institute of telecommunications

TU WIEN

Diplomarbeit

Bayesian Nonparametric Inference in State-Space Models with an Application to Extended Target Tracking

ausgeführt zur Erlangung des akademischen Grades
eines Diplom-Ingenieurs unter der Leitung von

Ao. Univ. Prof. Dipl.-Ing. Dr. techn. Franz Hlawatsch
Dipl.-Ing. Dr. techn. Günther Koliander
Dipl.-Ing. Rene Repp

am

Institute of Telecommunications

eingereicht an der Technischen Universität Wien
Fakultät für Elektrotechnik und Informationstechnik

von

Bernd Kreidl
1030519

Wien, 13.01.2021 _____

Abstract

Tracking a hidden state, such as the position and velocity of a target based on noisy measurements, is a common problem in statistical signal processing. There are scenarios where multiple measurements are generated per target and time step, which enable the additional estimation of the target's size, shape, and orientation. This problem is known as extended target tracking. For some extended target tracking scenarios, the targets can be grouped into several target classes where the targets in each class are defined by the same parameters. These parameters typically describe the target's size, shape, and/or dynamic behavior. If the classes and their parameters are known, the class affiliation and, thereby, the parameters of each target can be inferred using classification.

In this thesis, however, we propose joint tracking and clustering of extended targets for the case where the number of classes and the class parameters are unknown. Instead of estimating the parameters for each target separately, clustering enables more accurate parameter estimation by considering several targets jointly. We define a statistical model for joint tracking and clustering of extended targets in which the target classes and their parameters are described by a Dirichlet process and the target states given the target parameters by a state-space model. As our statistical model is a special case of a Bayesian nonparametric state-space model, we develop two Monte Carlo algorithms for inference in Bayesian nonparametric state-space models, one for batch processing and one for sequential processing. We present simulation results demonstrating the convergence of these algorithms and the performance gain due to joint tracking and clustering of extended targets compared to tracking without clustering. We observe a slight improvement for tracking and a significant improvement for parameter estimation. Thus, we conclude that joint tracking and clustering of extended targets can improve parameter estimation and tracking for scenarios with an inherent but unknown class structure.

Kurzfassung

Die Verfolgung eines unbeobachteten Zustands, insbesondere der Position und Geschwindigkeit eines Ziels aufgrund von verrauschten Messwerten, ist ein wichtiges Problem der statistischen Signalverarbeitung. In manchen Szenarien ruft ein Ziel mehrere Messwerte pro Zeitschritt hervor, welche die zusätzliche Schätzung der Größe, Form und Orientierung eines Ziels ermöglichen. Dieses Problem wird als Verfolgung ausgedehnter Ziele bezeichnet. In manchen Fällen können die ausgedehnten Ziele in Klassen eingeteilt werden, wobei die Ziele jeder Klasse durch dieselben Parameter definiert sind. Diese Parameter beschreiben häufig die Größe, die Form und/oder das dynamische Verhalten der Ziele. Wenn die Klassen und ihre Parameter bekannt sind, können die Klassenzuordnung und daher auch die Parameter der Ziele durch Klassifikation bestimmt werden.

In dieser Arbeit behandeln wir hingegen die gemeinsame Verfolgung und Clusterung ausgedehnter Ziele für den Fall, dass die Anzahl der Klassen sowie deren Parameter unbekannt sind. Anstatt die Parameter für jedes Ziel einzeln zu schätzen, können durch Clusterung die Parameter mehrerer Ziele gemeinsam und somit besser geschätzt werden. Wir definieren ein statistisches Modell zur gemeinsamen Verfolgung und Clusterung ausgedehnter Ziele, bei dem wir die Klassen und Parameter durch einen Dirichlet-Prozess und die Ziel-Zustände konditioniert auf ihre Parameter durch ein Zustandsraummodell beschreiben. Da unser statistisches Modell ein Spezialfall eines Bayesschen nicht-parametrischen Zustandsraummodells ist, entwickeln wir zwei Monte Carlo-Algorithmen zur Inferenz in Bayesschen nicht-parametrischen Zustandsraummodellen. Mittels Simulationen untersuchen wir das Konvergenzverhalten der Algorithmen und die Verbesserung der Ergebnisse, die durch die gemeinsame Verfolgung und Clusterung ausgedehnter Ziele im Vergleich zur Verfolgung ohne Clusterung erzielt werden kann. Dabei stellen wir eine leichte Verbesserung der Verfolgung und eine signifikante Verbesserung der Parameterschätzung fest. Unsere Simulationen zeigen somit, dass durch die gemeinsame Verfolgung und Clusterung ausgedehnter Ziele die Parameterschätzung und Verfolgung in Szenarien mit einer inhärenten, aber unbekannten Klassenstruktur verbessert werden kann.

Acknowledgments

I am profoundly grateful to my supervisors, Ao. Univ. Prof. Franz Hlawatsch, Dr. techn. Günther Koliander, and Dipl.-Ing. Rene Repp, for their continuous guidance, valuable feedback, and helpful insights. Furthermore, I would like to thank Dipl.-Ing. Thomas John Bucco and Dipl.-Ing. Thomas Kropfreiter for their advice and assistance.

Finally, I am deeply indebted to my family and friends. This thesis would not have been possible without their unconditional support and encouragement.

Contents

List of Figures

List of Abbreviations

BNP	Bayesian nonparametric
CG	clustering gain
CRP	Chinese restaurant process
DP	Dirichlet process
DPM	Dirichlet process mixture
i.i.d.	independent and identically distributed
IS	importance sampling
MC	Monte Carlo
MCMC	Markov chain Monte Carlo
MH	Metropolis-Hastings
MSE	mean square error
pdf	probabilty density function
PF	particle filter
PGAS	particle Gibbs sampler with ancestor sampling
PMCMC	particle Markov Chain Monte Carlo
RMPF	resample-move particle filter
SMC	sequential Monte Carlo
TS	time series

Chapter 1

Introduction

1.1 Motivation

An important task in statistical signal processing is target tracking, where the goal is to track the state of a target or the states of multiple targets using noisy measurements [1]. One exemplary application is to track the positions and velocities of multiple objects using measurements of a radar system [2]. Typically, target tracking considers at most one measurement per target and time step. However, modern radar systems with increased resolution may produce more than one measurement per target and time step. Besides tracking the state, these multiple measurements may also be used to estimate the size, shape, and orientation of each target. This problem is known as extended target tracking [3] and is, for example, relevant to marine vessel tracking using X-band radar [4] and car tracking using lidar [5].

Certain extended target tracking scenarios can be modeled as a parameter dependent state-space model [6], where each target has random parameters that define the target's state-space model. The parameters can describe the size, shape, expected number of measurements, and/or dynamic behavior of each target. Furthermore, if multiple targets are tracked, it can be reasonable to assume that some targets share the same parameters. According to these parameters, the targets can then be grouped into classes. If the parameters of each class are known, then classification can be performed by assigning each target to one of these classes [7, 8].

In this thesis, however, we consider the case, where we only know that some classes exist but the parameters of the individual classes are unknown. Here, we can still exploit this limited knowledge by clustering the targets and estimating the parameters of each class in addition to tracking the targets' states. Estimating the class parameters can be

expected to be more accurate than estimating the parameters for each target separately as we can use all measurements of all targets in each class instead of only the measurements of each individual target. Additionally, improved parameter estimation may also result in improved state estimation, as the target's state depends on the parameters.

An elegant approach to clustering in this context is to perform inference in a combined statistical model for the targets and their classes, which immediately enables joint tracking and clustering of extended targets. In this thesis, we define a Bayesian nonparametric (BNP) state-space model for joint tracking and clustering of extended targets, where the prior distribution of the targets' parameters (with an inherent class structure) is a Dirichlet process (DP) and the time-varying target states given the target parameters are distributed according to a state-space model. Note that the DP defines both the target classes and the class parameters. The main advantage of the DP is that it considers an unknown and random number of classes, which allows clustering without specifying the number of classes in advance. Further, we develop two Monte Carlo algorithms for inference in BNP state-space models, one for batch processing and one for sequential processing. We investigate the convergence of these algorithms and the performance gain due to joint tracking and clustering of extended targets compared to tracking without clustering.

1.2 State of the Art

A problem related to clustering is classification, where one assigns targets to predefined classes with known parameters. There is some previous work on joint tracking and classification [7,8] and on clustering after tracking [9]. However, to the best of our knowledge, joint tracking and clustering has not been considered so far. A parameter dependent state-space model for extended target tracking, which is related to a part of our statistical model for joint tracking and clustering of extended targets, was proposed in [6].

For clustering in our context, BNP statistical models [10], in particular the DP [11] [10, Chapter 4] and the Dirichlet process mixture (DPM) [12] [10, Chapter 5], are of special interest. Clustering of time series defined by state-space models and a DPM or a related statistical model is discussed in [13–19]. The statistical model for joint tracking and clustering of extended targets proposed in this thesis is a special case of a BNP state-space model [13].

The algorithm proposed in [13] is a batch processing algorithm that is used for inference

on the class parameters and the targets' class assignments, but not on the targets' states. However, as inference on the targets' states is required for tracking, we introduce a Markov chain Monte Carlo (MCMC) algorithm for batch processing that uses particle Markov Chain Monte Carlo (PMCMC) [20, 21] to sample the targets' states and Gibbs sampling [22, 23] to sample the class parameters and the targets' class assignments. Additionally, we introduce a sequential Monte Carlo (SMC) algorithm for sequential processing based on the resample-move particle filter (RMPF) [24–26] and sequential Markov chain Monte Carlo (SMCMC) [27–29].

1.3 Thesis Outline

After this introductory chapter, we continue this thesis by discussing MC methods in Chapter 2. More specifically, we present the general principle behind MC methods, as well as various SMC and MCMC algorithms.

In Chapter 3, we describe the basics of BNP statistical models with a focus on the DP and the DPM.

In Chapter 4, we propose two algorithms for inference in BNP state-space models. The first algorithm is an MCMC algorithm for batch processing based on PMCMC. The second algorithm is a sequential algorithm based on the RMPF and SMCMC.

In Chapter 5, we introduce a BNP state-space model for extended target tracking and apply the inference algorithms of Chapter 4. Furthermore, we evaluate the convergence of the algorithms and investigate the performance gain due to joint tracking and clustering.

Finally, Chapter 6 concludes this thesis by summarizing our contributions and suggesting future research directions.

Chapter 2

Monte Carlo Methods

This chapter gives a brief introduction to Monte Carlo (MC) methods [30]. MC methods are numerical methods that play an important role in many different fields, such as science, engineering, and finance. While there is a wide range of MC methods, we will focus on importance sampling (IS), sequential MC (SMC), and Markov chain MC (MCMC). We will discuss the general principle of these methods and their application to (parameter dependent) state-space models. This chapter is based on the introductory paper [31] and the book [30] unless stated otherwise.

While we will limit our discussion to real and continuous random vectors, every algorithm in this chapter can also be adapted to other types of random variables by replacing the probabilty density functions (pdfs), denoted by f, with the corresponding type of density and, if appropriate, replacing the integrals by sums. In this thesis, we will also consider discrete random variables, where we use probability mass functions, denoted by p, and mixed random variables, where we use mixed densities, denoted by v.

2.1 Motivation

To illustrate the general principle of MC methods, consider a random vector $\mathbf{x} \in \mathbb{R}^X$. We will call its pdf, denoted as $f_{\mathbf{x}}(\boldsymbol{x})$, the target pdf. Considering $K \in \mathbb{N}$ independent and identically distributed (i.i.d.) random vectors $\mathbf{x}^{(k)}$ with pdf $f_{\mathbf{x}}(\boldsymbol{x})$ and $k \in \{1, \ldots, K\}$, we can "approximate"[1] $f_{\mathbf{x}}(\boldsymbol{x})$ with

$$\hat{\mathsf{f}}_{\mathbf{x}}^{(K)}(\boldsymbol{x}) = \frac{1}{K} \sum_{k=1}^{K} \delta_{\mathbf{x}^{(k)}}(\boldsymbol{x}), \tag{2.1}$$

[1] Approximate in the sense that the distribution defined by $\hat{\mathsf{f}}_{\mathbf{x}}^{(K)}(\boldsymbol{x})$ converges to the distribution defined by $f_{\mathbf{x}}(\boldsymbol{x})$ as $K \to \infty$, but not pointwise convergence of $\hat{\mathsf{f}}_{\mathbf{x}}^{(K)}(\boldsymbol{x})$ to $f_{\mathbf{x}}(\boldsymbol{x})$.

where $\delta_{\boldsymbol{x}^{(k)}}(\boldsymbol{x})$ is the Dirac delta defined such that for any set $\mathcal{A} \subseteq \mathbb{R}^X$ and continuous function[2] $f(\boldsymbol{x})$

$$\int_{\mathcal{A}} f(\boldsymbol{x})\delta_{\boldsymbol{x}^{(k)}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = \begin{cases} f(\boldsymbol{x}^{(k)}) & \text{if } \boldsymbol{x}^{(k)} \in \mathcal{A} \\ 0 & \text{else} \end{cases}. \tag{2.2}$$

Note that $\hat{\mathsf{f}}_{\mathbf{x}}^{(K)}(\boldsymbol{x})$ is random because the $\mathbf{x}^{(k)}$ are random. To approximate an expectation

$$I = \mathbb{E}(h(\mathbf{x})) = \int_{\mathbb{R}^X} h(\boldsymbol{x}) f_{\mathbf{x}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}, \tag{2.3}$$

where h is an arbitrary function, we can calculate the expectation with respect to the "approximation" $\hat{\mathsf{f}}_{\mathbf{x}}^{(K)}(\boldsymbol{x})$ of the target pdf, that is,

$$\hat{\mathsf{I}}^{(K)} = \int_{\mathbb{R}^X} h(\boldsymbol{x})\hat{\mathsf{f}}_{\mathbf{x}}^{(K)}(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = \frac{1}{K}\sum_{k=1}^{K} h(\mathbf{x}^{(k)}). \tag{2.4}$$

By the strong law of large numbers, under mild conditions, we have $\hat{\mathsf{I}}^{(K)} \xrightarrow{\text{a.s.}} I$, that is, $\hat{\mathsf{I}}^{(K)}$ converges almost surely to I as $K \to \infty$. This shows that if we can generate K samples $\boldsymbol{x}^{(k)}$ from the target pdf, then we can approximate the often intractable integral in (2.3) by the tractable sum $\hat{I}^{(K)} = \frac{1}{K}\sum_{k=1}^{K} h(\boldsymbol{x}^{(k)})$. While sampling directly from the target pdf is difficult in many cases, we can use the methods discussed in the following sections to generate samples. These methods allow us to "approximate" the target pdf $f_{\mathbf{x}}(\boldsymbol{x})$ similarly to (2.1), and the expectation (2.3) similarly to (2.4).

2.2 Importance Sampling

We can restate (2.3) as

$$I = \int_{\mathbb{R}^X} h(\boldsymbol{x})w(\boldsymbol{x})g(\boldsymbol{x})\mathrm{d}\boldsymbol{x}, \tag{2.5}$$

where

$$w(\boldsymbol{x}) = \frac{f_{\mathbf{x}}(\boldsymbol{x})}{g(\boldsymbol{x})} \tag{2.6}$$

and $g(\boldsymbol{x})$ is an arbitrary pdf, with a support that includes the support of $f_{\mathbf{x}}(\boldsymbol{x})$, that is, $\operatorname{supp} g \supset \operatorname{supp} f_{\mathbf{x}}$. We will call $g(\boldsymbol{x})$ the proposal pdf. According to (2.5), I is represented as the expectation of $h(\boldsymbol{x})w(\boldsymbol{x})$ with respect to the distribution defined by the proposal

[2] Note that this definition can be made rigorous by interpreting $\delta_{\boldsymbol{x}^{(k)}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$ as the Dirac measure at the point $\boldsymbol{x}^{(k)}$.

pdf $g(\boldsymbol{x})$. Therefore, we can approximate (2.3) by considering i.i.d. random vectors $\mathbf{x}^{(k)}$ with pdf $g(\boldsymbol{x})$ for all $k \in \{1, \dots, K\}$, and calculating

$$\hat{\mathsf{I}}^{(K)} = \frac{1}{K} \sum_{k=1}^{K} w(\mathbf{x}^{(k)}) h(\mathbf{x}^{(k)}). \tag{2.7}$$

By the strong law of large numbers, under mild conditions, approximation (2.7) satisfies $\hat{\mathsf{I}}^{(K)} \xrightarrow{\text{a.s.}} I$.

It can be shown that the choice of the proposal pdf $g(\boldsymbol{x})$ that minimizes the variance of the approximation (2.7) is [30, p. 95]

$$g^*(\boldsymbol{x}) = \frac{|h(\boldsymbol{x})| f_{\mathbf{x}}(\boldsymbol{x})}{\int_{\mathbb{R}^X} |h(\boldsymbol{x}')| f_{\mathbf{x}}(\boldsymbol{x}') \mathrm{d}\boldsymbol{x}'}. \tag{2.8}$$

Therefore, using $g(\boldsymbol{x}) = f_{\mathbf{x}}(\boldsymbol{x})$, that is, directly sampling from the target pdf $f_{\mathbf{x}}(\boldsymbol{x})$, is in general not optimal in terms of approximation variance. However, as the optimal pdf $g^*(\boldsymbol{x})$ can be difficult to sample from, it is usually preferable to choose a proposal pdf $g(\boldsymbol{x})$ such that it is easy to sample from and that $g(\boldsymbol{x})$ is large where $|h(\boldsymbol{x})| f_{\mathbf{x}}(\boldsymbol{x})$ is large.

If we are only able to evaluate the target pdf up to a normalizing constant, we can still use the approximation

$$\hat{\mathsf{I}}^{(K)} = \sum_{k=1}^{K} \mathsf{W}^{(k)} h(\mathbf{x}^{(k)}), \tag{2.9}$$

where

$$\mathsf{W}^{(k)} \propto w(\mathbf{x}^{(k)}) = \frac{f_{\mathbf{x}}(\mathbf{x}^{(k)})}{g(\mathbf{x}^{(k)})} \tag{2.10}$$

with

$$\sum_{k=1}^{K} \mathsf{W}^{(k)} = 1, \tag{2.11}$$

as $\mathsf{W}^{(k)}$ is only proportional to $\frac{f_{\mathbf{x}}(\mathbf{x}^{(k)})}{g(\mathbf{x}^{(k)})}$ and, thus, $f_{\mathbf{x}}(\mathbf{x}^{(k)})$ needs to be known only up to a constant factor. We will call $\mathsf{W}^{(k)}$ the normalized weights. We have

$$\mathsf{W}^{(k)} = \frac{w(\mathbf{x}^{(k)})}{\sum_{k'=1}^{K} w(\mathbf{x}^{(k')})}. \tag{2.12}$$

By restating (2.9) as

$$\begin{aligned} \hat{\mathsf{I}}^{(K)} &= \sum_{k=1}^{K} \frac{w(\mathbf{x}^{(k)})}{\sum_{k'=1}^{K} w(\mathbf{x}^{(k')})} h(\mathbf{x}^{(k)}) \\ &= \frac{1}{\sum_{k'=1}^{K} w(\mathbf{x}^{(k')})} \sum_{k=1}^{K} w(\mathbf{x}^{(k)}) h(\mathbf{x}^{(k)}) \\ &= \frac{1}{\frac{1}{K} \sum_{k'=1}^{K} w(\mathbf{x}^{(k')})} \frac{1}{K} \sum_{k=1}^{K} w(\mathbf{x}^{(k)}) h(\mathbf{x}^{(k)}), \end{aligned} \tag{2.13}$$

where we have used (2.12) in the first step, it follows that $\hat{\mathsf{I}}^{(K)} \xrightarrow{\text{a.s.}} I$ under mild conditions because $\frac{1}{K}\sum_{k=1}^{K} w(\mathbf{x}^{(k)}) \xrightarrow{\text{a.s.}} 1$ by the strong law of large numbers (see (2.7) with $h(\boldsymbol{x}) = 1$). This approximation is equivalent to first "approximating" the target pdf by

$$\hat{\mathsf{f}}_{\mathbf{x}}^{(K)}(\boldsymbol{x}) = \sum_{k=1}^{K} \mathsf{W}^{(k)} \delta_{\mathbf{x}^{(k)}}(\boldsymbol{x}) \tag{2.14}$$

and then calculating the expectation (2.3) with respect to this "approximation" of the target pdf.

2.3 Sequential Monte Carlo Methods

2.3.1 General Principle

Many applications require the processing of sequential data that can be modeled by a sequence of random vectors $(\mathbf{x}_n)_{n\in\mathbb{N}}$ with $\mathbf{x}_n \in \mathbb{R}^X$. The target pdf, for each $n \in \mathbb{N}$, is then $f_{\mathbf{x}_{1:n}}(\boldsymbol{x}_{1:n})$, where $\mathbf{x}_{1:n} = (\mathbf{x}_{n'})_{n'\in\{1,\dots,n\}}$. We will refer to n as time, even though it does not neccessarily have a temporal interpretation. Similarly to the previous section, we are interested in generating sequences of random vectors $\mathbf{x}_{1:n}^{(n,k)} = \left(\mathbf{x}_{n'}^{(n,k)}\right)_{n'\in\{1,\dots,n\}}$ with $k \in \{1,\dots,K\}$ for each $n \in \mathbb{N}$ to "approximate" the target pdf $f_{\mathbf{x}_{1:n}}(\boldsymbol{x}_{1:n})$ by

$$\hat{\mathsf{f}}_{\mathbf{x}_{1:n}}^{(K)}(\boldsymbol{x}_{1:n}) = \sum_{k=1}^{K} \mathsf{W}^{(n,k)} \delta_{\mathbf{x}_{1:n}^{(n,k)}}(\boldsymbol{x}_{1:n}), \tag{2.15}$$

and to approximate an expectation

$$I_n = \mathbb{E}(h_n(\mathbf{x}_{1:n})) = \int_{\mathbb{R}^{X\times n}} h_n(\boldsymbol{x}_{1:n}) f_{\mathbf{x}_{1:n}}(\boldsymbol{x}_{1:n}) \mathrm{d}\boldsymbol{x}_{1:n} \tag{2.16}$$

by

$$\hat{\mathsf{I}}_n^{(K)} = \int_{\mathbb{R}^{X\times n}} h_n(\boldsymbol{x}_{1:n}) \hat{\mathsf{f}}_{\mathbf{x}_{1:n}}^{(K)}(\boldsymbol{x}_{1:n}) \mathrm{d}\boldsymbol{x}_{1:n} = \sum_{k=1}^{K} \mathsf{W}^{(n,k)} h_n\left(\mathbf{x}_{1:n}^{(n,k)}\right), \tag{2.17}$$

where $h_n(\boldsymbol{x}_{1:n})$ is an arbitrary function, $K \in \mathbb{N}$, and $\sum_{k=1}^{K} \mathsf{W}^{(n,k)} = 1$. Note that each sequence $\mathbf{x}_{1:n}^{(n,k)}$ has a superscript n to avoid confusion of the random vectors of the approximations at different times. Typically SMC methods "approximate" the target pdf based on an "approximation" of the target pdf of the last time step. An "approximation" of the marginalized target pdf $f_{\mathbf{x}_n}(\boldsymbol{x}_n)$ can be obtained by marginalizing $\hat{\mathsf{f}}_{\mathbf{x}_{1:n}}^{(K)}(\boldsymbol{x}_{1:n})$, which yields

$$\hat{\mathsf{f}}_{\mathbf{x}_n}^{(K)}(\boldsymbol{x}_n) = \sum_{k=1}^{K} \mathsf{W}^{(n,k)} \delta_{\mathbf{x}_n^{(n,k)}}(\boldsymbol{x}_n), \tag{2.18}$$

where $\mathsf{x}_n^{(n,k)}$ denotes the nth component of $\mathsf{x}_{1:n}^{(n,k)}$. We will refer to the random sequences of random vectors $\mathsf{x}_{1:n}^{(n,k)}$ as particles.

While using IS to generate the particles $\mathsf{x}_{1:n}^{(n,k)}$ as discussed in the previous section would work in principle, it will typically be difficult to choose an appropriate and tractable proposal pdf with growing n. Further, this choice is especially difficult if the random vectors $\mathsf{x}_{n'}$ for $n' \in \{1, \ldots, n\}$ are highly dependent.

2.3.2 Particle Filter

One algorithm to generate the random vectors $\mathsf{x}_{1:n}^{(n,k)}$ is the particle filter (PF), which is a recursive algorithm that applies IS to "approximate" $f_{\mathsf{x}_{1:n}}(\boldsymbol{x}_{1:n})$ based on an "approximation" of $f_{\mathsf{x}_{1:n-1}}(\boldsymbol{x}_{1:n-1})$ for each $n \geq 2$. To illustrate the general principle of this algorithm, first note that we can factor $f_{\mathsf{x}_{1:n}}(\boldsymbol{x}_{1:n})$ as

$$f_{\mathsf{x}_{1:n}}(\boldsymbol{x}_{1:n}) = f_{\mathsf{x}_n|\mathsf{x}_{1:n-1}}(\boldsymbol{x}_n \,|\, \boldsymbol{x}_{1:n-1}) f_{\mathsf{x}_{1:n-1}}(\boldsymbol{x}_{1:n-1}). \tag{2.19}$$

The PF "approximates" (2.19) by first "approximating" $f_{\mathsf{x}_{1:n-1}}(\boldsymbol{x}_{1:n-1})$ with $\hat{\mathsf{f}}_{\mathsf{x}_{1:n-1}}^{(K)}(\boldsymbol{x}_{1:n-1})$ and then $f_{\mathsf{x}_n|\mathsf{x}_{1:n-1}}(\boldsymbol{x}_n \,|\, \boldsymbol{x}_{1:n-1})$ based on IS.

More specifically, for $n = 1$, we generate i.i.d. particles $\mathsf{x}_1^{(1,k)}$ with proposal pdf $g_1\left(\boldsymbol{x}_1^{(1,k)}\right)$ and set (see (2.10))

$$\mathsf{W}^{(1,k)} \propto \frac{f_{\mathsf{x}_1}\left(\mathsf{x}_1^{(1,k)}\right)}{g_1\left(\mathsf{x}_1^{(1,k)}\right)} \tag{2.20}$$

with $\sum_{k=1}^K \mathsf{W}^{(1,k)} = 1$. For $n \geq 2$, we first generate i.i.d. random vectors $\mathsf{x}_{1:n-1}^{(n,k)}$ with pdf $\hat{\mathsf{f}}_{\mathsf{x}_{1:n-1}}^{(K)}\left(\boldsymbol{x}_{1:n-1}^{(n,k)}\right)$, where $\hat{\mathsf{f}}_{\mathsf{x}_{1:n-1}}^{(K)}\left(\boldsymbol{x}_{1:n-1}^{(n,k)}\right)$ is defined as in (2.15), that is, we select

$$\mathsf{x}_{1:n-1}^{(n,k)} = \mathsf{x}_{1:n-1}^{(n-1,k')} \tag{2.21}$$

with probability $\mathsf{W}^{(n-1,k')}$, $k' \in \{1, \ldots, K\}$. Second, we generate $\mathsf{x}_n^{(n,k)}$ given $\mathsf{x}_{1:n-1}^{(n,k)} = \boldsymbol{x}_{1:n-1}^{(n,k)}$ with proposal pdf $g_n\left(\boldsymbol{x}_n^{(n,k)} \,\middle|\, \boldsymbol{x}_{1:n-1}^{(n,k)}\right)$. The complete particle is then

$$\mathsf{x}_{1:n}^{(n,k)} = \left(\mathsf{x}_{1:n-1}^{(n,k)}, \mathsf{x}_n^{(n,k)}\right) \tag{2.22}$$

with normalized weight

$$\mathsf{W}^{(n,k)} \propto \frac{f_{\mathsf{x}_n|\mathsf{x}_{1:n-1}}\left(\mathsf{x}_n^{(n,k)} \,\middle|\, \mathsf{x}_{1:n-1}^{(n,k)}\right)}{g_n\left(\mathsf{x}_n^{(n,k)} \,\middle|\, \mathsf{x}_{1:n-1}^{(n,k)}\right)} \tag{2.23}$$

such that $\sum_{k=1}^K \mathsf{W}^{(n,k)} = 1$.

Algorithm 2.1 Particle filter

Input: $\left(\boldsymbol{x}_{1:n-1}^{(n-1,k)}, W^{(n-1,k)}\right)_{k\in\{1,\dots,K\}}, g_n$

1: **for all** $k = 1,\dots,K$ **do**
2: sample $\boldsymbol{x}_{1:n-1}^{(n,k)} = \boldsymbol{x}_{1:n-1}^{(n-1,k')}$ with probability $W^{(n-1,k')}$ for all $k' \in \{1,\dots,K\}$
3: sample $\boldsymbol{x}_n^{(n,k)}$ from $g_n\left(\boldsymbol{x}_n^{(n,k)} \,\middle|\, \boldsymbol{x}_{1:n-1}^{(n,k)}\right)$
4: set $\boldsymbol{x}_{1:n}^{(n,k)} = \left(\boldsymbol{x}_{1:n-1}^{(n,k)}, \boldsymbol{x}_n^{(n,k)}\right)$
5: **end for**
6: **for all** $k = 1,\dots,K$ **do**
7: calculate $W^{(n,k)}$ according to (2.23)
8: **end for**

Output: $\left(\boldsymbol{x}_{1:n}^{(n,k)}, W^{(n,k)}\right)_{k\in\{1,\dots,K\}}$

One sufficient condition for the choice of the proposal pdfs to ensure $\hat{\mathsf{I}}_n^{(K)} \xrightarrow{\text{a.s.}} I_n$ for all $n \in \mathbb{N}$ is that $\operatorname{supp} g_1 \supset \operatorname{supp} f_{\mathbf{x}_1}$, and $\operatorname{supp} g_n(\cdot \,|\, \boldsymbol{x}_{1:n-1}) \supset \operatorname{supp} f_{\mathbf{x}_n|\mathbf{x}_{1:n-1}}(\cdot \,|\, \boldsymbol{x}_{1:n-1})$ for all $\boldsymbol{x}_{1:n-1} \in \mathbb{R}^{X\times(n-1)}$ and $n \geq 2$. In contrast to IS, we cannot directly apply the law of large numbers, as the $\mathbf{x}_{1:n-1}^{(n,k)}$ are generated with pdf $\hat{\mathsf{f}}_{\mathbf{x}_{1:n-1}}^{(K)}\left(\boldsymbol{x}_{1:n-1}^{(n,k)}\right)$ instead of $f_{\mathbf{x}_{1:n-1}}\left(\boldsymbol{x}_{1:n-1}^{(n,k)}\right)$. However, it can still be shown that this algorithm converges under mild conditions [32].

The pseudo-code for one recursion of the PF algorithm is stated in Algorithm 2.1.

2.3.3 State-Space Models

Consider two sequences of random vectors $(\mathbf{x}_n)_{n\in\mathbb{N}}$ with $\mathbf{x}_n \in \mathbb{R}^X$ and $(\mathbf{y}_n)_{n\in\mathbb{N}}$ with $\mathbf{y}_n \in \mathbb{R}^Y$, with pdf $f_{\mathbf{x}_1}(\boldsymbol{x}_1)$ and conditional pdfs $f_{\mathbf{y}_1|\mathbf{x}_1}(\boldsymbol{y}_1 \,|\, \boldsymbol{x}_1)$, $f_{\mathbf{x}_n|\mathbf{x}_{1:n-1},\mathbf{y}_{1:n-1}}(\boldsymbol{x}_n \,|\, \boldsymbol{x}_{1:n-1}, \boldsymbol{y}_{1:n-1})$, and $f_{\mathbf{y}_n|\mathbf{x}_{1:n},\mathbf{y}_{1:n-1}}(\boldsymbol{y}_n \,|\, \boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n-1})$ for $n \geq 2$. We will speak of a state-space model if $\mathbf{x}_n$ is conditionally independent of $\mathbf{x}_{1:n-2}$ and $\mathbf{y}_{1:n-1}$ given $\mathbf{x}_{n-1}$, and $\mathbf{y}_n$ is conditionally independent of $\mathbf{x}_{1:n-1}$ and $\mathbf{y}_{1:n-1}$ given $\mathbf{x}_n$, that is,

$$f_{\mathbf{x}_n|\mathbf{x}_{1:n-1},\mathbf{y}_{1:n-1}}(\boldsymbol{x}_n \,|\, \boldsymbol{x}_{1:n-1}, \boldsymbol{y}_{1:n-1}) = f_{\mathbf{x}_n|\mathbf{x}_{n-1}}(\boldsymbol{x}_n \,|\, \boldsymbol{x}_{n-1}), \tag{2.24}$$

and

$$f_{\mathbf{y}_n|\mathbf{x}_{1:n},\mathbf{y}_{1:n-1}}(\boldsymbol{y}_n \,|\, \boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n-1}) = f_{\mathbf{y}_n|\mathbf{x}_n}(\boldsymbol{y}_n \,|\, \boldsymbol{x}_n) \tag{2.25}$$

for all $n \geq 2$. The sequences $(\mathbf{x}_n)_{n\in\mathbb{N}}$ and $(\mathbf{y}_n)_{n\in\mathbb{N}}$ will be referred to as the states and the observations of the model, respectively. It follows from (2.24) and (2.25) that a state-space model is defined by the initial pdf $f_{\mathbf{x}_1}(\boldsymbol{x}_1) = \zeta(\boldsymbol{x}_1)$, as well as the state transition model $f_{\mathbf{x}_n|\mathbf{x}_{n-1}}(\boldsymbol{x}_n \,|\, \boldsymbol{x}_{n-1}) = \xi(\boldsymbol{x}_n \,|\, \boldsymbol{x}_{n-1})$ for $n \geq 2$ and obervation model $f_{\mathbf{y}_n|\mathbf{x}_n}(\boldsymbol{y}_n \,|\, \boldsymbol{x}_n) = \chi(\boldsymbol{y}_n \,|\, \boldsymbol{x}_n)$ for $n \geq 1$.

Our goal is to generate sequences of random vectors $\mathbf{x}_{1:n}^{(n,k)}$ with $k \in \{1, \ldots, K\}$ for each $n \in \mathbb{N}$ to "approximate" the target pdf $f_{\mathbf{x}_{1:n}|\mathbf{y}_{1:n}}(\boldsymbol{x}_{1:n} \,|\, \boldsymbol{y}_{1:n})$ by

$$\hat{\mathsf{f}}_{\mathbf{x}_{1:n}|\mathbf{y}_{1:n}}^{(K)}(\boldsymbol{x}_{1:n} \,|\, \boldsymbol{y}_{1:n}) = \sum_{k=1}^{K} \mathsf{W}^{(n,k)} \delta_{\mathbf{x}_{1:n}^{(n,k)}}(\boldsymbol{x}_{1:n}), \tag{2.26}$$

and to approximate an expectation $I_n = \mathbb{E}(h_n(\mathbf{x}_{1:n}) \,|\, \mathbf{y}_{1:n} = \boldsymbol{y}_{1:n})$ by

$$\hat{\mathsf{I}}_n^{(K)} = \int_{\mathbb{R}^{X \times n}} h_n(\boldsymbol{x}_{1:n}) \hat{\mathsf{f}}_{\mathbf{x}_{1:n}|\mathbf{y}_{1:n}}^{(K)}(\boldsymbol{x}_{1:n} \,|\, \boldsymbol{y}_{1:n}) \mathrm{d}\boldsymbol{x}_{1:n} = \sum_{k=1}^{K} \mathsf{W}^{(n,k)} h_n\left(\mathbf{x}_{1:n}^{(n,k)}\right), \tag{2.27}$$

where $h_n(\mathbf{x}_{1:n})$ is an arbitrary function, $K \in \mathbb{N}$, and $\sum_{k=1}^{K} \mathsf{W}^{(n,k)} = 1$. Note that the difference from our previous equations (2.15) and (2.17) is that all pdfs contain the condition $\mathbf{y}_{1:n}$. Furthermore note that a closed-form calculation of the posterior distribution of the states given the observations $f_{\mathbf{x}_{1:n}|\mathbf{y}_{1:n}}(\boldsymbol{x}_{1:n} \,|\, \boldsymbol{y}_{1:n})$ and the integral I_n is possible only for some special cases, such as a linear Gaussian model with the Kalman filter and some functions $h_n(\boldsymbol{x}_{1:n})$.

If we apply the PF algorithm from Section 2.3.2 to a state-space model, then, for $n = 1$, our proposal pdf is $g_1(\boldsymbol{x}_1 \,|\, \boldsymbol{y}_1)$ and equation (2.20) can be expressed as

$$\begin{aligned} \mathsf{W}^{(1,k)} &\propto \frac{f_{\mathbf{x}_1|\mathbf{y}_1}\left(\mathbf{x}_1^{(1,k)} \,\middle|\, \boldsymbol{y}_1\right)}{g_1\left(\mathbf{x}_1^{(1,k)} \,\middle|\, \boldsymbol{y}_1\right)} \\ &\propto \frac{f_{\mathbf{y}_1|\mathbf{x}_1}\left(\boldsymbol{y}_1 \,\middle|\, \mathbf{x}_1^{(1,k)}\right) f_{\mathbf{x}_1}\left(\mathbf{x}_1^{(1,k)}\right)}{g_1\left(\mathbf{x}_1^{(1,k)} \,\middle|\, \boldsymbol{y}_1\right)} \\ &\propto \frac{\chi\left(\boldsymbol{y}_1 \,\middle|\, \mathbf{x}_1^{(1,k)}\right) \zeta\left(\mathbf{x}_1^{(1,k)}\right)}{g_1\left(\mathbf{x}_1^{(1,k)} \,\middle|\, \boldsymbol{y}_1\right)}. \end{aligned} \tag{2.28}$$

For $n \geq 2$, we first resample from $\hat{\mathsf{f}}_{\mathbf{x}_{1:n-1}|\mathbf{y}_{1:n-1}}^{(K)}(\boldsymbol{x}_{1:n-1} \,|\, \boldsymbol{y}_{1:n-1})$ as defined in (2.26). Then, for the importance sampling step we use the proposal pdf $g_n\left(\boldsymbol{x}_n^{(n,k)} \,\middle|\, \boldsymbol{x}_{1:n-1}^{(n,k)}, \boldsymbol{y}_{1:n}\right)$ and normalized weights (see (2.23))

$$\mathsf{W}^{(n,k)} \propto \frac{f_{\mathbf{x}_n|\mathbf{x}_{1:n-1},\mathbf{y}_{1:n}}\left(\mathbf{x}_n^{(n,k)} \,\middle|\, \mathbf{x}_{1:n-1}^{(n,k)}, \boldsymbol{y}_{1:n}\right)}{g_n\left(\mathbf{x}_n^{(n,k)} \,\middle|\, \mathbf{x}_{1:n-1}^{(n,k)}, \boldsymbol{y}_{1:n}\right)} \tag{2.29}$$

such that $\sum_{k=1}^{K} \mathsf{W}^{(n,k)} = 1$. We can further simplify (2.29) according to

$$\begin{aligned}
\mathsf{W}^{(n,k)} &\propto \frac{f_{\mathbf{x}_n|\mathbf{y}_n,\mathbf{x}_{1:n-1},\mathbf{y}_{1:n-1}}\left(\mathbf{x}_n^{(n,k)} \,\middle|\, \boldsymbol{y}_n, \mathbf{x}_{1:n-1}^{(n,k)}, \boldsymbol{y}_{1:n-1}\right)}{g_n\left(\mathbf{x}_n^{(n,k)} \,\middle|\, \mathbf{x}_{1:n-1}^{(n,k)}, \boldsymbol{y}_{1:n}\right)} \\
&\propto \frac{f_{\mathbf{y}_n|\mathbf{x}_{1:n},\mathbf{y}_{1:n-1}}\left(\boldsymbol{y}_n \,\middle|\, \mathbf{x}_{1:n}^{(n,k)}, \boldsymbol{y}_{1:n-1}\right) f_{\mathbf{x}_n|\mathbf{x}_{1:n-1},\mathbf{y}_{1:n-1}}\left(\mathbf{x}_n^{(n,k)} \,\middle|\, \mathbf{x}_{1:n-1}^{(n,k)}, \boldsymbol{y}_{1:n-1}\right)}{g_n\left(\mathbf{x}_n^{(n,k)} \,\middle|\, \mathbf{x}_{1:n-1}^{(n,k)}, \boldsymbol{y}_{1:n}\right)} \\
&= \frac{f_{\mathbf{y}_n|\mathbf{x}_n}\left(\boldsymbol{y}_n \,\middle|\, \mathbf{x}_n^{(n,k)}\right) f_{\mathbf{x}_n|\mathbf{x}_{n-1}}\left(\mathbf{x}_n^{(n,k)} \,\middle|\, \mathbf{x}_{n-1}^{(n,k)}\right)}{g_n\left(\mathbf{x}_n^{(n,k)} \,\middle|\, \mathbf{x}_{1:n-1}^{(n,k)}, \boldsymbol{y}_{1:n}\right)} \\
&= \frac{\chi\left(\boldsymbol{y}_n \,\middle|\, \mathbf{x}_n^{(n,k)}\right) \xi\left(\mathbf{x}_n^{(n,k)} \,\middle|\, \mathbf{x}_{n-1}^{(n,k)}\right)}{g_n\left(\mathbf{x}_n^{(n,k)} \,\middle|\, \mathbf{x}_{1:n-1}^{(n,k)}, \boldsymbol{y}_{1:n}\right)},
\end{aligned} \tag{2.30}$$

where we first applied Bayes' theorem and then (2.24) as well as (2.25). This shows that to "approximate" the marginal pdf $f_{\mathbf{x}_n|\mathbf{y}_{1:n}}(\boldsymbol{x}_n \,|\, \boldsymbol{y}_{1:n})$ (see (2.18)), it is sufficient to only keep the samples $\boldsymbol{x}_{n-1}^{(n-1,k)}$ and the current observation $\boldsymbol{y}_n$ instead of the whole sequences $\boldsymbol{x}_{1:n-1}^{(n-1,k)}$ and $\boldsymbol{y}_{1:n}$ if we choose the proposal distribution $g_n(\boldsymbol{x}_n \,|\, \boldsymbol{x}_{1:n-1}, \boldsymbol{y}_{1:n})$ independent of $\boldsymbol{x}_{1:n-2}$ and $\boldsymbol{y}_{1:n-1}$.

2.3.4 Resample-Move Particle Filter

Consider the PF from Section 2.3.2. The particles for time n are generated by resampling particles from time $n-1$ (see (2.21)) and extending them to time n (see (2.22)). The resampling step has the disadvantage that there may exist particles $\mathbf{x}_{1:n}^{(n,k)} = \left(\mathbf{x}_{1:n-1}^{(n,k)}, \mathbf{x}_n^{(n,k)}\right)$ and $\mathbf{x}_{1:n}^{(n,k')} = \left(\mathbf{x}_{1:n-1}^{(n,k')}, \mathbf{x}_n^{(n,k')}\right)$ with $\mathbf{x}_{1:n-1}^{(n,k)} = \mathbf{x}_{1:n-1}^{(n,k')}$. Therefore, this step depletes the particle values $\mathbf{x}_{n'}^{(n,k)}$ for time $n' \leq n-1$, as $\left|\left\{\mathbf{x}_1^{(n,k)} : k \in \{1,\ldots,K\}\right\}\right| \leq \left|\left\{\mathbf{x}_2^{(n,k)} : k \in \{1,\ldots,K\}\right\}\right| \leq \ldots \leq K$. This depletion can result in poor approximations (2.17) and (2.15) if the number of particles K is too small.

One approach to avoid this problem is to add an additional step to move the resampled particles from time $n-1$ by a transition kernel. If this transition kernel, defined by a conditional pdf $t_{n-1}(\boldsymbol{x}_{1:n-1} \,|\, \bar{\boldsymbol{x}}_{1:n-1})$, has $f_{\mathbf{x}_{1:n-1}}(\boldsymbol{x}_{1:n-1})$ as an invariant distribution, that is,

$$\int_{\mathbb{R}^{X\times(n-1)}} t_{n-1}(\boldsymbol{x}_{1:n-1} \,|\, \bar{\boldsymbol{x}}_{1:n-1}) f_{\mathbf{x}_{1:n-1}}(\bar{\boldsymbol{x}}_{1:n-1}) \mathrm{d}\bar{\boldsymbol{x}}_{1:n-1} = f_{\mathbf{x}_{1:n-1}}(\boldsymbol{x}_{1:n-1}), \tag{2.31}$$

then we obtain the resample-move particle filter (RMPF), first proposed in [24]. One recursion of the RMPF is presented in Algorithm 2.2.

However, while this algorithm can reduce the particle value depletion, the additional step to move the particles can be computationally expensive for large n, if the transition

Algorithm 2.2 Resample-move particle filter

Input: $\left(\boldsymbol{x}_{1:n-1}^{(n-1,k)}, W^{(n-1,k)}\right)_{k\in\{1,\dots,K\}}, t_{n-1}, g_n$

1: **for all** $k = 1, \dots, K$ **do**
2: sample $\bar{\boldsymbol{x}}_{1:n-1}^{(n,k)} = \boldsymbol{x}_{1:n-1}^{(n-1,k')}$ with probability $W^{(n-1,k')}$ for all $k' \in \{1, \dots, K\}$
3: sample $\boldsymbol{x}_{1:n-1}^{(n,k)}$ from $t_{n-1}\left(\boldsymbol{x}_{1:n-1}^{(n,k)} \,\middle|\, \bar{\boldsymbol{x}}_{1:n-1}^{(n,k)}\right)$
4: sample $\boldsymbol{x}_n^{(n,k)}$ from $g_n\left(\boldsymbol{x}_n^{(n,k)} \,\middle|\, \boldsymbol{x}_{1:n-1}^{(n,k)}\right)$
5: set $\boldsymbol{x}_{1:n}^{(n,k)} = \left(\boldsymbol{x}_{1:n-1}^{(n,k)}, \boldsymbol{x}_n^{(n,k)}\right)$
6: **end for**
7: **for all** $k = 1, \dots, K$ **do**
8: calculate $W^{(n,k)}$ according to (2.23)
9: **end for**

Output: $\left(\boldsymbol{x}_{1:n}^{(n,k)}, W^{(n,k)}\right)_{k\in\{1,\dots,K\}}$

kernels move the complete particles $\bar{\mathbf{x}}_{1:n}^{(n,k)}$. In order to reduce the computational complexity of this step, it can be sufficient to only move a part of each particle, for example with a transition kernel

$$t_{n-1}\left(\boldsymbol{x}_{1:n-1}^{(n,k)} \,\middle|\, \bar{\boldsymbol{x}}_{1:n-1}^{(n,k)}\right) = t_{n'+1:n-1}\left(\boldsymbol{x}_{n'+1:n-1}^{(n,k)} \,\middle|\, \bar{\boldsymbol{x}}_{n'+1:n-1}^{(n,k)}\right)\delta_{\bar{\boldsymbol{x}}_{1:n'}^{(n,k)}}\left(\boldsymbol{x}_{1:n'}^{(n,k)}\right), \tag{2.32}$$

where $n' < n-1$ and $t_{n'+1:n-1}\left(\boldsymbol{x}_{n'+1:n-1}^{(n,k)} \,\middle|\, \bar{\boldsymbol{x}}_{n'+1:n-1}^{(n,k)}\right)$ is a transition kernel with invariant distribution $f_{\mathbf{x}_{n'+1:n-1}}\left(\boldsymbol{x}_{n'+1:n-1}^{(n,k)}\right)$.

2.3.5 Parameter Dependent State-Space Models

While we can often describe a system by a state-space model, it can be necessary to extend it by a random but time-constant parameter that the states and observations depend on. Due to this dependency, it can be important to perform inference on this parameter jointly with the states. The inference on this parameter may even be the main objective of the task.

Similarly to Section 2.3.3, consider sequences of states $(\mathbf{x}_n)_{n\in\mathbb{N}}$, where $\mathbf{x}_n \in \mathbb{R}^X$, and observations $(\mathbf{y}_n)_{n\in\mathbb{N}}$, where $\mathbf{y}_n \in \mathbb{R}^Y$, as well as an additional random parameter $\mathbf{p} \in \mathbb{R}^D$, with $D \in \mathbb{N}$. We will speak of a parameter-dependent state-space model if (see (2.24))

$$f_{\mathbf{x}_n|\mathbf{x}_{1:n-1},\mathbf{y}_{1:n-1},\mathbf{p}}(\boldsymbol{x}_n \,|\, \boldsymbol{x}_{1:n-1}, \boldsymbol{y}_{1:n-1}, \boldsymbol{p}) = f_{\mathbf{x}_n|\mathbf{x}_{n-1},\mathbf{p}}(\boldsymbol{x}_n \,|\, \boldsymbol{x}_{n-1}, \boldsymbol{p}) \tag{2.33}$$

and (see (2.25))

$$f_{\mathbf{y}_n|\mathbf{x}_{1:n},\mathbf{y}_{1:n-1},\mathbf{p}}(\boldsymbol{y}_n \,|\, \boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n-1}, \boldsymbol{p}) = f_{\mathbf{y}_n|\mathbf{x}_n,\mathbf{p}}(\boldsymbol{y}_n \,|\, \boldsymbol{x}_n, \boldsymbol{p}) \tag{2.34}$$

for all $n \geq 2$. A parameter dependent state-space model is defined by its initial pdf $f_{\mathbf{x}_1|\mathbf{p}}(\boldsymbol{x}_1 \,|\, \boldsymbol{p}) = \zeta(\boldsymbol{x}_1 \,|\, \boldsymbol{p})$, the state transition model $f_{\mathbf{x}_n|\mathbf{x}_{n-1},\mathbf{p}}(\boldsymbol{x}_n \,|\, \boldsymbol{x}_{n-1}, \boldsymbol{p}) = \xi(\boldsymbol{x}_n \,|\, \boldsymbol{x}_{n-1}, \boldsymbol{p})$ for $n \geq 2$, the observation model $f_{\mathbf{y}_n|\mathbf{x}_n,\mathbf{p}}(\boldsymbol{y}_n \,|\, \boldsymbol{x}_n, \boldsymbol{p}) = \chi(\boldsymbol{y}_n \,|\, \boldsymbol{x}_n, \boldsymbol{p})$ for $n \geq 1$, and the pdf of the random parameter $f_{\mathbf{p}}(\boldsymbol{p})$. Here, our goal is to generate sequences of random vectors $\mathbf{x}_{1:n}^{(n,k)}$ and parameters $\mathbf{p}^{(n,k)}$ with $k \in \{1, \ldots, K\}$ for each $n \in \mathbb{N}$ to "approximate" the target pdf $f_{\mathbf{x}_{1:n},\mathbf{p}|\mathbf{y}_{1:n}}(\boldsymbol{x}_{1:n}, \boldsymbol{p} \,|\, \boldsymbol{y}_{1:n})$ by

$$\hat{\mathsf{f}}^{(K)}_{\mathbf{x}_{1:n},\mathbf{p}|\mathbf{y}_{1:n}}(\boldsymbol{x}_{1:n}, \boldsymbol{p} \,|\, \boldsymbol{y}_{1:n}) = \sum_{k=1}^{K} \mathsf{W}^{(n,k)} \delta_{(\mathbf{x}_{1:n}^{(n,k)}, \mathbf{p}^{(n,k)})}(\boldsymbol{x}_{1:n}, \boldsymbol{p}), \tag{2.35}$$

and to approximate an expectation $I_n = \mathbb{E}(h_n(\mathbf{x}_{1:n}, \mathbf{p}) \,|\, \mathbf{y}_{1:n} = \boldsymbol{y}_{1:n})$ by

$$\begin{aligned} \hat{\mathsf{I}}_n^{(K)} &= \int_{\mathbb{R}^D} \int_{\mathbb{R}^{X \times n}} h_n(\boldsymbol{x}_{1:n}, \boldsymbol{p}) \hat{\mathsf{f}}^{(K)}_{\mathbf{x}_{1:n},\mathbf{p}|\mathbf{y}_{1:n}}(\boldsymbol{x}_{1:n}, \boldsymbol{p} \,|\, \boldsymbol{y}_{1:n}) \mathrm{d}\boldsymbol{x}_{1:n} \mathrm{d}\boldsymbol{p} \\ &= \sum_{k=1}^{K} \mathsf{W}^{(n,k)} h_n\Big(\mathbf{x}_{1:n}^{(n,k)}, \mathbf{p}^{(n,k)}\Big), \end{aligned} \tag{2.36}$$

where $h_n(\boldsymbol{x}_{1:n}, \boldsymbol{p})$ is an arbitrary function, $K \in \mathbb{N}$, and $\sum_{k=1}^{K} \mathsf{W}^{(n,k)} = 1$.

If the PF is directly applied to an extended state $\check{\mathbf{x}}_n = (\mathbf{x}_n, \mathbf{p}_n)$ with time-independent parameter $\mathbf{p}_n = \mathbf{p}_{n-1} = \mathbf{p}$, then the algorithm will not move the parameter part of the particles $\check{\mathbf{x}}_{1:n}^{(n,k)} = \big(\mathbf{x}_{1:n}^{(n,k)}, \mathbf{p}_{1:n}^{(n,k)}\big)$ as time progresses. Therefore, the distribution of the parameter is at all times approximated by the initial samples or, due to the particle depletion discussed in Section 2.3.4, a subset of them.

