process, and a priori leads to a constant conditional expectation for the number of territories in time.

The initial distribution of the model is a joint distribution of the number of territories and the locations of their centroids. The initial locations of the territory centroids can either be distributed uniformly on $D_\mu$ or subject to Gaussian error (truncated to $D_\mu$) around some location estimate.

#### 2.2.2. Observation model

The observation model, conditional on the territory locations and lifetimes generated by the birth and death process, describes how each territory with its centroid on $D_\mu$ produces citizen science observations.

The 'baseline', underlying model for an observation from a single territory $i$ that exists at any given time point, is bivariate normal $N(\cdot;\mu_i,\Sigma_\mathrm{obs})$, where $\Sigma_\mathrm{obs}$ describes the size and shape of the territories. We assume that the territories are roughly circular in shape by setting $\Sigma_\mathrm{obs} = \sigma^2_\mathrm{obs} I$, where $I$ denotes the $2 \times 2$ identity matrix and $\sigma_\mathrm{obs} > 0$ is a standard deviation related to the territory size.

It is useful to interpret this territory model using the circular contours of the distribution $N(\mu_i, \sigma^2_\mathrm{obs} I)$. A circle of radius

$$\sqrt{\chi^2_{\alpha,2}}\sigma_\mathrm{obs} \tag{1}$$

centred at $\mu_i$ is assumed to enclose the instantaneous location of the wolves belonging to the territory with probability $\alpha$. Here, $\chi^2_{\alpha,2}$ corresponds to the $100\alpha\%$ quantile of the chi-squared distribution with two degrees of freedom.

Because of the temporal and spatial variability inherent to citizen science observation processes, the observation model modulates the number of observations produced from the territories based on a temporal intensity function $\lambda^{(\tau)}_\mathrm{obs}$, defined on a time interval of interest $[0,T)$, and a spatial intensity function $\lambda^{(s)}_\mathrm{obs}$ defined on the domain of the observed locations, $D_\mathrm{y}$. These intensity functions are assumed spatio-temporally separable, since there is limited data for their estimation, which is further discussed in Section 2.3. The values of these functions are tied to the number of observations the territories produce in time and space. For constant functions $\lambda^{(s)}_\mathrm{obs}(y) = l_x \in [0,\infty)$ and $\lambda^{(\tau)}_\mathrm{obs}(u) = l_t \in [0,\infty)$, our model assumes that the expected number of observations that a single territory produces on $D_\mathrm{y}$ in a unit of time is approximately $\lambda_\mathrm{obs} l_x l_t$, where $\lambda_\mathrm{obs}$ is a scalar multiplier for the intensity functions. The intensity functions $\lambda^{(\tau)}_\mathrm{obs}$ and $\lambda^{(s)}_\mathrm{obs}$ are assumed known (fixed), and we discuss their estimation in Sections 2.3–2.4.

In addition to the observations originating from the territories, the observation model also accommodates so called 'clutter' observations, that are understood as 'erroneous' observations not originating from actual territories. These observations are assumed to be distributed uniformly on $D_\mathrm{y}$ and their intensity is likewise modulated by $\lambda^{(\tau)}_\mathrm{obs}$ and $\lambda^{(s)}_\mathrm{obs}$, but multiplied by a different scalar parameter, $\lambda_\mathrm{c}$. The relative values of the scalar multipliers $\lambda_\mathrm{obs}$ and $\lambda_\mathrm{c}$ can be used to model the rate of the total number of observations believed to originate from the territories.

Mathematically, conditional on the territory locations and lifetimes, our observation model defines a three-dimensional inhomogeneous Poisson process in time and space, whose intensity function is given in Equation (5) of the supplementary material. The data generating model parameters in Fig. 1 are given by $\theta = (\lambda_\mathrm{obs}, \lambda_\mathrm{bd}, \lambda_\mathrm{c}, \sigma_\mathrm{obs}, D_\mathrm{y}, D_\mu)$ and the intensity functions $\lambda^{(\tau)}_\mathrm{obs}$ and $\lambda^{(s)}_\mathrm{obs}$.

### 2.3. Observation intensity modelling

The following two sections discuss how we estimate the intensity functions $\lambda^{(\tau)}_\mathrm{obs}$ and $\lambda^{(s)}_\mathrm{obs}$ in the observation model of Section 2.2.2. This section focuses on step 1 of the modelling framework in Fig. 1, detailing the intensity model we fit to the past Datasets A and B discussed in Section 2.1. The primary data for this step are the spatio-temporal points in Dataset A discussed in Section 2.1.1. We assume that each of these observations originated from an active wolf territory. In this section, we denote by $\psi = \{[s_i;t_i]\}$ the spatio-temporal point pattern of the wolf observations in Dataset A with locations $s_i \in D_\mathrm{y}$ and dates $t_i \in T_A$ =[2011-01-01, 2019-03-31].

We assume the arrival of Tassu reports $\psi$ can be approximated by an inhomogeneous Poisson process (Illian et al., 2008). Note that reporting depends on two consecutive events: An observer is at a territory, and they make and report an observation. The data contains no information if an observer was on a territory but did not observe wolf activity. The observer and reporting intensities are therefore confounded. The situation is notably different from presence–absence citizen science data, such as for birding analysed with point processes by Tang et al. (2021), as in addition to absences not being measured, the observers cannot be assumed to have been actively looking for wolf activity in the first place. Our situation is more akin to presence-only analysis (Renner et al., 2015; Ver Hoef et al., 2021), with the nuance that instead of estimating species abundances the goal is to estimate the connection between a wolf territory's presence and the emergence of the Tassu-reports. To account for the conditioning on a wolf territory's presence in this estimation, we constrain the Tassu-observations to the auxiliary territories in Dataset B, detailed in Section 2.1.2.

Given the observations on the auxiliary territories, we estimate the intensities by aggregating the observations and then using standard Poisson regression in the generalised additive models (GAM) framework. More specifically, we fit the model on observation units given by spatio-temporal grid cells $C_k = \{V_k \times t_k\}$ with temporal resolution $|t_k| = 1$ month and spatial resolution $|V_k| = 1$ km $\times$ 1 km. From this discretisation of $D_y \times T_A$ we only consider the subset of cells that intersect the auxiliary territories, which is $\approx 11.9$ million cells, and then count the Tassu reports in each such cell, denoted by $n_k = \#(\psi \cap C_k)$, resulting in 4816 non-empty cells. The auxiliary territory (polytope) which a cell $C_k$ overlaps is denoted by $A_k$. For each cell $C_k$ we let $u_k$ and $y_k$ represent its centroid in time and space, respectively, and finally define $\lambda_k^{(\tau)} := \int_{t_k} \lambda_{\text{obs}}^{(\tau)}(u)du$ and $\lambda_k^{(s)} := \int_{V_k} \lambda_{\text{obs}}^{(s)}(y)dy$.

Then the counts in the grid cells are modelled with a Poisson regression model of the following form

$$
\begin{aligned}
n_k &\sim \text{Poisson}\left(\frac{\lambda_k^{(\tau)}\lambda_k^{(s)}}{|A_k|}\right) \\
\log(\lambda_k^{(\tau)}\lambda_k^{(s)}) &= \text{month}(u_k) + \text{year}(u_k) + X(y_k)\beta + \text{smooth}(y_k) \\
X(y_k) &= [1_{\text{expert}}(A_k), \text{Corine}_k, \text{forestroad}_k]^T.
\end{aligned}
\tag{2}
$$

The offset $|A_k|$ is the area of the territory that grid cell $C_k$ belongs to, and accounts for an assumption of the instantaneous location of the wolves following a uniform distribution within $A_k$, i.e. a pack on a larger territory is harder to observe.

The 'year' and 'month' effect were included as factors to model seasonal effects and year to year differences. We used 'wolf years' as discussed in Section 2.1.1 for the factors 'year' and 'month'. We did not include weather station information (e.g. snow depth), mainly because their monthly aggregates are highly correlated with month-effects but also in order to avoid spatio-temporal interaction terms to reduce model complexity given we have only <1% non-zero units.

The spatial effects are modelled with covariates $X$ and a residual smooth term. The indicator $1_{\text{expert}}(A_k)$ was included to adjust for potential discrepancies between the different types of auxiliary territories (GPS tracks & expert estimates). The numerical covariates $\text{Corine}_k = [\text{Corine}_{k,2}, \dots, \text{Corine}_{k,8}]$ and $\text{forestroad}_k$ were described in Section 2.1.2 and capture environmental variability.

The estimation was carried out using the statistical software R and the GAM function `mgcv::bam` with a smooth term 'smooth($y_k$)' defined as a tensor product `te`-term in $x$ and $y$ coordinates of the cell centroid $y_k$. The smoothness penalty choice was left to the default which is generalised cross validation (Wood, 2017).

### 2.4. *Computing intensity functions for the data generating model*

Next, we discuss how we compute the intensity functions $\lambda_{\text{obs}}^{(\tau)}$ and $\lambda_{\text{obs}}^{(s)}$ based on the intensity model of Section 2.3 fit to the past Datasets A and B. This section focuses on step 2 in the modelling framework of Fig. 1. The aim is to obtain intensity functions for the data generating model that anticipate the spatio-temporal intensity of the observations in Dataset C. We model the intensity functions as piecewise constant such that $\lambda_{\text{obs}}^{(\tau)}$ takes on the value $c_{t_i}$ during wolf month $i$, $i = 1, \dots, 12$, and $\lambda_{\text{obs}}^{(s)}$ takes on the value $c_{V_j}$ in each cell $V_j \in D_{\text{y}}$. In summary, the $c_{t_i}$'s and $c_{V_j}$'s are computed by using quantities calculated from the predictions of the intensity model (2) in Eq. (4) below.

The computation proceeds as follows. First, assume that the intensity model (2) is fit using the Datasets A and B. Then, using the fitted model, we predict the spatial effect $\lambda_k^{(s)}$, excluding the term $1_{\text{expert}}(A_k)$ for all 1 km$^2$ grid cells $V_k \in D_{\text{y}}$. This grid is then smoothed using Gaussian blur, and we denote the value in the smoothed grid cell $V_k$ by $\tilde{\lambda}_k^{(s)}$. More specifically, we use a 'border preserving' Gaussian blur that only smooths cells that are within $D_y$ and normalises the blur weights in the smoothing window such that only non-zero intensity values contribute to the smoothed grid. We use $\sigma_{\text{obs}}$ as the standard deviation in the Gaussian blur and set the window size to the first integer larger than $2\sigma_{\text{obs}}$ (in kilometres). We report the $\sigma_{\text{obs}}$ value used in this step together with the results of Section 3. The smoothing of the predicted grid is motivated by an assumption of smoothness that our inference method places on the spatial intensity function. We will discuss this (Assumption C) in more detail in Section 2.5.

After computing the predicted and smoothed spatial effect, we also predict the temporal effect for the wolf year associated with Dataset C by setting

$$\tilde{\lambda}_i^{(\tau)} = \exp(\beta_{\text{year}}^* + \beta_{\text{month,i}}),$$

where $\tilde{\lambda}_i^{(\tau)}$ is the predicted temporal effect for wolf month $i = 1, 2, \dots, 12$ during the wolf year of interest. Here, $\beta_{\text{year}}^*$ corresponds to a predicted (wolf) year regression coefficient obtained by running a linear regression on the previous (wolf) years' regression coefficients (available from fitting model (2)). The coefficients $\beta_{\text{month,i}}$, on the other hand, correspond to the estimated (wolf) month coefficients from model (2). The linear regression for the year coefficients was carried out to take into account a slight increasing trend in the yearly regression coefficients of model (2).

To compute the distinct values $c_{t_i}$ and $c_{V_j}$ that the piecewise constant functions $\lambda_{\text{obs}}^{(\tau)}$ and $\lambda_{\text{obs}}^{(s)}$ take, we match the predicted intensities such that

$$\frac{\tilde{\lambda}_i^{(\tau)}\tilde{\lambda}_j^{(s)}}{|A_j|} = \int_{t_i}\int_{V_j} \lambda_{\text{obs}}^{(\tau)}(u)\lambda_{\text{obs}}^{(s)}(y)/|A_j|\mathrm{d}y\mathrm{d}u, \tag{3}$$

where $A_j$ is defined as in Section 2.3.

Evaluating the integral (3) and taking the logarithm yields

$$\log(\tilde{\lambda}_i^{(\tau)}) + \log(\tilde{\lambda}_j^{(s)}) = \log(|t_i|) + \log(|V_j|) + \log(c_{t_i}) + \log(c_{V_j}).$$

To obtain equality in this equation, $c_{t_i}$ and $c_{V_j}$ can be chosen such that

$$
\begin{aligned}
c_{t_i} &= \exp(\log(|t_i|^{-1}) + \log(\tilde{\lambda}_i^{(\tau)}) - K) \\
c_{V_j} &= \exp(\log(|V_j|^{-1}) + \log(\tilde{\lambda}_j^{(s)}) + K),
\end{aligned}
\tag{4}
$$

where $K$ is a constant that needs to be chosen. Our choice for $K$ is

$$K = -\max_j\{\log(|V_j|^{-1}) + \log(\tilde{\lambda}_j^{(s)})\},$$

which scales $\lambda_{\text{obs}}^{(s)} \in (0, 1]$.

Finally, for the purposes of Sections 3.3 and 3.4 we note that when $\lambda_{\text{obs}}^{(s)}$ is computed as discussed above, one of the territory centroids seen in Fig. 2 is outside the domain of $\lambda_{\text{obs}}^{(s)}$. Therefore, in the case of the wolf territory estimation, we additionally widen the domain of $\lambda_{\text{obs}}^{(s)}$ by 27 kilometres at the boundaries by repeating the maximal intensity value found in the neighbouring cells of the borders.

### 2.5. *Filtering algorithm and its constraints*

This section discusses step 3 of the framework in Fig. 1, the statistical inference of the territories given Dataset C, assuming the data generating model of Section 2.2. This section gives an overview of the inference, and the mathematical details are given in Section 1.6 of the supplementary material. In summary, given Dataset C and the data generating model of Section 2.2 with fixed parameters and intensity functions, the inference procedure outputs a sequence of joint filtering distributions of the territory centroid locations and their number at chosen times $t_1 < t_2 < \dots < t_n$ that are also input to the procedure. The filtering distribution at time $t$ is the distribution of the locations of the territory centroids and their number, conditional on the observations of Dataset C up to time $t$.

The inference method we use is somewhat involved and computationally intensive, and based on sequential Monte Carlo/particle filtering (cf. Doucet et al., 2000). More specifically, the method is a Rao-Blackwellised particle filter similar to the works of Särkkä et al. (2007), Vihola (2007), but employing a state-of-the-art optimal resampling algorithm developed by Fearnhead and Clifford (2003).

A summary of the method's operation can be given as follows. Denote by $\tilde{y}_{1:K} = (\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_K)$ the $K$ temporally ordered observations with observation times $\tilde{t}_1 < \tilde{t}_2 < \cdots < \tilde{t}_K$. Here, the input timepoints $t_1 < t_2 < \cdots < t_n$ are among the $\tilde{t}_i$'s and each $\tilde{y}_i$ is a spatial location from Dataset C, or $\tilde{y}_i = \emptyset$ (see also discussion on Assumption A below). The method works by processing the observations $\tilde{y}_{1:K}$ sequentially, such that the processing of $\tilde{y}_k$ yields the filtering distribution at time $\tilde{t}_k$. During the algorithm, each filtering distribution is characterised by a set of $M$ weighted particles. More specifically, at time index $k-1$, each of the $M$ particles represents a hypothesis about the locations and number of territory centroids on $D_\mu$, conditional on the observations $\tilde{y}_{1:k-1}$ seen so far (with $\tilde{y}_{1:0}$ understood as the empty set).

The observation $\tilde{y}_k$ is processed such that first a set of possible territory birth, territory death and observation association outcomes is built, which consist of all one step 'futures' that could happen for any existing hypothesis at time index $k-1$, conditional on $\tilde{y}_k$ and the time passed since $\tilde{y}_{k-1}$. This set of outcomes is then probabilistically pruned using the optimal resampling algorithm of Fearnhead and Clifford (2003), which yields a set of $M$ chosen outcomes and their normalised weights. Finally, based on the chosen outcomes and the previous hypotheses, a new set of $M$ updated and weighted hypotheses (particles) is constructed by adding and deleting territories and 'associating' $\tilde{y}_k$ (if $\tilde{y}_k \neq \emptyset$) to a territory using an approximate Kalman filter update. The filtering distribution at time index $k$ is characterised by these $M$ particles. The process then repeats for the observation $\tilde{y}_{k+1}$.

The ideal data generating model of Section 2.2 is time-discretised and approximated before the filter can be used. The approximations used can be justified by introducing a set of additional assumptions, some of which are primarily computational, and some of which help reduce the discrepancy between the approximate and the ideal model. The following list highlights these assumptions, which we will refer to as Assumptions A to D.

(A) The observed data can be processed sequentially, one at a time. In other words, each datum has an associated time, and the observation times are strictly increasing.

(B) Compared to the rate of observations arriving from the territories, birth and death events of territories are rare.

(C) The spatial intensity function $\lambda^{(s)}_{\text{obs}}$ is 'smooth'/'slowly varying' with respect to the territory size parameter $\Sigma_{\text{obs}}$. This means that for any centroid $\mu \in D_\mu$, $\lambda^{(s)}_{\text{obs}}(\mu)$ is a good approximation for $\lambda^{(s)}_{\text{obs}}(x)$ in the region where the distribution $N(\mu, \Sigma_{\text{obs}})$ has most of its probability mass.

(D) For most territories, the territory centroid $\mu_i$ is not close to the boundary of $D_\mu$, in the sense that a region of high probability of $N(\mu_i, \Sigma_{\text{obs}})$ is contained within $D_\mu$.

We conclude this section with a brief discussion on these assumptions. Assumption A is satisfied for many datasets that are collected in real time. However, as mentioned in Section 2.1, the observation times in Dataset C are pooled with a granularity of one day. In order to make Assumption A hold, we introduce a preprocessing step before the filtering that artificially disperses the daily pooled observations in time, generating a 'pseudotime' for each observation within the day that it occurred. This step introduces a bias, which is small, since the arrival intensity of the observations still remains practically the same as with the pooled data. The preprocessing of the data is related to the time discretisation of the model, which we make fine enough so that during filtering we may assume that practically at most one territory birth or death may occur during each time-discretised interval, and that the time-discretised model approximates the ideal model sufficiently well. We ensure this by introducing 'discretisation points' to the dataset that contain no spatial location (that is, $\tilde{y}_k = \emptyset$). For further discussion on these matters, see Sections 1.3–1.4 in the supplementary material.

Assumption B is necessary for the identification of the territories based on the data. In the present application, Assumption B holds since the births and deaths of wolf territories are relatively rare events on the daily timescale at which the observations in Dataset C arrive.

Assumption C is necessary for approximating certain intractable integrals that arise in the filtering algorithm and are related to the spatial intensity function $\lambda^{(s)}_{\text{obs}}$. In Section 2.4 we described a smoothing step for the predicted spatial grid, which was carried out in order to satisfy Assumption C.

Finally, Assumption D arises since our method does not involve explicit edge correction. The computations in the particle filter are approximate for territories and observations close to the boundary of the finite domain $D_\mu$. This may entail some bias, which is small under Assumption D. We investigate empirically the bias caused by the edge effect in Section 3.2.

### 2.6. Extracting the number and locations of the territories from the output of the particle filter

This section focuses on the final step of the framework of Fig. 1 and describes how we extract the number and locations of the territories from the filtering result. Our estimate for the number of territories at time $t$ is the marginal filtering distribution of the number of territories at time $t$, computed as follows. Denoting by $n_i$, $i = 1, 2, \ldots, p$ the unique numbers of territories found among the $M$ particles at time $t$, the marginal filtering distribution for the number of territories consists of the tuples $(n_1, \pi_1), (n_2, \pi_2), \ldots, (n_p, \pi_p)$, where $\pi_i$ is the sum of the normalised weights of the particles having exactly $n_i$ territories. In Section 3.4, we will summarise these distributions by taking their mean, mode and standard deviation, and by computing probability intervals $I_\alpha$, that is, intervals whose end points are given by the $(100-\alpha)$% and $\alpha$% quantiles of the distribution, where $\alpha$ is a given percentage point.

A common way to visualise the probabilistic location information of an unknown number of objects is to plot the so-called probability hypothesis density (Goodman et al., 1997) (PHD) of the filtering distribution, which in the present context corresponds to the expected intensity of territory centroids. More specifically, the PHD is defined for the territory centroids at time $t$ by

$$\mathrm{PHD}(\mu) = \sum_{j=1}^{M} w^{(j)} \sum_{i \in I_t^{(j)}} f_{ji}(\mu).$$

Here, $w^{(j)}$ is the $j$th normalised particle weight at time $t$, $I_t^{(j)}$ enumerates the territories in particle $j$ at time $t$, and $f_{ji}$ is the $i$th density in particle $j$ at time $t$. The densities $f_{ji}$, $i \in I_t^{(j)}$, each represent the knowledge about a particular territory centroid location within one of the particles (hypotheses). In the context of our model, these densities are either normal densities $N(\mu; m_{ji}, C_{ji})$, with known means $m_{ji}$ and covariances $C_{ji}$ computed by our particle filter, or uniform densities $U(\mu; D_\mu)$. The form of the density, uniform or normal, depends on whether an observation has been 'associated' with a particular territory centroid in the particle. For more details regarding this, see Section 1 of the supplementary material.

Our approach for visualising the territory locations differs slightly from 'standard' PHD, and is as follows. First, we compute the PHD, but with the $f_{ji}$'s 'at the observation level', meaning that for territories associated at least once, $f_{ji}$ corresponds to the normal density $N(y; m_{ji}, C_{ji} + \Sigma_{\text{obs}})$. For territories never associated, we take $f_{ji}(y) = U(y; D_\mu)$. After computing the PHD in this manner, we furthermore truncate the PHD values from above to the density value $N(0; 0, \Sigma_{\text{obs}})$. This value corresponds to the maximal contribution to the PHD value from a single territory which is known to exist and whose location has been estimated with maximal precision. In Section 3.4 we will visualise the estimated territory locations on the map using this computation. The rationale for this procedure is clearer visualisation of the regions where the filtering algorithm places the territories.

In Section 3 we focus mostly on the estimation of the number of territories, because it is relevant from the point of view of assessing the reproductive capacity of a wolf population. The number of territories is also easy to work with from a model validation perspective, since it is straightforward to compare it numerically to 'ground truths' of simulation experiments and to results of other estimation methods. In contrast, the PHD is important for graphical validation, visualisation and interpretation of the filtering result.

## 3. Results

We first discuss results regarding the observation intensity modelling in Section 3.1. We then move on to the experiments with the data generating model and the developed particle filter, starting with a synthetic scenario in Section 3.2, and then moving on to a semisynthetic scenario in Section 3.3 that resembles the situation with Dataset C but is still based on simulated observations. We conclude with the real data scenario based on Dataset C in Section 3.4.

### 3.1. Tassu observation arrival intensity estimation

Fig. 3 displays the estimated intensity functions $\lambda^{(\tau)}_{\text{obs}}$ and $\lambda^{(x)}_{\text{obs}}$ as well as the predicted spatial effect of the intensity model (2). Based on the figure, the temporal intensity function $\lambda^{(\tau)}_{\text{obs}}$ is seen to capture the rise in observation intensity in the winter time, and the spatial intensity function $\lambda^{(s)}_{\text{obs}}$ especially accounts for the high observation intensities in western Finland.

The overdispersion of the Poisson regression modelling of the observation intensities (Eq. (2)) fit was 1.83. After accounting for the overdispersion the yearly effects were not statistically significant (at 5% level), but monthly effects were clear: The fluctuations with respect to the baseline month of April ranged from a 78% (95% confidence interval [54, 91]) reduction in May to 371% ([206, 592]) increase in November, with smooth transitions in between.

The CORINE landcover variable effects were mostly increasing. When considering a 1% increase in the proportion of each cover class in turn, the estimated increase in intensity was: Cultivated fields 19% ([9, 29]); Closed forests 18% ([8, 28]); Open forests 16% ([7, 27]); Rivers and lakes 16% ([6, 26]); and Other wetlands 19% ([10, 30]). Effect of larger roads was not significant, but a 1% increase in forest roads increased the intensity by 5% ([1, 8]). The smooth component was significant, with a clear reduction effect in the central region and an increase in the west, south-west and south regions.

The explained deviance was 12.6%. For diagnostics we first checked the Pearson residuals aggregated at a month resolution, dropping the spatial dimension (see Figure 4 in the supplementary material). The cell counts showed slightly higher proportion of 0's than the model predictions, otherwise the overall quantiles were reasonably matched. There were no obvious patterns in time, apart from a potential positive trend during 2015. The residual variability was the same during 2011–2016 with only GPS tracking auxiliary territories and during 2017–2019 when both auxiliary territory-types were available. Pearson residuals exceeded +-2 during five months (2014-10, 2017-12, 2018-01, 2018-09, 2019-02) with no clear pattern, with three over-estimates (predicted v observed counts: 39 v 14, 257 v 175, 543 v 475), and two under-estimates (6 v 21, 9 v 21).

We then studied the Pearson residuals in space without the time dimension. To visually check troubling areas we aggregated the observed and predicted counts to 10 × 10 km cells. Observed counts had again slightly larger amount of 0's, and also some higher-than-expected values, the latter mostly from the 2017–2019 period. No obvious spatial structure was visible in the Pearson residual map, with large residuals dotted around the domain (see Figure 4 in the supplementary material). We checked a version where the largest count in the temporal sum per cell was omitted (before aggregation in space). The residual sizes were greatly reduced (max.abs. from 12 to 3). This sensitivity suggests that the observation counts are more concentrated in time and space than what we can capture with the model. Additionally, spatially contiguous regions of underestimation, particularly on the west coast, were revealed, indicating insufficient information in the spatial components of the model. A further check of before and after 2017 spatial sums revealed a tight cluster of unexpectedly high observation counts in the border region of eastern Kainuu.

### 3.2. Synthetic scenario and the edge effect

Our first territory estimation experiment is a purely synthetic, simple scenario. The purpose of the experiment is to ensure that the estimation algorithm works correctly. We also investigate explicitly the bias caused by the edge effect, as discussed in Section 2.5.

In this experiment we skip the intensity modelling discussed in Sections 2.3–2.4 and focus on the filtering of simulated datasets. We define the data generating model such that $D_\mu = [0, 100] \times [0, 100]$, $\lambda^{(\tau)}_{\text{obs}}(u) = 1$ and $\lambda^{(s)}_{\text{obs}}(y) = 1$ for all $u \in [0, 50]$ and $y \in D_{\text{y}} = \mathbb{R}^2$. For the remaining parameters we set $\lambda_{\text{bd}} = 0.0015$, $\lambda_{\text{c}} = 0$, $\lambda_{\text{obs}} = 1$. This configuration corresponds to a simple scenario for our particle filter, since there is no spatial inhomogeneity, and the largest approximation in the filtering arises from the finite domain.

Under these settings, for all combinations of the number of particles $M \in \{128, 256, 512, 1024, 2048\}$ and $\sigma_{\text{obs}} \in \{1, 2, 5, 10, 15\}$ we simulated 450 datasets as follows. First, we simulated the territory locations from the ideal birth and death model, and then conditional on the territories, simulated the observations from the ideal observation model, each time preprocessing the observations with the method discussed in Section 2.5. The initial distribution for the number of territories in the birth and death process was $\text{Poisson}(20)$ truncated to the interval $[10, 30]$. For each sampled initial territory, we set the uniform distribution on $D_\mu$ as the initial distribution for the centroid of the territory.

We filtered each simulated dataset assuming that the initial distribution above was known, and computed the deviations $\hat{N}_t - N_t$, for $t = 1, \ldots, 50$, where $\hat{N}_t$ is the estimated mean number of territories at time $t$ from the particle filter, and where $N_t$ is the true number of territories for the dataset. We investigated the bias $\mathbb{E}[\hat{N}_t - N_t]$, which would be zero for an ideal Bayes estimator, by computing the empirical mean of the 450 deviations per $t$, $M$ and $\sigma_{\text{obs}}$. Fig. 4 summarises the results of this experiment. With a small territory size compared to the size of the domain $D_\mu$, we observe little or no bias in the estimation of the number of territories, given that a sufficient $M$ is used in the filtering algorithm. For greater values of the territory size, the bias is more significant and appears to only slightly diminish with increasing $M$. This is expected, since with higher territory sizes, Assumption D of Section 2.5 is more likely to be violated, leading to observations outside or near the boundary of $D_\mu$.

### 3.3. Semisynthetic scenario and feasibility of territory estimation

Next, we consider a more realistic 'semisynthetic' scenario with closer resemblance to the situation with Dataset C. In this scenario, the idea was to fix the territory centroids to realistic locations, use plausible parameter values and the intensity functions estimated as discussed in Sections 2.3–2.4. We then simulated data to see how well the particle filter can recover the true number of territories under a more realistic setting.

More specifically, we assumed that $D_\mu$ corresponds to the domain of the Tassu data (Fig. 2 top left), and the territory centroid locations were fixed to the centroids (Fig. 2 bottom left) of the territories found by Luke in the wolf population assessment of 2019. We further assumed that each of these 47 territory centroids existed for a period of one year, and that the territory count did not change.

We used the intensity functions in Figs. 3(c) and 3(b) as $\lambda^{(\tau)}_{\text{obs}}$ and $\lambda^{(s)}_{\text{obs}}$ in the data generating model. To set the value for the territory size parameter $\Sigma_{\text{obs}} = \sigma^2_{\text{obs}} I$, we examined the shapes and sizes of the

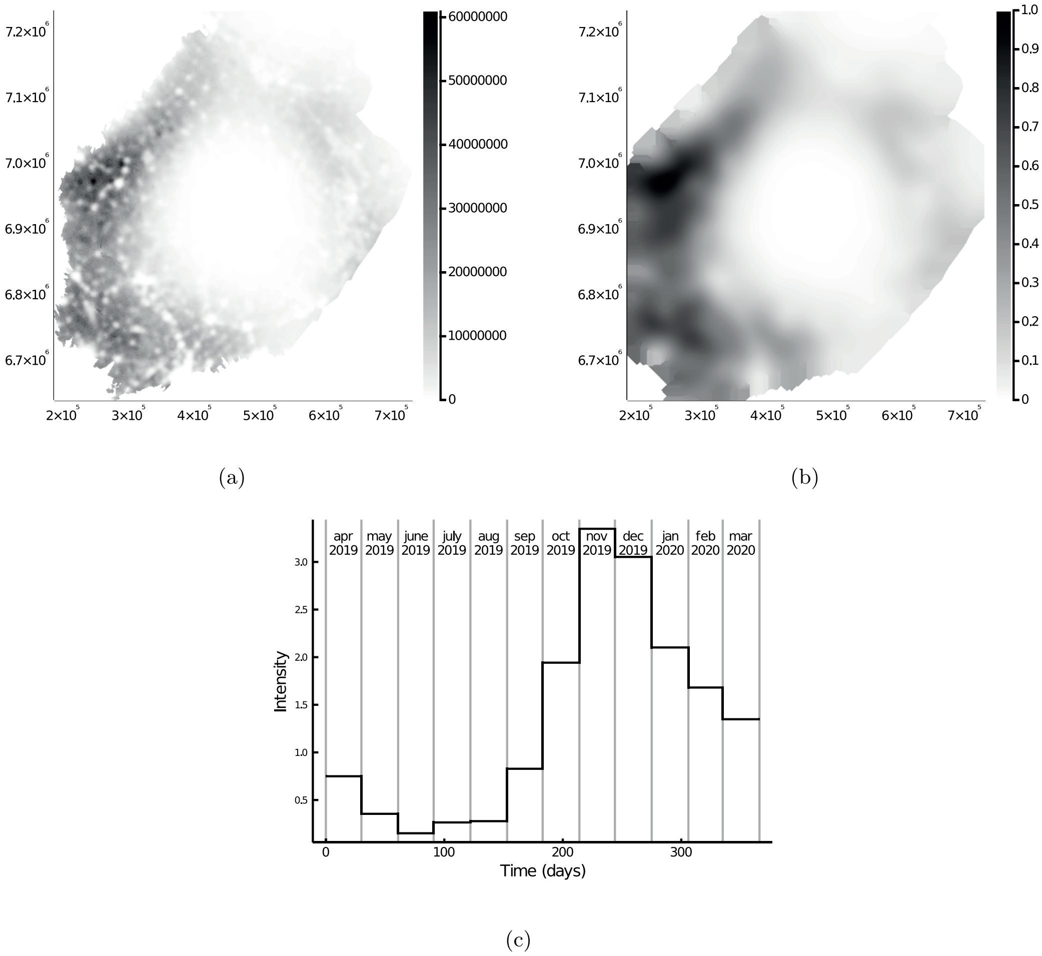


**Fig. 3.** Outputs of the intensity modelling of Sections 2.3–2.4. Plots (b) and (c) show the estimated spatial intensity function $\lambda^{(s)}_{\text{obs}}$ and temporal intensity function $\lambda^{(\tau)}_{\text{obs}}$, respectively. Plot (a) shows the predicted spatial effect $\lambda^{(s)}_{k}$ from the intensity model of Section 2.3 before applying the Gaussian blur as discussed in Section 2.4. The blur standard deviation was set to $\sigma_{\text{obs}} = 13376.67$ m and the domain of $\lambda^{(s)}_{\text{obs}}$ was additionally widened by 27 kilometres at the borders as discussed in Section 2.4.

territory (polygons) in Dataset B discussed in Section 2.1.2. As many of these territories were noncircular in shape (see Fig. 2 (bottom left) for some similar polygons) we estimated $\sigma_{\text{obs}}$ as follows. First, we computed the 95% quantile, $d_{0.95}$, from the empirical distribution of the diameters of the territory polygons in Dataset B. Then, we used Eq. (1) with $\alpha = 0.95$ to compute the $\sigma_{\text{obs}}$ value that corresponds to a radius of $d_{0.95}/2$, yielding the value $\sigma_{\text{obs}} = 13376.67$ m. Here, the 'diameter of a polygon' means the maximal length between any two points of the polygon. This procedure guarantees that typical territories 'fit' inside the 95% probability region of the territory model.