However, it is possible to solve this problem by applying the RMPF algorithm described in Section 2.3.4 to the extended state with a transition kernel that only moves the parameter part [24]. This can be done by using a transition kernel defined by the conditional pdf

$$\begin{aligned} t_{n-1}\Big(\check{\boldsymbol{x}}_{1:n-1}^{(n,k)} \,\Big|\, \bar{\check{\boldsymbol{x}}}_{1:n-1}^{(n,k)}, \boldsymbol{y}_{1:n-1}\Big) &= t_{n-1}\Big(\big(\boldsymbol{x}_{1:n-1}^{(n,k)}, \boldsymbol{p}_{1:n-1}^{(n,k)}\big) \,\Big|\, \big(\bar{\boldsymbol{x}}_{1:n-1}^{(n,k)}, \bar{\boldsymbol{p}}_{1:n-1}^{(n,k)}\big), \boldsymbol{y}_{1:n-1}\Big) \\ &= \delta_{\bar{\boldsymbol{x}}_{1:n-1}^{(n,k)}}\big(\boldsymbol{x}_{1:n-1}^{(n,k)}\big) f_{\mathbf{p}|\mathbf{x}_{1:n-1},\mathbf{y}_{1:n-1}}\Big(\boldsymbol{p}_{n-1}^{(n,k)} \,\Big|\, \boldsymbol{x}_{1:n-1}^{(n,k)}, \boldsymbol{y}_{1:n-1}\Big). \end{aligned} \tag{2.37}$$

From this transition kernel, we sample $\boldsymbol{x}_{1:n-1}^{(n,k)}$ (note that $\boldsymbol{x}_{1:n-1}^{(n,k)} = \bar{\boldsymbol{x}}_{1:n-1}^{(n,k)}$) and $\boldsymbol{p}_{1:n-1}^{(n,k)}$. Furthermore, we set $\boldsymbol{p}_{n'}^{(n,k)} = \boldsymbol{p}_{n-1}^{(n,k)}$ for all $n' \leq n-1$. Note that this transition kernel is independent of $\bar{\boldsymbol{p}}_{1:n-1}^{(n,k)}$.

As the parameter $\mathbf{p}$ is constant over time, we can factor the proposal pdf according to

$$\begin{aligned} g_n\Big(\check{\boldsymbol{x}}_n^{(n,k)} \,\Big|\, \check{\boldsymbol{x}}_{1:n-1}^{(n,k)}, \boldsymbol{y}_{1:n-1}\Big) &= g_n\Big(\big(\boldsymbol{x}_n^{(n,k)}, \boldsymbol{p}_n^{(n,k)}\big) \,\Big|\, \big(\boldsymbol{x}_{1:n-1}^{(n,k)}, \boldsymbol{p}_{1:n-1}^{(n,k)}\big), \boldsymbol{y}_{1:n-1}\Big) \\ &= \delta_{\boldsymbol{p}_{n-1}^{(n,k)}}\big(\boldsymbol{p}_n^{(n,k)}\big) \tilde{g}_n\Big(\boldsymbol{x}_n^{(n,k)} \,\Big|\, \boldsymbol{x}_{1:n-1}^{(n,k)}, \boldsymbol{p}_{n-1}^{(n,k)}, \boldsymbol{y}_{1:n}\Big), \end{aligned} \tag{2.38}$$

where $\tilde{g}_n\left(\boldsymbol{x}_n^{(n,k)} \,\middle|\, \boldsymbol{x}_{1:n-1}^{(n,k)}, \boldsymbol{p}_{n-1}^{(n,k)}, \boldsymbol{y}_{1:n}\right)$ is a proposal pdf for the state vector for all $n \geq 2$. That is, the proposal pdf is only used to sample the state $\boldsymbol{x}_n^{(n,k)}$ and leaves the parameter $\boldsymbol{p}_n^{(n,k)}$, which was moved by the transition kernel $t_{n-1}\left(\check{\boldsymbol{x}}_{1:n-1}^{(n,k)} \,\middle|\, \bar{\check{\boldsymbol{x}}}_{1:n-1}^{(n,k)}, \boldsymbol{y}_{1:n-1}\right)$, unchanged, that is, $\boldsymbol{p}_n^{(n,k)} - \boldsymbol{p}_{n-1}^{(n,k)}$. With this choice of the proposal pdf $g_n\left(\check{\boldsymbol{x}}_n^{(n,k)} \,\middle|\, \check{\boldsymbol{x}}_{1:n-1}^{(n,k)}, \boldsymbol{y}_{1:n-1}\right)$, we can express the normalized weights of equation (2.20) as (see (2.28))

$$\begin{aligned} \mathsf{W}^{(1,k)} &\propto \frac{f_{\mathbf{y}_1|\mathbf{x}_1,\mathbf{p}}\left(\boldsymbol{y}_1 \,\middle|\, \mathbf{x}_1^{(1,k)}, \mathbf{p}_1^{(1,k)}\right) f_{\mathbf{x}_1,\mathbf{p}}\left(\mathbf{x}_1^{(1,k)}, \mathbf{p}_1^{(1,k)}\right)}{g_1\left(\left(\mathbf{x}_1^{(1,k)}, \mathbf{p}_1^{(1,k)}\right) \,\middle|\, \boldsymbol{y}_1\right)} \\ &= \frac{\chi\left(\boldsymbol{y}_1 \,\middle|\, \mathbf{x}_1^{(1,k)}, \mathbf{p}_1^{(1,k)}\right) \zeta\left(\mathbf{x}_1^{(1,k)} \,\middle|\, \mathbf{p}_1^{(1,k)}\right) f_{\mathbf{p}}(\mathbf{p}_1^{(1,k)})}{g_1\left(\left(\mathbf{x}_1^{(1,k)}, \mathbf{p}_1^{(1,k)}\right) \,\middle|\, \boldsymbol{y}_1\right)} \end{aligned} \tag{2.39}$$

and of equation (2.23) as (see (2.30))

$$\begin{aligned} \mathsf{W}^{(n,k)} &\propto \frac{f_{\mathbf{y}_n|\mathbf{x}_n,\mathbf{p}}\left(\boldsymbol{y}_n \,\middle|\, \mathbf{x}_n^{(n,k)}, \mathbf{p}_n^{(n,k)}\right) f_{\mathbf{x}_n|\mathbf{x}_{n-1},\mathbf{p}}\left(\mathbf{x}_n^{(n,k)} \,\middle|\, \mathbf{x}_{n-1}^{(n,k)}, \mathbf{p}_{n-1}^{(n,k)}\right)}{\tilde{g}_n\left(\mathbf{x}_n^{(n,k)} \,\middle|\, \mathbf{x}_{1:n-1}^{(n,k)}, \mathbf{p}_{n-1}^{(n,k)}, \boldsymbol{y}_{1:n}\right)} \\ &= \frac{\chi\left(\boldsymbol{y}_n \,\middle|\, \mathbf{x}_n^{(n,k)}, \mathbf{p}_n^{(n,k)}\right) \xi\left(\mathbf{x}_n^{(n,k)} \,\middle|\, \mathbf{x}_{n-1}^{(n,k)}, \mathbf{p}_{n-1}^{(n,k)}\right)}{\tilde{g}_n\left(\mathbf{x}_n^{(n,k)} \,\middle|\, \mathbf{x}_{1:n-1}^{(n,k)}, \mathbf{p}_{n-1}^{(n,k)}, \boldsymbol{y}_{1:n}\right)} \end{aligned} \tag{2.40}$$

for $n \geq 2$.

In general, we need the complete sequences $\boldsymbol{x}_{1:n-1}^{(n,k)}$ and $\boldsymbol{y}_{1:n-1}$ to sample from the transition kernel defined in (2.37). Let us assume that there exists a sufficient statistic $\mathbf{s}_n \in \mathbb{R}^E$, with $E \in \mathbb{N}$, such that $f_{\mathbf{p}|\mathbf{x}_{1:n},\mathbf{y}_{1:n}}(\boldsymbol{p} \,|\, \boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n}) = f_{\mathbf{p}|\mathbf{s}_n}(\boldsymbol{p} \,|\, \boldsymbol{s}_n)$. Further, assume that $\mathbf{s}_n$ can be calculated recursively according to $\mathbf{s}_1 = u_1(\mathbf{x}_1, \mathbf{y}_1)$ with some function $u_1(\boldsymbol{x}_1, \boldsymbol{y}_1)$ and $\mathbf{s}_n = u(\mathbf{s}_{n-1}, \mathbf{x}_n, \mathbf{y}_n)$ for all $n \geq 2$ with some function $u(\boldsymbol{s}_{n-1}, \boldsymbol{x}_n, \boldsymbol{y}_n)$. These assumptions allow us to simplify the RMPF [25], as the conditional pdf defining the transition kernel (2.37) can be expressed as

$$t_{n-1}\left(\check{\boldsymbol{x}}_{1:n-1}^{(n,k)} \,\middle|\, \bar{\check{\boldsymbol{x}}}_{1:n-1}^{(n,k)}, \boldsymbol{y}_{1:n-1}\right) = \delta_{\bar{\boldsymbol{x}}_{1:n-1}^{(n,k)}}\left(\boldsymbol{x}_{1:n-1}^{(n,k)}\right) f_{\mathbf{p}|\mathbf{s}_{n-1}}\left(\boldsymbol{p}_{n-1}^{(n,k)} \,\middle|\, \bar{\boldsymbol{s}}_{n-1}^{(n,k)}\right), \tag{2.41}$$

where $\bar{\boldsymbol{s}}_{n-1}^{(n,k)}$ is the sufficient statistic corresponding to the particle $\bar{\boldsymbol{x}}_{1:n-1}^{(n,k)}$, that is, $\bar{\boldsymbol{s}}_1^{(n,k)} = u_1\left(\bar{\boldsymbol{x}}_1^{(n,k)}, \boldsymbol{y}_1\right)$ and $\bar{\boldsymbol{s}}_{n'}^{(n,k)} = u\left(\bar{\boldsymbol{s}}_{n'-1}^{(n,k)}, \bar{\boldsymbol{x}}_{n'}^{(n,k)}, \boldsymbol{y}_{n'}\right)$ for $n' \geq 2$. This shows that, similarly to the PF, to "approximate" the marginal pdfs $f_{\mathbf{x}_n,\mathbf{p}|\mathbf{y}_{1:n}}(\boldsymbol{x}_n, \boldsymbol{p} \,|\, \boldsymbol{y}_{1:n})$, it is sufficient to only keep the samples $\left(\boldsymbol{x}_{n-1}^{(n-1,k)}, \boldsymbol{p}_{n-1}^{(n-1,k)}\right)$ and the sufficient statistic $\boldsymbol{s}_{n-1}^{(n-1,k)}$ instead of the whole sequences $\boldsymbol{x}_{1:n-1}^{(n-1,k)}$ and $\boldsymbol{y}_{1:n-1}$, if we choose the proposal distribution g_n independent of $\boldsymbol{x}_{1:n-2}$ and $\boldsymbol{y}_{1:n-1}$. If there does not exist a sufficient statistic that can be calculated recursively, then we can approximate the posterior distribution of the parameter with a distribution where such a sufficient statistic exists [26].

2.4 Markov Chain Monte Carlo

2.4.1 General Principle

Another method to generate random vectors $\mathbf{x}^{(k)} \in \mathbb{R}^X$ with $k \in \{1, \dots, K\}$ to approximate the expectation (2.3) as in (2.4), is to sample a homogeneous discrete-time Markov chain which "explores" the target pdf.

A homogeneous discrete-time Markov chain is a sequence of random vectors $(\mathbf{x}^{(k)})_{k\in\mathbb{N}}$ with $\mathbf{x}^{(k)} \in \mathbb{R}^X$, that is constructed from a transition kernel defined by the conditional pdf $t(\boldsymbol{x}^{(k)} \,|\, \boldsymbol{x}^{(k-1)})$, such that $f_{\mathbf{x}^{(k)}|\mathbf{x}^{(k-1)},\dots,\mathbf{x}^{(1)}}(\boldsymbol{x}^{(k)} \,|\, \boldsymbol{x}^{(k-1)}, \dots, \boldsymbol{x}^{(1)}) = f_{\mathbf{x}^{(k)}|\mathbf{x}^{(k-1)}}(\boldsymbol{x}^{(k)} \,|\, \boldsymbol{x}^{(k-1)}) = t(\boldsymbol{x}^{(k)} \,|\, \boldsymbol{x}^{(k-1)})$ for all $k \geq 2$. The ergodic theorem ensures $\hat{\mathrm{I}}^{(K)} \xrightarrow{\text{a.s.}} I$ under mild conditions, if the Markov chain is ergodic with the target pdf as its stationary pdf, that is, if $f_{\mathbf{x}^{(k)}}(\boldsymbol{x}) = f_{\mathbf{x}}(\boldsymbol{x})$, then $f_{\mathbf{x}^{(k+1)}}(\boldsymbol{x}) = f_{\mathbf{x}}(\boldsymbol{x})$. This is the case if the Markov chain is irreducible and aperiodic [30, Section 6], and the target pdf is an invariant pdf of the transition kernel, that is, $\int_{\mathbb{R}^X} t(\boldsymbol{x}^{(k)} \,|\, \boldsymbol{x}^{(k-1)}) f_{\mathbf{x}}(\boldsymbol{x}^{(k-1)}) \mathrm{d}\boldsymbol{x}^{(k-1)} = f_{\mathbf{x}}(\boldsymbol{x}^{(k)})$. These properties ensure that the Markov chain "explores" the support of the target pdf without getting stuck in cycles, and that the pdf of the samples converges to the target pdf. We can sample such a Markov chain by first initializing $\boldsymbol{x}^{(1)} \in \operatorname{supp} f_{\mathbf{x}}$ arbitrarily, and then sampling $\boldsymbol{x}^{(k)}$ from $t(\boldsymbol{x}^{(k)} \,|\, \boldsymbol{x}^{(k-1)})$ at each step $k \geq 2$.

There are several generic methods to construct suitable transition kernels for a wide range of applications, some of which will be discussed below.

2.4.2 Metropolis-Hastings Algorithm

One of the most popular MCMC methods is the Metropolis-Hastings (MH) algorithm. This algorithm is a "standard approach," insofar as many other MCMC methods are closely related to it.

At each step $k \geq 2$, given the previous sample $\mathbf{x}^{(k-1)} = \boldsymbol{x}^{(k-1)}$, a candidate $\tilde{\boldsymbol{x}}^{(k)}$ is sampled from a conditional proposal pdf $g(\tilde{\boldsymbol{x}}^{(k)} | \boldsymbol{x}^{(k-1)})$. This candidate is then accepted

Algorithm 2.3 MH algorithm

Input: $\boldsymbol{x}^{(k-1)}, g$

1: sample $\tilde{\boldsymbol{x}}^{(k)}$ from $g(\tilde{\boldsymbol{x}}^{(k)} \,|\, \boldsymbol{x}^{(k-1)})$

2: set $\boldsymbol{x}^{(k)} = \tilde{\boldsymbol{x}}^{(k)}$ with probability $P_{\mathrm{a}}^{(k)}$ (see (2.42))
else set $\boldsymbol{x}^{(k)} = \boldsymbol{x}^{(k-1)}$

Output: $\boldsymbol{x}^{(k)}$

$\mathbf{x}^{(k)} = \tilde{\boldsymbol{x}}^{(k)}$ with probability

$$\begin{aligned} P_{\mathrm{a}}^{(k)} &= P_{\mathrm{a}}(\tilde{\boldsymbol{x}}^{(k)}, \boldsymbol{x}^{(k-1)}) \\ &= \min\left\{1, \frac{f_{\mathbf{x}}(\tilde{\boldsymbol{x}}^{(k)}) g(\boldsymbol{x}^{(k-1)} \,|\, \tilde{\boldsymbol{x}}^{(k)})}{f_{\mathbf{x}}(\boldsymbol{x}^{(k-1)}) g(\tilde{\boldsymbol{x}}^{(k)} \,|\, \boldsymbol{x}^{(k-1)})}\right\} \\ &= \begin{cases} 1, & \text{if } \frac{f_{\mathbf{x}}(\tilde{\boldsymbol{x}}^{(k)}) g(\boldsymbol{x}^{(k-1)} \,|\, \tilde{\boldsymbol{x}}^{(k)})}{f_{\mathbf{x}}(\boldsymbol{x}^{(k-1)}) g(\tilde{\boldsymbol{x}}^{(k)} \,|\, \boldsymbol{x}^{(k-1)})} > 1 \\ \frac{f_{\mathbf{x}}(\tilde{\boldsymbol{x}}^{(k)}) g(\boldsymbol{x}^{(k-1)} \,|\, \tilde{\boldsymbol{x}}^{(k)})}{f_{\mathbf{x}}(\boldsymbol{x}^{(k-1)}) g(\tilde{\boldsymbol{x}}^{(k)} \,|\, \boldsymbol{x}^{(k-1)})}, & \text{else} \end{cases}. \end{aligned} \tag{2.42}$$

If it is not accepted, then the previous sample is used again, that is, $\mathbf{x}^{(k)} = \boldsymbol{x}^{(k-1)}$. To find an expression for the resulting transition kernel $t(\boldsymbol{x}^{(k)} \,|\, \boldsymbol{x}^{(k-1)})$, note that the sample $\boldsymbol{x}^{(k)}$ is either the candidate generated with pdf $g(\boldsymbol{x}^{(k)} \,|\, \boldsymbol{x}^{(k-1)})$ and accepted with probability $P_{\mathrm{a}}(\boldsymbol{x}^{(k)}, \boldsymbol{x}^{(k-1)})$, or the last sample $\boldsymbol{x}^{(k-1)}$ with any candidate $\boldsymbol{x}$, that was generated by $g(\boldsymbol{x} \,|\, \boldsymbol{x}^{(k-1)})$ and then rejected with probability $1 - P_{\mathrm{a}}(\boldsymbol{x}, \boldsymbol{x}^{(k-1)})$. Therefore, the transition kernel is defined by the conditional pdf

$$t(\boldsymbol{x}^{(k)} \,|\, \boldsymbol{x}^{(k-1)}) = P_{\mathrm{a}}(\boldsymbol{x}^{(k)}, \boldsymbol{x}^{(k-1)}) g(\boldsymbol{x}^{(k)} \,|\, \boldsymbol{x}^{(k-1)}) + r(\boldsymbol{x}^{(k-1)}) \delta_{\boldsymbol{x}^{(k-1)}}(\boldsymbol{x}^{(k)}), \tag{2.43}$$

where

$$r(\boldsymbol{x}^{(k-1)}) = \int_{\mathbb{R}^X} (1 - P_{\mathrm{a}}(\boldsymbol{x}, \boldsymbol{x}^{(k-1)})) g(\boldsymbol{x} \,|\, \boldsymbol{x}^{(k-1)}) \mathrm{d}\boldsymbol{x}. \tag{2.44}$$

The pseudo-code for one step of this procedure is presented in Algorithm 2.3.

As with IS, equation (2.42) shows that in order to use the MH algorithm, we only have to be able to evaluate the target pdf $f_{\mathbf{x}}(\boldsymbol{x})$ up to a normalizing factor as it appears in both the numerator and the denominator. A sufficient condition for the convergence of the algorithm is that the conditional proposal pdf $g(\tilde{\boldsymbol{x}}^{(k)} \,|\, \boldsymbol{x}^{(k-1)})$ satisfies $\operatorname{supp} g(\cdot \,|\, \boldsymbol{x}) \supset \operatorname{supp} f_{\mathbf{x}}$ for all $\boldsymbol{x} \in \mathbb{R}^X$. In order to ensure fast convergence, it is recommended to choose the proposal pdf $g(\tilde{\boldsymbol{x}}^{(k)} \,|\, \boldsymbol{x}^{(k-1)})$ such that the acceptance probability $P_{\mathrm{a}}(\tilde{\boldsymbol{x}}^{(k)}, \boldsymbol{x}^{(k-1)})$ is close to one for most $\tilde{\boldsymbol{x}}^{(k)}, \boldsymbol{x}^{(k-1)} \in \mathbb{R}^X$ to prevent the Markov chain from getting stuck for a longer time.

Algorithm 2.4 MCMC algorithm with kernel cycles

Input: $\boldsymbol{x}^{(k-1)}, (t_m)_{m\in\{1,\dots,N_\mathrm{B}\}}$

1: **for all** $m = 1, \dots, N_\mathrm{B}$ **do**

2: sample $\boldsymbol{x}_m^{(k)}$ from $t_m\left(\boldsymbol{x}_m^{(k)} \,\middle|\, \boldsymbol{x}_m^{(k-1)}, \boldsymbol{x}_{\neg m}^{(k,k-1)}\right)$

3: **end for**

Output: $\boldsymbol{x}^{(k)}$

2.4.3 Cycles of MCMC Kernels

One problem of the MH algorithm is that for large dimension X, it can be difficult to find a conditional proposal pdf $g(\tilde{\boldsymbol{x}}^{(k)} \,|\, \boldsymbol{x}^{(k-1)})$ with a high acceptance probability $P_\mathrm{a}(\tilde{\boldsymbol{x}}^{(k)}, \boldsymbol{x}^{(k-1)})$. One way to circumvent this problem is to split the random vector $\mathsf{x} = [\mathsf{x}_1 \ \cdots \ \mathsf{x}_X]^\mathrm{T}$ into an arbitrary number $N_\mathrm{B} \leq X$ of blocks $\mathsf{x}_1, \dots, \mathsf{x}_{N_\mathrm{B}}$ and then use a transition kernel defined by the conditional pdf $t(\boldsymbol{x}^{(k)} \,|\, \boldsymbol{x}^{(k-1)})$ constructed from N_B lower dimensional transition kernels, one for each block. We can split the random vector x into N_B blocks x_m by first partitioning the set $\{1, \dots, X\}$ as $\{1, \dots, X\} = \bigcup_{m=1}^{N_\mathrm{B}} B_m$, where $B_m = \{b_{m,1}, \dots, b_{m,|B_m|}\} \subseteq \{1, \dots, X\}$ are non-empty disjoint sets for all $m \in \{1, \dots, N_\mathrm{B}\}$, and then setting $\mathsf{x}_m = [\mathsf{x}_{b_{m,1}} \ \cdots \ \mathsf{x}_{b_{m,|B_m|}}]^\mathrm{T} \in \mathbb{R}^{|B_m|}$ for all $m \in \{1, \dots, N_\mathrm{B}\}$. If, for all $m \in \{1, \dots, N_\mathrm{B}\}$, we choose the transition kernel defined by $t_m\left(\boldsymbol{x}_m^{(k)} \,\middle|\, \boldsymbol{x}_m^{(k-1)}, \boldsymbol{x}_{\neg m}^{(k,k-1)}\right)$ as an MCMC transition kernel (see Section 2.4.1) for $\mathsf{x}_m \middle| \left(\mathsf{x}_{\neg m} = \boldsymbol{x}_{\neg m}^{(k,k-1)}\right)$ with $\mathsf{x}_{\neg m} = \left[\mathsf{x}_1^\mathrm{T} \ \cdots \ \mathsf{x}_{m-1}^\mathrm{T} \ \mathsf{x}_{m+1}^\mathrm{T} \ \cdots \ \mathsf{x}_{N_\mathrm{B}}^\mathrm{T}\right]^\mathrm{T} \in \mathbb{R}^{X-|B_m|}$ and $\boldsymbol{x}_{\neg m}^{(k,k-1)} = \left[\boldsymbol{x}_1^{(k)\mathrm{T}} \ \cdots \ \boldsymbol{x}_{m-1}^{(k)\mathrm{T}} \ \boldsymbol{x}_{m+1}^{(k-1)\mathrm{T}} \ \cdots \ \boldsymbol{x}_{N_\mathrm{B}}^{(k-1)\mathrm{T}}\right]^\mathrm{T}$, then the transition kernel defined by

$$t(\boldsymbol{x}^{(k)} \,|\, \boldsymbol{x}^{(k-1)}) = \prod_{m=1}^{N_\mathrm{B}} t_m\left(\boldsymbol{x}_m^{(k)} \,\middle|\, \boldsymbol{x}_m^{(k-1)}, \boldsymbol{x}_{\neg m}^{(k,k-1)}\right) \tag{2.45}$$

is an MCMC transition kernel for the complete random vector x. Therefore, at each step $k \geq 2$, each block is sampled separately conditioned on the already sampled blocks of this step and the remaining blocks of the last step. The pseudo-code for one cycle is stated in Algorithm 2.4.

A drawback of this approach is that the Markov chain can move slowly if some blocks are highly dependent. Therefore, it is recommended to sample highly dependent components jointly by combining them into a single block.

One special case of this algorithm is the MH algorithm with kernel cycles, where we use an MH transition kernel for each block. For each block $m \in \{1, \dots, N_\mathrm{B}\}$, we sample a candidate $\tilde{\boldsymbol{x}}_m^{(k)}$ from a conditional proposal pdf $g_m\left(\tilde{\boldsymbol{x}}_m^{(k)} \,\middle|\, \boldsymbol{x}_m^{(k-1)}, \boldsymbol{x}_{\neg m}^{(k,k-1)}\right)$, which will be

Algorithm 2.5 MH algorithm with kernel cycles

Input: $\boldsymbol{x}^{(k-1)}, (g_m)_{m\in\{1,\dots,N_\mathrm{B}\}}$

1: **for all** $m = 1, \dots, N_\mathrm{B}$ **do**
2: sample $\tilde{\boldsymbol{x}}_m^{(k)}$ from $g_m\left(\tilde{\boldsymbol{x}}_m^{(k)} \,\middle|\, \boldsymbol{x}_m^{(k-1)}, \boldsymbol{x}_{\neg m}^{(k,k-1)}\right)$
3: set $\boldsymbol{x}_m^{(k)} = \tilde{\boldsymbol{x}}_m^{(k)}$ with probability $P_{\mathrm{a},m}^{(k)}$ (see (2.46))
else set $\boldsymbol{x}_m^{(k)} = \boldsymbol{x}_m^{(k-1)}$
4: **end for**

Output: $\boldsymbol{x}^{(k)}$

accepted with probability

$$P_{\mathrm{a},m}^{(k)} = \min\left\{1, \frac{f_{\mathbf{x}_m|\mathbf{x}_{\neg m}}\left(\tilde{\boldsymbol{x}}_m^{(k)} \,\middle|\, \boldsymbol{x}_{\neg m}^{(k,k-1)}\right) g_m\left(\boldsymbol{x}_m^{(k-1)} \,\middle|\, \tilde{\boldsymbol{x}}_m^{(k)}, \boldsymbol{x}_{\neg m}^{(k,k-1)}\right)}{f_{\mathbf{x}_m|\mathbf{x}_{\neg m}}\left(\boldsymbol{x}_m^{(k-1)} \,\middle|\, \boldsymbol{x}_{\neg m}^{(k,k-1)}\right) g_m\left(\tilde{\boldsymbol{x}}_m^{(k)} \,\middle|\, \boldsymbol{x}_m^{(k-1)}, \boldsymbol{x}_{\neg m}^{(k,k-1)}\right)}\right\}. \tag{2.46}$$

Note that compared to (2.42), we have the additional condition $\mathbf{x}_{\neg m} = \boldsymbol{x}_{\neg m}^{(k,k-1)}$ in all pdfs. The complete procedure for each step of this algorithm is stated in Algorithm 2.5.

2.4.4 Gibbs Sampling

By further specializing the MH algorithm with kernel cycles, we can derive the Gibbs sampler, where transition kernels $t_\mathrm{m}\left(\boldsymbol{x}_m^{(k)} \,\middle|\, \boldsymbol{x}_m^{(k-1)}, \boldsymbol{x}_{\neg m}^{(k,k-1)}\right)$ are MH transition kernels with the conditional pdf of $\mathbf{x}_m|\mathbf{x}_{\neg m}$ as the proposal pdf, that is, $g_m\left(\tilde{\boldsymbol{x}}_m^{(k)} \,\middle|\, \boldsymbol{x}_m^{(k-1)}, \boldsymbol{x}_{\neg m}^{(k,k-1)}\right) = f_{\mathbf{x}_m|\mathbf{x}_{\neg m}}\left(\tilde{\boldsymbol{x}}_m^{(k)} \,\middle|\, \boldsymbol{x}_{\neg m}^{(k,k-1)}\right)$ (note that g_m no longer involves $\boldsymbol{x}_{\neg m}^{(k,k-1)}$ as a condition). In this case

$$\begin{aligned}
&\frac{f_{\mathbf{x}_m|\mathbf{x}_{\neg m}}\left(\tilde{\boldsymbol{x}}_m^{(k)} \,\middle|\, \boldsymbol{x}_{\neg m}^{(k,k-1)}\right) g_m\left(\boldsymbol{x}_m^{(k-1)} \,\middle|\, \tilde{\boldsymbol{x}}_m^{(k)}, \boldsymbol{x}_{\neg m}^{(k,k-1)}\right)}{f_{\mathbf{x}_m|\mathbf{x}_{\neg m}}\left(\boldsymbol{x}_m^{(k-1)} \,\middle|\, \boldsymbol{x}_{\neg m}^{(k,k-1)}\right) g_m\left(\tilde{\boldsymbol{x}}_m^{(k)} \,\middle|\, \boldsymbol{x}_m^{(k-1)}, \boldsymbol{x}_{\neg m}^{(k,k-1)}\right)} \\
&\quad = \frac{f_{\mathbf{x}_m|\mathbf{x}_{\neg m}}\left(\tilde{\boldsymbol{x}}_m^{(k)} \,\middle|\, \boldsymbol{x}_{\neg m}^{(k,k-1)}\right) f_{\mathbf{x}_m|\mathbf{x}_{\neg m}}\left(\boldsymbol{x}_m^{(k-1)} \,\middle|\, \boldsymbol{x}_{\neg m}^{(k,k-1)}\right)}{f_{\mathbf{x}_m|\mathbf{x}_{\neg m}}\left(\boldsymbol{x}_m^{(k-1)} \,\middle|\, \boldsymbol{x}_{\neg m}^{(k,k-1)}\right) f_{\mathbf{x}_m|\mathbf{x}_{\neg m}}\left(\tilde{\boldsymbol{x}}_m^{(k)} \,\middle|\, \boldsymbol{x}_{\neg m}^{(k,k-1)}\right)} \qquad (2.47) \\
&\quad = 1 \qquad (2.48)
\end{aligned}$$

and therefore the acceptance probability (2.46) is given by

$$P_{\mathrm{a},m}^{(k)} = \min\{1, 1\} = 1. \tag{2.49}$$

This simplifies Algorithm 2.5, as every candidate is accepted. The drawback of this method is that we need to be able to sample each block from the conditional distributions defined by $f_{\mathbf{x}_m|\mathbf{x}_{\neg m}}\left(\boldsymbol{x}_m^{(k)} \,\middle|\, \boldsymbol{x}_{\neg m}^{(k,k-1)}\right)$. The pseudo-code for one iteration of the Gibbs sampler is stated in Algorithm 2.6.

Algorithm 2.6 Gibbs sampler

Input: $\boldsymbol{x}^{(k-1)}$

1: **for all** $m = 1, \ldots, N_{\mathrm{B}}$ **do**

2: sample $\boldsymbol{x}_m^{(k)}$ from $f_{\mathbf{x}_m|\mathbf{x}_{\neg m}}\left(\boldsymbol{x}_m^{(k)} \,\middle|\, \boldsymbol{x}_{\neg m}^{(k,k-1)}\right)$

3: **end for**

Output: $\boldsymbol{x}^{(k)}$

2.4.5 Particle Markov Chain Monte Carlo

Particle MCMC methods, first introduced in [20], are MCMC methods using transition kernels based on SMC. These kernels allow efficient sampling of possibly highly dependent states within an MCMC algorithm. In this section, we will focus on the particle Gibbs sampler with ancestor sampling (PGAS) [21]. Let us, similarly to Section 2.3, consider a sequence of random vectors $\mathbf{x}_{1:N}$, where $\mathbf{x}_n \in \mathbb{R}^X$, with target pdf $f_{\mathbf{x}_{1:N}}(\boldsymbol{x}_{1:N})$ for some $N \in \mathbb{N}$. Our goal is to generate samples $\boldsymbol{x}_{1:N}^{(k)}$, $k = 1, \ldots, K$, of $f_{\mathbf{x}_{1:N}}(\boldsymbol{x}_{1:N})$. The PGAS transition kernel for the states is based on a modified PF with $L \in \mathbb{N}$ particles $\mathbf{x}_{1:n}^{(k,n,l)}$, $l = 1, \ldots, L$, similar to the PF described in Section 2.3.2. The transition kernel of this algorithm is defined by

$$\mathsf{t}\left(\boldsymbol{x}_{1:N}^{(k)} \,\middle|\, \boldsymbol{x}_{1:N}^{(k-1)}\right) = \sum_{l=1}^{L} \mathsf{W}^{(k,N,l)} \delta_{\mathbf{x}_{1:N}^{(k,N,l)}}\left(\boldsymbol{x}_{1:N}^{(k)}\right). \tag{2.50}$$

That is, we sample from a particle approximation of $f_{\mathbf{x}_{1:N}}(\boldsymbol{x}_{1:N})$: we choose $\boldsymbol{x}_{1:N}^{(k)}$ as the particle $\mathbf{x}_{1:N}^{(k,N,l)}$ with probability $\mathsf{W}^{(k,N,l)}$. However, compared to the PF of Section 2.3.2, this transition kernel depends on a complete reference particle $\boldsymbol{x}_{1:N}^{(k-1)}$ in that we deterministically set $\mathbf{x}_n^{(k,n,L)} = \boldsymbol{x}_n^{(k-1)}$ for each $n \in \{1, \ldots, N\}$ during the construction of the particle approximation (2.50). This change results in at least one probable sample at each time n, even for an imprecise proposal density or a low number of particles L.

More specifically, for $n = 1$, we generate the first $L - 1$ particles $\mathbf{x}_1^{(k,1,l)}$ with $l \leq L - 1$ as i.i.d. with proposal pdf $g_1\left(\boldsymbol{x}_1^{(k,1,l)}\right)$. For the last particle, we deterministically set $\mathbf{x}_1^{(k,1,L)} = \boldsymbol{x}_1^{(k-1)}$. Finally, we calculate (see (2.20))

$$\mathsf{W}^{(k,1,l)} \propto \frac{f_{\mathbf{x}_1}\left(\mathbf{x}_1^{(k,1,l)}\right)}{g_1\left(\mathbf{x}_1^{(k,1,l)}\right)} \tag{2.51}$$

for all $l \in \{1, \ldots, L\}$ with $\sum_{l=1}^{L} \mathsf{W}^{(k,1,l)} = 1$.

For $n \in \{2, \ldots, N\}$, we generate particles $\boldsymbol{x}_{1:N}^{(k,N,l)}$ recursively using a modified PF (since $\mathbf{x}_n^{(k,n,L)} = \boldsymbol{x}_n^{(k-1)}$ for each $n \in \{1, \ldots, N\}$). We first perform resampling, that is, we

generate $L-1$ i.i.d. random vectors $\mathbf{x}_{1:n-1}^{(k,n,l)}$, $l = 1, \ldots, L-1$, with pdf (see (2.15))

$$\hat{\mathsf{f}}_{\mathbf{x}_{1:n-1}}^{(k,L)}\left(\boldsymbol{x}_{1:n-1}^{(k,n,l)}\right) = \sum_{l'=1}^{L} \mathsf{W}^{(k,n-1,l')} \delta_{\mathbf{x}_{1:n-1}^{(k,n-1,l')}}\left(\boldsymbol{x}_{1:n-1}^{(k,n,l)}\right), \tag{2.52}$$

that is, we set $\mathbf{x}_{1:n-1}^{(k,n,l)} = \mathbf{x}_{1:n-1}^{(k,n-1,l')}$ with probability $\mathsf{W}^{(k,n-1,l')}$ for all $l' \in \{1, \ldots, L\}$. To complete these particles, we sample $\mathbf{x}_n^{(k,n,l)}$ given $\mathbf{x}_{1:n-1}^{(k,n,l)} = \boldsymbol{x}_{1:n-1}^{(k,n,l)}$ with proposal pdf $g_n\left(\boldsymbol{x}_n^{(k,n,l)} \,\middle|\, \boldsymbol{x}_{1:n-1}^{(k,n,l)}\right)$ for all $l \in \{1, \ldots, L-1\}$ and form the complete particle as $\mathbf{x}_{1:n}^{(k,n,l)} = \left(\mathbf{x}_{1:n-1}^{(k,n,l)}, \mathbf{x}_n^{(k,n,l)}\right)$.

To obtain the last particle $\mathbf{x}_{1:n}^{(k,n,L)}$ with $\mathbf{x}_n^{(k,n,L)} = \boldsymbol{x}_n^{(k-1)}$, we have to resample $\mathbf{x}_{1:n-1}^{(k,n,L)}$ from among the previous particles $\mathbf{x}_{1:n-1}^{(k,n-1,l)}$ conditioned on $\mathbf{x}_n^{(k,n,L)} = \boldsymbol{x}_n^{(k-1)}$. It can be shown that the probability of $\mathbf{x}_{1:n-1}^{(k,n,L)} = \mathbf{x}_{1:n-1}^{(k,n-1,l)}$ given $\mathbf{x}_n^{(k,n,L)} = \boldsymbol{x}_n^{(k-1)}$ is

$$\mathsf{W}'^{(k,n-1,l)} \propto \mathsf{W}^{(k,n-1,l)} f_{\mathbf{x}_{n:N}|\mathbf{x}_{1:n-1}}\left(\boldsymbol{x}_{n:N}^{(k-1)} \,\middle|\, \mathbf{x}_{1:n-1}^{(k,n-1,l)}\right) \tag{2.53}$$

for all $l \in \{1, \ldots, L\}$ with $\sum_{l=1}^{L} \mathsf{W}'^{(k,n-1,l)} = 1$. In other words, we generate $\mathbf{x}_{1:n-1}^{(k,n,L)}$ with pdf

$$\hat{\mathsf{f}}_{\mathbf{x}_{1:n-1}^{(k,n,L)}|\mathbf{x}_n^{(k,n,L)}}^{(k,L)}\left(\boldsymbol{x}_{1:n-1}^{(k,n,L)} \,\middle|\, \boldsymbol{x}_n^{(k-1)}\right) = \sum_{l=1}^{L} \mathsf{W}'^{(k,n-1,l)} \delta_{\mathbf{x}_{1:n-1}^{(k,n-1,l)}}\left(\boldsymbol{x}_{1:n-1}^{(k,n,L)}\right), \tag{2.54}$$

that is, we set $\mathbf{x}_{1:n-1}^{(k,n,L)} = \mathbf{x}_{1:n-1}^{(k,n-1,l)}$ with probability $\mathsf{W}'^{(k,n-1,l)}$. The complete particle is then $\mathbf{x}_{1:n}^{(k,n,L)} = \left(\mathbf{x}_{1:n-1}^{(k,n,L)}, \boldsymbol{x}_n^{(k-1)}\right)$. The associated weights are calculated as (see (2.23))

$$\mathsf{W}^{(k,n,l)} \propto \frac{f_{\mathbf{x}_n|\mathbf{x}_{1:n-1}}\left(\mathbf{x}_n^{(k,n,l)} \,\middle|\, \mathbf{x}_{1:n-1}^{(k,n,l)}\right)}{g_n\left(\mathbf{x}_n^{(k,n,l)} \,\middle|\, \mathbf{x}_{1:n-1}^{(k,n,l)}\right)} \tag{2.55}$$

for all $l \in \{1, \ldots, L\}$ with $\sum_{l=1}^{L} \mathsf{W}^{(k,n,l)} = 1$. Finally, the new sample $\boldsymbol{x}_{1:N}^{(k)}$ is obtained by sampling from the transition kernel $\mathsf{t}\left(\boldsymbol{x}_{1:N}^{(k)} \,\middle|\, \boldsymbol{x}_{1:N}^{(k-1)}\right)$ in (2.50), that is, $\boldsymbol{x}_{1:N}^{(k)}$ is chosen as particle $\mathbf{x}_{1:N}^{(k,N,l)}$ with probability $\mathsf{W}^{(k,N,l)}$, for $l \in \{1, \ldots, L\}$. The algorithm to sample from the PGAS transition kernel is stated in Algorithm 2.7.

Suprisingly, while a transition kernel that directly samples from the PF "approximation" of the target pdf with a finite number of particles would not have the target pdf as an invariant pdf, it can be shown that the addition of the reference particle $\boldsymbol{x}_{1:N}^{(k-1)}$ in the PGAS transition kernel defined by (2.50) ensures this property for all $L \geq 2$ [21, Theorem 1]. Furthermore, the number of particles can usually be chosen much smaller for the PGAS transition kernel than what is necessary for a PF in a similar scenario.

Algorithm 2.7 Particle Gibbs with ancestor sampling

Input: $\boldsymbol{x}_{1:N}^{(k-1)}$, $(g_n)_{n\in\{1,\dots,N\}}$

1: **for all** $l = 1, \dots, L-1$ **do**
2: sample $\boldsymbol{x}_1^{(k,1,l)}$ from $g_1\left(\boldsymbol{x}_1^{(k,1,l)}\right)$
3: **end for**
4: set $\boldsymbol{x}_1^{(k,1,L)} = \boldsymbol{x}_1^{(k-1)}$
5: **for all** $l = 1, \dots, L$ **do**
6: calculate $W^{(k,1,l)}$ according to (2.51)
7: **end for**
8: **for all** $n = 2, \dots, N$ **do**
9: **for all** $l = 1, \dots, L-1$ **do**
10: sample $\boldsymbol{x}_{1:n-1}^{(k,n,l)} = \boldsymbol{x}_{1:n-1}^{(k,n-1,l')}$ with probability $W^{(k,n-1,l')}$ for all $l' \in \{1, \dots, L\}$
11: sample $\boldsymbol{x}_n^{(k,n,l)}$ from $g_n\left(\boldsymbol{x}_n^{(k,n,l)} \,\middle|\, \boldsymbol{x}_{1:n-1}^{(k,n,l)}\right)$
12: set $\boldsymbol{x}_{1:n}^{(k,n,l)} = \left(\boldsymbol{x}_{1:n-1}^{(k,n,l)}, \boldsymbol{x}_n^{(k,n,l)}\right)$
13: **end for**
14: **for all** $l = 1, \dots, L$ **do**
15: calculate $W'^{(k,n-1,l)}$ according to (2.53)
16: **end for**
17: sample $\boldsymbol{x}_{1:n-1}^{(k,n,L)} = \boldsymbol{x}_{1:n-1}^{(k,n-1,l)}$ with probability $W'^{(k,n-1,l)}$ for all $l \in \{1, \dots, L\}$
18: set $\boldsymbol{x}_{1:n}^{(k,n,L)} = \left(\boldsymbol{x}_{1:n-1}^{(k,n,L)}, \boldsymbol{x}_n^{(k-1)}\right)$
19: **for all** $l = 1, \dots, L$ **do**
20: calculate $W^{(k,n,l)}$ according to (2.55)
21: **end for**
22: **end for**
23: sample $\boldsymbol{x}_{1:N}^{(k)} = \boldsymbol{x}_{1:N}^{(k,N,l)}$ with probability $W^{(k,N,l)}$ for all $l \in \{1, \dots, L\}$ (see (2.50))

Output: $\boldsymbol{x}_{1:N}^{(k)}$

2.4.6 Parameter Dependent State-Space Models

As an alternative to the RMPF of Section 2.3.4, we can also use MCMC with the PGAS transition kernel of the previous section for parameter estimation in state-space models.

Let us reconsider the parameter dependent state-space model of Section 2.3.3. That is, we have the sequences of states $(\mathbf{x}_n)_{n\in\mathbb{N}}$, where $\mathbf{x}_n \in \mathbb{R}^X$, and observations $(\mathbf{y}_n)_{n\in\mathbb{N}}$, where $\mathbf{y}_n \in \mathbb{R}^Y$, as well as an additional random parameter $\mathbf{p} \in \mathbb{R}^D$. They are defined by the initial pdf $f_{\mathbf{x}_1|\mathbf{p}}(\boldsymbol{x}_1 \,|\, \mathbf{p}) = \zeta(\boldsymbol{x}_1 \,|\, \mathbf{p})$, the state transition model $f_{\mathbf{x}_n|\mathbf{x}_{n-1},\mathbf{p}}(\boldsymbol{x}_n \,|\, \boldsymbol{x}_{n-1}, \boldsymbol{p}) = \xi(\boldsymbol{x}_n \,|\, \boldsymbol{x}_{n-1}, \boldsymbol{p})$ for $n \geq 2$ and observation model $f_{\mathbf{y}_n|\mathbf{x}_n,\mathbf{p}}(\boldsymbol{y}_n \,|\, \boldsymbol{x}_n, \boldsymbol{p}) = \chi(\boldsymbol{y}_n \,|\, \boldsymbol{x}_n, \boldsymbol{p})$ for $n \geq 1$, and the pdf of the random parameter $f_{\mathbf{p}}(\boldsymbol{p})$. Our target pdf is the joint posterior pdf $f_{\mathbf{x}_{1:N},\mathbf{p}|\mathbf{y}_{1:N}}(\boldsymbol{x}_{1:N}, \boldsymbol{p} \,|\, \boldsymbol{y}_{1:N})$ for a given time interval with length $N \in \mathbb{N}$. We are interested in generating sequences of random vectors $\mathbf{x}_{1:N}^{(k)}$ and parameters $\mathbf{p}^{(k)}$ with $k \in \{1, \ldots, K\}$ to "approximate" the target pdf $f_{\mathbf{x}_{1:N},\mathbf{p}|\mathbf{y}_{1:N}}(\boldsymbol{x}_{1:N}, \boldsymbol{p} \,|\, \boldsymbol{y}_{1:N})$ by (see (2.35))

$$\hat{\mathsf{f}}^{(K)}_{\mathbf{x}_{1:N},\mathbf{p}|\mathbf{y}_{1:N}}(\boldsymbol{x}_{1:N}, \boldsymbol{p} \,|\, \boldsymbol{y}_{1:N}) = \sum_{k=1}^{K} \mathsf{W}^{(k)} \delta_{(\mathbf{x}_{1:N}^{(k)}, \mathbf{p}^{(k)})}(\boldsymbol{x}_{1:N}, \boldsymbol{p}), \tag{2.56}$$

and to approximate an expectation $I_N = \mathbb{E}(h_N(\mathbf{x}_{1:N}, \mathbf{p}) \,|\, \mathbf{y}_{1:N} = \boldsymbol{y}_{1:N})$ by (see (2.36))

$$\begin{aligned} \hat{\mathsf{I}}_N^{(K)} &= \int_{\mathbb{R}^D} \int_{\mathbb{R}^{X\times n}} h_N(\boldsymbol{x}_{1:N}, \boldsymbol{p}) \hat{\mathsf{f}}^{(K)}_{\mathbf{x}_{1:N},\mathbf{p}|\mathbf{y}_{1:N}}(\boldsymbol{x}_{1:N}, \boldsymbol{p} \,|\, \boldsymbol{y}_{1:N}) \mathrm{d}\boldsymbol{x}_{1:N} \mathrm{d}\boldsymbol{p} \\ &= \sum_{k=1}^{K} \mathsf{W}^{(k)} h_N\left(\mathbf{x}_{1:N}^{(k)}, \mathbf{p}^{(k)}\right), \end{aligned} \tag{2.57}$$

where $h_N(\boldsymbol{x}_{1:N}, \boldsymbol{p})$ is an arbitrary function and $\sum_{k=1}^{K} \mathsf{W}^{(k)} = 1$.

To generate samples $\left(\mathbf{x}_{1:N}^{(k)}, \mathbf{p}^{(k)}\right)$, we can use cycles of MCMC kernels (Section 2.4.3), where we first sample $\mathbf{x}_{1:N}^{(k)}$ with Algorithm 2.7 and then $\mathbf{p}^{(k)}$ from an arbitrary MCMC transition kernel $t_{\mathbf{p}}\left(\boldsymbol{p}^{(k)} \,\middle|\, \boldsymbol{p}^{(k-1)}, \boldsymbol{x}_{1:N}^{(k)}, \boldsymbol{y}_{1:N}\right)$. We can make some simplifications for state-space models compared to the general formulation of Algorithm 2.7. For this scenario, the transition kernel is (see (2.50))

$$\mathsf{t}_{\mathrm{PGAS}}\left(\boldsymbol{x}_{1:N}^{(k)} \,\middle|\, \boldsymbol{x}_{1:N}^{(k-1)}, \boldsymbol{p}^{(k-1)}, \boldsymbol{y}_{1:N}\right) = \sum_{l=1}^{L} \mathsf{W}^{(k,N,l)} \delta_{\mathbf{x}_{1:N}^{(k,N,l)}}\left(\boldsymbol{x}_{1:N}^{(k)}\right). \tag{2.58}$$

At the first time step, we use a proposal pdf $g_1\left(\boldsymbol{x}_1^{(k,1,l)} \,\middle|\, \boldsymbol{y}_1, \boldsymbol{p}^{(k-1)}\right)$ and the normalized

weights (2.51) can be simplified to

$$\begin{aligned} \mathsf{W}^{(k,1,l)} &\propto \frac{f_{\mathsf{x}_1|\mathsf{y}_1,\mathsf{p}}\left(\mathbf{x}_1^{(k,1,l)} \,\middle|\, \boldsymbol{y}_1, \boldsymbol{p}^{(k-1)}\right)}{g_1\left(\mathbf{x}_1^{(k,1,l)} \,\middle|\, \boldsymbol{y}_1, \boldsymbol{p}^{(k-1)}\right)} \\ &\propto \frac{f_{\mathsf{y}_1|\mathsf{x}_1,\mathsf{p}}\left(\boldsymbol{y}_1 \,\middle|\, \mathbf{x}_1^{(k,1,l)}, \boldsymbol{p}^{(k-1)}\right) f_{\mathsf{x}_1|\mathsf{p}}\left(\mathbf{x}_1^{(k,1,l)} \,\middle|\, \boldsymbol{p}^{(k-1)}\right)}{g_1\left(\mathbf{x}_1^{(k,1,l)} \,\middle|\, \boldsymbol{y}_1, \boldsymbol{p}^{(k-1)}\right)} \\ &= \frac{\chi\left(\boldsymbol{y}_1 \,\middle|\, \mathbf{x}_1^{(k,1,l)}, \boldsymbol{p}^{(k-1)}\right) \zeta\left(\mathbf{x}_1^{(k,1,l)} \,\middle|\, \boldsymbol{p}^{(k-1)}\right)}{g_1\left(\mathbf{x}_1^{(k,1,l)} \,\middle|\, \boldsymbol{y}_1, \boldsymbol{p}^{(k-1)}\right)} \end{aligned} \tag{2.59}$$

for all $l \in \{1, \ldots, L\}$, where we have used Bayes' theorem in the first step and the properties of our parameter dependent state-space model in the second. For the time steps $n \in \{2, \ldots, N\}$, we use a proposal pdf $g_n\left(\boldsymbol{x}_n^{(k,n,l)} \,\middle|\, \boldsymbol{x}_{1:n-1}^{(k,n,l)}, \boldsymbol{y}_{1:n}, \boldsymbol{p}^{(k-1)}\right)$. We can then express (2.53) as

$$\begin{aligned} \mathsf{W}'^{(k,n-1,l)} &\propto \mathsf{W}^{(k,n-1,l)} f_{\mathsf{x}_n|\mathsf{x}_{n-1},\mathsf{p}}\left(\boldsymbol{x}_n^{(k-1)} \,\middle|\, \boldsymbol{x}_{n-1}^{(k,n-1,l)}, \boldsymbol{p}^{(k-1)}\right) \\ &= \mathsf{W}^{(k,n-1,l)} \xi\left(\boldsymbol{x}_n^{(k-1)} \,\middle|\, \boldsymbol{x}_{n-1}^{(k,n-1,l)}, \boldsymbol{p}^{(k-1)}\right) \end{aligned} \tag{2.60}$$

and simplify (2.55) to

$$\begin{aligned} \mathsf{W}^{(k,n,l)} &\propto \frac{f_{\mathsf{x}_n|\mathsf{x}_{1:n-1},\mathsf{y}_{1:n},\mathsf{p}}\left(\mathbf{x}_n^{(k,n,l)} \,\middle|\, \boldsymbol{x}_{1:n-1}^{(k,n,l)}, \boldsymbol{y}_{1:n}, \boldsymbol{p}^{(k-1)}\right)}{g_n\left(\mathbf{x}_n^{(k,n,l)} \,\middle|\, \boldsymbol{x}_{1:n-1}^{(k,n,l)}, \boldsymbol{y}_{1:n}, \boldsymbol{p}^{(k-1)}\right)} \\ &\propto \frac{f_{\mathsf{y}_n|\mathsf{x}_n,\mathsf{p}}\left(\boldsymbol{y}_n \,\middle|\, \mathbf{x}_n^{(k,n,l)}, \boldsymbol{p}^{(k-1)}\right) f_{\mathsf{x}_n|\mathsf{x}_{n-1},\mathsf{p}}\left(\mathbf{x}_n^{(k,n,l)} \,\middle|\, \boldsymbol{x}_{n-1}^{(k,n,l)}, \boldsymbol{p}^{(k-1)}\right)}{g_n\left(\mathbf{x}_n^{(k,n,l)} \,\middle|\, \boldsymbol{x}_{1:n-1}^{(k,n,l)}, \boldsymbol{y}_{1:n}, \boldsymbol{p}^{(k-1)}\right)} \\ &= \frac{\chi\left(\boldsymbol{y}_n \,\middle|\, \mathbf{x}_n^{(k,n,l)}, \boldsymbol{p}^{(k-1)}\right) \xi\left(\mathbf{x}_n^{(k,n,l)} \,\middle|\, \boldsymbol{x}_{n-1}^{(k,n,l)}, \boldsymbol{p}^{(k-1)}\right)}{g_n\left(\mathbf{x}_n^{(k,n,l)} \,\middle|\, \boldsymbol{x}_{1:n-1}^{(k,n,l)}, \boldsymbol{y}_{1:n}, \boldsymbol{p}^{(k-1)}\right)} \end{aligned} \tag{2.61}$$

for all $l \in \{1, \ldots, L\}$, by again using Bayes' theorem and the properties of our parameter dependent state-space model.