For the remaining filter parameters, we used the values $\lambda_{\text{obs}} = 1.0$, $\lambda_{\text{c}} = 0.475$ and $\lambda_{\text{bd}} = 0.0015$. The choice of the relative values of $\lambda_{\text{obs}}$ and $\lambda_{\text{c}}$ here corresponds to a situation where 1% of the total observation intensity is assumed to arise from clutter observations, when the spatial and temporal intensity functions are constant one, and there are 47 territories for a period of one year.

The choice of $\lambda_{\text{bd}}$ corresponds to a mean territory lifetime of $\frac{1}{0.0015} \approx$ 667 days, a little less than two years. This choice averages between the fact that in reality some wolf territories are short-lived, but some can exists for years. The chosen value also a priori predicts reasonable changes in the territory count over a period of one year, while maintaining a good agreement between the ideal and approximate birth and death models (see Figure 1 in the supplementary material).

With these settings, we simulated a total of 240 datasets for each particle count $M \in \{2^7, \ldots, 2^{12}\}$, and applied the particle filter to each, estimating the mean number of territories at approximately weekly intervals. Each time, the filter was initialised with the initial number of territories following Poisson(47) truncated to [37, 57], with each territory centroid initially following the uniform distribution on $D_\mu$. Fig. 5 shows the deviations computed by subtracting the true number of territories from the estimated mean territory count trajectories for each simulation and all particle counts. In addition, the average deviation and the average absolute deviation are shown. On average, the particle filter appears to recover the true number of territories quite accurately. With increasing numbers of particles, a slight underestimation of the true territory count is revealed. The average absolute deviation further indicates that the discrepancy from the true territory count is typically less than 3 given that a moderate amount of data has been processed.

### 3.4. Application to the Tassu dataset

Next, we applied the developed particle filter with 16 384 particles to Dataset C. As the initial distribution for the territory centroids, we used the centroids seen in Fig. 2 (bottom left), with Gaussian noise with covariance $\Sigma_{\text{obs}}$ added to each. This way, the prior knowledge from the population assessment of March 2019 can be utilised in the filter.

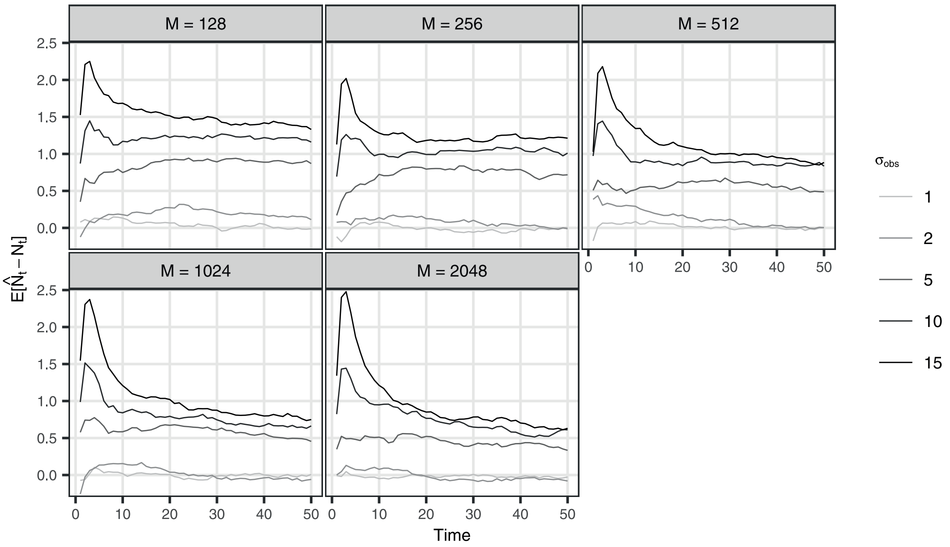


**Fig. 4.** The estimated bias based on 450 simulations for each $M$, $\sigma_{obs}$ and $t$ in the simulation experiment described in the text.

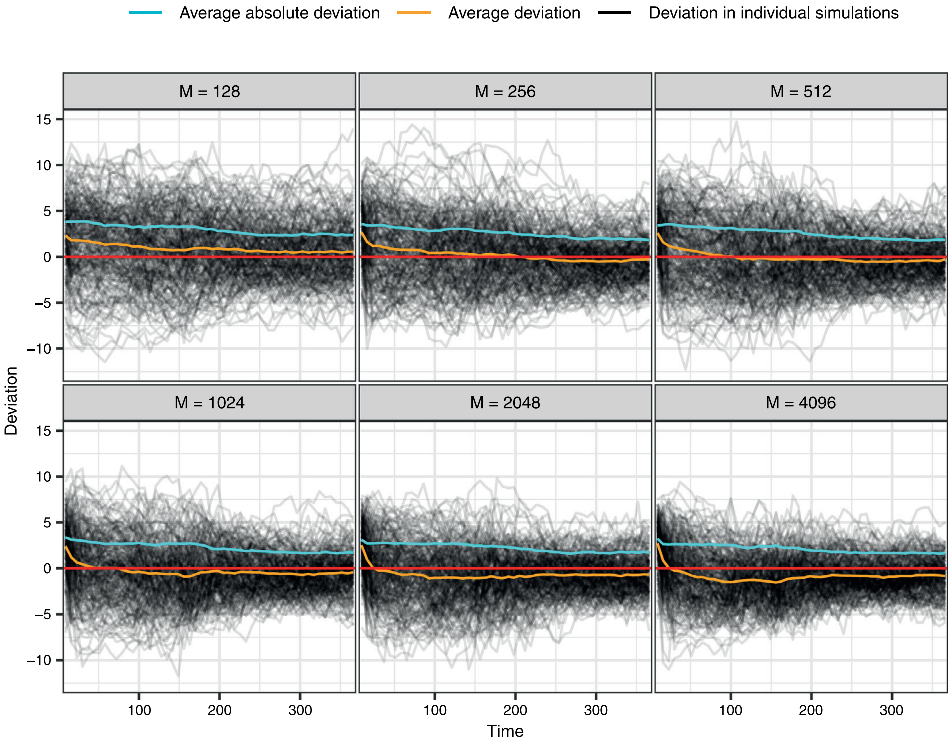


**Fig. 5.** The true number of territories subtracted from territory count estimates (on black) obtained by applying the particle filter to 240 datasets simulated conditional on the intensity functions in Fig. 3 and territory locations fixed to the centroids of the territory polygons in Fig. 2 (bottom left) for a period of one year. The orange and light blue lines represent the average deviation and the average absolute deviation between the true territory count and estimates, respectively.

We report results for four model variants, as follows. Models 1 and 2 correspond to the same model configuration we used in the semisynthetic experiment, but with Model 1 having $\lambda_c = 0$. Models 3 and 4 correspond to models 1 and 2, respectively, but with another set of intensity functions $\lambda^{(\tau)}_{obs}$ and $\lambda^{(s)}_{obs}$, obtained by dropping the terms Corine$_k$, forestroad$_k$ and smooth$(y_k)$ from the intensity model (2), and then estimating $\lambda^{(\tau)}_{obs}$ and $\lambda^{(s)}_{obs}$ as before, as described in Section 2.4. The resulting spatial intensity function in models 3 and 4 is constant one in $D_y$. For a plot of $\lambda^{(s)}_{obs}$ and $\lambda^{(\tau)}_{obs}$ for models 3 and 4, see Figure 2 in the supplementary material.

Fig. 6 displays the estimated territory locations, and Table 1 shows summary statistics of the filtering distribution for the number of territories based on models 1–4 at the end of March 2020, after the

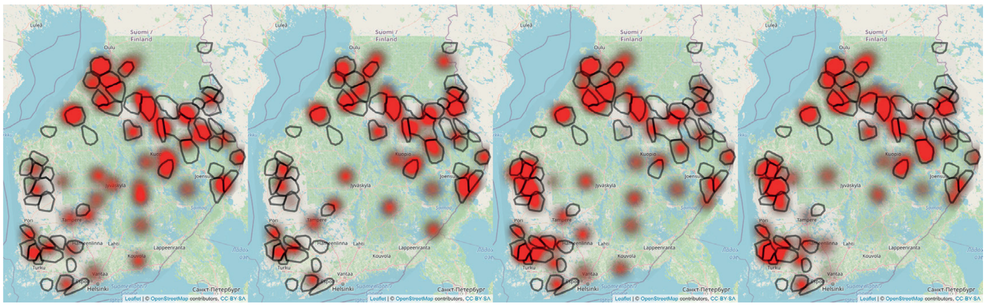


**Fig. 6.** The estimated territory locations (red) at the end of March 2020 and the territory polygons of the wolf territories in March 2020 (black) from the assessment by Luke. Higher intensities of the red colour depict higher plausibility of a territory location. The intensity of the red colour is computed such that the opaque red colour corresponds to the truncated PHD value discussed in Section 2.6. The individual maps show the results for models 1 through 4 from left to right.

**Table 1**
Mean, mode, standard deviation and probability intervals of the estimated distribution for the number of wolf territories in March 2020. The estimate 'Luke' corresponds to the estimate by the Natural Resources Institute Finland (Heikkinen et al., 2020). $I_\alpha$ denotes the $\alpha$% probability interval.

| Estimate | Mean | Mode | St. Dev. | $I_{90}$ | $I_{95}$ | $I_{99}$ |
|---|---|---|---|---|---|---|
| Luke | 46.49 | 47 | 1.93 | [43, 50] | [43, 50] | [41, 51] |
| Model 1 | 49.44 | 49 | 1.55 | [47, 52] | [46, 52] | [46, 53] |
| Model 2 | 44.38 | 44 | 1.14 | [43, 46] | [42, 47] | [42, 48] |
| Model 3 | 55.47 | 55 | 1.60 | [53, 58] | [53, 59] | [52, 60] |
| Model 4 | 55.37 | 53 | 2.12 | [52, 59] | [52, 59] | [51, 60] |

full Dataset C has been filtered. The computations underlying these quantities were carried out as explained in Section 2.6.

The overlaid territory polygons in Fig. 6 correspond to the territories found by Luke in the wolf territory assessment of spring 2020. The plots show that each of the models 1–4 seem to find many of the wolf territories found by Luke. However, it appears that models 1 and 3 with $\lambda_c = 0$ place multiple 'extra' territories to central and southeast Finland, that are not found among the official territories. These territories are most likely not real wolf territories, since only a small number of observations have been reported from these areas; see Fig. 2 (top left). The territories most likely arise since the observation intensity is low (see Fig. 3(b)), and perhaps underestimated in these regions, causing the filter to attempt to explain these observations with additional territories. Based on the plots for model 2 and 4, setting $\lambda_c = 0.475$ seems to mitigate this issue a bit, as these observations are no longer interpreted as real observations from wolf territories. We also experimented with a higher value for $\lambda_c$, but this resulted in a territory count distribution not well aligned with the count estimates by Luke (see also discussion below). There is also no reason to believe that a substantial proportion of the data would not originate from the wolf territories.

Another observation from Fig. 6 is that models 3 and 4, with the simplified intensity functions, seem to somewhat better capture the cluster of territories in west and southwest Finland, in comparison to models 1 and 2. This difference in the results occurs since the spatial intensity function for models 1 and 2 assumes that in these regions, the reporting intensity of the observations is higher than in other parts of Finland. This in turn results in less territories being needed to explain the observations arriving from these regions, under models 1 and 2.

Based on the territory count distributions summarised in Table 1, the territory counts for models 1 and 2 are best aligned with the estimate of Luke. In comparison, the territory count for models 3 and 4 is somewhat overestimated. Fig. 7 reports the sample standard deviations of the obtained mean territory counts at approximately weekly time points when we repeated the filtering of the Tassu data 195 times with the configuration of model 2 and different particle counts. The observation is that the variability in the estimated mean territory counts is seen to diminish with increasing numbers of particles, but still remains noticeable even with 16 384 particles. We also experimented with different $\lambda_{bd}$, $\Sigma_{obs}$ and intensity functions, but the phenomenon persisted. In contrast, when a dataset is simulated from the model, the results of this experiment are markedly different, as is seen from the second pane in the figure. Similarly, there is also some variability in the estimated territory locations. Figure 3 in the supplementary material shows the estimated territory locations after 10 independent runs of the filter to the Tassu data.

## 4. Discussion

We presented a statistical modelling framework for the analysis of citizen science data from territorial animals. At the core of our framework is the data generating model discussed in Section 2.2, that consists of a birth and death model giving rise to the animal territories, and an observation model that links the citizen science observations to the territories. In the developed data generating model, the high variability common to citizen science observation processes is modelled through a temporal and a spatial intensity function, which are assumed fixed and known. The Rao-Blackwellised particle filter described in Section 2.5, estimates the sequence of filtering distributions for the locations and number of territories that describe the knowledge of the animal territories in time as more data is brought in.

We found that the fitted intensity model and the estimated intensity functions were able to capture general trends in the arrival of the citizen science observations as is seen from the estimated intensity functions in Fig. 3. Clearly, the model captures the higher arrival intensity of observations in the winter, which mainly occurs because of snow that leaves wolf tracks visible for potentially long periods of time. The estimated spatial intensity function, on the other hand, captures the high intensity of observations in western Finland and the low intensity of observations in middle Finland compared to other regions.

The intensity model did, however, struggle to explain some characteristics adequately. For instance, sometimes the observations arrived in unexpected bursts, or were highly localised in space, and these features the model was not flexible enough to capture (cf. Figure 4 in the supplementary material). A potential remedy for this might be the addition of random effects e.g. an additional noise component to each spatial pixel, but it would be better to include interpretable rare-event overdispersion components based on the social analysis of the mechanisms for reporting the wolf observations. In fact, the outliers are worth a closer look to gain such insight. The model also struggled with an excess of 0-count cells due to how the conditioning on the auxiliary territories in Dataset B was constructed.

Another improvement for the intensity model could be a zero-inflation component with its own regression structure on, for example, environmental covariates. However, an even better option would be to forgo the aggregation and model the data as a spatio-temporal (marked)

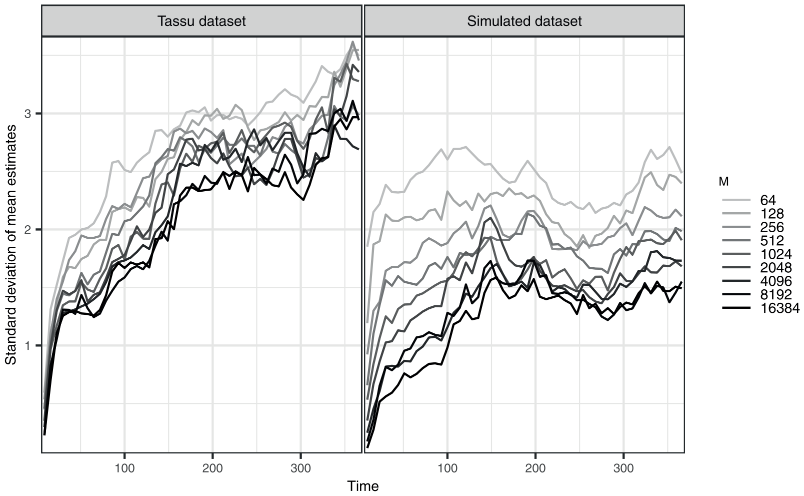


**Fig. 7.** Sample standard deviations over estimates of the mean territory count obtained by running the particle filter 195 times to Dataset C (Tassu dataset) and a single simulated dataset, for varying numbers of particles $M$.

point pattern (Sicacha-Parada et al., 2021; Tang et al., 2021). Modelled this way, events such as an observation being exactly on a road, or snow cover of previous days would not be averaged out, yet coarser resolutions for downstream analysis could still be easily computed. Other possible improvements include using the type of observation (sightings, paw prints, game-cameras etc.) in the intensity model and the improvement of the process of deriving the auxiliary wolf territories from the GPS trajectories. Finally, it might be beneficial to investigate the possibility to relax the space–time separability assumption, which would allow for the incorporation of weather data such as snow cover and/or allow for individual time trends for western and eastern Finland, for example. Such complex models could then be estimated from longer period or more frequent observation series, if available.

Our synthetic experiment in Section 3.2 investigated the mean territory count estimates of our particle filter under a simple data generating model, and found that our method works as expected when Assumption D of Section 2.5 holds. With high territory sizes, this assumption is violated, and there is some bias in the estimation of the mean territory count. This edge effect arises due to the finiteness of the domain $D_\mu$. In particular, the violation of Assumption D reduces the accuracy of the approximation (25) (in the supplementary material), which likely causes the bias. In this experiment, territory parameters on a scale of 1%–2% of the width of the rectangular region led to small bias. Noting that the distance between the western and eastern borders of Finland is approximately 500 kilometres, we therefore expect that the bias caused by the edge effect should be small in the realistic experiments discussed in Sections 3.3–3.4.

The semisynthetic experiment of Section 3.3 showed that under realistic conditions resembling the situation with the Tassu dataset, the estimation of the territory count is possible. There was a slight underestimation of the true territory count with increasing numbers of particles, which we think occurs because of the discrepancy between the scenario and the ideal model, or the approximations used. This experiment was a proof of concept that showed that the territory estimation can be feasible even with presence-only data, when the model is correct.

While experimenting with different parameter values for the data generating model, we found that in general the model and especially the count estimation is most sensitive to the values of the territory size parameter $\Sigma_{\text{obs}}$ and the intensity functions, and least sensitive to the choice of the constant and equal birth rate $\lambda_{\text{bd}}$. This is expected, since large territory sizes increase the probability of an observation being associated with a territory that is far away, decreasing the relative probability of a new territory emerging. In a similar vein, the intensity functions are directly tied to the amount of territories needed to explain the number of observations arriving. With a fixed dataset, lowering the observation intensity results in more births, since a higher number of territories is then needed to explain the data.

The concluding analysis in Section 3.4 applied the developed data generating model and particle filter to analyse citizen science observations of wolves from April 2019 to March 2020. The results show similar patterns as the counts and locations reported by the Natural Resources Institute Finland (Luke) in the official assessment of March 2020. However, the results do not reach the accuracy of expert judgement. This cannot be expected, because the official assessment also incorporated additional information sources, such as DNA samples, GPS collared wolves and mortality records.

Based on Table 1, the inference of Model 2 incorporating the estimated spatial intensity function and clutter observations resulted in a slightly smaller estimated territory count and smaller variability than the official estimate by Luke. In general, such differences can occur because the estimates of Luke are based on a very different model and assumptions, and also take advantage of additional data. When using the citizen science data only, it is also possible that the particle filter might not find territories which rarely produce observations, leading into underestimation of the territory count. Furthermore, the uncertainty reported by the particle filter can be underestimated, because the results are based on a single particle filter run. If Monte Carlo variability is taken into account, the uncertainty is inflated (see also discussion below).

We noticed that with models using a constant one spatial intensity, the cluster of territories in the west and southwest Finland was captured better than with models that used the estimated intensity functions, as was seen from Fig. 6. This might suggest that the observation intensities in the more complex models are overestimated in these regions at the time of the observations arriving. Indeed, based on the Pearson residuals in Figure 4 of the supplementary material, the intensity model fit could not capture all of the variation in these regions. Despite this, it appears that using the spatially varying intensity improves and is likely a requirement for the accurate estimation of the territory count. This is supported by the territory count distribution in Table 1, which indicates that the territory count estimates of the more complex models were better aligned with the official territory count estimates of Luke.

In general, we found that estimating the territory count and locations of the territories reliably is challenging by only using the citizen science observations from the Tassu database. The challenges faced may be partly explained by unsatisfactory observation intensity modelling, but it appears that the inference algorithm is also struggling with real data. Fig. 7 (and 3 in the supplementary material) show repeated runs of the filter in our concluding analysis, indicating some Monte Carlo variability. We believe that the main challenge for the inference is the presence-only nature of the observations. These observations are quite informative about the births of new territories since a territory needs to exists so that an observation may occur. In contrast, the observations are not very informative about the deaths of the territories, because the information about a death of a territory is indirect and only mediated by the absence of observations arriving from a particular area. This difficulty with the data might explain the Monte Carlo variability in our particle filter, and further suggests that it could still benefit from further specialisation to the territory estimation task. For instance, this specialisation could come in the form of further heuristics that eliminate territories more efficiently, when no observations have been associated for a long time.

All in all, we think that the results obtained suggest that the developed modelling framework might be useful as an additional tool in the annual wolf population assessment, reducing the amount of subjectivity in the estimation process by providing a preliminary statistical interpretation of the citizen-made observations. Integration of the DNA samples, GPS collared wolves and other data with the results of the particle filter would still remain a task for the panel of experts. Our analysis with Datasets A, B and C of Section 2.1 showcased the intended use of the modelling framework in the context of the wolf population assessments. First, the intensity functions required by the data generating model are estimated based on the latest historical data. Then, the particle filter is initialised with the territory count and location distribution available from the latest population assessment. Finally, the yearly observations are analysed in a batch to obtain a model-based view on the status of the wolf population at the time of the next population assessment.

Filtering a year of data from April 2019 to March 2020 allowed both the comparison to the official wolf population assessment of 2020 and the use of prior information from the assessment of 2019. However, this yearly batch estimation is not the only way in which the developed data generating model and particle filter could be used. In fact, one of the motivations for the development of the data generating model and the particle filter was that the developed modelling framework could also be used in an online fashion. This way, smaller batches of new observations could be used to update the posterior distribution of the territory locations and track the population in finer timescales. In the context of the wolf territory estimation, the results from such an online estimation could provide dynamic feedback for the volunteers collecting the data, highlighting that their work is important. Such feedback might also be used to direct the effort of the volunteers to areas with the most uncertainty about the existence of territories.

We envision that our modelling scheme could also be a noteworthy tool for refining the population assessment of other large carnivores. For example, the female Eurasian lynx (*Lynx lynx*) are known to show territorial behaviour with cubs. This could be exploited in the estimation of lynx reproduction. It might also be possible to couple the developed framework with other developments for assessing animal populations, such as the spatial capture–recapture (SCR) model which estimates wolf density based on DNA samples (Bischof et al., 2020). For example, the two approaches might be used sequentially. The SCR model could be used first to estimate the spatial wolf density based on DNA samples, and then the density could provide another source of prior information about potential territory locations for our data generating model. On the other hand, in case that the DNA samples are collected by volunteers, the modelling of the sampling effort in the SCR model could be done by similar techniques as in this work.

Besides the context of territorial animals, our methods might be relevant in target tracking where the modelling of the temporal and spatial variability of the observation process is required. The methods may also be regarded as a form of 'dynamic clustering'. In different applications, the modular nature of the developed framework can be exploited to carry out the intensity modelling in a way that fits the application.

There are a number of ways how the developed data generating model and the inference algorithm could be improved in future works. The core of the filtering computation consists of evaluating the posterior probabilities for the birth, death and association variables (see Section 1.6 of the supplementary material). The main approximations made in the computation arise from the intractable integrals related to the spatial domains $D_\mu$ and $D_\mathrm{y}$ and the spatial intensity function $\lambda_\mathrm{obs}^{(s)}$. We concentrated on the situation where $\lambda_\mathrm{obs}^{(s)}$ may be assumed to be slowly varying by Assumption C, allowing for straightforward approximation of the intractable integrals in the probability computations and the measurement update. Introducing a numerical integration scheme could mitigate the bias from the approximations in the former. It might also be possible to incorporate a 'no-overlap' condition to the model that penalises large 'overlaps' of the territories.

The developed particle filtering approach assumes fixed parameter values. Even with the limited and noisy citizen science data, it might be possible to estimate some parameters of the data generating model, such as the intensity scaling parameter $\lambda_\mathrm{obs}$ and/or parameters related to the birth and death intensity functions $\lambda_\mathrm{b}$ and $\lambda_\mathrm{d}$. This could result in a better fit of the data and model, and might result in improved location and count estimates. In theory, estimating the model parameters is possible using for example the particle marginal Metropolis–Hastings algorithm (Andrieu et al., 2010) similar to Kokkala and Särkkä (2015). Using these methods would however require the derivation of an unbiased estimate of the marginal likelihood of the observed data under the employed optimal resampling scheme of Fearnhead and Clifford (2003). For the current model and a moderate to large dataset, such estimation procedures are also computationally intensive and would likely require a tailored parallel implementation. The methods might also be difficult to tune in practice.

The data generating model could, in principle, readily incorporate moving territories as well, perhaps using an Ornstein–Uhlenbeck-type movement model as in the work of Johnson et al. (2008). We did not attempt this with the wolf territory estimation, because our main interest was in immobile territories. In addition, we suspect that the additional flexibility in the model allowing for movement would make the inference task substantially more difficult or even infeasible. Furthermore, we do not believe that in our dataset each territory is observed frequently enough to make the inference with an additional movement model practical.

The observation model could also be further refined. We did not separate between the different observation types, but some observation types can be more reliable than others and may be subject to different observation intensity ('detectability'); consider for instance tracks vs. sightings. We chose to simplify since we do not believe that adequate intensity estimation is possible for the different observation types separately with our dataset. In another context, however, it would in principle be possible to modify the observation model to also accommodate different observation types. The intensity function of the observation model (see Equation (5) in the supplementary material) could be augmented with 'independent data streams' for the different observation types, each with their own spatio-temporal and clutter intensities. This could lead to interesting observation models, such as ones that designate higher clutter intensity for more uncertain observations types such as sightings.

Another means of model improvement are more fine-grained birth–death models. In ecological applications, for example, further information such as mating seasons or typical lifespans of the species could be encoded into the birth and death intensity functions $\lambda_\mathrm{b}$ and $\lambda_\mathrm{d}$

in the general model described in the supplementary material. However, these kinds of models would likely achieve their full potential when coupled with parameter estimation discussed above. In our wolf territory estimation, we did not investigate time-varying birth/death intensity functions and opted for a model where a territory can emerge at any time. Even though wolves only reproduce in the spring, new wolf territories can form at any time of the year when vagrant wolves pair up and establish new territories. Furthermore, the majority of the citizen science wolf observations are made in the winter time, and it is possible that the first observation from a territory formed in the spring comes later, in the winter.

## CRediT authorship contribution statement

**Santeri Karppinen:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization, Project administration. **Tuomas Rajala:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization, Project administration. **Samu Mäntyniemi:** Conceptualization, Writing – original draft, Resources, Writing – review & editing, Supervision, Project administration. **Ilpo Kojola:** Resources, Writing – original draft, Project administration. **Matti Vihola:** Conceptualization, Methodology, Project administration, Writing – original draft, Writing – review & editing, Supervision, Funding acquisition.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data accessibility statement

The code used to reach the conclusions of the paper is available at the repositories

- https://github.com/antiphon/tassu-intensity (intensity analysis)
- https://github.com/skarppinen/tassu-filtering (particle filter implementation and experiments of Sections 3.2–3.4).

The repositories (and links therein) also contain simulation data and results related to the experiments. The data on the Finnish wolf population is not publicly available because it contains sensitive information on local wolf behaviour that could be exploited in poaching or disturbing wolves.

## Acknowledgements

We thank Gabriel Kasmi, who was involved with early experiments related to this research. SK and MV were supported by the Academy of Finland research grant 315619. The authors wish to acknowledge CSC — IT Center for Science, Finland, for computational resources.

## Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.ecolmodel.2022.110101.

## References

Andrieu, C., Doucet, A., Holenstein, R., 2010. Particle Markov chain Monte Carlo methods. J. R. Stat. Soc. Ser. B Stat. Methodol. 72 (3), 269–342.

Bischof, R., Milleret, C., Dupont, P., Chipperfield, J., Tourani, M., Ordiz, A., de Valpine, P., Turek, D., Royle, J.A., Gimenez, O., Flagstad, Ø., Åkesson, M., Svensson, L., Brøseth, H., Kindberg, J., 2020. Estimating and forecasting spatial population dynamics of apex predators using transnational genetic monitoring. Proc. Natl. Acad. Sci. (ISSN: 0027-8424) 117 (48), 30531–30538. http://dx.doi.org/10.1073/pnas.2011383117, URL https://www.pnas.org/content/117/48/30531.

Bragina, E.V., Ives, A., Pidgeon, A., Kuemmerle, T., Baskin, L., Gubar, Y., Piquer-Rodríguez, M., Keuler, N., Petrosyan, V., Radeloff, V., 2015. Rapid declines of large mammal populations after the collapse of the Soviet union. Conserv. Biol. 29 (3), 844–853.

Conrad, C., Hichley, K., 2011. A review of citizen science and community-based environmental monitoring: issues and opportunities. Environ. Monit. Assess. 176, 273–291.

Cretois, B., Linnell, J.D., Grainger, M., Nilsen, E.B., Rød, J.K., 2020. Hunters as citizen scientists: Contributions to biodiversity monitoring in Europe. Global Ecol. Conserv. 23, e01077.

Doucet, A., Godsill, S., Andrieu, C., 2000. On sequential Monte Carlo sampling methods for Bayesian filtering. Stat. Comput. 10 (3), 197–208.

Fearnhead, P., Clifford, P., 2003. On-line inference for hidden Markov models via particle filters. J. R. Stat. Soc. Ser. B Stat. Methodol. 65 (4), 887–899.

Finnish Environment Institute SYKE, 2018. CORINE Land Cover 2018. Data is downloaded from the data download service of SYKE on 18.3.2019 under the license CC 4.0 BY.

Finnish Transport Infrastructure Agency, 2021. Digiroad. Data is downloaded from the download and viewing service of the Finnish Transport Infrastructure Agency on 19.01.2021 under the license CC 4.0 BY.

Goodman, I.R., Mahler, R.P.S., Nguyen, H.T., 1997. Mathematics of Data Fusion. Series B: Mathematical and Statistical Methods, vol. 39, Kluwer Academic Publishers, AA Dordrecht, The Netherlands, ISBN: 0-7923-4674-2.

Gregory, R.D., van Strien, A., Vorisek, P., Gmelig Meyling, A.W., Noble, D., Roy, D., 2005. Developing indicators for European birds. Philos. Trans. R. Soc. Lond. Ser. B Biol. Sci. 360, 269–288.

Heikkinen, S., Kojola, I., Mäntyniemi, S., Holmala, K., Härkälä, A., 2020. Susikanta suomessa maaliskuussa 2020. Luonnonvarakeskus 37, URL http://urn.fi/URN:ISBN:978-952-326-979-8.

Illian, J., Penttinen, A., Stoyan, H., Stoyan, D., 2008. Statistical Analysis and Modelling of Spatial Point Patterns. John Wiley & Sons.

Isaac, N., 2014. Statistics for citizen science: extracting signals from noisy ecological data. Methods Ecol. Evol. 5, 1052–1060.

Johnson, D.S., London, J.M., Lea, M.-A., Durban, J.W., 2008. Continuous-time correlated random walk model for animal telemetry data. Ecology 89 (5), 1208–1215.

Kojola, I., Hallikainen, V., Mikkola, K., Gurarie, E., Heikkinen, S., Kaartinen, S., Nikula, A., Nivala, V., 2016. Wolf visitations close to human residences in Finland: the role of age, residence density, and time of day. Biol. Cons. 198, 9–14.

Kojola, I., Heikkinen, S., Holmala, K., 2018. Balancing costs and confidence: volunteer-provided point observations, GPS telemetry and the genetic monitoring of Finland's wolves. Mammal Res. 63, 415–423. http://dx.doi.org/10.1007/s13364-018-0371-3.

Kokkala, J., Särkkä, S., 2015. Combining particle MCMC with Rao-Blackwellized Monte Carlo data association for parameter estimation in multiple target tracking. Digital Signal Process. 47, 84–95.

Lawrence, A., 2006. No personal motive? Volunteers, biodiversity, and false dichotomies of participation. Ethics Place Environ. 9, 279–298.

Linnell, J.D., Kaczensky, P., Wotschikowsky, U., Lescureux, N., Boitani, L., 2015. Framing the relationship between people and nature in the context of European conservation. Conserv. Biol. 29 (4), 978–985.

Pellikka, J., Hiedanpää, J., 2017. Looking for a common ground: useful knowledge and adaptation in wolf politics in southwestern Finland. Wildlife Biol. 2017 (4).

Renner, I.W., Elith, J., Baddeley, A., Fithian, W., Hastie, T., Phillips, S.J., Popovic, G., Warton, D.I., 2015. Point process models for presence-only analysis. In: O'Hara, R.B. (Ed.), Methods Ecol. Evol. (ISSN: 2041210X) 6 (4), 366–379. http://dx.doi.org/10.1111/2041-210X.12352, URL http://doi.wiley.com/10.1111/2041-210X.12352.

Royle, J., Kéry, M., 2007. A Bayesian state-space formulation of dynamic occupancy models. Ecology 88 (7), 1813–1823.

Särkkä, S., Vehtari, A., Lampinen, J., 2007. Rao-Blackwellized particle filter for multiple target tracking. Inf. Fusion 8 (1), 2–15.

Sicacha-Parada, J., Steinsland, I., Cretois, B., Borgelt, J., 2021. Accounting for spatial varying sampling effort due to accessibility in citizen science data: A case study of moose in Norway. Spat. Statist. (ISSN: 22116753) 42, 100446. http://dx.doi.org/10.1016/j.spasta.2020.100446, URL https://linkinghub.elsevier.com/retrieve/pii/S2211675320300403.

Silvertown, J., 2009. A new dawn for citizen science. Trends Ecol. Evol. 24, 467–471.