Chapter 3

Bayesian Nonparametrics

This chapter gives an introduction to Bayesian nonparametric (BNP) models. We will base this chapter on the recent book [10] unless stated otherwise. A statistical model is a family of distributions that differ in the choice of a parameter. If this parameter is infinite dimensional, then the statistical model is called nonparametric. We are referring to a model as Bayesian if this parameter is modeled as a random variable defined by a prior distribution. While there are many important BNP models, we will limit this introduction to the Dirichlet process (DP) and the Dirichlet process mixture (DPM). The DP, first introduced in [11], is fundamental to BNP, as it exhibits important properties and serves as a building block in more complex BNP models, such as the DPM. One common application of the DPM is clustering. In contrast to many clustering algorithms, clustering based on the DPM does not require a predefined number of classes and allows the number of classes to grow with an increasing number of data points. Further popular BNP models are the hierarchical DP [10, Example 5.12], the Gaussian process [10, Chapter 11], and the Indian buffet process [10, Section 14.10].

3.1 Dirichlet Process

3.1.1 Construction and Definition

In this section, we will construct and define the DP as a random discrete pdf $\mathsf{f}_{\mathrm{DP}}(\boldsymbol{p})$, closely following the definition in [33]. We note that the DP is usually defined as a random discrete measure, with the most general definition given in [10, Definition 4.1]. However, the definition of the DP as a random discrete measure would require the use of measure theory, which we will avoid by defining the DP as a random discrete pdf.

Let us consider a random discrete pdf on $\mathbb{R}^D$, with $D \in \mathbb{N}$, that can be expressed as

the weighted sum of Dirac deltas

$$\mathsf{f}_{\mathsf{p}}(\boldsymbol{p}) = \sum_{c=1}^{\infty} \mathsf{B}_c \delta_{\mathsf{p}_c^*}(\boldsymbol{p}) \tag{3.1}$$

with random positions $\mathbf{p}_c^* \in \mathbb{R}^D$ and random weights $\mathsf{B}_c \in [0,1]$ for all $c \in \mathbb{N}$ such that

$$\sum_{c=1}^{\infty} \mathsf{B}_c = 1 \quad \text{a.s..} \tag{3.2}$$

One way to construct the random weights $(\mathsf{B}_c)_{c\in\mathbb{N}}$ is the following recursive procedure. The first weight B_1 is equal to V_1, which is a random variable between 0 and 1. For $c \geq 2$, the weight B_c is equal to $1 - \sum_{c'=1}^{c-1} \mathsf{B}_{c'}$ multiplied by the random variable $\mathsf{V}_c \in [0,1]$, that is,

$$\mathsf{B}_c = \mathsf{V}_c \Big(1 - \sum_{c'=1}^{c-1} \mathsf{B}_{c'}\Big). \tag{3.3}$$

It can be shown that we can restate (3.3) as

$$\mathsf{B}_c = \mathsf{V}_c \prod_{c'=1}^{c-1} (1 - \mathsf{V}_{c'}). \tag{3.4}$$

If we choose the random variables $(\mathsf{V}_c)_{c\in\mathbb{N}}$ as independent and such that $\sum_{c=1}^{\infty} \mathbb{E}(\log(1-\mathsf{V}_c)) = -\infty$, then it can be shown, similarly to [34, Lemma 1], that the weights $(\mathsf{B}_c)_{c\in\mathbb{N}}$ satisfy (3.2).

To obtain a further specialized case of feasible random variables $(\mathsf{V}_c)_{c\in\mathbb{N}}$, first note that

$$\mathbb{E}(\log(1-\mathsf{V}_c)) \leq \log(\mathbb{E}(1-\mathsf{V}_c)) = \log(1-\mathbb{E}(\mathsf{V}_c)), \tag{3.5}$$

where we have used Jensen's inequality in the first step. By choosing the random variables $(\mathsf{V}_c)_{c\in\mathbb{N}}$ as i.i.d. with $\mathbb{E}(\mathsf{V}_c) > 0$, we have

$$\sum_{c=1}^{\infty} \mathbb{E}(\log(1-\mathsf{V}_c)) \leq \sum_{c=1}^{\infty} \log(1-\mathbb{E}(\mathsf{V}_c)) = \log(1-\mathbb{E}(\mathsf{V}_1)) \sum_{c=1}^{\infty} 1 = -\infty, \tag{3.6}$$

where we have used (3.5) in the first step. Therefore, we can, for example, ensure (3.2) by choosing the random variables $(\mathsf{V}_c)_{c\in\mathbb{N}}$ as i.i.d. with $\mathbb{E}(\mathsf{V}_c) > 0$.

This recursive procedure to construct the weights $(\mathsf{B}_c)_{c\in\mathbb{N}}$ is often called **stick breaking**, as the weights can be imagined as pieces of a stick with total length 1. A graphical representation of the stick-breaking analogy is provided in Figure 3.1.

Consider now the special case of the stick breaking procedure with i.i.d. random variables $(\mathsf{V}_c)_{c\in\mathbb{N}}$ with a Beta$(1,\alpha)$ distribution, where $\alpha \in \mathbb{R}^+$. The pdf of V_c is

$$f_{\mathsf{V}_c}(V_c) = \frac{\Gamma(1+\alpha)}{\Gamma(\alpha)} (1-V_c)^{\alpha-1} \tag{3.7}$$

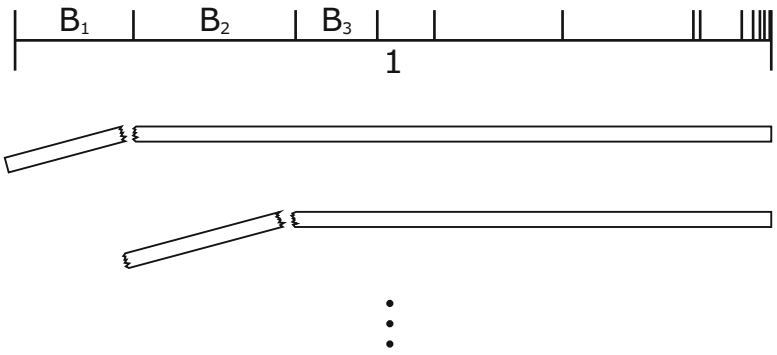

Fig. 3.1: *Graphical representation of the stick breaking procedure.*

for all $V_c \in [0,1]$ and $c \in \mathbb{N}$, where $\Gamma(x) = \int_0^\infty y^{x-1}e^{-y}\mathrm{d}y$ for all $x \in \mathbb{R}^+$. The expectation of V_c is

$$\mathbb{E}(\mathsf{V}_c) = \frac{1}{1+\alpha} > 0. \tag{3.8}$$

Further consider i.i.d. random positions $\mathbf{p}_c^*$ with some pdf $f_{\mathrm{G}}(\boldsymbol{p}_c^*)$. The resulting random pdf

$$\mathsf{f}_{\mathrm{DP}}(\boldsymbol{p}) = \sum_{c=1}^{\infty} \mathsf{B}_c \delta_{\mathbf{p}_c^*}(\boldsymbol{p}) \tag{3.9}$$

is called a DP with base pdf $f_{\mathrm{G}}(\boldsymbol{p})$ and concentration α. Similarly to Chapter 2, note that while we use a pdf $f_{\mathrm{G}}(\boldsymbol{p})$ for simplicity in this chapter, we can also, with the obvious changes, use densities of other types of random variables. The distribution of the DP $\mathsf{f}_{\mathrm{DP}}(\boldsymbol{p})$ is briefly denoted as $\mathrm{DP}(\alpha, f_{\mathrm{G}})$. This choice of a $\mathrm{Beta}(1,\alpha)$ distribution will allow us to formulate important properties of the DP later in this chapter. The concentration α determines the weight distribution of the DP $\mathsf{f}_{\mathrm{DP}}(\boldsymbol{p})$, where a small α leads to more weight assigned to small indices c. Note that for $\alpha = 1$, in particular, the random variables V_c are distributed according to a $\mathrm{Beta}(1,1)$ distribution, which is the uniform distribution on the interval $[0,1]$. Further note that if the base pdf $f_{\mathrm{G}}(\boldsymbol{p})$ is continuous, then it can be shown that there does not exist a pdf defining the distribution $\mathrm{DP}(\alpha, f_{\mathrm{G}})$. The above definition of the DP is restated in Definition 3.1.

Definition 3.1: *Let $\alpha \in \mathbb{R}^+$, $D \in \mathbb{N}$, and $f_{\mathrm{G}}(\boldsymbol{p})$ be a pdf on $\mathbb{R}^D$. The random pdf*

$$\mathsf{f}_{\mathrm{DP}}(\boldsymbol{p}) = \sum_{c=1}^{\infty} \mathsf{B}_c \delta_{\mathbf{p}_c^*}(\boldsymbol{p}) \tag{3.10}$$

with

$$\mathsf{B}_1 = \mathsf{V}_1, \quad \mathsf{B}_c = \mathsf{V}_c \prod_{c'=1}^{c-1} (1 - \mathsf{V}_{c'}) \quad \textit{for all } c \geq 2, \tag{3.11}$$

where

$$\mathsf{V}_1, \mathsf{V}_2, \ldots \overset{\text{i.i.d.}}{\sim} \text{Beta}(1, \alpha), \tag{3.12}$$

and with

$$\mathbf{p}_1^*, \mathbf{p}_2^*, \ldots \overset{\text{i.i.d.}}{\sim} f_{\text{G}} \tag{3.13}$$

is called a ***Dirichlet process (DP)*** *with* ***base pdf*** $f_{\text{G}}(\boldsymbol{p})$ *and* ***concentration*** α*, and we denote its distribution by* $\text{DP}(\alpha, \mathsf{f}_{\text{G}})$*, that is,*

$$\mathsf{f}_{\text{DP}} \sim \text{DP}(\alpha, f_{\text{G}}). \tag{3.14}$$

3.1.2 Properties

Consider the following construction of random vectors $\mathbf{p}_i \in \mathbb{R}^D$ for all $i \in \mathbb{N}$. First we define a DP

$$\mathsf{f}_{\text{DP}} \sim \text{DP}(\alpha, f_{\text{G}}) \tag{3.15}$$

with $\alpha \in \mathbb{R}^+$ and the continuous pdf $f_{\text{G}}(\boldsymbol{p})$. Then, we let the random vectors $(\mathbf{p}_i)_{i \in \mathbb{N}}$ be conditionally i.i.d. with pdf f_{DP}, that is,

$$\mathbf{p}_1, \mathbf{p}_2, \ldots \,|\, (\mathsf{f}_{\text{DP}} = f_{\text{DP}}) \overset{\text{i.i.d.}}{\sim} f_{\text{DP}}. \tag{3.16}$$

Note that we require $f_{\text{G}}(\boldsymbol{p})$ to be continuous to ensure that $\mathbf{p}_{c'}^* \neq \mathbf{p}_{c''}^*$ a.s. iff $c' \neq c''$. Therefore, if we use a density of a different type of random variable instead of $f_{\text{G}}(\boldsymbol{p})$, then we have to choose it such that we have $\mathbf{p}_{c'}^* \neq \mathbf{p}_{c''}^*$ a.s. iff $c' \neq c''$.

Marginal Distribution According to (3.10), we have

$$\mathsf{f}_{\text{DP}}(\boldsymbol{p}) = \sum_{c=1}^{\infty} \mathsf{B}_c \delta_{\mathbf{p}_c^*}(\boldsymbol{p}) \tag{3.17}$$

for some random weights $(\mathsf{B}_c)_{c \in \mathbb{N}}$ and random positions $(\mathbf{p}_c^*)_{c \in \mathbb{N}}$. It therefore follows that, for each $i \in \mathbb{N}$,

$$\mathbf{p}_i = \mathbf{p}_c^* \text{ with probability } \mathsf{B}_c \text{ for all } c \in \mathbb{N}. \tag{3.18}$$

As each $\mathbf{p}_c^*$ is individually distributed according to the pdf $f_{\text{G}}(\boldsymbol{p}_c^*)$, the marginal pdf of each $\mathbf{p}_i$ is also $f_{\text{G}}(\boldsymbol{p}_i)$, that is,

$$f_{\mathbf{p}_i}(\boldsymbol{p}_i) = f_{\text{G}}(\boldsymbol{p}_i). \tag{3.19}$$

Number of Distinct Values If we denote the number of distinct values among the first I random vectors $\mathbf{p}_1, \ldots, \mathbf{p}_I$ as $\mathsf{C}_I \in \mathbb{N}$, then it can be shown that $\mathbb{E}(\mathsf{C}_I)/(\alpha \log(I)) \to 1$ [10, Proposition 4.8], that is, the expected number of distinct values C_I grows logarithmically and is close to $\alpha \log(I)$ for large I.

Posterior Distribution Another important property is the distribution of the random pdf f_{DP} given $\mathbf{p}_{1:I}$, for some $I \in \mathbb{N}$. As shown in [11, Theorem 1], this random pdf is again a DP with the same concentration α but a different base pdf, that is,

$$\mathsf{f}_{\mathrm{DP}} \,|\, (\mathbf{p}_{1:I} = \boldsymbol{p}_{1:I}) \sim \mathrm{DP}(\alpha, \tilde{f}_{\mathrm{G},I}) \tag{3.20}$$

with

$$\tilde{f}_{\mathrm{G},I}(\boldsymbol{p}) = \frac{1}{\alpha + I}\left(\alpha f_{\mathrm{G}}(\boldsymbol{p}) + \sum_{i=1}^{I} \delta_{\boldsymbol{p}_i}(\boldsymbol{p})\right). \tag{3.21}$$

To further elucidate this property, we can use (3.18) to express (3.21) as

$$\begin{aligned}\tilde{f}_{\mathrm{G},I}(\boldsymbol{p}) &= \frac{1}{\alpha + I}\left(\alpha f_{\mathrm{G}}(\boldsymbol{p}) + I \sum_{i=1}^{I} \frac{1}{I}\delta_{\boldsymbol{p}_i}(\boldsymbol{p})\right) \\ &= \frac{1}{\alpha + I}\left(\alpha f_{\mathrm{G}}(\boldsymbol{p}) + I \sum_{c=1}^{C_I} \hat{B}_{I,c}\delta_{\boldsymbol{p}_c^*}(\boldsymbol{p})\right),\end{aligned} \tag{3.22}$$

where C_I is the number of distinct values $\boldsymbol{p}_c^*$ among the vectors $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_I$ and $\hat{B}_{I,c}$ is the empirical probability of $\boldsymbol{p}_c^*$, that is,

$$\hat{B}_{I,c} = \frac{1}{I}\sum_{i=1}^{I} \mathbb{1}(\boldsymbol{p}_i = \boldsymbol{p}_c^*). \tag{3.23}$$

It follows that (3.22), and therefore also (3.20), is invariant under permutation of the given samples $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_I$.

Induced Partition One property of the DP is that it induces a random partition[1] on the natural numbers $\mathbb{N}$. As stated above, each random vector $\mathbf{p}_i$ is equal to $\mathbf{p}_c^*$ with probability B_c for all $c \in \mathbb{N}$. This motivates an equivalent formulation of (3.15) and (3.16) using latent indicator variables $\mathsf{c}_1, \mathsf{c}_2, \ldots \in \mathbb{N}$. More specifically, with $(\mathsf{B}_c)_{c\in\mathbb{N}}$ and $(\mathbf{p}_c^*)_{c\in\mathbb{N}}$ as in Definition 3.1, we define c_i conditionally i.i.d. given $(\mathsf{B}_c)_{c\in\mathbb{N}}$ according to

$$\mathbb{P}\left(\mathsf{c}_i = c \,\middle|\, \left((\mathsf{B}_c)_{c\in\mathbb{N}} = (B_c)_{c\in\mathbb{N}}\right)\right) = B_c \tag{3.24}$$

[1] A partition of a set $\mathfrak{X}$ is a set $\mathfrak{P}$ of non-empty and disjoint subsets of $\mathfrak{X}$ such that $\bigcup_{\mathcal{X}\in\mathfrak{P}} \mathcal{X} = \mathfrak{X}$.

for all $c \in \mathbb{N}$, and we set

$$\mathbf{p}_i = \mathbf{p}^*_{\mathsf{c}_i} \tag{3.25}$$

for all $i \in \mathbb{N}$. This formulation shows that the DP naturally induces a random distribution on the set of all partitions of $\mathbb{N}$ by grouping indices i with equal indicator variables c_i. If we choose an arbitrary continuous base pdf $f_{\mathrm{G}}(\boldsymbol{p})$ and concentration α for the DP, then the induced random distribution on the set of all partitions is called the Chinese restaurant process (CRP) with concentration α [10, Section 14.1.1]. The distribution of the CRP is only parametrized by the concentration α as the random vectors $(\mathbf{p}^*_c)_{c\in\mathbb{N}}$ in Definition 3.1 have no influence on the CRP, provided the base pdf $f_{\mathrm{G}}(\boldsymbol{p})$ is continuous. Note that the resulting partition has C_I subsets for the first I random vectors $\mathbf{p}_{1:I}$ and is invariant under permutation of the indicator variables $(\mathsf{c}_i)_{i\in\mathbb{N}}$ as they are conditionally i.i.d. given $(\mathsf{B}_c)_{c\in\mathbb{N}}$.

3.1.3 Sampling from the Dirichlet Process

We are often interested in generating samples from a statistical model. One way to generate samples $\boldsymbol{p}_{1:I}$ from a DP would be to first sample the pdf f_{DP} from $\mathrm{DP}(\alpha, f_{\mathrm{G}})$ (see (3.15)), and then generating the samples $\boldsymbol{p}_{1:I}$ from $f_{\mathrm{DP}}(\boldsymbol{p})$ (see (3.16)). This, however, is infeasible in practice as the pdf $f_{\mathrm{DP}}(\boldsymbol{p})$ includes an infinite number of Dirac deltas.

Another approach to generating the samples, which is feasible in practice as it avoids sampling a complete DP, is based on the DP's posterior distribution (see (3.20)). By applying the chain rule, we have

$$f_{\mathbf{p}_{1:I}}(\boldsymbol{p}_{1:I}) = f_{\mathbf{p}_1}(\boldsymbol{p}_1) \prod_{i=2}^{I} f_{\mathbf{p}_i|\mathbf{p}_{1:i-1}}(\boldsymbol{p}_i \,|\, \boldsymbol{p}_{1:i-1}). \tag{3.26}$$

Therefore, we can start by sampling $\mathbf{p}_1$ from $f_{\mathbf{p}_1}(\boldsymbol{p}_1)$, and then sample $\mathbf{p}_i$ given $\mathbf{p}_{1:i-1}$ from $f_{\mathbf{p}_i|\mathbf{p}_{1:i-1}}(\boldsymbol{p}_i \,|\, \boldsymbol{p}_{1:i-1})$ for all $i \in \{2, \ldots, I\}$. These pdfs can be simplified by using the properties discussed in Section 3.1.2. Using (3.19), we can express $f_{\mathbf{p}_1}(\boldsymbol{p}_1)$ as

$$f_{\mathbf{p}_1}(\boldsymbol{p}_1) = f_{\mathrm{G}}(\boldsymbol{p}_1). \tag{3.27}$$

To simplify $f_{\mathbf{p}_i|\mathbf{p}_{1:i-1}}(\boldsymbol{p}_i \,|\, \boldsymbol{p}_{1:i-1})$, first note that according to (3.20)–(3.23), with $i-1$ instead of I, we have

$$\mathsf{f}_{\mathrm{DP}} \,|\, (\mathbf{p}_{1:i-1} = \boldsymbol{p}_{1:i-1}) \sim \mathrm{DP}(\alpha, \tilde{f}_{\mathrm{G},i-1}), \tag{3.28}$$

with

$$\tilde{f}_{\mathrm{G},i-1}(\boldsymbol{p}) = \frac{1}{\alpha + i - 1}\left(\alpha f_{\mathrm{G}}(\boldsymbol{p}) + (i-1)\sum_{c=1}^{C_{i-1}} \hat{B}_{i-1,c}\delta_{\boldsymbol{p}_c^*}(\boldsymbol{p})\right), \tag{3.29}$$

where C_{i-1} is the number of distinct values $\boldsymbol{p}_c^*$ among the vectors $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_{i-1}$ and

$$\hat{B}_{i-1,c} = \frac{1}{i-1}\sum_{i'=1}^{i-1} \mathbb{1}(\boldsymbol{p}_{i'} = \boldsymbol{p}_c^*). \tag{3.30}$$

As $\mathbf{p}_i|\mathbf{p}_{1:i-1}$ is distributed according to (3.28), we can apply (3.19) to (3.28) and obtain

$$f_{\mathbf{p}_i|\mathbf{p}_{1:i-1}}(\boldsymbol{p}_i \,|\, \boldsymbol{p}_{1:i-1}) = \tilde{f}_{\mathrm{G},i-1}(\boldsymbol{p}_i). \tag{3.31}$$

In other words, it follows from (3.27) that we sample $\boldsymbol{p}_1$ from $f_{\mathrm{G}}(\boldsymbol{p})$. Similarly, for $i \in \{2, \ldots, I\}$, it follows from (3.31) and (3.29) that we either sample $\boldsymbol{p}_i$ from $f_{\mathrm{G}}(\boldsymbol{p})$ with probability $\frac{\alpha}{\alpha+i-1}$, or we set $\boldsymbol{p}_i = \boldsymbol{p}_c^*$ with probability $\frac{(i-1)\hat{B}_{i-1,c}}{\alpha+i-1}$ for all $c \in \{1, \ldots, C_{i-1}\}$.

3.2 Dirichlet Process Mixture

3.2.1 Construction and Definition

One important application of the DP is its use as the pdf of the prior distribution for a parameter $\mathbf{p} \in \mathbb{R}^D$ of a statistical model for a random vector $\mathbf{x} \in \mathbb{R}^X$. More specifically, in this case, the parameter $\mathbf{p}$ is distributed according to the DP $\mathsf{f}_{\mathrm{DP}} \sim \mathrm{DP}(\alpha, f_{\mathrm{G}})$, that is,

$$\mathbf{p} \,|\, (\mathsf{f}_{\mathrm{DP}} = f_{\mathrm{DP}}) \sim f_{\mathrm{DP}}. \tag{3.32}$$

Further, the random vector $\mathbf{x}$ is parametrized by $\mathbf{p}$, that is,

$$\mathbf{x} \,|\, (\mathbf{p} = \boldsymbol{p}) \sim \psi(\cdot \,|\, \boldsymbol{p}) \tag{3.33}$$

with some pdf $\psi(\boldsymbol{x} \,|\, \boldsymbol{p})$. Note that instead of a pdf $\psi(\boldsymbol{x} \,|\, \boldsymbol{p})$, we can also, with the obvious changes, choose a density of a different type of random variable. Consistently with (3.32) and (3.33), we assume that given $\mathbf{p}$, $\mathbf{x}$ is conditionally independent of f_{DP}. It follows from (3.32) and (3.33) that the pdf of $\mathbf{x}$ given $\mathsf{f}_{\mathrm{DP}}(\boldsymbol{p})$ can be expressed as

$$\begin{aligned} f_{\mathbf{x}|\mathsf{f}_{\mathrm{DP}}}(\boldsymbol{x} \,|\, f_{\mathrm{DP}}) &= \int_{\mathbb{R}^D} f_{\mathbf{x}|\mathbf{p},\mathsf{f}_{\mathrm{DP}}}(\boldsymbol{x} \,|\, \boldsymbol{p}, f_{\mathrm{DP}}) f_{\mathbf{p}|\mathsf{f}_{\mathrm{DP}}}(\boldsymbol{p} \,|\, f_{\mathrm{DP}}) \mathrm{d}\boldsymbol{p} \\ &= \int_{\mathbb{R}^D} f_{\mathbf{x}|\mathbf{p}}(\boldsymbol{x} \,|\, \boldsymbol{p}) f_{\mathrm{DP}}(\boldsymbol{p}) \mathrm{d}\boldsymbol{p} \\ &= \int_{\mathbb{R}^D} \psi(\boldsymbol{x} \,|\, \boldsymbol{p}) f_{\mathrm{DP}}(\boldsymbol{p}) \mathrm{d}\boldsymbol{p}. \end{aligned} \tag{3.34}$$

Using (3.10), we finally obtain

$$\begin{aligned} f_{\mathbf{x}|\mathsf{f}_{\mathrm{DP}}}(\boldsymbol{x} \,|\, f_{\mathrm{DP}}) &= \int_{\mathbb{R}^D} \psi(\boldsymbol{x} \,|\, \boldsymbol{p}) \sum_{c=1}^{\infty} B_c \delta_{\boldsymbol{p}_c^*}(\boldsymbol{p}) \mathrm{d}\boldsymbol{p} \\ &= \sum_{c=1}^{\infty} B_c \int_{\mathbb{R}^D} \psi(\boldsymbol{x} \,|\, \boldsymbol{p}) \delta_{\boldsymbol{p}_c^*}(\boldsymbol{p}) \mathrm{d}\boldsymbol{p} \\ &= \sum_{c=1}^{\infty} B_c \psi(\boldsymbol{x} \,|\, \boldsymbol{p}_c^*) \end{aligned} \tag{3.35}$$

for some $(B_c)_{c\in\mathbb{N}}$ and $(\boldsymbol{p}_c^*)_{c\in\mathbb{N}}$. As (3.35) is a mixture based on a DP, this model is called a Dirichlet process mixture (DPM). This definition of the DPM is restated more concisely in Definition 3.2.

Definition 3.2: *Let $\alpha \in \mathbb{R}^+$; $D, N \in \mathbb{N}$; $f_{\mathrm{G}}(\boldsymbol{p})$ be a pdf on $\mathbb{R}^D$; and $\psi(\boldsymbol{x} \,|\, \boldsymbol{p})$ be a pdf on $\mathbb{R}^X$ for each $\boldsymbol{p} \in \mathbb{R}^D$. A random pdf defined by $\psi(\boldsymbol{x} \,|\, \mathbf{p})$, with $\mathbf{p}|(\mathsf{f}_{\mathrm{DP}} = f_{\mathrm{DP}}) \sim f_{\mathrm{DP}}$ and $\mathsf{f}_{\mathrm{DP}} \sim \mathrm{DP}(\alpha, f_{\mathrm{G}})$ is called a **Dirichlet process mixture (DPM)**.*

3.2.2 Inference

Consider a DP

$$\mathsf{f}_{\mathrm{DP}} \sim \mathrm{DP}(\alpha, f_{\mathrm{G}}) \tag{3.36}$$

with some $\alpha \in \mathbb{R}^+$, some continuous pdf $f_{\mathrm{G}}(\boldsymbol{p})$, and conditionally i.i.d. random vectors $(\mathbf{p}_i)_{i\in\mathbb{N}}$ such that

$$\mathbf{p}_1, \mathbf{p}_2, \ldots \,|\, (\mathsf{f}_{\mathrm{DP}} = f_{\mathrm{DP}}) \overset{\text{i.i.d.}}{\sim} f_{\mathrm{DP}}. \tag{3.37}$$

Further consider conditionally i.i.d. random vectors $(\mathbf{x}_i)_{i\in\mathbb{N}}$ with a distribution parametrized by $(\mathbf{p}_i)_{i\in\mathbb{N}}$, that is,

$$\mathbf{x}_i \,|\, (\mathbf{p}_i = \boldsymbol{p}_i) \sim \psi(\cdot \,|\, \boldsymbol{p}_i) \tag{3.38}$$

for all $i \in \mathbb{N}$, with some continuous pdf $\psi(\boldsymbol{x} \,|\, \boldsymbol{p})$ for $\boldsymbol{p} \in \mathbb{R}^D$ and with $\mathbf{x}_i$ is conditionally independent of $\mathbf{p}_i'$ for all $i' \neq i$ given $\mathbf{p}_i$. That is, the random vectors $(\mathbf{x}_i)_{i\in\mathbb{N}}$ are distributed according to a DPM.

Similarly to Chapter 2, we are mainly interested in generating samples $\boldsymbol{p}_{1:I}^{(k)}$ for $k \in \{1, \ldots, K\}$, or equivalently, $c_{1:I}^{(k)}$ and $\boldsymbol{p}_c^{(k)}$ for $c \in \left\{c_1^{(k)}, \ldots, c_I^{(k)}\right\}$ and $k \in \{1, \ldots, K\}$ (see Section 3.1.2), to obtain MC approximations. For this, we will discuss two Gibbs sampling algorithms below.

Algorithm 3.1 Gibbs sampler for DPM

Input: $\boldsymbol{p}_{1:I}^{(k-1)}, \boldsymbol{x}_{1:I}$

1: **for all** $i = 1, \ldots, I$ **do**

2: sample $\boldsymbol{p}_i^{(k)}$ from $f_{\mathbf{p}_i|\mathbf{p}_{\neg i},\mathbf{x}_{1:I}}\left(\boldsymbol{p}_i^{(k)} \,|\, \boldsymbol{p}_{\neg i}^{(k)}, \boldsymbol{x}_{1:I}\right)$ (see (3.39))

3: **end for**

Output: $\boldsymbol{p}_{1:I}^{(k)}$

Gibbs Sampler It can be shown that [10, Theorem 5.3]

$$\begin{aligned} f_{\mathbf{p}_i|\mathbf{p}_{\neg i},\mathbf{x}_{1:I}}(\boldsymbol{p}_i \,|\, \boldsymbol{p}_{\neg i}, \boldsymbol{x}_{1:I}) &= f_{\mathbf{p}_i|\mathbf{p}_{\neg i},\mathbf{x}_i}(\boldsymbol{p}_i \,|\, \boldsymbol{p}_{\neg i}, \boldsymbol{x}_i) \\ &\propto \frac{1}{\alpha + I - 1}\left(\alpha\psi(\boldsymbol{x}_i \,|\, \boldsymbol{p}_i) f_{\mathrm{G}}(\boldsymbol{p}_i) + \sum_{i' \in \{1,\ldots,I\}\setminus\{i\}} \psi(\boldsymbol{x}_i \,|\, \boldsymbol{p}_{i'})\delta_{\boldsymbol{p}_{i'}}(\boldsymbol{p}_i)\right), \end{aligned} \tag{3.39}$$

where $\mathbf{p}_{\neg i} = (\mathbf{p}_1, \ldots, \mathbf{p}_{i-1}, \mathbf{p}_{i+1}, \ldots, \mathbf{p}_I)$. Therefore, $\mathbf{p}_i$ given $\mathbf{p}_{\neg i}$ and $\mathbf{x}_{1:I}$ is distributed according to a pdf proportional to $\psi(\boldsymbol{x}_i \,|\, \boldsymbol{p}_i) f_{\mathrm{G}}(\boldsymbol{p}_i)$ with probability

$$b_{i,0} \propto \frac{\alpha}{\alpha + I - 1} \tag{3.40}$$

and equal to $\boldsymbol{p}_{i'}$, for $i' \in \{1, \ldots, I\} \setminus \{i\}$, with probability

$$b_{i,i'} \propto \frac{\psi(\boldsymbol{x}_i \,|\, \boldsymbol{p}_{i'})}{\alpha + I - 1}, \tag{3.41}$$

where

$$\sum_{i' \in \{0,1,\ldots,I\}\setminus\{i\}} b_{i,i'} = 1. \tag{3.42}$$

If we are able to sample from (3.39), then we can use Gibbs sampling (see Section 2.4.4) to generate the samples $\boldsymbol{p}_{1:I}^{(k)}$ [22]. The procedure for each step of this Gibbs sampler is stated in Algorithm 3.1, where $\boldsymbol{p}_{\neg i}^{(k)} = \left(\boldsymbol{p}_1^{(k)}, \ldots, \boldsymbol{p}_{i-1}^{(k)}, \boldsymbol{p}_{i+1}^{(k-1)}, \ldots, \mathbf{p}_I^{(k-1)}\right)$. One drawback of this algorithm is its slow convergence as it updates each $\boldsymbol{p}_i^{(k)}$ separately, even though several of them can be equal. This algorithm is especially applicable if the base pdf $f_{\mathrm{G}}(\boldsymbol{p})$ is a conjugate prior[2] of $\psi(\boldsymbol{x} \,|\, \boldsymbol{p})$, as this will typically simplify sampling from $\psi(\boldsymbol{x}_i \,|\, \boldsymbol{p}_i) f_{\mathrm{G}}(\boldsymbol{p}_i)$ in (3.39).

[2]Let $\mathbf{q} \sim f_{\mathrm{G}}$ and $\mathbf{r}_1, \ldots, \mathbf{r}_J | (\mathbf{q} = \boldsymbol{q}) \overset{\text{i.i.d.}}{\sim} \psi(\cdot \,|\, \boldsymbol{q})$ with $J \in \mathbb{N}$. The pdf $f_{\mathrm{G}}(\boldsymbol{q})$ is said to be a conjugate prior of $\psi(\boldsymbol{r} \,|\, \boldsymbol{q})$ if the posterior distribution $f_{\mathbf{q}|\mathbf{r}_{1:J}}(\boldsymbol{q} \,|\, \boldsymbol{r}_{1:J})$ is in the same family of pdfs as $f_{\mathrm{G}}(\boldsymbol{q})$. That is, $f_{\mathrm{G}}(\boldsymbol{q}) = \phi(\boldsymbol{q} \,|\, \boldsymbol{u})$ and $f_{\mathbf{q}|\mathbf{r}_{1:J}}(\boldsymbol{q} \,|\, \boldsymbol{r}_{1:J}) = \phi(\boldsymbol{q} \,|\, \boldsymbol{u}')$, for all $\boldsymbol{r}_{1:J}$, for some parametrized pdf $\phi(\boldsymbol{q} \,|\, \boldsymbol{u}')$ and some parameters $\boldsymbol{u}$ and $\boldsymbol{u}'$.

Gibbs Sampler Using Indicator Variables Similarly to (3.24) and (3.25), it is possible to introduce indicator variables to obtain an equivalent formulation of the model (3.36)–(3.38), that is, we define c_i conditionally i.i.d. given $(\mathsf{B}_c)_{c\in\mathbb{N}}$ with

$$\mathbb{P}\left(\mathsf{c}_i = c \middle| \left((\mathsf{B}_c)_{c\in\mathbb{N}} = (B_c)_{c\in\mathbb{N}}\right)\right) = B_c \tag{3.43}$$

for all $c \in \mathbb{N}$, and

$$\mathbf{p}_i = \mathbf{p}^*_{\mathsf{c}_i}, \tag{3.44}$$

$$\mathbf{x}_i \,|\, (\mathbf{p}_i = \boldsymbol{p}_i) \sim \psi(\cdot \,|\, \boldsymbol{p}_i), \tag{3.45}$$

for all $i \in \mathbb{N}$, with $(\mathsf{B}_c)_{c\in\mathbb{N}}$ and $(\mathbf{p}^*_c)_{c\in\mathbb{N}}$ as in Definition 3.1. We can use this model to construct an alternative Gibbs sampler [23], where we first sample the indicator variables c_i, and then the parameters $\mathbf{p}^*_c$ for each distinct value of the c_i.

We start by noting that it follows from (3.43)–(3.45) that

$$f_{\mathbf{p}_i|\mathbf{p}_{\neg i},\mathbf{x}_{1:I}}(\boldsymbol{p}_i \,|\, \boldsymbol{p}_{\neg i}, \boldsymbol{x}_{1:I}) = f_{\mathbf{p}_i|\mathsf{c}_{\neg i},\mathbf{p}^*_{\mathsf{c}_{\neg i}},\mathbf{x}_{1:I}}(\boldsymbol{p}_i \,|\, \boldsymbol{c}_{\neg i}, \boldsymbol{p}^*_{\boldsymbol{c}_{\neg i}}, \boldsymbol{x}_{1:I}), \tag{3.46}$$

where $\mathbf{c}_{\neg i} = (\mathsf{c}_1, \ldots, \mathsf{c}_{i-1}, \mathsf{c}_{i+1}, \ldots, \mathsf{c}_I)$ and $\mathbf{p}^*_{\mathbf{c}_{\neg i}} = (\mathbf{p}^*_c)_{c\in\{\mathsf{c}_1,\ldots,\mathsf{c}_{i-1},\mathsf{c}_{i+1},\ldots,\mathsf{c}_I\}}$. Further, we can restate (3.39) using the indicator variables c_i and the parameters $\mathbf{p}^*_c$ according to

$$\begin{aligned}
&f_{\mathbf{p}_i|\mathbf{p}_{\neg i},\mathbf{x}_{1:I}}(\boldsymbol{p}_i \,|\, \boldsymbol{p}_{\neg i}, \boldsymbol{x}_{1:I}) \\
&\quad\propto \frac{1}{\alpha + I - 1}\left(\alpha\psi(\boldsymbol{x}_i \,|\, \boldsymbol{p}_i) f_{\mathrm{G}}(\boldsymbol{p}_i) + \sum_{i'\in\{1,\ldots,I\}\setminus\{i\}} \psi(\boldsymbol{x}_i \,|\, \boldsymbol{p}^*_{c_{i'}}) \delta_{\boldsymbol{p}^*_{c_{i'}}}(\boldsymbol{p}_i)\right) \\
&\quad= \frac{1}{\alpha + I - 1}\Bigg(\alpha\psi(\boldsymbol{x}_i \,|\, \boldsymbol{p}_i) f_{\mathrm{G}}(\boldsymbol{p}_i) \\
&\qquad + \sum_{c\in\{c_1,\ldots,c_{i-1},c_{i+1},\ldots,c_I\}} \left(\sum_{i'\in\{1,\ldots,I\}\setminus\{i\}} \mathbb{1}(c_{i'} = c)\right) \psi(\boldsymbol{x}_i \,|\, \boldsymbol{p}^*_c) \delta_{\boldsymbol{p}^*_c}(\boldsymbol{p}_i)\Bigg).
\end{aligned} \tag{3.47}$$

Combining (3.46) and (3.47), we obtain

$$\begin{aligned}
&f_{\mathbf{p}_i|\mathsf{c}_{\neg i},\mathbf{p}^*_{\mathsf{c}_{\neg i}},\mathbf{x}_{1:I}}(\boldsymbol{p}_i \,|\, \boldsymbol{c}_{\neg i}, \boldsymbol{p}^*_{\boldsymbol{c}_{\neg i}}, \boldsymbol{x}_{1:I}) \\
&\quad\propto \frac{1}{\alpha + I - 1}\Bigg(\alpha\psi(\boldsymbol{x}_i \,|\, \boldsymbol{p}_i) f_{\mathrm{G}}(\boldsymbol{p}_i) \\
&\qquad + \sum_{c\in\{c_1,\ldots,c_{i-1},c_{i+1},\ldots,c_I\}} \left(\sum_{i'\in\{1,\ldots,I\}\setminus\{i\}} \mathbb{1}(c_{i'} = c)\right) \psi(\boldsymbol{x}_i \,|\, \boldsymbol{p}^*_c) \delta_{\boldsymbol{p}^*_c}(\boldsymbol{p}_i)\Bigg).
\end{aligned} \tag{3.48}$$

For all $i \in \{1, \ldots, I\}$, the probability of the event $\mathsf{c}_i = c$, with $c \in \{c_1, \ldots, c_{i-1}, c_{i+1}, \ldots, c_I\}$, given $\mathbf{c}_{\neg i} = \boldsymbol{c}_{\neg i}$, $\mathbf{p}^*_{\mathbf{c}_{\neg i}} = \boldsymbol{p}^*_{\boldsymbol{c}_{\neg i}}$, and $\mathbf{x}_{1:I} = \boldsymbol{x}_{1:I}$ is (see (3.39))

$$\begin{aligned}
b_{i,c} &= \mathbb{P}(\mathsf{c}_i = c \,|\, \mathbf{c}_{\neg i} = \boldsymbol{c}_{\neg i}, \mathbf{p}^*_{\mathbf{c}_{\neg i}} = \boldsymbol{p}^*_{\boldsymbol{c}_{\neg i}}, \mathbf{x}_{1:I} = \boldsymbol{x}_{1:I}) \\
&= \mathbb{P}(\mathbf{p}_i = \boldsymbol{p}^*_c \,|\, \mathbf{c}_{\neg i} = \boldsymbol{c}_{\neg i}, \mathbf{p}^*_{\mathbf{c}_{\neg i}} = \boldsymbol{p}^*_{\boldsymbol{c}_{\neg i}}, \mathbf{x}_{1:I} = \boldsymbol{x}_{1:I}) \\
&= \int_{\{\boldsymbol{p}^*_c\}} f_{\mathbf{p}_i|\mathsf{c}_{\neg i},\mathbf{p}^*_{\mathsf{c}_{\neg i}},\mathbf{x}_{1:I}}(\boldsymbol{p}_i \,|\, \boldsymbol{c}_{\neg i}, \boldsymbol{p}^*_{\boldsymbol{c}_{\neg i}}, \boldsymbol{x}_{1:I}) \mathrm{d}\boldsymbol{p}_i,
\end{aligned} \tag{3.49}$$

where we have used that $\mathbf{p}^*_{c'} \neq \mathbf{p}^*_{c''}$ a.s. iff $c' \neq c''$ as $f_{\mathrm{G}}(\boldsymbol{p})$ is assumed continuous. The probability of c_i being distinct from all the other indicator variables, that is, $\mathsf{c}_i \notin \{c_1, \ldots, c_{i-1}, c_{i+1}, \ldots, c_I\}$, given the same random variables as above, is

$$\begin{aligned} b_{i,0} &= \mathbb{P}(\mathsf{c}_i \notin \{c_1, \ldots, c_{i-1}, c_{i+1}, \ldots, c_I\} \,|\, \mathbf{c}_{\neg i} = \boldsymbol{c}_{\neg i}, \mathbf{p}^*_{\mathbf{c}_{\neg i}} = \boldsymbol{p}^*_{\boldsymbol{c}_{\neg i}}, \mathbf{x}_{1:I} = \boldsymbol{x}_{1:I}) \\ &= \mathbb{P}(\mathbf{p}_i \notin \{\boldsymbol{p}^*_{c_1}, \ldots, \boldsymbol{p}^*_{c_{i-1}}, \boldsymbol{p}^*_{c_{i+1}}, \ldots, \boldsymbol{p}^*_{c_I}\} \,|\, \mathbf{c}_{\neg i} = \boldsymbol{c}_{\neg i}, \mathbf{p}^*_{\mathbf{c}_{\neg i}} = \boldsymbol{p}^*_{\boldsymbol{c}_{\neg i}}, \mathbf{x}_{1:I} = \boldsymbol{x}_{1:I}) \\ &= \int_{\mathbb{R}^D \setminus \{\boldsymbol{p}^*_{c_1}, \ldots, \boldsymbol{p}^*_{c_{i-1}}, \boldsymbol{p}^*_{c_{i+1}}, \ldots, \boldsymbol{p}^*_{c_I}\}} f_{\mathbf{p}_i | \mathbf{c}_{\neg i}, \mathbf{p}^*_{\mathbf{c}_{\neg i}}, \mathbf{x}_{1:I}}(\boldsymbol{p}_i \,|\, \boldsymbol{c}_{\neg i}, \boldsymbol{p}^*_{\boldsymbol{c}_{\neg i}}, \boldsymbol{x}_{1:I}) \mathrm{d}\boldsymbol{p}_i. \end{aligned} \tag{3.50}$$

Note that

$$\sum_{c \in \mathcal{C}_{\neg i}} b_{i,c} = 1 \tag{3.51}$$

with $\mathcal{C}_{\neg i} = \{0\} \cup \{c_1, \ldots, c_{i-1}, c_{i+1}, \ldots, c_I\}$. By applying (3.48), we can simplify (3.49) to

$$\begin{aligned} b_{i,c} &\propto \int_{\{\boldsymbol{p}^*_c\}} \frac{1}{\alpha + I - 1} \Bigg(\alpha \psi(\boldsymbol{x}_i \,|\, \boldsymbol{p}_i) f_{\mathrm{G}}(\boldsymbol{p}_i) \\ &\quad + \sum_{c' \in \{c_1, \ldots, c_{i-1}, c_{i+1}, \ldots, c_I\}} \Bigg(\sum_{i' \in \{1, \ldots, I\} \setminus \{i\}} \mathbb{1}(c_{i'} = c') \Bigg) \psi(\boldsymbol{x}_i \,|\, \boldsymbol{p}^*_{c'}) \delta_{\boldsymbol{p}^*_{c'}}(\boldsymbol{p}_i) \Bigg) \mathrm{d}\boldsymbol{p}_i \\ &= \frac{\alpha}{\alpha + I - 1} \int_{\{\boldsymbol{p}^*_c\}} \psi(\boldsymbol{x}_i \,|\, \boldsymbol{p}_i) f_{\mathrm{G}}(\boldsymbol{p}_i) \mathrm{d}\boldsymbol{p}_i \\ &\quad + \frac{1}{\alpha + I - 1} \int_{\{\boldsymbol{p}^*_c\}} \sum_{c' \in \{c_1, \ldots, c_{i-1}, c_{i+1}, \ldots, c_I\}} \Bigg(\sum_{i' \in \{1, \ldots, I\} \setminus \{i\}} \mathbb{1}(c_{i'} = c') \Bigg) \psi(\boldsymbol{x}_i \,|\, \boldsymbol{p}^*_{c'}) \delta_{\boldsymbol{p}^*_{c'}}(\boldsymbol{p}_i) \mathrm{d}\boldsymbol{p}_i \\ &= \frac{1}{\alpha + I - 1} \int_{\{\boldsymbol{p}^*_c\}} \sum_{c' \in \{c_1, \ldots, c_{i-1}, c_{i+1}, \ldots, c_I\}} \Bigg(\sum_{i' \in \{1, \ldots, I\} \setminus \{i\}} \mathbb{1}(c_{i'} = c') \Bigg) \psi(\boldsymbol{x}_i \,|\, \boldsymbol{p}^*_{c'}) \delta_{\boldsymbol{p}^*_{c'}}(\boldsymbol{p}_i) \mathrm{d}\boldsymbol{p}_i. \end{aligned} \tag{3.52}$$

Solving the integral in (3.52), we obtain

$$b_{i,c} = \frac{1}{\alpha + I - 1} \Bigg(\sum_{i' \in \{1, \ldots, I\} \setminus \{i\}} \mathbb{1}(c_{i'} = c) \Bigg) \psi(\boldsymbol{x}_i \,|\, \boldsymbol{p}^*_c), \tag{3.53}$$

where $\mathbb{1}$ is the indicator function and with a proportionality constant such that (3.51) is

true. Similarly, by applying (3.39), we can simplify (3.50) to

$$\begin{aligned}
b_{i,0} &\propto \int_{\mathbb{R}^D \setminus \{\boldsymbol{p}_1,\dots,\boldsymbol{p}_{i-1},\boldsymbol{p}_{i+1},\dots,\boldsymbol{p}_I\}} \frac{1}{\alpha + I - 1} \Bigg(\alpha \psi(\boldsymbol{x}_i \,|\, \boldsymbol{p}_i) f_{\mathrm{G}}(\boldsymbol{p}_i) \\
&\quad + \sum_{c' \in \{c_1,\dots,c_{i-1},c_{i+1},\dots,c_I\}} \Bigg(\sum_{i' \in \{1,\dots,I\} \setminus \{i\}} \mathbb{1}(c_{i'} = c') \Bigg) \psi(\boldsymbol{x}_i \,|\, \boldsymbol{p}^*_{c'}) \delta_{\boldsymbol{p}^*_{c'}}(\boldsymbol{p}_i) \Bigg) \mathrm{d}\boldsymbol{p}_i \\
&= \frac{\alpha}{\alpha + I - 1} \int_{\mathbb{R}^D \setminus \{\boldsymbol{p}_1,\dots,\boldsymbol{p}_{i-1},\boldsymbol{p}_{i+1},\dots,\boldsymbol{p}_I\}} \psi(\boldsymbol{x}_i \,|\, \boldsymbol{p}_i) f_{\mathrm{G}}(\boldsymbol{p}_i) \mathrm{d}\boldsymbol{p}_i \\
&\quad + \frac{1}{\alpha + I - 1} \int_{\mathbb{R}^D \setminus \{\boldsymbol{p}_1,\dots,\boldsymbol{p}_{i-1},\boldsymbol{p}_{i+1},\dots,\boldsymbol{p}_I\}} \sum_{c' \in \{c_1,\dots,c_{i-1},c_{i+1},\dots,c_I\}} \\
&\quad \Bigg(\sum_{i' \in \{1,\dots,I\} \setminus \{i\}} \mathbb{1}(c_{i'} = c') \Bigg) \psi(\boldsymbol{x}_i \,|\, \boldsymbol{p}^*_{c'}) \delta_{\boldsymbol{p}^*_{c'}}(\boldsymbol{p}_i) \mathrm{d}\boldsymbol{p}_i \\
&= \frac{\alpha}{\alpha + I - 1} \int_{\mathbb{R}^D} \psi(\boldsymbol{x}_i \,|\, \boldsymbol{p}_i) f_{\mathrm{G}}(\boldsymbol{p}_i) \mathrm{d}\boldsymbol{p}_i
\end{aligned} \tag{3.54}$$

with the same proportionality constant as in (3.53). Using (3.37), (3.38), we can further simplify (3.54) to

$$b_{i,0} \propto \frac{\alpha}{\alpha + I - 1} \int_{\mathbb{R}^D} f_{\mathbf{x}_i|\mathbf{p}_i}(\boldsymbol{x}_i \,|\, \boldsymbol{p}_i) f_{\mathbf{p}_i}(\boldsymbol{p}_i) \mathrm{d}\boldsymbol{p}_i = \frac{\alpha}{\alpha + I - 1} f_{\mathbf{x}_i}(\boldsymbol{x}_i). \tag{3.55}$$

As we have

$$f_{\mathbf{p}_i|\mathbf{x}_i}(\tilde{\boldsymbol{p}} \,|\, \boldsymbol{x}_i) f_{\mathbf{x}_i}(\boldsymbol{x}_i) = f_{\mathbf{x}_i|\mathbf{p}_i}(\boldsymbol{x}_i \,|\, \tilde{\boldsymbol{p}}) f_{\mathbf{p}_i}(\tilde{\boldsymbol{p}}) \tag{3.56}$$

for any $\tilde{\boldsymbol{p}} \in \mathbb{R}^D$, it follows from (3.55) that

$$b_{i,0} = \frac{\alpha}{\alpha + I - 1} \frac{f_{\mathbf{x}_i|\mathbf{p}_i}(\boldsymbol{x}_i \,|\, \tilde{\boldsymbol{p}}) f_{\mathbf{p}_i}(\tilde{\boldsymbol{p}})}{f_{\mathbf{p}_i|\mathbf{x}_i}(\tilde{\boldsymbol{p}} \,|\, \boldsymbol{x}_i)} = \frac{\alpha}{\alpha + I - 1} \frac{\psi(\boldsymbol{x}_i \,|\, \tilde{\boldsymbol{p}}) f_{\mathrm{G}}(\tilde{\boldsymbol{p}})}{f_{\mathbf{p}_i|\mathbf{x}_i}(\tilde{\boldsymbol{p}} \,|\, \boldsymbol{x}_i)} \tag{3.57}$$

for any $\tilde{\boldsymbol{p}} \in \mathbb{R}^D$ with $f_{\mathbf{p}_i|\mathbf{x}_i}(\tilde{\boldsymbol{p}} \,|\, \boldsymbol{x}_i) > 0$.

The pdf of $\mathbf{p}^*_c$ given $\mathbf{c}_{1:I}$, $\mathbf{p}^*_{\neg c} = (\mathbf{p}^*_{c'})_{c' \in \{\mathbf{c}_1,\dots,\mathbf{c}_I\} \setminus \{c\}}$, and $\mathbf{x}_{1:I}$ is

$$f_{\mathbf{p}^*_c|\mathbf{c}_{1:I},\mathbf{p}^*_{\neg c},\mathbf{x}_{1:I}}(\boldsymbol{p}^*_c \,|\, \boldsymbol{c}_{1:I}, \boldsymbol{p}^*_{\neg c}, \boldsymbol{x}_{1:I}) \propto f_{\mathbf{x}_{1:I}|\mathbf{p}^*_c,\mathbf{c}_{1:I},\mathbf{p}^*_{\neg c}}(\boldsymbol{x}_{1:I} \,|\, \boldsymbol{p}^*_c, \boldsymbol{c}_{1:I}, \boldsymbol{p}^*_{\neg c}) f_{\mathbf{p}^*_c|\mathbf{c}_{1:I},\mathbf{p}^*_{\neg c}}(\boldsymbol{p}^*_c \,|\, \boldsymbol{c}_{1:I}, \boldsymbol{p}^*_{\neg c}), \tag{3.58}$$

for all $c \in \{c_1, \dots, c_I\}$, where we have used Bayes' theorem. Let us simplify both factors in (3.58). The first factor, $f_{\mathbf{x}_{1:I}|\mathbf{p}^*_c,\mathbf{c}_{1:I},\mathbf{p}^*_{\neg c}}(\boldsymbol{x}_{1:I} \,|\, \boldsymbol{p}^*_c, \boldsymbol{c}_{1:I}, \boldsymbol{p}^*_{\neg c})$, can be simplified to

$$\begin{aligned}
f_{\mathbf{x}_{1:I}|\mathbf{p}^*_c,\mathbf{c}_{1:I},\mathbf{p}^*_{\neg c}}(\boldsymbol{x}_{1:I} \,|\, \boldsymbol{p}^*_c, \boldsymbol{c}_{1:I}, \boldsymbol{p}^*_{\neg c}) &= \prod_{i=1}^{I} f_{\mathbf{x}_i|\mathbf{p}^*_c,\mathbf{c}_{1:I},\mathbf{p}^*_{\neg c}}(\boldsymbol{x}_i \,|\, \boldsymbol{p}^*_c, \boldsymbol{c}_{1:I}, \boldsymbol{p}^*_{\neg c}) \\
&\propto \prod_{i:c_i=c} f_{\mathbf{x}_i|\mathbf{p}^*_c,\mathbf{c}_{1:I},\mathbf{p}^*_{\neg c}}(\boldsymbol{x}_i \,|\, \boldsymbol{p}^*_c, \boldsymbol{c}_{1:I}, \boldsymbol{p}^*_{\neg c}) \\
&= \prod_{i:c_i=c} f_{\mathbf{x}_i|\mathbf{p}_i}(\boldsymbol{x}_i \,|\, \boldsymbol{p}^*_c) = \prod_{i:c_i=c} \psi(\boldsymbol{x}_i \,|\, \boldsymbol{p}^*_c),
\end{aligned} \tag{3.59}$$

Algorithm 3.2 Gibbs sampler for DPM with indicator variables

Input: $\boldsymbol{c}_{1:I}^{(k-1)}, c_{\max}^{(k-1)}, \boldsymbol{p}_{c_{1:I}^{(k-1)}}^{*(k-1)}, \boldsymbol{x}_{1:I}$

1: **for all** $i = 1, \ldots, I$ **do**
2: sample $c_i^{(k)} = c$ with probability $b_{i,c}$ for all $c \in \left\{0, c_1^{(k)}, \ldots, c_{i-1}^{(k)}, c_{i+1}^{(k-1)}, \ldots, c_I^{(k-1)}\right\}$ (see (3.53) and (3.57))
3: if $c_i^{(k)} = 0$
set $c_i^{(k)} = \max\left\{\max_{i' \in \{1,\ldots,i-1\}} c_{i'}^{(k)}, c_{\max}^{(k-1)}\right\} + 1$ and
sample $\boldsymbol{p}_{c_i^{(k)}}^{*(k-1)}$ from $f_{\mathbf{p}_c^*|\mathbf{c}_{1:i},\mathbf{c}_{i+1:I},\mathbf{p}_{\neg c}^*,\mathbf{x}_{1:I}}\left(\boldsymbol{p}_c^{*(k-1)} \,\middle|\, \boldsymbol{c}_{1:i}^{(k)}, \boldsymbol{c}_{i+1:I}^{(k-1)}, \boldsymbol{p}_{\neg c}^{*(k-1)}, \boldsymbol{x}_{1:I}\right)$ (see (3.61))
4: **end for**
5: set $c_{\max}^{(k)} = \max\left\{\max_{i \in \{1,\ldots,I\}} c_i^{(k)}, c_{\max}^{(k-1)}\right\}$
6: **for all** $c \in \left\{c_1^{(k)}, \ldots, c_I^{(k)}\right\}$ **do**
7: sample $\boldsymbol{p}_c^{*(k)}$ from $f_{\mathbf{p}_c^*|\mathbf{c}_{1:I},\mathbf{p}_{\neg c}^*,\mathbf{x}_{1:I}}\left(\boldsymbol{p}_c^{*(k)} \,\middle|\, \boldsymbol{c}_{1:I}^{(k)}, \boldsymbol{p}_{\neg c}^{*(k)}, \boldsymbol{x}_{1:I}\right)$ (see (3.61))
8: **end for**

Output: $\boldsymbol{c}_{1:I}^{(k)}, c_{\max}^{(k)}, \boldsymbol{p}_{\boldsymbol{c}_{1:I}^{(k)}}^{*(k)}$

where we have used the independence assumptions defined in at the beginning of this section, (3.38), and that we are interested in (3.58) as a function of $\boldsymbol{p}_c^*$. The second factor, $f_{\mathbf{p}_c^*|\mathbf{c}_{1:I},\mathbf{p}_{\neg c}^*}(\boldsymbol{p}_c^* \,|\, \boldsymbol{c}_{1:I}, \boldsymbol{p}_{\neg c}^*)$, can be simplified to

$$f_{\mathbf{p}_c^*|\mathbf{c}_{1:I},\mathbf{p}_{\neg c}^*}(\boldsymbol{p}_c^* \,|\, \boldsymbol{c}_{1:I}, \boldsymbol{p}_{\neg c}^*) = f_{\mathbf{p}_c^*}(\boldsymbol{p}_c^*) = f_{\mathrm{G}}(\boldsymbol{p}_c^*), \tag{3.60}$$

where we have used that the parameters $\mathbf{p}_c^*$ are i.i.d. (see Definition 3.1). By inserting (3.59) and (3.60) into (3.58), we obtain

$$f_{\mathbf{p}_c^*|\mathbf{c}_{1:I},\mathbf{p}_{\neg c}^*,\mathbf{x}_{1:I}}(\boldsymbol{p}_c^* \,|\, \boldsymbol{c}_{1:I}, \boldsymbol{p}_{\neg c}^*, \boldsymbol{x}_{1:I}) = \left(\prod_{i:c_i=c} \psi(\boldsymbol{x}_i \,|\, \boldsymbol{p}_c^*)\right) f_{\mathrm{G}}(\boldsymbol{p}_c^*). \tag{3.61}$$

The pseudo code of the resulting Gibbs sampler is provided in Algorithm 3.2, where $\boldsymbol{p}_{\neg c}^{*(k)}$ is defined as $\boldsymbol{p}_{\neg c}^*$ above with $\boldsymbol{p}_{c'}^{*(k)}$ instead of $\boldsymbol{p}_{c'}^*$ if it has already been sampled and $\boldsymbol{p}_{c'}^{*(k-1)}$ otherwise. Note that in Algorithm 3.2, if an indicator variable $c_i^{(k)}$ is sampled different from every other indicator variable, that is, $c_i^{(k)} \neq c_{i'}^{(k-1)}$ for all $i' \in \{1, \ldots, I\}$, then it is temporarily set to 0 in Line 2 and assigned a new unique value in Line 3. Furthermore, as $c_i^{(k)}$ is a new class, we also sample a new parameter $\boldsymbol{p}_{c_i^{(k)}}^{*(k-1)}$ according to (3.61) in Line 3.

Just as for the Gibbs sampler without indicator variables presented above, it is beneficial if the base pdf $f_{\mathrm{G}}(\boldsymbol{p})$ is a conjugate prior of $\psi(\boldsymbol{x}|\boldsymbol{p})$ to simplify sampling from (3.61). This is demonstrated next.

Example 3.3: Consider the Gaussian pdfs $f_{\mathrm{G}}(p) = \mathcal{N}(p; \mu, \sigma_1^2)$ and $\psi(x \,|\, p) = \mathcal{N}(x; p, \sigma_2^2)$, where $\mu \in \mathbb{R}$, and $\sigma_1^2, \sigma_2^2 > 0$. It follows from (3.61) that

$$\mathsf{p}_c^* \,|\, (\mathbf{c}_{1:I} = \boldsymbol{c}_{1:I}, \mathsf{p}_{\neg c}^* = p_{\neg c}^*, \mathbf{x}_{1:I} = \boldsymbol{x}_{1:I}) \sim \mathcal{N}(\mu', \sigma_1'^2) \tag{3.62}$$

with

$$\mu' = \frac{1}{\frac{1}{\sigma_1^2} + \frac{\sum_{i=1}^I \mathbb{1}(c_i = c)}{\sigma_2^2}} \left(\frac{\mu}{\sigma_1^2} + \frac{\sum_{i=1}^I \mathbb{1}(c_i = c) x_i}{\sigma_2^2} \right) \tag{3.63}$$

and

$$\sigma_1'^2 = \left(\frac{1}{\sigma_1^2} + \frac{\sum_{i=1}^I \mathbb{1}(c_i = c)}{\sigma_2^2} \right)^{-1}, \tag{3.64}$$

as $f_{\mathrm{G}}(p)$ is a conjugate prior to[3] $\psi(x \,|\, p)$.

Note that in Example (3.3), (3.61) is a Gaussian distribution (3.62) and, therefore, easy to sample.

Expectation of the Posterior Distribution of the DP While there exists a closed form solution for the expectation of the posterior distribution of f_{DP} given the samples $\mathbf{x}_{1:I} = \boldsymbol{x}_{1:I}$ [10, Proposition 5.2], it is of limited use due to its computational complexity. There is however a wide variety of algorithms to approximate it.

One approach is to first use an MC algorithm to generate samples $\boldsymbol{p}_{1:I}^{(k)}$ for $k \in \{1, \ldots, K\}$ that approximate the posterior distribution of $\mathbf{p}_{1:I}$ given $\mathbf{x}_{1:I}$. With the help of (3.20), it can be shown that the DP $\mathsf{f}_{\mathrm{DP}}(\boldsymbol{p})$ given $\mathbf{x}_{1:I}$ can be approximated by $\hat{\mathsf{f}}_{\mathrm{DP},I}^{(K)}(\boldsymbol{p})$, which is distributed according to

$$\hat{\mathsf{f}}_{\mathrm{DP},I}^{(K)} \sim \frac{1}{K} \sum_{k=1}^{K} \mathrm{DP}\Big(\alpha, \tilde{f}_{\mathrm{G},I}^{(k)}\Big) \tag{3.65}$$

with (see 3.21)

$$\tilde{f}_{\mathrm{G},I}^{(k)}(\boldsymbol{p}) = \frac{1}{\alpha + I} \left(\alpha f_{\mathrm{G}}(\boldsymbol{p}) + \sum_{i=1}^{I} \delta_{\boldsymbol{p}_i^{(k)}}(\boldsymbol{p}) \right), \tag{3.66}$$

which is to be interpreted in the sense that $\hat{\mathsf{f}}_{\mathrm{DP},I}^{(K)}(\boldsymbol{p})$ is distributed according to $\mathrm{DP}\Big(\alpha, \tilde{f}_{\mathrm{G},I}^{(k)}\Big)$ with probability $\frac{1}{K}$ for all $k \in \{1, \ldots, K\}$. In other words, the expectation of the posterior distribution of f_{DP} given the samples $\mathbf{x}_{1:I} = \boldsymbol{x}_{1:I}$ can be approximated by $\frac{1}{K} \sum_{k=1}^{K} \mathrm{DP}\Big(\alpha, \tilde{f}_{\mathrm{G},I}^{(k)}\Big)$.

[3] If $\mathsf{p} \sim \mathcal{N}(\mu, \sigma_1^2)$ and $\mathsf{x}_i | (\mathsf{p} = p) \sim \mathcal{N}(p, \sigma_2^2)$ for $i \in \{1, \ldots, I\}$, then $\mathsf{p} | (\mathsf{x}_1 = x_1, \ldots, \mathsf{x}_I = x_I) \sim \mathcal{N}(\mu', \sigma_1'^2)$ with $\mu' = \frac{1}{\frac{1}{\sigma_1^2} + \frac{I}{\sigma_2^2}} \left(\frac{\mu_1}{\sigma_1^2} + \frac{\sum_{i=1}^I x_i}{\sigma_2^2} \right)$ and $\sigma_1'^2 = \left(\frac{1}{\sigma_1^2} + \frac{I}{\sigma_2^2} \right)^{-1}$. In other words, the Gaussian distribution is the conjugate prior for the mean of a Gaussian distribution with known variance.

Clustering As mentioned in the introduction of this chapter, one important application of the DPM is clustering. The objective of clustering is to partition a set of observed data points $\{\boldsymbol{x}_i : i = 1, \ldots, I\}$ into subsets, called classes or clusters, according to some criterion. Equivalently, the solution to a clustering problem is a partition of the index set $\{1, \ldots, I\}$. If we assume that the data points $\boldsymbol{x}_i$ were sampled from a DPM as defined in (3.36)–(3.38), then we can use the partion induced by the indicator variables, that is, the indices i with equal indicator variables c_i are grouped into the same subset. We can choose the most likely partition that is induced by the the indicator variables as our solution to the clustering problem (see CRP in Section 3.1.2). More specifically, the most likely partition is induced by

$$\hat{\boldsymbol{c}}_{\text{joint},1:I} = \text{argmax}_{\boldsymbol{c}_{1:I} \in \mathbb{N}^I} \mathbb{P}(\mathbf{c}_{1:I} = \boldsymbol{c}_{1:I} \,|\, \mathbf{x}_{1:I} = \boldsymbol{x}_{1:I}). \tag{3.67}$$

In the case of an MC algorithm, for example, Algorithm (3.1) or Algorithm (3.2), we can approximate the indicator variables $\mathbf{c}_{1:I}$ that induce the most likely partition according to

$$\hat{\boldsymbol{c}}_{\text{joint},1:I} \approx \text{argmax}_{\boldsymbol{c}_{1:I} \in \mathbb{N}^I} \sum_{k=1}^{K} \mathbb{1}\left(\boldsymbol{c}_{1:I}^{(k)} \sim \boldsymbol{c}_{1:I}\right) \tag{3.68}$$

with the equivalence relation $\sim$, where $\boldsymbol{c}_{1:I}^{(k)} \sim \boldsymbol{c}_{1:I}$ is true iff the induced partition of $\boldsymbol{c}_{1:I}^{(k)}$ is equal to the induced partition of $\boldsymbol{c}_{1:I}$, where the samples $\boldsymbol{c}_{1:I}^{(k)}$ are generated by the MC algorithm. In practice, for large I, we can also use the heuristic approximation

$$\hat{c}_i \approx \text{argmax}_{c \in \mathbb{N}} \sum_{k=1}^{K} \mathbb{1}\left(c_i^{(k)} = c\right) \tag{3.69}$$

for all $i \in \{1, \ldots, I\}$. One important aspect of using the DPM for clustering is that the number of classes C_i (see C_i in Section 3.1.2) is random and does not have to be set in advance. As the DPM is defined as a random mixture with an infinite number of components, closely related to the DP, the number of classes C_I grows with the number of data points I [35] (see Section 3.1.2). If the number of classes is assumed to be random but not growing with the number of data points, then it is recommended to use a mixture of finite mixtures [36], which is closely related to the DPM and allows for similar inference algorithms.

Chapter 4

Inference in Bayesian Nonparametric State-Space Models

In recent years, with the rise of big data, inference based on large data sets to extract useful information has become increasingly important. Often, these data sets represent a large number of time series (TSs). One approach to extract hidden structures is clustering. Clustering of TSs is an active research topic that has been applied to a wide range of different data sets [37]. As mentioned in Section 2.3.5, TSs can often be modeled by a parameter dependent state-space model. In the present chapter, we introduce two algorithms for inference in parameter dependent state-space models, where the parameters are assumed to be distributed according to a Dirichlet process (DP) (see Section 3.1). These statistical models were first introduced in [13] and we will call them Bayesian nonparametric (BNP) state-space models. They combine parameter dependent state-space models with the Dirichlet process mixture (DPM) (see Section 3.2), where the TSs take the role of the random vectors $\mathbf{x}_i$ in Section 3.2. The DPM results in a clustering of the TSs, where the TSs in each group have equal parameter values.

In addition to the clustering, the equal parameter values in each group should also improve the estimation of the parameter of each TS, as we can jointly use the observations of all of the group's TSs instead of using the observations of each TS separately. As the states depend on the parameters, the improved estimation of the parameters should also improve the state estimation. One example where such a statistical model could be applied is extended target tracking of marine vessels using X-band radar [4], as it is likely that a radar system in a harbor should track similar vessels, or the same vessel several times at different points in time. Here, we could model the vessels using a BNP state-space model with the states describing the vessels' kinematic states and the parameters describing their shapes, sizes, and dynamic behaviors. We will further explore extended target tracking

using BNP state-space models in Chapter 5.

To the best of our knowledge, inference and clustering for general BNP State-Space Models has only been considered in [13]. Further, inference and clustering for special cases of related parameter dependent state-space models has been previously considered for, for example, clustering of hidden Markov models using a DP [14–17], linear Gaussian models using a Dirichlet mixture model [18], and models related to linear Gaussian models using a DP [19]. These methods, however, rely on analytical solutions that are not available for state-space models in general.

We present algorithms for TSs of equal length in Section 4.1 and generalize them to TSs of different lengths in Section 4.2. In Sections 4.1.2 and 4.2.2, we present a Particle Markov chain Monte Carlo (PMCMC) algorithm for batch processing based on the particle Gibbs sampler with ancestor sampling (see Section 2.4.5), and in Sections 4.1.3 and 4.2.3, we present a sequential Monte Carlo (SMC) algorithm for sequential processing based on the resample-move particle filter (see Section 2.3.4).

4.1 Time Series of Equal Length

4.1.1 Statistical Model and Inference

Assume that a data set contains $I \in \mathbb{N}$ TSs $(\boldsymbol{y}_{i,1:N})_{i\in\{1,\dots,I\}}$ of equal length $N \in \mathbb{N}$, with $\boldsymbol{y}_{i,n} \in \mathbb{R}^Y$ for all $n \in \{1,\dots,N\}$, that are modeled as observations of a parameter dependent state-space model (see Section 2.3.5), where the prior distribution of the parameter is chosen as a DP (see Section 3.1) [13]. Let us define the parameter dependent state-space model as follows. For each TS indexed by $i \in \mathbb{N}$ — that is, we model an infinite number of TSs — we denote the sequence of states as $(\mathbf{x}_{i,n})_{n\in\mathbb{N}}$ with $\mathbf{x}_{i,n} \in \mathbb{R}^X$ for all $n \in \mathbb{N}$, the sequence of observations as $(\mathbf{y}_{i,n})_{n\in\mathbb{N}}$ with $\mathbf{y}_{i,n} \in \mathbb{R}^Y$ for all $n \in \mathbb{N}$, and the parameter as $\mathbf{p}_i \in \mathbb{R}^D$, with dimensions $X, Y, D \in \mathbb{N}$. The states and observations, $(\mathbf{x}_{i,n}, \mathbf{y}_{i,n})_{n\in\mathbb{N}}$, are modeled as mutually independent for different i given the parameters $(\mathbf{p}_{i'})_{i'\in\mathbb{N}}$ and independent from the parameters $(\mathbf{p}_{i'})_{i'\in\mathbb{N}\setminus\{i\}}$ given $\mathbf{p}_i$. They are, for all $i \in \mathbb{N}$, defined by conditional pdfs $f_{\mathbf{x}_{i,1}|\mathbf{p}_i}(\boldsymbol{x}_{i,1} \,|\, \boldsymbol{p}_i) = \zeta(\boldsymbol{x}_{i,1}|\boldsymbol{p}_i)$ and $f_{\mathbf{x}_{i,n}|\mathbf{x}_{i,n-1},\mathbf{p}_i}(\boldsymbol{x}_{i,n} \,|\, \boldsymbol{x}_{i,n-1}, \boldsymbol{p}_i) = \xi(\boldsymbol{x}_{i,n} \,|\, \boldsymbol{x}_{i,n-1}, \boldsymbol{p}_i)$ for all $n \geq 2$, as well as $f_{\mathbf{y}_{i,n}|\mathbf{x}_{i,n},\mathbf{p}_i}(\boldsymbol{y}_{i,n} \,|\, \boldsymbol{x}_{i,n}, \boldsymbol{p}_i) = \chi(\boldsymbol{y}_{i,n} \,|\, \boldsymbol{x}_{i,n}, \boldsymbol{p}_i)$ for all $n \geq 1$. Furthermore, for parameter dependent state-space models, recall that the state $\mathbf{x}_{i,n}$ is conditionally independent of the states $\mathbf{x}_{i,1:n-2}$ given the previous state $\mathbf{x}_{i,n-1}$ and parameter $\mathbf{p}_i$, and that the observation $\mathbf{y}_{i,n}$ is conditionally independent of the states $\mathbf{x}_{i,1:n-1}$ and observations

$\mathbf{y}_{i,1:n-1}$ given the current state $\mathbf{x}_{i,n}$ and parameter $\mathbf{p}_i$ for all $n \geq 2$, that is, (see (2.33))

$$f_{\mathbf{x}_{i,n}|\mathbf{x}_{i,1:n-1},\mathbf{y}_{i,1:n-1},\mathbf{p}_i}(\boldsymbol{x}_{i,n} \,|\, \boldsymbol{x}_{i,1:n-1}, \boldsymbol{y}_{i,1:n-1}\boldsymbol{p}_i) = \xi(\boldsymbol{x}_{i,n} \,|\, \boldsymbol{x}_{i,n-1}, \boldsymbol{p}_i) \tag{4.1}$$

and (see (2.34))

$$f_{\mathbf{y}_{i,n}|\mathbf{x}_{i,1:n},\mathbf{y}_{i,1:n-1},\mathbf{p}_i}(\boldsymbol{y}_{i,n} \,|\, \boldsymbol{x}_{i,1:n}, \boldsymbol{y}_{i,1:n-1}, \boldsymbol{p}_i) = \chi(\boldsymbol{y}_{i,n} \,|\, \boldsymbol{x}_{i,n}, \boldsymbol{p}_i). \tag{4.2}$$

Note that, with the obvious changes, we can also use densities of other types of random variables instead of $\zeta(\boldsymbol{x}_{i,1}|\boldsymbol{p}_i)$, $\xi(\boldsymbol{x}_{i,n} \,|\, \boldsymbol{x}_{i,n-1}, \boldsymbol{p}_i)$, and $\chi(\boldsymbol{y}_{i,n} \,|\, \boldsymbol{x}_{i,n}, \boldsymbol{p}_i)$.

The parameters $(\mathbf{p}_i)_{i\in\mathbb{N}}$ are distributed according to a DP with concentration α and base pdf $f_{\mathrm{G}}(\boldsymbol{p})$, that is, (see Definition 3.1)

$$\mathsf{f}_{\mathrm{DP}} \sim \mathrm{DP}(\alpha, f_{\mathrm{G}}) \tag{4.3}$$

and (see (3.16))

$$\mathbf{p}_i \,|\, (\mathsf{f}_{\mathrm{DP}} = f_{\mathrm{DP}}) \overset{\text{i.i.d.}}{\sim} f_{\mathrm{DP}}. \tag{4.4}$$

As discussed in Chapter 3, the pdf of $\mathbf{p}_i$ can be expressed as $\mathsf{f}_{\mathbf{p}}(\boldsymbol{p}) = \sum_{c=1}^{\infty} \mathsf{B}_c \delta_{\mathbf{p}_c^*}(\boldsymbol{p})$ (see (3.17)) and therefore (see (3.18))

$$\mathbf{p}_i = \mathbf{p}_{\mathsf{c}_i}^*, \tag{4.5}$$

where $(\mathsf{c}_i)_{i\in\mathbb{N}}$ are indicator variables. Note that this model is a Dirichlet process mixture (DPM) as discussed in Section 3.2, where the states and observations $(\mathbf{x}_{i,1:N}, \mathbf{y}_{i,1:N})$ are equivalent to the random vectors $\mathbf{x}_i$ in Section 3.2.2. Similarly, the pdf $f_{\mathbf{x}_{i,1:N},\mathbf{y}_{i,1:N}|\mathbf{p}_i}(\boldsymbol{x}_{i,1:N}, \boldsymbol{y}_{i,1:N} \,|\, \boldsymbol{p}_i)$ is equivalent to $\psi(\boldsymbol{x}_i \,|\, \boldsymbol{p}_i)$ (see (3.38)). Here, the states $(\mathbf{x}_{i,n})_{n\in\mathbb{N}}$ are latent random vectors and not directly observed.

The goal of the algorithms presented in this chapter is to calculate the expectation

$$J = \mathbb{E}(h(\mathbf{x}_{1:I,1:N}, \mathbf{c}_{1:I}, \mathbf{p}_{\mathbf{c}_{1:I}}^*) \,|\, \mathbf{y}_{1:I,1:N} = \boldsymbol{y}_{1:I,1:N}), \tag{4.6}$$

and in particular

$$J = \mathbb{E}(h(\mathbf{x}_{1:I,1:N}) \,|\, \mathbf{y}_{1:I,1:N} = \boldsymbol{y}_{1:I,1:N}), \tag{4.7}$$

where $h(\boldsymbol{x}_{1:I,1:N}, \boldsymbol{c}_{1:I}, \boldsymbol{p}_{c_{1:I}}^*)$ and $h(\boldsymbol{x}_{1:I,1:N})$ are arbitrary functions, or to infer the most likely class assignment (see (3.67))

$$\hat{\boldsymbol{c}}_{\mathrm{joint},1:I} = \mathrm{argmax}_{\boldsymbol{c}_{1:I}\in\mathbb{N}^I} \mathbb{P}(\mathbf{c}_{1:I} = \boldsymbol{c}_{1:I} \,|\, \mathbf{y}_{1:I,1:N} = \boldsymbol{y}_{1:I,1:N}). \tag{4.8}$$

The algorithms generate $K \in \mathbb{N}$ samples of the states $\boldsymbol{x}_{1:I,1:N}^{(k)}$, indicator variables $\boldsymbol{c}_{1:I}^{(k)}$, and parameters $\boldsymbol{p}_{\boldsymbol{c}_{1:I}^{(k)}}^{*(k)}$ given the observations $\mathbf{y}_{1:I,1:N} = \boldsymbol{y}_{1:I,1:N}$, for all $k \in \{1, \ldots, K\}$. The samples can be used to approximate the expectation (4.6) by (see Section 2.1)

$$J \approx \frac{1}{K} \sum_{k=1}^{K} h\Big(\boldsymbol{x}_{1:I,1:N}^{(k)}, \boldsymbol{c}_{1:I}^{(k)}, \boldsymbol{p}_{\boldsymbol{c}_{1:I}^{(k)}}^{*(k)}\Big), \tag{4.9}$$

the most likely class assignment (4.8) by (see (3.68))

$$\hat{\boldsymbol{c}}_{\text{joint},1:I} \approx \text{argmax}_{\boldsymbol{c}_{1:I} \in \mathbb{N}^I} \sum_{k=1}^{K} \mathbb{1}\Big(\boldsymbol{c}_{1:I}^{(k)} \sim \boldsymbol{c}_{1:I}\Big) \tag{4.10}$$

with the equivalence relation $\sim$, where $\boldsymbol{c}_{1:I}^{(k)} \sim \boldsymbol{c}_{1:I}$ is true iff the induced partition of $\boldsymbol{c}_{1:I}^{(k)}$ is equal to the induced partition of $\boldsymbol{c}_{1:I}$, or by the heuristic approximation (see (3.69))

$$\hat{c}_i \approx \text{argmax}_{c \in \mathbb{N}} \sum_{k=1}^{K} \mathbb{1}\Big(c_i^{(k)} = c\Big) \tag{4.11}$$

for all $i \in \{1, \ldots, I\}$.

4.1.2 Particle Markov Chain Monte Carlo Algorithm

This section presents a PMCMC algorithm that generates samples of the statistical model described in Section 4.1.1. The algorithm uses cycles of Markov chain Monte Carlo (MCMC) kernels (see Section 2.4.3) for the states $\mathbf{x}_{1:I,1:N}$, as well as the indicator variables $\mathbf{c}_{1:I}$ and parameters $\mathbf{p}_{\mathbf{c}_{1:I}}^{*}$. The algorithm then generates the samples $\boldsymbol{x}_{1:I,1:N}^{(k)}$, $\boldsymbol{c}_{1:I}^{(k)}$, and $\boldsymbol{p}_{\boldsymbol{c}_{1:I}^{(k)}}^{*(k)}$ by sampling from transition kernels as discussed in Section 2.4.3. More specifically, we use a particle Gibbs sampler with ancestor sampling (PGAS) (see Section 2.4.6) to sample the states, and Gibbs sampling (see Section 3.2.2) to sample the indicator variables and the parameters. However, in contrast to the PGAS algorithm for inference in parameter dependent state-space models discussed in Section 2.4.6, this algorithm considers multiple TSs and the parameters are distributed according to a DP prior. In comparison, the algorithm introduced in [13] uses cycles of MCMC kernels to obtain indicator variable samples $\boldsymbol{c}_{1:I}^{(k)}$, and parameter samples $\boldsymbol{p}_{\boldsymbol{c}_{1:I}^{(k)}}^{*(k)}$, while the algorithm introduced in this section uses cycles of MCMC kernels to generate state samples $\boldsymbol{x}_{1:I,1:N}^{(k)}$, indicator variable samples $\boldsymbol{c}_{1:I}^{(k)}$, and parameter samples $\boldsymbol{p}_{\boldsymbol{c}_{1:I}^{(k)}}^{*(k)}$, which additionally allows inference on the states $\mathbf{x}_{1:I,1:N}$.

Initialization The algorithm is initialized with state particles $\boldsymbol{x}_{1:I,1:N}^{(1)}$, where each $\boldsymbol{x}_{i,n}^{(1)}$ is chosen arbitrarily from $\mathbb{R}^X$, with indicator variable samples $\boldsymbol{c}_{1:I}^{(1)}$, where $c_i^{(1)} = i$, and with parameter samples $\boldsymbol{p}_{\boldsymbol{c}_{1:I}^{(1)}}^{*(1)}$, where each $\boldsymbol{p}_{c_i^{(1)}}^{*(1)}$ is chosen arbitrarily from $\mathbb{R}^D$.

Sampling the States The PGAS kernel used to sample the states $\mathbf{x}_{i,1:N}$ can be expressed as (see (2.58))

$$t_{\text{PGAS}}\Big(\boldsymbol{x}_{i,1:N}^{(k)} \,\Big|\, \boldsymbol{x}_{i,1:N}^{(k-1)}, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}, \boldsymbol{y}_{i,1:N}\Big) = \sum_{l=1}^{L} W_i^{(k,N,l)} \delta_{\boldsymbol{x}_{i,1:N}^{(k,N,l)}}\Big(\boldsymbol{x}_{i,1:N}^{(k)}\Big), \tag{4.12}$$

where $L \in \mathbb{N}$ and the weights $W_i^{(k,N,l)}$ and particles $\boldsymbol{x}_{i,1:N}^{(k,N,l)}$ are generated as discussed below. That is, we have $\boldsymbol{x}_{i,1:N}^{(k)} = \boldsymbol{x}_{i,1:N}^{(k,N,l)}$ with probability $W_i^{(k,N,l)}$. Note that the kernel used to sample the states $\mathbf{x}_{i,1:N}$ does not depend on the other states and observations, that is, $\mathbf{x}_{i',1:N}$ and $\mathbf{y}_{i',1:N}$ for $i' \in \{1,\ldots,I\} \setminus \{i\}$, as the states and observations are modeled as mutually independent given the parameters $\mathbf{p}_{1:I}$ or equivalently $\mathbf{p}^*_{\mathbf{c}_{1:I}}$ and $\mathbf{c}_{1:I}$ (see Section 4.1.1). For the first time step, $n = 1$, the particles $\boldsymbol{x}_{i,1}^{(k,1,l)}$ are sampled from a proposal pdf $g_1\Big(\boldsymbol{x}_{i,1}^{(k,1,l)} \,\Big|\, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}, \boldsymbol{y}_{i,1}\Big)$ for all $l \in \{1,\ldots L-1\}$, and for $l = L$ we deterministically set $\boldsymbol{x}_{i,1}^{(k,1,L)} = \boldsymbol{x}_{i,1}^{(k-1)}$. Furthermore, for $l \in \{1,\ldots,L\}$, the associated weight $W_i^{(k,1,l)}$ is calculated according to (see (2.59))

$$W_i^{(k,1,l)} \propto \frac{\chi\Big(\boldsymbol{y}_{i,1} \,\Big|\, \boldsymbol{x}_{i,1}^{(k,1,l)}, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}\Big)\zeta\Big(\boldsymbol{x}_{i,1}^{(k,1,l)} \,\Big|\, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}\Big)}{g_1\Big(\boldsymbol{x}_{i,1}^{(k,1,l)} \,\Big|\, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}, \boldsymbol{y}_{i,1}\Big)}, \tag{4.13}$$

with the proportionality constant such that $\sum_{l=1}^{L} W_i^{(k,1,l)} = 1$. For the time steps $n \in \{2,\ldots,N\}$, for all $l \in \{1,\ldots,L-1\}$, we sample $\boldsymbol{x}_{i,1:n-1}^{(k,n,l)}$ such that $\boldsymbol{x}_{i,1:n-1}^{(k,n,l)} = \boldsymbol{x}_{i,1:n-1}^{(k,n-1,l')}$ with probability $W_i^{(k,n-1,l')}$ for all $l' \in \{1,\ldots,L\}$. Then, we set $\boldsymbol{x}_{i,1:n}^{(k,n,l)} = \Big(\boldsymbol{x}_{i,1:n-1}^{(k,n,l)}, \boldsymbol{x}_{i,n}^{(k,n,l)}\Big)$, where the vector $\boldsymbol{x}_{i,n}^{(k,n,l)}$ is sampled from proposal pdf $g\Big(\boldsymbol{x}_{i,n}^{(k,n,l)} \,\Big|\, \boldsymbol{x}_{i,n-1}^{(k,n,l)}, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}, \boldsymbol{y}_{i,n}\Big)$. The vector $\boldsymbol{x}_{i,n}^{(k,n,L)}$ is deterministically set to $\boldsymbol{x}_{i,n}^{(k-1)}$. For all $l \in \{1,\ldots,L\}$, $\boldsymbol{x}_{i,1:n-1}^{(k,n,L)}$ is equal to $\boldsymbol{x}_{i,1:n-1}^{(k,n-1,l)}$ with probability (see (2.60))

$$W_i'^{(k,n-1,l)} \propto W_i^{(k,n-1,l)} \xi\Big(\boldsymbol{x}_{i,n}^{(k-1)} \,\Big|\, \boldsymbol{x}_{i,n-1}^{(k,n-1,l)}, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}\Big), \tag{4.14}$$

where $\sum_{l=1}^{L} W_i'^{(k,n-1,l)} = 1$. As before, we set $\boldsymbol{x}_{i,1:n}^{(k,n,L)} = \Big(\boldsymbol{x}_{i,1:n-1}^{(k,n,L)}, \boldsymbol{x}_{i,n}^{(k,n,L)}\Big)$. Finally, for $l \in \{1,\ldots,L\}$, the weight $W_i^{(k,n,l)}$ is calculated according to (see (2.61))

$$W_i^{(k,n,l)} \propto \frac{\chi\Big(\boldsymbol{y}_{i,n} \,\Big|\, \boldsymbol{x}_{i,n}^{(k,n,l)}, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}\Big)\xi\Big(\boldsymbol{x}_{i,n}^{(k,n,l)} \,\Big|\, \boldsymbol{x}_{i,n-1}^{(k,n-1,l)}, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}\Big)}{g\Big(\boldsymbol{x}_{i,n}^{(k,n,l)} \,\Big|\, \boldsymbol{x}_{i,n-1}^{(k,n-1,l)}, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}, \boldsymbol{y}_{i,n}\Big)} \tag{4.15}$$

with $\sum_{l=1}^{L} W_i^{(k,n,l)} = 1$.

Sampling the Indicator Variables and Parameters As the indicator variables $\mathbf{c}_{1:I}$ and parameters $\mathbf{p}^*_{\mathbf{c}_{1:I}}$ are generated by a DPM, we can sample them by adopting the Gibbs

sampler described in Algorithm 3.2 in Section 3.2. We sample $\mathsf{c}_i = c$ with the conditional probability (see (3.49))

$$b_{i,c}^{(k)} = \mathbb{P}\left(\mathsf{c}_i = c \,\middle|\, \mathbf{c}_{\neg i} = \boldsymbol{c}_{\neg i}^{(k)}, \mathbf{p}^*_{\mathbf{c}_{\neg i}} = \boldsymbol{p}^{*(k-1)}_{\boldsymbol{c}_{\neg i}^{(k)}}, \mathbf{x}_{1:I,1:N} = \boldsymbol{x}_{1:I,1:N}^{(k)}, \mathbf{y}_{1:I,1:N} = \boldsymbol{y}_{1:I,1:N}\right) \tag{4.16}$$

for all $c \in \left\{c_1^{(k)}, \ldots, c_{i-1}^{(k)}, c_{i+1}^{(k-1)}, \ldots, c_I^{(k-1)}\right\}$, where $\mathbf{c}_{\neg i} = (\mathsf{c}_1, \ldots, \mathsf{c}_{i-1}, \mathsf{c}_{i+1}, \ldots, \mathsf{c}_I) = \left(c_1^{(k)}, \ldots, c_{i-1}^{(k)}, c_{i+1}^{(k-1)}, \ldots, c_I^{(k-1)}\right) = \boldsymbol{c}_{\neg i}^{(k)}$ and $\mathbf{p}^*_{\mathbf{c}_{\neg i}} = (\mathbf{p}^*_{\mathsf{c}})_{\mathsf{c}\in\{\mathsf{c}_1,\ldots,\mathsf{c}_{i-1},\mathsf{c}_{i+1},\ldots,\mathsf{c}_I\}} = \left(\boldsymbol{p}_c^{*(k-1)}\right)_{c\in\{c_1^{(k)},\ldots,c_{i-1}^{(k)},c_{i+1}^{(k-1)},\ldots,c_I^{(k-1)}\}} = \boldsymbol{p}^{*(k-1)}_{\boldsymbol{c}_{\neg i}^{(k)}}$, and we sample c_i as being distinct from all the other indicator variables, that is, $\mathsf{c}_i \notin \left\{c_1^{(k)}, \ldots, c_{i-1}^{(k)}, c_{i+1}^{(k-1)}, \ldots, c_I^{(k-1)}\right\}$, with the conditional probability (see (3.55))

$$\begin{aligned} b_{i,0}^{(k)} = \mathbb{P}\Big(\mathsf{c}_i \notin \left\{c_1^{(k)}, \ldots, c_{i-1}^{(k)}, c_{i+1}^{(k-1)}, \ldots, c_I^{(k-1)}\right\} \,\Big|\, \mathbf{c}_{\neg i} = \boldsymbol{c}_{\neg i}^{(k)}, \mathbf{p}^*_{\mathbf{c}_{\neg i}} = \boldsymbol{p}^{*(k-1)}_{\boldsymbol{c}_{\neg i}^{(k)}}, \mathbf{x}_{1:I,1:N} = \boldsymbol{x}_{1:I,1:N}^{(k)}, \\ \mathbf{y}_{1:I,1:N} = \boldsymbol{y}_{1:I,1:N}\Big). \end{aligned} \tag{4.17}$$

The samples of the indicator variables $c_i^{(k)}$ that are distinct from the other indicator variables need to be assigned a new value, that is, $c_i^{(k)} = \max\left\{\max_{i'\in\{1,\ldots,i-1\}} c_{i'}^{(k)}, c_{\max}^{(k-1)}\right\} + 1$ with $c_{\max}^{(k-1)} = \max_{k'\in\{1,\ldots,k-1\}, i'\in\{1,\ldots,I\}} c_{i'}^{(k')}$. Furthermore, a parameter $\boldsymbol{p}^{*(k-1)}_{c_i^{(k)}}$ has to be sampled, which will be discussed presently. Before we further simplify (4.16) and (4.17), note that

$$\begin{aligned} &f_{\mathbf{x}_{i,1:n},\mathbf{y}_{i,1:n}|\mathbf{p}_i}\left(\boldsymbol{x}_{i,1:n}, \boldsymbol{y}_{i,1:n} \,\middle|\, \boldsymbol{p}\right) \\ &\quad= f_{\mathbf{y}_{i,1:n}|\mathbf{x}_{i,1:n},\mathbf{p}_i}\left(\boldsymbol{y}_{i,1:n} \,\middle|\, \boldsymbol{x}_{i,1:n}, \boldsymbol{p}\right) f_{\mathbf{x}_{i,1:n}|\mathbf{p}_i}\left(\boldsymbol{x}_{i,1:n} \,\middle|\, \boldsymbol{p}\right) \\ &\quad= \left(\prod_{n'=1}^{n} f_{\mathbf{y}_{i,n'}|\mathbf{x}_{i,n'},\mathbf{p}_i}\left(\boldsymbol{y}_{i,n'} \,\middle|\, \boldsymbol{x}_{i,n'}, \boldsymbol{p}\right)\right) f_{\mathbf{x}_{i,1}|\mathbf{p}_i}\left(\boldsymbol{x}_{i,1} \,\middle|\, \boldsymbol{p}\right) \prod_{n''=2}^{n} f_{\mathbf{x}_{i,n''}|\mathbf{x}_{i,n''-1},\mathbf{p}_i}\left(\boldsymbol{x}_{i,n''} \,\middle|\, \boldsymbol{x}_{i,n''-1}, \boldsymbol{p}\right) \\ &\quad= \left(\prod_{n'=1}^{n} \chi\left(\boldsymbol{y}_{i,n'} \,\middle|\, \boldsymbol{x}_{i,n'}, \boldsymbol{p}\right)\right) \zeta\left(\boldsymbol{x}_{i,1} \,\middle|\, \boldsymbol{p}\right) \prod_{n''=2}^{n} \xi\left(\boldsymbol{x}_{i,n''} \,\middle|\, \boldsymbol{x}_{i,n''-1}, \boldsymbol{p}\right). \end{aligned} \tag{4.18}$$

Using the same derivation as for (3.53) with $\mathbf{x}_i$ replaced by $(\mathbf{x}_{i,1:N}, \mathbf{y}_{i,1:N})$, and further (4.18), we can simplify (4.16) to

$$\begin{aligned} b_{i,c}^{(k)} &\propto \frac{1}{\alpha + I - 1}\left(\sum_{i'=1}^{i-1} \mathbb{1}\left(c_{i'}^{(k)} = c\right) + \sum_{i'=i+1}^{I} \mathbb{1}\left(c_{i'}^{(k-1)} = c\right)\right) f_{\mathbf{x}_{i,1:N},\mathbf{y}_{i,1:N}|\mathbf{p}_i}\left(\boldsymbol{x}_{i,1:N}^{(k)}, \boldsymbol{y}_{i,1:N} \,\middle|\, \boldsymbol{p}_c^{*(k-1)}\right) \\ &= \frac{1}{\alpha + I - 1}\left(\sum_{i'=1}^{i-1} \mathbb{1}\left(c_{i'}^{(k)} = c\right) + \sum_{i'=i+1}^{I} \mathbb{1}\left(c_{i'}^{(k-1)} = c\right)\right)\left(\prod_{n=1}^{N} \chi\left(\boldsymbol{y}_{i,n} \,\middle|\, \boldsymbol{x}_{i,n}^{(k)}, \boldsymbol{p}_c^{*(k-1)}\right)\right) \\ &\quad\times \zeta\left(\boldsymbol{x}_{i,1}^{(k)} \,\middle|\, \boldsymbol{p}_c^{*(k)}\right) \prod_{n'=2}^{N} \xi\left(\boldsymbol{x}_{i,n'}^{(k)} \,\middle|\, \boldsymbol{x}_{i,n'-1}^{(k)}, \boldsymbol{p}_c^{*(k)}\right). \end{aligned} \tag{4.19}$$

Similarly, by using the same derivation as for (3.57) with $\mathbf{x}_i$ replaced by $(\mathbf{x}_{i,1:N}, \mathbf{y}_{i,1:N})$, and furthermore using (4.18), we can simplify (4.17) to

$$\begin{aligned} b_{i,0}^{(k)} &\propto \frac{\alpha}{\alpha+I-1} \frac{f_{\mathbf{x}_{i,1:N},\mathbf{y}_{i,1:N}|\mathbf{p}_i}\left(\boldsymbol{x}_{i,1:N}^{(k)}, \boldsymbol{y}_{i,1:N} \,\middle|\, \tilde{\boldsymbol{p}}\right) f_{\mathbf{p}}(\tilde{\boldsymbol{p}})}{f_{\mathbf{p}_i|\mathbf{x}_{i,1:N},\mathbf{y}_{i,1:N}}\left(\tilde{\boldsymbol{p}} \,\middle|\, \boldsymbol{x}_{i,1:N}^{(k)}, \boldsymbol{y}_{i,1:N}\right)} \\ &= \frac{\alpha}{\alpha+I-1} \frac{f_{\mathbf{p}}(\tilde{\boldsymbol{p}})}{f_{\mathbf{p}_i|\mathbf{x}_{i,1:N},\mathbf{y}_{i,1:N}}\left(\tilde{\boldsymbol{p}} \,\middle|\, \boldsymbol{x}_{i,1:N}^{(k)}, \boldsymbol{y}_{i,1:N}\right)} \left(\prod_{n=1}^{N} \chi\left(\boldsymbol{y}_{i,n} \,\middle|\, \boldsymbol{x}_{i,n}^{(k)}, \tilde{\boldsymbol{p}}\right)\right) \\ &\quad \times \zeta\left(\boldsymbol{x}_{i,1}^{(k)} \,\middle|\, \tilde{\boldsymbol{p}}\right) \prod_{n'=2}^{N} \xi\left(\boldsymbol{x}_{i,n'}^{(k)} \,\middle|\, \boldsymbol{x}_{i,n'-1}^{(k)}, \tilde{\boldsymbol{p}}\right), \end{aligned} \tag{4.20}$$

with the same proportionality constant as in (4.18), and an arbitrary $\tilde{\boldsymbol{p}} \in \mathbb{R}^D$ with $f_{\mathbf{p}}(\tilde{\boldsymbol{p}}) \neq 0$. Note that we assume here that we are able to calculate $f_{\mathbf{p}_i|\mathbf{x}_{i,1:N},\mathbf{y}_{i,1:N}}\left(\tilde{\boldsymbol{p}} \,\middle|\, \boldsymbol{x}_{i,1:N}^{(k)}, \boldsymbol{y}_{i,1:N}\right)$.

The parameter samples $\boldsymbol{p}_c^{*(k)}$ for $c \in \{c_1^{(k)}, \ldots, c_I^{(k)}\}$ are sampled from the conditional pdf of $\mathbf{p}_c^*$ given every other random variable (see (3.58))

$$f_{\mathbf{p}_c^*|\mathbf{x}_{1:I,1:N},\mathbf{c}_{1:I},\mathbf{p}_{\neg c}^*,\mathbf{y}_{1:I,1:N}}\left(\boldsymbol{p}_c^{*(k)} \,\middle|\, \boldsymbol{x}_{1:I,1:N}^{(k)}, \boldsymbol{c}_{1:I}^{(k)}, \boldsymbol{p}_{\neg c}^{*(k)}, \boldsymbol{y}_{1:I,1:N}\right), \tag{4.21}$$

with $\mathbf{p}_{\neg c}^* = (\mathbf{p}_{c'}^*)_{c' \in \{\mathbf{c}_1,\ldots,\mathbf{c}_I\}\setminus\{c\}}$ and $\boldsymbol{p}_{\neg c}^{*(k)} = \left(\boldsymbol{p}_{c'}^{*(k)'}\right)_{c' \in \{c_1^{(k)},\ldots,c_I^{(k)}\}\setminus\{c\}}$, where $\boldsymbol{p}_{c'}^{*(k)'}$ is the newest sample of $\mathbf{p}_{c'}^*$, that is, either $\boldsymbol{p}_{c'}^{*(k-1)}$ or $\boldsymbol{p}_{c'}^{*(k)}$. Note that the parameter $\boldsymbol{p}_{c_i^{(k)}}^{*(k-1)}$ for a newly generated class, as discussed above, is also sampled from (4.21), but with the newest currently available samples of the other random variables. Using Bayes' theorem, that the states and observations, $(\mathbf{x}_{i,1:N}, \mathbf{y}_{i,1:N})$, are mutually independent for different i given $\mathbf{p}_{1:I}$ or equivalently $\mathbf{p}_{\mathbf{c}_{1:I}}^*$ and $\mathbf{c}_{1:I}$, and that the parameters $\mathbf{p}_c^*$ are i.i.d. and independent of $\mathbf{c}_{1:I}$, we can simplify (4.21) to

$$\begin{aligned} &f_{\mathbf{p}_c^*|\mathbf{x}_{1:I,1:N},\mathbf{c}_{1:I},\mathbf{p}_{\neg c}^*,\mathbf{y}_{1:I,1:N}}\left(\boldsymbol{p}_c^{*(k)} \,\middle|\, \boldsymbol{x}_{1:I,1:N}^{(k)}, \boldsymbol{c}_{1:I}^{(k)}, \boldsymbol{p}_{\neg c}^{*(k)}, \boldsymbol{y}_{1:I,1:N}\right) \\ &\quad \propto f_{\mathbf{x}_{1:I,1:N},\mathbf{y}_{1:I,1:N}|\mathbf{p}_c^*,\mathbf{c}_{1:I},\mathbf{p}_{\neg c}^*}\left(\boldsymbol{x}_{1:I,1:N}^{(k)}, \boldsymbol{y}_{1:I,1:N} \,\middle|\, \boldsymbol{p}_c^{*(k)}, \boldsymbol{c}_{1:I}^{(k)}, \boldsymbol{p}_{\neg c}^{*(k)}\right) f_{\mathbf{p}_c^*|\mathbf{c}_{1:I},\mathbf{p}_{\neg c}^*}\left(\boldsymbol{p}_c^{*(k)} \,\middle|\, \boldsymbol{c}_{1:I}^{(k)}, \boldsymbol{p}_{\neg c}^{*(k)}\right) \\ &\quad = \left(\prod_{i=1}^{I} f_{\mathbf{x}_{i,1:N},\mathbf{y}_{i,1:N}|\mathbf{p}_c^*,\mathbf{c}_{1:I},\mathbf{p}_{\neg c}^*}\left(\boldsymbol{x}_{i,1:N}^{(k)}, \boldsymbol{y}_{i,1:N} \,\middle|\, \boldsymbol{p}_c^{*(k)}, \boldsymbol{c}_{1:I}^{(k)}, \boldsymbol{p}_{\neg c}^{*(k)}\right)\right) f_{\mathbf{p}_c^*|\mathbf{c}_{1:I},\mathbf{p}_{\neg c}^*}\left(\boldsymbol{p}_c^{*(k)} \,\middle|\, \boldsymbol{c}_{1:I}^{(k)}, \boldsymbol{p}_{\neg c}^{*(k)}\right) \\ &\quad = \left(\prod_{i=1}^{I} f_{\mathbf{x}_{i,1:N},\mathbf{y}_{i,1:N}|\mathbf{p}_c^*,\mathbf{c}_{1:I},\mathbf{p}_{\neg c}^*}\left(\boldsymbol{x}_{i,1:N}^{(k)}, \boldsymbol{y}_{i,1:N} \,\middle|\, \boldsymbol{p}_c^{*(k)}, \boldsymbol{c}_{1:I}^{(k)}, \boldsymbol{p}_{\neg c}^{*(k)}\right)\right) f_{\mathbf{p}_c^*}\left(\boldsymbol{p}_c^{*(k)}\right). \end{aligned} \tag{4.22}$$

As $\mathbf{p}_{\mathbf{c}_{1:I}}^*$ and $\mathbf{c}_{1:I}$ is equivalent to $\mathbf{p}_{1:I}$, the states and observations, $(\mathbf{x}_{i,1:N}, \mathbf{y}_{i,1:N})$, are

Algorithm 4.1 PMCMC algorithm for a BNP state-space model

Input: $\boldsymbol{x}_{1:I,1:N}^{(k-1)}, \boldsymbol{c}_{1:I}^{(k-1)}, c_{\max}^{(k-1)}, \boldsymbol{p}_{\boldsymbol{c}_{1:I}^{(k-1)}}^{*(k-1)}, \boldsymbol{y}_{1:I,1:N}, L, g_1, g$

1: **for all** $i = 1, \ldots, I$ **do**
2: sample $\boldsymbol{x}_{i,1:N}^{(k)}$ using Algorithm 4.2 with input $\boldsymbol{x}_{i,1:N}^{(k-1)}, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}, \boldsymbol{y}_{i,1:N}, L, g_1, g$
3: **end for**
4: sample $\boldsymbol{c}_{1:I}^{(k)}$, $c_{\max}^{(k)}$, and $\boldsymbol{p}_{\boldsymbol{c}_{1:I}^{(k)}}^{*(k)}$ using Algorithm 4.3 with input $\boldsymbol{x}_{1:I,1:N}^{(k)}, \boldsymbol{c}_{1:I}^{(k-1)}, c_{\max}^{(k-1)}, \boldsymbol{p}_{\boldsymbol{c}_{1:I}^{(k)}}^{*(k-1)}, \boldsymbol{y}_{1:I,1:N}$

Output: $\boldsymbol{x}_{1:I,1:N}^{(k)}, \boldsymbol{c}_{1:I}^{(k)}, c_{\max}^{(k)}, \boldsymbol{p}_{\boldsymbol{c}_{1:I}^{(k)}}^{*(k)}$

independent of $\mathbf{p}_{\neg i}$ given $\mathbf{p}_i$, and $f_{\mathbf{p}_c^*}\left(\boldsymbol{p}_c^{*(k)}\right) = f_{\mathrm{G}}\left(\boldsymbol{p}_c^{*(k)}\right)$, we can further simplify (4.22) to

$$\begin{aligned}
&f_{\mathbf{p}_c^*|\mathbf{x}_{1:I,1:N},\mathbf{c}_{1:I},\mathbf{p}_{\neg c}^*,\mathbf{y}_{1:I,1:N}}\left(\boldsymbol{p}_c^{*(k)} \,\middle|\, \boldsymbol{x}_{1:I,1:N}^{(k)}, \boldsymbol{c}_{1:I}^{(k)}, \boldsymbol{p}_{\neg c}^{*(k)}, \boldsymbol{y}_{1:I,1:N}\right) \\
&\quad \propto \left(\prod_{i=1}^{I} f_{\mathbf{x}_{i,1:N},\mathbf{y}_{i,1:N}|\mathbf{p}_i}\left(\boldsymbol{x}_{i,1:N}^{(k)}, \boldsymbol{y}_{i,1:N} \,\middle|\, \boldsymbol{p}_{c_i}^{*(k)}\right)\right) f_{\mathbf{p}_c^*}\left(\boldsymbol{p}_c^{*(k)}\right) \\
&\quad \propto \left(\prod_{i:\boldsymbol{c}_i^{(k)}=c} f_{\mathbf{x}_{i,1:N},\mathbf{y}_{i,1:N}|\mathbf{p}_i}\left(\boldsymbol{x}_{i,1:N}^{(k)}, \boldsymbol{y}_{i,1:N} \,\middle|\, \boldsymbol{p}_{c_i}^{*(k)}\right)\right) f_{\mathbf{p}_c^*}\left(\boldsymbol{p}_c^{*(k)}\right) \\
&\quad \propto \left(\prod_{i:\boldsymbol{c}_i^{(k)}=c} f_{\mathbf{x}_{i,1:N},\mathbf{y}_{i,1:N}|\mathbf{p}_i}\left(\boldsymbol{x}_{i,1:N}^{(k)}, \boldsymbol{y}_{i,1:N} \,\middle|\, \boldsymbol{p}_{c_i}^{*(k)}\right)\right) f_{\mathrm{G}}\left(\boldsymbol{p}_c^{*(k)}\right). \qquad (4.23)
\end{aligned}$$

That is, in order to use this algorithm, we need to be able to sample from (4.23).