Tang, B., Clark, J.S., Gelfand, A.E., 2021. Modeling spatially biased citizen science effort through the eBird database. Environ. Ecol. Stat. http://dx.doi.org/10.1007/s10651-021-00508-1, (ISSN: 1352-8505, 1573-3009). URL https://link.springer.com/10.1007/s10651-021-00508-1.

Ver Hoef, J.M., Johnson, D., Angliss, R., Higham, M., 2021. Species density models from opportunistic citizen science data. Methods Ecol. Evol. 2041–210X.13679. http://dx.doi.org/10.1111/2041-210X.13679, (ISSN: 2041-210X, 2041-210X). URL https://onlinelibrary.wiley.com/doi/10.1111/2041-210X.13679.

Vihola, M., 2007. Rao-Blackwellised particle filtering in random set multitarget tracking. IEEE Trans. Aerosp. Electron. Syst. 43 (2), 689–705.

Wood, S., 2017. Generalized Additive Models: An Introduction with R, second ed. Chapman and Hall/CRC.

Zhao, Q., Royle, A., Boomer, G.S., 2017. Spatially explicit dynamic N-mixture models. Popul. Ecol. 59 (4), 293–300.

# SUPPLEMENTARY MATERIAL FOR 'IDENTIFYING TERRITORIES USING PRESENCE-ONLY CITIZEN SCIENCE DATA: AN APPLICATION TO THE FINNISH WOLF POPULATION'

SANTERI KARPPINEN, TUOMAS RAJALA, SAMU MÄNTYNIEMI, ILPO KOJOLA, AND MATTI VIHOLA

## 1. Mathematical details of the data generating model and particle filter

This section gives more thorough mathematical descriptions of the ideal model, its approximate version and the particle filtering algorithm (Sections 2.2 and 2.5 of the main text, respectively).

### 1.1. Ideal birth and death process.

The birth and death process governs how the territories emerge and disappear in a time interval of interest, which we denote by $[0, T)$. In this section, we will use natural numbers to distinguish between the territories. This numbering of the territories is arbitrary and exists mainly for the purposes of this section as a convenience for describing the model.

Let $\mu_t^{(j)}$ be the territory centroid of territory $j$ at time $t \in [0, T)$. Recall that we assume that the territory centroids are constant; the time subscript will be relevant later. At time 0 we assume that the initial number of territories $N_0$ follows some distribution on the natural numbers. Conditional on $N_0$, the density of the territory centroids has the form

$$\prod_{i=1}^{N_0} f_i(\mu_0^{(i)}),$$

where each (known) component $f_i$ is either $N(\mu_0^{(i)}; x_i, \Sigma_i)$ (truncated to $D_\mu$) with some known location estimate $x_i$ and constant diagonal covariance matrix $\Sigma_i$, or $\mathrm{Unif}(D_\mu)$, where $D_\mu$ is the domain of the territories. We enumerate these initial territories from 1 to $N_0$ (in some order), and in general we will denote by $\beta_j$ the birth time of the $j$th territory, and by $\delta_j$ the death time of the $j$th territory.

With these definitions, the indices of the territories alive at any time $t$ are given by

$$I_t = \{j \in \mathbb{N} : \beta_j \leq t < \delta_j\},$$

and in particular, $I_0 = \{1, 2, \ldots, N_0\}$. Furthermore, in general we will denote the number of territories at time $t$ with $N_t$, noting that $N_t = |I_t|$, the cardinality of the set $I_t$. For any new territory $j \in \mathbb{N}$ emerging at time $t$, the territory centroid $\mu_t^{(j)}$ follows $\mathrm{Unif}(D_\mu)$.

We assume that for the territories $j = N_0 + 1, N_0 + 2, \ldots$ (since $\beta_j = 0$ for all initial territories), the birth time $\beta_j$ has the cumulative distribution function

$$\mathbb{P}(\beta_j \leq t) = 1 - \exp\left(-\int_{\beta_{j-1}}^{t} \lambda_{\mathrm{b}}(u) N_u + \lambda_{\mathrm{b}0} \mathrm{d}u\right), \text{ for } t \geq \beta_{j-1} \tag{1}$$

where $\lambda_{\mathrm{b}}(u)$ is a birth intensity function that gives the birth intensity induced by a single existing territory at time $u$, and $\lambda_{\mathrm{b}0}$ is a constant baseline birth intensity. Here, it is assumed that the probability of a new territory being born increases with growing number of existing territories, modulated by the function $\lambda_{\mathrm{b}}$, which can for example be used to model seasonal variation in the amount of new territories appearing.

Based on (1), it follows that

$$\mathbb{P}(\beta_j \le t \mid \beta_j > \tau) = 1 - \exp\left(-\int_\tau^t \lambda_{\mathrm{b}}(u)N_u + \lambda_{\mathrm{b}0}\,\mathrm{d}u\right), \text{ for } t \ge \tau > \beta_{j-1}. \tag{2}$$

This corresponds to the probability that the $j$th territory is born before time $t$ given that it has not been born before time $\tau$.

Similarly, but independent of the other territories, we assume that each territory dies with rate given by the death intensity function $\lambda_{\mathrm{d}}(u)$. This means, that the time of death for territory $j$, $\delta_j$, has the cumulative distribution function:

$$\mathbb{P}(\delta_j \le t) = 1 - \exp\left(-\int_{\beta_j}^t \lambda_{\mathrm{d}}(u)\mathrm{d}u\right), \text{ for } t \ge \beta_j, \tag{3}$$

which again gives

$$\mathbb{P}(\delta_j \le t \mid \delta_j > \tau) = 1 - \exp\left(-\int_\tau^t \lambda_{\mathrm{d}}(u)du\right), \text{ for } t \ge \tau > \beta_j. \tag{4}$$

Note that the number of territories $N_u$ in the birth and death process described by (2) and (4) depends in principle on all births and deaths, but the process can be simulated sequentially by applying the inverse distribution method, because $N_u$ for $u \le \beta_{n+1}$ depends only on $(\beta_j, \delta_j)$ for $j \le n$.

The absolute values of the birth and death rates are related to the 'persistence' of the model: lower values mean fewer births and deaths. For constant rates $\lambda_{\mathrm{b}}(u) = \lambda_{\mathrm{b}}$ and $\lambda_{\mathrm{d}}(u) = \lambda_{\mathrm{d}}$, it holds that:

- If $\lambda_{\mathrm{b}0} = 0$, and $\lambda_{\mathrm{b}} = \lambda_{\mathrm{d}} > 0$, then $(N_t)$ has constant conditional expectation, given that $N_t$ has not reached zero yet.
- If $\lambda_{\mathrm{b}} > \lambda_{\mathrm{d}}$, then the process $N_t$ can drift to infinity; in case $\lambda_{\mathrm{b}0} > 0$, then it almost surely does.
- If $\lambda_{\mathrm{b}0} > 0$ and $\lambda_{\mathrm{b}} < \lambda_{\mathrm{d}}$, then the process $N_t$ has a nontrivial stationary distribution.

1.2. **Ideal observation model.** Conditional on the territory lifetimes consisting of the locations of the centroids $\mu_{[0,T)}$ and indices $I_{[0,T)}$, we assume that the observation times and locations are generated by a three-dimensional inhomogeneous Poisson process (IPP). The intensity of the IPP is assumed to depend on a time-varying intensity function $\lambda_{\mathrm{obs}}^{(\tau)}$ and a spatially varying intensity function $\lambda_{\mathrm{obs}}^{(s)}$, which are key in accounting for the spatial and temporal variability in citizen science observation processes. We allow for two types of observations: observations that originate from the territories, and observations that are "erroneous" clutter observations that do not originate from any territory. The total intensity at time $u \in [0, T)$ at the point $y \in D_{\mathrm{y}} \subset \mathbb{R}^2$ ($D_\mu \subseteq D_{\mathrm{y}}$) is assumed to be

$$\lambda_{\mathrm{obs,tot}}(u, y \mid I_u, \mu_u) = \sum_{i \in I_u} \lambda_{\mathrm{obs}}\lambda_{\mathrm{obs}}^{(\tau)}(u)N(y; \mu_u^{(i)}, \Sigma_{\mathrm{obs}})\lambda_{\mathrm{obs}}^{(s)}(y) + \lambda_{\mathrm{c}}\lambda_{\mathrm{obs}}^{(\tau)}(u)U(y; D_{\mathrm{y}})\lambda_{\mathrm{obs}}^{(s)}(y), \tag{5}$$

where $N(y; x, \Sigma)$ and $U(y; D_{\mathrm{y}})$ are the Gaussian density with mean $x$ and covariance $\Sigma$ and the uniform density with domain $D_{\mathrm{y}}$, evaluated at the point $y$, respectively. The parameter $\Sigma_{\mathrm{obs}}$ controls the territory size, and the parameters $\lambda_{\mathrm{obs}}$ and $\lambda_{\mathrm{c}}$ scale the relative intensity of the territory and clutter observations, respectively.

It follows from (5) that the likelihood function for the observation times $\tau$ ($0 = \tau_0 < \tau_1 < \cdots < \tau_K < T$) and locations $y$ $(y_1, y_2, \ldots, y_K)$ is given by

(6)
$$p(\tau_{1:K}, y_{1:K}|I_{[0,T)}, \mu_{[0,T)}) = \left[\prod_{k=1}^{K} \lambda_{\mathrm{obs,tot}}(\tau_k, y_k \mid I_{\tau_k}, \mu_{\tau_k}) \exp\left(-\int_{D_{\mathrm{y}}}\int_{\tau_{k-1}}^{\tau_k} \lambda_{\mathrm{obs,tot}}(u, x)\mathrm{d}u\mathrm{d}x\right)\right] \times \exp\left(-\int_{D_{\mathrm{y}}}\int_{\tau_K}^{T} \lambda_{\mathrm{obs,tot}}(u, x)\mathrm{d}u\mathrm{d}x\right),$$

where the dependence on the parameters is left implicit for brevity.

For the purposes of the filtering algorithm described in Section 1.6, we note that we can decompose this likelihood as follows

$$p(\tau_{1:K}, y_{1:K}|I_{[0,T)}, \mu_{[0,T)}) = p(\tau_{1:K} \mid I_{[0,T)}, \mu_{[0,T)})p(y_{1:K} \mid \tau_{1:K}, I_{[0,T)}, \mu_{[0,T)}),$$

where

$$\begin{aligned} p(\tau_{1:K} \mid I_{[0,T)}, \mu_{[0,T)}) &= \int_{D_{\mathrm{y}}^K} p(\tau_{1:K}, y_{1:K} \mid I_{[0,T)}, \mu_{[0,T)})\mathrm{d}y_1\mathrm{d}y_2 \ldots \mathrm{d}y_K \\ &= \left[\prod_{k=1}^{K} \lambda_{\mathrm{obs,tot}}(\tau_k \mid I_{\tau_k}, \mu_{\tau_k}) \exp\left(-\int_{D_{\mathrm{y}}}\int_{\tau_{k-1}}^{\tau_k} \lambda_{\mathrm{obs,tot}}(u, x)\mathrm{d}u\mathrm{d}x\right)\right] \quad (7) \\ &\quad \times \exp\left(-\int_{D_{\mathrm{y}}}\int_{\tau_K}^{T} \lambda_{\mathrm{obs,tot}}(u, x)\mathrm{d}u\mathrm{d}x\right), \end{aligned}$$

with

$$\lambda_{\mathrm{obs,tot}}(\tau_k \mid I_{\tau_k}, \mu_{\tau_k}) = \int_{D_{\mathrm{y}}} \lambda_{\mathrm{obs,tot}}(\tau_k, y \mid I_{\tau_k}, \mu_{\tau_k})\mathrm{d}y. \quad (8)$$

Furthermore, we remark that

$$p(y_{1:K} \mid \tau_{1:K}, I_{[0,T)}, \mu_{[0,T)}) \propto \prod_{k=1}^{K} \lambda_{\mathrm{obs,tot}}(\tau_k, y_k \mid I_{\tau_k}, \mu_{\tau_k}),$$

which allows us to write

$$p(y_{1:K} \mid \tau_{1:K}, I_{[0,T)}, \mu_{[0,T)}) = \prod_{k=1}^{K} p(y_k \mid \mu_{\tau_k}^{(I_{\tau_k})}),$$

where

$$p(y_k \mid \mu_{\tau_k}^{(I_{\tau_k})}) = \sum_{c_k \in \{0\} \cup I_{\tau_k}} p(c_k \mid \mu_{\tau_k}^{(I_{\tau_k})})p(y_k \mid c_k, \mu_{\tau_k}^{(I_{\tau_k})}),$$

with

$$p(y_k \mid c_k, \mu_{\tau_k}^{(I_{\tau_k})}) \propto \begin{cases} U(y_k; D_{\mathrm{y}})\lambda_{\mathrm{obs}}^{(s)}(y_k), & c_k = 0 \\ N(y_k \mid \mu_{\tau_k}^{(c_k)}, \Sigma_{\mathrm{obs}})\lambda_{\mathrm{obs}}^{(s)}(y_k), & \text{otherwise,} \end{cases} \quad (9)$$

and

$$p(c_k \mid \mu_{\tau_k}^{(I_{\tau_k})}) \propto \begin{cases} \lambda_{\mathrm{c}}\lambda_{\mathrm{obs}}^{(\tau)}(\tau_k)\int_{D_{\mathrm{y}}} U(y; D_{\mathrm{y}})\lambda_{\mathrm{obs}}^{(s)}(y)\mathrm{d}y, & c_k = 0 \\ \lambda_{\mathrm{obs}}\lambda_{\mathrm{obs}}^{(\tau)}(\tau_k)\int_{D_{\mathrm{y}}} N(y \mid \mu_{\tau_k}^{(c_k)}, \Sigma_{\mathrm{obs}})\lambda_{\mathrm{obs}}^{(s)}(y)\mathrm{d}y, & \text{otherwise.} \end{cases} \tag{10}$$

Here, the variables $c_k$ are 'association variables' which denote the territory (or clutter, in case $c_k = 0$) from which the observation $y_k$ originated from.

1.3. **Time discretisation and data preprocessing.** We begin our discussion of the time-discretised approximate model by discretising the time interval of interest, $[0, T)$, to intervals $\Delta_k = [t_{k-1}, t_k)$ such that the maximal length of any $\Delta_k$ is $\Delta_{\max}$ and the $\Delta_k$'s need not be of equal length. When the discretisation is fine, we may use the simple Euler-type approximation to approximate the integral of a function $f$:

$$|\Delta_k| f(t_{k-1}) \approx \int_{t_{k-1}}^{t_k} f(u)\mathrm{d}u \approx |\Delta_k| f(t_k). \tag{11}$$

Discretising $[0, T)$ with the maximal interval length set to $\Delta_{\max}$ introduces a new set of observations $\tilde{y}_{1:\tilde{K}}$, each of which is either of two different types: the observation can contain both the observation time and a location datum, that is, $\tilde{y}_k = (t_k, y_k)$, or in case two observations are more than $\Delta_{\max}$ apart, only the time point, that is, $\tilde{y}_k = (t_k, \emptyset)$. We will refer to the second type of observations as discretisation points. Furthermore, we introduce auxiliary variables $E_k = 0$ or $E_k = 1$ to differentiate between these two types of observations, that is, $E_k = 1$ when observation $k$ is a discretisation point.

By Assumption A in Section 2.5 of the main text, we assume that the observations arrive sequentially. With the wolf data, as discussed in the main text, the observation times are not recorded accurately, leading to data which is pooled with a granularity of approximately one day. Because of this, we introduce a data preprocessing step given in Algorithm 1, that artificially disperses the pooled observations in time by dividing the time horizon $[0, T)$ to intervals of length $\Delta_{\max}$ and places the observations (if any) within each interval equidistantly in random order. If no observations fall within any such interval of length $\Delta_{\max}$, the process also adds a discretisation point, ensuring that $|\Delta_k| \leq \Delta_{\max}$ for all $k$. With the wolf data, we set $\Delta_{\max} = 1$. The preprocessing of the data introduces a bias, which is small, because its effect on the intensity of the arrival times of the observations will in practice be small.

After the preprocessing step, we denote the obtained time-discretisation with $\tilde{\tau} = t_{0:\tilde{K}}$, where $t_0 = 0$ and $t_{\tilde{K}} = T$. If the preprocessing step is not required (that is, the data has accurate time stamps), $\tilde{\tau}$ can be interpreted as the original dataset that simply has discretisation points added as required. Note that a discretisation point may also need to be added to the data in the case that the filtering distribution at a certain time point is of interest. In the following sections, we will assume that $\tilde{\tau}$ is distributed according to the IPP of Section 1.2.

1.4. **Approximate birth and death model.** The exact transition probability for the birth and death process of Section 1.1 is intractable, since there are an infinite number of possible birth and death scenarios that could potentially occur during each $\Delta_k$. Under a fine time discretisation, the following simplifications are justified:

(i) At most one birth can occur during any $\Delta_k$.
(ii) At most one death can occur during any $\Delta_k$.
(iii) It is not possible for both a birth and a death to occur during any $\Delta_k$.

**Algorithm 1** The preprocessing algorithm for pooled observations; $\mathcal{Y}_i$ denotes the vector of observations falling to the interval $[\Delta_{\max}(i-1), \Delta_{\max} i)$.

```
Set $k = 0$
Set $T_{cur} = 0.0$
for $i = 1, 2, \ldots$ do
    $n_i^* = \max\{1, \#\mathcal{Y}_i\}$
    if $\#\mathcal{Y}_i = 0$ then
        Set $y_{k+1} = \phi$ ('no observations')
    else
        Set $y_{k+1:k+n_i^*} \leftarrow \mathcal{Y}_i$ (by picking a random order)
    end if
    for $j = 1, \ldots, n_i^*$ do
        Set $t_{k+j} = T_{cur} + \Delta_{\max}/n_i^*$
        Set $T_{cur} = t_{k+j}$
    end for
    $k \leftarrow k + n_i^*$
end for
return locations $y$, timepoints $t$
```

Note that the events where the number of births or deaths is more than one have probabilities of order $|\Delta_k|^2$ in contrast with a single birth or a death, which has a probability of order $|\Delta_k|$. Furthermore, in light of the data preprocessing described in Section 1.3, the maximal value for any $|\Delta_k|$ is controlled by $\Delta_{\max}$.

To construct the approximate birth and death model, three probabilities must be specified with respect to the time interval $\Delta_k$: the probability that a new territory is born, the probability that territory $i$ dies, and the probability that the amount of territories remains the same.

We define the probability of a birth, $p_{\text{birth}}$, by setting

$$\begin{aligned} p_{\text{birth}} &:= \mathbb{P}(\text{at least one birth occurs during } \Delta_k)\mathbb{P}(\text{no deaths occur during } \Delta_k) \\ &\approx (1 - e^{-|\Delta_k|(\lambda_{\text{b}}(t_{k-1})N_{t_{k-1}} + \lambda_{\text{b}0})})e^{-|\Delta_k|\lambda_{\text{d}}(t_{k-1})N_{t_{k-1}}}, \end{aligned}$$

where we have used (11) to approximate the integral in (2). Here, $p_{\text{birth}}$ is defined conditional on no deaths occuring, by assumption (iii).

Similarly, we define the probability for the death of territory $i$, $p_{\text{death}}$, as follows:

$$\begin{aligned} p_{\text{death}} &:= \mathbb{P}(\text{no birth during } \Delta_k)\mathbb{P}(\text{at least one death during } \Delta_k)/N_{t_{k-1}} \\ &\approx e^{-|\Delta_k|(\lambda_{\text{b}}(t_{k-1})N_{t_{k-1}} + \lambda_{\text{b}0})}(1 - e^{-|\Delta_k|N_{t_{k-1}}\lambda_{\text{d}}(t_{k-1})})/N_{t_{k-1}}, \end{aligned}$$

where we have again used (11) on (4) to compute the last term. Here, the probability of at least one death occuring during $\Delta_k$ is distributed evenly among the territories that were alive at time $t_{k-1}$.

Finally, the probability that no births or deaths occur, $p_{\text{nothing}}$, is the complement of $p_{\text{birth}}$ and ($N_{t_{k-1}}$ times) $p_{\text{death}}$:

$$\begin{aligned} p_{\text{nothing}} &= 1 + 2e^{-|\Delta_k|((\lambda_{\text{b}}(t_{k-1}) + \lambda_{\text{d}}(t_{k-1}))N_{t_{k-1}} + \lambda_{\text{b}0})} \\ &\quad - e^{-|\Delta_k|\lambda_{\text{d}}(t_{k-1})N_{t_{k-1}}} - e^{-|\Delta_k|(\lambda_{\text{b}}(t_{k-1})N_{t_{k-1}} + \lambda_{\text{b}0})}. \end{aligned}$$

To summarise, the approximate birth and death model can be written as the discrete probability distribution:

$$p(b_k, d_k \mid I_{t_{k-1}}) = \begin{cases} p_{\text{birth}}, & b_k = 1,\ d_k = 0 \\ p_{\text{death}}, & b_k = 0,\ d_k \in I_{t_{k-1}} \\ p_{\text{nothing}}, & b_k = 0,\ d_k = 0 \\ 0, & \text{otherwise}, \end{cases} \tag{12}$$

where $b_k \in \{0, 1\}$ indicates whether a birth occurs during $\Delta_k$, and $d_k \in \{0\} \cup I_{t_{k-1}}$ indicates which territory (if any) died during $\Delta_k$.

Given the values of $b_k$ and $d_k$, the value of $I_{t_k}$ is deterministic,

$$I_{t_k} = \begin{cases} I_{t_{k-1}} \cup L, & \text{if } b_k = 1, d_k = 0 \\ I_{t_{k-1}} \setminus d_k, & \text{if } b_k = 0, d_k \in I_{t_{k-1}} \\ I_{t_{k-1}}, & \text{if } b_k = 0, d_k = 0, \end{cases} \tag{13}$$

where $L$ is the next index in the numbering of the territories. Finally, the centroids of the territories alive at time $t_k$ are given by:

$$\mu_{t_k}^{(I_{t_k})} \mid \mu_{t_{k-1}}^{(I_{t_{k-1}})}, b_k, d_k = \begin{cases} \mu_{t_{k-1}}^{(I_{t_{k-1}})} \cup \mu_{\text{new}}, & b_k = 1,\ d_k = 0 \\ \mu_{t_{k-1}}^{(I_{t_{k-1}} \setminus \{i\})}, & b_k = 0,\ d_k = i \in I_{t_{k-1}} \\ \mu_{t_{k-1}}^{(I_{t_{k-1}})}, & b_k = 0,\ d_k = 0, \end{cases} \tag{14}$$

where $\mu_{\text{new}} \sim \text{Unif}(D_\mu)$.

The approximate birth and death model behaves similarly to the ideal birth and death process. Empirical evidence for this can be seen from Figure 1 which compares the sample standard deviations of the number of territories at 365 time points spaced by intervals of unit length. The standard deviations were computed over 50000 simulated trajectories for the number of territories under the ideal and approximate models, when the initial number of territories was each time set to 47. The figure depicts the models in the configuration we use with the wolf data, that is, $\lambda_{\text{b}}(u) = \lambda_{\text{d}}(u) = \lambda_{\text{bd}}$ for all $u \in [0, 365)$ and $\lambda_{\text{b0}} = 0$. Note that under this configuration the ideal and approximate model also have the same expected numbers of territories at any time point. If higher birth and death intensities are used, $|\Delta_k|$ should be decreased to ensure a good approximation.

1.5. **Approximate observation model.** We use (11) to approximate the inner integrals of (7) and obtain the time-discretised likelihood of the observation times

$$\begin{aligned} p\big(\tilde{\tau} \mid (I_{t_1}, \mu_{t_1}, \ldots, I_{t_{\tilde{K}}}, \mu_{t_{\tilde{K}}})\big) \approx & \left[ \prod_{\{k : E_k = 0\}} \lambda_{\text{obs,tot}}(t_k \mid I_{t_k}, \mu_{t_k}) \right] \\ & \times \left[ \prod_{k=1}^{\tilde{K}} \exp\Big( - |\Delta_k| \lambda_{\text{obs,tot}}(t_k \mid I_{t_k}, \mu_{t_k}) \Big) \right]. \end{aligned} \tag{15}$$

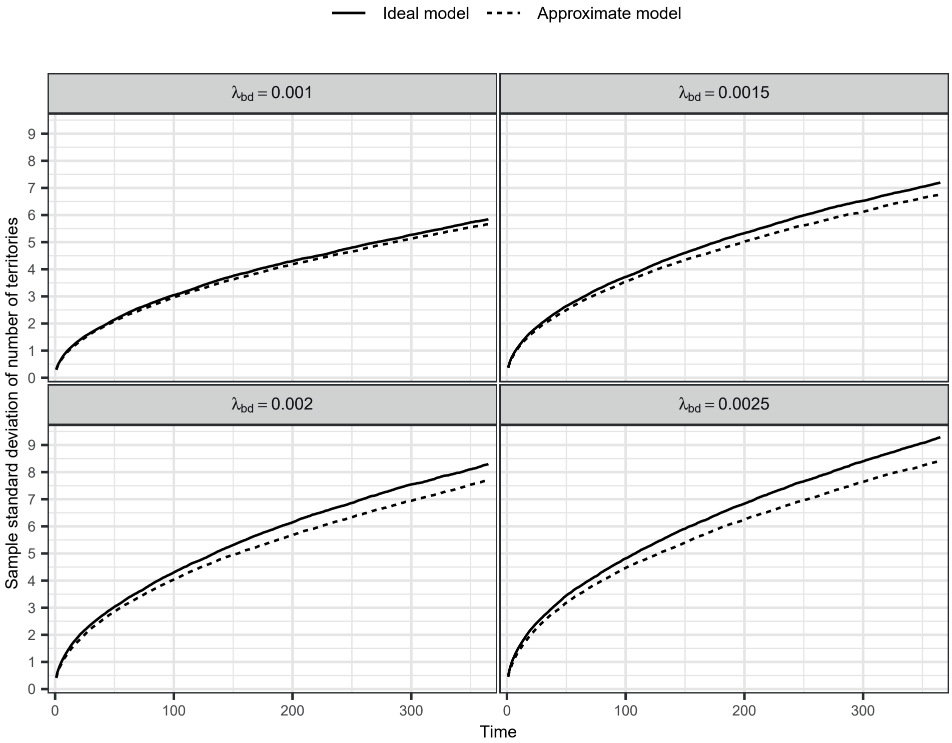


FIGURE 1. Comparison of the standard deviations of the number of territories between the ideal and approximate birth death models based on 50000 simulations with small, constant and equal birth and death intensities. The interval length $|\Delta_k|$ was set to 1 for all $k$ in the simulation.

From here onwards, without explicit reference to $E_k$, we shall use $p(t_k \mid I_{t_k}, \mu_{t_k})$ to denote the individual factors of this approximate likelihood, as follows:

$$p(t_k \mid I_{t_k}, \mu_{t_k}) = \lambda_{\text{obs,tot}}(t_k \mid I_{t_k}, \mu_{t_k}) \exp\big( - |\Delta_k| \lambda_{\text{obs,tot}}(t_k \mid I_{t_k}, \mu_{t_k})\big), \qquad \text{if } E_k = 0 \tag{16}$$

$$p(t_k \mid I_{t_k}, \mu_{t_k}) = \exp\big( - |\Delta_k| \lambda_{\text{obs,tot}}(t_k \mid I_{t_k}, \mu_{t_k})\big), \qquad \text{if } E_k = 1, \tag{17}$$

where $\lambda_{\text{obs,tot}}(t_k \mid I_{t_k}, \mu_{t_k})$ is given by (8) with $\tau_k$ replaced by $t_k$. Finally, the densities (9) and (10) remain as in the ideal model, again replacing $\tau_k$ with $t_k$.

1.6. **State estimation using a Rao-Blackwellised particle filter.** Next, we discuss the inference of the distribution of the territory centroid locations $\mu_{t_k}^{(I_{t_k})}$ for $k = 1, 2, \ldots, \tilde{K}$ given the observed data $\tilde{y}$. This section will only consider the time-discretised model detailed in the previous sections, and therefore we will index the state variables interchangeably with time or time index (that is, $I_{t_k} = I_k$, for example).

We are interested in the filtering distributions $p(\mu_k^{(I_k)} \mid \tilde{y}_{1:k})$ for $k = 1, \ldots, \tilde{K}$. Let $r_k$ denote $(b_k, d_k, c_k)$ for $k \geq 1$, and $I_0$ for $k = 0$. That is, the sequence $r_{0:k}$ contains all knowledge about the births, deaths and associations that occurred until time $t_k$, and the territory indices that were alive at any time up to $t_k$.

The filtering distribution of the territory centroid locations given the data can be inferred using a Rao-Blackwellised particle filter (RBPF) [cf. Doucet et al., 2001]. More specifically, our method is very similar to the particle filter using the optimal resampling strategy described in the work of Fearnhead and Clifford [2003].

A key building block of their particle filter is the computation of the posterior probabilities

$$p(r_k \mid r_{0:k-1}, \tilde{y}_{1:k}) = \frac{p(r_k, \tilde{y}_k \mid r_{0:k-1}, \tilde{y}_{1:k-1})}{\sum_{r'_k} p(r'_k, \tilde{y}_k \mid r_{0:k-1}, \tilde{y}_{1:k-1})}. \tag{18}$$

We will next discuss how we can approximately evaluate the probabilities (18) in the context of our model. To begin, we note that

$$p(r_k, \tilde{y}_k \mid r_{0:k-1}, \tilde{y}_{1:k-1}) = \int_{D_\mu^{N_k}} p(r_k, \tilde{y}_k, \mu_k^{(I_k)} \mid r_{0:k-1}, \tilde{y}_{1:k-1}) \mathrm{d}\mu_k^{(I_k)}, \tag{19}$$

where $D_\mu^{N_k}$ is the $N_k$-dimensional ($N_k = |I_k|$) Cartesian product of $D_\mu$. Then, since $I_{k-1}$ is deterministic given $r_{0:k-1}$, the integrand can be written as

$$\begin{aligned} &p(r_k, \tilde{y}_k, \mu_k^{(I_k)} \mid r_{0:k-1}, \tilde{y}_{1:k-1}) \\ &= p(b_k, d_k \mid I_{k-1}) p(\mu_k^{(I_k)} \mid b_k, d_k, r_{0:k-1}, y_{1:k-1}) p(t_k \mid I_k, \mu_k) p(c_k \mid \mu_k^{(I_k)}) p(y_k \mid \mu_k^{(I_k)}, c_k), \end{aligned} \tag{20}$$

where

$$p(\mu_k^{(I_k)} \mid b_k, d_k, r_{0:k-1}, y_{1:k-1}) = \prod_{i \in I_k} f_i(\mu_k^{(i)}).$$

Here, the territory-specific densities $f_i$ are either $N(\cdot; m_k^{(i)}, C_k^{(i)})$ where $m_k^{(i)}, C_k^{(i)}$ are the predictive mean and covariance of territory $i$, which are functions of $b_k, d_k, r_{0:k-1}, y_{1:k-1}$, or $f_i(\mu_k^{(i)}) = U(\cdot; D_\mu)$ in case the territory has not been associated with an observation yet. Next, we shall consider the approximate evaluation of the integral (19), which by (20) can be written as

$$p(b_k, d_k \mid I_{k-1}) \int_{D_\mu^{N_k}} p(\mu_k^{(I_k)} \mid b_k, d_k, r_{0:k-1}, y_{1:k-1}) p(t_k \mid I_k, \mu_k) p(c_k \mid \mu_k^{(I_k)}) p(y_k \mid \mu_k^{(I_k)}, c_k) \mathrm{d}\mu_k^{(I_k)}.$$

Noting the cancellation of normalisation constants and factors in the product

$$p(t_k \mid I_k, \mu_k) p(c_k \mid \mu_k^{(I_k)}) p(y_k \mid \mu_k^{(I_k)}, c_k),$$

(by Equations (9), (10) and (16)) a direct computation yields

$$\begin{aligned} &\int_{D_\mu^{N_k}} p(\mu_k^{(I_k)} \mid b_k, d_k, r_{0:k-1}, y_{1:k-1}) p(t_k \mid I_k, \mu_k) p(c_k \mid \mu_k^{(I_k)}) p(y_k \mid \mu_k^{(I_k)}, c_k) \mathrm{d}\mu_k^{(I_k)} \\ &= [\lambda_{\mathrm{obs}}^{1(c_k>0)} (\lambda_{\mathrm{c}} U(y_k; D_{\mathrm{y}}))^{1(c_k=0)} \lambda_{\mathrm{obs}}^{(\tau)}(t_k) \lambda_{\mathrm{obs}}^{(s)}(y_k)]^{1(E_k=0)} \exp\left(-|\Delta_k| \lambda_{\mathrm{c}} \lambda_{\mathrm{obs}}^{(\tau)}(t_k) \tilde{C}_U\right) \\ &\times \int_{D_\mu^{N_k}} \prod_{i \in I_k} [f_i(\mu_k^{(i)}) \exp\left(-|\Delta_k| \lambda_{\mathrm{obs}} \lambda_{\mathrm{obs}}^{(\tau)}(t_k) \tilde{C}_N(\mu_k^{(i)})\right)] N(y_k; \mu_k^{(c_k)}, \Sigma_{\mathrm{obs}})^{1(E_k=0, c_k>0)} \mathrm{d}\mu_k^{(I_k)}, \end{aligned} \tag{21}$$

where $\tilde{C}_U = \int_{D_{\mathrm{y}}} \lambda_{\mathrm{obs}}^{(s)}(y)U(y; D_{\mathrm{y}})\mathrm{d}y$, $\tilde{C}_N(\mu_k^{(i)}) = \int_{D_{\mathrm{y}}} \lambda_{\mathrm{obs}}^{(s)}(y)N(y; \mu_k^{(i)}, \Sigma_{\mathrm{obs}})\mathrm{d}y$ and $1(\cdot)$ stands for the indicator function. The integral on the last line can also be written as

$$\begin{aligned}
&\prod_{i\in I_k\setminus\{c_k\}} \int_{D_\mu} [f_i(\mu_k^{(i)}) \exp(-|\Delta_k|\lambda_{\mathrm{obs}}\lambda_{\mathrm{obs}}^{(\tau)}(t_k)\tilde{C}_N(\mu_k^{(i)}))]\mathrm{d}\mu_k^{(i)} \\
&\times \left[\int_{D_\mu} f_{c_k}(\mu_k^{(c_k)}) \exp(-|\Delta_k|\lambda_{\mathrm{obs}}\lambda_{\mathrm{obs}}^{(\tau)}(t_k)\tilde{C}_N(\mu_k^{(c_k)}))N(y_k; \mu_k^{(c_k)}, \Sigma_{\mathrm{obs}})^{1(E_k=0)}\mathrm{d}\mu_k^{(c_k)}\right]^{1(c_k>0)} \qquad (22)\\
&:= \left(\prod_{i\in I_k\setminus\{c_k\}} \tilde{I}_1^{(i)}\right) \tilde{I}_2^{(c_k)},
\end{aligned}$$

which highlights the fact that territories not associated with $y_k$ each contribute $\tilde{I}_1^{(i)}$ to the posterior probability, whereas the associated territory contributes $\tilde{I}_2^{(c_k)}$ (given $E_k = 0$ and $y_k$ not clutter).