Summary The pseudo-code for one iteration of the complete algorithm is stated in Algorithm 4.1. It first samples the states using Algorithm 4.2 and then the indicator variables and parameters using Algorithm 4.3.

4.1.3 Sequential Monte Carlo Algorithm

4.1.3.1 General Principle

This section presents an SMC algorithm to sequentially generate K samples, or particles, of the states $\mathbf{x}_{1:I,1:n}$, indicator variables $\mathbf{c}_{1:I}$, and parameters $\mathbf{p}_{\mathbf{c}_{1:I}}^*$ for all $n \in \mathbb{N}$. This algorithm can be seen as a combination of the resample-move particle filter (RMPF) (see Section 2.3.4) and the algorithm presented in Section 4.1.2. At each time step $n \in \mathbb{N}$, the algorithm first extends the state particles of the previous time step $\left(\boldsymbol{x}_{1:I,1:n-1}^{(n-1,k)}\right)_{k\in\{1,\ldots,K\}}$ to intermediate state particles of the current time step $\left(\bar{\boldsymbol{x}}_{1:I,1:n}^{(n,k)}\right)_{k\in\{1,\ldots,K\}}$. These intermediate particles are then used within an MCMC algorithm to jointly generate the state

Algorithm 4.2 PMCMC algorithm for a BNP state-space model: states cycle

Input: $\boldsymbol{x}_{i,1:N}^{(k-1)}, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}, \boldsymbol{y}_{i,1:N}, L, g_1, g$

for all $l = 1, \ldots, L-1$ **do**
  sample $\boldsymbol{x}_{i,1}^{(k,1,l)}$ from $g_1\left(\boldsymbol{x}_{i,1}^{(k,1,l)} \middle| \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}, \boldsymbol{y}_{i,1}\right)$
end for
set $\boldsymbol{x}_{i,1}^{(k,1,L)} = \boldsymbol{x}_{i,1}^{(k-1)}$
for all $l = 1, \ldots, L$ **do**
  calculate $W_i^{(k,1,l)}$ according to (4.13)
end for
for all $n = 2, \ldots, N$ **do**
  for all $l = 1, \ldots, L-1$ **do**
    sample $\boldsymbol{x}_{i,1:n-1}^{(k,n,l)} = \boldsymbol{x}_{i,1:n-1}^{(k,n-1,l')}$ with probability $W_i^{(k,n-1,l')}$ for all $l' \in \{1, \ldots, L\}$
    sample $\boldsymbol{x}_{i,n}^{(k,n,l)}$ from $g\left(\boldsymbol{x}_{i,n}^{(k,n,l)} \middle| \boldsymbol{x}_{i,1:n-1}^{(k,n,l)}, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}, \boldsymbol{y}_{i,n}\right)$
    set $\boldsymbol{x}_{i,1:n}^{(k,n,l)} = \left(\boldsymbol{x}_{i,1:n-1}^{(k,n,l)}, \boldsymbol{x}_{i,n}^{(k,n,l)}\right)$
  end for
  for all $l = 1, \ldots, L$ **do**
    calculate $W_i'^{(k,n-1,l)}$ according to (4.14)
  end for
  sample $\boldsymbol{x}_{i,1:n-1}^{(k,n,L)} = \boldsymbol{x}_{i,1:n-1}^{(k,n-1,l')}$ with probability $W_i'^{(k,n-1,l')}$ for all $l' \in \{1, \ldots, L\}$
  set $\boldsymbol{x}_{i,1:n}^{(k,n,L)} = \left(\boldsymbol{x}_{i,1:n-1}^{(k,n,L)}, \boldsymbol{x}_{i,n}^{(k-1)}\right)$
  for all $l = 1, \ldots, L$ **do**
    calculate $W_i^{(k,n,l)}$ according to (4.15)
  end for
end for
sample $\boldsymbol{x}_{i,1:N}^{(k)}$ from (4.12)

Output: $\boldsymbol{x}_{i,1:N}^{(k)}$

Algorithm 4.3 PMCMC algorithm for a BNP state-space model: indicator variables and parameters cycle

Input: $\boldsymbol{x}_{1:I,1:N}^{(k)}, \boldsymbol{c}_{1:I}^{(k-1)}, c_{\max}^{(k-1)}, \boldsymbol{p}_{\boldsymbol{c}_{1:I}^{(k-1)}}^{*(k-1)}, \boldsymbol{y}_{1:I,1:N}$

1: **for all** $i = 1, \ldots, I$ **do**
2: sample $c_i^{(k)} = c$ with probability $b_{i,c}^{(k)}$ (see (4.19) and (4.20))
for all $c \in \left\{0, c_1^{(k)}, \ldots, c_{i-1}^{(k)}, c_{i+1}^{(k-1)}, \ldots, c_I^{(k-1)}\right\}$
3: if $c_i^{(k)} = 0$
set $c_i^{(k)} = \max\left\{\max_{i' \in \{1,\ldots,i-1\}} c_{i'}^{(k)}, c_{\max}^{(k-1)}\right\} + 1$
sample $\boldsymbol{p}_{c_i^{(k)}}^{*(k-1)}$ from (4.23)
4: **end for**
5: set $c_{\max}^{(k)} = \max\left\{\max_{i \in \{1,\ldots,I\}} c_i^{(k)}, c_{\max}^{(k-1)}\right\}$
6: **for all** $c \in \left\{c_1^{(k)}, \ldots, c_I^{(k)}\right\}$ **do**
7: sample $\boldsymbol{p}_c^{*(k)}$ from (4.23)
8: **end for**

Output: $\boldsymbol{c}_{1:I}^{(k)}, c_{\max}^{(k)}, \boldsymbol{p}_{\boldsymbol{c}_{1:I}^{(k)}}^{*(k)}$

particles $\left(\boldsymbol{x}_{1:I,1:n}^{(n,k)}\right)_{k \in \{1,\ldots,K\}}$, the indicator variable samples $\left(\boldsymbol{c}_{1:I}^{(n,k)}\right)_{k \in \{1,\ldots,K\}}$, and the parameter samples $\left(\boldsymbol{p}_{\boldsymbol{c}_{1:I}^{(n,k)}}^{*(n,k)}\right)_{k \in \{1,\ldots,K\}}$. Note that we have added an additional time index to the indicator variable and parameter samples as they are sampled at each time step. In contrast, the RMPF, at each time step, first resamples the particles, extends them, and then uses a transition kernel to resample the parameters of each TS separately. This modification to jointly sample the states, indicator variables, and parameters using the MCMC algorithm is necessary because, due to the DP, the TSs are not independent. Another related algorithm is sequential Markov chain Monte Carlo [38, Section 2.2] [27–29], where random vectors $\mathbf{x}_{1:n}$ are sampled sequentially at each time $n \in \mathbb{N}$ using an MCMC algorithm. There, the MCMC algorithm is used to sample $\mathbf{x}_{1:n}$ from the approximated pdf $f_{\mathbf{x}_{1:n}}(\boldsymbol{x}_{1:n}) \approx f_{\mathbf{x}_n|\mathbf{x}_{1:n-1}}(\boldsymbol{x}_n \,|\, \boldsymbol{x}_{1:n-1}) \hat{\mathsf{f}}_{\mathbf{x}_{1:n-1}}(\boldsymbol{x}_{1:n-1})$, where $\hat{\mathsf{f}}_{\mathbf{x}_{1:n-1}}(\boldsymbol{x}_{1:n-1})$ is the MC approximation of $f_{\mathbf{x}_{1:n-1}}(\boldsymbol{x}_{1:n-1})$ built from the samples of the time step $n-1$. The main difference of our approach from sequential Markov chain Monte Carlo is that we first extend our state particles and then resample them within our MCMC step.

Intermediate State Particles To initialize the algorithm at the first time step, we start by setting $c_i^{(0,k)} = i$ and sampling $\boldsymbol{p}_{c_i^{(0,k)}}^{*(0,k)}$ from the base pdf $f_{\mathrm{G}}\left(\boldsymbol{p}_{c_i^{(0,k)}}^{*(0,k)}\right)$ for all TSs $i \in \{1, \ldots, I\}$. That is, we initialize the intermediate state particles with a separate class for each TS. The intermediate state particles $\left(\bar{\boldsymbol{x}}_{1:I}^{(1,k)}\right)_{k \in \{1,\ldots,K\}}$ are initialized by sam-

pling $\bar{\boldsymbol{x}}_i^{(1,k)}$ from a proposal pdf $g_1\Big(\bar{\boldsymbol{x}}_i^{(1,k)} \,\Big|\, \boldsymbol{p}^{*(0,k)}_{c_i^{(0,k)}}, \boldsymbol{y}_{i,1}\Big)$. These intermediate state particles are then later resampled using the MCMC algorithm to obtain the final state particles $\Big(\boldsymbol{x}_{1:I}^{(1,k)}\Big)_{k\in\{1,\dots,K\}}$ for the first time step.

For the time steps $n \geq 2$, the algorithm updates the state particles of the previous time step $\Big(\boldsymbol{x}_{1:I,1:n-1}^{(n-1,k)}\Big)_{k\in\{1,\dots,K\}}$ to intermediate state particles of the current time step $\Big(\bar{\boldsymbol{x}}_{1:I,1:n}^{(n,k)}\Big)_{k\in\{1,\dots,K\}}$. More specifically, we sample $\bar{\boldsymbol{x}}_{i,n}^{(n,k)}$ from a proposal pdf $g\Big(\bar{\boldsymbol{x}}_{i,n}^{(n,k)} \,\Big|\, \boldsymbol{x}_{i,n-1}^{(n-1,k)}, \boldsymbol{p}^{*(n-1,k)}_{c_i^{(n-1,k)}}, \boldsymbol{y}_{i,n}\Big)$ and then set $\bar{\boldsymbol{x}}_{i,1:n}^{(n,k)} = \Big(\boldsymbol{x}_{i,1:n-1}^{(n-1,k)}, \bar{\boldsymbol{x}}_{i,n}^{(n,k)}\Big)$ for each TS. As in the first time step, these intermediate state particles are later resampled using an MCMC algorithm to obtain the final state particles $\Big(\boldsymbol{x}_{1:I,1:n}^{(n,k)}\Big)_{k\in\{1,\dots,K\}}$.

Initialization of the MCMC Algorithm The MCMC algorithm is first initialized with state particles $\Big(\boldsymbol{x}_{1:I,1:n}^{(n,1)}\Big)_{k\in\{1,\dots,K\}}$, where $\boldsymbol{x}_{i,1:n}^{(n,1)}$ is chosen arbitrarily from $\Big\{\bar{\boldsymbol{x}}_{i,1:n}^{(n,k)} : k \in \{1,\dots,K\}\Big\}$, with indicator variables samples $\boldsymbol{c}_{1:I}^{(n,1)}$, where $c_i^{(n,1)} = i$, and with parameter samples $\boldsymbol{p}^{*(n,1)}_{\boldsymbol{c}_{1:I}^{(n,1)}}$, where $\boldsymbol{p}^{*(n,1)}_{c_i^{(n,1)}}$ is chosen arbitrarily from $\mathbb{R}^D$.

Sampling the States At each iteration of the MCMC algorithm, the states are sampled from a transition kernel based on an approximation of the true conditional distribution of the states that is developed in the following. We start by noting that the MCMC algorithm should ideally use a transition kernel to generate samples $\boldsymbol{x}_{1:I,1:n}^{(n,k)}$ of the states $\mathbf{x}_{1:I,1:n}$ given the indicator variables $\mathbf{c}_{1:I}$, parameters $\mathbf{p}^*_{\mathbf{c}_{1:I}}$, and observations $\mathbf{y}_{1:I,1:n}$, that is, a transition kernel with target pdf

$$\begin{aligned} &f_{\mathbf{x}_{1:I,1:n}|\mathbf{c}_{1:I},\mathbf{p}^*_{\mathbf{c}_{1:I}},\mathbf{y}_{1:I,1:n}}\Big(\boldsymbol{x}_{1:I,1:n} \,\Big|\, \boldsymbol{c}_{1:I}^{(n,k-1)}, \boldsymbol{p}^{*(n,k-1)}_{\boldsymbol{c}_{1:I}^{(n,k-1)}}, \boldsymbol{y}_{1:I,1:n}\Big) \\ &\quad= \prod_{i=1}^{I} f_{\mathbf{x}_{i,1:n}|\mathbf{c}_{1:I},\mathbf{p}^*_{\mathbf{c}_{1:I}},\mathbf{y}_{1:I,1:n}}\Big(\boldsymbol{x}_{i,1:n} \,\Big|\, \boldsymbol{c}_{1:I}^{(n,k-1)}, \boldsymbol{p}^{*(n,k-1)}_{\boldsymbol{c}_{1:I}^{(n,k-1)}}, \boldsymbol{y}_{1:I,1:n}\Big) \\ &\quad= \prod_{i=1}^{I} f_{\mathbf{x}_{i,1:n}|\mathbf{p}_i,\mathbf{y}_{i,1:n}}\Big(\boldsymbol{x}_{i,1:n} \,\Big|\, \boldsymbol{p}^{*(n,k-1)}_{\boldsymbol{c}_i^{(n,k-1)}}, \boldsymbol{y}_{i,1:n}\Big), \end{aligned} \tag{4.24}$$

where we have used the conditional independence of the TSs, and (4.5) as well as that the states and observervations, $(\mathbf{x}_{i,1:N}, \mathbf{y}_{i,1:N})$, are independent of $\mathbf{p}_{\neg i}$ given $\mathbf{p}_i$. Directly sampling from the pdf (4.24) at time step n would require us to sample the n states $\mathbf{x}_{i,1:n}$ for each $i \in \{1,\dots,I\}$. That is, the computational complexity of sampling from the pdf (4.24) grows with n. Therefore, we propose to approximate each $f_{\mathbf{x}_{i,1:n}|\mathbf{p}_i,\mathbf{y}_{i,1:n}}\Big(\boldsymbol{x}_{i,1:n} \,\Big|\, \boldsymbol{p}^{*(n,k-1)}_{\boldsymbol{c}_i^{(n,k-1)}}, \boldsymbol{y}_{i,1:n}\Big)$ in (4.24) by

$$f_{\mathbf{x}_{i,1:n}|\mathbf{p}_i,\mathbf{y}_{i,1:n}}\Big(\boldsymbol{x}_{i,1:n} \,\Big|\, \boldsymbol{p}^{*(n,k-1)}_{\boldsymbol{c}_i^{(n,k-1)}}, \boldsymbol{y}_{i,1:n}\Big) \approx \sum_{k'=1}^{K} W_i^{(n,k,k')} \delta_{\bar{\boldsymbol{x}}_{i,1:n}^{(n,k')}}(\boldsymbol{x}_{i,1:n}), \tag{4.25}$$

where at the first time step (see (2.39))

$$W_i^{(1,k,k')} \propto \frac{\chi\Big(\boldsymbol{y}_{i,1} \,\Big|\, \bar{\boldsymbol{x}}_{i,1}^{(1,k')}, \boldsymbol{p}_{c_i^{(1,k-1)}}^{*(1,k-1)}\Big)\zeta\Big(\bar{\boldsymbol{x}}_{i,1}^{(1,k')} \,\Big|\, \boldsymbol{p}_{c_i^{(1,k-1)}}^{*(1,k-1)}\Big)}{g_1\Big(\bar{\boldsymbol{x}}_{i,1}^{(1,k')} \,\Big|\, \boldsymbol{p}_{c_i^{(0,k')}}^{*(0,k')}, \boldsymbol{y}_{i,1}\Big)}, \tag{4.26}$$

with $\sum_{k'=1}^{K} W_i^{(1,k,k')} = 1$, and at the time steps $n \geq 2$ (see (2.40))

$$W_i^{(n,k,k')} \propto \frac{\chi\Big(\boldsymbol{y}_{i,n} \,\Big|\, \bar{\boldsymbol{x}}_{i,n}^{(n,k')}, \boldsymbol{p}_{c_i^{(n,k-1)}}^{*(n,k-1)}\Big)\xi\Big(\bar{\boldsymbol{x}}_{i,n}^{(n,k')} \,\Big|\, \bar{\boldsymbol{x}}_{i,n-1}^{(n-1,k')}, \boldsymbol{p}_{c_i^{(n,k-1)}}^{*(n,k-1)}\Big)}{g\Big(\bar{\boldsymbol{x}}_{i,n}^{(n,k')} \,\Big|\, \bar{\boldsymbol{x}}_{i,n-1}^{(n-1,k')}, \boldsymbol{p}_{c_i^{(n-1,k')}}^{*(n-1,k')}, \boldsymbol{y}_{i,n}\Big)}, \tag{4.27}$$

with $\sum_{k'=1}^{K} W_i^{(n,k,k')} = 1$.

The advantage of this approximation is that it allows us to sample the states from the discrete distribution over K particles defined by (4.25) for each TS at each time step. In fact, the computational complexity of sampling from the approximation (4.25) is constant over time. However, we are still required to save the complete particles $\bar{\boldsymbol{x}}_{i,1:n}^{(n,k')}$, which grow with n, for all $k' \in \{1, \ldots, K\}$ and $i \in \{1, \ldots, I\}$. Saving the complete particles $\bar{\boldsymbol{x}}_{i,1:n}^{(n,k')}$ is often required even if (4.6) only depends on $\mathbf{x}_{i,N}$ instead of $\mathbf{x}_{i,1:N}$ for all $i \in \{1, \ldots, I\}$, as, in general, $\mathbf{c}_i$ and $\mathbf{p}_{\mathbf{c}_i}^*$ depend on the complete sequence $\mathbf{x}_{i,1:N}$. As will be explored in detail in Section 4.1.3.4, despite the dependence of $\mathbf{c}_i$ and $\mathbf{p}_{\mathbf{c}_i}^*$ on $\mathbf{x}_{i,1:n}$, we can sometimes introduce recursively calculated sufficient statistics $\mathbf{s}_{i,n} = (\mathbf{s}_{i,n,1}, \mathbf{s}_{i,n,2})$ with fixed dimension $E_1 \times E_2$ for all $n \in \mathbb{N}$ and $i \in \{1, \ldots, I\}$ such that $\mathbf{c}_i$ and $\mathbf{p}_{\mathbf{c}_i}^*$ given $\mathbf{s}_{i,n}$ are independent of $\mathbf{x}_{i,1:n}$. These sufficient statistics then allow us to only save the last intermediate sample $\bar{\boldsymbol{x}}_{i,n}^{(n,k')}$ and the corresponding sufficient statistics $\bar{\boldsymbol{s}}_{i,n}^{(n,k')}$ instead of the complete particle $\bar{\boldsymbol{x}}_{i,1:n}^{(n,k')}$ for all $k' \in \{1, \ldots, K\}$ and $i \in \{1, \ldots, I\}$. We will present procedures to sample from approximation (4.25) based on Gibbs sampling in Section 4.1.3.2 and based on the MH algorithm in Section 4.1.3.3.

Sampling the Indicator Variables and Parameters Within the MCMC algorithm, the indicator variables $\mathbf{c}_{1:I}$ and parameters $\mathbf{p}_{\boldsymbol{c}_{1:I}}^*$ are sampled using Gibbs sampling, similarly to the algorithm discussed in Section 4.1.2. That is, we sample $\mathsf{c}_i = c$ with the conditional probability (see (4.16))

$$b_{i,c}^{(n,k)} = \mathbb{P}\left(\mathsf{c}_i = c \,\middle|\, \mathbf{c}_{\neg i} = \boldsymbol{c}_{\neg i}^{(n,k)}, \mathbf{p}_{\mathbf{c}_{\neg i}}^* = \boldsymbol{p}_{\boldsymbol{c}_{\neg i}^{(n,k)}}^{*(n,k-1)}, \mathbf{x}_{1:I,1:n} = \boldsymbol{x}_{1:I,1:n}^{(n,k)}, \mathbf{y}_{1:I,1:n} = \boldsymbol{y}_{1:I,1:n}\right) \tag{4.28}$$

for all $c \in \left\{c_1^{(n,k)}, \ldots, c_{i-1}^{(n,k)}, c_{i+1}^{(n,k-1)}, \ldots, c_I^{(n,k-1)}\right\}$, where $\mathbf{c}_{\neg i} = (\mathsf{c}_1, \ldots, \mathsf{c}_{i-1}, \mathsf{c}_{i+1}, \ldots, \mathsf{c}_I) = \left(c_1^{(n,k)}, \ldots, c_{i-1}^{(n,k)}, c_{i+1}^{(n,k-1)}, \ldots, c_I^{(n,k-1)}\right) = \boldsymbol{c}_{\neg i}^{(n,k)}$ and $\mathbf{p}_{\mathbf{c}_{\neg i}}^* = (\mathbf{p}_{\mathsf{c}}^*)_{\mathsf{c} \in \{\mathsf{c}_1, \ldots, \mathsf{c}_{i-1}, \mathsf{c}_{i+1}, \ldots, \mathsf{c}_I\}} = \left(\boldsymbol{p}_c^{*(n,k-1)}\right)_{c \in \{c_1^{(n,k)}, \ldots, c_{i-1}^{(n,k)}, c_{i+1}^{(n,k-1)}, \ldots, c_I^{(n,k-1)}\}} = \boldsymbol{p}_{\boldsymbol{c}_{\neg i}^{(n,k)}}^{*(n,k-1)}$, and we sample c_i being distinct

from all the other indicator variables, that is, $\mathsf{c}_i \notin \left\{c_1^{(n,k)}, \dots, c_{i-1}^{(n,k)}, c_{i+1}^{(n,k-1)}, \dots, c_I^{(n,k-1)}\right\}$, with the conditional probability (see (4.17))

$$b_{i,0}^{(n,k)} = \mathbb{P}\left(\mathsf{c}_i \notin \left\{c_1^{(n,k)}, \dots, c_{i-1}^{(n,k)}, c_{i+1}^{(n,k-1)}, \dots, c_I^{(n,k-1)}\right\} \,\middle|\, \mathsf{c}_{\neg i} = \boldsymbol{c}_{\neg i}^{(n,k)}, \mathbf{p}^*_{\mathsf{c}_{\neg i}} = \boldsymbol{p}^{*(n,k-1)}_{\boldsymbol{c}_{\neg i}^{(n,k)}}, \mathbf{x}_{1:I,1:n} = \boldsymbol{x}_{1:I,1:n}^{(n,k)}, \mathbf{y}_{1:I,1:n} = \boldsymbol{y}_{1:I,1:n}\right). \tag{4.29}$$

The samples of the indicator variables $c_i^{(n,k)}$ that are distinct from the other indicator variables need to be assigned a new value, that is, $c_i^{(n,k)} = \max\left\{\max_{i' \in \{1,\dots,i-1\}} c_{i'}^{(n,k)}, c_{\max}^{(n,k-1)}\right\} + 1$ with $c_{\max}^{(n,k-1)} = \max_{k' \in \{1,\dots,k-1\}, i' \in \{1,\dots,I\}} c_{i'}^{(n,k')}$. Furthermore, a parameter $\boldsymbol{p}^{*(n,k-1)}_{c_i^{(n,k)}}$ has to be sampled, which will be discussed presently. Similarly to (4.19), we can simplify (4.28) to

$$b_{i,c}^{(n,k)} \propto \frac{1}{\alpha + I - 1}\left(\sum_{i'=1}^{i-1} \mathbb{1}\left(c_{i'}^{(n,k)} = c\right) + \sum_{i'=i+1}^{I} \mathbb{1}\left(c_{i'}^{(n,k-1)} = c\right)\right)\left(\prod_{n'=1}^{n} \chi\left(\boldsymbol{y}_{i,n'} \,\middle|\, \boldsymbol{x}_{i,n'}^{(n,k)}, \boldsymbol{p}_c^{*(n,k-1)}\right)\right) \times \zeta\left(\boldsymbol{x}_{i,1}^{(n,k)} \,\middle|\, \boldsymbol{p}_c^{*(n,k)}\right) \prod_{n''=2}^{n} \xi\left(\boldsymbol{x}_{i,n''}^{(n,k)} \,\middle|\, \boldsymbol{x}_{i,n''-1}^{(n,k)}, \boldsymbol{p}_c^{*(n,k)}\right). \tag{4.30}$$

Furthermore, similarly to (4.20), we can simplify (4.29) to

$$b_{i,0}^{(n,k)} \propto \frac{\alpha}{\alpha + I - 1} \frac{f_{\mathbf{p}}(\tilde{\boldsymbol{p}})}{f_{\mathbf{p}_i|\mathbf{x}_{i,1:n},\mathbf{y}_{i,1:n}}\left(\tilde{\boldsymbol{p}} \,\middle|\, \boldsymbol{x}_{i,1:n}^{(n,k)}, \boldsymbol{y}_{i,1:n}\right)}\left(\prod_{n'=1}^{n} \chi\left(\boldsymbol{y}_{i,n'} \,\middle|\, \boldsymbol{x}_{i,n'}^{(n,k)}, \tilde{\boldsymbol{p}}\right)\right) \times \zeta\left(\boldsymbol{x}_{i,1}^{(n,k)} \,\middle|\, \tilde{\boldsymbol{p}}\right) \prod_{n''=2}^{n} \xi\left(\boldsymbol{x}_{i,n''}^{(n,k)} \,\middle|\, \boldsymbol{x}_{i,n''-1}^{(n,k)}, \tilde{\boldsymbol{p}}\right), \tag{4.31}$$

with the same proportionality constant as (4.30), and an arbitrary $\tilde{\boldsymbol{p}} \in \mathbb{R}^D$ with $f_{\mathbf{p}}(\tilde{\boldsymbol{p}}) \neq 0$. Note that we assume here that we are able to calculate $f_{\mathbf{p}_i|\mathbf{x}_{i,1:n},\mathbf{y}_{i,1:n}}\left(\tilde{\boldsymbol{p}} \,\middle|\, \boldsymbol{x}_{i,1:n}^{(n,k)}, \boldsymbol{y}_{i,1:n}\right)$.

The parameters $\mathbf{p}_c^*$ for $c \in \left\{c_1^{(n,k)}, \dots, c_I^{(n,k)}\right\}$ are sampled from their conditional distribution given all the other random variables, that is, $\boldsymbol{p}_c^{*(n,k)}$ is sampled from (4.21)

$$f_{\mathbf{p}_c^*|\mathbf{x}_{1:I,1:n},\mathbf{c}_{1:I},\mathbf{p}^*_{\neg c},\mathbf{y}_{1:I,1:n}}\left(\boldsymbol{p}_c^{*(n,k)} \,\middle|\, \boldsymbol{x}_{1:I,1:n}^{(n,k)}, \boldsymbol{c}_{1:I}^{(n,k)}, \boldsymbol{p}_{\neg c}^{*(n,k)}, \boldsymbol{y}_{1:I,1:n}\right) \tag{4.32}$$

with $\mathbf{p}^*_{\neg c} = (\mathbf{p}^*_{c'})_{c' \in \{\mathsf{c}_1,\dots,\mathsf{c}_I\}\setminus\{c\}}$ and $\boldsymbol{p}_{\neg c}^{*(n,k)} = \left(\boldsymbol{p}_{c'}^{*(n,k)'}\right)_{c' \in \{c_1^{(n,k)},\dots,c_I^{(n,k)}\}\setminus\{c\}}$, where $\boldsymbol{p}_{c'}^{*(n,k)'}$ is the newest sample of $\mathbf{p}^*_{c'}$, that is, either $\boldsymbol{p}_{c'}^{*(n,k-1)}$ or $\boldsymbol{p}_{c'}^{*(n,k)}$. Note that the parameter $\boldsymbol{p}^{*(n,k-1)}_{c_i^{(n,k)}}$ for a newly generated class, as discussed above, is also sampled from (4.32), but with the newest currently available samples of the other random variables. Using the same derivation as for (4.23) with N replaced by n and using the samples of the time step n, we can show that

$$f_{\mathbf{p}_c^*|\mathbf{x}_{1:I,1:n},\mathbf{c}_{1:I},\mathbf{p}^*_{\neg c},\mathbf{y}_{1:I,1:n}}\left(\boldsymbol{p}_c^{*(n,k)} \,\middle|\, \boldsymbol{x}_{1:I,1:n}^{(n,k)}, \boldsymbol{c}_{1:I}^{(n,k)}, \boldsymbol{p}_{\neg c}^{*(n,k)}, \boldsymbol{y}_{1:I,1:n}\right) \propto \left(\prod_{i:\boldsymbol{c}_i^{(k)}=c} f_{\mathbf{x}_{i,1:n},\mathbf{y}_{i,1:n}|\mathbf{p}_i}\left(\boldsymbol{x}_{i,1:n}^{(n,k)}, \boldsymbol{y}_{i,1:n} \,\middle|\, \boldsymbol{p}_{c_i}^{*(n,k)}\right)\right) f_{\mathrm{G}}\left(\boldsymbol{p}_c^{*(n,k)}\right). \tag{4.33}$$

Therefore, in order to use this algorithm, we need to be able to sample from (4.33).

Similarly to the generation of the state particles $\boldsymbol{x}_{1:I,1:n}^{(n,k)}$ using the approximation in (4.25), sampling the indicator variables $\boldsymbol{c}_{1:I}^{(n,k)}$ and the parameters $\boldsymbol{p}_{\boldsymbol{c}_{1:I}^{(n,k)}}^{*(n,k)}$ using (4.30)–(4.33) requires us to process the state particles $\boldsymbol{x}_{1:I,1:n}^{(n,k)}$ and observations $\boldsymbol{y}_{1:I,1:n}$, which also grow with n. Therefore, in general, the computational complexity of this algorithm grows with n. As noted in the paragraph above, it is, however, sometimes possible to introduce sufficient statistics, with the same dimensionality at each time step, which allow us to avoid saving and computing an increasing amount of data as time progresses. This scenario will be discussed in Section 4.1.3.4.

While the development of a proof of convergence for this algorithm is out of the scope of this thesis, we argue that the performance evaluation in Section 5.3 suggests convergence to the true distribution with an increasing number of samples.

4.1.3.2 Gibbs Sampling Algorithm

In this section, we present a version of the SMC algorithm discussed above that generates samples of the states $\mathbf{x}_{1:I,1:n}$, indicator variables $\mathbf{c}_{1:I}$, and parameters $\mathbf{p}_{\mathbf{c}_{1:I}}^{*}$. Here, the MCMC algorithm samples the states directly from their approximated conditional pdf (4.25). That is, we have to calculate the weights $\left(W_{1:I}^{(n,k,k')}\right)_{k'\in\{1,\dots,K\}}$ at each iteration of the MCMC algorithm according to (4.26) and (4.27) using the current samples. The algorithm for the first time step is stated in Algorithm 4.4 and for the time steps $n \geq 2$ in Algorithm 4.5, with the MCMC algorithm stated in Algorithm 4.6.

4.1.3.3 Metropolis-Hastings Algorithm

One drawback of the algorithm presented in Section 4.1.3.2 is that the Gibbs sampling step in Algorithm 4.6 requires the calculation of IK^2 weights, that is, K weights for each of the I TSs and K samples, which makes this method infeasible for large K. Therefore, we now propose an algorithm where we use the MH algorithm from Section 2.4.2 to sample the states from approximation (4.25), which only requires the calculation of IK weights and IK acceptance probabilities at each time step.

In order to apply the MH algorithm, we need to define a proposal pdf whose support includes the support of (4.25). This can be done by using the particles $\left(\bar{\boldsymbol{x}}_{1:I,1:n}^{(n,k)}\right)_{k\in\{1,\dots,K\}}$ and calculating their weights for the parameters that were used to generate them. That

Algorithm 4.4 SMC algorithm using Gibbs sampling for a BNP state-space model: initialization

Input: $\boldsymbol{y}_{1:I,1}, K, g_1$

1: **for all** $i = 1, \ldots, I$ **do**
2: **for all** $k = 1, \ldots, K$ **do**
3: set $c_i^{(0,k)} = i$
4: sample $\boldsymbol{p}_{c_i^{(0,k)}}^{*(0,k)}$ from $f_{\mathrm{G}}\left(\boldsymbol{p}_{c_i^{(0,k)}}^{*(0,k)}\right)$
5: sample $\bar{\boldsymbol{x}}_{i,1}^{(1,k)}$ from $g_1\left(\bar{\boldsymbol{x}}_{i,1}^{(1,k)} \,\middle|\, \boldsymbol{p}_{c_i^{(0,k)}}^{*(0,k)}, \boldsymbol{y}_{i,1}\right)$
6: **end for**
7: **end for**
8: sample $\left(\boldsymbol{x}_{1:I,1}^{(1,k)}, \boldsymbol{c}_{1:I}^{(1,k)}, \boldsymbol{p}_{\boldsymbol{c}_{1:I}^{(1,k)}}^{*(1,k)}\right)_{k\in\{1,\ldots,K\}}$ using Algorithm 4.6 with input $(\bar{\boldsymbol{x}}_{1:I,1}^{(1,k)})_{k\in\{1,\ldots,K\}}, \boldsymbol{y}_{1:I,1}, K, g_1$

Output: $\left(\boldsymbol{x}_{1:I,1}^{(1,k)}, \boldsymbol{c}_{1:I,1}^{(1,k)}, \boldsymbol{p}_{\boldsymbol{c}_{1:I,1}^{(1,k)}}^{*(1,k)}\right)_{k\in\{1,\ldots,K\}}$

Algorithm 4.5 SMC algorithm using Gibbs sampling for a BNP state-space model: iteration

Input: $\left(\boldsymbol{x}_{1:I,1:n-1}^{(n-1,k)}, \boldsymbol{c}_{1:I}^{(n-1,k)}, \boldsymbol{p}_{\boldsymbol{c}_{1:I}^{(n-1,k)}}^{*(n-1,k)}\right)_{k\in\{1,\ldots,K\}}, \boldsymbol{y}_{1:I,1:n}, K, g$

1: **for all** $i = 1, \ldots, I$ **do**
2: **for all** $k = 1, \ldots, K$ **do**
3: sample $\bar{\boldsymbol{x}}_{i,n}^{(n,k)}$ from $g\left(\bar{\boldsymbol{x}}_{i,n}^{(n,k)} \,\middle|\, \boldsymbol{x}_{i,n-1}^{(n-1,k)}, \boldsymbol{p}_{c_i^{(n-1,k)}}^{*(n-1,k)}, \boldsymbol{y}_{i,n}\right)$
4: set $\bar{\boldsymbol{x}}_{i,1:n}^{(n,k)} = \left(\boldsymbol{x}_{i,1:n-1}^{(n-1,k)}, \bar{\boldsymbol{x}}_{i,n}^{(n,k)}\right)$
5: **end for**
6: **end for**
7: sample $\left(\boldsymbol{x}_{1:I,1:n}^{(n,k)}, \boldsymbol{c}_{1:I}^{(n,k)}, \boldsymbol{p}_{\boldsymbol{c}_{1:I}^{(n,k)}}^{*(n,k)}\right)_{k\in\{1,\ldots,K\}}$ using Algorithm 4.6 with input $\left(\bar{\boldsymbol{x}}_{1:I,1:n}^{(n,k)}\right)_{k\in\{1,\ldots,K\}}, \boldsymbol{y}_{1:I,1:n}, K, g$

Output: $\left(\boldsymbol{x}_{1:I,1:n}^{(n,k)}, \boldsymbol{c}_{1:I}^{(n,k)}, \boldsymbol{p}_{\boldsymbol{c}_{1:I}^{(n,k)}}^{*(n,k)}\right)_{k\in\{1,\ldots,K\}}$

Algorithm 4.6 SMC algorithm using Gibbs sampling for a BNP state-space model: MCMC part

Input: $\left(\bar{\boldsymbol{x}}_{1:I,1:n}^{(n,k)}\right)_{k\in\{1,\dots,K\}}, \boldsymbol{y}_{1:I,1:n}, K, g_1$ or g

for all $i = 1,\dots,I$ **do**
  initialize $\boldsymbol{x}_{i,1:n}^{(n,1)}$ arbitrarily from $\left\{\bar{\boldsymbol{x}}_{i,1:n}^{(n,k)} : k \in \{1,\dots,K\}\right\}$
  set $c_i^{(n,1)} = i$
  initialize $\boldsymbol{p}_{c_i^{(n,1)}}^{*(n,1)}$ arbitrarily from $\mathbb{R}^D$
end for
set $c_{\max}^{(n,1)} = I$
for all $k = 2,\dots,K$ **do**
  for all $i = 1,\dots,I$ **do**
    for all $k' = 1,\dots,K$ **do**
      calculate $W_i^{(n,k,k')}$ according to (4.26) or (4.27)
    end for
    sample $\boldsymbol{x}_{i,1:n}^{(n,k)}$ from (4.25)
  end for
  for all $i = 1,\dots,I$ **do**
    sample $c_i^{(n,k)} = c$ with probability $b_{i,c}^{(n,k)}$ (see (4.30) and (4.31))
    for all $c \in \left\{0, c_1^{(n,k)},\dots,c_{i-1}^{(n,k)}, c_{i+1}^{(n,k-1)},\dots,c_I^{(n,k-1)}\right\}$
    if $c_i^{(n,k)} = 0$
      set $c_i^{(n,k)} = \max\left\{\max_{i'\in\{1,\dots,i-1\}} c_{i'}^{(n,k)}, c_{\max}^{(n,k-1)}\right\} + 1$
      sample $\boldsymbol{p}_{c_i^{(n,k)}}^{*(n,k-1)}$ from (4.33)
  end for
  set $c_{\max}^{(n,k)} = \max\left\{\max_{i\in\{1,\dots,I\}} c_i^{(n,k)}, c_{\max}^{(n,k-1)}\right\}$
  for all $c \in \left\{c_1^{(n,k)},\dots,c_I^{(n,k)}\right\}$ **do**
    sample $\boldsymbol{p}_c^{*(n,k)}$ from (4.33)
  end for
end for

Output: $\left(\boldsymbol{x}_{1:I,1:n}^{(n,k)}, \boldsymbol{c}_{1:I}^{(n,k)}, \boldsymbol{p}_{c_{1:I}^{(n,k)}}^{*(n,k)}\right)_{k\in\{1,\dots,K\}}$

is, we define the proposal pdf for the MH step as (see (4.25))

$$g_{\mathrm{MH},n}\Big(\boldsymbol{x}_{i,1:n} \,\Big|\, \Big(\bar{\boldsymbol{x}}_{i,1:n}^{(n,k)}, \boldsymbol{p}_{c_i^{(n-1,k)}}^{*(n-1,k)}\Big)_{k\in\{1,\dots,K\}}, \boldsymbol{y}_{i,n}\Big) = \sum_{k=1}^{K} W_i^{\prime(n,k)} \delta_{\bar{\boldsymbol{x}}_{i,1:n}^{(n,k)}}\Big(\boldsymbol{x}_{i,1:n}\Big), \tag{4.34}$$

where at the first time step (see (4.26))

$$W_i^{\prime(1,k)} \propto \frac{\chi\Big(\boldsymbol{y}_{i,1} \,\Big|\, \bar{\boldsymbol{x}}_{i,1}^{(1,k)}, \boldsymbol{p}_{c_i^{(0,k)}}^{*(0,k)}\Big)\zeta\Big(\bar{\boldsymbol{x}}_{i,1}^{(1,k)} \,\Big|\, \boldsymbol{p}_{c_i^{(0,k)}}^{*(0,k)}\Big)}{g_1\Big(\bar{\boldsymbol{x}}_{i,1}^{(1,k)} \,\Big|\, \boldsymbol{p}_{c_i^{(0,k)}}^{*(0,k)}, \boldsymbol{y}_{i,1}\Big)}, \tag{4.35}$$

and at the time steps $n \geq 2$ (see (4.27))

$$W_i^{\prime(n,k)} \propto \frac{\chi\Big(\boldsymbol{y}_{i,n} \,\Big|\, \bar{\boldsymbol{x}}_{i,n}^{(n,k)}, \boldsymbol{p}_{c_i^{(n-1,k)}}^{*(n-1,k)}\Big)\xi\Big(\bar{\boldsymbol{x}}_{i,n}^{(n,k)} \,\Big|\, \bar{\boldsymbol{x}}_{i,n-1}^{(n,k)}, \boldsymbol{p}_{c_i^{(n-1,k)}}^{*(n-1,k)}\Big)}{g\Big(\bar{\boldsymbol{x}}_{i,n}^{(n,k)} \,\Big|\, \bar{\boldsymbol{x}}_{i,n-1}^{(n,k)}, \boldsymbol{p}_{c_i^{(n-1,k)}}^{*(n-1,k)}, \boldsymbol{y}_{i,n}\Big)} \tag{4.36}$$

with $\sum_{k=1}^{K} W_i^{\prime(n,k)} = 1$. Note that the proposal pdf (4.34) is equal to the approximation (4.25) using the parameter $\boldsymbol{p}_{c_i^{(n-1,k)}}^{*(n-1,k)}$ that was used to generated the intermediate particle $\bar{\boldsymbol{x}}_{i,1:n}^{(n,k)}$ instead of the parameter $\boldsymbol{p}_{c_i^{(n,k-1)}}^{*(n,k-1)}$ of the last MCMC iteration.

At each iteration, the algorithm first samples a proposal particle $\boldsymbol{x}_{i,1:n}^{\prime(n,k)}$ from (4.34) and then accepts it $\boldsymbol{x}_{i,1:n}^{(n,k)} = \boldsymbol{x}_{i,1:n}^{\prime(n,k)}$ with probability $P_{\mathrm{a},i}^{(n,k)}$, which will be defined presently, and rejects it $\boldsymbol{x}_{i,1:n}^{(n,k)} = \boldsymbol{x}_{i,1:n}^{(n,k-1)}$ with probability $1 - P_{\mathrm{a},i}^{(n,k)}$. We define the acceptance probability $P_{\mathrm{a},i}^{(n,k)}$ as (see (2.42))

$$P_{\mathrm{a},i}^{(n,k)} = \min\left\{1, \frac{f_{\boldsymbol{p}_i|\mathbf{x}_{i,1:n},\mathbf{y}_{i,1:n}}\Big(\boldsymbol{p}_{c_i^{(n,k-1)}}^{*(n,k-1)} \,\Big|\, \boldsymbol{x}_{i,1:n}^{\prime(n,k)}, \boldsymbol{y}_{i,1:n}\Big) W_{\mathrm{MH},i}^{\prime(n,k-1)}}{f_{\boldsymbol{p}_i|\mathbf{x}_{i,1:n},\mathbf{y}_{i,1:n}}\Big(\boldsymbol{p}_{c_i^{(n,k-1)}}^{*(n,k-1)} \,\Big|\, \boldsymbol{x}_{i,1:n}^{(n,k-1)}, \boldsymbol{y}_{i,1:n}\Big) W_{\mathrm{MH},i}^{\prime(n,k,\mathrm{prop})}}\right\}, \tag{4.37}$$

where $W_{\mathrm{MH},i}^{\prime(n,k,\mathrm{prop})}$ and $W_{\mathrm{MH},i}^{\prime(n,k-1)}$ are the probabilities of $\boldsymbol{x}_{i,1:n}^{\prime(n,k)}$ and $\boldsymbol{x}_{i,1:n}^{(n,k-1)}$, respectively, when they are sampled from the proposal pdf $g_{\mathrm{MH},n}\Big(\boldsymbol{x}_{i,1:n} \,\Big|\, \Big(\bar{\boldsymbol{x}}_{i,1:n}^{(n,k)}, \boldsymbol{p}_{c_i^{(n-1,k)}}^{*(n-1,k)}\Big)_{k\in\{1,\dots,K\}}, \boldsymbol{y}_{i,n}\Big)$ in (4.34). That is, for all $k' \in \{1,\dots,K\}$, $W_{\mathrm{MH},i}^{\prime(n,k,\mathrm{prop})} = W_i^{\prime(n,k')}$ if $\boldsymbol{x}_{i,1:n}^{\prime(n,k)} = \bar{\boldsymbol{x}}_{i,1:n}^{(n,k')}$. Similarly, again for all $k' \in \{1,\dots,K\}$, $W_{\mathrm{MH},i}^{\prime(n,k-1)} = W_i^{\prime(n,k')}$ if $\boldsymbol{x}_{i,1:n}^{(n,k-1)} = \bar{\boldsymbol{x}}_{i,1:n}^{(n,k')}$. Note that compared to (2.42), we use the probabilities $W_{\mathrm{MH},i}^{\prime(n,k,\mathrm{prop})}$ and $W_{\mathrm{MH},i}^{\prime(n,k-1)}$ instead $g(\tilde{\boldsymbol{x}}^{(k)} \,|\, \boldsymbol{x}^{(k-1)})$ and $g(\boldsymbol{x}^{(k-1)} \,|\, \tilde{\boldsymbol{x}}^{(k)})$, respectively, as $g_{\mathrm{MH},n}\Big(\boldsymbol{x}_{i,1:n} \,\Big|\, \Big(\bar{\boldsymbol{x}}_{i,1:n}^{(n,k)}, \boldsymbol{p}_{c_i^{(n-1,k)}}^{*(n-1,k)}\Big)_{k\in\{1,\dots,K\}}, \boldsymbol{y}_{i,n}\Big)$ defines a discrete distribution.

The complete algorithm for the first time step is outlined in Algorithm 4.7 and for the time steps $n \geq 2$ in Algorithm 4.8, with the MCMC algorithm stated in Algorithm 4.9.

4.1.3.4 Sufficient Statistics

For many applications, the expectation (4.6) only depends on the state current time step n. That is we can express $h(\boldsymbol{x}_{1:I,1:n}, \boldsymbol{c}_{1:I}, \boldsymbol{p}_{\boldsymbol{c}_{1:I}}^{*})$ in (4.6) as

$$h(\boldsymbol{x}_{1:I,1:n}, \boldsymbol{c}_{1:I}, \boldsymbol{p}_{\boldsymbol{c}_{1:I}}^{*}) = h'(\boldsymbol{x}_{1:I,n}, \boldsymbol{c}_{1:I}, \boldsymbol{p}_{\boldsymbol{c}_{1:I}}^{*}), \tag{4.38}$$

Algorithm 4.7 SMC algorithm using MH for a BNP state-space model: initialization

Input: $\boldsymbol{y}_{1:I,1}, K, g_1$

for all $i = 1, \ldots, I$ **do**
  for all $k = 1, \ldots, K$ **do**
    set $c_i^{(0,k)} = i$
    sample $\boldsymbol{p}_{c_i^{(0,k)}}^{*(0,k)}$ from $f_{\mathrm{G}}\left(\boldsymbol{p}_{c_i^{(0,k)}}^{*(0,k)}\right)$
    sample $\bar{\boldsymbol{x}}_{i,1}^{(1,k)}$ from $g_1\left(\bar{\boldsymbol{x}}_{i,1}^{(1,k)} \,\middle|\, \boldsymbol{p}_{c_i^{(0,k)}}^{*(0,k)}, \boldsymbol{y}_{i,1}\right)$
    calculate $W_i'^{(1,k)}$ according to (4.35)
  end for
end for
sample $\left(\boldsymbol{x}_{1:I,1}^{(1,k)}, \boldsymbol{c}_{1:I}^{(1,k)}, \boldsymbol{p}_{\boldsymbol{c}_{1:I}^{(1,k)}}^{*(1,k)}\right)_{k\in\{1,\ldots,K\}}$ using Algorithm 4.9 with input $\left(\bar{\boldsymbol{x}}_{1:I,1}^{(1,k)}, W_{1:I}'^{(1,k)}\right)_{k\in\{1,\ldots,K\}}, K, \boldsymbol{y}_{1:I,1}$

Output: $\left(\boldsymbol{x}_{1:I,1}^{(1,k)}, \boldsymbol{c}_{1:I}^{(1,k)}, \boldsymbol{p}_{\boldsymbol{c}_{1:I}^{(1,k)}}^{*(1,k)}\right)_{k\in\{1,\ldots,K\}}$

Algorithm 4.8 SMC algorithm using MH for a BNP state-space model: iteration

Input: $\left(\boldsymbol{x}_{1:I,1:n-1}^{(n-1,k)}, \boldsymbol{c}_{1:I}^{(n-1,k)}, \boldsymbol{p}_{\boldsymbol{c}_{1:I}^{(n-1,k)}}^{*(n-1,k)}\right)_{k\in\{1,\ldots,K\}}, \boldsymbol{y}_{1:I,1:n}, K, g$

for all $i = 1, \ldots, I$ **do**
  for all $k = 1, \ldots, K$ **do**
    sample $\bar{\boldsymbol{x}}_{i,n}^{(n,k)}$ from $g\left(\bar{\boldsymbol{x}}_{i,n}^{(n,k)} \,\middle|\, \boldsymbol{x}_{i,n-1}^{(n-1,k)}, \boldsymbol{p}_{c_i^{(n-1,k)}}^{*(n-1,k)}, \boldsymbol{y}_{i,n}\right)$
    set $\bar{\boldsymbol{x}}_{i,1:n}^{(n,k)} = (\boldsymbol{x}_{i,1:n-1}^{(n-1,k)}, \bar{\boldsymbol{x}}_{i,n}^{(n,k)})$
    calculate $W_i'^{(n,k)}$ according to (4.36)
  end for
end for
sample $\left(\boldsymbol{x}_{1:I,1:n}^{(n,k)}, \boldsymbol{c}_{1:I}^{(n,k)}, \boldsymbol{p}_{\boldsymbol{c}_{1:I}^{(n,k)}}^{*(n,k)}\right)_{k\in\{1,\ldots,K\}}$ using Algorithm 4.9 with input $\left(\bar{\boldsymbol{x}}_{1:I,1:n}^{(n,k)}, W_{1:I}'^{(n,k)}\right)_{k\in\{1,\ldots,K\}}, K, \boldsymbol{y}_{1:I,1:n}$

Output: $\left(\boldsymbol{x}_{1:I,1:n}^{(n,k)}, \boldsymbol{c}_{1:I}^{(n,k)}, \boldsymbol{p}_{\boldsymbol{c}_{1:I}^{(n,k)}}^{*(n,k)}\right)_{k\in\{1,\ldots,K\}}$

Algorithm 4.9 SMC algorithm using MH for a BNP state-space model: MCMC part

Input: $\left(\bar{\boldsymbol{x}}_{1:I,1:n}^{(n,k)}, W_{1:I}^{\prime(n,k)}\right)_{k\in\{1,\ldots,K\}}, K, \boldsymbol{y}_{1:I,1:n}$

for all $i = 1,\ldots,I$ **do**
  initialize $\boldsymbol{x}_{i,1:n}^{(n,1)}$ arbitrarily from $\left\{\bar{\boldsymbol{x}}_{i,1:n}^{(n,k)} : k \in \{1,\ldots,K\}\right\}$
  set $c_i^{(n,1)} = i$
  initialize $\boldsymbol{p}_{c_i^{(n,1)}}^{*(n,1)}$ arbitrarily from $\mathbb{R}^D$
end for
set $c_{\max}^{(n,1)} = I$
for all $k = 2,\ldots,K$ **do**
  for all $i = 1,\ldots,I$ **do**
    sample $\boldsymbol{x}_{i,1:n}^{\prime(n,k)} = \bar{\boldsymbol{x}}_{i,1:n}^{(n,k')}$ with probability $W_i^{\prime(n,k')}$ for all $k' \in \{1,\ldots,K\}$
    set $\boldsymbol{x}_{i,1:n}^{(n,k)} = \boldsymbol{x}_{i,1:n}^{\prime(n,k)}$ with probability $P_{\mathrm{a},i}^{(n,k)}$ (see (4.37))
    else set $\boldsymbol{x}_{i,1:n}^{(n,k)} = \boldsymbol{x}_{i,1:n}^{(n,k-1)}$
  end for
  for all $i = 1,\ldots,I$ **do**
    sample $c_i^{(n,k)} = c$ with probability $b_{i,c}^{(n,k)}$ (see (4.30) and (4.31))
    for all $c \in \left\{0, c_1^{(n,k)},\ldots,c_{i-1}^{(n,k)}, c_{i+1}^{(n,k-1)},\ldots,c_I^{(n,k-1)}\right\}$
    if $c_i^{(n,k)} = 0$
      set $c_i^{(n,k)} = \max\left\{\max_{i'\in\{1,\ldots,i-1\}} c_{i'}^{(n,k)}, c_{\max}^{(n,k-1)}\right\} + 1$
      sample $\boldsymbol{p}_{c_i^{(n,k)}}^{*(n,k-1)}$ from (4.33)
  end for
  set $c_{\max}^{(n,k)} = \max\left\{\max_{i\in\{1,\ldots,I\}} c_i^{(n,k)}, c_{\max}^{(n,k-1)}\right\}$
  for all $c \in \left\{c_1^{(n,k)},\ldots,c_I^{(n,k)}\right\}$ **do**
    sample $\boldsymbol{p}_c^{*(n,k)}$ from (4.33)
  end for
end for

Output: $\left(\boldsymbol{x}_{1:I,1:n}^{(n,k)}, \boldsymbol{c}_{1:I}^{(n,k)}, \boldsymbol{p}_{\boldsymbol{c}_{1:I}^{(n,k)}}^{*(n,k)}\right)_{k\in\{1,\ldots,K\}}$

with some function $h'(\boldsymbol{x}_{1:I,n}, \boldsymbol{c}_{1:I}, \boldsymbol{p}^*_{\boldsymbol{c}_{1:I}})$, and therefore

$$\begin{aligned} J_n &= \mathbb{E}\left(h(\mathbf{x}_{1:I,1:n}, \mathbf{c}_{1:I}, \mathbf{p}^*_{\mathbf{c}_{1:I}}) \,\middle|\, \mathbf{y}_{1:I,1:n} = \boldsymbol{y}_{1:I,1:n}\right) \\ &= \mathbb{E}\left(h'(\mathbf{x}_{1:I,n}, \mathbf{c}_{1:I}, \mathbf{p}^*_{\mathbf{c}_{1:I}}) \,\middle|\, \mathbf{y}_{1:I,1:n} = \boldsymbol{y}_{1:I,1:n}\right) \\ &\approx \frac{1}{K}\sum_{k=1}^{K} h'\left(\boldsymbol{x}^{(n,k)}_{1:I,n}, \boldsymbol{c}^{(n,k)}_{1:I}, \boldsymbol{p}^{*(n,k)}_{\boldsymbol{c}^{(n,k)}_{1:I}}\right). \end{aligned} \tag{4.39}$$

While the expectation (4.39) does not directly involve the states of the previous time steps $\mathbf{x}_{1:I,1:n-1}$, we may still have to sample them in practice to generate samples of the indicator variables $\mathbf{c}_{1:I}$ and parameters $\mathbf{p}^*_{\mathbf{c}_{1:I}}$. Similarly, we also have to consider all observations up to the current time step, $\mathbf{y}_{1:I,1:n}$. That is, we often sample $\mathbf{c}_{1:I}$ and $\mathbf{p}^*_{\mathbf{c}_{1:I}}$ given $\mathbf{x}_{1:I,1:n}$ and $\mathbf{y}_{1:I,1:n}$ (see (4.30)–(4.33)). As discussed previously in Section 4.1.3, this generally means that the computational complexity and the amount of data saved grow with n.

However, it is sometimes possible to simplify the expressions (4.30)–(4.33) by introducing recursively calculated sufficient statistics, with constant dimensionality for all time steps, so that the computational complexity and the amount of data saved at each time step remain constant over time. More formally, we assume that we have sufficient statistics $\mathbf{s}_{i,n} = (\mathbf{s}_{i,n,1}, \mathbf{s}_{i,n,2}) \in \mathbb{R}^{E_1 \times E_2}$ for all $n \in \mathbb{N}$ and $i \in \{1, \ldots, I\}$ with $E_1, E_2 \in \mathbb{N}$, where $\mathbf{s}_{i,1} = u_1(\mathbf{x}_1, \mathbf{y}_1)$ and, for $n \geq 2$, $\mathbf{s}_{i,n} = u(\mathbf{s}_{i,n-1}, \mathbf{x}_n, \mathbf{y}_n)$ for some functions u_1 and u, such that

$$f_{\mathbf{p}_i|\mathbf{x}_{i,1:n},\mathbf{y}_{i,1:n}}(\boldsymbol{p}_{i,n} \,|\, \boldsymbol{x}_{i,1:n}, \boldsymbol{y}_{i,1:n}) = f_{\mathbf{p}_i|\mathbf{s}_{i,n,1}}(\boldsymbol{p}_{i,n} \,|\, \boldsymbol{s}_{i,n,1}), \tag{4.40}$$

$$f_{\mathbf{p}^*_c|\mathbf{x}_{1:I,1:n},\mathbf{c}_{1:I},\mathbf{y}_{1:I,1:n}}(\boldsymbol{p}^*_{c,n} \,|\, \boldsymbol{x}_{1:I,1:n}, \boldsymbol{c}_{1:I,n}, \boldsymbol{y}_{1:I,1:n}) = f_{\mathbf{p}^*_c|\mathbf{c}_{1:I},\mathbf{s}_{1:I,n,2}}(\boldsymbol{p}^*_{c,n} \,|\, \boldsymbol{c}_{1:I,n}, \boldsymbol{s}_{1:I,n,2}). \tag{4.41}$$

We can then simplify (4.30) and (4.31) using (4.40), as well as (4.32) using (4.41). This allows us to sample the indicator variables and parameters given the sufficient statistics without directly using the states and observations.

4.2 Time Series of Different Lengths

4.2.1 Statistical Model and Inference

The statistical model of Section 4.1.1 can be generalized to TSs of possibly different lengths. The data set now contains $I \in \mathbb{N}$ TSs $(\boldsymbol{y}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}})_{i\in\{1,\ldots,I\}}$ with $\boldsymbol{y}_{i,n} \in \mathbb{R}^Y$ for all $n \in \{N_{\mathrm{S},i}, \ldots, N_{\mathrm{E},i}\}$, between starting time $N_{\mathrm{S},i} \in \mathbb{N}$ and ending time $N_{\mathrm{E},i} \in \mathbb{N}$. The TSs are again modeled as observations of a parameter dependent state-space model

(see Section 2.3.5), where the prior distribution of the parameter is chosen as a DP (see Section 3.1). Similarly to Section 4.1.1, we define the parameter dependent state-space model as follows. For each TS indexed by $i \in \mathbb{N}$, we denote the sequence of states as $\mathbf{x}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}}$ with $\mathbf{x}_{i,n} \in \mathbb{R}^X$, the sequence of observations as $\mathbf{y}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}}$ with $\mathbf{y}_{i,n} \in \mathbb{R}^Y$, and the parameter as $\mathbf{p}_i \in \mathbb{R}^D$, with dimensions $X, Y, D \in \mathbb{N}$. The states and observations $(\mathbf{x}_{i,n}, \mathbf{y}_{i,n})_{n \in \{N_{\mathrm{S},i},\ldots,N_{\mathrm{E},i}\}}$, are modeled as mutually independent for different i given the parameters $(\mathbf{p}_{i'})_{i \in \mathbb{N}}$. They are defined, for all $i \in \mathbb{N}$, by conditional pdfs $f_{\mathbf{x}_{i,N_{\mathrm{S},i}}|\mathbf{p}_i}(\boldsymbol{x}_{i,N_{\mathrm{S},i}} \,|\, \boldsymbol{p}_i) = \zeta(\boldsymbol{x}_{i,N_{\mathrm{S},i}} \,|\, \boldsymbol{p}_i)$, $f_{\mathbf{x}_{i,n}|\mathbf{x}_{i,n-1},\mathbf{p}_i}(\boldsymbol{x}_{i,n} \,|\, \boldsymbol{x}_{i,n-1}, \boldsymbol{p}_i) = \xi(\boldsymbol{x}_{i,n} \,|\, \boldsymbol{x}_{i,n-1}, \boldsymbol{p}_i)$ for all $n \in \{N_{\mathrm{S},i}+1, \ldots, N_{\mathrm{E},i}\}$, and $f_{\mathbf{y}_{i,n}|\mathbf{x}_{i,n},\mathbf{p}_i}(\boldsymbol{y}_{i,n} \,|\, \boldsymbol{x}_{i,n}, \boldsymbol{p}_i) = \chi(\boldsymbol{y}_{i,n} \,|\, \boldsymbol{x}_{i,n}, \boldsymbol{p}_i)$ for all $n \in \{N_{\mathrm{S},i}, \ldots, N_{\mathrm{E},i}\}$.

As in Section 4.1.1, the parameters $(\mathbf{p}_i)_{i \in \mathbb{N}}$ are distributed according to a DP with concentration α and base pdf $f_{\mathrm{G}}(\boldsymbol{p}_i)$ (see Definition 3.1).

In this case, we are interested in generating samples to approximate the expectations (4.6) and (4.7), and the most likely class assignments (4.8), with the obvious changes for TSs of different lengths.

4.2.2 Particle Markov Chain Monte Carlo Algorithm

The PMCMC algorithm discussed in Section 4.1.2 can be adapted to TSs of different lengths with minimal changes. That is, for each TS $i \in \{1, \ldots, I\}$, we simply run the algorithm from time step $N_{\mathrm{S},i}$ to time step $N_{\mathrm{E},i}$ instead of from 1 to N.

4.2.3 Sequential Monte Carlo Algorithm

4.2.3.1 General Principle

In this section, we will adapt the SMC algorithm for TSs of equal length discussed in Section 4.1.3 to TSs of different lengths. Compared to Section 4.2.2, this adaptation is more challenging. As the TSs can start at different times, we are required to be able to initialize TSs at each time step, instead of only at the first time step. Further, at each time step, some TSs may have already ended; however, they still influence the class assignment and the parameter inference.

To distinguish between TSs that are starting at the current time step n or have already ended in a previous time step, we introduce the index set of all TSs starting at the current time step, $\mathfrak{I}_{\mathrm{S},n} = \{i \in \{1, \ldots, I\} : N_{\mathrm{S},i} = n\}$, the index set of all currently active TSs, that is, started but not ended, $\mathfrak{I}_{\mathrm{A},n} = \{i \in \{1, \ldots, I\} : N_{\mathrm{S},i} \leq n \leq N_{\mathrm{E},i}\}$, and the index set of all TSs that ended before the current time step, $\mathfrak{I}_{\mathrm{E},n} = \{i \in \{1, \ldots, I\} : N_{\mathrm{E},i} < n\}$,

for all $n \in \mathbb{N}$. Note that $\mathfrak{I}_{\mathrm{S},n} \subseteq \mathfrak{I}_{\mathrm{A},n}$. In contrast to Section 4.1.3, we will have to iterate over the indices of the currently active and already ended TSs, that is, $i \in \mathfrak{I}_{\mathrm{A},n} \cup \mathfrak{I}_{\mathrm{E},n}$, instead of the indices of the TSs $i \in \{1, \ldots, I\}$. Therefore, we use an arbitrary bijective mapping τ_n from $\{1, \ldots, |\mathfrak{I}_{\mathrm{A},n} \cup \mathfrak{I}_{\mathrm{E},n}|\}$ to $\mathfrak{I}_{\mathrm{A},n} \cup \mathfrak{I}_{\mathrm{E},n}$ to induce an order on this set.

Intermediate State Particles At each time step $n \in \mathbb{N}$, we have to initialize the intermediate state particles of the currently starting TSs. That is, for all $i \in \mathfrak{I}_{\mathrm{S},n}$, we sample $\left(\bar{\boldsymbol{x}}_{i,n}^{(n,k)}\right)_{k\in\{1,\ldots,K\}}$ using the initialization procedure to generate the intermediate state particles for time step 1 of Section 4.1.3 with the obvious changes for TSs of different lengths. Similarly, for each active but not currently starting TS $i \in \mathfrak{I}_{\mathrm{A},n} \setminus \mathfrak{I}_{\mathrm{S},n}$, we adapt the procedure to generate the intermediate state particles for time steps $n \geq 2$ of Section 4.1.3 to obtain the intermediate state particles $\left(\bar{\boldsymbol{x}}_{i,N_{\mathrm{S},i}:n}^{(n,k)}\right)_{k\in\{1,\ldots,K\}}$. Due to the dependence of the indicator variables and parameters on the already ended TSs, we also need to sample the states of the ended TSs using the MCMC algorithm. Therefore, we use the intermediate state particles of the last time step where these TSs were active, that is, $\left(\bar{\boldsymbol{x}}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}}^{(N_{\mathrm{E},i},k)}\right)_{k\in\{1,\ldots,K\}}$ generated at time step $N_{\mathrm{E},i}$ for each $i \in \mathfrak{I}_{\mathrm{E},n}$.

After the proposal particles are generated, an MCMC algorithm is used at each time step $n \in \mathbb{N}$ to generate samples of the states, $\left(\boldsymbol{x}_{i,N_{\mathrm{S},i}:n}^{(n,k)}\right)_{i\in\mathfrak{I}_{\mathrm{A},n}}$ and $\left(\boldsymbol{x}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}}^{(n,k)}\right)_{i\in\mathfrak{I}_{\mathrm{E},n}}$, of the indicator variables, $\left(c_i^{(n,k)}\right)_{i\in\mathfrak{I}_{\mathrm{A},n}\cup\mathfrak{I}_{\mathrm{E},n}}$, and of the parameters $\left(\boldsymbol{p}_c^{*(n,k)}\right)_{c\in\{c_i^{(n,k)}:i\in\mathfrak{I}_{\mathrm{A},n}\cup\mathfrak{I}_{\mathrm{E},n}\}}$, for all $k \in \{1, \ldots, K\}$.

Initialization of the MCMC Algorithm The MCMC algorithm is initialized with samples of the states $\left(\boldsymbol{x}_{i,N_{\mathrm{S},i}:n}^{(n,1)}\right)_{i\in\mathfrak{I}_{\mathrm{A},n}}$ and $\left(\boldsymbol{x}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}}^{(n,1)}\right)_{i\in\mathfrak{I}_{\mathrm{E},n}}$, where $\boldsymbol{x}_{i,N_{\mathrm{S},i}:n}^{(n,1)}$ is chosen arbitrarily from $\left\{\bar{\boldsymbol{x}}_{i,N_{\mathrm{S},i}:n}^{(n,k)} : k \in \{1, \ldots, K\}\right\}$ for all $i \in \mathfrak{I}_{\mathrm{A},n}$ and $\boldsymbol{x}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}}^{(n,1)}$ is chosen arbitrarily from $\left\{\bar{\boldsymbol{x}}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}}^{(N_{\mathrm{E},i},k)} : k \in \{1, \ldots, K\}\right\}$ for all $i \in \mathfrak{I}_{\mathrm{E},n}$, with samples of the indicator variables $\left(c_i^{(n,1)}\right)_{i\in\mathfrak{I}_{\mathrm{A},n}\cup\mathfrak{I}_{\mathrm{E},n}}$, where $c_i^{(n,1)} = \tau_n^{-1}(i)$, and with samples of the parameters $\left(\boldsymbol{p}_c^{*(n,1)}\right)_{c\in\{c_i^{(n,1)}:i\in\mathfrak{I}_{\mathrm{A},n}\cup\mathfrak{I}_{\mathrm{E},n}\}}$, where $\boldsymbol{p}_c^{*(n,1)}$ is chosen arbitrarily from $\mathbb{R}^D$.

Sampling the States We sample the states from the approximation (see (4.25))

$$f_{\mathbf{x}_{i,N_{\mathrm{S},i}:n}|\mathbf{p}_i,\mathbf{y}_{i,N_{\mathrm{S},i}:n}}\left(\boldsymbol{x}_{i,N_{\mathrm{S},i}:n} \,\middle|\, \boldsymbol{p}_{c_i^{(n,k-1)}}^{*(n,k-1)}, \boldsymbol{y}_{i,N_{\mathrm{S},i}:n}\right) \approx \sum_{k'=1}^{K} W_i^{(n,k,k')} \delta_{\bar{\boldsymbol{x}}_{i,N_{\mathrm{S},i}:n}^{(n,k')}}\left(\boldsymbol{x}_{i,N_{\mathrm{S},i}:n}\right) \tag{4.42}$$

for $i \in \mathfrak{I}_{\mathrm{A},n}$, and

$$f_{\mathbf{x}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}}|\mathbf{p}_i,\mathbf{y}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}}}\left(\boldsymbol{x}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}} \,\middle|\, \boldsymbol{p}_{c_i^{(n,k-1)}}^{*(n,k-1)}, \boldsymbol{y}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}}\right) \approx \sum_{k'=1}^{K} W_i^{(n,k,k')} \delta_{\bar{\boldsymbol{x}}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}}^{(N_{\mathrm{E},i},k')}}\left(\boldsymbol{x}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}}\right) \tag{4.43}$$

for $i \in \mathfrak{I}_{\mathrm{E},n}$, with $W_i^{(n,k,k')}$ defined as follows. For $i \in \mathfrak{I}_{\mathrm{S},n}$, we define $W_i^{(n,k,k')}$ as in (4.26) with the time step index 1 replaced by $N_{\mathrm{S},i}$. For $i \in \mathfrak{I}_{\mathrm{A},n} \setminus \mathfrak{I}_{\mathrm{S},n}$, we define $W_i^{(n,k,k')}$ as in (4.27). Finally, for $i \in \mathfrak{I}_{\mathrm{E},n}$, we define $W_i^{(n,k,k')}$ as in (4.27) with the time step index n replaced by $N_{\mathrm{E},i}$. We will present procedures to sample from this approximation based on Gibbs sampling in Section 4.2.3.2 and based on the MH algorithm Section 4.2.3.3.

Sampling the Indicator Variables and Parameters For $i \in \{1, \ldots, |\mathfrak{I}_{\mathrm{A},n} \cup \mathfrak{I}_{\mathrm{E},n}|\}$, the conditional probability of $\mathsf{c}_{\tau_n(i)} = c$, with $c \in \left\{c_i^{(n,k-1)} : i \in \mathfrak{I}_{\mathrm{A},n} \cup \mathfrak{I}_{\mathrm{E},n}\right\}$, is given in (4.30) and the conditional probability of $\mathsf{c}_{\tau_n(i)} \notin \left\{c_i^{(n,k-1)} : i \in \mathfrak{I}_{\mathrm{A},n} \cup \mathfrak{I}_{\mathrm{E},n}\right\}$ is given in (4.31), with $\mathbf{c}_{\neg i} = \boldsymbol{c}_{\neg i}^{(n,k)}$ replaced by $\mathbf{c}_{\neg \tau_n(i)} = (\mathsf{c}_{\tau_n(1)}, \ldots, \mathsf{c}_{\tau_n(i-1)}, \mathsf{c}_{\tau_n(i+1)}, \ldots, \mathsf{c}_{\tau_n(|\mathfrak{I}_{\mathrm{A},n} \cup \mathfrak{I}_{\mathrm{E},n}|)}) = \left(c_{\tau_n(1)}^{(n,k)}, \ldots, c_{\tau_n(i-1)}^{(n,k)}, c_{\tau_n(i+1)}^{(n,k-1)}, \ldots, c_{\tau_n(|\mathfrak{I}_{\mathrm{A},n} \cup \mathfrak{I}_{\mathrm{E},n}|)}^{(n,k-1)}\right) = \boldsymbol{c}_{\neg \tau_n(i)}^{(n,k)}$, $c_{\max}^{(n,k-1)} = \max_{k' \in \{1,\ldots,k-1\}, i' \in \{1,\ldots,I\}} c_{i'}^{(n,k')}$ replaced by $c_{\max}^{(n,k-1)} = \max_{k' \in \{1,\ldots,k-1\}, i' \in \{1,\ldots,|\mathfrak{I}_{\mathrm{A},n} \cup \mathfrak{I}_{\mathrm{E},n}|\}} c_{\tau_n(i')}^{(n,k')}$, and further obvious changes for TSs of different lengths.

The parameters $\mathbf{p}_c^*$ for $c \in \left\{c_i^{(n,k)} : i \in \mathfrak{I}_{\mathrm{A},n} \cup \mathfrak{I}_{\mathrm{E},n}\right\}$ are sampled from their conditional pdf given the other random variables, similar to (4.33) with the obvious changes for TSs of different lengths.

The complete algorithm for TSs of different lengths using Gibbs sampling for the states is presented in Section 4.2.3.2, and the algorithm using the MH algorithm to sample the states in Section 4.2.3.3.

4.2.3.2 Gibbs Sampling Algorithm

The samples of the states $(\mathbf{x}_{i,N_{\mathrm{S},i}:n})_{i \in \mathfrak{I}_{\mathrm{A},n}}$ and $(\mathbf{x}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}})_{i \in \mathfrak{I}_{\mathrm{E},n}}$, indicator variables $(\mathsf{c}_i)_{i \in \mathfrak{I}_{\mathrm{A},n} \cup \mathfrak{I}_{\mathrm{E},n}}$, and parameters $(\mathbf{p}_{\mathsf{c}}^*)_{\mathsf{c} \in \{\mathsf{c}_i : i \in \mathfrak{I}_{\mathrm{A},n} \cup \mathfrak{I}_{\mathrm{E},n}\}}$ can be generated using Gibbs sampling (see Section 4.1.3.2). As in Section 4.1.3.2, the states are directly sampled from their approximated conditional pdf, that is, from (4.42) and (4.43), which requires the calculation of the weights $\left(W_i^{(n,k,k')}\right)_{k' \in \{1,\ldots,K\}}$ (see (4.26) and (4.27)) for all $i \in \mathfrak{I}_{\mathrm{A},n} \cup \mathfrak{I}_{\mathrm{E},n}$ at each iteration of the MCMC algorithm using the current samples. The pseudo-code for one iteration of the SMC algorithm is stated in Algorithm 4.10, with the MCMC algorithm stated in Algorithm 4.11. Note that Algorithm 4.10 includes the initialization of the starting TSs as they can start at any time step, that is, Algorithm 4.10 is equivalent to Algorithms 4.4 and 4.5 for TSs of equal length.

Algorithm 4.10 SMC algorithm using Gibbs sampling for a BNP state-space model with different lengths: iteration

Input: $\left(\left(\boldsymbol{x}_{i,N_{\mathrm{S},i}:n-1}^{(n-1,k)}\right)_{i\in\mathfrak{I}_{\mathrm{A},n}\setminus\mathfrak{I}_{\mathrm{S},n}}, \left(\bar{\boldsymbol{x}}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}}^{(N_{\mathrm{E},i},k)}\right)_{i\in\mathfrak{I}_{\mathrm{E},n}}, \left(c_i^{(n-1,k)}\right)_{i\in\mathfrak{I}_{\mathrm{A},n}\setminus\mathfrak{I}_{\mathrm{S},n}}, \left(\boldsymbol{p}_c^{*(n-1,k)}\right)_{c\in\{c_i^{(n-1,k)}:i\in\mathfrak{I}_{\mathrm{A},n}\setminus\mathfrak{I}_{\mathrm{S},n}\}}\right)_{k\in\{1,\ldots,K\}}, (\boldsymbol{y}_{i,N_{\mathrm{S},i}:n})_{i\in\mathfrak{I}_{\mathrm{A},n}}, (\boldsymbol{y}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}})_{i\in\mathfrak{I}_{\mathrm{E},n}}, K, g_1, g, \tau_n$

1: **for all** $i \in \mathfrak{I}_{\mathrm{S},n}$ **do**
2: **for all** $k = 1, \ldots, K$ **do**
3: set $c_i^{(0,k)} = \tau_n^{-1}(i)$
4: sample $\boldsymbol{p}_{c_i^{(0,k)}}^{*(0,k)}$ from $f_{\mathrm{G}}\left(\boldsymbol{p}_{c_i^{(0,k)}}^{*(0,k)}\right)$
5: sample $\bar{\boldsymbol{x}}_i^{(n,k)}$ from $g_1\left(\bar{\boldsymbol{x}}_i^{(n,k)} \middle| \boldsymbol{p}_{c_i^{(0,k)}}^{*(0,k)}, \boldsymbol{y}_{i,n}\right)$
6: **end for**
7: **end for**
8: **for all** $i \in \mathfrak{I}_{\mathrm{A},n} \setminus \mathfrak{I}_{\mathrm{S},n}$ **do**
9: **for all** $k = 1, \ldots, K$ **do**
10: sample $\bar{\boldsymbol{x}}_{i,n}^{(n,k)}$ from $g\left(\bar{\boldsymbol{x}}_{i,n}^{(n,k)} \middle| \boldsymbol{x}_{i,N_{\mathrm{S},i}:n-1}^{(n-1,k)}, \boldsymbol{p}_{c_i^{(n-1,k)}}^{*(n-1,k)}, \boldsymbol{y}_{i,n}\right)$
11: set $\bar{\boldsymbol{x}}_{i,N_{\mathrm{S},i}:n}^{(n,k)} = \left(\boldsymbol{x}_{i,N_{\mathrm{S},i}:n-1}^{(n-1,k)}, \bar{\boldsymbol{x}}_{i,n}^{(n,k)}\right)$
12: **end for**
13: **end for**
14: sample $\left(\left(\boldsymbol{x}_{i,N_{\mathrm{S},i}:n}^{(n,k)}\right)_{i\in\mathfrak{I}_{\mathrm{A},n}}, \left(\boldsymbol{x}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}}^{(n,k)}\right)_{i\in\mathfrak{I}_{\mathrm{E},n}}, \left(c_i^{(n,k)}\right)_{i\in\mathfrak{I}_{\mathrm{A},n}\cup\mathfrak{I}_{\mathrm{E},n}}, \left(\boldsymbol{p}_c^{*(n,k)}\right)_{c\in\{c_i^{(n,k)}:i\in\mathfrak{I}_{\mathrm{A},n}\cup\mathfrak{I}_{\mathrm{E},n}\}}\right)_{k\in\{1,\ldots,K\}}$ using Algorithm 4.11 with input $\left(\left(\bar{\boldsymbol{x}}_{i,N_{\mathrm{S},i}:n}^{(n,k)}\right)_{i\in\mathfrak{I}_{\mathrm{A},n}}, \left(\bar{\boldsymbol{x}}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}}^{(N_{\mathrm{E},i},k)}\right)_{i\in\mathfrak{I}_{\mathrm{E},n}}\right)_{k\in\{1,\ldots,K\}}$, $(\boldsymbol{y}_{i,N_{\mathrm{S},i}:n})_{i\in\mathfrak{I}_{\mathrm{A},n}}, (\boldsymbol{y}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}})_{i\in\mathfrak{I}_{\mathrm{E},n}}, K, g_1, g, \tau_n$

Output: $\left(\left(\boldsymbol{x}_{i,N_{\mathrm{S},i}:n}^{(n,k)}\right)_{i\in\mathfrak{I}_{\mathrm{A},n}}, \left(\boldsymbol{x}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}}^{(n,k)}\right)_{i\in\mathfrak{I}_{\mathrm{E},n}}, \left(\bar{\boldsymbol{x}}_{i,N_{\mathrm{S},i}:n}^{(n,k)}\right)_{i\in\mathfrak{I}_{\mathrm{A},n}}, \left(\bar{\boldsymbol{x}}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}}^{(N_{\mathrm{E},i},k)}\right)_{i\in\mathfrak{I}_{\mathrm{E},n}}, \left(c_i^{(n,k)}\right)_{i\in\mathfrak{I}_{\mathrm{A},n}\cup\mathfrak{I}_{\mathrm{E},n}}, \left(\boldsymbol{p}_c^{*(n,k)}\right)_{c\in\{c_i^{(n,k)}:i\in\mathfrak{I}_{\mathrm{A},n}\cup\mathfrak{I}_{\mathrm{E},n}\}}\right)_{k\in\{1,\ldots,K\}}$

Algorithm 4.11 SMC algorithm using Gibbs sampling for a BNP state-space model with different lengths: MCMC part

Input: $\left(\left(\bar{\boldsymbol{x}}_{i,N_{\mathrm{S},i}:n}^{(n,k)}\right)_{i\in\mathfrak{I}_{\mathrm{A},n}}, \left(\bar{\boldsymbol{x}}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}}^{(N_{\mathrm{E},i},k)}\right)_{i\in\mathfrak{I}_{\mathrm{E},n}}\right)_{k\in\{1,\dots,K\}}$,
$(\boldsymbol{y}_{i,N_{\mathrm{S},i}:n})_{i\in\mathfrak{I}_{\mathrm{A},n}}, (\boldsymbol{y}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}})_{i\in\mathfrak{I}_{\mathrm{E},n}}, K, g_1, g, \tau_n$

for all $i \in \mathfrak{I}_{\mathrm{A},n}$ **do**
  initialize $\boldsymbol{x}_{i,N_{\mathrm{S},i}:n}^{(n,1)}$ arbitrarily from $\left\{\bar{\boldsymbol{x}}_{i,N_{\mathrm{S},i}:n}^{(n,k)} : k \in \{1,\dots,K\}\right\}$
  set $c_i^{(n,1)} = \tau_n^{-1}(i)$
  initialize $\boldsymbol{p}_{c_i^{(n,1)}}^{*(n,1)}$ arbitrarily from $\mathbb{R}^D$
end for
for all $i \in \mathfrak{I}_{\mathrm{E},n}$ **do**
  initialize $\boldsymbol{x}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}}^{(n,1)}$ arbitrarily from $\left\{\bar{\boldsymbol{x}}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}}^{(N_{\mathrm{E},i},k)} : k \in \{1,\dots,K\}\right\}$
  set $c_i^{(n,1)} = \tau_n^{-1}(i)$
  initialize $\boldsymbol{p}_{c_i^{(n,1)}}^{*(n,1)}$ arbitrarily from $\mathbb{R}^D$
end for
set $c_{\max}^{(n,1)} = |\mathfrak{I}_{\mathrm{A},n} \cup \mathfrak{I}_{\mathrm{E},n}|$
for all $k = 2,\dots,K$ **do**
  for all $i \in \mathfrak{I}_{\mathrm{A},n}$ **do**
    for all $k' = 1,\dots,K$ **do**
      calculate $W_i^{(n,k,k')}$ according to (4.26) or (4.27) with the necessary changes
    end for
    sample $\boldsymbol{x}_{i,N_{\mathrm{S},i}:n}^{(n,k)}$ from (4.42)
  end for
  for all $i \in \mathfrak{I}_{\mathrm{E},n}$ **do**
    for all $k' = 1,\dots,K$ **do**
      calculate $W_i^{(n,k,k')}$ according to (4.26) or (4.27) with the necessary changes
    end for
    sample $\boldsymbol{x}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}}^{(n,k)}$ from (4.43)
  end for
  for all $i = 1,\dots,|\mathfrak{I}_{\mathrm{A},n} \cup \mathfrak{I}_{\mathrm{E},n}|$ **do**
    sample $c_{\tau_n(i)}^{(n,k)} = c$ with probability $b_{\tau_n(i),c}^{(n,k)}$ (see (4.30) and (4.31))
      for all $c \in \left\{0, c_{\tau_n(1)}^{(n,k)},\dots,c_{\tau_n(i-1)}^{(n,k)}, c_{\tau_n(i+1)}^{(n,k-1)},\dots,c_{\tau_n(|\mathfrak{I}_{\mathrm{A},n}\cup\mathfrak{I}_{\mathrm{E},n}|)}^{(n,k-1)}\right\}$
    if $c_{\tau_n(i)}^{(n,k)} = 0$
      set $c_{\tau_n(i)}^{(n,k)} = \max\left\{\max_{i'\in\{1,\dots,i-1\}} c_{\tau_n(i')}^{(n,k)}, c_{\max}^{(n,k-1)}\right\} + 1$
      sample $\boldsymbol{p}_{c_{\tau_n(i)}^{(n,k)}}^{*(n,k-1)}$ from (4.33) with the necessary changes
  end for
  set $c_{\max}^{(n,k)} = \max\left\{\max_{i'\in\{1,\dots,|\mathfrak{I}_{\mathrm{A},n}\cup\mathfrak{I}_{\mathrm{E},n}|\}} c_{\tau_n(i')}^{(n,k)}, c_{\max}^{(n,k-1)}\right\}$
  for all $c \in \left\{c_i^{(n,k)} : i \in \mathfrak{I}_{\mathrm{A},n} \cup \mathfrak{I}_{\mathrm{E},n}\right\}$ **do**
    sample $\boldsymbol{p}_c^{*(n,k)}$ from (4.33) with the necessary changes
  end for
end for

Output: $\left(\left(\boldsymbol{x}_{i,N_{\mathrm{S},i}:n}^{(n,k)}\right)_{i\in\mathfrak{I}_{\mathrm{A},n}}, \left(\boldsymbol{x}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}}^{(n,k)}\right)_{i\in\mathfrak{I}_{\mathrm{E},n}}, \left(c_i^{(n,k)}\right)_{i\in\mathfrak{I}_{\mathrm{A},n}\cup\mathfrak{I}_{\mathrm{E},n}}, \left(\boldsymbol{p}_c^{*(n,k)}\right)_{c\in\{c_i^{(n,k)}:i\in\mathfrak{I}_{\mathrm{A},n}\cup\mathfrak{I}_{\mathrm{E},n}\}}\right)_{k\in\{1,\dots,K\}}$

4.2.3.3 Metropolis-Hastings Algorithm

As with TSs of equal length, we can also use an MH algorithm instead of Gibbs sampling (see Section 4.1.3.3). For this, we use the MH algorithm discussed in Section 4.1.3.3 with the obvious changes for TSs of different lengths. The algorithm for one iteration is outlined in Algorithm 4.12, with the MCMC algorithm stated in Algorithm 4.13.

4.2.3.4 Sufficient Statistics

We can simplify the algorithms for TSs of different lengths introduced above by using sufficient statistics as discussed in Section 4.1.3.4, with obvious changes due to the different TS lengths.

Algorithm 4.12 SMC algorithm using MH for a BNP state-space model with different lengths: iteration

Input: $\left(\left(\boldsymbol{x}_{i,N_{\mathrm{S},i}:n-1}^{(n-1,k)}\right)_{i\in\mathfrak{I}_{\mathrm{A},n}\setminus\mathfrak{I}_{\mathrm{S},n}}, \left(\boldsymbol{p}_{c}^{*(n-1,k)}\right)_{c\in\{c_i^{(n-1,k)}:i\in\mathfrak{I}_{\mathrm{A},n}\setminus\mathfrak{I}_{\mathrm{S},n}\}}\right)_{k\in\{1,\dots,K\}}$, $\left(\left(\bar{\boldsymbol{x}}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}}^{(N_{\mathrm{E},i},k)}, W_i^{\prime(N_{\mathrm{E},i},k)}\right)_{i\in\mathfrak{I}_{\mathrm{E},n}}\right)_{k\in\{1,\dots,K\}}$, $(\boldsymbol{y}_{i,N_{\mathrm{S},i}:n})_{i\in\mathfrak{I}_{\mathrm{A},n}}$, $(\boldsymbol{y}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}})_{i\in\mathfrak{I}_{\mathrm{E},n}}$, K, g_1, g, τ_n

for all $i \in \mathfrak{I}_{\mathrm{S},n}$ **do**
  for all $k = 1, \dots, K$ **do**
    set $c_i^{(0,k)} = \tau_n^{-1}(i)$
    sample $\boldsymbol{p}_{c_i^{(0,k)}}^{*(0,k)}$ from $f_{\mathrm{G}}\left(\boldsymbol{p}_{c_i^{(0,k)}}^{*(0,k)}\right)$
    sample $\bar{\boldsymbol{x}}_i^{(n,k)}$ from $g_1\left(\bar{\boldsymbol{x}}_i^{(n,k)} \middle| \boldsymbol{p}_{c_i^{(0,k)}}^{*(0,k)}, \boldsymbol{y}_{i,n}\right)$
    calculate $W_i^{\prime(n,k)}$ according to (4.35) with the necessary changes
  end for
end for
for all $i \in \mathfrak{I}_{\mathrm{A},n} \setminus \mathfrak{I}_{\mathrm{S},n}$ **do**
  for all $k = 1, \dots, K$ **do**
    sample $\bar{\boldsymbol{x}}_{i,n}^{(n,k)}$ from $g\left(\bar{\boldsymbol{x}}_{i,n}^{(n,k)} \middle| \boldsymbol{x}_{i,n-1}^{(n-1,k)}, \boldsymbol{p}_{c_i^{(n-1,k)}}^{*(n-1,k)}, \boldsymbol{y}_{i,n}\right)$
    set $\bar{\boldsymbol{x}}_{i,N_{\mathrm{S},i}:n}^{(n,k)} = \left(\boldsymbol{x}_{i,N_{\mathrm{S},i}:n-1}^{(n-1,k)}, \bar{\boldsymbol{x}}_{i,n}^{(n,k)}\right)$
    calculate $W_i^{\prime(n,k)}$ according to (4.36) with the necessary changes
  end for
end for
sample $\left(\left(\boldsymbol{x}_{i,N_{\mathrm{S},i}:n}^{(n,k)}\right)_{i\in\mathfrak{I}_{\mathrm{A},n}}, \left(\boldsymbol{x}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}}^{(n,k)}\right)_{i\in\mathfrak{I}_{\mathrm{E},n}}, \left(c_i^{(n,k)}\right)_{i\in\mathfrak{I}_{\mathrm{A},n}\cup\mathfrak{I}_{\mathrm{E},n}}, \left(\boldsymbol{p}_{c}^{*(n,k)}\right)_{c\in\{c_i^{(n,k)}:i\in\mathfrak{I}_{\mathrm{A},n}\cup\mathfrak{I}_{\mathrm{E},n}\}}\right)_{k\in\{1,\dots,K\}}$ using Algorithm 4.13 with input $\left(\left(\bar{\boldsymbol{x}}_{i,N_{\mathrm{S},i}:n}^{(n,k)}, W_i^{\prime(n,k)}\right)_{i\in\mathfrak{I}_{\mathrm{A},n}}, \left(\bar{\boldsymbol{x}}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}}^{(N_{\mathrm{E},i},k)}, W_i^{\prime(N_{\mathrm{E},i},k)}\right)_{i\in\mathfrak{I}_{\mathrm{E},n}}\right)_{k\in\{1,\dots,K\}}$, $(\boldsymbol{y}_{i,N_{\mathrm{S},i}:n})_{i\in\mathfrak{I}_{\mathrm{A},n}}$, $(\boldsymbol{y}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}})_{i\in\mathfrak{I}_{\mathrm{E},n}}, K, g_1, g, \tau_n$

Output: $\left(\left(\boldsymbol{x}_{i,N_{\mathrm{S},i}:n}^{(n,k)}\right)_{i\in\mathfrak{I}_{\mathrm{A},n}}, \left(\boldsymbol{x}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}}^{(n,k)}\right)_{i\in\mathfrak{I}_{\mathrm{E},n}}, \left(\bar{\boldsymbol{x}}_{i,N_{\mathrm{S},i}:n}^{(n,k)}, W_i^{\prime(n,k)}\right)_{i\in\mathfrak{I}_{\mathrm{A},n}}, \left(\bar{\boldsymbol{x}}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}}^{(N_{\mathrm{E},i},k)}, W_i^{\prime(N_{\mathrm{E},i},k)}\right)_{i\in\mathfrak{I}_{\mathrm{E},n}}, \left(c_i^{(n,k)}\right)_{i\in\mathfrak{I}_{\mathrm{A},n}\cup\mathfrak{I}_{\mathrm{E},n}}, \left(\boldsymbol{p}_{c}^{*(n,k)}\right)_{c\in\{c_i^{(n,k)}:i\in\mathfrak{I}_{\mathrm{A},n}\cup\mathfrak{I}_{\mathrm{E},n}\}}\right)_{k\in\{1,\dots,K\}}$

Algorithm 4.13 SMC algorithm using MH for a BNP state-space model with different lengths: MCMC part

Input: $\left(\left(\bar{\boldsymbol{x}}_{i,N_{\mathrm{S},i}:n}^{(n,k)}, W_i'^{(n,k)}\right)_{i\in\mathfrak{I}_{\mathrm{A},n}}, \left(\bar{\boldsymbol{x}}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}}^{(N_{\mathrm{E},i},k)}, W_i'^{(N_{\mathrm{E},i},k)}\right)_{i\in\mathfrak{I}_{\mathrm{E},n}}\right)_{k\in\{1,\ldots,K\}}, (\boldsymbol{y}_{i,N_{\mathrm{S},i}:n})_{i\in\mathfrak{I}_{\mathrm{A},n}}, (\boldsymbol{y}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}})_{i\in\mathfrak{I}_{\mathrm{E},n}}, K, g_1, g, \tau_n$

for all $i \in \mathfrak{I}_{\mathrm{A},n}$ **do**
  initialize $\boldsymbol{x}_{i,N_{\mathrm{S},i}:n}^{(n,1)}$ arbitrarily from $\left\{\bar{\boldsymbol{x}}_{i,N_{\mathrm{S},i}:n}^{(n,k)} : k \in \{1,\ldots,K\}\right\}$
  set $c_i^{(n,1)} = \tau_n^{-1}(i)$
  initialize $\boldsymbol{p}_{c_i^{(n,1)}}^{*(n,1)}$ arbitrarily from $\mathbb{R}^D$
end for
for all $i \in \mathfrak{I}_{\mathrm{E},n}$ **do**
  initialize $\boldsymbol{x}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}}^{(n,1)}$ arbitrarily from $\left\{\bar{\boldsymbol{x}}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}}^{(N_{\mathrm{E},i},k)} : k \in \{1,\ldots,K\}\right\}$
  set $c_i^{(n,1)} = \tau_n^{-1}(i)$
  initialize $\boldsymbol{p}_{c_i^{(n,1)}}^{*(n,1)}$ arbitrarily from $\mathbb{R}^D$
end for
set $c_{\max}^{(n,1)} = |\mathfrak{I}_{\mathrm{A},n} \cup \mathfrak{I}_{\mathrm{E},n}|$
for all $k = 2,\ldots,K$ **do**
  for all $i \in \mathfrak{I}_{\mathrm{A},n}$ **do**
    sample $\boldsymbol{x}_{i,N_{\mathrm{S},i}:n}^{\prime(n,k)} = \bar{\boldsymbol{x}}_{i,N_{\mathrm{S},i}:n}^{(n,k')}$ with probability $W_i'^{(n,k')}$ for all $k' \in \{1,\ldots,K\}$
    set $\boldsymbol{x}_{i,N_{\mathrm{S},i}:n}^{(n,k)} = \boldsymbol{x}_{i,N_{\mathrm{S},i}:n}^{\prime(n,k)}$ with probability $P_{\mathrm{a},i}^{(n,k)}$ (see (4.37))
    else set $\boldsymbol{x}_{i,N_{\mathrm{S},i}:n}^{(n,k)} = \boldsymbol{x}_{i,N_{\mathrm{S},i}:n}^{(n,k-1)}$
  end for
  for all $i \in \mathfrak{I}_{\mathrm{E},n}$ **do**
    sample $\boldsymbol{x}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}}^{\prime(n,k)} = \bar{\boldsymbol{x}}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}}^{(N_{\mathrm{E},i},k')}$ with probability $W_i'^{(N_{\mathrm{E},i},k')}$ for all $k' \in \{1,\ldots,K\}$
    set $\boldsymbol{x}_{i,N_{\mathrm{S},i}:N_{\mathrm{E}}}^{(n,k)} = \boldsymbol{x}_{i,N_{\mathrm{S},i}:N_{\mathrm{E}}}^{\prime(n,k)}$ with probability $P_{\mathrm{a},i}^{(n,k)}$ (see (4.37))
    else set $\boldsymbol{x}_{i,N_{\mathrm{S},i}:N_{\mathrm{E}}}^{(n,k)} = \boldsymbol{x}_{i,N_{\mathrm{S},i}:N_{\mathrm{E}}}^{(n,k-1)}$
  end for
  for all $i = 1,\ldots,|\mathfrak{I}_{\mathrm{A},n} \cup \mathfrak{I}_{\mathrm{E},n}|$ **do**
    sample $c_{\tau_n(i)}^{(n,k)} = c$ with probability $b_{\tau_n(i),c}^{(n,k)}$ (see (4.30) and (4.31))
      for all $c \in \left\{0, c_{\tau_n(1)}^{(n,k)},\ldots,c_{\tau_n(i-1)}^{(n,k)}, c_{\tau_n(i+1)}^{(n,k-1)},\ldots,c_{\tau_n(|\mathfrak{I}_{\mathrm{A},n}\cup\mathfrak{I}_{\mathrm{E},n}|)}^{(n,k-1)}\right\}$
      set $c_{\tau_n(i)}^{(n,k)} = \max\left\{\max_{i'\in\{1,\ldots,i-1\}} c_{\tau_n(i')}^{(n,k)}, c_{\max}^{(n,k-1)}\right\} + 1$
      sample $\boldsymbol{p}_{c_{\tau_n(i)}^{(n,k)}}^{*(n,k-1)}$ from (4.33) with the necessary changes
  end for
  set $c_{\max}^{(n,k)} = \max\left\{\max_{i'\in\{1,\ldots,|\mathfrak{I}_{\mathrm{A},n}\cup\mathfrak{I}_{\mathrm{E},n}|\}} c_{\tau_n(i')}^{(n,k)}, c_{\max}^{(n,k-1)}\right\}$
  for all $c \in \left\{c_i^{(n,k)} : i \in \mathfrak{I}_{\mathrm{A},n} \cup \mathfrak{I}_{\mathrm{E},n}\right\}$ **do**
    sample $\boldsymbol{p}_{c,n}^{*(k)}$ from (4.33) with the necessary changes
  end for
end for

Output: $\left(\left(\boldsymbol{x}_{i,N_{\mathrm{S},i}:n}^{(n,k)}\right)_{i\in\mathfrak{I}_{\mathrm{A},n}}, \left(\boldsymbol{x}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}}^{(n,k)}\right)_{i\in\mathfrak{I}_{\mathrm{E},n}}, \left(c_i^{(n,k)}\right)_{i\in\mathfrak{I}_{\mathrm{A},n}\cup\mathfrak{I}_{\mathrm{E},n}}, \left(\boldsymbol{p}_c^{*(n,k)}\right)_{c\in\{c_i^{(n,k)}:i\in\mathfrak{I}_{\mathrm{A},n}\cup\mathfrak{I}_{\mathrm{E},n}\}}\right)_{k\in\{1,\ldots,K\}}$

Chapter 5

Application to Extended Target Tracking

In our context, target tracking refers to the estimation of a sequence of target states based on a sequence of noisy measurements. Usually the target is a moving object that is observed by a sensor and we are interested in its kinematic state. In many scenarios it is assumed that a target gives rise to at most one measurement per time step. There are, however, applications where this assumption is not valid and each target may produce multiple measurements per time step. In these scenarios, it can be of interest to additionally estimate the size and orientation, which is often referred to as the extent, of the target. This is called extended target/object tracking. A recent overview of extended target tracking is provided in [3]. Figure 5.1 shows an example of the trajectory and the measurements of an elliptically shaped target.

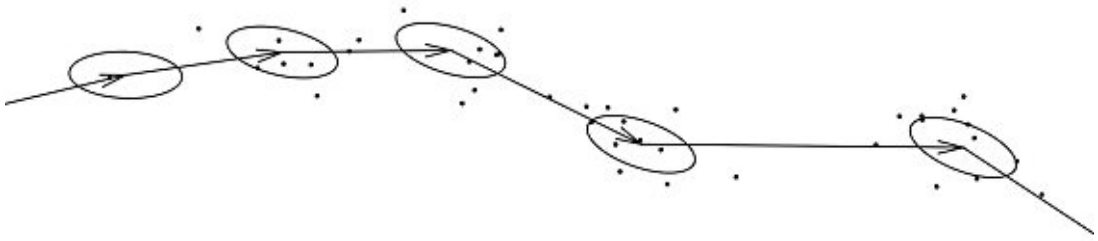

Fig. 5.1: *Example of the trajectory and the measurements of an elliptically shaped target.*

As extended target tracking can be applied in a wide range of applications, there are many different models for the target state, extent, and measurements. One common choice is the spatial model, described in [6, 39, 40], where a random number of measurements is assumed to be spatially distributed around the target's position. In this chapter, we adapt

the multiplicative noise model with known extent from [6], which is a special case of the spatial model, to our purposes.

In some scenarios there are several known target classes defining certain parameters of the targets that belong to these classes. As we usually do not know a priori to which class each target belongs, we can use joint tracking and classification algorithms [7,8] to infer it. In practice, however, it is not always the case that the target classes are known. Clustering of multiple targets for joint class identification and target classification is considered in [9], where it is required that the target has already been tracked. The algorithm in [9] takes a complete feature sequence as an input, which may depend on the complete estimated state sequences of the targets. A different approach is *joint tracking and clustering*. Here, we jointly infer the class structure and track the targets, in contrast to inferring the class structure and tracking the targets separately as in [9]. As the parameters, which are defined by the class, may influence the state estimate, joint tracking and clustering can improve state, parameter, and class assignment estimation. To the best of the author's knowledge, there is no previous work on joint tracking and clustering.

We start by statistically modeling the extended target tracking problem as a Bayesian nonparametric state-space model (see Chapter 4) in Section 5.1. Then, in Section 5.2, we discuss the application of the algorithms of Chapter 4, and evaluate the algorithms' convergence as well as the performance gain from joint tracking and clustering in Section 5.3.

5.1 Statistical Model

In order to statistically model the problem outlined above, we model the dynamic behavior and the measurements of each target. We consider a known number of targets $I \in \mathbb{N}$ where each target $i \in \{1, \ldots, I\}$ is observed in discrete time from time $N_{\mathrm{S},i} \in \mathbb{N}$ to time $N_{\mathrm{E},i} \in \mathbb{N}$ with $N_{\mathrm{S},i} \leq N_{\mathrm{E},i}$. For time steps $n \in \{N_{\mathrm{S},i}, \ldots, N_{\mathrm{E},i}\}$ the behavior of target i is described by a parameter dependent state-space model (see Section 4.1.1) as illustrated in Figure 5.2. Each target i has a kinematic state $\mathbf{x}_{i,n}$, composed of its position and velocity, for all times $n \in \{N_{\mathrm{S},i}, \ldots, N_{\mathrm{E},i}\}$. The temporal evolution of this kinematic state is described by the motion model defined in Section 5.1.1 below. At each time $n \in \{N_{\mathrm{S},i}, \ldots, N_{\mathrm{E},i}\}$, for each target i, we observe a random number $\mathsf{M}_{i,n}$ of random vectors $\mathbf{y}_{i,n,m}$ for $m \in \{1, \ldots, \mathsf{M}_{i,n}\}$, or equivalently the random sequence $\mathbf{y}_{i,n} = \left(\mathbf{y}_{i,n,1}, \ldots, \mathbf{y}_{i,n,\mathsf{M}_{i,n}}\right)$, which is described by the measurement model in Section 5.1.2. Furthermore, the motion and measurement models

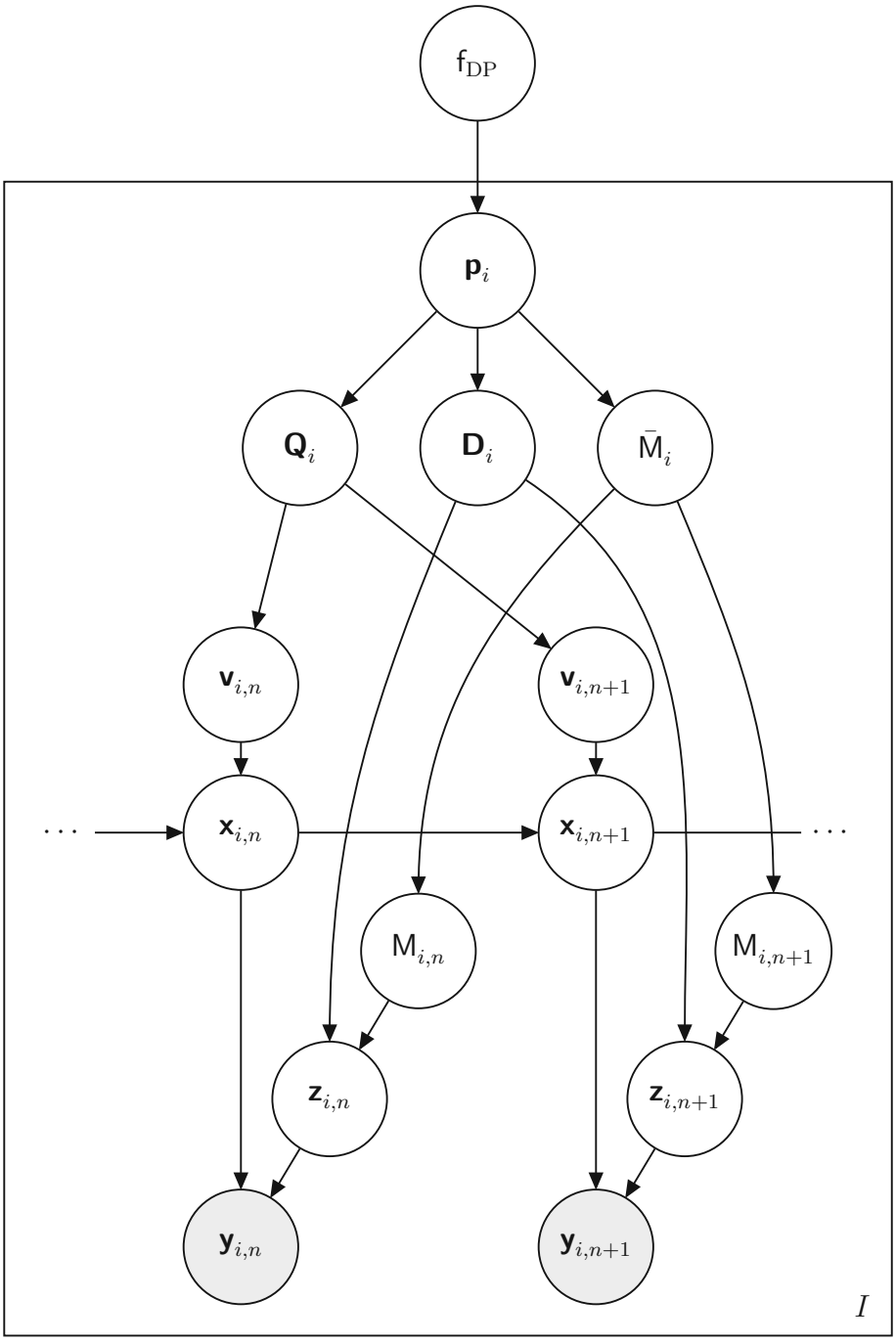

Fig. 5.2: *Bayesian network for our statistical model for extended target tracking. Observed random variables are represented by shaded disks.*

of each target i depend on a random parameter triple $\mathbf{p}_i = (\mathbf{D}_i, \mathbf{Q}_i, \bar{\mathsf{M}}_i)$, where $\mathbf{D}_i$ describes the target's extent, $\mathbf{Q}_i$ describes its dynamic behavior, and $\bar{\mathsf{M}}_i$ is the average number of measurements per time step associated with target i. Each target in our statistical model belongs to a class, which is described by a class indicator c_i. The targets in each class share the same parameter triple, that is, $\mathbf{p}_i = \mathbf{p}^*_{\mathsf{c}_i}$, where $\mathbf{p}^*_c = (\mathbf{D}^*_c, \mathbf{Q}^*_c, \bar{\mathsf{M}}^*_c)$ is the parameter triple of class c. The parameters and classes are modeled in Section 5.1.3.