To evaluate (21), we approximate the two different kinds of integrals $\tilde{I}_1^{(i)}$ and $\tilde{I}_2^{(c_k)}$ in (22). We will begin by considering the approximation of $\tilde{I}_1^{(i)}$. Assume first that $f_i = N(m_k^{(i)}, C_k^{(i)})$. Then, by using Assumptions C (twice) and D in Section 2.5 of the main text, we approximate

$$\begin{aligned}
\tilde{I}_1^{(i)} &= \int_{D_\mu} N(\mu_k^{(i)}; m_k^{(i)}, C_k^{(i)}) \exp(-|\Delta_k|\lambda_{\mathrm{obs}}\lambda_{\mathrm{obs}}^{(\tau)}(t_k)\tilde{C}_N(\mu_k^{(i)}))\mathrm{d}\mu_k^{(i)} \\
&\approx \int_{D_\mu} N(\mu_k^{(i)}; m_k^{(i)}, C_k^{(i)}) \exp(-|\Delta_k|\lambda_{\mathrm{obs}}\lambda_{\mathrm{obs}}^{(\tau)}(t_k)\lambda_{\mathrm{obs}}^{(s)}(\mu_k^{(i)}))\mathrm{d}\mu_k^{(i)} \qquad (23)\\
&\approx \exp(-|\Delta_k|\lambda_{\mathrm{obs}}\lambda_{\mathrm{obs}}^{(\tau)}(t_k)\lambda_{\mathrm{obs}}^{(s)}(m_k^{(i)})).
\end{aligned}$$

Consider then that $f_i = U(D_\mu)$. We approximate

$$\begin{aligned}
\tilde{I}_1^{(i)} &= \int_{D_\mu} U(\mu_k^{(i)}; D_\mu) \exp(-|\Delta_k|\lambda_{\mathrm{obs}}\lambda_{\mathrm{obs}}^{(\tau)}(t_k)\tilde{C}_N(\mu_k^{(i)}))\mathrm{d}\mu_k^{(i)} \\
&\approx \int_{D_\mu} U(\mu_k^{(i)}; D_\mu) \exp(-|\Delta_k|\lambda_{\mathrm{obs}}\lambda_{\mathrm{obs}}^{(\tau)}(t_k)\lambda_{\mathrm{obs}}^{(s)}(\mu_k^{(i)}))\mathrm{d}\mu_k^{(i)} \qquad (24)\\
&\approx \exp\left(-|\Delta_k|\lambda_{\mathrm{obs}}\lambda_{\mathrm{obs}}^{(\tau)}(t_k) \int_{D_\mu} \lambda_{\mathrm{obs}}^{(s)}(\mu_k^{(i)})U(\mu_k^{(i)}; D_\mu)\mathrm{d}\mu_k^{(i)}\right),
\end{aligned}$$

where the first approximation can be justified again by using Assumption C, and the second one using the first order Taylor series $\exp(-x) \approx 1 - x$ (twice, first from left to right, then from right to left). In summary, (23) and (24) together give the approximation for $\tilde{I}_1^{(i)}$:

$$\tilde{I}_1^{(i)} \approx \begin{cases} \exp\left(-|\Delta_k|\lambda_{\mathrm{obs}}\lambda_{\mathrm{obs}}^{(\tau)}(t_k) \int_{D_\mu} \lambda_{\mathrm{obs}}^{(s)}(\mu_k^{(i)})U(\mu_k^{(i)}; D_\mu)\mathrm{d}\mu_k^{(i)}\right), & \text{if } f_i = U(D_\mu), \\ \exp(-|\Delta_k|\lambda_{\mathrm{obs}}\lambda_{\mathrm{obs}}^{(\tau)}(t_k)\lambda_{\mathrm{obs}}^{(s)}(m_k^{(i)})), & \text{if } f_i = N(m_k^{(i)}, C_k^{(i)}), \end{cases} \qquad (25)$$

where the integral in the first case can be precomputed.

Next, we consider the approximation of the integral $\tilde{I}_2^{(c_k)}$ in (22). First note that if $E_k \neq 0$ or $c_k = 0$, $\tilde{I}_2^{(c_k)} = \tilde{I}_1^{(c_k)}$ and we can use approximation (25). Otherwise, assuming $E_k = 0$ and

$c_k > 0$, and denoting

$$C_{\tilde{I}_2} = \int_{D_\mu} f_{c_k}(\mu_k^{(c_k)}) N(y_k; \mu_k^{(c_k)}, \Sigma_{\text{obs}}) \mathrm{d}\mu_k^{(c_k)},$$

$\tilde{I}_2^{(c_k)}$ may be approximated by

$$\begin{aligned}
\tilde{I}_2^{(c_k)} &= \int_{D_\mu} f_{c_k}(\mu_k^{(c_k)}) \exp\left(-|\Delta_k| \lambda_{\text{obs}} \lambda_{\text{obs}}^{(\tau)}(t_k) \tilde{C}_N(\mu_k^{(c_k)})\right) N(y_k; \mu_k^{(c_k)}, \Sigma_{\text{obs}}) \mathrm{d}\mu_k^{(c_k)} \\
&= C_{\tilde{I}_2} \int_{D_\mu} \exp\left(-|\Delta_k| \lambda_{\text{obs}} \lambda_{\text{obs}}^{(\tau)}(t_k) \tilde{C}_N(\mu_k^{(c_k)})\right) \frac{N(y_k; \mu_k^{(c_k)}, \Sigma_{\text{obs}}) f_{c_k}(\mu_k^{(c_k)})}{C_{\tilde{I}_2}} \mathrm{d}\mu_k^{(c_k)} \qquad (26) \\
&\approx \exp\left(-|\Delta_k| \lambda_{\text{obs}} \lambda_{\text{obs}}^{(\tau)}(t_k) \lambda_{\text{obs}}^{(s)}(m_{k,+}^{(c_k)})\right) \int_{D_\mu} f_{c_k}(\mu_k^{(c_k)}) N(y_k; \mu_k^{(c_k)}, \Sigma_{\text{obs}}) \mathrm{d}\mu_k^{(c_k)},
\end{aligned}$$

where the approximation may be justified by Assumption C. Here, $m_{k,+}^{(c_k)}$ is the mean of the Gaussian distribution $p(\mu_k^{(c_k)} \mid y_{1:k}, r_{0:k})$ if $f_{c_k} = N(m_k^{(c_k)}, C_k^{(c_k)})$ (available from the Kalman update step), and $m_{k,+}^{(c_k)} = y_k$ if $f_{c_k} = U(D_\mu)$. Furthermore,

$$\int_{D_\mu} f_{c_k}(\mu_k^{(c_k)}) N(y_k; \mu_k^{(c_k)}, \Sigma_{\text{obs}}) \mathrm{d}\mu_k^{(c_k)} \approx \begin{cases} K_l^{(c_k)}, & \text{if } f_{c_k} = N(m_k^{(c_k)}, C_k^{(c_k)}) \\ 1/|D_\mu|, & \text{if } f_{c_k} = U(D_\mu), \end{cases} \qquad (27)$$

where $K_l^{(c_k)}$ is the likelihood of the observation, again available from the Kalman update step. The approximation (27) is accurate when $y_k$ is within $D_\mu$, and not close to its boundary.

Taking everything together, (19) equals

$$\begin{aligned}
&p(b_k, d_k \mid I_{k-1}) [\lambda_{\text{obs}}^{1(c_k>0)} (\lambda_{\text{c}} U(y_k; D_{\text{y}}))^{1(c_k=0)} \lambda_{\text{obs}}^{(\tau)}(t_k) \lambda_{\text{obs}}^{(s)}(y_k)]^{1(E_k=0)} \\
&\times \exp\left(-|\Delta_k| \lambda_{\text{c}} \lambda_{\text{obs}}^{(\tau)}(t_k) \tilde{C}_U\right) \left( \prod_{i \in I_k \setminus c_k} \tilde{I}_1^{(i)} \right) \tilde{I}_2^{(c_k)},
\end{aligned} \qquad (28)$$

which we approximate using (25), (26) and (27). Note that in this expression, the values of $b_k$ and $d_k$ influence the cardinality of the set $I_k$.

We utilise the approximate unnormalised probabilities in Equation (28) in the particle filter of Fearnhead and Clifford [2003] which couples an exhaustive one step lookahead for each particle with a clever resampling step that guarantees the uniqueness of the particles in each filtering distribution and is optimal among resampling algorithms that minimise a squared error loss function. A single step of the method, starting from a set of $M$ weighted particles and processing the next observation $\tilde{y}_k$, can be summarised as follows:

(1) For each particle, construct the set of outcomes that can occur to it in the next time interval. For particle $i$ in the case that $E_k = 0$, there are $R(N_{i,k-1}) = O(N_{i,k-1}^2)$ outcomes (consisting of all valid combinations of $b_k \in \{0, 1\}, d_k \in \{0\} \cup I_{i,k-1}, c_k \in \{0\} \cup I_{i,k}$), where $N_{i,k-1} = |I_{i,k-1}|$, the number of territories in particle $i$ prior to processing observation $\tilde{y}_k$. If $E_k = 1$, the values of $c_k$ do not have to be considered, and $R(N_{i,k-1}) = O(N_{i,k-1})$. In total, there are $K = \sum_{i=1}^{M} R(N_{i,k-1})$ outcomes. Denote the (normalised) weights of these $K$ outcomes (possible future particles) by $q^{(j)}$, $j = 1, \ldots, K$. Each (unnormalised) $q^{(j)}$ is computed by multiplying the value of (28) for the outcome by the original weight of the particle.

(2) Compute the unique value for a constant, denoted by $c$, such that $M = \sum_{j=1}^{K} \min(cq^{(j)}, 1)$. This computation is discussed in detail in Section 1.7.
(3) Partition the new set of $K$ outcomes to two sets, set 1 and set 2. For $j = 1, \dots, K$, if $q^{(j)} \geq 1/c$, place outcome $j$ to set 1; otherwise place outcome $j$ in set 2. Denote by $L$ the number of outcomes placed to set 1.
(4) Use stratified resampling to resample $M - L$ outcomes from set 2. The stratified resampling algorithm is given in Appendix B of [Fearnhead and Clifford, 2003].
(5) Output a set of $M$ weighted particles that have been constructed based on the $L$ outcomes in set 1, each with original weights, and based on the $M - L$ outcomes resampled from set 2, each assigned the weight $1/c$.

Each particle in the $M$ output particles sampled by this method represents a hypothesis of

$$p(\mu_k^{(I_k)} \mid r_{0:k}, y_{1:k}) = \prod_{j \in I_k} p(\mu_k^{(j)} \mid r_{0:k}, y_{1:k}), \tag{29}$$

where each factor of the product is computed as follows. If $c_k$ points to a territory that has been associated at least once before, we have

$$p(\mu_k^{(c_k)} \mid y_{1:k}, r_{0:k}) \approx N(\mu_k^{(c_k)}; m_{k,+}^{(c_k)}, C_{k,+}^{(c_k)}), \tag{30}$$

where $m_{k,+}^{(c_k)}, C_{k,+}^{(c_k)}$ refer to the mean and covariance available from the Kalman filter update step, when the normal distribution $p(\mu_k^{(c_k)} \mid b_k, d_k, r_{0:k-1}, y_{1:k-1})$ is updated with the observation $y_k$.

Similarly, if $c_k$ points to a newborn territory,

$$p(\mu_k^{(c_k)} \mid y_{1:k}, r_{0:k}) \approx N(\mu_k^{(c_k)}; y_k, \Sigma_{\text{obs}}), \tag{31}$$

when $D_\mu$ is large and $y_k$ is sufficiently far from the boundary of $D_\mu$. Finally, for $j \neq c_k$, we have

$$p(\mu_k^{(j)} \mid y_{1:k}, r_{0:k}) = p(\mu_k^{(j)} \mid b_k, d_k, r_{0:k-1}, y_{1:k-1}),$$

since the observation $y_k$ is only informative about the territory it was associated with.

1.7. **Computing the constant $c$.** The following computation was described by Fearnhead and Clifford [2003]. Denote by $J$ the set of normalised weights,

$$J = \{q^{(1)}, \dots, q^{(K)}\}.$$

Furthermore, define

$$I^{(M)} := \left\{ \kappa \in J : \sum_{j=1}^{K} \min\left(\frac{q^{(j)}}{\kappa}, 1\right) \leq M \right\}.$$

By splitting the sum in the definition for $I^{(M)}$ to two parts, the inequality in the condition can also be written as

$$\kappa^{-1} B_\kappa + A_\kappa \leq M, \tag{32}$$

where $A_\kappa$ is the number of elements in $J$ that are greater or equal to $\kappa$, and $B_\kappa$ is the sum of the remaining elements in $J$. The value for $c$ is given as follows

$$c = \begin{cases} M, & \text{if } I^{(M)} = \emptyset \\ (M - A_{\kappa_{\min}})/B_{\kappa_{\min}}, & \text{if } I^{(M)} \neq \emptyset, \end{cases} \tag{33}$$

where $\kappa_{\min}$ is the smallest weight in the set $I^{(M)}$. The values of $A_\kappa$ and $B_\kappa$ (or determining that $I^{(M)} = \emptyset$) can be found using a selection algorithm, which has an average running time of $O(M)$.

In the description of this algorithm, two helper routines given in Algorithms 2 and 3 (see Cormen et al. [2009] p. 171 and p. 179, respectively) are used.

**Algorithm 2** *partition*(array A, index l, index u)

```
$x = A[u]$
$i = l - 1$
for $j = l$ to $u - 1$ do
    if $A[j] \leq x$ then
        $i = i + 1$
        Exchange $A[i]$ with $A[j]$.
    end if
end for
Exchange $A[i + 1]$ with $A[u]$.
return i + 1
```

**Algorithm 3** *random-partition*(array A, index l, index u)

```
$i = random(l, u)$
Exchange $A[u]$ with $A[i]$
return partition(A, l, u)
```

In Algorithm 3, *random(l, u)* draws an index at random between $l$ and $u$ (inclusive). Algorithm 2 returns an index $k$ such that for the array $A$ and index $k$, the condition

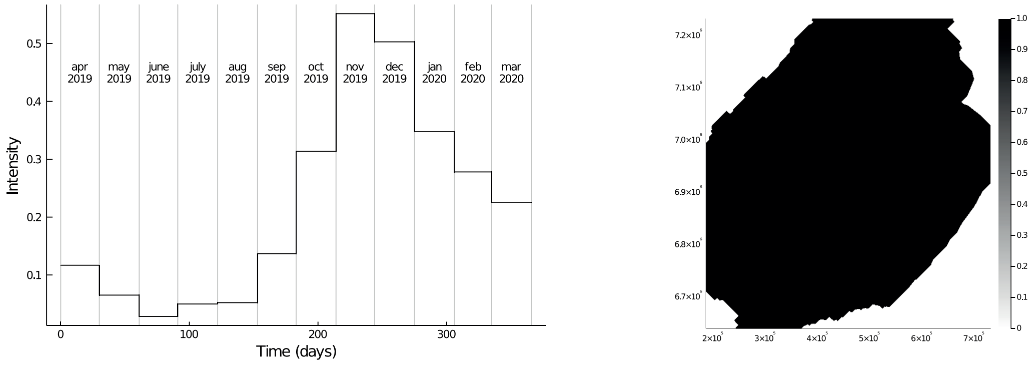
$$\begin{aligned} A[j] &\leq A[k] \text{ for } j = 1, \dots k - 1, \text{ and} \\ A[j] &> A[k] \text{ for } j = k + 1, \dots M. \end{aligned} \tag{34}$$

holds. In other words, Algorithm 2 splits the elements in $A$ to 'partitions' less than $A[k]$ and greater than $A[k]$. Note that the two partitions themselves are not necessarily in order, and that here it is assumed that the array $A$ is mutated 'in place'. The only difference in *random-partition* is that the 'pivot element' $x$ is chosen at random between $l$ and $u$ before calling *partition*.

With the help of Algorithms 2 and 3, pseudocode for computing the constant $c$ is as follows:

- Initialise $l = 1$, $u = M$.
- Loop while $l < u \mid\mid i \neq u$.
  - Partition the weights in $q$ by calling *random-partition*(q, l, u). The output is an index $i$. Set $\kappa = q[i]$. Note that (34) holds for the array $q$ and index $i$.
  - Check if the condition (32) is satisfied for $\kappa$. Note that $A_\kappa = M - i + 1$ and $B_\kappa = sum(q[1 : (i - 1)])$.
  - If (32) is satisfied, we know $\kappa_{\min} \leq \kappa$, so we set $u = i$; otherwise we set $l = i + 1$.
- If $l > u$, $I^{(M)} = \emptyset$; otherwise $\kappa_{\min}$ has been found and $I^{(M)} \neq \emptyset$. Use (33) to compute the constant $c$.

## 2. Additional figures



(a)

(b)

Figure 2. The temporal intensity function $\lambda_{\text{obs}}^{(\tau)}$ (a) and spatial intensity function $\lambda_{\text{obs}}^{(s)}$ (b) estimated for models 3 and 4 in Section 3.4 of the main text. The computation was done as discussed in Sections 2.3–2.4 of the main text, but with the terms $Corine_k$, $forestroad_k$ and $\text{smooth}(y_k)$ dropped from the intensity model (2). The domain of $\lambda_{\text{obs}}^{(s)}$ was additionally widened by 27 kilometres at the borders as described in Section 2.4 of the main text.

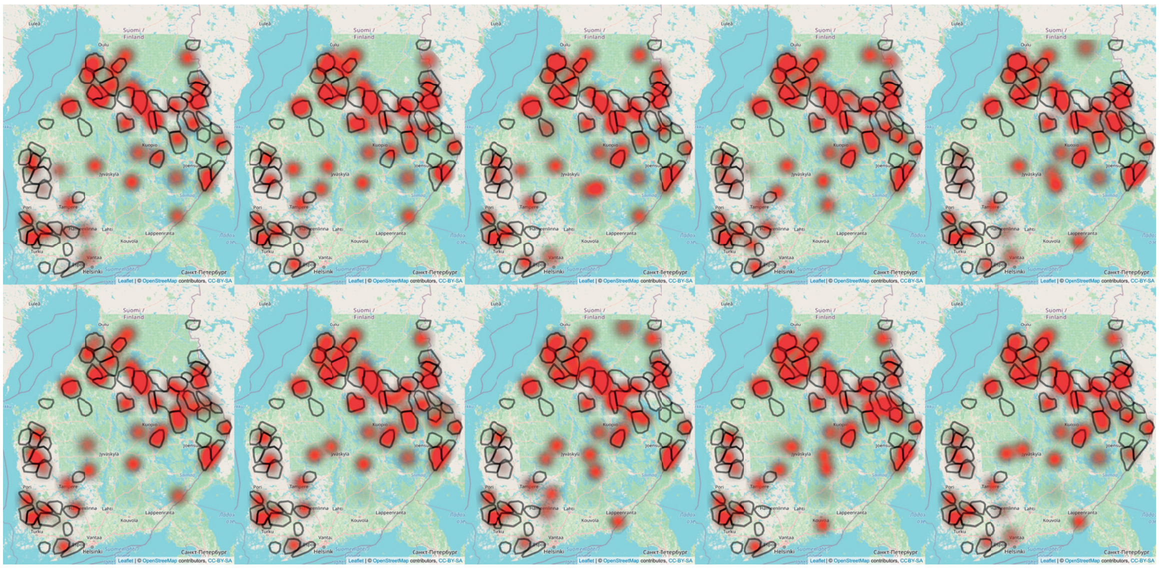


Figure 3. Estimated territory locations obtained by filtering Dataset C ten times with the filter configuration corresponding to model 2 in Section 3.4 of the main text.

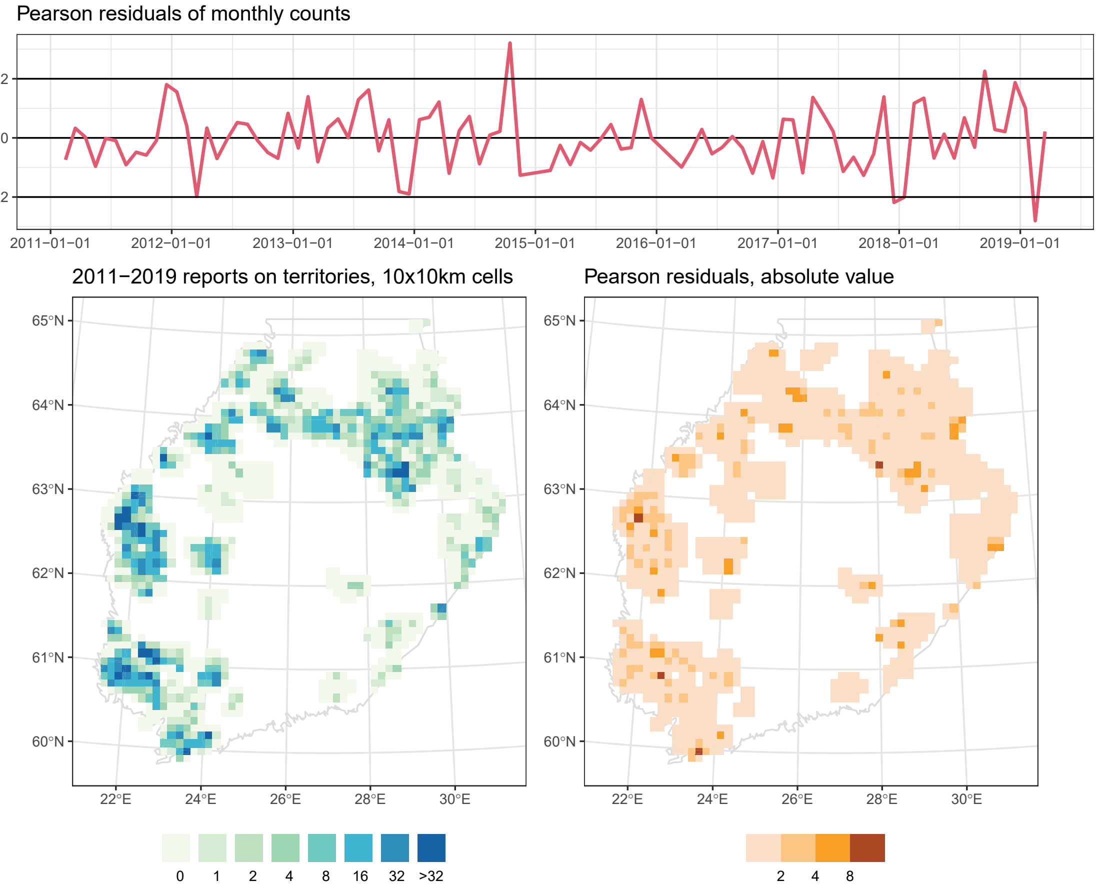


FIGURE 4. Fit diagnostics for the intensity model in Section 3.1 of the main text. *Top:* Pearson residuals of monthly counts. *Bottom:* Observed counts (left) and magnitudes of Pearson residuals of counts (right) in spatial cells of size 10x10km.

## REFERENCES

T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms.* MIT press, 2009.

A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo methods in practice.* Springer-Verlag New York, 2001. doi: 10.1007/978-1-4757-3437-9.

P. Fearnhead and P. Clifford. On-line inference for hidden Markov models via particle filters. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 65(4):887–899, 2003.

# IV

# CONDITIONAL PARTICLE FILTERS WITH BRIDGE BACKWARD SAMPLING

by

Karppinen, S., Singh, S.S., and Vihola, M. 2022

Preprint

# CONDITIONAL PARTICLE FILTERS WITH BRIDGE BACKWARD SAMPLING

SANTERI KARPPINEN, SUMEETPAL S. SINGH, MATTI VIHOLA

Abstract. The performance of the conditional particle filter (CPF) with backward sampling is often impressive even with long data records. Two known exceptions are when the observations are weakly informative and the dynamic model is slowly mixing. These are both present when sampling finely time-discretised continuous-time path integral models, but can occur with hidden Markov models too. Multinomial resampling, which is commonly employed in the (backward sampling) CPF, resamples excessively for weakly informative observations and thereby introduces extra variance. A slowly mixing dynamic model renders the backward sampling step ineffective. We detail two conditional resampling strategies suitable for the weakly informative regime: the so-called 'killing' resampling and the systematic resampling with mean partial order. To avoid the degeneracy issue of backward sampling, we introduce a generalisation that involves backward sampling with an auxiliary 'bridging' CPF step, which is parameterised by a blocking sequence. We present practical tuning strategies for choosing an appropriate blocking. Our experiments demonstrate that the CPF with a suitable resampling and the developed 'bridge backward sampling' can lead to substantial efficiency gains in the weakly informative regime.

## 1. Introduction

Conditional particle filter (CPF) with multinomial resampling and backward sampling (BS) [1, 33] can perform well with challenging state-space models and long data records [23]. However, when the observations are weakly informative, its multinomial resampling steps introduce excess noise, and when the dynamic model is slow mixing, its backward sampling step has only a limited effect. The aim of this paper is to devise a more effective CPF for such 'weak potentials' and slowly mixing scenarios, which arise for instance with time-discretisations of Feynman–Kac (FK) path integral models.

Motivated by successes of CPF with BS (hereafter CPF-BS) in the discrete time domain, we were interested to seek for a BS analogue which is stable with respect to refined time-discretisations. It is relatively easy to see that the direct application of BS degenerates under such refined discretisations, except for limited cases such as when the driving Markov process admits jumps, such as considered in [26], or in a univariate case where the trajectories can cross with positive probability. To address the inherent inefficiency of multinomial resampling, we draw inspiration from the recent works of [2] and [5] where other types of resampling are shown to be more effective for weakly informative potentials. Arnaudon and Del Moral [2] propose a continuous-time version of the CPF with 'killing' resampling, however, this is an idealised algorithm in the sense that practical diffusion models need to be time-discretised. The work of [5] studies the stability of resampling for particle filters (and not the CPF) as the time discretisation is refined. There, a new systematic resampling method is proposed, which incorporates a 'mean partition' step, which has not been developed for the CPF yet. The mean partition step does further decrease superfluous resampling [5], and thus reduces the so-called particle degeneracy.

The main contributions of this paper are as follows.

- We detail two new conditional resampling algorithms: the ‘killing’ and systematic resampling with mean partition (Section 5). These are conditional versions of resampling algorithms, which are stable in the weak potentials setting (in the continuous-time limit) [5]. We also detail a generic sufficient condition for conditional resamplings (Assumption 7), which guarantees validity of the CPF (Theorem 2), and complements the result of [4].
- We introduce a new CPF with bridge backward sampling (CPF-BBS) (Section 6), which may be regarded as a generalisation of BS to an arbitrary ‘blocking sequence,’ and which can avoid the degeneracy problem of CPF-BS with refined discretisations.
- The performance of the CPF-BBS relies on an appropriately chosen blocking sequence, which depends on the model at hand. Therefore, a significant portion of our work focuses on finding practical, computationally inexpensive and robust tuning criteria for choosing such a sequence (Section 7). We introduce a method for blocking sequence selection that requires a small number of independent runs of the standard particle filter for the model of interest.

Our developments related to blocking sequence selection can be of independent interest, and potentially useful with other methods based on blocking, such as the blocked particle Gibbs [29]. Systematic resampling in the context of CPF has been proposed before [4] but not the more efficient mean partition version. Furthermore, [4] does not discuss or demonstrate efficiency in the context of weak potentials, which is our primary motivation.

The CPF-BBS is a general method, but requires evaluation of and simulation from the conditional distributions of (multiple steps of the) proposal distributions. In practice, this typically means that the proposals are linear-Gaussian, arising for instance from a linear stochastic differential equation (SDE). The latter can occur in single molecule studies [7], and one of our numerical examples demonstrates how animal movement modelling based on telemetry data [21] can be combined with a path integral model for the so-called step-selection analyses [cf. 30]. Linear-Gaussian state dynamics are common with structural time series models [14] too, and smoothing distribution approximations can lead to weak potentials [cf. 32].

The CPF-BBS features ‘bridging’ CPF steps, which resemble the intermediate block importance sampling suggested in [24], and the MCMC rejuvenation considered in [24, 3]; there are similarities also with the bridging particle filter suggested in [12]; see also [27]. We believe that our approach is more efficient than direct importance bridging, and because our approach can be intuitively related to a continuous-time analogue (through [5]), it is expected to behave well with respect to refinement of time-discretisation, unlike the MCMC bridging.

Our experiments (Section 9) demonstrate how the developed resamplings outperform standard multinomial resampling in the weak potential setting, and we establish empirically an order between their performance, which follows a similar pattern as the results of [5] for the standard particle filter applied to FK path integral models. Empirical results using the CPF-BBS show a significant improvement over CPF-BS in the weak potential setting, and reveal how the method is stable with respect to refined discretisation. Finally, our tuning algorithm appears to deliver blocking sequences that reach near-optimal performance with little additional specification from the user.

## 2. Preliminaries and notation

We aim at inference of a probability density on $\mathsf{X}^T$ with the following form:

$$\pi(x_{1:T}) = \frac{\eta(x_{1:T})}{\mathcal{Z}}, \quad \text{where} \quad \eta(x_{1:T}) := M_1(x_1)G_1(x_1)\prod_{k=2}^{T} M_k(x_k \mid x_{k-1})G_k(x_{k-1:k}). \tag{1}$$

The model above, defined in terms of $M_{1:T}$ and $G_{1:T}$, is often referred to as a Feynman-Kac (FK) model [10]. Here, $M_1(x_1)$ and $M_k(x_k \mid x_{k-1})$ define, respectively, an initial distribution and 'proposal' transition densities of a Markov chain on $\mathsf{X}$, and $G_1 : \mathsf{X} \to [0,\infty)$ and $G_k : \mathsf{X}^2 \to [0,\infty)$ for $k \geq 2$ are called 'potential' or 'weight' functions (see the discussion below). The probability $\pi$ is well-defined assuming that the normalising constant $\mathcal{Z} := \int \eta(x_{1:T})\mathrm{d}x_{1:T} \in (0,\infty)$.

Above, and hereafter, 'd$x$' stands for a $\sigma$-finite dominating measure on $\mathsf{X}$, integers are equipped with the counting measure, and product spaces are equipped with products of the dominating measures. We use the shorthand notation for sequences: for $\{x_i\}_i$, $\{y^j\}_j$ and $\{z_i^j\}_{i,j}$ we write $x_{a:b} = (x_a,\ldots,x_b)$, $y^{(a:b)} = (y^{(a)},\ldots,y^{(b)})$ and $z_{a:b}^{(j_{a:b})} = (z_a^{(j_a)},\ldots,z_b^{(j_b)})$. We also denote $[N] := \{1,\ldots,N\}$. Test and potential functions are implicitly assumed measurable.

The FK model can be seen as a slight generalisation of the hidden Markov model (HMM), which has a latent Markov state $X_{1:T}$ with initial (prior) density $m_1(x_1)$ and transition probability densities $m_k(x_k \mid x_{k-1})$, and conditional independent observations $y_{1:T}^*$ with observation densities $g_k(y_k \mid x_k)$. The posterior (or smoothing) distribution of $X_{1:T} \mid y_{1:T}^*$ is of form (1) if we choose $M_k \equiv m_k$ and $G_k(\,\cdot\,) \equiv g_k(y_k^* \mid \cdot\,)$. However, it is often beneficial to choose another 'proposal' family $M_k \not\equiv m_k$, in which case the 'weights' are $G_1(x_1) := g_1(y_k^* \mid x_1)m_1(x_1)/M_1(x_1)$ and $G_k(x_{k-1},x_k) := g_k(y_k^* \mid x_k)m_k(x_k \mid x_{k-1})/M_k(x_k \mid x_{k-1})$ for $k \geq 2$.