5.1.1 Motion Model

Each target state $\mathbf{x}_{i,n} = [\mathsf{x}_{i,n,1}\ \mathsf{x}_{i,n,2}\ \dot{\mathsf{x}}_{i,n,1}\ \dot{\mathsf{x}}_{i,n,2}]^{\mathrm{T}} \in \mathbb{R}^4$ is composed of the position $[\mathsf{x}_{i,n,1}\ \mathsf{x}_{i,n,2}]^{\mathrm{T}}$ and velocity $[\dot{\mathsf{x}}_{i,n,1}\ \dot{\mathsf{x}}_{i,n,2}]^{\mathrm{T}}$. At time $N_{\mathrm{S},i}$, the state $\mathbf{x}_{i,N_{\mathrm{S},i}}$ is modeled as i.i.d. across i according to

$$\mathbf{x}_{i,N_{\mathrm{S},i}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{x}}), \tag{5.1}$$

where $\boldsymbol{\Sigma}_{\mathbf{x}} \in \mathbb{R}^{4\times 4}$ is an uninformative covariance matrix.

There are several ways to model the motion of the target [1]. We use the nearly constant velocity model, where the state at time $n \in \{N_{\mathrm{S},i}+1, \ldots, N_{\mathrm{E},i}\}$ is related to the previous state by

$$\mathbf{x}_{i,n} = \boldsymbol{F}\mathbf{x}_{i,n-1} + \mathbf{v}_{i,n}. \tag{5.2}$$

Here, $\mathbf{v}_{i,n} \in \mathbb{R}^4$ is called the driving noise and $\boldsymbol{F} \in \mathbb{R}^{4\times 4}$ is the state transition matrix given by

$$\boldsymbol{F} = \begin{bmatrix} \mathbf{I}_2 & \Delta T \mathbf{I}_2 \\ \mathbf{0}_{2\times 2} & \mathbf{I}_2 \end{bmatrix}, \tag{5.3}$$

with $\Delta T \in \mathbb{R}$ denoting the sampling period. Furthermore, we assume that the driving noise $\mathbf{v}_{i,n}$ given the random covariance matrix $\mathbf{Q}_i \in \mathbb{R}^{4\times 4}$, which will be explained below, is zero-mean Gaussian and i.i.d. across n and independent across i, that is,

$$\mathbf{v}_{i,n} \,|\, (\mathbf{Q}_i = \boldsymbol{Q}_i) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{Q}_i), \tag{5.4}$$

where

$$\mathbf{Q}_i = \begin{bmatrix} \mathsf{q}^2_{i,1}\mathbf{I}_2 & \mathbf{0}_{2\times 2} \\ \mathbf{0}_{2\times 2} & \mathsf{q}^2_{i,2}\mathbf{I}_2 \end{bmatrix} \tag{5.5}$$

is a random parameter with $\mathsf{q}^2_{i,1} \in \mathbb{R}_{>0}$ and $\mathsf{q}^2_{i,2} \in \mathbb{R}_{>0}$ being the position and velocity driving noise variances, respectively. The distribution of $\mathsf{q}^2_{i,1}$ and $\mathsf{q}^2_{i,2}$ will be defined in

Section 5.1.3. It follows from (5.2)–(5.5) that the state transition distribution given $\mathbf{Q}_i$ is also Gaussian, that is,

$$\mathbf{x}_{i,n} \,|\, (\mathbf{x}_{i,n-1} = \boldsymbol{x}_{i,n-1}, \mathbf{Q}_i = \boldsymbol{Q}_i) \sim \mathcal{N}(\boldsymbol{F}\boldsymbol{x}_{i,n-1}, \boldsymbol{Q}_i). \tag{5.6}$$

Note that $\mathbf{x}_{i,n}$ is independent across i as both $\mathbf{x}_{i,N_{\mathrm{S},i}}$ and $\mathbf{v}_{i,n}$ are independent across i.

By further defining the states as being conditionally independent of the parameters $\mathbf{D}_i$ and $\bar{\mathsf{M}}_i$ given $\mathbf{Q}_i$, we can now completely describe our motion model by (see (5.1))

$$\zeta\big(\boldsymbol{x}_{i,N_{\mathrm{S},i}} \,|\, \boldsymbol{p}_i\big) = f_{\mathbf{x}_{i,N_{\mathrm{S},i}}|\mathbf{p}_i}\big(\boldsymbol{x}_{i,N_{\mathrm{S},i}} \,|\, \boldsymbol{p}_i\big) = f_{\mathbf{x}_{i,N_{\mathrm{S},i}}}\big(\boldsymbol{x}_{i,N_{\mathrm{S},i}}\big) = \mathcal{N}\big(\boldsymbol{x}_{i,N_{\mathrm{S},i}}; \mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{x}}\big) \tag{5.7}$$

and (see (5.6))

$$\xi(\boldsymbol{x}_{i,n} \,|\, \boldsymbol{x}_{i,n-1}, \boldsymbol{p}_i) = f_{\mathbf{x}_{i,n}|\mathbf{x}_{i,n-1},\mathbf{p}_i}(\boldsymbol{x}_{i,n} \,|\, \boldsymbol{x}_{i,n-1}, \boldsymbol{p}_i) = \mathcal{N}(\boldsymbol{x}_{i,n}; \boldsymbol{F}\boldsymbol{x}_{i,n-1}, \boldsymbol{Q}_i) \tag{5.8}$$

for all $n \in \{N_{\mathrm{S},i} + 1, \ldots, N_{\mathrm{E},i}\}$.

An illustration of the motion model is given in Figure 5.3.

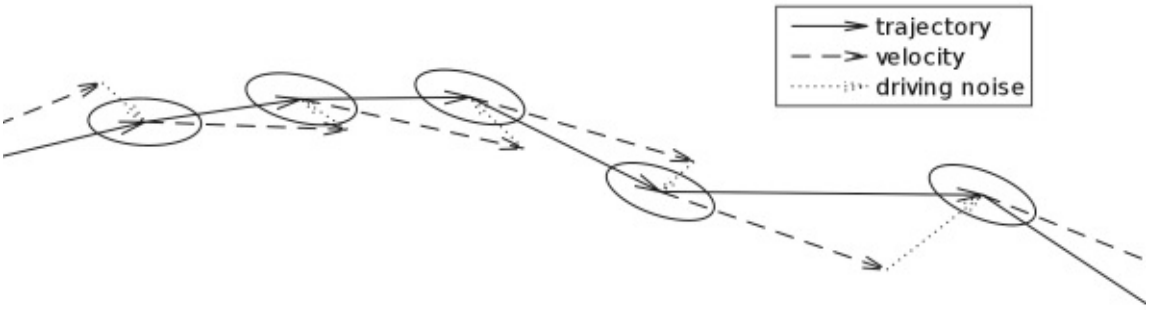

Fig. 5.3: *Illustration of the motion model.*

5.1.2 Measurement Model

For each target i, we observe at each time $n \in \{N_{\mathrm{S},i}, \ldots, N_{\mathrm{E},i}\}$ a random number $\mathsf{M}_{i,n} \in \mathbb{N}$ of measurements $\mathbf{y}_{i,n,m}$, $m \in \{1, \ldots, \mathsf{M}_{i,n}\}$, that, given a random parameter $\bar{\mathsf{M}}_i \in \mathbb{R}_{>0}$, is independent across i and i.i.d. across n according to a Poisson distribution

$$\mathsf{M}_{i,n} \,|\, (\bar{\mathsf{M}}_i = \bar{M}_i) \sim \mathrm{Pois}(\bar{M}_i). \tag{5.9}$$

Note that $\bar{M}_i$ is the average number of measurements of target i, that is, $\mathbb{E}(\mathsf{M}_{i,n}|\bar{\mathsf{M}}_i = \bar{M}_i) = \bar{M}_i$.

To model the measurements $\mathbf{y}_{i,n,m} \in \mathbb{R}^2$, we adapt the model of [41]. Ideally, in many applications, we would like to model the measurements $\mathbf{y}_{i,n,m}$ as noisy observations of random points on the target's extent. That is, each measurement $\mathbf{y}_{i,n,m}$ could be constructed

by randomly selecting a point on the target's extent and then adding measurement noise to this point.

In order to simplify the inference, we approximate this ideal construction by a Gaussian distribution [41]. More specifically, we define

$$\mathbf{y}_{i,n,m} = \boldsymbol{H}\mathbf{x}_{i,n} + \boldsymbol{R}(\mathbf{x}_{i,n})\mathbf{z}_{i,n,m}, \tag{5.10}$$

for $n \in \{N_{\mathrm{S},i}, \ldots, N_{\mathrm{E},i}\}$, where $\boldsymbol{H} \in \mathbb{R}^{2\times 4}$, $\boldsymbol{R}(\mathbf{x}_{i,n}) \in \mathbb{R}^{2\times 2}$, and $\mathbf{z}_{i,n,m} \in \mathbb{R}^2$ will be defined in detail presently. The first component of the sum in (5.10), $\boldsymbol{H}\mathbf{x}_{i,n}$, is used to center the measurements around the target's position, and the second component, $\boldsymbol{R}(\mathbf{x}_{i,n})\mathbf{z}_{i,n,m}$, is used to model the measurements spreading across the target's extent with added measurement noise. The measurement matrix

$$\boldsymbol{H} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{0}_{2\times 2} \end{bmatrix} \tag{5.11}$$

singles out the position of the target, $[\mathsf{x}_{i,n,1} \quad \mathsf{x}_{i,n,2}]^{\mathrm{T}}$, from its state $\mathbf{x}_{i,n} = [\mathsf{x}_{i,n,1} \ \mathsf{x}_{i,n,2} \ \dot{\mathsf{x}}_{i,n,1} \ \dot{\mathsf{x}}_{i,n,2}]^{\mathrm{T}}$. The pre-measurements $\mathbf{z}_{i,n,m}$ determine, up to a rotation by the matrix $\boldsymbol{R}(\mathbf{x}_{i,n})$, the shape and size of the target as well as the measurement noise. To model elliptical shapes, we choose $\mathbf{z}_{i,n,m}$ to be distributed according to

$$\mathbf{z}_{i,n,m} \,|\, (\mathbf{D}_i = \boldsymbol{D}_i) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{D}_i), \tag{5.12}$$

conditionally independent across i and i.i.d. across n and m given $\mathbf{D}_i$, with the random parameter

$$\mathbf{D}_i = \begin{bmatrix} \mathsf{d}_{i,1}^2 & 0 \\ 0 & \mathsf{d}_{i,2}^2 \end{bmatrix} \tag{5.13}$$

and the half axis lengths $\mathsf{d}_{i,1}, \mathsf{d}_{i,2} \in \mathbb{R}_{>0}$, which are determined by the target's shape and size, and the measurement noise. We summarize the pre-measurements $\mathbf{z}_{i,n,m}$ for target i at time n in the random sequence

$$\mathbf{z}_{i,n} = \left(\mathbf{z}_{i,n,1}, \ldots, \mathbf{z}_{i,n,\mathsf{M}_{i,n}}\right). \tag{5.14}$$

To align the target's extent with the direction of the target's velocity $[\dot{\mathsf{x}}_{i,n,1} \ \dot{\mathsf{x}}_{i,n,2}]^{\mathrm{T}}$, we define the rotation matrix

$$\boldsymbol{R}(\boldsymbol{x}_{i,n}) = \frac{1}{\sqrt{\dot{x}_{i,n,1}^2 + \dot{x}_{i,n,2}^2}} \begin{bmatrix} \dot{x}_{i,n,1} & -\dot{x}_{i,n,2} \\ \dot{x}_{i,n,2} & \dot{x}_{i,n,1} \end{bmatrix}. \tag{5.15}$$

That is, $\boldsymbol{R}(\mathsf{x}_{i,n})$ rotates a two-dimensional vector counterclockwise by the angle $\angle[\dot{\mathsf{x}}_{i,n,1}\ \dot{\mathsf{x}}_{i,n,2}]^{\mathrm{T}} \in (0, 2\pi]$. Therefore, this matrix aligns the first half axis with the direction of the target's movement. It follows that $\mathsf{y}_{i,n,m}$ given $\mathsf{x}_{i,n}$ and D_i is distributed according to

$$\mathsf{y}_{i,n,m} \,|\, (\mathsf{x}_{i,n} = \boldsymbol{x}_{i,n}, \mathsf{D}_i = \boldsymbol{D}_i) \sim \mathcal{N}(\boldsymbol{H}\boldsymbol{x}_{i,n}, \boldsymbol{R}(\boldsymbol{x}_{i,n})\boldsymbol{D}_i\boldsymbol{R}(\boldsymbol{x}_{i,n})^{\mathrm{T}}), \tag{5.16}$$

independent across i and n, and i.i.d. across m. The overall observation of the state-space model for target i at time n is then given the measurement sequence

$$\mathsf{y}_{i,n} = \left(\mathsf{y}_{i,n,1}, \ldots, \mathsf{y}_{i,n,\mathsf{M}_{i,n}}\right). \tag{5.17}$$

Note that the number of vectors in $\mathsf{z}_{i,n}$ and also in $\mathsf{y}_{i,n}$ is $\mathsf{M}_{i,n}$, and therefore random. We could, for example, treat $\mathsf{z}_{i,n}$ and $\mathsf{y}_{i,n}$ rigorously using random finite sets [42]. However, for the purpose of this text, we will treat $\mathsf{z}_{i,n}$ and $\mathsf{y}_{i,n}$ as mixed random variables, combining the continuous random variables $\mathsf{z}_{i,n,m}$ and $\mathsf{y}_{i,nm}$ for $m \in \{1, \ldots, \mathsf{M}_{i,n}\}$ with the discrete random variable $\mathsf{M}_{i,n}$. Thus, we can describe our measurement model by the mixed density (see Section 4.1.1)

$$\chi(\boldsymbol{y}_{i,n} \,|\, \boldsymbol{x}_{i,n}, \boldsymbol{p}_i) = v_{\mathsf{y}_{i,n}|\mathsf{x}_{i,n},\mathsf{p}_i}(\boldsymbol{y}_{i,n} \,|\, \boldsymbol{x}_{i,n}, \boldsymbol{p}_i) = v_{\mathsf{y}_{i,n}|\mathsf{x}_{i,n},\mathsf{D}_i,\mathsf{Q}_i,\bar{\mathsf{M}}_i}(\boldsymbol{y}_{i,n} \,|\, \boldsymbol{x}_{i,n}, \boldsymbol{D}_i, \boldsymbol{Q}_i, \bar{M}_i) \tag{5.18}$$

for all $n \in \{N_{\mathrm{S},i}, \ldots, N_{\mathrm{E},i}\}$. By defining the measurements $\boldsymbol{y}_{i,n}$ as conditionally independent of Q_i given $\mathsf{x}_{i,n}$, D_i, and $\bar{\mathsf{M}}_i$, we can simplify (5.18) to

$$\begin{aligned}
\chi(\boldsymbol{y}_{i,n} \,|\, \boldsymbol{x}_{i,n}, \boldsymbol{p}_i) &= \sum_{M'=0}^{\infty} v_{\mathsf{y}_{i,n},\mathsf{M}_{i,n}|\mathsf{x}_{i,n},\mathsf{D}_i,\bar{\mathsf{M}}_i}(\boldsymbol{y}_{i,n}, M' \,|\, \boldsymbol{x}_{i,n}, \boldsymbol{D}_i, \bar{M}_i) \\
&= v_{\mathsf{y}_{i,n},\mathsf{M}_{i,n}|\mathsf{x}_{i,n},\mathsf{D}_i,\bar{\mathsf{M}}_i}(\boldsymbol{y}_{i,n}, M_{i,n} \,|\, \boldsymbol{x}_{i,n}, \boldsymbol{D}_i, \bar{M}_i) \\
&= f_{\mathsf{y}_{i,n}|\mathsf{M}_{i,n},\mathsf{x}_{i,n},\mathsf{D}_i,\bar{\mathsf{M}}_i}(\boldsymbol{y}_{i,n} \,|\, M_{i,n}, \boldsymbol{x}_{i,n}, \boldsymbol{D}_i, \bar{M}_i) p_{\mathsf{M}_{i,n}|\mathsf{x}_{i,n},\mathsf{D}_i,\bar{\mathsf{M}}_i}(M_{i,n} \,|\, \boldsymbol{x}_{i,n}, \boldsymbol{D}_i, \bar{M}_i),
\end{aligned} \tag{5.19}$$

where we have used that $\mathsf{y}_{i,n}$ defines $\mathsf{M}_{i,n}$. By further defining $\mathsf{y}_{i,n}$ as conditionally independent of $\bar{\mathsf{M}}_i$ given $\mathsf{M}_{i,n}$, and $\mathsf{M}_{i,n}$ as conditionally independent of $\mathsf{x}_{i,n}$ and D_i given $\bar{\mathsf{M}}_i$, it follows that

$$\chi(\boldsymbol{y}_{i,n} \,|\, \boldsymbol{x}_{i,n}, \boldsymbol{p}_i) = f_{\mathsf{y}_{i,n}|\mathsf{M}_{i,n},\mathsf{x}_{i,n},\mathsf{D}_i}(\boldsymbol{y}_{i,n} \,|\, M_{i,n}, \boldsymbol{x}_{i,n}, \boldsymbol{D}_i) p_{\mathsf{M}_{i,n}|\bar{\mathsf{M}}_i}(M_{i,n} \,|\, \bar{M}_i). \tag{5.20}$$

Inserting (5.9) and (5.16) into (5.20), we finally obtain

$$\chi(\boldsymbol{y}_{i,n} \,|\, \boldsymbol{x}_{i,n}, \boldsymbol{p}_i) = \left(\prod_{m=1}^{M_{i,n}} \mathcal{N}(\boldsymbol{y}_{i,n,m}; \boldsymbol{H}\boldsymbol{x}_{i,n}, \boldsymbol{R}(\boldsymbol{x}_{i,n})\boldsymbol{D}_i\boldsymbol{R}(\boldsymbol{x}_{i,n})^{\mathrm{T}}) \right) \mathrm{Pois}(M_{i,n}; \bar{M}_i) \tag{5.21}$$

Fig. 5.4: *Example of an elliptical target extent and corresponding measurements.*

with $\boldsymbol{y}_{i,n} = \left(\boldsymbol{y}_{i,n,1}, \ldots, \boldsymbol{y}_{i,n,M_{i,n}}\right)$. Figure 5.4 shows an example of an elliptical target extent and example measurements.

Our model differs from [41] in that it assumes $\mathbf{D}_i$ as a time-constant random parameter and not as part of the target state. The underlying assumption of our model is that the targets are rigid and can only change their orientations, but not their shapes, over time.

5.1.3 Parameters

The motion and measurement models in Sections 5.1.1 and 5.1.2 are parametrized by the random parameter triple $\mathbf{p}_i = (\mathbf{D}_i, \mathbf{Q}_i, \bar{\mathsf{M}}_i) \in \mathbb{R}^{2\times 2} \times \mathbb{R}^{4\times 4} \times \mathbb{R}$, which we will use for clustering the targets. We will use a Dirichlet process to define the joint distribution of all parameter triples $\mathbf{p}_{1:I} = (\mathbf{p}_1, \ldots, \mathbf{p}_I)$. However, let us first discuss the marginal distribution of a single parameter triple $\mathbf{p}_i$ before we introduce this Dirichlet process.

We define the elements of $\mathbf{p}_i$, that is, $\mathbf{D}_i$, $\mathbf{Q}_i$, and $\bar{\mathsf{M}}_i$, as mutually independent and with their marginal distributions defined as follows. We define $\mathbf{D}_i$ as (see (5.13))

$$\mathbf{D}_i = \begin{bmatrix} \mathsf{d}_{i,1}^2 & 0 \\ 0 & \mathsf{d}_{i,2}^2 \end{bmatrix}, \tag{5.22}$$

where $\mathsf{d}_{i,1}^2$ and $\mathsf{d}_{i,2}^2$ are independent and distributed according to

$$\mathsf{d}_{i,1}^2 \sim \Gamma^{-1}(\alpha_{\mathrm{d},1}, \beta_{\mathrm{d},1}), \tag{5.23}$$

$$\mathsf{d}_{i,2}^2 \sim \Gamma^{-1}(\alpha_{\mathrm{d},2}, \beta_{\mathrm{d},2}) \tag{5.24}$$

with the hyperparameters $\alpha_{\mathrm{d},1}, \beta_{\mathrm{d},1}, \alpha_{\mathrm{d},2}, \beta_{\mathrm{d},2} \in \mathbb{R}_{>0}$. Further, we define $\mathbf{Q}_i$ as (see (5.5))

$$\mathbf{Q}_i = \begin{bmatrix} \mathsf{q}_{i,1}^2 \mathbf{I}_2 & \mathbf{0}_{2\times 2} \\ \mathbf{0}_{2\times 2} & \mathsf{q}_{i,2}^2 \mathbf{I}_2 \end{bmatrix}, \tag{5.25}$$

where $\mathsf{q}_{i,1}^2$ and $\mathsf{q}_{i,2}^2$ are independent and distributed according to

$$\mathsf{q}_{i,1}^2 \sim \Gamma^{-1}(\alpha_{\mathrm{v},1}, \beta_{\mathrm{v},1}), \tag{5.26}$$

$$\mathsf{q}_{i,2}^2 \sim \Gamma^{-1}(\alpha_{\mathrm{v},2}, \beta_{\mathrm{v},2}) \tag{5.27}$$

with the hyperparameters $\alpha_{\mathrm{v},1}, \beta_{\mathrm{v},1}, \alpha_{\mathrm{v},2}, \beta_{\mathrm{v},2} \in \mathbb{R}_{>0}$. Finally, we define

$$\bar{\mathsf{M}}_i \sim \Gamma(\alpha_{\mathrm{M}}, \beta_{\mathrm{M}}) \tag{5.28}$$

with the hyperparameters $\alpha_{\mathrm{M}}, \beta_{\mathrm{M}} \in \mathbb{R}_{>0}$. By combining (5.22)–(5.28), we obtain

$$\begin{aligned} f_{\mathbf{p}_i}(\boldsymbol{p}_i) &= f_{\mathbf{D}_i}(\boldsymbol{D}_i) f_{\mathbf{Q}_i}(\boldsymbol{Q}_i) f_{\bar{\mathsf{M}}_i}(\bar{M}_i) \\ &= \Gamma^{-1}(d_{i,1}^2; \alpha_{\mathrm{d},1}, \beta_{\mathrm{d},1})\, \Gamma^{-1}(d_{i,2}^2; \alpha_{\mathrm{d},2}, \beta_{\mathrm{d},2}) \\ &\quad \times \Gamma^{-1}(q_{i,1}^2; \alpha_{\mathrm{v},1}, \beta_{\mathrm{v},1})\, \Gamma^{-1}(q_{i,2}^2; \alpha_{\mathrm{v},2}, \beta_{\mathrm{v},2}) \\ &\quad \times \Gamma(\bar{M}_i; \alpha_{\mathrm{M}}, \beta_{\mathrm{M}}) \end{aligned} \tag{5.29}$$

with

$$\boldsymbol{p}_i = (\boldsymbol{D}_i, \boldsymbol{Q}_i, \bar{M}_i) = \left(\begin{bmatrix} d_{i,1}^2 & 0 \\ 0 & d_{i,2}^2 \end{bmatrix}, \begin{bmatrix} q_{i,1}^2 \mathbf{I}_2 & \mathbf{0}_{2\times 2} \\ \mathbf{0}_{2\times 2} & q_{i,2}^2 \mathbf{I}_2 \end{bmatrix}, \bar{M}_i \right). \tag{5.30}$$

As mentioned above, the joint distribution of all parameter triples $\mathbf{p}_{1:I}$ is defined by a Dirichlet process according to Definition 3.1, that is,

$$\mathsf{f}_{\mathrm{DP}} \sim \mathrm{DP}(\alpha_{\mathrm{DP}}, f_G), \tag{5.31}$$

where $\alpha_{\mathrm{DP}} \in \mathbb{R}_{>0}$ is the concentration parameter and $f_G(\boldsymbol{p}_i)$ is the base pdf, and

$$\mathbf{p}_i \,|\, (\mathsf{f}_{\mathrm{DP}} = f_{\mathrm{DP}}) \overset{\text{i.i.d.}}{\sim} f_{\mathrm{DP}}. \tag{5.32}$$

As $f_G(\boldsymbol{p}_i) = f_{\mathbf{p}_i}(\boldsymbol{p}_i)$ (see (3.19)), we have (see (5.29))

$$\begin{aligned} f_G(\boldsymbol{p}_i) &= \Gamma^{-1}(d_{i,1}^2; \alpha_{\mathrm{d},1}, \beta_{\mathrm{d},1})\, \Gamma^{-1}(d_{i,2}^2; \alpha_{\mathrm{d},2}, \beta_{\mathrm{d},2}) \\ &\quad \times \Gamma^{-1}(q_{i,1}^2; \alpha_{\mathrm{v},1}, \beta_{\mathrm{v},1})\, \Gamma^{-1}(q_{i,2}^2; \alpha_{\mathrm{v},2}, \beta_{\mathrm{v},2}) \\ &\quad \times \Gamma(\bar{M}_i; \alpha_{\mathrm{M}}, \beta_{\mathrm{M}}). \end{aligned} \tag{5.33}$$

The targets in each class have equal parameter triples (see (3.25)), that is,

$$\mathbf{p}_i = \mathbf{p}_{\mathsf{c}_i}^*, \tag{5.34}$$

where $\mathsf{c}_i \in \mathbb{N}$ is the random class indicator (which assigns a class to target i) and

$$\mathbf{p}_c^* = (\mathbf{D}_c^*, \mathbf{Q}_c^*, \bar{\mathsf{M}}_c^*) \tag{5.35}$$

is the random parameter triple of class c.

5.1.4 Independence Assumptions

Let us briefly summarize some (conditional) independence assumptions that will be needed in the following section. Some of the following assumptions were already discussed above or can be found in Figure 5.2.

A.1) The states and measurements $(\mathbf{x}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}}, \mathbf{y}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}})$ given the parameters $\mathbf{p}_i$ are conditionally independent of the states and measurements $(\mathbf{x}_{i',N_{\mathrm{S},i'}:N_{\mathrm{E},i'}}, \mathbf{y}_{i',N_{\mathrm{S},i'}:N_{\mathrm{E},i'}})$ as well as the parameters $\mathbf{p}_{i'}$ of every other target $i' \neq i$.

A.2) The state $\mathbf{x}_{i,n}$ is conditionally independent of the states $\mathbf{x}_{i,N_{\mathrm{S},i}:n-2}$ given the previous state $\mathbf{x}_{i,n-1}$ and the parameters $\mathbf{p}_i$.

A.3) The measurement $\mathbf{y}_{i,n}$ is conditionally independent of the states $\mathbf{x}_{i,N_{\mathrm{S},i}:n-1}$ and measurements $\mathbf{y}_{i,N_{\mathrm{S},i}:n-1}$ given the state $\mathbf{x}_{i,n}$ and the parameters $\mathbf{p}_i$.

A.4) The state $\mathbf{x}_{i,N_{\mathrm{S},i}}$ is independent of the parameters $\mathbf{p}_i$.

A.5) The states $\mathbf{x}_{i,N_{\mathrm{S},i}:n}$ are conditionally independent of the parameters $\mathbf{D}_i$ and $\bar{\mathsf{M}}_i$ given the parameter $\mathbf{Q}_i$.

A.6) The numbers of measurements $\mathsf{M}_{i,N_{\mathrm{S},i}:n}$ are conditionally independent of the states $\mathbf{x}_{i,N_{\mathrm{S},i}:n}$, and the parameters $\mathbf{Q}_i$ and $\mathbf{D}_i$ given the parameter $\bar{\mathsf{M}}_i$.

A.7) The measurements $\mathbf{y}_{i,n}$ are conditionally independent of the parameters $\mathbf{Q}_i$ and $\bar{\mathsf{M}}_i$ given the number of measurements $\mathsf{M}_{i,n}$, the state $\mathbf{x}_{i,n}$, and the parameter $\mathbf{D}_i$.

A.8) The pre-measurement $\mathbf{z}_{i,n,m}$ is conditionally independent of the number of measurements $\mathsf{M}_{i,n}$ given the parameter $\mathbf{D}_i$.

A.9) The pre-measurement $\mathbf{z}_{i,n,m}$ is conditionally independent of the state $\mathbf{x}_{i,n}$ given the parameter $\mathbf{D}_i$.

A.10) The parameter $\mathbf{D}_i$ is independent of the numbers of measurements $\mathsf{M}_{i,1:n}$ and the states $\mathbf{x}_{i,N_{\mathrm{S},i}:n}$.

A.11) The parameter $\mathbf{D}_i$ is conditionally independent of the states $\mathbf{x}_{i,N_{\mathrm{S},i}:n}$ given the pre-measurements $\mathbf{z}_{i,N_{\mathrm{S},i}:n}$.

A.12) The parameter $\mathbf{Q}_i$ is conditionally independent of the states $\mathbf{x}_{i,N_{\mathrm{S},i}:n}$ given the driving noise $\mathbf{v}_{i,N_{\mathrm{S},i}+1:n}$.

5.2 Inference

In this section, we will concretize the distributions used in the algorithms in Chapter 4 when applied to the statistical model for extended target tracking described in Section 5.1. Furthermore, we will introduce a simple MCMC algorithm for inference in our statistical model, which will be used as a reference algorithm to illustrate the performance of the algorithms introduced in Chapter 4. This reference algorithm, however, can only be derived for some special cases of a Bayesian nonparametric state-space model, while the algorithms introduced in Chapter 4 can be applied to a wide range of Bayesian nonparametric state-space models. For simplicity, we will only discuss the case $N_{\mathrm{S},i} = 1$ and $N_{\mathrm{E},i} = N$ for all targets $i \in \{1, \ldots, I\}$. This setting can easily be adapted to the general case as discussed in Section 4.2.

As discussed in Section 4.1.1, the algorithms are used to generate samples in order to approximate an expectation (see (4.6))

$$J = \mathbb{E}(h(\mathbf{x}_{1:I,1:N}, \mathbf{c}_{1:I}, \mathbf{p}^*_{\mathbf{c}_{1:I}}) \,|\, \mathbf{y}_{1:I,1:N} = \boldsymbol{y}_{1:I,1:N}), \tag{5.36}$$

or to infer the most likely class assignment (see (4.8))

$$\hat{\boldsymbol{c}}_{\mathrm{joint},1:I} = \mathrm{argmax}_{\boldsymbol{c}_{1:I} \in \mathbb{N}^I} \mathbb{P}(\mathbf{c}_{1:I} = \boldsymbol{c}_{1:I} \,|\, \mathbf{y}_{1:I,1:N} = \boldsymbol{y}_{1:I,1:N}). \tag{5.37}$$

5.2.1 Conditional Probability Distributions

Before we discuss the specific algorithms, let us first derive, within our statistical model, the conditional probability distributions that are used in the algorithms presented in Chapter 4.

Conditional pdf $f_{\mathbf{p}_i|\mathbf{x}_{i,1:n},\mathbf{y}_{i,1:n}}$ We first consider the conditional pdf of the parameter triple $\mathbf{p}_i$ given the states $\mathbf{x}_{i,1:n}$ and measurements $\mathbf{y}_{i,1:n}$. By applying Bayes' theorem and the chain rule we obtain

$$\begin{aligned} f_{\mathbf{p}_i|\mathbf{x}_{i,1:n},\mathbf{y}_{i,1:n}}(\boldsymbol{p}_i \,|\, \boldsymbol{x}_{i,1:n}, \boldsymbol{y}_{i,1:n}) &\propto v_{\mathbf{x}_{i,1:n},\mathbf{y}_{i,1:n}|\mathbf{p}_i}(\boldsymbol{x}_{i,1:n}, \boldsymbol{y}_{i,1:n} \,|\, \boldsymbol{p}_i) f_{\mathbf{p}_i}(\boldsymbol{p}_i) \\ &= \sum_{M'=0}^{\infty} v_{\mathbf{x}_{i,1:n},\mathbf{y}_{i,1:n},\mathsf{M}_{i,1:n}|\mathbf{p}_i}(\boldsymbol{x}_{i,1:n}, \boldsymbol{y}_{i,1:n}, M' \,|\, \boldsymbol{p}_i) f_{\mathbf{p}_i}(\boldsymbol{p}_i). \end{aligned} \tag{5.38}$$

Using that $\mathsf{y}_{i,n}$ defines $\mathsf{M}_{i,n}$, the chain rule, and (5.29), we have

$$\begin{aligned}
&f_{\mathsf{p}_i|\mathsf{x}_{i,1:n},\mathsf{y}_{i,1:n}}(\boldsymbol{p}_i \,|\, \boldsymbol{x}_{i,1:n}, \boldsymbol{y}_{i,1:n}) \\
&\quad = v_{\mathsf{x}_{i,1:n},\mathsf{y}_{i,1:n},\mathsf{M}_{i,1:n}|\mathsf{p}_i}(\boldsymbol{x}_{i,1:n}, \boldsymbol{y}_{i,1:n}, M_{i,1:n} \,|\, \boldsymbol{p}_i) f_{\mathsf{p}_i}(\boldsymbol{p}_i) \\
&\quad = v_{\mathsf{x}_{i,1:n},\mathsf{y}_{i,1:n},\mathsf{M}_{i,1:n}|\mathsf{D}_i,\mathsf{Q}_i,\bar{\mathsf{M}}_i}(\boldsymbol{x}_{i,1:n}, \boldsymbol{y}_{i,1:n}, M_{i,1:n} \,|\, \boldsymbol{D}_i, \boldsymbol{Q}_i, \bar{M}_i) f_{\mathsf{D}_i}(\boldsymbol{D}_i) f_{\mathsf{Q}_i}(\boldsymbol{Q}_i) f_{\bar{\mathsf{M}}_i}(\bar{M}_i) \\
&\quad = f_{\mathsf{y}_{i,1:n}|\mathsf{M}_{i,1:n},\mathsf{x}_{i,1:n},\mathsf{D}_i,\mathsf{Q}_i,\bar{\mathsf{M}}_i}(\boldsymbol{y}_{i,1:n} \,|\, M_{i,1:n}, \boldsymbol{x}_{i,1:n}, \boldsymbol{D}_i, \boldsymbol{Q}_i, \bar{M}_i) f_{\mathsf{D}_i}(\boldsymbol{D}_i) \\
&\qquad \times f_{\mathsf{x}_{i,1:n}|\mathsf{D}_i,\mathsf{Q}_i,\bar{\mathsf{M}}_i}(\boldsymbol{x}_{i,1:n} \,|\, \boldsymbol{D}_i, \boldsymbol{Q}_i, \bar{M}_i) f_{\mathsf{Q}_i}(\boldsymbol{Q}_i) \\
&\qquad \times p_{\mathsf{M}_{i,1:n}|\mathsf{x}_{i,1:n},\mathsf{D}_i,\mathsf{Q}_i,\bar{\mathsf{M}}_i}(M_{i,1:n} \,|\, \boldsymbol{x}_{i,1:n}, \boldsymbol{D}_i, \boldsymbol{Q}_i, \bar{M}_i) f_{\bar{\mathsf{M}}_i}(\bar{M}_i). \qquad (5.39)
\end{aligned}$$

Let us simplify the factors in (5.39), starting with

$$\begin{aligned}
&f_{\mathsf{y}_{i,1:n}|\mathsf{M}_{i,1:n},\mathsf{x}_{i,1:n},\mathsf{D}_i,\mathsf{Q}_i,\bar{\mathsf{M}}_i}(\boldsymbol{y}_{i,1:n} \,|\, M_{i,1:n}, \boldsymbol{x}_{i,1:n}, \boldsymbol{D}_i, \boldsymbol{Q}_i, \bar{M}_i) f_{\mathsf{D}_i}(\boldsymbol{D}_i) \\
&\quad = f_{\mathsf{y}_{i,1:n}|\mathsf{M}_{i,1:n},\mathsf{x}_{i,1:n},\mathsf{D}_i}(\boldsymbol{y}_{i,1:n} \,|\, M_{i,1:n}, \boldsymbol{x}_{i,1:n}, \boldsymbol{D}_i) f_{\mathsf{D}_i}(\boldsymbol{D}_i) \\
&\quad = f_{\mathsf{y}_{i,1:n}|\mathsf{M}_{i,1:n},\mathsf{x}_{i,1:n},\mathsf{D}_i}(\boldsymbol{y}_{i,1:n} \,|\, M_{i,1:n}, \boldsymbol{x}_{i,1:n}, \boldsymbol{D}_i) f_{\mathsf{D}_i|\mathsf{M}_{i,1:n},\mathsf{x}_{i,1:n}}(\boldsymbol{D}_i \,|\, M_{i,1:n}, \boldsymbol{x}_{i,1:n}), \qquad (5.40)
\end{aligned}$$

where the first step can be shown using assumptions A.7 and A.3, and the second step follows from assumption A.10. Using Bayes' theorem, that (5.10) is a bijective transformation of $\mathsf{z}_{i,n,m}$ given $\mathsf{x}_{i,n}$, and then finally assumption A.11, we can further simplify (5.40) to

$$\begin{aligned}
&f_{\mathsf{y}_{i,1:n}|\mathsf{M}_{i,1:n},\mathsf{x}_{i,1:n},\mathsf{D}_i,\mathsf{Q}_i,\bar{\mathsf{M}}_i}(\boldsymbol{y}_{i,1:n} \,|\, M_{i,1:n}, \boldsymbol{x}_{i,1:n}, \boldsymbol{D}_i, \boldsymbol{Q}_i, \bar{M}_i) f_{\mathsf{D}_i}(\boldsymbol{D}_i) \\
&\quad \propto f_{\mathsf{D}_i|\mathsf{y}_{i,1:n},\mathsf{M}_{i,1:n},\mathsf{x}_{i,1:n}}(\boldsymbol{D}_i \,|\, \boldsymbol{y}_{i,1:n}, M_{i,1:n}, \boldsymbol{x}_{i,1:n}) \\
&\quad = f_{\mathsf{D}_i|\mathsf{z}_{i,1:n},\mathsf{M}_{i,1:n},\mathsf{x}_{i,1:n}}(\boldsymbol{D}_i \,|\, \boldsymbol{z}_{i,1:n}, M_{i,1:n}, \boldsymbol{x}_{i,1:n}) \\
&\quad = f_{\mathsf{D}_i|\mathsf{z}_{i,1:n},\mathsf{M}_{i,1:n}}(\boldsymbol{D}_i \,|\, \boldsymbol{z}_{i,1:n}, M_{i,1:n}) \qquad (5.41)
\end{aligned}$$

with $\mathsf{z}_{i,n} = \left(\mathsf{z}_{i,n,1}, \ldots, \mathsf{z}_{i,n,\mathsf{M}_{i,n}}\right)$ and (see (5.10))

$$\mathsf{z}_{i,n,m} = \boldsymbol{R}(\mathsf{x}_{i,n})^{-1}(\mathsf{y}_{i,n,m} - \boldsymbol{H}\mathsf{x}_{i,n}) = \boldsymbol{R}(\mathsf{x}_{i,n})^{\mathrm{T}}(\mathsf{y}_{i,n,m} - \boldsymbol{H}\mathsf{x}_{i,n}). \qquad (5.42)$$

In order to further simplify (5.41), first note that we have

$$\begin{aligned}
f_{\mathsf{z}_{i,1:n}|\mathsf{M}_{i,1:n},\mathsf{D}_i}(\boldsymbol{z}_{i,1:n} \,|\, M_{i,1:n}, \boldsymbol{D}_i) &= \prod_{n'=1}^{n} \prod_{m=1}^{M_{i,n'}} f_{\mathsf{z}_{i,n',m}|\mathsf{D}_i}(\boldsymbol{z}_{i,n',m} \,|\, \boldsymbol{D}_i) \\
&= \prod_{n'=1}^{n} \prod_{m=1}^{M_{i,n'}} \mathcal{N}(\boldsymbol{z}_{i,n',m}; \boldsymbol{0}, \boldsymbol{D}_i), \qquad (5.43)
\end{aligned}$$

where we have used assumption A.8, (5.12), and that the $\mathsf{z}_{i,n,m}$ were assumed conditionally i.i.d. across n and m given D_i. As D_i is a diagonal matrix with independent inverse

gamma distributed entries (see (5.22)–(5.24)) and as the inverse gamma distribution is the conjugate prior for the variance of the Gaussian distribution,[1] it finally follows from (5.41) and (5.43) that

$$\begin{aligned} f_{\mathsf{y}_{i,1:n}|\mathsf{M}_{i,1:n},\mathsf{x}_{i,1:n},\mathsf{D}_i,\mathsf{Q}_i,\bar{\mathsf{M}}_i}(\boldsymbol{y}_{i,1:n} \,|\, M_{i,1:n}, \boldsymbol{x}_{i,1:n}, \boldsymbol{D}_i, \boldsymbol{Q}_i, \bar{M}_i) f_{\mathsf{D}_i}(\boldsymbol{D}_i) \\ \propto \Gamma^{-1}\left(d_{i,1}^2; \alpha_{\mathrm{d},1} + \frac{1}{2}\sum_{n'=1}^{n} M_{i,n'}, \beta_{\mathrm{d},1} + \frac{1}{2}\sum_{n'=1}^{n}\sum_{m=1}^{M_{i,n'}} z_{i,n',m,1}^2\right) \\ \times \Gamma^{-1}\left(d_{i,2}^2; \alpha_{\mathrm{d},2} + \frac{1}{2}\sum_{n'=1}^{n} M_{i,n'}, \beta_{\mathrm{d},2} + \frac{1}{2}\sum_{n'=1}^{n}\sum_{m=1}^{M_{i,n'}} z_{i,n',m,2}^2\right). \end{aligned} \tag{5.44}$$

with $z_{i,n,m,1}$, $z_{i,n,m,2}$, $d_{i,1}^2$, and $d_{i,2}^2$ being the elements of $\boldsymbol{z}_{i,n,m}$ and $\boldsymbol{D}_i$, that is,

$$\boldsymbol{z}_{i,n,m} = [z_{i,n,m,1} \; z_{i,n,m,2}]^{\mathrm{T}} = \boldsymbol{R}(\boldsymbol{x}_{i,n})^{\mathrm{T}}(\boldsymbol{y}_{i,n,m} - \boldsymbol{H}\boldsymbol{x}_{i,n}) \tag{5.45}$$

and (see (5.13))

$$\boldsymbol{D}_i = \begin{bmatrix} d_{i,1}^2 & 0 \\ 0 & d_{i,2}^2 \end{bmatrix}. \tag{5.46}$$

The second factor in (5.39) to simplify is

$$\begin{aligned} f_{\mathsf{x}_{i,1:n}|\mathsf{D}_i,\mathsf{Q}_i,\bar{\mathsf{M}}_i}(\boldsymbol{x}_{i,1:n} \,|\, \boldsymbol{D}_i, \boldsymbol{Q}_i, \bar{M}_i) f_{\mathsf{Q}_i}(\boldsymbol{Q}_i) &= f_{\mathsf{x}_{i,1:n}|\mathsf{Q}_i}(\boldsymbol{x}_{i,1:n} \,|\, \boldsymbol{Q}_i) f_{\mathsf{Q}_i}(\boldsymbol{Q}_i) \\ &\propto f_{\mathsf{Q}_i|\mathsf{x}_{i,1:n}}(\boldsymbol{Q}_i \,|\, \boldsymbol{x}_{i,1:n}), \end{aligned} \tag{5.47}$$

where we have used assumption A.5 and Bayes' theorem. As

$$\mathsf{v}_{i,n} = \mathsf{x}_{i,n} - \boldsymbol{F}\mathsf{x}_{i,n-1} \tag{5.48}$$

for $n \in \{2, \ldots, N\}$, by (5.2), we can express $\mathsf{v}_{i,2:n}$ using $\mathsf{x}_{i,1:n}$, and thus, we can restate (5.47) as

$$\begin{aligned} f_{\mathsf{x}_{i,1:n}|\mathsf{D}_i,\mathsf{Q}_i,\bar{\mathsf{M}}_i}(\boldsymbol{x}_{i,1:n} \,|\, \boldsymbol{D}_i, \boldsymbol{Q}_i, \bar{M}_i) f_{\mathsf{Q}_i}(\boldsymbol{Q}_i) &\propto f_{\mathsf{Q}_i|\mathsf{v}_{i,2:n},\mathsf{x}_{i,1:n}}(\boldsymbol{Q}_i \,|\, \boldsymbol{v}_{i,2:n}, \boldsymbol{x}_{i,1:n}) \\ &= f_{\mathsf{Q}_i|\mathsf{v}_{i,2:n}}(\boldsymbol{Q}_i \,|\, \boldsymbol{v}_{i,2:n}), \end{aligned} \tag{5.49}$$

where we have used assumption A.12. In order to further simplify (5.49), first note that we have

$$f_{\mathsf{v}_{i,2:n}|\mathsf{Q}_i}(\boldsymbol{v}_{i,2:n} \,|\, \boldsymbol{Q}_i) = \prod_{n'=2}^{n} f_{\mathsf{v}_{i,n'}|\mathsf{Q}_i}(\boldsymbol{v}_{i,n'} \,|\, \boldsymbol{Q}_i) = \prod_{n'=2}^{n} \mathcal{N}(\boldsymbol{v}_{i,n'}; \boldsymbol{0}, \boldsymbol{Q}_i), \tag{5.50}$$

[1] If $\mathsf{p} \sim \Gamma^{-1}(\alpha, \beta)$ and $\mathsf{x}_i|(\mathsf{p} = p) \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mu, p)$ for $i \in \{1, \ldots, I\}$, then $\mathsf{p}|(\mathsf{x}_{1:I} = x_{1:I}) \sim \Gamma^{-1}(\alpha', \beta')$ with $\alpha' = \alpha + \frac{I}{2}$ and $\beta' = \beta + \frac{1}{2}\sum_{i=1}^{I}(x_i - \mu)^2$.

as the $\mathsf{v}_{i,n}$ are i.i.d. across n given Q_i, and Gaussian according to (5.4). Using again the fact that the inverse gamma distribution is the conjugate prior for the variance of the Gaussian distribution and that Q_i is a diagonal matrix with inverse gamma distributed entries (see (5.25)–(5.27)), we obtain

$$\begin{aligned} &f_{\mathsf{x}_{i,1:n}|\mathsf{D}_i,\mathsf{Q}_i,\bar{\mathsf{M}}_i}(\boldsymbol{x}_{i,1:n} \,|\, \boldsymbol{D}_i, \boldsymbol{Q}_i, \bar{M}_i) f_{\mathsf{Q}_i}(\boldsymbol{Q}_i) \\ &\quad \propto \Gamma^{-1}\left(q_{i,1}^2; \alpha_{\mathrm{v},1} + n - 1, \beta_{\mathrm{v},1} + \frac{1}{2}\sum_{n'=2}^{n}(v_{i,n',1}^2 + v_{i,n',2}^2)\right) \\ &\qquad \times \Gamma^{-1}\left(q_{i,2}^2; \alpha_{\mathrm{v},2} + n - 1, \beta_{\mathrm{v},2} + \frac{1}{2}\sum_{n'=2}^{n}(v_{i,n',3}^2 + v_{i,n',4}^2)\right) \end{aligned} \tag{5.51}$$

with $v_{i,n,1}$, $v_{i,n,2}$, $v_{i,n,3}$, $v_{i,n,4}$, $q_{i,1}^2$, and $q_{i,2}^2$ being the elements of $\boldsymbol{v}_{i,n}$ and $\boldsymbol{Q}_i$, that is,

$$\boldsymbol{v}_{i,n} = [v_{i,n,1}\ v_{i,n,2}\ v_{i,n,3}\ v_{i,n,4}]^{\mathrm{T}} \tag{5.52}$$

and (see (5.5))

$$\boldsymbol{Q}_i = \begin{bmatrix} q_{i,1}^2 \mathbf{I}_2 & \mathbf{0}_{2\times 2} \\ \mathbf{0}_{2\times 2} & q_{i,2}^2 \mathbf{I}_2 \end{bmatrix}. \tag{5.53}$$

The last factor in (5.39) that we will simplify is

$$\begin{aligned} p_{\mathsf{M}_{i,1:n}|\mathsf{x}_{i,1:n},\mathsf{D}_i,\mathsf{Q}_i,\bar{\mathsf{M}}_i}(M_{i,1:n} \,|\, \boldsymbol{x}_{i,1:n}, \boldsymbol{D}_i, \boldsymbol{Q}_i, \bar{M}_i) f_{\bar{\mathsf{M}}_i}(\bar{M}_i) &= p_{\mathsf{M}_{i,1:n}|\bar{\mathsf{M}}_i}(M_{i,1:n} \,|\, \bar{M}_i) f_{\bar{\mathsf{M}}_i}(\bar{M}_i) \\ &\propto f_{\bar{\mathsf{M}}_i|\mathsf{M}_{i,1:n}}(\bar{M}_i \,|\, M_{i,1:n}), \end{aligned} \tag{5.54}$$

where we have used assumption A.6 and Bayes' theorem. Note that the $\mathsf{M}_{i,1:n}$ given $\bar{\mathsf{M}}_i$ are Poisson distributed according to (5.9) and that $\bar{\mathsf{M}}_i$ is gamma distributed according to (5.28). As the gamma distribution is the conjugate prior of the Poisson distribution,[2] it follows that

$$\begin{aligned} &p_{\mathsf{M}_{i,1:n}|\mathsf{x}_{i,1:n},\mathsf{D}_i,\mathsf{Q}_i,\bar{\mathsf{M}}_i}(M_{i,1:n} \,|\, \boldsymbol{x}_{i,1:n}, \boldsymbol{D}_i, \boldsymbol{Q}_i, \bar{M}_i) f_{\bar{\mathsf{M}}_i}(\bar{M}_i) \\ &\quad \propto \Gamma\left(\bar{M}_i; \alpha_{\mathrm{M}} + \sum_{n'=1}^{n} M_{i,n'}, \beta_{\mathrm{M}} + n\right). \end{aligned} \tag{5.55}$$

Having simplified all factors of (5.39) with (5.44), (5.55), and (5.51), we finally arrive

[2] If $\mathsf{p} \sim \Gamma(\alpha, \beta)$ and $\mathsf{x}_i|(\mathsf{p} = p) \overset{\text{i.i.d.}}{\sim} \mathrm{Pois}(p)$ for $i \in \{1, \ldots, I\}$, then $\mathsf{p}|(\mathsf{x}_{1:I} = x_{1:I}) \sim \Gamma(\alpha', \beta')$ with $\alpha' = \alpha + \sum_{i=1}^{I} x_i$ and $\beta' = \beta + I$.

at

$$
\begin{aligned}
f_{\mathbf{p}_i|\mathbf{x}_{i,1:n},\mathbf{y}_{i,1:n}}(\boldsymbol{p}_i \,|\, \boldsymbol{x}_{i,1:n}, \boldsymbol{y}_{i,1:n}) = \Gamma^{-1}&\left(d_{i,1}^2; \alpha_{\mathrm{d},1} + \frac{1}{2}\sum_{n'=1}^{n} M_{i,n'}, \beta_{\mathrm{d},1} + \frac{1}{2}\sum_{n'=1}^{n}\sum_{m=1}^{M_{i,n'}} z_{i,n',m,1}^2\right) \\
&\times \Gamma^{-1}\left(d_{i,2}^2; \alpha_{\mathrm{d},2} + \frac{1}{2}\sum_{n'=1}^{n} M_{i,n'}, \beta_{\mathrm{d},2} + \frac{1}{2}\sum_{n'=1}^{n}\sum_{m=1}^{M_{i,n'}} z_{i,n',m,2}^2\right) \\
&\times \Gamma^{-1}\left(q_{i,1}^2; \alpha_{\mathrm{v},1} + n - 1, \beta_{\mathrm{v},1} + \frac{1}{2}\sum_{n'=2}^{n} (v_{i,n',1}^2 + v_{i,n',2}^2)\right) \\
&\times \Gamma^{-1}\left(q_{i,2}^2; \alpha_{\mathrm{v},2} + n - 1, \beta_{\mathrm{v},2} + \frac{1}{2}\sum_{n'=2}^{n} (v_{i,n',3}^2 + v_{i,n',4}^2)\right) \\
&\times \Gamma\left(\bar{M}_i; \alpha_{\mathrm{M}} + \sum_{n'=1}^{n} M_{i,n'}, \beta_{\mathrm{M}} + n\right) \qquad (5.56)
\end{aligned}
$$

with

$$\boldsymbol{p}_i = (\boldsymbol{D}_i, \boldsymbol{Q}_i, \bar{M}_i), \qquad (5.57)$$

$$\boldsymbol{D}_i = \begin{bmatrix} d_{i,1}^2 & 0 \\ 0 & d_{i,2}^2 \end{bmatrix}, \qquad (5.58)$$

$$\boldsymbol{Q}_i = \begin{bmatrix} q_{i,1}^2 \mathbf{I}_2 & \mathbf{0}_{2\times 2} \\ \mathbf{0}_{2\times 2} & q_{i,2}^2 \mathbf{I}_2 \end{bmatrix}, \qquad (5.59)$$

$$\boldsymbol{y}_{i,n'} = \left(\boldsymbol{y}_{i,n',1}, \ldots, \boldsymbol{y}_{i,n',M_{i,n'}}\right), \qquad (5.60)$$

$$\boldsymbol{z}_{i,n',m} = \boldsymbol{R}(\boldsymbol{x}_{i,n'})^{\mathrm{T}}(\boldsymbol{y}_{i,n',m} - \boldsymbol{H}\boldsymbol{x}_{i,n'}) = [z_{i,n',m,1}, z_{i,n',m,2}]^{\mathrm{T}}, \qquad (5.61)$$

$$\boldsymbol{v}_{i,n'} = \boldsymbol{x}_{i,n'} - \boldsymbol{F}\boldsymbol{x}_{i,n'-1} = [v_{i,n',1}\ v_{i,n',2}\ v_{i,n',3}\ v_{i,n',4}]^{\mathrm{T}}. \qquad (5.62)$$

Conditional pdf $f_{\mathbf{p}_c^*|\mathbf{x}_{1:I,1:n},\mathbf{c}_{1:I},\mathbf{p}_{\neg c}^*,\mathbf{y}_{1:I,1:n}}$ We will also need the conditional pdf of the class parameter triple $\mathbf{p}_c^*$ given the states $\mathbf{x}_{1:I,1:n}$, indicator variables $\mathbf{c}_{1:I}$, other class parameter triples $\mathbf{p}_{\neg c}^*$, and measurements $\mathbf{y}_{1:I,1:n}$. Using (4.23), we have

$$
\begin{aligned}
&f_{\mathbf{p}_c^*|\mathbf{x}_{1:I,1:n},\mathbf{c}_{1:I},\mathbf{p}_{\neg c}^*,\mathbf{y}_{1:I,1:n}}(\boldsymbol{p}_c^* \,|\, \boldsymbol{x}_{1:I,1:n}, \boldsymbol{c}_{1:I}, \boldsymbol{p}_{\neg c}^*, \boldsymbol{y}_{1:I,1:n}) \\
&\quad \propto \left(\prod_{i:c_i=c} v_{\mathbf{x}_{i,1:n},\mathbf{y}_{i,1:n}|\mathbf{p}_i}(\boldsymbol{x}_{i,1:n}, \boldsymbol{y}_{i,1:n} \,|\, \boldsymbol{p}_c^*)\right) f_{\mathrm{G}}(\boldsymbol{p}_c^*). \qquad (5.63)
\end{aligned}
$$

Further, using the same derivation as in (5.39), but using the states and observations of all targets i with $\mathsf{c}_i = c$ instead of just a single target, we can restate (5.63) as

$$\begin{aligned}
&f_{\mathsf{p}_c^*|\mathsf{x}_{1:I,1:n},\mathsf{c}_{1:I},\mathsf{p}_{\neg c}^*,\mathsf{y}_{1:I,1:n}}(\boldsymbol{p}_c^* \,|\, \boldsymbol{x}_{1:I,1:n}, \boldsymbol{c}_{1:I}, \boldsymbol{p}_{\neg c}^*, \boldsymbol{y}_{1:I,1:n}) \\
&\quad \propto \prod_{i:c_i=c} \Big(f_{\mathsf{y}_{i,1:n}|\mathsf{M}_{i,1:n},\mathsf{x}_{i,1:n},\mathsf{D}_i,\mathsf{Q}_i,\bar{\mathsf{M}}_i}(\boldsymbol{y}_{i,1:n} \,|\, M_{i,1:n}, \boldsymbol{x}_{i,1:n}, \boldsymbol{D}_c^*, \boldsymbol{Q}_c^*, \bar{M}_c^*) f_{\mathsf{D}_i}(\boldsymbol{D}_c^*) \\
&\qquad \times f_{\mathsf{x}_{i,1:n}|\mathsf{D}_i,\mathsf{Q}_i,\bar{\mathsf{M}}_i}(\boldsymbol{x}_{i,1:n} \,|\, \boldsymbol{D}_c^*, \boldsymbol{Q}_c^*, \bar{M}_c^*) f_{\mathsf{Q}_i}(\boldsymbol{Q}_c^*) \\
&\qquad \times p_{\mathsf{M}_{i,1:n}|\mathsf{x}_{i,1:n},\mathsf{D}_i,\mathsf{Q}_i,\bar{\mathsf{M}}_i}(M_{i,1:n} \,|\, \boldsymbol{x}_{i,1:n}, \boldsymbol{D}_c^*, \boldsymbol{Q}_c^*, \bar{M}_c^*) f_{\bar{\mathsf{M}}_i}(\bar{M}_c^*) \Big).
\end{aligned} \tag{5.64}$$

By using almost the same derivations as for (5.44), (5.55), and (5.51), but again using the states and observations of all targets i with $\mathsf{c}_i = c$ instead of just a single target, and then inserting the results into (5.64), we obtain (see (5.56))

$$\begin{aligned}
&f_{\mathsf{p}_c^*|\mathsf{x}_{1:I,1:n},\mathsf{c}_{1:I},\mathsf{p}_{\neg c}^*,\mathsf{y}_{1:I,1:n}}(\boldsymbol{p}_c^* \,|\, \boldsymbol{x}_{1:I,1:n}, \boldsymbol{c}_{1:I}, \boldsymbol{p}_{\neg c}^*, \boldsymbol{y}_{1:I,1:n}) \\
&\quad = \Gamma^{-1}\left(d_{c,1}^{*2}; \alpha_{\mathrm{d},1} + \frac{1}{2} \sum_{i:c_i=c} \sum_{n'=1}^{n} M_{i,n'}, \beta_{\mathrm{d},1} + \frac{1}{2} \sum_{i:c_i=c} \sum_{n'=1}^{n} \sum_{m=1}^{M_{i,n'}} z_{i,n',m,1}^2 \right) \\
&\qquad \times \Gamma^{-1}\left(d_{c,2}^{*2}; \alpha_{\mathrm{d},2} + \frac{1}{2} \sum_{i:c_i=c} \sum_{n'=1}^{n} M_{i,n'}, \beta_{\mathrm{d},2} + \frac{1}{2} \sum_{i:c_i=c} \sum_{n'=1}^{n} \sum_{m=1}^{M_{i,n'}} z_{i,n',m,2}^2 \right) \\
&\qquad \times \Gamma\left(\bar{M}_c^*; \alpha_{\mathrm{M}} + \sum_{i:c_i=c} \sum_{n'=1}^{n} M_{i,n'}, \beta_{\mathrm{M}} + \sum_{i:c_i=c} n \right) \\
&\qquad \times \Gamma^{-1}\left(q_{c,1}^{*2}; \alpha_{\mathrm{v},1} + (n-1) \sum_{i:c_i=c} 1, \beta_{\mathrm{v},1} + \frac{1}{2} \sum_{i:c_i=c} \sum_{n'=2}^{n} (v_{i,n',1}^2 + v_{i,n',2}^2) \right) \\
&\qquad \times \Gamma^{-1}\left(q_{c,2}^{*2}; \alpha_{\mathrm{v},2} + (n-1) \sum_{i:c_i=c} 1, \beta_{\mathrm{v},2} + \frac{1}{2} \sum_{i:c_i=c} \sum_{n'=2}^{n} (v_{i,n',3}^2 + v_{i,n',4}^2) \right)
\end{aligned} \tag{5.65}$$

with (5.57)–(5.62). Note that in (5.65) we use the data of all targets in each class, instead of the data of a single target as in (5.56).