## 3. The particle filter

The particle filter is a sequential Monte Carlo algorithm, which includes sampling from Markov dynamics $M_k$, and *resampling* proportional to weights arising from $G_k$. The resampling operation $r(a^{(1:N)} \mid g^{(1:N)})$ defines a probability distribution on $[N]^N$ which depends on non-negative 'unnormalised weights' $g^{(1:N)}$. That is, if $A^{(1:N)} \sim r(\,\cdot \mid g^{(1:N)})$ in $[N]$, then $\mathbb{P}(A^{(1:N)} = a^{(1:N)}) = r(a^{(1:N)} \mid g^{(1:N)})$. We will only consider *unbiased resamplings* [6] $r$, which means that for all $j \in [N]$:

$$\bigg(\sum_{i=1}^N g^{(i)}\bigg)\mathbb{E}_{r(\,\cdot\,\mid g^{(1:N)}))}\bigg[\frac{1}{N}\sum_{i=1}^N \mathbf{1}\left(A^{(i)} = j\right)\bigg] = g^{(j)}. \tag{2}$$

Algorithm 1 describes the particle filter targetting the FK model (1) using $N$ particles, and an unbiased resampling $r(\,\cdot \mid g^{(1:N)}))$. The boldface notation $\mathbf{X}_1^{(i)} = X_1^{(i)}$ and $\mathbf{X}_{k+1}^{(i)} = (X_k^{(A_k^{(i)})}, X_{k+1}^{(i)})$ stands for the latest particles augmented with their ancestors, generated during the algorithm. In what follows, we use underline to denote 'all particles' at one time instant, so for instance $\underline{X}_k = X_k^{(1:N)}$.

Consider then the following density on $\mathsf{X}^{NT} \times [N]^{N(T-1)}$ which corresponds to all the random variables generated during Algorithm 1:

$$\zeta^{(N)}(\underline{x}_{1:T},\underline{a}_{1:T-1}) = \prod_{i=1}^N M_1(x_1^{(i)})\prod_{k=1}^{T-1}\bigg(r\big(\underline{a}_k \mid G_k(\underline{\boldsymbol{x}}_k)\big)\prod_{i=1}^N M_{k+1}(x_{k+1}^{(i)} \mid x_k^{(a_k^{(i)})})\bigg). \tag{3}$$

**Algorithm 1** PF$(r, M_{1:T}, G_{1:T}, N)$

```
Draw $X_1^{(i)} \sim M_1(\cdot)$ for $i \in [N]$.
Set $\mathbf{X}_1^{(i)} = X_1^{(i)}$ for $i \in [N]$.
for $k = 1, \dots, T-1$ do
    Set $W_k^{(i)} = G_k(\mathbf{X}_k^{(i)})$ for $i \in [N]$.
    Draw $A_k^{(1:N)} \sim r(\cdot \mid W_k^{(1:N)})$
    Draw $X_{k+1}^{(i)} \sim M_{k+1}(\cdot \mid X_k^{(A_k^{(i)})})$ for $i \in [N]$.
    Set $\mathbf{X}_{k+1}^{(i)} = (X_k^{(A_k^{(i)})}, X_{k+1}^{(i)})$ for $i \in [N]$.
end for
Set $W_T^{(i)} = G_T(\mathbf{X}_T^{(i)})$ for $i \in [N]$.
output $(X_{1:T}^{(1:N)}, A_{1:T-1}^{(1:N)}, W_{1:T}^{(1:N)})$
```

The normalising constant estimate calculated from $\underline{X}_{1:T}$ and $\underline{A}_{1:T-1}$ in the output of Algorithm 1, is defined as follows:

$$\hat{\mathcal{Z}}(\underline{x}_{1:T}, \underline{a}_{1:T-1}) = \prod_{k=1}^{T} \left( \frac{1}{N} \sum_{i=1}^{N} G_k(\mathbf{x}_k^{(i)}) \right). \tag{4}$$

Thanks to the unbiasedness of the resampling (2), the following 'unbiasedness property' [cf. 10] holds, which is key for the validity of particle Markov chain Monte Carlo [1]:

$$\mathbb{E}[\hat{\mathcal{Z}}(\underline{X}_{1:T}, \underline{A}_{1:T-1}) f(X_{1:T}^*)] = \mathcal{Z}\mathbb{E}_\pi[f(X_{1:T})]. \tag{5}$$

Equation (5) holds for any test function $f : \mathsf{X}^T \to \mathbb{R}$ for which the expectation on the right is well-defined, as long as the trajectory $X_{1:T}^* = X_{1:T}^{(B_{1:T})}$ is chosen among all particles in a suitable manner, that is, with suitably generated indices $B_{1:T}$.

The most direct approach is to draw $B_T \sim \mathrm{Categ}(\omega_T^{(1:N)})$ with (unnormalised weights) $\omega_T^{(i)} = G_T(\mathbf{X}_T^{(i)})$, and to set the rest of the indices recursively by '*ancestor tracing*': $B_k := A_k^{(B_{k+1})}$. In our case, where the potentials depend on at most two consecutive states, it is possible to replace ancestor tracing by '*backward sampling*': for $k = T-1, \dots, 1$:

$$B_k \sim \mathrm{Categ}\big(\omega_k^{(1:N)}\big), \quad \omega_k^{(i)} := G_k(\boldsymbol{X}_k^{(i)}) G_{k+1}(X_k^{(i)}, X_{k+1}^{(B_{k+1})}) M_{k+1}(X_{k+1}^{(B_{k+1})} \mid X_k^{(i)}). \tag{6}$$

The unbiasedness (5) was shown in [9] for ancestor tracing and multinomial resampling, and has been extended for other unbiased resamplings [e.g. 1]. For the context of the present work, we refer the reader to [31, Appendix D] for a proof of the unbiasedness in (5) assuming only unbiasedness of resampling (2), and which accomodates backward sampling with any resampling.

## 4. The conditional particle filter

The CPF introduced in [1] implements a $\pi$-invariant Markov transition kernel from an input so called 'reference' path $X_{1:T}^* \in \mathsf{X}^T$ to a newly chosen path $\tilde{X}_{1:T}^* \in \mathsf{X}^T$. The original scheme of [1] assumed multinomial sampling with ancestor tracing, and [33] suggested, in a discussion note to [1], that backward sampling may also be used (with multinomial resampling); later an algorithmic variant of BS called 'ancestor sampling' (AS) [25] was also introduced. In fact, the corresponding Markov kernels are reversible with respect to $\pi$, and BS/AS is guaranteed to outperform AT in the asymptotic variance sense [4]. The improvement has been found substantial in many empirical studies; see also [23] for a theoretical result supporting such findings. Finally, other resampling schemes are presented in [4], but not the ones devised in this work.

Algorithm 2 presents a generic version of the CPF with $N$ particles and with ancestor tracing, using a generic conditional resampling $r^{(p,n)}$. The conditional resampling scheme draws the ancestor indices (on line 3 of Algorithm 3) conditional on the ancestor of the reference. This makes it possible to write Algorithm 2 such that the reference trajectory can be located at arbitrary indices $B_{1:T}$, unlike earlier formulations, which assume reference at index 1 [e.g. 4]. The arbitrary reference indices turn out to be convenient for us, when we introduce the bridge backward sampling CPF in Section 6. Definition 1 gives a sufficient condition that $r^{(p,n)}$ is a valid conditional resampling for use with Algorithm 2.

**Algorithm 2** CPF-AT($r^{(p,n)}, X^*_{1:T}, B_{1:T}; M_{1:T}, G_{1:T}, N$)

```
$(\underline{X}_{1:T}, \underline{A}_{1:T-1}, \tilde{B}_T) \leftarrow$ CPF($r^{(p,n)}, X^*_{1:T}, B_{1:T}; M_{1:T}, G_{1:T}, N$)
$\tilde{B}_{1:T-1} \leftarrow$ AncestorTrace($\underline{A}_{1:T-1}, \tilde{B}_T$).
output $(\tilde{X}^*_{1:T}, \tilde{B}_{1:T})$ where $\tilde{X}^*_{1:T} = X_{1:T}^{(\tilde{B}_{1:T})}$.
```

**Algorithm 3** CPF($r^{(p,n)}, X^*_{1:T}, B_{1:T}; M_{1:T}, G_{1:T}, N$)

```
1: Draw $X_1^{(-B_1)} \sim M_1(\,\cdot\,)$ and set $X_1^{(B_1)} \leftarrow X_1^*$ and $\mathbf{X}_1^{(i)} = X_1^{(i)}$ for $i \in [N]$.
2: for $k = 1, \ldots, T-1$ do
3:     $A_k^{(1:N)} \leftarrow r^{(B_k, B_{k+1})}\big(\,\cdot \mid G_k(\boldsymbol{X}_k^{(1:N)})\big)$
4:     Draw $X_{k+1}^{(i)} \sim M_{k+1}(\,\cdot \mid X_k^{(A_k^{(i)})})$ for $i \neq B_{k+1}$ and set $X_{k+1}^{(B_{k+1})} = X^*_{k+1}$.
5:     Set $\mathbf{X}_{k+1}^{(i)} = (X_k^{(A_k^{(i)})}, X_{k+1}^{(i)})$ for $i \in [N]$.
6: end for
7: Draw $\tilde{B}_T \sim \mathrm{Categ}\big(G_T(\boldsymbol{X}_T^{(1:N)})\big)$.
8: output $(\underline{X}_{1:T}, \underline{A}_{1:T-1}, \tilde{B}_T)$
```

**Algorithm 4** AncestorTrace($\underline{a}_{\ell:u-1}, b_u$)

```
for $v = u-1, u-2, \ldots, \ell$ do $b_v \leftarrow a_v^{(b_{v+1})}$
output $b_{\ell:u-1}$.
```

**Definition 1.** The conditional resampling scheme $r^{(p,n)}(\,\cdot \mid g^{(1:N)})$ is valid, if it is a conditional of an unconditional unbiased symmetric resampling scheme $r(\,\cdot \mid g^{(1:N)})$. That is, for all $g^{(1:N)} \geq 0$ such that $\sum_{\ell=1}^N g^{(\ell)} > 0$, and all $p, n \in \{1{:}N\}$,

(i) $\mathbb{P}_{r^{(p,n)}(\,\cdot\,|g^{(1:N)})}(A^{(n)} = p) = 1$,

(ii) $r^{(p,n)}(a^{(1:n)} \mid g^{(1:N)}) = \mathbb{P}_{r(\,\cdot\,|g^{(1:N)})}(A^{(-n)} = a^{(-n)} \mid A^{(n)} = p)$,

(iii) $\mathbb{P}_{r(\,\cdot\,|g^{(1:N)})}(A^{(n)} = p) = \frac{g^{(p)}}{\sum_{i=1}^N g^{(i)}}$.

**Theorem 2.** *Algorithm 2 with a valid conditional resampling $r^{(p,n)}$ defines a Markov update $(X^*_{1:T}, B_{1:T}) \to (\tilde{X}^*_{1:T}, \tilde{B}_{1:T})$ that is reversible with respect to $\pi \times U([N]^T)$.*

Theorem 2, whose proof is given in Appendix A, complements the result of [4] by accomodating our version of the CPF, where the reference is placed at arbitrary position, and allows for the resamplings which we discuss next.

## 5. Conditional resamplings for the weak potentials scenario

The simplest unbiased resampling, that is, satisfying (2) is *multinomial resampling*, where $A^{(k)}$ are drawn independently from the categorical distribution $\mathrm{Categ}(w^{(1:N)})$ with normalised weights $w^{(j)} = \frac{g^{(j)}}{\sum_{i=1}^N g^{(i)}}$. However, in the context of this work, multinomial resampling is wasteful, and we focus instead on conditional versions of two resampling algorithms, that were found stable in refined discretisations [5].

The first of these is the 'killing' resampling, defined as follows [cf. 11]:

$$r_{\text{kill}}(a^{(1:N)} \mid g^{(1:N)}) := \prod_{i=1}^N \left[ \mathbf{1}\left(a^{(i)} = i\right) \frac{g^{(i)}}{g^*} + \left(1 - \frac{g^{(i)}}{g^*}\right) \sum_{j=1}^N \mathbf{1}\left(a^{(i)} = j\right) \frac{g^{(j)}}{\sum_{\ell=1}^N g^{(\ell)}} \right], \tag{7}$$

where $g^* = \max_{i\in\{1:N\}} g^{(i)}$ (and in case $g^* = 0$, $\rho$ may be defined arbitrarily). The killing resampling is valid also with any other choice of $g^*$ as long as $g^{(j)} \le g^*$, but we consider the above one minimising the resampling rate. Like multinomial resampling, the components of the random vector $A^{(1:N)}$ are independent but not identically distributed.

Killing resampling is simple and stable, but [5] found two resamplings that yield a smaller resampling rate, and appear to admit slightly better performance: the Srinivasan sampling process (SSP) and systematic resampling, both with a mean partition order (to be defined below). It seems difficult to implement a conditional version of SSP (in an efficient manner), but systematic resampling with mean partition can be implemented by extending the algorithm of [4].

The systematic resampling with mean partition (Definition 5) is a variant of 'standard' systematic resampling (Definition 3) where the weights $w^{1:N}$ are processed in a particular 'mean partition' order (Definition 4). The mean partition may be found in $O(N)$ time, and our implementation is based on Hoare's scheme [20]; see Algorithm 12 in Appendix E.

**Definition 3.** (Systematic resampling). Input normalised weights $w^{1:N}$. Simulate a single $\tilde{U} \sim U(0,1)$, set $\check{U}^i := (i-1+\tilde{U})/N$ and define the resampling indices as $A^i := F^{-1}(\check{U}^i)$ for $i \in [N]$. Here, the generalised inverse $F^{-1}(u)$ is defined for $u \in (0,1)$ as the unique index $i \in [N]$ such that $F(i-1) < u \le F(i)$, with $F(i) := \sum_{j=1}^i w^j$.

**Definition 4.** (Mean partition order) Suppose that $u^{1:N} \in \mathbb{R}^N$. A permutation $\varpi : [N] \to [N]$ is a *mean partition* order for $u^{1:N}$, if the re-indexed vector $u_\varpi^i := u^{\varpi(i)}$ satisfies $u_\varpi^1, \ldots, u_\varpi^m \le \bar{u}$ and $u_\varpi^{m+1}, \ldots, u_\varpi^N > \bar{u}$ for some $m \in [N]$, with $\bar{u}$ denoting the mean of the vector $u$.

**Definition 5.** (Systematic resampling with mean partition). Let $F_\varpi^{-1}$ denote the generalised inverse distribution function corresponding to the re-indexed weights $w_\varpi^{1:N}$, where $\varpi$ is a mean partition order as in Definition 4. Set $A^{\varpi(i)} := \varpi(F_\varpi^{-1}(\check{U}^i))$, where $\check{U}^{1:N}$ are defined as in Definition 3.

Algorithms 5 and 6 describe the conditional variants of killing resampling and systematic resampling with mean partition, respectively.

**Lemma 6.** *The following define valid conditional resamplings (Definition 1):*

*(i) conditional killing $\rho_{\text{kill}}^{(i,k)}(\,\cdot \mid g^{(1:N)})$ of Algorithm 5, and*

*(ii) conditional systematic resampling with mean partition $\rho_{\text{syst}}^{(i,k)}$ of Algorithm 6.*

Proof of Lemma 6 is given in Appendix A.

**Algorithm 5** Conditional killing resampling $\rho_{\text{kill}}^{(i,k)}(\,\cdot \mid g^{(1:N)})$.

```
Draw $\bar{A}^{(1:N)} \sim \rho_{\text{kill}}(\,\cdot \mid g^{(1:N)})$.
Draw $J \in [N]$ such that $\mathbb{P}(J = j) = h(j \mid i) := \begin{cases} \frac{1}{N}\Big(1 + \frac{\sum_{\ell \neq i} g^{(\ell)}}{g^*}\Big), & j = i \\ \frac{1}{N}\Big(1 - \frac{g^{(j)}}{g^*}\Big), & j \neq i. \end{cases}$
Set $S := [\![J - k]\!]_N$, where $[\![\ell]\!]_N := 1 + (\ell - 1 \mod N)$.
Set $\bar{A}^{([\![k+S]\!]_N)} \leftarrow i$.
output $A^{(1:N)}$ where $A^{(j)} = \bar{A}^{([\![j+S]\!]_N)}$
```

**Algorithm 6** Conditional systematic resampling with mean partition $\rho_{\text{syst}}^{(i,k)}(\,\cdot \mid g^{(1:N)})$.

```
For $j \in [N]$, define $W^j := \dfrac{g^{(j)}}{\sum_{i=1}^N g^{(i)}}$ and $U^j := \dfrac{j - 1 + U}{N}$, where $U \sim U(0,1)$.
Set $r = NW^i - \lfloor NW^i \rfloor$ and $p = \dfrac{r(\lfloor NW^i \rfloor + 1)}{NW^i}$.
With probability $p$, draw $\bar{U} \sim U(0, r)$ and set $N^i = \lfloor NW^i \rfloor + 1$; otherwise draw $\bar{U} \sim U(r, 1)$
    and set $N^i = \lfloor NW^i \rfloor$.
Set $\varpi \leftarrow \textsc{MeanPartitionOrder}(W^{1:N})$          ▷ Algorithm 12 in Appendix E
Set $s = \varpi^{-1}(i)$ and $\tilde{\varpi} = \sigma_{1-s}(\varpi)$, so that $\tilde{\varpi}(1) = \varpi(s) = i$.
Draw $\bar{A}^{\tilde{\varpi}(1:N)} = \tilde{\varpi}(F_{\tilde{\varpi}}^{-1}(U^{[N]}))$.
Draw $C \sim U([N])$ and set $A^j = \bar{A}^{\sigma_{k-C}(j)}$, for $j \in [N]$.
output $A^{1:N}$.
```

## 6. The conditional particle filter with bridge backward sampling

The backward/ancestor sampling CPF [33, 25] often has impressive performance even with large $T$ [23]. The selection probabilities in the backward sampling step (6) include the transition density $M_{k+1}(X_{k+1}^{(B_{k+1})} \mid X_k^{(i)})$. When $M_{k+1}$ is slowly mixing, this density is typically very small for all $i$ except for the ancestor $i = A_k^{(B_{k+1})}$, and therefore the backward sampling step reduces to ancestor tracing.

We discuss next the conditional particle filter with bridge backward sampling (CPF-BBS), which is a generalisation of CPF with backward sampling (CPF-BS) [33] suitable for slowly mixing $M_k$. The backward sampling step is replaced by a 'bridging' procedure which spans over multiple time steps, and requires tractable dynamics $\{M_k\}$, in the following sense:

**Assumption 7.** Denote $M_{\ell:u}(x_{\ell:u}) := \prod_{k=\ell+1}^{u} M_k(x_k \mid x_{k-1})$ for any $2 \leq \ell < u \leq T$. Then, we are able to simulate from and evaluate the density of the conditional distribution of $x_\ell$ given $x_{\ell-1}$ and $x_u$:

$$\bar{M}_\ell(x_\ell \mid x_{\ell-1}, x_u) := \frac{\int M_{\ell-1:u}(x_{\ell-1:u}) \mathrm{d}x_{\ell+1:u-1}}{M_{u|\ell-1}(x_u \mid x_{\ell-1})}.$$

We further assume that we are able to evaluate the conditional density of $x_u$ given $x_\ell$:

$$M_{u|\ell}(x_u \mid x_\ell) := \int M_{\ell:u}(x_\ell, z_{\ell+1:u-1}, x_u) \mathrm{d}z_{\ell+1:u-1}.$$

Algorithm 7 gives the pseudocode of the CPF-BBS algorithm.

The first step (line 1) invokes the forward CPF (Algorithm 3). To facilitate the bridging procedure (line 5) that replaces the usual backward sampling step in CPF-BS, a fixed 'blocking sequence' $1 = T_1 < \cdots < T_L = T$ is utilised that gives rise to the blocks $(T_{i-1},\ T_i)$, $i = 2, \ldots, L$, where $T_{i-1}$ and $T_i$ are referred to as the block lower and upper

**Algorithm 7** CPF-BBS($X^*_{1:T}, B_{1:T}; T_{1:L}$)

```
1: $(\underline{X}_{1:T}, \underline{A}_{1:T-1}, \tilde{B}_T) \leftarrow \mathrm{CPF}(X^*_{1:T}, B_{1:T})$ and set $\tilde{X}^*_T \leftarrow X_T^{(\tilde{B}_T)}$
2: for $k = L, L-1, \ldots, 2$ do
3:     $\ell \leftarrow T_{k-1}$; $u \leftarrow T_k$; $B^*_u \leftarrow \tilde{B}_u$.
4:     $B^*_{\ell:u-1} \leftarrow \textsc{AncestorTrace}(\underline{A}_{\ell:u-1}, B^*_u)$
5:     $(\tilde{X}^*_{\ell:u-1}, \tilde{B}_{\ell:u-1}) \leftarrow \textsc{BridgeCPF}(\underline{\boldsymbol{X}}_\ell, B^*_{\ell:u-1}, X_{\ell+1:u}^{(B^*_{\ell+1:u})})$
6: end for
7: output $(\tilde{X}^*_{1:T}, \tilde{B}_{1:T})$
```

boundaries, respectively. Algorithm 8 is invoked (line 5) to attempt to change the ancestor of state $X_u^{(B^*_u)}$ at time $l$ from $B^*_l$ to a different particle from the pool $\underline{\boldsymbol{X}}_l$. Success of this step hinges on Algorithm 8 being able to generate particles that could equally well explain the future state $X_u^{(B^*_u)}$ being conditioned on (see line 8 of Algorithm 8), for which the conditional densities of Assumption 7 are needed in its forward simulation procedure (lines 1–7). The new ancestor from the pool $\underline{\boldsymbol{X}}_l$ is then found by ancestral tracing (line 9). Success in this step relies on an efficient resampling strategy (line 3) to avoid particle degeneracy so that many particles from $\underline{\boldsymbol{X}}_l$ survive to line 8. Clearly the choice of the blocking sequence is also important and for this a practical design choice procedure is devised in Section 7.1.

**Algorithm 8** BridgeCPF($\underline{\boldsymbol{x}}_\ell, b^*_{\ell:u-1}, x^*_{\ell+1:u}$)

```
1: $W_\ell^{(1:N)} \leftarrow M_{u|\ell}(x^*_u \mid \underline{x}_\ell)^{\frac{1}{u-\ell}}$; $\underline{\tilde{\boldsymbol{X}}}_\ell \leftarrow \underline{\boldsymbol{x}}_\ell$
2: for $v = \ell+1 : u-1$ do
3:     $\tilde{A}_{v-1}^{(1:N)} \leftarrow r^{(b^*_{v-1}, b^*_v)}(\,\cdot\, \mid G_{v-1}(\tilde{\boldsymbol{X}}_{v-1}^{(1:N)}) W_{v-1}^{(1:N)})$
4:     Draw $\tilde{X}_v^{(i)} \sim \bar{M}_v(\,\cdot \mid \tilde{X}_{v-1}^{(\tilde{A}_{v-1}^{(i)})}, \tilde{X}^*_u)$ for $i \neq b^*_v$ and set $\tilde{X}_v^{(b^*_v)} = x^*_v$
5:     Set $\tilde{\boldsymbol{X}}_v^{(i)} \leftarrow (X_{v-1}^{(\tilde{A}_{v-1}^{(i)})}, \tilde{X}_v^{(i)})$ for $i \in [N]$.
6:     $W_v^{(1:N)} \leftarrow W_{v-1}^{(\tilde{A}_{v-1}^{(1:N)})}$
7: end for
8: Draw $\tilde{B}_{u-1} \sim \mathrm{Categ}(\tilde{\omega}_{u-1}^{(1:N)})$ where $\tilde{\omega}_{u-1}^{(j)} = G_{u-1}(\tilde{\boldsymbol{X}}_{u-1}^{(j)}) G_u(\tilde{X}_{u-1}^{(j)}, x^*_u) W_{u-1}^{(j)}$
9: $\tilde{B}_{\ell:u-2} \leftarrow \textsc{AncestorTrace}(\underline{\tilde{A}}_{\ell:u-2}, \tilde{B}_{u-1})$
10: output $\big((x_\ell^{(\tilde{B}_\ell)}, \tilde{X}_{\ell+1:u-1}^{(\tilde{B}_{\ell+1:u-1})}), \tilde{B}_{\ell:u-1}\big)$
```

We record the following consistency result, ensuring the CPF-BBS is valid, whose proof is given in Appendix B:

**Theorem 8.** *Consider Algorithm 7 as a Markov update* $(X^*_{1:T}, B_{1:T}) \to (\tilde{X}_{1:T}, \tilde{B}_{1:T})$. *Then, it leaves* $\pi \times U([N]^T)$ *invariant.*

With dense blocking sequence $T_{1:T} = 1{:}T$, the bridging CPF and its tracing (lines 2–7 and 9 of Algorithm 8, respectively) are eliminated, and therefore the CPF-BBS simplifies to the backward sampling CPF (CPF-BS) of [33]. This means that the CPF-BBS can be viewed as a true generalisation of CPF-BS for arbitrary blockings.

The other extreme case, that is, the trivial blocking sequence $T_1 = 1$, $T_2 = T$ leads to running a CPF and then *another* CPF with same initial particles and targeting the conditional distribution $\pi(x_{1:T-1} \mid \tilde{X}^*_T)$ (cf. Lemma 12). This may not be practically useful, but can give insight about what the ‘bridge CPF’ is about.

We conclude this section with two remarks about methods related to Algorithm 7.

(i) If we modify the algorithm by replacing BridgeCPF by the following algorithm:

```
1: Set $\tilde{X}_k^{(B_\ell^*)} = X_k^{(B_k^*)}$ for $k = \ell{:}(u-1)$.
2: For $i \neq B_\ell^*$, set $\tilde{X}_\ell^{(i)} = X_\ell^{(i)}$ and $\tilde{X}_k^{(i)} \sim \bar{M}_v(\,\cdot \mid \tilde{X}_{k-1}^{(i)}, X_u^{(B_u^*)})$ for $k = (\ell+1){:}(u-1)$.
3: Choose $\tilde{X}_{\ell:u-1}^{(i)}$ with probability proportional to $M_{u|\ell}(\tilde{X}_u^* \mid \tilde{X}_\ell^{(i)}) \prod_{v=\ell}^{u-1} G_v(\tilde{X}_v^{(i)})$,
```

then we get a CPF version of the extended importance sampling for particle filters suggested in [13].

(ii) Algorithm 7 has similarities with the blocked particle Gibbs (or blocked CPF) of [29] but differs in two crucial points:

- We suggest a block-wide 'lookahead' which is possible to implement thanks to Assumption 7, instead of using a modified potential only at the last time instant.
- The block update is not conditioned on a single start point, but all particles which were generated by the 'forward' CPF. (Algorithm 3).

While these differences may seem technical, they can have substantial effect on the efficiency of the method. We believe that CPF-BBS often leads to more efficient algorithm in the same computational complexity, but note that the blocked particle Gibbs is directly parallelisable unlike the CPF-BBS.

Finally, we note that the reverse update order of the blocks occurs since a block's update depends on the value at the lower boundary of the subsequent block. Although not pursued here, it is possible that a forward only implementation (following [25]) could be devised to achieve a similarly better mixing CPF algorithm.

## 7. Blocking sequence selection

The CPF-BBS (Algorithm 7) is valid with any choice of the blocking sequence $T_{1:L}$. However, its choice affects simulation efficiency, that is, the mixing of the Markov chain. In this section, we discuss a computationally inexpensive method that can be used in practice to determine a suitable blocking sequence prior to running the CPF-BBS in order to facilitate efficient mixing.

We begin in Section 7.1 by discussing a proxy for the integrated autocorrelation time (IACT) of the Markov chain output by the CPF-BBS. Then, Section 7.2 details an estimator we have developed for the proxy. Finally, Section 7.3 describes a practical algorithm for blocking sequence selection that is based on the estimator of Section 7.2. We will study the methods presented in this section empirically in Section 9.

### 7.1. The probability of lower boundary updates (PLU).

A theoretically attractive candidate strategy for blocking sequence selection is monitoring the IACT for variables of interest, based on the output of the CPF-BBS. Efficient inference could then be obtained by choosing the blocking sequence that minimises the IACT. However, this approach is typically computationally demanding or even infeasible, since the estimation of the IACT is notoriously difficult and often requires extensive simulation of Markov chains.

For these reasons, we base the selection of the blocking sequence on a proxy for IACT that is easier to work with. We call the proxy the 'probability of lower boundary updates' (PLU), and its definition for the block $(\ell, u)$, using the notation of Algorithm 8, is:

$$\mathrm{PLU}(\ell, u) := \mathbb{P}(X_\ell^{(\tilde{B}_\ell)} \neq X_\ell^{(b_\ell^*)}). \tag{8}$$

In other words, $\mathrm{PLU}(\ell, u)$ measures the probability that the bridge CPF (Algorithm 8) on block $(\ell, u)$ updates the value at the block lower boundary $\ell$. Intuitively, higher values of PLU should be associated with lower IACT, and our experiments in Section 9 support this.

7.2. **Approximate estimator for the PLU.** Even though $\mathrm{PLU}(\ell, u)$ is much easier to estimate than IACT, it still requires iterating the CPF-BBS for each candidate blocking, which is computationally demanding. We have developed an estimator for $\mathrm{PLU}(\ell, u)$ which avoids this, and is based on a single 'stationary' CPF state (the generated particles $X_{1:T}^{(1:N}$ and reference indices $B_{1:T}$), which is used for any block boundaries $\ell, u$. The practical algorithm postponed to Section 7.3 will be based on this idea, but assumes further that such a stationary CPF state can be well approximated by an independent particle filter.

The estimator from a single CPF state is presented in (12) below and is based on two 'asymptotic' characterisations for PLU, for small and large blocksizes, respectively. The idea behind the characterisations is that the event $X_\ell^{(\tilde{B}_\ell)} \neq X_\ell^{(b_\ell^*)}$ occurs when a trajectory traced back from the generated particle tree in the bridge CPF has a different value at the block lower boundary than the reference.

Consider first the case of a small blocksize, that is, $u - \ell \approx 1$. In this case, PLU is approximately characterised by:

$$\mathrm{PLU}_{\mathrm{M}}(\ell, u) := 1 - \frac{M_{u|\ell}(X_u^* \mid X_\ell^*)}{\sum_{j=1}^N M_{u|\ell}(X_u^* \mid X_\ell^{(j)})}, \tag{9}$$

where $X_\ell^* := X_\ell^{(B_\ell)}$ and $X_u^* := X_u^{(B_u)}$ refer to the $\ell$th and $u$th value of a reference trajectory. The rationale for (9) comes from CPF-BS being a special case of the CPF-BBS for the dense blocking with unit blocksizes. Letting $b_t$ and $b_{t+1}$ denote the indices of the current reference, the probability of choosing $b_t$ in backward sampling [33] is given by:

$$\mathbb{P}(B_t = b_t \mid B_{t+1} = b_{t+1}) \propto w_t^{(b_t)} M_{t+1}(X_{t+1}^{(b_{t+1})} \mid X_t^{(b_t)}) G_{t+1}(X_t^{(b_t)}, X_{t+1}^{(b_{t+1})}).$$

Here, under the weak potential setting with approximately constant potentials, the right hand side approximately reduces to $M_{u|\ell}(X_u^{(b_u)} \mid X_\ell^{(b_\ell)})$ since $\ell = t, u = t + 1$ with a unit blocksize. The probability of choosing a non-reference is therefore approximately given by (9).

On the other hand, if the blocksize is large, $\mathrm{PLU}(\ell, u)$ is approximately characterised by:

$$\mathrm{PLU}_{\mathrm{G}}(\ell, u) := \left(1 - \frac{1}{N}\right) \prod_{k=\ell}^{u-1} \left(1 - \frac{p_k N}{(N-1)^2}\right), \tag{10}$$

where the quantity $p_k$ equals the probability that a resampling event occurs, divided by $N$. In the case of systematic resampling with mean partitioned weights $W_k^{(1:N)}$, (see Appendix A, Lemma 28 of [5]) and the weak potential setting, $p_k$ may be calculated as follows (for normalised $W_k^{(1:N)}$):

$$p_k = \frac{1}{2} \sum_{i=1}^N \left| W_k^{(i)} - \frac{1}{N} \right|. \tag{11}$$

The justification of (10) comes from a calculation detailed in Appendix F, which shows that $\mathrm{PLU}_{\mathrm{G}}(\ell, u)$ approximately equals the expected proportion of particles whose ancestor at time $\ell$ is not the reference after an 'artificial' conditional particle system has evolved for $u - \ell$ time steps from time $\ell$. Therefore, $\mathrm{PLU}_{\mathrm{G}}(\ell, u)$ maybe loosely interpreted as approximating the probability of choosing nonreference at time $\ell$, when the ancestry of a particle chosen uniformly at time $u$ is traced back until time $\ell$.

Our estimator for $\mathrm{PLU}(\ell, u)$ is constructed by 'interpolating' (9) and (10) such that

$$\widehat{\mathrm{PLU}}(\ell, u) := \mathrm{PLU}_{\mathrm{G}}(\ell, u)\mathrm{PLU}_{\mathrm{M}}(\ell, u)\left(1 - \frac{1}{N}\right)^{-1}, \tag{12}$$

where the scaling is added so that the estimator approximately reduces to (9) and (10) for short and long blocks, respectively, in the weak potential setting.

The estimator in (12) was derived assuming an access to CPF state with $N$ particles. It is also possible to estimate the $\widehat{\mathrm{PLU}}(\ell, u)$ from a CPF (or particle filter) state which has a different number of particles $N_0$ (which is often useful to take 'large' in practice so that $N_0 \gg N$). In this case, we can estimate $\mathrm{PLU}_{\mathrm{G}}$ and $\mathrm{PLU}_{\mathrm{M}}$ as follows, and then use (12) with the desired $N$ in the scaling.

To estimate $\mathrm{PLU}_{\mathrm{G}}$, we simply compute $p_k$ using (11) from the $N_0$ particles and substitute it directly to (10) with the desired $N < N_0$. For $\mathrm{PLU}_{\mathrm{M}}$ we use the alternative estimator of the form

$$\mathrm{PLU}_{\mathrm{M}}(\ell, u) = 1 - \frac{c(\ell, u)}{c(\ell, u) + N - 1}, \tag{13}$$

which follows by assuming that

$$M_{u|\ell}(X_u^* \mid X_\ell^*) \approx c(\ell, u) M_{u|\ell}^{(T)}, \tag{14}$$

where

$$M_{u|\ell}^{(T)} = \frac{1}{N_0 - 1} \sum_{j \neq B_\ell} M_{u|\ell}(X_u^* \mid X_\ell^{(j)}). \tag{15}$$

In other words, the block transition density for the reference is assumed to be approximately equal to a constant $c(\ell, u)$ times a 'typical' value of the block transition densities for particles not including the reference. The estimator (13) may be derived by appropriate substitution of (14) and (15) into (9).

### 7.3. Algorithm for blocking sequence selection.

In this section we describe a practical method based on (12) to choose the blocking sequence. Algorithm 9 describes a method that uses (12) to evaluate $S$ candidate blocking sequences $(T^{(s)}_{1:L^{(s)}})_{s=1,2,\ldots,S}$ in the context of the FK model $(M_{1:T}, G_{1:T})$. The additional parameters $N$ and $n$ stand for the number of particles and number of iterations, which are tuning parameters of the blocking candidate evaluation. Here, we use indexing notation where $A[i, j, k]$ stands for the element in row $i$, column $j$ and slice $k$ in an array $A$. Furthermore, the columns of arrays need not have the

**Algorithm 9** EvaluateBlockingCandidates($\{T^{(1)}_{1:L^{(1)}}, \ldots, T^{(S)}_{1:L^{(S)}}\}$, $M_{1:T}$, $G_{1:T}$, $N$, $n$)

```
for $j = 1, \ldots, n$ do
    $\underline{X}_{1:T}, \underline{A}_{1:T-1}, \underline{W}_{1:T} \leftarrow \mathrm{PF}(\rho_{\mathrm{syst}}, M_{1:T}, G_{1:T}, N)$
    Draw $B_T \sim \mathrm{Categ}(W_T^{(1:N)})$; $B_{1:T-1} \leftarrow \textsc{AncestorTrace}(A_{1:T-1}^{(1:N)}, B_T)$
    $\phi_{\mathrm{PLU}}[:,:,j] \leftarrow \textsc{EstimatePLU}(\{T^{(1)}_{1:L^{(1)}}, \ldots, T^{(S)}_{1:L^{(S)}}\}, \underline{X}_{1:T}, \underline{W}_{1:T-1}, B_{1:T})$
end for
for $s = 1, \ldots, S$ do
    Set $\bar{\phi}_{\mathrm{PLU}}[i, s] = \textsc{Mean}(\phi_{\mathrm{PLU}}[i, s, :])$ for $i = 1, \ldots, L^{(s)} - 1$.
end for
return $\bar{\phi}_{\mathrm{PLU}}$
```

same number of rows, and indexing operations with ':' mean 'all elements' in the particular dimension.

One iteration of the main loop in Algorithm 9 consists of running the standard particle filter (Algorithm 1) with mean partitioned systematic resampling followed by a traceback using ancestor tracing (Algorithm 4) in lines 2–3. Then, given the output of the particle filter, we estimate PLU using Algorithm 10 (see below) on line 4 for each block $(\ell, u)$ within each blocking sequence. The computation is a straightforward application of Equations (9)–(12) using the particle filtering results. Finally, lines 6–8 summarise the estimates of PLU by taking their mean over the $n$ replicate runs of the particle filter. The element $\bar{\phi}_{\mathrm{PLU}}[i, s]$ in the output of Algorithm 9 describes in terms of PLU, how efficient the $i$th block in the blocking sequence $s$ was.

**Algorithm 10** EstimatePLU($\{T^{(1)}_{1:L^{(1)}}, \ldots, T^{(S)}_{1:L^{(S)}}\}$, $\underline{X}_{1:T}, \underline{W}_{1:T-1}, B_{1:T}$)

```
Compute $p_k$ for $k = 1, \ldots, T-1$ using (11).
for $s = 1, \ldots, S$ do
    for $i = 1, \ldots, L^{(s)} - 1$ do
        Set $\ell = T^{(s)}_i$; $u = T^{(s)}_{i+1}$; $X^*_\ell = X^{(B_\ell)}_\ell$; $X^*_u = X^{(B_u)}_u$
        Compute $\mathrm{PLU}_M(\ell, u)$ using (9) and $\mathrm{PLU}_G(\ell, u)$ using (10)
        Set $\phi_{\mathrm{PLU}}[i, s] = \widehat{\mathrm{PLU}}(\ell, u)$ given in (12)
    end for
end for
return $\phi_{\mathrm{PLU}}$
```

Algorithm 9 may in principle be used to evaluate any blocking sequence, but we suggest to use it with Algorithm 13 given in Appendix E.2 that constructs blocking sequences where the blocksizes $T_{k+1} - T_k$ are powers of two. More precisely, if $T = 2^{p^*} + 1$ for some $p^*$, Algorithm 13 returns blocking sequences $T^{(i)}_{1:L^{(i)}}$ for $i = 1, 2, \ldots, p^* + 1$, where the blocksizes of the $i$th sequence are all constant $2^{i-1}$. If $T \neq 2^{p^*} + 1$, similar sequences are returned, but with a possible 'residual block' of length $< 2^{i-1}$ as the last block in each sequence $i$.

Finally, Algorithm 11 describes a method based on Algorithms 9 and 13 for choosing a single blocking sequence to be used with the CPF-BBS and a given FK model. In summary, Algorithm 11 first constructs the candidate blocking sequences using Algorithm 13. Then, Algorithm 9 is run to obtain $\bar{\phi}_{\mathrm{PLU}}$ given these sequences. The data $\bar{\phi}_{\mathrm{PLU}}$ is then reinterpreted as a set of elements $D_{\mathrm{PLU}}$, whose element $(\ell, b, p)$ describes the estimated PLU, $p$, of the block with lower boundary $\ell$ and upper boundary $\ell + b$. Finally, $D_{\mathrm{PLU}}$ is processed such that blocking sequences with largest blocksizes are considered first, and at each block lower boundary, the best performing blocksize in terms of the estimated PLU is selected to the output blocking sequence.

## 8. Linear diffusions with path integral weights

We discuss next a class of continuous-time models and their discretisations, for which the methods of Section 6–7 are particularly useful. We will consider instances of these models also in the experiments (Section 9).

We start with the continuous-time model on a time interval $[0, \tau]$. The prior dynamics $\mathbb{M}$ correspond to the solution of a linear stochastic differential equation (SDE):

$$\mathrm{d}X_t = \mathbf{F} X_t \mathrm{d}t + \mathbf{K} \mathrm{d}\mathbf{B}_t, \qquad X_0 \sim N(\mu_{\mathrm{init}}, \Sigma_{\mathrm{init}}) \tag{16}$$

**Algorithm 11** ChooseBlocking$(M_{1:T}, G_{1:T}, N, n)$

```
$\{T^{(1)}_{1:L^{(1)}}, \ldots, T^{(p)}_{1:L^{(p)}}\} \leftarrow$ DyadicCandidateBlockings$(T)$
$\bar{\phi}_{\mathrm{PLU}} \leftarrow$ EvaluateBlockingCandidates$(\{T^{(1)}_{1:L^{(1)}}, \ldots, T^{(p)}_{1:L^{(p)}}\}, M_{1:T}, G_{1:T}, N, n)$
Compute $D_{\mathrm{PLU}}$, a container with elements of the form $(\ell, b, p)$ based on $\bar{\phi}_{\mathrm{PLU}}$.
Initialise $D$, an empty container for elements of the form $(\ell, b)$.
for $s = p, p-1, \ldots, 1$ do
    Get lower boundaries and blocksizes $(\ell_k, b_k)$ for $k = 1, \ldots, L^{(s)} - 1$ from $T^{(s)}_{1:L^{(s)}}$.
    for $k = 1, \ldots, L^{(s)} - 1$ do
        Denote by $D^{(\ell_k)}_{\mathrm{PLU}}$ all elements of $D_{\mathrm{PLU}}$ whose block lower boundary equals $\ell_k$.
        if maximal $p$ is reached when blocksize equals $b_k$ among elements of $D^{(\ell_k)}_{\mathrm{PLU}}$ then
            Add $(\ell_k, b_k)$ to $D$.
            Remove all elements of $D_{\mathrm{PLU}}$ with $\ell$ such that $\ell_k \leq \ell < \ell_k + b_k$.
        end if
    end for
end for
return Blocking sequence constructed from elements of $D$.
```

where $\mathbf{B}_t$ is a $d$-dimensional Brownian motion and $\mathbf{F}$ and $\mathbf{K}$ are matrices of appropriate dimension, and $\mu_{\mathrm{init}}$ and $\Sigma_{\mathrm{init}}$ are the mean and covariance of the initial distribution, respectively. The law of interest is $\mathbb{M}$ weighted by non-negative weights of the form $w(x_{[0,\tau]}) = \exp(-\int_0^\tau V(x_u)\mathrm{d}u)$, where $V : \mathsf{X} \to [0, \infty]$ are 'potential' functions that 'penalise' the trajectories of $\mathbb{M}$. That is, the distribution of interest is proportional to $\mathbb{M}(\mathrm{d}x_{[0,\tau]})w(x_{[0,\tau]})$.

In practice, we assume a time discretisation of $[0, \tau]$, $0 = t_1 < t_2 < \cdots < t_T = \tau$, which leads to the discrete-time FK-model (1). The dynamics $M_{1:T}$ in (1) correspond to the marginals of $X_{[0,\tau]} \sim \mathbb{M}$, that is:

$$
\begin{aligned}
M_1 &= \mathrm{Law}(X_{t_1}) = N(\mu_{\mathrm{init}}, \Sigma_{\mathrm{init}}) \\
M_k(\cdot \mid x) &= \mathrm{Law}(X_{t_k} \mid X_{t_{k-1}} = x) \qquad \text{for } 2 \leq k \leq T,
\end{aligned}
\tag{17}
$$

which are linear-Gaussian. Appendix C details how $M_k$ can be derived from the parameters of the SDE, and also how their necessary conditional distributions required by Assumption 7 can be determined. The potential functions $G_{1:T}$ in (1) stem from approximating the path integral by a Riemann sum:

$$
w(x_{[0,\tau]}) = \prod_{k=1}^{T-1} \exp\left( - \int_{t_k}^{t_{k+1}} V(x_u)\mathrm{d}u \right) \approx \prod_{k=1}^{T-1} \exp\left( - |\Delta_k| V(x_{t_k}) \right),
\tag{18}
$$

where $\Delta_k = [t_k, t_{k+1})$ and $|\Delta_k| = t_{k+1} - t_k$. This leads to potentials of the following form:

$$
\begin{aligned}
G_1(x_{t_1}) &= \exp\big( - (t_2 - t_1) V(x_{t_1}) \big) \\
G_k(x_{t_{k-1}}, x_{t_k}) &= \exp\big( - (t_{k+1} - t_k) V(x_{t_k}) \big) \text{ for } 2 \leq k \leq T-1, \text{ and } G_T \equiv 1.
\end{aligned}
\tag{19}
$$

*Remark* 9. The scenario detailed above can be generalised and/or modified in a number of ways. Indeed, the potentials $G_k$ can also include purely discrete-time elements, as in our Cox process experiment (Section 9.2). The law $\mathbb{M}$, or equivalently $M_k$, can also correspond to the law of linear SDE *conditioned on* a number of linear-Gaussian observations. In such a case, the distributions $M_k$ are still linear-Gaussian, and we can derive the required conditional laws. This can be useful in many practical settings, and indeed was essential for our movement model example (Section 9.3).

## 9. Experiments

9.1. **Comparing conditional resamplings with Algorithm 7.** We first investigate the performance of the CPF-BBS (Algorithm 7) using the conditional resamplings $\rho_{\text{kill}}$ and $\rho_{\text{syst}}$. For reference, we also study conditional multinomial resampling with conditioning indices $i$ and $k$, $\rho_{\text{mult}}^{(i,k)}$. This conditional resampling may be simply implemented by first drawing the ancestor indices $A^{(1:N)} \sim \text{Categ}(w^{(1:N)})$ as in standard multinomial resampling, and then enforcing the condition $A^{(k)} = i$ (since $A^{(1:N)}$ are independent).

In this section, we study a correlated random walk incorporating a path integral type potential function, hereafter called the CTCRW-P model. The dynamics of the model $X_t = (V_t \; L_t)^T$ are driven by the SDE

$$\begin{aligned} \mathrm{d}V_t &= -\beta_v V_t \mathrm{d}t + \sigma \mathrm{d}B_t \\ \mathrm{d}L_t &= [-\beta_x L_t + V_t]\mathrm{d}t, \end{aligned} \tag{20}$$

where $B_t$ is the standard Brownian motion, $\sigma$, $\beta_v$ and $\beta_x$ are parameters, and $(L_t)_{t\geq 0}$ and $(V_t)_{t\geq 0}$ represent location and velocity processes, respectively. The FK representation (17) & (19) of CTCRW-P is given by $M_1 := N(0, S)$, $M_k(\,\cdot \mid x) := N(T_{t_{k-1},t_k}x, Q_{t_{k-1},t_k})$, for $2 \leq k \leq T$ and $V(X_t) := L_t^2/(2\eta^2)$. Here, $\eta$ is a parameter, and $T_{t_{k-1},t_k}$, $Q_{t_{k-1},t_k}$ and $S$ are the transition matrix, conditional covariance matrix and stationary covariance matrix, respectively, arising in the solution of the linear SDE (20). Their expressions are given in Appendix D.1, in Equations (44), (45)–(46) and (47)–(48), respectively.

We ran the CPF-BBS targetting CTCRW-P with the configurations $N \in \{2, 4, 8, 16, 32\}$, blocktime $\in \{2^{-7}, 2^{-6}, \ldots 2^{6}\}$ and $r \in \{\rho_{\text{syst}}, \rho_{\text{kill}}, \rho_{\text{mult}}\}$. Here, blocktime parameterises the blocking sequence in terms of the 'physical time' of the discretised SDE. The blocksizes $T_{k+1} - T_k$ in Algorithm 7, may simply be obtained by dividing blocktime by $|\Delta_k|$ (see below). For each run of the CPF-BBS, we used 21000 iterations with the first 1000 discarded as burnin.

We set $\tau = 2^6$, $|\Delta_k| = 2^{-7}$, $\eta = 1.0$ and $\sigma \in \{0.125, 0.5, 2.0\}$, which controls the variability in the velocity process. Each time, given $\sigma$, we solved for the parameters $\beta_x$ and $\beta_v$ such that the stationary covariance matrix (47) had unit variances on the diagonal. This was done to ensure that the variability of the process remains similar as $\sigma$ changes.

The simulations were run with all combinations of the algorithm and model configurations described above. We estimated PLU (discussed in Section 7.1) by tallying iterations where $x_\ell^{(\tilde{B}_\ell)} \neq x_\ell^{(b_\ell^*)}$ and dividing by their total, and estimated the IACT for $L_{0.0}$ using batch means [17]. Figure 1 summarises the results of this experiment. The mean PLU shown in the top row is computed over the number of blocks (given here by $\tau$/blocktime). The figure shows systematic and killing resampling performing better than multinomial resampling, which can be seen from the lower IACTs and higher mean PLU. The performance with multinomial resampling is poor here, as expected, since the model has weak potentials with $|\Delta_k| = 2^{-7}$. In contrast, killing and systematic resampling behave nearly uniformly, with systematic resampling performing slightly better. This finding aligns well with the theoretical and empirical findings in [5] for the particle filter in a similar context of path integral potentials and $|\Delta_k|$ close to 0.

The CPF-BBS coincides with the CPF-BS when blocktime $= |\Delta_k|$, which corresponds to the first value on the horizontal axis. Even though increasing $N$ naturally improves the performance of the CPF-BS too, the CPF-BBS has better simulation efficiency with an appropriately chosen blocktime, for any $N$ in the simulation. Note that the estimation of the IACT is quite noisy here, since the mixing is poor especially with multinomial resampling and with poorly chosen blocking sequences induced by the value of blocktime. In contrast,

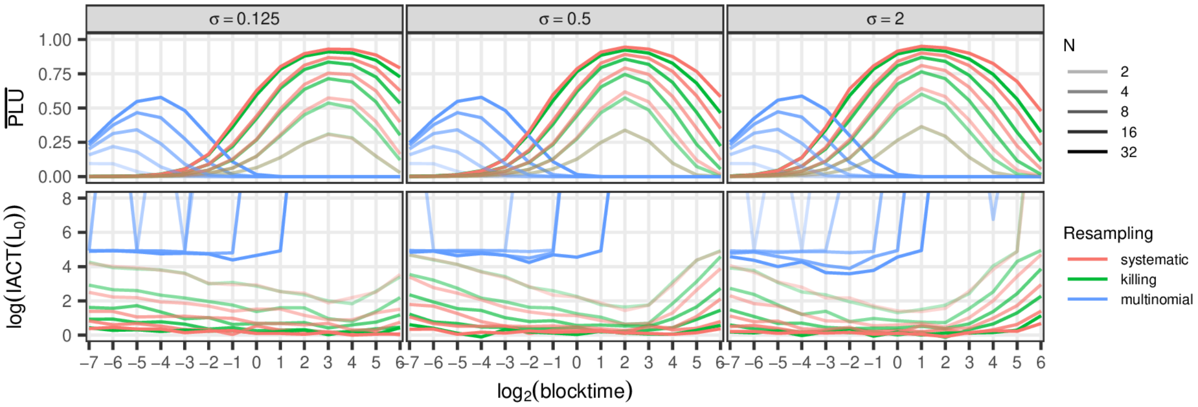


FIGURE 1. The estimated mean PLUs and the logarithm of IACT with varying $\sigma$ for the location state variable at time 0.0 in the CTCRW-P model. The value of $|\Delta_k|$ was set to $2^{-7}$. The performance of CPF-BS is seen at the far left, with blocktime $= 2^{-7}$.

the computed mean PLU appears less noisy, and in the case of systematic and killing resampling the best blocktime in terms of IACT is identified.

We also investigated the relationship of PLU with $\text{IACT}_{32.0}$, and the findings were similar. A further experiment fixing $\sigma = 1.0$ and varying $\eta \in \{0.125, 0.5, 2.0\}$ instead also resulted in similar findings (see supplementary Figure 8).

9.2. **Choice of the blocking sequence.** As already illustrated empirically with Figure 1 and discussed in Section 7, the choice of the blocking sequence is a tuning parameter affecting the sampling efficiency of the CPF-BBS. Figure 2 exemplifies this further by showing another look at the results obtained from the experiment in the previous section. Here, the logarithm of the inverse relative efficiency (IRE) is plotted at each timepoint when systematic resampling was used. The IRE is obtained by scaling the IACT by the number of particles, and measures the asymptotic efficiencies of estimators with varying computational costs [18]. The panes from left to right show the results with varying blocktime and represent a range of algorithms from the CPF-BS (blocktime $= 2^{-7}$) to an algorithm similar to running the CPF twice (blocktime $= 2^6$). The optimal algorithms use only 4 particles, motivating the search for an appropriate blocktime (or blocking sequence). By visual inspection, it appears the optimal blocktimes here are around $2^1 - 2^2$ for $\sigma = 2.0$, $2^2 - 2^3$ for $\sigma = 0.5$ and $2^3 - 2^4$ for $\sigma = 0.125$. Here, a decrease in the value of $\sigma$ results in a larger optimal blocktime, since decreasing $\sigma$ leads to increased 'stiffness' in the dynamics of $M_{1:T}$. The optimal blocktimes represent balances where the blocks are large enough so that bridging between the lower and upper boundaries is sufficiently likely, and small enough so that degeneracy of the particle tree within the block is avoided.

Next, we investigate how well the estimates of $\bar{\phi}_{\text{PLU}}$ computed using Algorithm 9 coincide with PLU. We studied the relationship of $\bar{\phi}_{\text{PLU}}$ and PLU with respect to blocktime (that is, with blocking sequences constructed with constant blocksizes) using the CTCRW-P model with $N \in \{2^1, 2^2, \ldots, 2^{10}\}$ and the parameter $\sigma \in \{0.03125, 0.125, 0.5, 2, 4\}$. The rest of the model configuration was as in Section 9.1. To estimate PLU, we ran 1100 iterations of Algorithm 7 with the first 100 discarded burnin, monitoring for each block the proportion of iterations where $x_\ell^{(\tilde{B}_\ell)} \neq x_\ell^{(b_\ell^*)}$. In Algorithm 9, we used $n = 50$ runs of the particle filter and $N$ as reported above. Figure 3 visualises the results for $N \leq 2^7$ (the results for $N > 2^7$ yield

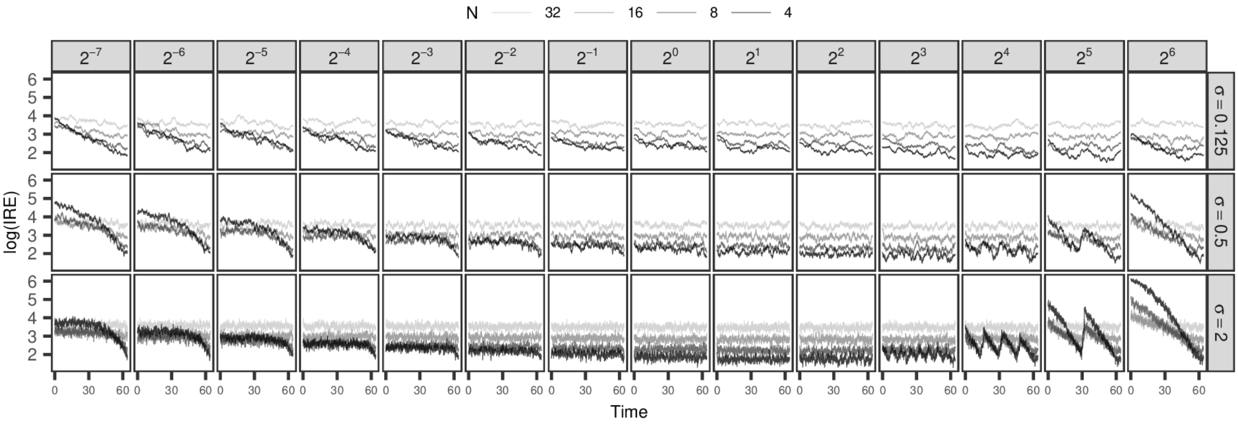


FIGURE 2. The logarithms of inverse relative efficiency obtained at each timepoint with conditional systematic resampling in the experiment discussed in Section 9.1. The columns show the results with varying blocktime in Algorithm 7.

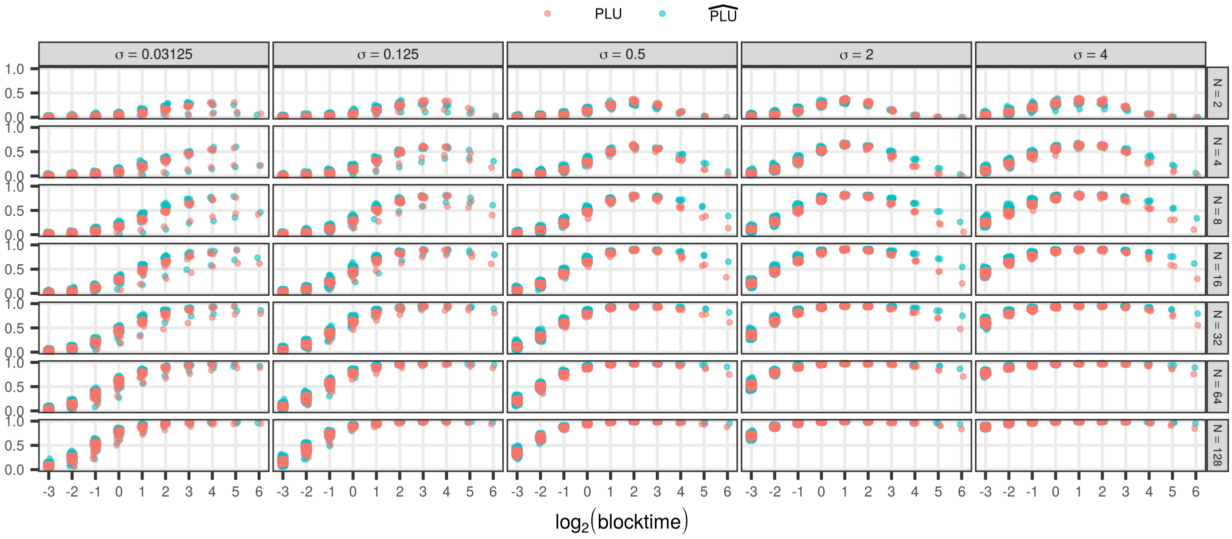


FIGURE 3. The PLU (orange) and $\bar{\phi}_{\text{PLU}}$ (light blue) for each block induced by the blocktime on the horizontal axis for the CTCRW-P. The points are slightly jittered for visualisation.

no further conclusions). The estimated PLU and $\bar{\phi}_{\text{PLU}}$ appear to be in close agreement, with only slight discrepancies seen for large blocktimes. This finding motivates the use of $\bar{\phi}_{\text{PLU}}$ as a maximisation criterion for finding a blocksize that likely results in a high overall PLU as well.

Next, we turned to study Algorithm 11 for selecting the blocking sequence based on $\bar{\phi}_{\text{PLU}}$. We investigated this with a model that slightly differs from the form (19), and is a Cox process model incorporating a reflected Brownian motion (CP-RBM) first appearing in [5] and briefly detailed (with minor changes) below.

The CP-RBM model assumes an inhomogeneous Poisson process (IPP) in time, generating observation sequences $\tilde{\tau}$. The intensity function of the IPP is piecewise constant, and given by

$$\lambda(t) = \beta \exp\left(-\alpha X_{t_k}\right), \quad \text{for } t \in [t_k, t_{k+1}). \tag{21}$$

The process $(X_{t_k})_{k=1,\dots,T}$ is distributed such that

$$X_{t_1} \sim N^{(r)}(0,1,a,b), \text{ and } X_{t_k} \mid X_{t_{k-1}} = x_{t_{k-1}} \sim N^{(r)}(x_{t_{k-1}}, |\Delta_k|\sigma^2, a, b), \tag{22}$$

where $N^{(r)}(\mu,\sigma^2,a,b)$ is a distribution we call the 'reflected normal distribution', with parameters $\mu$, $\sigma$ and bounds $a$ and $b$. To simulate from $N^{(r)}(\mu,\sigma^2,a,b)$, one first draws $Z \sim N(\mu,\sigma^2)$ and then sets $X = \text{reflect}(Z,a,b)$, where 'reflect' is an operation that recursively reflects (that is, mirrors over a boundary) $Z$ with respect to $a$ (if $Z < a$) or $b$ (if $Z > b$) until a value within $(a,b)$ is obtained and outputted.

To apply the CPF-BBS with the CP-RBM, we use the following FK representation that differs from that of [5] such that the reflection of the process $X$ is accounted for in the potential functions:

$$\begin{aligned} M_1 &:= N(0,1), \text{ and } M_k(\cdot \mid x) := N(x, |\Delta_{k-1}|\sigma^2), \text{ for } 2 \le k \le T \\ G_1(x) &:= \frac{N^{(r)}(x;0,1,a,b)\exp\left(-|\Delta_1|\beta\exp\left(-\alpha x\right)\right)}{N(x;0,1)} \left(\beta\exp\left(-\alpha x\right)\right)^{\mathbf{1}(\exists i \text{ s.t } \tilde{\tau}_i \in \Delta_1)} \\ G_k(x,y) &:= \frac{N^{(r)}(y;x,|\Delta_{k-1}|\sigma^2,a,b)\exp\left(-|\Delta_k|\beta\exp\left(-\alpha y\right)\right)}{N(y;x,|\Delta_{k-1}|\sigma^2)} \times \\ &\qquad \left(\beta\exp\left(-\alpha y\right)\right)^{\mathbf{1}(\exists i \text{ s.t } \tilde{\tau}_i \in \Delta_k)}, \text{ for } 2 \le k \le T \end{aligned} \tag{23}$$

where $|\Delta_T| = 0$. This FK model is valid for the inference of the CP-RBM in the situation that the time discretisation is made fine enough such that each $\Delta_k$ contains at most one observation. The density $N^{(r)}(x;\mu,\sigma^2,a,b)$ contains an infinite sum, which we truncate to the first ten terms; the formula is given in Appendix D.2.

We first drew a realisation of the process $X$ using (22) with $|\Delta_k| = 2^{-6}$, $\sigma = 0.3$, $a = 0$, $b = 3$ and time interval length $\tau = 2^8$. Then, conditional on this realisation, we simulated one dataset, $\tilde{\tau}$, from the IPP defined by (21) with $\alpha = 1$ and $\beta = 0.5$. Finally, we refined the time discretisation such that (23) could be used.

For the blocking sequences, we considered the sequences induced by the constant blocktimes $\{2^{-6}, 2^{-5}, \dots, 2^5\}$ and a (nonhomogeneous) blocking sequence constructed using Algorithm 11 with $n = 50$ and $N = 8$. Here, a minor change to the choice of candidate blockings (that is, Algorithm 13) was done: instead of constructing them using block*sizes* (integers) in powers of two as discussed in Section 7, we constructed them using the power of two block*times* $2^{-6} - 2^5$ as this is more natural for a continuous-time model. For each blocking sequence, we then applied the CPF-BBS with $N = 8$ for 26000 iterations with the first 1000 discarded as burnin.

Figure 4 summarises the results of the experiment. The top pane shows the true simulated state, the observations $\tilde{\tau}$ and the 50% and 95% credible intervals of the distributions $X_t \mid \tilde{\tau}$ for $t \in t_1, t_2, \dots, t_T$. The middle pane compares the IACTs obtained from the samples of said distributions with the different blocking strategies; the nonhomogeneous blocking is highlighted in red. The IACTs for the blocking sequences constructed for blocktimes $> 2^3$ were greater than for the depicted blocking strategies. Finally, the bottom pane visualises the nonhomogeneous blocking sequence obtained using Algorithm 11.

In terms of the IACT the blocking sequence returned by Algorithm 11 appears to perform similarly to the best choices for the blocking sequences constructed with constant blocktimes, indicating that the method here provides adequate performance without trial runs of the CPF-BBS. The bottom pane shows how the blocktime of the nonhomogeneous blocking switches between $1/2, 1$ and $2$.

9.3. **Movement modelling with terrain preference.** We conclude with an application of the CPF-BBS to a movement modelling scenario. Here, we are interested in modelling

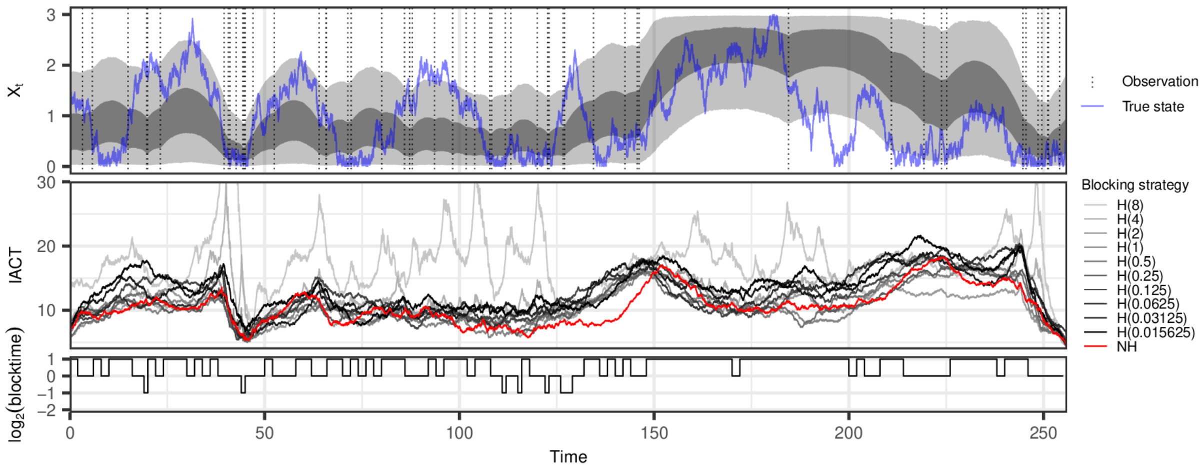


Figure 4. Top: the observations $\tilde{\tau}$, the true simulated state and the 50% and 95% credible intervals for $X_t \mid \tilde{\tau}$ (shaded) of the CP-RBM model. Middle: the IACTs with homogeneous blocking with $\mathsf{H}$(blocktime) (shades of gray), and with the nonhomogeneous blocking $\mathsf{NH}$ (red). Bottom: the nonhomogeneous blocking from Algorithm 11 using $n = 50$ and $N = 8$.

the movement of an object on a plane based on noisy observations and knowledge of terrain in which the object moves. We assume that the object has a 'preference' for spending time in certain types of terrain.

To model such a setting, we build on the continuous-time correlated random walk (CTCRW) model developed for animal movement modelling based on telemetry data [21]. The dynamics of the CTCRW model arise from a special case of the SDE (20), obtained by setting $\beta_x = 0$ and denoting $\beta := \beta_v$. Using this SDE independently in $x$ and $y$ dimensions yields a 4-dimensional state $X_t = (V_t^{(x)}, L_t^{(x)}, V_t^{(y)}, L_t^{(y)})^T$ and a movement model on the plane, which we call the CTCRW SDE. The full CTCRW model also incorporates two-dimensional location observations $y = (y_k)_{k=1,2,\ldots,K_y}$ observed at times $(\tilde{t}_k)_{k=1,2,\ldots,K_y}$. Each observation is related to the location state variables, $\mathbf{L}_t = (L_t^{(x)}, L_t^{(y)})^T$, with $y_k = \mathbf{L}_{\tilde{t}_k} + \epsilon_k$, where $\epsilon_k \sim N(0, \eta^2 I_2)$, where $\eta$ is a standard deviation and $I_2$ stands for the $2 \times 2$ identity matrix. We use the initial distribution $X_{t_1} \sim N((0, y_{11}, 0, y_{12})^T, \mathrm{diag}(\sigma_V^2, \sigma_L^2, \sigma_V^2, \sigma_L^2))$, where $y_{11}$ and $y_{12}$ are the first and second coordinates of the first observation, respectively, $\sigma_V^2 = \sigma^2/(2\beta)$ (the stationary variance of the velocity component) and $\sigma_L^2$ is a parameter. The details regarding the solution of the CTCRW SDE are given in Section D.3.

Our model, which we denote CTCRW-T ($T$ standing for 'terrain') differs from the CTCRW model of [21] by incorporating the effect of terrain. We use a discretisation of the CTCRW SDE conditioned on the observations as the sequence of $M_k$'s in the FK representation of the CTCRW-T. More specifically, we define

$$\begin{aligned} M_1 &= \mathrm{Law}(X_{t_1} \mid Y = y), \\ M_k(\,\cdot \mid x) &= \mathrm{Law}(X_{t_k} \mid X_{t_{k-1}} = x, Y = y), \qquad \text{for } 2 \le k \le T, \end{aligned} \tag{24}$$

where $X_t$ stands for the state of the CTCRW model at time $t$ and $Y$ stands for all observations and $y$ their realised values. The distributions in (24) are Gaussian, and their computation can be carried out by very similar conditioning as discussed in Appendix C.

The CTCRW-T models terrain preference through its potential that is of the form (18) with $V(x) = -\log(v_i)$ when $x$ is in terrain $i$. We call the values $v_i \in [0, 1]$, $i = 1, \ldots, K_T$, 'terrain coefficients', which induce the potential values for each of the $K_T$ terrain types.

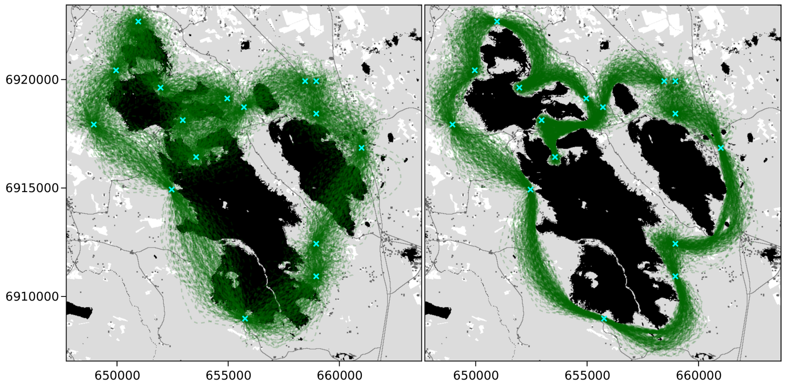


FIGURE 5. Comparison of CTCRW (left) and CTCRW-T (right) with 250 simulated location trajectories. The observations used by both models are shown with crosses. The CTCRW-T also uses terrain, which includes lakes (black).

We apply the CTCRW-T model in a region of Finland containing lakes, plotted in the background of Figure 5. The colors of the background map depict the value of $V$, with black representing larger values, that is, lower potential. We define the terrain types based on the Corine Land Cover classification [16] which classifies each 20 × 20 metre cell in Finland to one of five classes. The terrain types and their associated terrain coefficients (in parentheses) are ‘Artificial surfaces’ (0.2), ‘Agricultural areas’ (0.6), ‘Forests and seminatural areas’ (0.5), ‘Water bodies’ (0.0) and ‘Wetlands’ (0.5). The terrain coefficient of ‘Water bodies’ is set to zero, since we want to constrain the movement on land only.

With the potential map constructed this way, we set $\tau = 16$ and handpicked 16 observed locations in a clockwise pattern around the lakes, spacing the observation times equidistantly in time. The observed locations appear as crosses in Figure 5. The CTCRW model parameters $\beta$ and $\sigma$ were fit via maximum likelihood, and we set $\eta = 50$ and $\sigma_L = 50$.

We then applied the CPF-BBS with systematic resampling, $N = 16$ and blocktime $= 1.0$ for 11000 iterations, discarding the first 1000 as burnin. $|\Delta_k|$ was set to $2^{-7}$. The right pane of Figure 5 shows 250 of the simulated location trajectories from the CTCRW-T model. In comparison, the left pane shows trajectories simulated from the CTCRW model conditioned on the observed locations, simulated using (24). We observe that the trajectories simulated from the CTCRW-T model are influenced by the conditioning on the observations, while avoiding water bodies, as desired.

We also tested the performance of the CPF-BBS with the blocking sequence obtained using Algorithm 11 (using $N = 512$ and $n = 25$), as well as CPF-BS in this example. Here, the number of particles for Algorithm 11 had to be set slightly higher to ensure that a sufficient number of particles end up in regions of positive potential (due to the hard constraint induced by ‘Water bodies’). Figure 6 compares the three algorithms by plotting the IACT of the state variable $L_t^{(x)}$ with respect to time. The plots for the other state variables were similar. Clearly, the simulation efficiencies of both variants of the CPF-BBS are superior here in comparison to the CPF-BS. Between the optimised blocking and constant blocking, the finding is similar as with the CP-RBM model: the blocking optimisation via Algorithm 11 yields similar results as the ‘hand tuned’ constant blocking with blocktime $= 1.0$. The supplementary material also includes an animation that visualises the values of all sampled trajectories at each timepoint of the simulation, showing slower exploration of the target distribution using the CPF-BS.

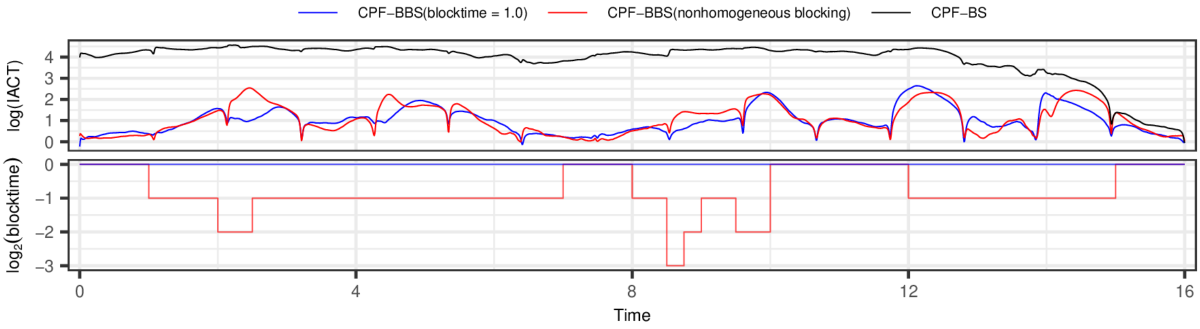


FIGURE 6. The logarithmic IACT (top) of the state $L_t^{(x)}$ in the CTCRW-T model with $|\Delta_k| = 2^{-7}$ for CPF-BS and the CPF-BBS with blocktime 1.0 and the blocking from Algorithm 11 (bottom).

We also experimented with the three algorithms using a higher value for $|\Delta_k|$, a situation where a greater discretisation error in the approximation (18) may be tolerated. We found that when $|\Delta_k|$ was increased to 0.125, the resulting IACTs of $L_t^{(x)}$ were similar between the three algorithms (see Figure 7 in the supplementary material).

## 10. DISCUSSION

The methods presented in this paper make inference more efficient (and feasible) for an important class of statistical models, which includes hidden Markov models (HMMs) involving weakly informative observations and, in particular, time-discretisations of continuous-time path integral models.

Our first contribution was presenting two new conditional resampling algorithms for CPFs in such a context: the killing resampling $\rho_{\text{kill}}$, and the systematic resampling with mean partitioned weights $\rho_{\text{syst}}$. Our empirical experiments with the developed resampling algorithms revealed that $\rho_{\text{syst}}$ performs slightly better than $\rho_{\text{kill}}$, coinciding with the recent findings of [5] for the standard particle filter in a similar context. Based on our findings, we recommend to use $\rho_{\text{syst}}$ with the CPF in the weak potential regime.

Our main contribution is a new CPF, which we call the conditional particle filter with bridge backward sampling (CPF-BBS), which may be regarded as a generalisation of the celebrated CPF with backward sampling (CPF-BS) [33]. The key ingredient of the CPF-BBS which avoids performance issues of the CPF-BS in the weak potentials and slowly mixing context, is the bridging CPF step that updates the latent trajectory subject to a blocking sequence that acts as a tuning parameter of the method. Since tuning the blocking sequence by ‘trial and error’ is laborous and costly, we presented a computationally cheap procedure for finding an appropriate blocking sequence. The procedure is based on a proxy of the integrated autocorrelation time of the output Markov chain, the so-called probability of lower boundary updates (PLU), which measures the probability that the bridge CPF updates the value at the block lower boundary. We derived an estimator for PLU that we suggest to use for blocking sequence tuning via Algorithm 11 that uses a small number of trial runs of the standard particle filter with ancestor tracing to estimate PLU prior to running the CPF-BBS.

The CPF-BBS is generally applicable, assuming that the conditional distributions $M_{u|\ell}$ and $M_{k|k-1,u}$ related to the individual blocks $(\ell, u)$ and the dynamics $M_{1:T}$ may be computed. This may seem restrictive, but it is important to note that $M_k$ need not necessarily correspond to the model, but may be any ‘proposal’ distributions satisfying Assumption

7. Careless choice of $M_k$ might however result in informative potentials $G_k$ and therefore poor performance. The contrary is also possible: with suitably chosen $M_k$, the $G_k$ can be weakly informative, even if the HMM observations are informative. This can be achieved by designing $M_k$ by suitable 'lookaheads' [19], such as an approximate smoothing distribution from a Laplace approximation [cf. 32, Section 8.1].

The experiments suggested that our estimator for PLU is in good agreement with the true PLU. Algorithm 11, which finds an appropriate blocking automatically, showed promising behaviour in our experiments, leading to performance similar to 'hand tuning' the blocking sequence. Using Algorithm 11 is easy: it only requires the user to specify the number of iterations and number particles used in the selection to obtain adequate performance 'out of the box'. In all of the examples we studied, we found 50 iterations to suffice for block selection, but we presume that the number of particles has to be chosen in a model by model basis.

The performance of the CPF-BBS in practice was promising: we found that the method can provide a substantial performance improvement over CPF-BS in the weak potential setting. This was particularly clear with our movement modelling experiment, which can be of independent interest for certain applications.

We believe that PLU and the ideas in the estimator we derived for it can be of interest in other contexts, too. In Section 7.2 we discussed the possibility of obtaining estimates for PLU for $N \neq N_0$, where $N_0$ is the number of particles used for the necessary computations. We found empirically (results not reported) that the agreement between PLU and $\widehat{\mathrm{PLU}}$ remains similar as in Figure 3 if we use this alternative estimation procedure. This method could potentially be elaborated to a heuristic for choosing the number of particles $N$ for the CPF-BBS. One potential way forward is to determine a 'cutoff level' for how large a PLU is 'large enough,' and the smallest $N$ reaching this level would be chosen. However, further developments of these ideas are out of the scope of the present paper.

Finally, we note that in some applications relevant for the weakly informative context, the initial distribution $M_1$ can be diffuse (relative to the smoothing distribution) — even an (improper) uniform measure. In such a case, the CPF and also the CPF-BBS will suffer from poor mixing, but there are relatively direct extensions that are applicable also with the CPF-BBS. Indeed, [15] discuss general state augmentations that can be useful, and a straightforward implementation is often possible in terms of $M_1$-reversible transitions [22].

## Acknowledgements

SK and MV were funded by the Academy of Finland project 315619 and the Finnish Centre of Excellence in Randomness and Structures. The authors wish to acknowledge CSC — IT Center for Science, Finland, for computational resources.

## References

[1] Christophe Andrieu, Arnaud Doucet, and Roman Holenstein. Particle Markov chain Monte Carlo methods. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 72(3):269–342, 2010.

[2] Marc Arnaudon and Pierre Del Moral. A duality formula and a particle Gibbs sampler for continuous time Feynman-Kac measures on path spaces. *Electron. J. Probab.*, 25:1–54, 2020.

[3] Christopher K Carter, Eduardo F Mendes, and Robert Kohn. An extended space approach for particle Markov chain Monte Carlo methods. Preprint arXiv:1406.5795, 2014.

[4] Nicolas Chopin and Sumeetpal S Singh. On particle Gibbs sampling. *Bernoulli*, 21(3):1855–1883, 2015.

[5] Nicolas Chopin, Sumeetpal S. Singh, Tomás Soto, and Matti Vihola. On resampling schemes for particle filters with weakly informative observations. Preprint arXiv:2203.10037, 2022.

[6] Dan Crisan, Pierre Del Moral, and Terry Lyons. Discrete filtering using branching and interacting particle systems. *Markov Process. Related Fields*, 5(3):293–318, 1999.

[7] A. Marie d'Avigneau, Sumeetpal S. Singh, and Raimund J. Ober. Limits of accuracy for parameter estimation and localization in single-molecule microscopy via sequential monte carlo methods. *SIAM Journal on Imaging Sciences*, 15(1):139–171, 2022.

[8] Piet de Jong and Murray J Mackinnon. Covariances for smoothed estimates in state space models. *Biometrika*, 75(3):601–602, 1988.

[9] Pierre Del Moral. Non-linear filtering: interacting particle resolution. *Markov Process. Related Fields*, 2(4):555–581, 1996.

[10] Pierre Del Moral. *Feynman-Kac Formulae*. Springer, 2004.

[11] Pierre Del Moral. *Mean field simulation for Monte Carlo integration*. Chapman and Hall/CRC, 2013.

[12] Pierre Del Moral and Lawrence M Murray. Sequential Monte Carlo with highly informative observations. *SIAM/ASA Journal on Uncertainty Quantification*, 3(1):969–997, 2015.

[13] Arnaud Doucet, Mark Briers, and Stéphane Sénécal. Efficient block sampling strategies for sequential Monte Carlo methods. *J. Comput. Graph. Statist.*, 15(3):693–711, 2006.

[14] J. Durbin and S. J. Koopman. *Time series analysis by state space methods*. Oxford University Press, New York, 2nd edition, 2012.

[15] Paul Fearnhead and Loukia Meligkotsidou. Augmentation schemes for particle MCMC. *Statist. Comput.*, 26(6):1293–1306, 2016.

[16] Finnish Environment Institute SYKE. CORINE Land Cover 2018. The data are downloaded from the Data Download Service of SYKE on 03.12.2018., 2018.

[17] James M Flegal and Galin L Jones. Batch means and spectral variance estimators in Markov chain Monte Carlo. *Ann. Statist.*, 38(2):1034–1070, 2010.

[18] Peter W Glynn and Ward Whitt. The asymptotic efficiency of simulation estimators. *Operations research*, 40(3):505–520, 1992.

[19] Pieralberto Guarniero, Adam M. Johansen, and Anthony Lee. The iterated auxiliary particle filter. *J. Amer. Statist. Assoc.*, 112(520):1636–1647, 2017.

[20] Charles AR Hoare. Quicksort. *The Computer Journal*, 5(1):10–16, 1962.

[21] Devin S Johnson, Joshua M London, Mary-Anne Lea, and John W Durban. Continuous-time correlated random walk model for animal telemetry data. *Ecology*, 89(5):1208–1215, 2008.

[22] Santeri Karppinen and Matti Vihola. Conditional particle filters with diffuse initial distributions. *Statist. Comput.*, 31(3):1–14, 2021.

[23] Anthony Lee, Sumeetpal S. Singh, and Matti Vihola. Coupled conditional backward sampling particle filter. *Ann. Statist.*, 48(5):3066–3089, 2020.

[24] Fredrik Lindsten, Pete Bunch, Sumeetpal S Singh, and Thomas B Schön. Particle ancestor sampling for near-degenerate or intractable state transition models. Preprint arXiv:1505.06356, 2015.

[25] Fredrik Lindsten, Michael I Jordan, and Thomas B Schön. Particle Gibbs with ancestor sampling. *J. Mach. Learn. Res.*, 15(1):2145–2184, 2014.

[26] Blazej Miasojedow and Wojciech Niemiro. Particle Gibbs algorithms for Markov jump processes. Preprint arXiv:1505.01434, 2015.

[27] Marcin Mider, Moritz Schauer, and Frank van der Meulen. Continuous-discrete smoothing of diffusions. *Electron. J. Statist.*, 15(2):4295–4342, 2021.

[28] Simo Särkkä and Arno Solin. *Applied stochastic differential equations*, volume 10. Cambridge University Press, 2019.

[29] Sumeetpal S Singh, Frederik Lindsten, and Eric Moulines. Blocking strategies and stability of particle Gibbs samplers. *Biometrika*, 104(4):953–969, 2017.

[30] Henrik Thurfjell, Simone Ciuti, and Mark S Boyce. Applications of step-selection functions in ecology and conservation. *Movement ecology*, 2(1):1–12, 2014.

[31] Matti Vihola, Jouni Helske, and Jordan Franks. Importance sampling type estimators based on approximate marginal Markov chain Monte Carlo. Preprint arXiv:1609.02541v3, 2017.

[32] Matti Vihola, Jouni Helske, and Jordan Franks. Importance sampling type estimators based on approximate marginal MCMC. *Scand. J. Stat.*, 47(4):1339–1376, 2020.

[33] Nick Whiteley. Discussion on Particle Markov chain Monte Carlo methods. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 72(3):306–307, 2010.

# SUPPLEMENTARY MATERIAL

## Appendix A. Validity of CPF with killing and systematic resampling

We start by stating an easy lemma, whose proof is immediate.

**Lemma 10.** *For a valid conditional resampling scheme $r^{(p,n)}(\,\cdot \mid g^{(1:N)})$ and its unconditional version $r$, it holds that:*

*(i)* $\mathbb{E}_{r(\,\cdot\,|g^{(1:N)})}\big[\sum_{i=1}^N \mathbf{1}\left(A^{(i)} = j\right)\big] = N\frac{g^{(j)}}{\sum_{i=1}^N g^{(i)}}$ *for all $j \in \{1{:}N\}$, and*

*(ii)* $r(a^{(1:N)} \mid g^{(1:N)})\mathbf{1}\left(a^{(n)} = p\right) = \frac{g^{(p)}}{\sum_{\ell=1}^N g^{(\ell)}} r^{(p,n)}(a^{(1:N)} \mid g^{(1:N)})$ *for all $a^{(i)}$, $n$ and $p$ in $\{1{:}N\}$.*

In what follows, we denote $\underline{M}_1(\underline{x}_1) = \prod_{i=1}^N M_1(x_1^{(i)})$ and $\underline{M}_k(\underline{x}_k \mid x_{k-1}^{(\underline{a})}) = \prod_{i=1}^N M_k(x_k^{(i)} \mid x_{k-1}^{(a^{(i)})})$.

*Proof of Theorem 2.* Assume that $X_{1:T}^* \sim \pi$ and $B_{1:T} \sim U(\{1{:}N\}^T)$ independently. The joint distribution of $B_{1:T}$, the particles $\underline{X}_{1:T}$, and the ancestories $\underline{A}_{1:T-1}$ generated by the CPF, may be written as

$$\frac{\underline{M}_1(\underline{x}_1)}{\mathcal{Z}N^T}\left[\prod_{k=2}^T r^{(b_{k-1},b_k)}\big(\underline{a}_{k-1} \mid G_{k-1}(\underline{\mathbf{x}}_{k-1})\big)\underline{M}_k(\underline{x}_k \mid x_{k-1}^{(\underline{a}_{k-1})})G_{k-1}(\mathbf{x}_{k-1}^{(b_{k-1})})\right]G_T(\mathbf{x}_T^{(b_T)})$$

$$= \frac{\underline{M}_1(x_1^{(1:N)})}{\mathcal{Z}}\left(\prod_{k=1}^T \frac{1}{N}\sum_{\ell=1}^N G_k(\mathbf{x}_k^{(\ell)})\right)$$

$$\left(\prod_{k=2}^T \mathbf{1}\left(a_{k-1}^{(b_k)} = b_{k-1}\right) r(\underline{a}_{k-1} \mid G_{k-1}(\underline{\mathbf{x}}_{k-1}))\underline{M}_k(\underline{x}_k \mid x_{k-1}^{(\underline{a}_{k-1})})\right)\frac{G_T(\mathbf{x}_T^{(b_T)})}{\sum_{\ell=1}^N G_T(\mathbf{x}_T^{(\ell)})}, \tag{25}$$

by Lemma 10. Including the variables $\tilde{B}_{1:T}$ to (25) adds the following factor:

$$\left[\prod_{k=2}^{T} \mathbf{1}\left(a_{k-1}^{(\tilde{b}_k)} = \tilde{b}_{k-1}\right)\right] \frac{G_T(\mathbf{x}_T^{(\tilde{b}_T)})}{\sum_{\ell=1}^N G_T(\mathbf{x}_T^{(\ell)})} \tag{26}$$

The joint distribution—product of (25) and (26)—is clearly symmetric with respect to $(b_{1:T}, x_{1:T}^{(b_{1:T})})$ and $(\tilde{b}_{1:T}, x_{1:T}^{(\tilde{b}_{1:T})})$. □

*Proof of Lemma 6 (i).* Suppose that $\bar{A}^{(1:N)} \sim \rho(\,\cdot \mid g^{(1:N)})$, where $\rho$ is given in (7). We first observe that $\rho$ is unbiased:

$$\mathbb{E}\left[\sum_{i=1}^{N} \mathbf{1}\left(\bar{A}^{(i)} = j\right)\right] = \frac{g^{(j)}}{g^*} + \sum_{i=1}^{N}\left(1 - \frac{g^{(i)}}{g^*}\right)\frac{g^{(j)}}{\sum_{\ell=1}^N g^{(\ell)}} = N\frac{g^{(j)}}{\sum_{\ell=1}^N g^{(\ell)}}. \tag{27}$$

Let $S \in \{1{:}N\}$ be an independent uniformly distributed random variable, and consider $A^{(1:N)} = \bar{A}^{(\sigma_S(1:N))}$, where

$$\sigma_s(i) := [\![ i+s ]\!]_N, \qquad \text{with} \qquad [\![ j ]\!]_N := 1 + (j-1 \mod N)$$

is a cyclic permutation of $1{:}N$. Then $A^{(1:N)} \sim \hat{\rho}(\,\cdot \mid g^{(1:N)})$, where

$$\hat{\rho}(a^{(1:N)} \mid g^{(1:N)}) := \frac{1}{N}\sum_{s=1}^{N} \rho(a^{(\sigma_s(1:N))} \mid g^{(1:N)}), \tag{28}$$

which also clearly unbiased, and from (27), it follows that

$$\mathbb{P}(A^{(k)} = j) = \frac{1}{N}\mathbb{E}\Big[\sum_{i=1}^{N} \mathbf{1}\left(\bar{A}^{(i)} = j\right)\Big] = \frac{g^{(j)}}{\sum_{\ell=1}^N g^{(\ell)}}. \tag{29}$$

Next we derive the conditional distribution of $A^{(-k)}$ given $A^{(k)} = i$. First, because $A^{(k)} = \bar{A}^{(\sigma_S(k))}$, we have

$$\mathbb{P}\left(\sigma_S(k) = j \mid A^{(k)} = i\right) = \frac{\mathbb{P}(\sigma_S(k) = j)\mathbb{P}(\bar{A}^{(j)} = i)}{\sum_{\ell=1}^N \mathbb{P}\left(\sigma_S(k) = \ell\right)\mathbb{P}\left(\bar{A}^{(\ell)} = i\right)} = \frac{\sum_{\ell=1}^N g^{(\ell)}}{N g^{(i)}}\mathbb{P}\left(\bar{A}^{(j)} = i\right),$$

and $\mathbb{P}(\bar{A}^{(j)} = i) = \frac{g^{(i)}}{g^*}\mathbf{1}\left(j = i\right) + \left(1 - \frac{g^{(j)}}{g^*}\right)\frac{g^{(i)}}{\sum_{\ell=1}^N g^{(\ell)}}$, so a simple calculation yields

$$\mathbb{P}(\sigma_S(k) = j \mid A^{(k)} = i) = h(j \mid i), \quad \text{where} \quad h(j \mid i) := \begin{cases} \frac{1}{N}\left(1 + \frac{\sum_{\ell \neq i} g^{(\ell)}}{g^*}\right), & j = i \\ \frac{1}{N}\left(1 - \frac{g^{(j)}}{g^*}\right), & j \neq i. \end{cases} \tag{30}$$

Note that $\sigma_S(k) = j$ is equivalent with $S = [\![ j + (N-k) ]\!]_N$.

We conclude that $A^{(1:N)} \sim \hat{\rho}(\,\cdot \mid g^{(1:N)})$ may be drawn by first drawing $B$ from the marginal distribution of $A^{(k)}$, that is, $\mathbb{P}(B = i) = g^{(i)}/\sum_{\ell=1}^N g^{(\ell)}$, drawing $J \sim h(\,\cdot \mid B)$, setting $S = [\![ J - k ]\!]_N$ and $\bar{A}^{(\sigma_S(k))} = B$ and $A^{(j)} = \bar{A}^{(\sigma_S(j))}$ for $i \in \{1{:}N\}$. □

**Lemma 11.** *Suppose that $\varpi$ is a permutation of $[N]$, and $\varpi_*$ is a cyclic shift of $\varpi$, that is, $\varpi_*(i) = \varpi(\sigma_s(i))$ for some $s \in [N]$, and that*

$$\bar{A}^{1:N} = \varpi(F_{\varpi}^{-1}(U^{1:N}))$$
$$\bar{A}_*^{1:N} = \varpi_*(F_{\varpi_*}^{-1}(U^{1:N})),$$

*where* $U^j = \dfrac{j-1+U}{N}$ *with* $U \sim U(0,1)$.

*Then, it holds that $A^{1:N}$ and $A_*^{1:N}$ have the same distribution, where*

$$A^j = \bar{A}^{\sigma_C(j)}$$
$$A_*^j = \bar{A}_*^{\sigma_C(j)},$$

*and $C \sim U([N])$ is a random shift offset.*

*Proof.* Without loss of generality, we may consider the case $s = 1$ and $\varpi(i) = i$, in which case $\varpi_*(i) = \sigma_1(i)$.

Define $\tilde{U}^i := (U^i - w^1 \mod 1)$, let $j = \arg\min_i \tilde{U}^i$, and let $\tilde{U}_*^i = \tilde{U}^{\sigma_j(i)}$. Observe that $\tilde{U}_*^{1:N}$ and $U^{1:N}$ have the same distribution, so the claim follows once we show that $\bar{A}^{1:N} = F^{-1}(U^{1:N})$ and $\bar{A}_*^{1:N} = \sigma_1(F_{\sigma_1}^{-1}(\tilde{U}_*^{1:N}))$ are equal, up to a cyclic shift. Indeed, we will see that for all $i \in [N]$ and $0 \le k \le N-1$:

$$\bar{A}_*^{\sigma_{-j}(i)} = \sigma_1(F_{\sigma_1}^{-1}(\tilde{U}^i)) = k+1 \iff \bar{A}^i = k+1.$$

Let us first assume $k \geq 1$, then the expression on the left is equivalent to

$$F_{\sigma_1}(k-1) < \tilde{U}^i \le F_{\sigma_1}(k) \iff F(k) < \tilde{U}^i + w^1 \le F(k+1),$$

because $F_{\sigma_1}(\ell) = F(\ell+1) - w^1$. Whenever $U^i - w^1 \geq 0$, we have $\tilde{U}^i + w^1 = U^i$, and the expression on the right simplifies to $\bar{A}^i = F^{-1}(U^i) = k+1$, as desired.

Suppose then that $U^i - w^1 < 0$, in which case $\bar{A}^i = F^{-1}(U^i) = 1$. But then also $\tilde{U}^i = U^i - w^1 + 1 \in (1 - w^1, 1)$, which is equivalent to $F_{\sigma_1}^{-1}(\tilde{U}^i) = N$. □

*Proof of Lemma 6 (ii).* Assume that $\varpi$ is a permutation (such as the mean partition order). Let $I^{1:N} = F_{\tilde{\varpi}}^{-1}(U^{1:N})$, with

$$U^i = \frac{i-1+U}{N},$$

with $U \sim U(0,1)$, that is, standard systematic resampling (Definition 3) with weights $W_{\tilde{\varpi}}^{1:N}$, where $W_{\tilde{\varpi}}^j = W^{\tilde{\varpi}(j)}$ and $\tilde{\varpi}(j) = \varpi(\sigma_{s-1}(j))$, with $s = \varpi^{-1}(i)$.

Hence, $\tilde{\varpi}$ satisfies

$$\tilde{\varpi}(1) = \varpi(\sigma_{s-1}(1)) = \varpi(s) = i. \tag{31}$$

Define $\bar{A}^{1:N}$ such that

$$\bar{A}^j = \tilde{\varpi}(I^j).$$

Then, by Lemma 11, it holds that

$$A^j = \bar{A}^{\sigma_C(j)}, \text{ for } j \in [N], \text{ with } C \sim U([N]),$$

have the same distribution as the indices from systematic resampling with order $\varpi$ that have been shifted by $\sigma_C$. In particular, note that Definition 1 (iii) holds for the latter.

Consider then the count of indices equal to $i$:

$$N^i = \#\{j : A^j = i\}.$$

Since $A^j = i \iff I^{\sigma_C(j)} = 1$ and the indices $I^{1:N}$ are ascending, it holds that

$$N^i = \max\{j \geq 1 \,:\, I^j = 1\},$$

where max is zero in case the set is empty. The event $N^i = n$ is equivalent with

$$\begin{aligned} &\frac{n-1+U}{N} < F_{\tilde{\varpi}}(1) \le \frac{n+U}{N} \\ &\iff n-1+U < Nw^i \le n+U \\ &\iff Nw^i - (n-1) > U \geq Nw^i - n. \end{aligned}$$

We deduce that only two values of $n$ have nonzero probability (for $U \in (0,1)$), since:

$$\begin{aligned} n = \lfloor Nw^i \rfloor &\iff U \in [r, 1) \\ n = \lfloor Nw^i \rfloor + 1 &\iff U \in (0, r), \end{aligned}$$

where $r = Nw^i - \lfloor Nw^i \rfloor$. Furthermore, the conditional probabilities for the events $N^i = n$ are given as:

$$\mathbb{P}(N^i = n \mid A^k = i) = \frac{\mathbb{P}(N^i = n, A^k = i)}{\mathbb{P}(A^k = i)} = \frac{\mathbb{P}(N^i = n, A^k = i)}{w^i},$$

where the numerator satisfies

$$\begin{aligned} \mathbb{P}(N^i = n, A^k = i) &= \sum_{c=1}^{N} \mathbb{P}(C = c, A^k = i \mid N^i = n)\mathbb{P}(N^i = n) \\ &= \sum_{c=1}^{N} \mathbb{P}(A^k = i \mid C = c, N^i = n)\mathbb{P}(C = c \mid N^i = n)\mathbb{P}(N^i = n). \end{aligned}$$

Since

- $\mathbb{P}(N^i = \lfloor Nw^i \rfloor + 1) = r$ and $\mathbb{P}(N^i = \lfloor Nw^i \rfloor) = 1 - r$ (from above),
- $\mathbb{P}(C = c \mid N^i = n) = 1/N$ (because $C$ is independent of $I^{1:N}$ and therefore $N^i$),
- $\mathbb{P}(A^k = i \mid C = c, N^i = n)$ are deterministic, either zero or one, and precisely $n$ are one,

it holds that

$$\mathbb{P}(N^i = \lfloor Nw^i \rfloor + 1 \mid A^k = i) = \frac{(\lfloor Nw^i \rfloor + 1)r}{Nw^i} := p,$$

and

$$\mathbb{P}(N^i = \lfloor Nw^i \rfloor \mid A^k = i) = 1 - p.$$

Observe also that the random variable $U$ conditional on $A^k = i$ and $N^i = n$ has the density $U(0, r)$ if $N^i = \lfloor NW^i \rfloor + 1$ and $U(r, 1)$ if $n = \lfloor NW^i \rfloor$. This follows since $U$ is conditionally independent from the event $A^k = i$ given $N^i = n$, since $U$ only depends on $A^k = i$ through $N^i$. Similarly,

$$\mathbb{P}(C = c \mid A^k = i, N^i = n, U) = \mathbb{P}(C = c \mid A^k = i, N^i = n) = \frac{1}{n}1(\sigma_C(k) \in [1, n]).$$

In practice, we can simulate $C$ from this distribution as follows:

(1) Draw $\bar{C} \sim U\{1, \ldots, n\}$ corresponding to $\sigma_C(k)$ in the above probability,
(2) set $C = \bar{C} - k$,

since $\sigma_C(k) = \sigma_{\bar{C}-k}(k) \in [1, n]$ is equivalent to $\bar{C} \in [1, n]$. □

## Appendix B. Validity of CPF-BBS

We start by two auxiliary results about marginal distributions after partial ancestor tracing and a partial CPF. In what follows, we assume that $G_k(x_{1:k}) = G_k(x_{k-1:k})$ for $k \in \{2{:}T\}$. Using the definition of $\underline{M}_k$ as in Appendix A, let us fix some notation: for $u = 1, \ldots, T$, denote by $\check{\pi}_u^{(N)}(\underline{x}_{1:u}, \underline{a}_{1:u-1}, b_u)$:

$$\frac{1}{\mathcal{Z}}\underline{M}_1(\underline{x}_1)\prod_{k=1}^{u-1}\bigg[\bigg(\frac{1}{N}\sum_{j=1}^{N} G_k(\boldsymbol{x}_k^{(j)})\bigg) r\big(\underline{a}_k \mid G_k(\underline{\boldsymbol{x}}_k)\big)\underline{M}_{k+1}\big(\underline{x}_{k+1} \mid x_k^{(\underline{a}_k)}\big)\bigg]\frac{G_u(\boldsymbol{x}_u^{(b_u)})}{N},$$

and $\eta_{u:T}(x_{u:T}) := \prod_{k=u+1}^{T} M_k(x_k \mid x_{k-1}) G_k(x_{k-1:k})$ with $\eta_{T:T}(x_T) \equiv 1$, then the following define probability distributions for $u = 1, \ldots, T$:

$$\mu_u^{(N)}(\underline{x}_{1:u}, \underline{a}_{1:u-1}, b_u, x^*_{u+1:T}) := \check{\pi}_u^{(N)}(\underline{x}_{1:u}, \underline{a}_{1:u-1}, b_u)\eta_{u:T}(x_u^{(b_u)}, x^*_{u+1:T}).$$

**Lemma 12.** *Suppose that $r^{(p,n)}$ is a valid conditional resampling scheme, with respect to resampling $r$ in Definition 1. Suppose $(\underline{X}_{1:u}, \underline{A}_{1:u-1}, B_u, X^*_{u+1:T}) \sim \mu_u^{(N)}$, and $\ell \in \{1{:}u-1\}$.*

*(i) If $B_{\ell:u-1} \leftarrow \textsc{AncestorTrace}(\underline{A}_{\ell:u-1}, B_u)$, then the marginal density of $\underline{X}_{1:\ell}, \underline{A}_{1:\ell-1}$, $X_{\ell+1:u}^{(B_{\ell+1:u})}$, $B_{\ell:u}$ and $X^*_{u+1:T}$ is*

$$\mu_\ell^{(N)}(\underline{x}_{1:\ell}, \underline{a}_{1:\ell-1}, b_\ell, x_{\ell+1:u}^{(b_{\ell+1:u})}, x^*_{u+1:T})/N^{u-\ell}.$$

*(ii) If further $(X^*_{\ell:u-1}, \tilde{B}_{\ell:u-1}) \leftarrow \textsc{BridgeCPF}(\underline{\boldsymbol{X}}_\ell, B_{\ell:u-1}, X_{\ell:u}^{B_{\ell:u}})$, $\tilde{B}_u = B_u$ and $X^*_u = X_u^{(B_u)}$, then the marginal density of $\underline{X}_{1:\ell}, \underline{A}_{1:\ell-1}$, $X^*_{\ell+1:u}$ and $\tilde{B}_{\ell:u}$ is*

$$\mu_\ell^{(N)}(\underline{x}_{1:\ell}, \underline{a}_{1:\ell-1}, \tilde{b}_\ell, x^*_{\ell+1:T})/N^{u-\ell}.$$

*Proof.* In the case (i), the joint density of all variables may be written as

$$\begin{aligned}
&\check{\pi}_u^{(N)}(\underline{x}_{1:u}, \underline{a}_{1:u-1}, b_u)\left(\prod_{k=\ell}^{u-1} \mathbf{1}\left(b_k = a_k^{(b_{k+1})}\right)\right)\eta_{u:T}(x_u^{(b_u)}, x^*_{u+1:T}) \\
&= \frac{\check{\pi}_\ell^{(N)}(\underline{x}_{1:\ell}, \underline{a}_{1:\ell-1}, b_\ell)}{G_\ell(\boldsymbol{x}_\ell^{(b_\ell)})}\left[\prod_{k=\ell}^{u-1}\left(\frac{1}{N}\sum_{i=1}^{N} G_k(\boldsymbol{x}_k^{(i)})\right)\mathbf{1}\left(b_k = a_k^{(b_{k+1})}\right) r(\underline{a}_k \mid G_k(\underline{\boldsymbol{x}}_k))\right. \\
&\qquad\qquad \left.\underline{M}_{k+1}(\underline{x}_{k+1} \mid x_k^{(\underline{a}_k)})\right] G_u(\boldsymbol{x}_u^{(b_u)})\eta_{u:T}(x_u^{(b_u)}, x^*_{u+1:T}) \\
&= \frac{\check{\pi}_\ell^{(N)}(\underline{x}_{1:\ell}, \underline{a}_{1:\ell-1}, b_\ell)\eta_{\ell:T}(x_{\ell:u}^{(b_{\ell:u})}, x^*_{u+1:T})}{N^{u-\ell}} \prod_{k=\ell}^{u-1} r^{(b_k, b_{k+1})}(\underline{a}_k \mid G_k(\underline{\boldsymbol{x}}_k)) \prod_{i \neq b_{k+1}} M_{k+1}(x_{k+1}^{(i)} \mid x_k^{(a_k^{(i)})}),
\end{aligned}$$

by Lemma 10 (ii). The result (i) follows as we marginalise $x_u^{(i)}$ for $i \neq b_u$, $\underline{a}_{u-1}$, $x_{u-1}^{(i)}$ for $i \neq b_{u-1}$, $\ldots$, $\underline{a}_\ell$.

For (ii), define $\tilde{G}_k^{(\ell,u)}(x_{1:k} \mid x_u) = G_k(x_{k-1:k}) M_{u|\ell}(x_u \mid x_\ell)^{(u-\ell)^{-1}}$, and notice that

$$G_\ell(x_{\ell-1:\ell})\eta_{\ell:T}(x_{\ell:T}) = \left(\prod_{k=\ell+1}^{u} \tilde{G}_{k-1}^{(\ell,u)}(x_{1:k-1} \mid x_u)\bar{M}_k(x_k \mid x_{k-1}, x_u)\right) G_u(x_{u-1:u})\eta_{u:T}(x_{u:T}),$$

where $\bar{M}_u(x_u \mid \cdot, x_u) \equiv 1$. Adding the variables generated in lines 2–7 of Algorithm 8 leads to

$$\begin{aligned}
&\frac{\check{\pi}_\ell^{(N)}(\underline{x}_{1:\ell}, \underline{a}_{1:\ell-1}, b_\ell)}{N^{u-\ell} G_\ell(\boldsymbol{x}_\ell^{(b_\ell)})}\left[\prod_{k=\ell+1}^{u-1} \tilde{G}_{k-1}(\check{\boldsymbol{x}}_{k-1}^{(b_{k-1})} \mid x_u^{(b_u)}) r^{(b_{k-1}, b_k)}\big(\underline{\tilde{a}}_{k-1} \mid \tilde{G}_{k-1}(\underline{\check{\boldsymbol{x}}}_{k-1} \mid x_u^{(b_u)})\big)\right. \\
&\quad \left.\left(\prod_{i=1}^{N} \bar{M}_k(\tilde{x}_k^{(i)} \mid \tilde{x}_{k-1}^{(\tilde{a}_{k-1}^{(i)})}, x_u^{(b_u)})\right)\right] \tilde{G}_{u-1}(\check{\boldsymbol{x}}_{u-1}^{(b_{u-1})} \mid x_u^{(b_u)}) G_u(x_{u-1:u}^{(b_{u-1:u})})\eta_{u:T}(x_u^{(b_u)}, x^*_{u+1:T}),
\end{aligned} \tag{32}$$

where $\underline{\check{\boldsymbol{x}}}_\ell = \underline{x}_\ell$ and $\check{\boldsymbol{x}}_k^{(i)} = (\check{\boldsymbol{x}}_{k-1}^{(\tilde{a}_{k-1}^{(i)})}, \tilde{x}_k^{(i)})$. Thanks to Lemma 10 (ii)

$$\begin{aligned}
&\tilde{G}_{k-1}(\check{\boldsymbol{x}}_{k-1}^{(b_{k-1})} \mid x_u^{(b_u)}) r^{(b_{k-1}, b_k)}\big(\underline{\tilde{a}}_{k-1} \mid \tilde{G}_{k-1}(\underline{\check{\boldsymbol{x}}}_{k-1} \mid x_u^{(b_u)})\big) \\
&= \left(\sum_{i=1}^{N} \tilde{G}_{k-1}(\check{\boldsymbol{x}}_{k-1}^{(i)} \mid x_u^{(b_u)})\right)\mathbf{1}\left(b_{k-1} = \tilde{a}_{k-1}^{(b_k)}\right) r\big(\underline{\tilde{a}}_{k-1} \mid \tilde{G}_{k-1}(\underline{\check{\boldsymbol{x}}}_{k-1} \mid x_u^{(b_u)})\big).
\end{aligned}$$

Because the fraction in (32) does not depend on $b_\ell$, we may now marginalise over $b_\ell, \ldots, b_{u-1}$ and add the distribution of $\tilde{B}_{u-1} \sim \mathrm{Categ}(\tilde{\omega}_{u-1}^{(1:N)})$ where $\tilde{\omega}_{u-1}^{(j)} = \tilde{G}_{u-1}(\check{\boldsymbol{X}}_{u-1}^{(j)}) G_u(\tilde{X}_{u-1}^{(j)}, X_u^{(B_u)})$, leading into

$$\frac{\check{\pi}_\ell^{(N)}(\underline{x}_{1:\ell}, \underline{a}_{1:\ell-1}, b_\ell)}{N^{u-\ell} G_\ell(\boldsymbol{x}_\ell^{(b_\ell)})} \Bigg[ \prod_{k=\ell+1}^{u-1} \Bigg( \sum_{i=1}^N \tilde{G}_{k-1}(\check{\boldsymbol{x}}_{k-1}^{(i)} \mid x_u^{(b_u)}) \Bigg) r\big(\underline{\tilde{a}}_{k-1} \mid \tilde{G}_{k-1}(\underline{\check{\boldsymbol{x}}}_{k-1} \mid x_u^{(b_u)})\big)$$
$$\Bigg( \prod_{i=1}^N \bar{M}_k(\tilde{x}_k^{(i)} \mid \tilde{x}_{k-1}^{(\tilde{a}_{k-1}^{(i)})}, x_u^{(b_u)}) \Bigg) \Bigg] \tilde{G}_{u-1}(\check{\boldsymbol{x}}_{u-1}^{(\tilde{b}_{u-1})} \mid x_u^{(b_u)}) G_u(x_{u-1:u}^{(\tilde{b}_{u-1}, b_u)}) \eta_{u:T}(x_u^{(b_u)}, x_{u+1:T}^*).$$

Introducing $\tilde{b}_{\ell:u-2}$ by AncestorTrace leads to addition of terms $\mathbf{1}\,(\tilde{b}_{k-1} = \tilde{a}_{k-1}^{(b_k)})$. Then, calculations similar as above, but in reverse order, lead to (32) with $b_{\ell:u-1}$ replaced with $\tilde{b}_{\ell:u-1}$. The result follows by marginalising over $\tilde{X}_{\ell+1:u-1}, \tilde{A}_{\ell+1:u-2}$. □

*Proof of Theorem 8.* We start by observing that by Theorem 2, $(\underline{X}_{1:T}, \underline{A}_{T-1}, \tilde{B}_T) \sim \check{\pi}_T^{(N)} = \mu_T^{(N)}$. Then, the proof relies on an iterative application of Lemma 12 (i) and (ii), with $(\ell, u) = (T_{L-1}, T_L), \ldots, (T_1, T_2)$, which concludes that $(\underline{X}_1, X_{2:T}^*, \tilde{B}_{1:T}) \sim \mu_1^{(N)}(\underline{x}_1, \tilde{b}_1, x_{2:T}^*)/N^{T-1}$ so $(\tilde{X}_{1:T}^*, \tilde{B}_{1:T}) \sim \mathcal{Z}^{-1} \eta_{1:T}(\tilde{x}_{1:T})/N^T = \pi(\tilde{x}_{1:T})/N^T$. □

## Appendix C. Computing the conditional distributions in Assumption 7 for discretisations of linear SDEs

The practical application of Algorithm 7 requires for each block the computation of the conditional distributions $M_{u|\ell}$ and $M_{k|k-1,u}$, where $k$ is a time index, and $\ell$ and $u$ refer to the block lower and upper boundaries, respectively. This section discusses how these distributions may be computed when:

- $M_{1:T}$ stem from a discretisation of a linear SDE
- $M_{1:T}$ stem from a discretisation of a linear SDE that is conditioned on a set of noisy linear Gaussian observations.

Note that it is enough to only consider the second case, since the first one may be obtained by omitting the conditioning on the observations (see discussion at the end of this section).

Following [28], the conditional means and variance (matrices) of the SDE (16) are given for $t > s$ by

$$\mathbb{E}[X_t \mid X_s = x_s] = \mathrm{expm}(\mathbf{F}(t-s)) x_s \tag{33}$$

$$\mathrm{Var}[X_t \mid X_s = x_s] = \int_s^t \mathrm{expm}(\mathbf{F}(t-\tau)) \mathbf{K}\mathbf{K}^T \mathrm{expm}(\mathbf{F}(t-\tau))^T \mathrm{d}\tau, \tag{34}$$

where expm denotes the matrix exponential. We introduce the notation

$$T_{s,t} := \mathrm{expm}(\mathbf{F}(t-s)), \; Q_{s,t} := \mathrm{Cov}[X_t \mid X_s = x_s]. \tag{35}$$

Assuming a Gaussian initial distribution, we have:

$$\begin{gathered} X_{t_k} \mid X_{t_{k-1}} = x_{t_{k-1}} \sim N(T_{t_{k-1},t_k} x_{t_{k-1}}, Q_{t_{k-1},t_k}) \\ X_{t_1} \sim N(\mu_{\text{init}}, \Sigma_{\text{init}}), \end{gathered} \tag{36}$$

where the time discretisation corresponds to

$$0 = t_1 < t_2 < \cdots < t_T = \tau \tag{37}$$

as in Section 8, and where $\mu_{\text{init}}$ and $\Sigma_{\text{init}}$ are the initial mean and variance, respectively.

Suppose then that there are observations $\tilde{Y} = (\tilde{Y}_k)_{k=1,\dots,K_y}$ observed at times $\tilde{t}_k$, $k = 1, \dots, K_y$, where each observation time is one of the times in time discretisation (37). Further suppose the $\tilde{Y}_k$ are distributed as

$$\tilde{Y}_k \mid X_{\tilde{t}_k} = x_{\tilde{t}_k} \sim N(Z_k x_{\tilde{t}_k}, H_k), \tag{38}$$

where $Z_k$ and $H_k$ are matrices and observation variances, respectively. We may then define the augmented observations $Y = (Y_k)_{k=1,\dots,T}$ with times (37). The random vector $Y$ is distributed like the observations $\tilde{Y}$ at their respective times, and has missing elements otherwise.

Consider then the joint distribution of $X_{t_k}$, $X_{t_{k-1}}$ and $X_{t_u}$ for $k = \ell+1, \dots, u-1$, conditioned on the observations $Y_{1:T} = y_{1:T}$. Since all variables involved are jointly Gaussian, this conditional distribution is:

$$N\left(\begin{bmatrix} \mu_{k|T} \\ \mu_{k-1|T} \\ \mu_{u|T} \end{bmatrix}, \begin{bmatrix} \Sigma_{k|T} & \Sigma_{k,k-1|T} & \Sigma_{k,u|T} \\ \Sigma_{k-1,k|T} & \Sigma_{k-1|T} & \Sigma_{k-1,u|T} \\ \Sigma_{u,k|T} & \Sigma_{u,k-1|T} & \Sigma_{u|T} \end{bmatrix}\right), \tag{39}$$

where we have used the notation

$$\begin{aligned} \mu_{k|n} &:= \mathbb{E}[X_{t_k} \mid Y_{1:n} = y_{1:n}], \\ \Sigma_{k|n} &:= \mathrm{Var}[X_{t_k} \mid Y_{1:n} = y_{1:n}], \\ \Sigma_{p,s|n} &:= \mathrm{Cov}[X_{t_p}, X_{t_s} \mid Y_{1:n} = y_{1:n}]. \end{aligned}$$

Here, conditioning on a missing observation should be understood as the observation being removed from the condition.

To obtain the (cross)covariances in (39), the following backwards recursion for $s = t-1, t-2, \dots$ from [8] may be used (with a matrix transpose applied to the result as needed):

$$\Sigma_{s,t|T} = \Sigma_{s|s} T^T_{t_s,t_{s+1}} \Sigma^{-1}_{s+1|s} \Sigma_{s+1,t|T}. \tag{40}$$

An inspection of Equations (39) and (40) reveals that all the quantities required are computed routinely by the Kalman filter and smoother [cf. 14] applied to the linear Gaussian state space model composed of (36) and (38). Note that the Kalman filter automatically handles any missing values in the observation sequence.

For $k = \ell + 1$, by elementary properties of the Gaussian distribution, the distribution $M_{u|\ell}$, that is $X_{t_u} \mid X_{t_\ell} = x_{t_\ell}, Y_{1:T} = y_{1:T}$, is

$$\begin{aligned} N\Big(&\mu_{u|T} + \Sigma^T_{\ell,u|T}\Sigma^{-1}_{\ell|T}(x_{t_\ell} - \mu_{\ell|T}), \\ &\Sigma_{u|T} - \Sigma^T_{\ell,u|T}\Sigma^{-1}_{\ell|T}\Sigma_{\ell,u|T}\Big). \end{aligned} \tag{41}$$

Similarly, for $k = \ell+1, \dots, u-1$, the distribution $M_{k|k-1,u}$, that is, $X_{t_k} \mid X_{t_{k-1}} = x_{t_{k-1}}, X_{t_u} = x_{t_u}, Y_{1:T} = y_{1:T}$, is

$$\begin{aligned} N\Bigg(&\mu_{k|T} + \Sigma_{k,(k-1,u)|T}\Sigma^{-1}_{(k-1,u)|T}((x_{t_{k-1}}\ x_{t_u})^T - \mu_{(k-1,u)|T}), \\ &\Sigma_{k|T} - \Sigma_{k,(k-1,u)|T}\Sigma^{-1}_{(k-1,u)|T}\Sigma^T_{k,(k-1,u)|T}\Bigg), \end{aligned} \tag{42}$$

where

$$\begin{aligned}\mu_{(k-1,u)|T} &:= (\mu_{k-1|T} \quad \mu_{u|T})^T, \\ \Sigma_{k,(k-1,u)|T} &:= [\Sigma_{k,k-1|T} \quad \Sigma_{k,u|T}], \\ \Sigma_{(k-1,u)|T} &:= \begin{bmatrix} \Sigma_{k-1|T} & \Sigma_{k-1,u|T} \\ \Sigma_{u,k-1|T} & \Sigma_{u|T} \end{bmatrix}.\end{aligned} \tag{43}$$

In the case where $M_{1:T}$ simply corresponds to a discretisation of the linear SDE (16), the above computations can be repeated with the conditioned means, variances and covariances replaced with their unconditional counterparts. In practice, an easy way to compute the unconditional means and variances is to set all observations missing in the Kalman filter. The unconditional covariances can then be obtained from (40) as before.

## Appendix D. Models

This section gives additional details related to the models appearing in Section 9.

### D.1. **CTCRW-P.**

The CTCRW-P SDE (20) may be placed into the form of the linear SDE (16) by setting

$$X_t = (V_t \; L_t)^T, \; \mathbf{F} = \begin{bmatrix} -\beta_v & 0 \\ 1 & -\beta_x \end{bmatrix} \text{ and } \mathbf{K} = \begin{bmatrix} \sigma & 0 \\ 0 & 0 \end{bmatrix}.$$

The expressions for $T_{s,t}$ and $Q_{s,t}$ in (35) are given as follows. A direct computation yields

$$\operatorname{expm}(Ft) = \begin{bmatrix} \exp(-\beta_v t) & 0 \\ \dfrac{\exp(-\beta_x t) - \exp(-\beta_v t)}{\beta_v - \beta_x} & \exp(-\beta_x t) \end{bmatrix}, \tag{44}$$

when $\beta_v \neq \beta_x$. If $\beta_v = \beta_x$, the first element of the second row is replaced by $t \exp(-\beta_v t)$. The transition matrix $T_{s,t}$ may be obtained from (44) by substituting $t - s$ for $t$.

If $\beta_v \neq \beta_x$, the elements $q_{ij}, 1 \leq i, j \leq 2$ of $Q_{s,t}$ are given by

$$\begin{aligned} &q_{11} = \frac{\sigma^2}{2\beta_v}(1 - \exp(-2\beta_v(t-s))) \\ &q_{12} = q_{21} = \frac{\sigma^2}{\beta_v - \beta_x}\left[\frac{1}{\beta_v + \beta_x}(1 - \exp(-(\beta_v + \beta_x)(t-s))) - \frac{1}{2\beta_v}(1 - \exp(-2\beta_v(t-s)))\right] \\ &q_{22} = \frac{\sigma^2}{(\beta_v - \beta_x)^2}\left[\frac{1}{2\beta_x}(1 - \exp(-2\beta_x(t-s))) + \frac{1}{2\beta_v}(1 - \exp(-2\beta_v(t-s))) \right. \\ &\qquad \left. - \frac{2}{\beta_x + \beta_v}(1 - \exp(-(\beta_x + \beta_v)(t-s)))\right]. \end{aligned} \tag{45}$$

If $\beta_v = \beta_x$, the element $q_{11}$ remains as in (45), but the elements $q_{12}$ and $q_{22}$ become:

$$\begin{aligned} q_{12} &= \frac{\sigma^2}{4\beta_v^2}\left[1 + \exp\Big(-2\beta_v(t-s)\Big)\Big(-2\beta_v(t-s) - 1\Big)\right], \\ q_{22} &= \frac{\sigma^2}{4\beta_v^3}\left[1 - \exp\Big(-2\beta_v(t-s)\Big)\Big(1 + 2\beta_v(t-s)(\beta_v(t-s) + 1)\Big)\right]. \end{aligned} \tag{46}$$

Finally, the stationary covariance matrix $S$ with elements $s_{ij}$ used in the initial distribution of the CTCRW-P model is obtained by taking the limit $(t-s) \to \infty$ in the previous equations:

$$\begin{aligned} s_{11} &= \frac{\sigma^2}{2\beta_v} \\ s_{12} = s_{21} &= \frac{\sigma^2}{\beta_v - \beta_x}\left[\frac{1}{\beta_v + \beta_x} - \frac{1}{2\beta_v}\right] \\ s_{22} &= \frac{\sigma^2}{(\beta_v - \beta_x)^2}\left[\frac{1}{2\beta_x} + \frac{1}{2\beta_v} - \frac{2}{\beta_x + \beta_v}\right], \end{aligned} \tag{47}$$

when $\beta_v \neq \beta_x$. When $\beta_v = \beta_x$, the elements $s_{12}$ and $s_{22}$ are

$$s_{12} = \frac{\sigma^2}{4\beta_v^2} \quad \text{and} \quad s_{22} = \frac{\sigma^2}{4\beta_v^3}. \tag{48}$$

D.2. **CP-RBM.** The density of the reflected normal distribution $N^{(r)}(\mu, \sigma^2, a, b)$, for any point $x$ in the support $(a, b)$, is given by

$$N^{(r)}(x; \mu, \sigma^2, a, b) = N(x; \mu, \sigma^2) + \sum_{k=1}^{\infty} N(g_a^{(k)}(x); \mu, \sigma^2) + N(g_b^{(k)}(x); \mu, \sigma^2), \tag{49}$$

where

$$\begin{aligned} g_a^{(k)}(x) &:= (-1)^k x + ka - kb + (a+b)\mathbf{1}\,(\text{k odd}) \\ g_b^{(k)}(x) &:= (-1)^k x + kb - ka + (a+b)\mathbf{1}\,(\text{k odd})\,. \end{aligned} \tag{50}$$

Equation (49) may be derived by noting that the density of any point $x \in (a, b)$ is equal to the sum of normal densities at points that (eventually) reflect to $x$. These points consist of $x$ itself and the reflection points outside $(a, b)$ given by the sequences in (50). In practice, we truncate the infinite sum in (49) to the first 10 terms, which provides a reasonable approximation for the values of $\sigma$, $a$ and $b$ we use.

D.3. **CTCRW.** The CTCRW SDE can be placed in the form of the linear SDE (16) by setting

$$X_t = (V_t\ L_t)^T, \ \mathbf{F} = \begin{bmatrix} -\beta & 0 \\ 1 & 0 \end{bmatrix} \text{ and } \mathbf{K} = \begin{bmatrix} \sigma & 0 \\ 0 & 0 \end{bmatrix}.$$

The expressions for $T_{s,t}$ and $Q_{s,t}$ in (35) are then given as follows.

$$T_{s,t} = \begin{bmatrix} \exp\left(-(t-s)\beta\right) & 0 \\ \dfrac{1 - \exp\left(-(t-s)\beta\right)}{\beta} & 1 \end{bmatrix}, \tag{51}$$

and the matrix $Q_{s,t}$ has elements $q_{ij}$, $i, j = 1, 2$, such that

$$\begin{aligned} q_{11} &= \frac{\sigma^2}{2\beta}\left(1 - \exp\left(-(t-s)2\beta\right)\right), \\ q_{21} = q_{12} &= \frac{\sigma^2}{2\beta^2}\left(1 - 2\exp\left(-(t-s)\beta\right) + \exp\left(-(t-s)2\beta\right)\right), \\ q_{22} &= \frac{\sigma^2}{\beta^2}\left((t-s) - \frac{2}{\beta}\left(1 - \exp\left(-(t-s)\beta\right)\right) + \frac{1}{2\beta}\left(1 - \exp\left(-(t-s)2\beta\right)\right)\right). \end{aligned} \tag{52}$$

## Appendix E. Miscellaneous algorithms

### E.1. Algorithm for finding mean partition order of weights.

The following algorithm finds a mean partition order $I$ of the input weights $w^{1:N}$. Note that the algorithm does not modify the weights $w^{1:N}$ and that the operation 'break' means exiting from the current (innermost) 'while' loop.

**Algorithm 12** MeanPartitionOrder($w^{1:N}$)

```
Set $p = \text{Mean}(w^{1:N})$ ('pivot').
Set $i_\ell = 0$ and $i_u = N + 1$.
Initialise $I$ as the index set $[N]$.
while True do
    while $i_\ell < \min(i_u, N)$ do
        Set $i_\ell = i_\ell + 1$.
        Break if $w^{I^{(i_\ell)}} > p$.
    end while
    while $i_u > i_\ell$ do
        Set $i_u = i_u - 1$.
        Break if $w^{I^{(i_u)}} < p$.
    end while
    Break if $i_\ell$ equals $i_u$.
    Swap indices $i_\ell$ and $i_u$ of $I$.
end while
output $I$
```

### E.2. Algorithm for constructing dyadic blocking sequences.

**Algorithm 13** DyadicCandidateBlockings($T \in \{2, 3, \ldots\}$)

```
Denote by $p^*$ the largest $p$ such that $2^p + 1 \leq T$.
for $i = 1, \ldots, p^* + 1$ do
    Set blocksize = $2^{i-1}$
    Set $\ell = 1; u = 0$.
    Set $k = 0$ (block index)
    while $l < T$ do
        Set $k = k + 1$
        Set $u = \ell +$ blocksize
        Set $T_k^{(i)} = \ell; T_{k+1}^{(i)} = \min(u, T)$
        Set $\ell = u$
    end while
end for
return Candidate blocking sequences $T_{1:L^{(i)}}^{(i)}$ for $i = 1, 2, \ldots, p^* + 1$
```

## Appendix F. Derivation of PLU$_\text{G}$

Consider the following artificial conditional particle system that approximates a continuous time conditional particle filter with near constant weights:

- The system has $N$ particles.
- One of the particles corresponds to the 'reference', which can not die.
- At most one resampling event occurs at any time $k$, with probability $p_R^{(k)}$.
- If a resampling event occurs:

  - a dying particle is chosen uniformly among the $N-1$ particles (excluding the reference).
  - a particle is selected for 'reproduction' uniformly among the $N-1$ particles (excluding the dying particle).
- If a resampling event does not occur, no particles die or reproduce.

Further suppose that the particle population is divided into two groups, 'ill' and 'healthy', where the ill population is to be interpreted as the particles having reproduced from the reference or any of its descendants. Denote by $H_k$ and $I_k := N - H_k$ the number of healthy and number of ill (including reference) at time $k$, respectively. Initially, $H_1 = N-1$.

**Theorem 13.** *For the artificial particle system of this section, it holds for any $T \geq 1$ that*

$$\mathbb{E}[H_T] = (N-1)\prod_{k=1}^{T-1}\left(1 - \frac{p_R^{(k)}}{(N-1)^2}\right). \tag{53}$$

*Proof.* For any $k \geq 2$ we have

$$H_k \mid H_{k-1} = \begin{cases} H_{k-1}+1, & \text{prob. } p_{\text{increase}}^{(k-1)} \\ H_{k-1}-1, & \text{prob. } p_{\text{decrease}}^{(k-1)} \\ H_{k-1}, & \text{prob. } p_{\text{nothing}}^{(k-1)}, \end{cases}$$

where

$$\begin{aligned} p_{\text{increase}}^{(k-1)} &= p_R^{(k-1)}\frac{I_{k-1}-1}{N-1}\cdot\frac{H_{k-1}}{N-1} \text{ (resampling occurs, ill dies, healthy reproduces)} \\ p_{\text{decrease}}^{(k-1)} &= p_R^{(k-1)}\frac{H_{k-1}}{N-1}\cdot\frac{I_{k-1}}{N-1} \text{ (resampling occurs, healthy dies, ill reproduces)}, \\ p_{\text{nothing}}^{(k-1)} &= 1 - p_{\text{increase}}^{(k-1)} - p_{\text{decrease}}^{(k-1)}. \end{aligned}$$

Therefore,

$$\begin{aligned} \mathbb{E}[H_T \mid H_{T-1}] &= H_{T-1} + \mathbb{E}[H_T - H_{T-1} \mid H_{T-1}] \\ &= H_{T-1} + p_R^{(T-1)}\frac{I_{T-1}-1}{N-1}\cdot\frac{H_{T-1}}{N-1} - p_R^{(T-1)}\frac{H_{T-1}}{N-1}\cdot\frac{I_{T-1}}{N-1} \\ &= \Big(1 - \frac{p_R^{(T-1)}}{(N-1)^2}\Big)H_{T-1}, \end{aligned}$$

and

$$\mathbb{E}[H_T] = \mathbb{E}[\mathbb{E}[H_T \mid H_{T-1}]] = \Big(1 - \frac{p_R^{(T-1)}}{(N-1)^2}\Big)\mathbb{E}[H_{T-1}],$$

which yields (53) by repeated application. □

The direct consequence of this result is that $\text{PLU}_\text{G}(\ell, u)$ equals $\mathbb{E}[H_u/N]$ with $p_R^{(k)} = p_k N$ (defined in Section 7) and $\ell$ considered as the 'first' time point.

## Appendix G. Supplementary figures

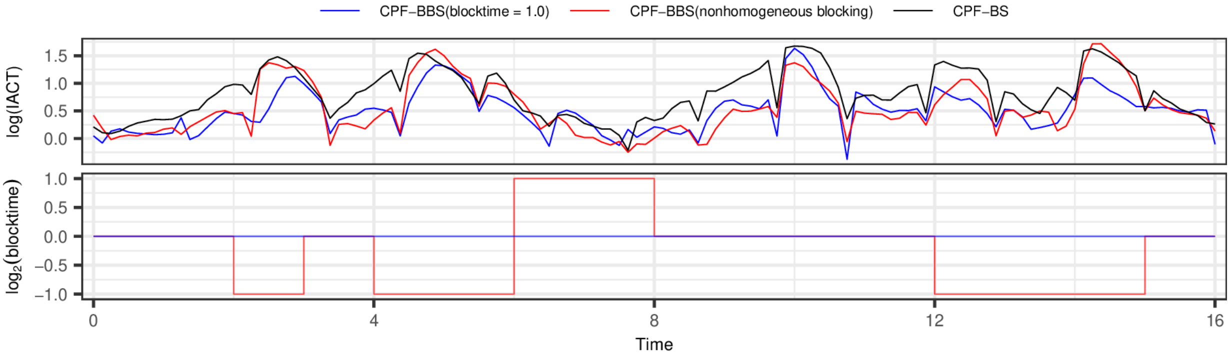


Figure 7. The logarithmic IACT (top) of the state $L_t^{(x)}$ in the CTCRW-T model with $|\Delta_k| = 0.125$ for CPF-BS and the CPF-BBS with blocktime 1.0 and the blocking from Algorithm 11 (bottom).

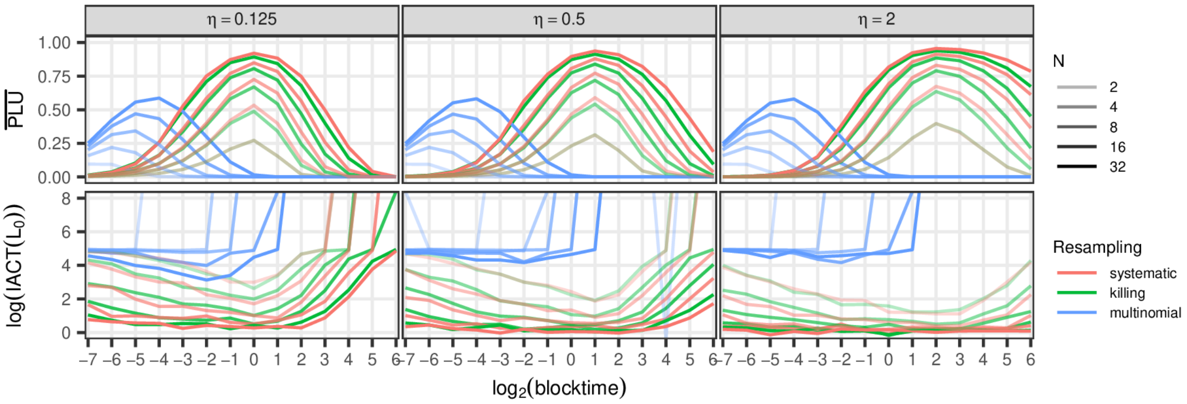


Figure 8. The estimated mean PLUs and the logarithm of IACT with varying $\eta$ for the location state variable at time 0.0 in the CTCRW-P model. The value of $|\Delta_k|$ was set to $2^{-7}$. The performance of CPF-BS is seen at the far left, with blocktime $= 2^{-7}$.