5.2.2 Reference Algorithm

To illustrate the performance of the algorithms introduced in Chapter 4 for the statistical model of this chapter, we will first derive a simple MCMC algorithm that uses cycles of MCMC kernels (see Section 2.4.3) as a reference. That is, we develop kernels to sample each random vector of the states $\mathbf{x}_{1:I,1:N}$, indicator variables $\mathbf{c}_{1:I}$, and parameter triples $\mathbf{p}_{\mathbf{c}_{1:I}}^*$ separately given the current samples of every other random vector. Note that this algorithm can only be derived for some special cases of a Bayesian nonparametric state-space model.

The algorithm starts with an intialization of arbitrary samples $\boldsymbol{x}_{1:I,1:N}^{(1)}$, $\boldsymbol{c}_{1:I}^{(1)}$, and $\boldsymbol{p}_{1:\boldsymbol{c}_{1:I}^{(1)}}^{*(1)}$.

5.2.2.1 Sampling the States

We use a separate Metropolis-Hastings kernel (see Sections 2.4.2 and 2.4.3) to sample the states $\mathbf{x}_{1:I,n}$ at each time n of each target i. We treat the states and measurements $(\mathbf{x}_{i,1:N}, \mathbf{y}_{i,1:N})$ of each target i separately as the states and measurements $(\mathbf{x}_{i,1:N}, \mathbf{y}_{i,1:N})$ given the parameter triple $\mathbf{p}_i$ or equivalently $\mathbf{p}^*_{\mathsf{c}_i}$ are independent of the states and measurements $(\mathbf{x}_{i',1:N}, \mathbf{y}_{i',1:N})$ of every other target $i' \neq i$ (see assumption A.1).

Time Step $n \in \{2, \dots, N-1\}$ Before we define the Metropolis-Hastings kernel for the first time step, let us first consider the time steps $n \in \{2, \dots, N-1\}$. We choose our proposal pdf for $\mathbf{x}_{i,n}$ for all $n \in \{2, \dots, N-1\}$ as the conditional pdf of the state $\mathbf{x}_{i,n}$ given every other state $\mathbf{x}_{i,1:n-1}$, $\mathbf{x}_{i,n+1:N}$ and the parameter triple $\mathbf{p}^*_{\mathsf{c}_i}$, that is, every random variable except the state $\mathbf{x}_{i,n}$ and the measurements $\mathbf{y}_{i,1:N}$, that is,

$$\begin{aligned} &g_{i,n}\Big(\boldsymbol{x}_{i,n} \,\Big|\, \boldsymbol{x}^{(k)}_{i,1:n-1}, \boldsymbol{x}^{(k-1)}_{i,n+1:N}, \boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}}\Big) \\ &\quad = f_{\mathbf{x}_{i,n}|\mathbf{x}_{i,1:n-1},\mathbf{x}_{i,n+1:N},\mathbf{p}^*_{\mathsf{c}_i}}\Big(\boldsymbol{x}_{i,n} \,\Big|\, \boldsymbol{x}^{(k)}_{i,1:n-1}, \boldsymbol{x}^{(k-1)}_{i,n+1:N}, \boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}}\Big). \end{aligned} \tag{5.66}$$

Using assumption A.2, it can be shown that the state $\mathbf{x}_{i,n}$ is conditionally independent of the states $\mathbf{x}_{i,N_{\mathrm{S},i}:n-2}$ and $\mathbf{x}_{i,n+2:N_{\mathrm{E},i}}$ given the previous state $\mathbf{x}_{i,n-1}$, the next state $\mathbf{x}_{i,n+1}$, and the parameter triple $\mathbf{p}^*_{\mathsf{c}_i}$, and therefore, we can simplify (5.66) to

$$\begin{aligned} &g_{i,n}\Big(\boldsymbol{x}_{i,n} \,\Big|\, \boldsymbol{x}^{(k)}_{i,1:n-1}, \boldsymbol{x}^{(k-1)}_{i,n+1:N}, \boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}}\Big) \\ &\quad = f_{\mathbf{x}_{i,n}|\mathbf{x}_{i,n-1},\mathbf{x}_{i,n+1},\mathbf{p}^*_{\mathsf{c}_i}}\Big(\boldsymbol{x}_{i,n} \,\Big|\, \boldsymbol{x}^{(k)}_{i,n-1}, \boldsymbol{x}^{(k-1)}_{i,n+1}, \boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}}\Big) \end{aligned} \tag{5.67}$$

Using Bayes' theorem, we obtain further

$$\begin{aligned} &g_{i,n}\Big(\boldsymbol{x}_{i,n} \,\Big|\, \boldsymbol{x}^{(k)}_{i,1:n-1}, \boldsymbol{x}^{(k-1)}_{i,n+1:N}, \boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}}\Big) \\ &\quad \propto f_{\mathbf{x}_{i,n+1}|\mathbf{x}_{i,n-1},\mathbf{x}_{i,n},\mathbf{p}^*_{\mathsf{c}_i}}\Big(\boldsymbol{x}^{(k-1)}_{i,n+1} \,\Big|\, \boldsymbol{x}^{(k)}_{i,n-1}, \boldsymbol{x}_{i,n}, \boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}}\Big) f_{\mathbf{x}_{i,n}|\mathbf{x}_{i,n-1},\mathbf{p}^*_{\mathsf{c}_i}}\Big(\boldsymbol{x}_{i,n} \,\Big|\, \boldsymbol{x}^{(k)}_{i,n-1}, \boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}}\Big). \end{aligned} \tag{5.68}$$

The first factor in (5.68) can be expressed as

$$\begin{aligned} &f_{\mathbf{x}_{i,n+1}|\mathbf{x}_{i,n-1},\mathbf{x}_{i,n},\mathbf{p}^*_{\mathsf{c}_i}}\Big(\boldsymbol{x}^{(k-1)}_{i,n+1} \,\Big|\, \boldsymbol{x}^{(k)}_{i,n-1}, \boldsymbol{x}_{i,n}, \boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}}\Big) \\ &\quad = f_{\mathbf{x}_{i,n+1}|\mathbf{x}_{i,n},\mathbf{p}^*_{\mathsf{c}_i}}\Big(\boldsymbol{x}^{(k-1)}_{i,n+1} \,\Big|\, \boldsymbol{x}_{i,n}, \boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}}\Big). \end{aligned} \tag{5.69}$$

where we have used assumption A.2. Using assumptions A.2 and A.5, it can be shown that $\mathbf{x}_{i,n+1}$ is conditionally independent of $\mathbf{D}^*_{\mathsf{c}_i}$ and $\bar{\mathbf{M}}^*_{\mathsf{c}_i}$ given $\mathbf{x}_{i,n}$ and $\mathbf{Q}^*_{\mathsf{c}_i}$, and therefore,

we have

$$f_{\mathbf{x}_{i,n+1}|\mathbf{x}_{i,n-1},\mathbf{x}_{i,n},\mathbf{p}^*_{\mathbf{c}_i}}\left(\boldsymbol{x}^{(k-1)}_{i,n+1}\,\middle|\,\boldsymbol{x}^{(k)}_{i,n-1},\boldsymbol{x}_{i,n},\boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}}\right) = f_{\mathbf{x}_{i,n+1}|\mathbf{x}_{i,n},\mathbf{Q}^*_{\mathbf{c}_i}}\left(\boldsymbol{x}^{(k-1)}_{i,n+1}\,\middle|\,\boldsymbol{x}_{i,n},\boldsymbol{Q}^{*(k-1)}_{c_i^{(k-1)}}\right). \tag{5.70}$$

Inserting (5.6) into (5.70), we obtain

$$\begin{aligned} &f_{\mathbf{x}_{i,n+1}|\mathbf{x}_{i,n-1},\mathbf{x}_{i,n},\mathbf{p}^*_{\mathbf{c}_i}}\left(\boldsymbol{x}^{(k-1)}_{i,n+1}\,\middle|\,\boldsymbol{x}^{(k)}_{i,n-1},\boldsymbol{x}_{i,n},\boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}}\right)\\ &\quad= \mathcal{N}\left(\boldsymbol{x}^{(k-1)}_{i,n+1};\boldsymbol{F}\boldsymbol{x}_{i,n},\boldsymbol{Q}^{*(k-1)}_{c_i^{(k-1)}}\right)\\ &\quad= \mathcal{N}\left(\boldsymbol{F}^{-1}\boldsymbol{x}^{(k-1)}_{i,n+1};\boldsymbol{x}_{i,n},\boldsymbol{F}^{-1}\boldsymbol{Q}^{*(k-1)}_{c_i^{(k-1)}}(\boldsymbol{F}^{-1})^{\mathrm{T}}\right). \end{aligned} \tag{5.71}$$

The second factor in (5.68) can be expressed as

$$f_{\mathbf{x}_{i,n}|\mathbf{x}_{i,n-1},\mathbf{p}^*_{\mathbf{c}_i}}\left(\boldsymbol{x}_{i,n}\,\middle|\,\boldsymbol{x}^{(k)}_{i,n-1},\boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}}\right) = \mathcal{N}\left(\boldsymbol{x}_{i,n};\boldsymbol{F}\boldsymbol{x}^{(k)}_{i,n-1},\boldsymbol{Q}^{*(k-1)}_{c_i^{(k-1)}}\right), \tag{5.72}$$

where we have also used (5.6). As the conjugate prior for the mean of a Gaussian distribution is the Gaussian distribution,[3] and as (5.72) is the prior for the mean of (5.71), it follows that we can simplify (5.68) to obtain the final expression for our proposal pdf

$$g_{i,n}\left(\boldsymbol{x}_{i,n}\,\middle|\,\boldsymbol{x}^{(k)}_{i,1:n-1},\boldsymbol{x}^{(k-1)}_{i,n+1:N},\boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}}\right) = \mathcal{N}\left(\boldsymbol{x}_{i,n};\boldsymbol{\mu}^{(k)}_{g_{i,n}},\boldsymbol{\Sigma}^{(k)}_{g_{i,n}}\right) \tag{5.73}$$

with

$$\begin{aligned} \boldsymbol{\mu}^{(k)}_{g_{i,n}} &= \left(\left(\boldsymbol{Q}^{*(k-1)}_{c_i^{(k-1)}}\right)^{-1} + \left(\boldsymbol{F}^{-1}\boldsymbol{Q}^{*(k-1)}_{c_i^{(k-1)}}(\boldsymbol{F}^{-1})^{\mathrm{T}}\right)^{-1}\right)^{-1}\\ &\quad\times\left(\left(\boldsymbol{Q}^{*(k-1)}_{c_i^{(k-1)}}\right)^{-1}\boldsymbol{F}\boldsymbol{x}^{(k)}_{i,n-1} + \left(\boldsymbol{F}^{-1}\boldsymbol{Q}^{*(k-1)}_{c_i^{(k-1)}}(\boldsymbol{F}^{-1})^{\mathrm{T}}\right)^{-1}\boldsymbol{F}^{-1}\boldsymbol{x}^{(k-1)}_{i,n+1}\right)\\ &= \left(\left(\boldsymbol{Q}^{*(k-1)}_{c_i^{(k-1)}}\right)^{-1} + \boldsymbol{F}^{\mathrm{T}}\left(\boldsymbol{Q}^{*(k-1)}_{c_i^{(k-1)}}\right)^{-1}\boldsymbol{F}\right)^{-1}\\ &\quad\times\left(\left(\boldsymbol{Q}^{*(k-1)}_{c_i^{(k-1)}}\right)^{-1}\boldsymbol{F}\boldsymbol{x}^{(k)}_{i,n-1} + \boldsymbol{F}^{\mathrm{T}}\left(\boldsymbol{Q}^{*(k-1)}_{c_i^{(k-1)}}\right)^{-1}\boldsymbol{x}^{(k-1)}_{i,n+1}\right) \end{aligned} \tag{5.74}$$

and

$$\begin{aligned} \boldsymbol{\Sigma}^{(k)}_{g_{i,n}} &= \left(\left(\boldsymbol{Q}^{*(k-1)}_{c_i^{(k-1)}}\right)^{-1} + \left(\boldsymbol{F}^{-1}\boldsymbol{Q}^{*(k-1)}_{c_i^{(k-1)}}(\boldsymbol{F}^{-1})^{\mathrm{T}}\right)^{-1}\right)^{-1}\\ &= \left(\left(\boldsymbol{Q}^{*(k-1)}_{c_i^{(k-1)}}\right)^{-1} + \boldsymbol{F}^{\mathrm{T}}\left(\boldsymbol{Q}^{*(k-1)}_{c_i^{(k-1)}}\right)^{-1}\boldsymbol{F}\right)^{-1}. \end{aligned} \tag{5.75}$$

Following Section 2.4.3, our target pdf is

$$f_{\mathbf{x}_{i,n}|\mathbf{x}_{i,1:n-1},\mathbf{x}_{i,n+1:N},\mathbf{x}_{\neg i,1:N},\mathbf{p}^*_{\mathbf{c}_{1:I}},\mathbf{c}_{1:I},\mathbf{y}_{1:I,1:N}}(\boldsymbol{x}_{i,n}\,|\,\boldsymbol{x}_{i,1:n-1},\boldsymbol{x}_{i,n+1:N},\boldsymbol{x}_{\neg i,1:N},\boldsymbol{p}^*_{\boldsymbol{c}_{1:I}},\boldsymbol{c}_{1:I},\boldsymbol{y}_{1:I,1:N}). \tag{5.76}$$

[3] If $\mathbf{p}\sim\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma}_1)$ and $\mathbf{x}_i|(\mathbf{p}=\boldsymbol{p})\sim\mathcal{N}(\boldsymbol{p},\boldsymbol{\Sigma}_2)$ for $i\in\{1,\dots,I\}$, then $\mathbf{p}|(\mathbf{x}_{1:I}=\boldsymbol{x}_{1:I})\sim\mathcal{N}(\boldsymbol{\mu}',\boldsymbol{\Sigma}'_1)$ with $\boldsymbol{\mu}'=(\boldsymbol{\Sigma}_1^{-1}+I\boldsymbol{\Sigma}_2^{-1})^{-1}\left(\boldsymbol{\Sigma}_1^{-1}\mu+\boldsymbol{\Sigma}_2^{-1}\sum_{i=1}^{I}\boldsymbol{x}_i\right)$ and $\boldsymbol{\Sigma}'_1=\left(\boldsymbol{\Sigma}_1^{-1}+I\boldsymbol{\Sigma}_2^{-1}\right)^{-1}$.

Using assumption A.1 and then assumptions A.2 and A.3, it can be shown that

$$\begin{aligned} & f_{\mathbf{x}_{i,n}|\mathbf{x}_{i,1:n-1},\mathbf{x}_{i,n+1:N},\mathbf{x}_{\neg i,1:N},\mathbf{p}^*_{c_{1:I}},\mathbf{c}_{1:I},\mathbf{y}_{1:I,1:N}}(\boldsymbol{x}_{i,n} \mid \boldsymbol{x}_{i,1:n-1}, \boldsymbol{x}_{i,n+1:N}, \boldsymbol{x}_{\neg i,1:N}, \boldsymbol{p}^*_{c_{1:I}}, \boldsymbol{c}_{1:I}, \boldsymbol{y}_{1:I,1:N}) \\ & \quad = f_{\mathbf{x}_{i,n}|\mathbf{x}_{i,1:n-1},\mathbf{x}_{i,n+1:N},\mathbf{p}^*_{c_i},\mathbf{y}_{i,1:N}}(\boldsymbol{x}_{i,n} \mid \boldsymbol{x}_{i,1:n-1}, \boldsymbol{x}_{i,n+1:N}, \boldsymbol{p}^*_{c_i}, \boldsymbol{y}_{i,1:N}) \\ & \quad = f_{\mathbf{x}_{i,n}|\mathbf{x}_{i,n-1},\mathbf{x}_{i,n+1},\mathbf{p}^*_{c_i},\mathbf{y}_{i,n}}(\boldsymbol{x}_{i,n} \mid \boldsymbol{x}_{i,n-1}, \boldsymbol{x}_{i,n+1}, \boldsymbol{p}^*_{c_i}, \boldsymbol{y}_{i,n}). \end{aligned} \tag{5.77}$$

Therefore, a candidate $\bar{\boldsymbol{x}}^{(k)}_{i,n}$, sampled from $g_{i,n}\left(\bar{\boldsymbol{x}}^{(k)}_{i,n} \,\middle|\, \boldsymbol{x}^{(k)}_{i,1:n-1}, \boldsymbol{x}^{(k-1)}_{i,n+1:N}, \boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}}\right)$, is accepted $\boldsymbol{x}^{(k)}_{i,n} = \bar{\boldsymbol{x}}^{(k)}_{i,n}$ with probability

$$P^{(k)}_{\mathrm{a},i,n} = \min\left\{1, \rho_n\left(\bar{\boldsymbol{x}}^{(k)}_{i,n}, \boldsymbol{x}^{(k-1)}_{i,n}, \boldsymbol{x}^{(k)}_{i,n-1}, \boldsymbol{x}^{(k-1)}_{i,n+1}, \boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}}, \boldsymbol{y}_{i,n}\right)\right\}, \tag{5.78}$$

where

$$\begin{aligned} & \rho_n\left(\bar{\boldsymbol{x}}^{(k)}_{i,n}, \boldsymbol{x}^{(k-1)}_{i,n}, \boldsymbol{x}^{(k)}_{i,n-1}, \boldsymbol{x}^{(k-1)}_{i,n+1}, \boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}}, \boldsymbol{y}_{i,n}\right) \\ & \quad = \frac{f_{\mathbf{x}_{i,n}|\mathbf{x}_{i,n-1},\mathbf{x}_{i,n+1},\mathbf{p}^*_{c_i},\mathbf{y}_{i,n}}\left(\bar{\boldsymbol{x}}^{(k)}_{i,n} \,\middle|\, \boldsymbol{x}^{(k)}_{i,n-1}, \boldsymbol{x}^{(k-1)}_{i,n+1}, \boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}}, \boldsymbol{y}_{i,n}\right)}{f_{\mathbf{x}_{i,n}|\mathbf{x}_{i,n-1},\mathbf{x}_{i,n+1},\mathbf{p}^*_{c_i},\mathbf{y}_{i,n}}\left(\boldsymbol{x}^{(k-1)}_{i,n} \,\middle|\, \boldsymbol{x}^{(k)}_{i,n-1}, \boldsymbol{x}^{(k-1)}_{i,n+1}, \boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}}, \boldsymbol{y}_{i,n}\right)} \\ & \qquad \times \frac{g_{i,n}\left(\boldsymbol{x}^{(k-1)}_{i,n} \,\middle|\, \boldsymbol{x}^{(k)}_{i,1:n-1}, \boldsymbol{x}^{(k-1)}_{i,n+1:N}, \boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}}\right)}{g_{i,n}\left(\bar{\boldsymbol{x}}^{(k)}_{i,n} \,\middle|\, \boldsymbol{x}^{(k)}_{i,1:n-1}, \boldsymbol{x}^{(k-1)}_{i,n+1:N}, \boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}}\right)}, \end{aligned} \tag{5.79}$$

or rejected $\boldsymbol{x}^{(k)}_{i,n} = \boldsymbol{x}^{(k-1)}_{i,n}$ with probability $1 - P^{(k)}_{\mathrm{a},i,n}$.

In order to simplify (5.79), we insert (5.67) to obtain

$$\begin{aligned} & \rho_n\left(\bar{\boldsymbol{x}}^{(k)}_{i,n}, \boldsymbol{x}^{(k-1)}_{i,n}, \boldsymbol{x}^{(k)}_{i,n-1}, \boldsymbol{x}^{(k-1)}_{i,n+1}, \boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}}, \boldsymbol{y}_{i,n}\right) \\ & \quad = \frac{f_{\mathbf{x}_{i,n}|\mathbf{x}_{i,n-1},\mathbf{x}_{i,n+1},\mathbf{p}^*_{c_i},\mathbf{y}_{i,n}}\left(\bar{\boldsymbol{x}}^{(k)}_{i,n} \,\middle|\, \boldsymbol{x}^{(k)}_{i,n-1}, \boldsymbol{x}^{(k-1)}_{i,n+1}, \boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}}, \boldsymbol{y}_{i,n}\right)}{f_{\mathbf{x}_{i,n}|\mathbf{x}_{i,n-1},\mathbf{x}_{i,n+1},\mathbf{p}^*_{c_i},\mathbf{y}_{i,n}}\left(\boldsymbol{x}^{(k-1)}_{i,n} \,\middle|\, \boldsymbol{x}^{(k)}_{i,n-1}, \boldsymbol{x}^{(k-1)}_{i,n+1}, \boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}}, \boldsymbol{y}_{i,n}\right)} \\ & \qquad \times \frac{f_{\mathbf{x}_{i,n}|\mathbf{x}_{i,n-1},\mathbf{x}_{i,n+1},\mathbf{p}^*_{c_i}}\left(\boldsymbol{x}^{(k-1)}_{i,n} \,\middle|\, \boldsymbol{x}^{(k)}_{i,n-1}, \boldsymbol{x}^{(k-1)}_{i,n+1}, \boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}}\right)}{f_{\mathbf{x}_{i,n}|\mathbf{x}_{i,n-1},\mathbf{x}_{i,n+1},\mathbf{p}^*_{c_i}}\left(\bar{\boldsymbol{x}}^{(k)}_{i,n} \,\middle|\, \boldsymbol{x}^{(k)}_{i,n-1}, \boldsymbol{x}^{(k-1)}_{i,n+1}, \boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}}\right)}. \end{aligned} \tag{5.80}$$

Before we continue with the simplification of (5.80), first note that

$$
\begin{aligned}
&\frac{f_{\mathsf{x}_{i,n}|\mathsf{x}_{i,n-1},\mathsf{x}_{i,n+1},\mathsf{p}^*_{\mathsf{c}_i},\mathsf{y}_{i,n}}\left(\boldsymbol{x}_{i,n}\,\middle|\,\boldsymbol{x}^{(k)}_{i,n-1},\boldsymbol{x}^{(k-1)}_{i,n+1},\boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}},\boldsymbol{y}_{i,n}\right)}{f_{\mathsf{x}_{i,n}|\mathsf{x}_{i,n-1},\mathsf{x}_{i,n+1},\mathsf{p}^*_{\mathsf{c}_i}}\left(\boldsymbol{x}_{i,n}\,\middle|\,\boldsymbol{x}^{(k)}_{i,n-1},\boldsymbol{x}^{(k-1)}_{i,n+1},\boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}}\right)}\\
&\quad=\frac{f_{\mathsf{x}_{i,n}|\mathsf{x}_{i,n-1},\mathsf{x}_{i,n+1},\mathsf{p}^*_{\mathsf{c}_i},\mathsf{y}_{i,n},\mathsf{M}_{i,n}}\left(\boldsymbol{x}_{i,n}\,\middle|\,\boldsymbol{x}^{(k)}_{i,n-1},\boldsymbol{x}^{(k-1)}_{i,n+1},\boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}},\boldsymbol{y}_{i,n},M_{i,n}\right)}{f_{\mathsf{x}_{i,n}|\mathsf{x}_{i,n-1},\mathsf{x}_{i,n+1},\mathsf{p}^*_{\mathsf{c}_i},\mathsf{M}_{i,n}}\left(\boldsymbol{x}_{i,n}\,\middle|\,\boldsymbol{x}^{(k)}_{i,n-1},\boldsymbol{x}^{(k-1)}_{i,n+1},\boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}},M_{i,n}\right)}\\
&\quad=\frac{f_{\mathsf{y}_{i,n}|\mathsf{M}_{i,n},\mathsf{x}_{i,n},\mathsf{x}_{i,n-1},\mathsf{x}_{i,n+1},\mathsf{p}^*_{\mathsf{c}_i}}\left(\boldsymbol{y}_{i,n}\,\middle|\,M_{i,n},\boldsymbol{x}_{i,n},\boldsymbol{x}^{(k)}_{i,n-1},\boldsymbol{x}^{(k-1)}_{i,n+1},\boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}}\right)}{f_{\mathsf{y}_{i,n}|\mathsf{M}_{i,n},\mathsf{x}_{i,n-1},\mathsf{x}_{i,n+1},\mathsf{p}^*_{\mathsf{c}_i}}\left(\boldsymbol{y}_{i,n}\,\middle|\,M_{i,n},\boldsymbol{x}^{(k)}_{i,n-1},\boldsymbol{x}^{(k-1)}_{i,n+1},\boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}}\right)}\\
&\qquad\times\frac{f_{\mathsf{x}_{i,n}|\mathsf{x}_{i,n-1},\mathsf{x}_{i,n+1},\mathsf{p}^*_{\mathsf{c}_i},\mathsf{M}_{i,n}}\left(\boldsymbol{x}_{i,n}\,\middle|\,\boldsymbol{x}^{(k)}_{i,n-1},\boldsymbol{x}^{(k-1)}_{i,n+1},\boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}},M_{i,n}\right)}{f_{\mathsf{x}_{i,n}|\mathsf{x}_{i,n-1},\mathsf{x}_{i,n+1},\mathsf{p}^*_{\mathsf{c}_i},\mathsf{M}_{i,n}}\left(\boldsymbol{x}_{i,n}\,\middle|\,\boldsymbol{x}^{(k)}_{i,n-1},\boldsymbol{x}^{(k-1)}_{i,n+1},\boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}},M_{i,n}\right)}\\
&\quad=\frac{f_{\mathsf{y}_{i,n}|\mathsf{M}_{i,n},\mathsf{x}_{i,n},\mathsf{x}_{i,n-1},\mathsf{x}_{i,n+1},\mathsf{p}^*_{\mathsf{c}_i},}\left(\boldsymbol{y}_{i,n}\,\middle|\,M_{i,n},\boldsymbol{x}_{i,n},\boldsymbol{x}^{(k)}_{i,n-1},\boldsymbol{x}^{(k-1)}_{i,n+1},\boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}}\right)}{f_{\mathsf{y}_{i,n}|\mathsf{M}_{i,n},\mathsf{x}_{i,n-1},\mathsf{x}_{i,n+1},\mathsf{p}^*_{\mathsf{c}_i}}\left(\boldsymbol{y}_{i,n}\,\middle|\,M_{i,n},\boldsymbol{x}^{(k)}_{i,n-1},\boldsymbol{x}^{(k-1)}_{i,n+1},\boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}}\right)},
\end{aligned}
\tag{5.81}
$$

where the first step can be shown using assumption A.6 and we have used Bayes' theorem in the second step. Inserting (5.81) into (5.80) twice, we further obtain

$$
\begin{aligned}
&\rho_n\left(\bar{\boldsymbol{x}}^{(k)}_{i,n},\boldsymbol{x}^{(k-1)}_{i,n},\boldsymbol{x}^{(k)}_{i,n-1},\boldsymbol{x}^{(k-1)}_{i,n+1},\boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}},\boldsymbol{y}_{i,n}\right)\\
&\quad=\frac{f_{\mathsf{y}_{i,n}|\mathsf{M}_{i,n},\mathsf{x}_{i,n},\mathsf{x}_{i,n-1},\mathsf{x}_{i,n+1},\mathsf{p}^*_{\mathsf{c}_i}}\left(\boldsymbol{y}_{i,n}\,\middle|\,M_{i,n},\bar{\boldsymbol{x}}^{(k)}_{i,n},\boldsymbol{x}^{(k)}_{i,n-1},\boldsymbol{x}^{(k-1)}_{i,n+1},\boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}}\right)}{f_{\mathsf{y}_{i,n}|\mathsf{M}_{i,n},\mathsf{x}_{i,n-1},\mathsf{x}_{i,n+1},\mathsf{p}^*_{\mathsf{c}_i}}\left(\boldsymbol{y}_{i,n}\,\middle|\,M_{i,n},\boldsymbol{x}^{(k)}_{i,n-1},\boldsymbol{x}^{(k-1)}_{i,n+1},\boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}}\right)}\\
&\qquad\times\frac{f_{\mathsf{y}_{i,n}|\mathsf{M}_{i,n},\mathsf{x}_{i,n-1},\mathsf{x}_{i,n+1},\mathsf{p}^*_{\mathsf{c}_i}}\left(\boldsymbol{y}_{i,n}\,\middle|\,M_{i,n},\boldsymbol{x}^{(k)}_{i,n-1},\boldsymbol{x}^{(k-1)}_{i,n+1},\boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}}\right)}{f_{\mathsf{y}_{i,n}|\mathsf{M}_{i,n},\mathsf{x}_{i,n},\mathsf{x}_{i,n-1},\mathsf{x}_{i,n+1},\mathsf{p}^*_{\mathsf{c}_i}}\left(\boldsymbol{y}_{i,n}\,\middle|\,M_{i,n},\boldsymbol{x}^{(k-1)}_{i,n},\boldsymbol{x}^{(k)}_{i,n-1},\boldsymbol{x}^{(k-1)}_{i,n+1},\boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}}\right)}\\
&\quad=\frac{f_{\mathsf{y}_{i,n}|\mathsf{M}_{i,n},\mathsf{x}_{i,n},\mathsf{x}_{i,n-1},\mathsf{x}_{i,n+1},\mathsf{p}^*_{\mathsf{c}_i}}\left(\boldsymbol{y}_{i,n}\,\middle|\,M_{i,n},\bar{\boldsymbol{x}}^{(k)}_{i,n},\boldsymbol{x}^{(k)}_{i,n-1},\boldsymbol{x}^{(k-1)}_{i,n+1},\boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}}\right)}{f_{\mathsf{y}_{i,n}|\mathsf{M}_{i,n},\mathsf{x}_{i,n},\mathsf{x}_{i,n-1},\mathsf{x}_{i,n+1},\mathsf{p}^*_{\mathsf{c}_i}}\left(\boldsymbol{y}_{i,n}\,\middle|\,M_{i,n},\boldsymbol{x}^{(k-1)}_{i,n},\boldsymbol{x}^{(k)}_{i,n-1},\boldsymbol{x}^{(k-1)}_{i,n+1},\boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}}\right)}.
\end{aligned}
\tag{5.82}
$$

It can be shown, using assumption A.3, that we can simplify 5.82 to

$$
\begin{aligned}
&\rho_n\left(\bar{\boldsymbol{x}}^{(k)}_{i,n},\boldsymbol{x}^{(k-1)}_{i,n},\boldsymbol{x}^{(k)}_{i,n-1},\boldsymbol{x}^{(k-1)}_{i,n+1},\boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}},\boldsymbol{y}_{i,n}\right)\\
&\quad=\frac{f_{\mathsf{y}_{i,n}|\mathsf{M}_{i,n},\mathsf{x}_{i,n},\mathsf{p}^*_{\mathsf{c}_i}}\left(\boldsymbol{y}_{i,n}\,\middle|\,M_{i,n},\bar{\boldsymbol{x}}^{(k)}_{i,n},\boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}}\right)}{f_{\mathsf{y}_{i,n}|\mathsf{M}_{i,n},\mathsf{x}_{i,n},\mathsf{p}^*_{\mathsf{c}_i}}\left(\boldsymbol{y}_{i,n}\,\middle|\,M_{i,n},\boldsymbol{x}^{(k-1)}_{i,n},\boldsymbol{p}^{*(k-1)}_{c_i^{(k-1)}}\right)}\\
&\quad=\frac{f_{\mathsf{y}_{i,n}|\mathsf{M}_{i,n},\mathsf{x}_{i,n},\mathsf{D}^*_{\mathsf{c}_i}}\left(\bar{\boldsymbol{y}}^{(k)}_{i,n}\,\middle|\,M_{i,n},\bar{\boldsymbol{x}}^{(k)}_{i,n},\boldsymbol{D}^{*(k-1)}_{c_i^{(k-1)}}\right)}{f_{\mathsf{y}_{i,n}|\mathsf{M}_{i,n},\mathsf{x}_{i,n},\mathsf{D}^*_{\mathsf{c}_i}}\left(\boldsymbol{y}^{(k-1)}_{i,n}\,\middle|\,M_{i,n},\boldsymbol{x}^{(k-1)}_{i,n},\boldsymbol{D}^{*(k-1)}_{c_i^{(k-1)}}\right)},
\end{aligned}
\tag{5.83}
$$

where we have used assumption A.7 in the second step. Finally, as (5.10) is a bijective

affine transformation, it follows that

$$\begin{aligned}
&\rho_n\Big(\bar{\boldsymbol{x}}_{i,n}^{(k)}, \boldsymbol{x}_{i,n}^{(k-1)}, \boldsymbol{x}_{i,n-1}^{(k)}, \boldsymbol{x}_{i,n+1}^{(k-1)}, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}, \boldsymbol{y}_{i,n}\Big) \\
&= \frac{f_{\mathbf{z}_{i,n}|\mathsf{M}_{i,n},\mathbf{x}_{i,n},\mathbf{D}_{c_i}^*}\Big(\bar{\boldsymbol{z}}_{i,n}^{(k)} \,\Big|\, M_{i,n}, \bar{\boldsymbol{x}}_{i,n}^{(k)}, \boldsymbol{D}_{c_i^{(k-1)}}^{*(k-1)}\Big)}{f_{\mathbf{z}_{i,n}|\mathsf{M}_{i,n},\mathbf{x}_{i,n},\mathbf{D}_{c_i}^*}\Big(\boldsymbol{z}_{i,n}^{(k-1)} \,\Big|\, M_{i,n}, \boldsymbol{x}_{i,n}^{(k-1)}, \boldsymbol{D}_{c_i^{(k-1)}}^{*(k-1)}\Big)} \\
&= \frac{f_{\mathbf{z}_{i,n}|\mathsf{M}_{i,n},\mathbf{D}_{c_i}^*}\Big(\bar{\boldsymbol{z}}_{i,n}^{(k)} \,\Big|\, M_{i,n}, \boldsymbol{D}_{c_i^{(k-1)}}^{*(k-1)}\Big)}{f_{\mathbf{z}_{i,n}|\mathsf{M}_{i,n},\mathbf{D}_{c_i}^*}\Big(\boldsymbol{z}_{i,n}^{(k-1)} \,\Big|\, M_{i,n}, \boldsymbol{D}_{c_i^{(k-1)}}^{*(k-1)}\Big)} \\
&= \frac{\prod_{m=1}^{M_{i,n}} \mathcal{N}\Big(\bar{\boldsymbol{z}}_{i,n,m}^{(k)}; \mathbf{0}, \boldsymbol{D}_{c_i^{(k-1)}}^{*(k-1)}\Big)}{\prod_{m=1}^{M_{i,n}} \mathcal{N}\Big(\boldsymbol{z}_{i,n,m}^{(k-1)}; \mathbf{0}, \boldsymbol{D}_{c_i^{(k-1)}}^{*(k-1)}\Big)},
\end{aligned} \tag{5.84}$$

with $\bar{\boldsymbol{z}}_{i,n,m}^{(k)} = \boldsymbol{R}(\bar{\boldsymbol{x}}_{i,n}^{(k)})^{\mathrm{T}}(\boldsymbol{y}_{i,n,m} - \boldsymbol{H}\bar{\boldsymbol{x}}_{i,n}^{(k)})$ and $\boldsymbol{z}_{i,n,m}^{(k-1)} = \boldsymbol{R}(\boldsymbol{x}_{i,n}^{(k-1)})^{\mathrm{T}}(\boldsymbol{y}_{i,n,m} - \boldsymbol{H}\boldsymbol{x}_{i,n}^{(k-1)})$, where we have used assumption A.9 in the second step as well as assumption A.8, (5.12), and that the $\mathbf{z}_{i,n,m}$ were assumed conditionally i.i.d. across n and m given $\mathbf{D}_i$ in the second step. Therefore, (5.78) becomes

$$P_{\mathrm{a},i,n}^{(k)} = \min\left\{1, \frac{\prod_{m=1}^{M_{i,n}} \mathcal{N}\Big(\bar{\boldsymbol{z}}_{i,n,m}^{(k)}; \mathbf{0}, \boldsymbol{D}_{c_i^{(k-1)}}^{*(k-1)}\Big)}{\prod_{m=1}^{M_{i,n}} \mathcal{N}\Big(\boldsymbol{z}_{i,n,m}^{(k-1)}; \mathbf{0}, \boldsymbol{D}_{c_i^{(k-1)}}^{*(k-1)}\Big)}\right\}. \tag{5.85}$$

Time Step 1 Similarly to (5.66), we choose our proposal pdf for the first time step as

$$g_{i,1}\Big(\boldsymbol{x}_{i,1} \,\Big|\, \boldsymbol{x}_{i,2:N}^{(k-1)}, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}\Big) = f_{\mathbf{x}_{i,1}|\mathbf{x}_{i,2:N},\mathbf{p}_{c_i}^*}\Big(\boldsymbol{x}_{i,1} \,\Big|\, \boldsymbol{x}_{i,2:N}^{(k-1)}, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}\Big). \tag{5.86}$$

We can simplify (5.86), using same derivation as for (5.68) but without the condition on $\mathbf{x}_{i,1:n-1}$ or $\mathbf{x}_{i,n-1}$ in all occurring pdfs, to

$$g_{i,1}\Big(\boldsymbol{x}_{i,1} \,\Big|\, \boldsymbol{x}_{i,2:N}^{(k-1)}, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}\Big) \propto f_{\mathbf{x}_{i,2}|\mathbf{x}_{i,1},\mathbf{p}_{c_i}^*}\Big(\boldsymbol{x}_{i,2}^{(k-1)} \,\Big|\, \boldsymbol{x}_{i,1}, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}\Big) f_{\mathbf{x}_{i,1}|\mathbf{p}_{c_i}^*}\Big(\boldsymbol{x}_{i,1} \,\Big|\, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}\Big). \tag{5.87}$$

The first factor in (5.87) is (5.72) for $n = 2$, and the second factor is

$$f_{\mathbf{x}_{i,1}|\mathbf{p}_{c_i}^*}\Big(\boldsymbol{x}_{i,1} \,\Big|\, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}\Big) = \mathcal{N}(\boldsymbol{x}_{i,1}; \mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{x}}), \tag{5.88}$$

where we have used assumption A.4 and (5.1). Similarly to (5.73), we can use the facts that (5.88) is the prior for the mean of the first factor in (5.87) and that the conjugate prior for the mean of a Gaussian distribution is the Gaussian distribution, and obtain

$$g_{i,1}\Big(\boldsymbol{x}_{i,1} \,\Big|\, \boldsymbol{x}_{i,2:N}^{(k-1)}, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}\Big) = \mathcal{N}\Big(\boldsymbol{x}_{i,1}; \boldsymbol{\mu}_{g_{i,1}}^{(k)}, \boldsymbol{\Sigma}_{g_{i,1}}^{(k)}\Big) \tag{5.89}$$

with

$$\begin{aligned}\boldsymbol{\mu}_{g_{i,1}}^{(k)} &= \left(\boldsymbol{\Sigma}_{\mathbf{x}}^{-1} + \left(\boldsymbol{F}^{-1}\boldsymbol{Q}_{c_i^{(k-1)}}^{*(k-1)}(\boldsymbol{F}^{-1})^{\mathrm{T}}\right)^{-1}\right)^{-1} \\ &\quad \times \left(\boldsymbol{\Sigma}_{\mathbf{x}}^{-1}\mathbf{0} + \left(\boldsymbol{F}^{-1}\boldsymbol{Q}_{c_i^{(k-1)}}^{*(k-1)}(\boldsymbol{F}^{-1})^{\mathrm{T}}\right)^{-1}\boldsymbol{F}^{-1}\boldsymbol{x}_{i,2}^{(k-1)}\right) \\ &= \left(\boldsymbol{\Sigma}_{\mathbf{x}}^{-1} + \boldsymbol{F}^{\mathrm{T}}\left(\boldsymbol{Q}_{c_i^{(k-1)}}^{*(k-1)}\right)^{-1}\boldsymbol{F}\right)^{-1}\boldsymbol{F}^{\mathrm{T}}\left(\boldsymbol{Q}_{c_i^{(k-1)}}^{*(k-1)}\right)^{-1}\boldsymbol{x}_{i,2}^{(k-1)} \end{aligned} \tag{5.90}$$

and

$$\begin{aligned}\boldsymbol{\Sigma}_{g_{i,1}}^{(k)} &= \left(\boldsymbol{\Sigma}_{\mathbf{x}}^{-1} + \left(\boldsymbol{F}^{-1}\boldsymbol{Q}_{c_i^{(k-1)}}^{*(k-1)}(\boldsymbol{F}^{-1})^{\mathrm{T}}\right)^{-1}\right)^{-1} \\ &= \left(\boldsymbol{\Sigma}_{\mathbf{x}}^{-1} + \boldsymbol{F}^{\mathrm{T}}\left(\boldsymbol{Q}_{c_i^{(k-1)}}^{*(k-1)}\right)^{-1}\boldsymbol{F}\right)^{-1}. \end{aligned} \tag{5.91}$$

The target pdf for the first time step is

$$f_{\mathbf{x}_{i,1}|\mathbf{x}_{i,2:N},\mathbf{x}_{\neg i,1:N},\mathbf{p}^*_{\mathbf{c}_{1:I}},\mathbf{c}_{1:I},\mathbf{y}_{1:I,1:N}}(\boldsymbol{x}_{i,1} \,|\, \boldsymbol{x}_{i,2:N}, \boldsymbol{x}_{\neg i,1:N}, \boldsymbol{p}^*_{\boldsymbol{c}_{1:I}}, \boldsymbol{c}_{1:I}, \boldsymbol{y}_{1:I,1:N}). \tag{5.92}$$

Using assumption A.1 and then assumptions A.2 and A.3, it can be shown that

$$\begin{aligned}&f_{\mathbf{x}_{i,1}|\mathbf{x}_{i,2:N},\mathbf{x}_{\neg i,1:N},\mathbf{p}^*_{\mathbf{c}_{1:I}},\mathbf{c}_{1:I},\mathbf{y}_{1:I,1:N}}(\boldsymbol{x}_{i,1} \,|\, \boldsymbol{x}_{i,2:N}, \boldsymbol{x}_{\neg i,1:N}, \boldsymbol{p}^*_{\boldsymbol{c}_{1:I}}, \boldsymbol{c}_{1:I}, \boldsymbol{y}_{1:I,1:N}) \\ &\quad = f_{\mathbf{x}_{i,1}|\mathbf{x}_{i,2:N},\mathbf{p}^*_{\mathbf{c}_i},\mathbf{y}_{i,1:N}}(\boldsymbol{x}_{i,1} \,|\, \boldsymbol{x}_{i,2:N}, \boldsymbol{p}^*_{c_i}, \boldsymbol{y}_{i,1:N}) \\ &\quad = f_{\mathbf{x}_{i,1}|\mathbf{x}_{i,2},\mathbf{p}^*_{\mathbf{c}_i},\mathbf{y}_{i,1}}(\boldsymbol{x}_{i,1} \,|\, \boldsymbol{x}_{i,2}, \boldsymbol{p}^*_{c_i}, \boldsymbol{y}_{i,1}). \end{aligned} \tag{5.93}$$

Therefore, a candidate $\bar{\boldsymbol{x}}_{i,1}^{(k)}$, sampled from $g_{i,1}\left(\bar{\boldsymbol{x}}_{i,1}^{(k)} \,\middle|\, \boldsymbol{x}_{i,2}^{(k-1)}, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}\right)$, is accepted $\boldsymbol{x}_{i,1}^{(k)} = \bar{\boldsymbol{x}}_{i,1}^{(k)}$ with probability

$$P_{\mathrm{a},i,1}^{(k)} = \min\left\{1, \rho_1\left(\bar{\boldsymbol{x}}_{i,1}^{(k)}, \boldsymbol{x}_{i,1}^{(k-1)}, \boldsymbol{x}_{i,2}^{(k-1)}, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}, \boldsymbol{y}_{i,1}\right)\right\}, \tag{5.94}$$

where

$$\begin{aligned}&\rho_1\left(\bar{\boldsymbol{x}}_{i,1}^{(k)}, \boldsymbol{x}_{i,1}^{(k-1)}, \boldsymbol{x}_{i,2}^{(k-1)}, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}, \boldsymbol{y}_{i,1}\right) \\ &\quad = \frac{f_{\mathbf{x}_{i,1}|\mathbf{x}_{i,2},\mathbf{p}^*_{\mathbf{c}_i},\mathbf{y}_{i,1}}\left(\bar{\boldsymbol{x}}_{i,1}^{(k)} \,\middle|\, \boldsymbol{x}_{i,2}^{(k-1)}, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}, \boldsymbol{y}_{i,1}\right) g_{i,1}\left(\boldsymbol{x}_{i,1}^{(k-1)} \,\middle|\, \boldsymbol{x}_{i,2}^{(k-1)}, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}\right)}{f_{\mathbf{x}_{i,1}|\mathbf{x}_{i,2},\mathbf{p}^*_{\mathbf{c}_i},\mathbf{y}_{i,1}}\left(\boldsymbol{x}_{i,1}^{(k-1)} \,\middle|\, \boldsymbol{x}_{i,2}^{(k-1)}, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}, \boldsymbol{y}_{i,1}\right) g_{i,1}\left(\bar{\boldsymbol{x}}_{i,1}^{(k)} \,\middle|\, \boldsymbol{x}_{i,2}^{(k-1)}, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}\right)}, \end{aligned} \tag{5.95}$$

or rejected $\boldsymbol{x}_{i,1}^{(k)} = \boldsymbol{x}_{i,1}^{(k-1)}$ with probability $1 - P_{\mathrm{a},i,1}^{(k)}$.

Using the same derivation as for (5.84) with $n = 1$ and without the condition on $\mathbf{x}_{i,1:n-1}$ or $\mathbf{x}_{i,n-1}$ in all occurring pdfs, we can simplify (5.95) to

$$\rho_1\left(\bar{\boldsymbol{x}}_{i,1}^{(k)}, \boldsymbol{x}_{i,1}^{(k-1)}, \boldsymbol{x}_{i,2}^{(k-1)}, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}, \boldsymbol{y}_{i,1}\right) = \frac{\prod_{m=1}^{M_{i,1}} \mathcal{N}\left(\bar{\boldsymbol{z}}_{i,1,m}^{(k)}; \mathbf{0}, \boldsymbol{D}_{c_i^{(k-1)}}^{*(k-1)}\right)}{\prod_{m=1}^{M_{i,1}} \mathcal{N}\left(\boldsymbol{z}_{i,1,m}^{(k-1)}; \mathbf{0}, \boldsymbol{D}_{c_i^{(k-1)}}^{*(k-1)}\right)}, \tag{5.96}$$

with $\bar{\boldsymbol{z}}_{i,1,m}^{(k)} = \boldsymbol{R}(\bar{\boldsymbol{x}}_{i,1}^{(k)})^{\mathrm{T}}(\boldsymbol{y}_{i,1,m} - \boldsymbol{H}\bar{\boldsymbol{x}}_{i,1}^{(k)})$ and $\boldsymbol{z}_{i,1,m}^{(k-1)} = \boldsymbol{R}(\boldsymbol{x}_{i,1}^{(k-1)})^{\mathrm{T}}(\boldsymbol{y}_{i,1,m} - \boldsymbol{H}\boldsymbol{x}_{i,1}^{(k-1)})$, and therefore (5.94) becomes

$$P_{\mathrm{a},i,1}^{(k)} = \min\left\{1, \frac{\prod_{m=1}^{M_{i,1}} \mathcal{N}\left(\bar{\boldsymbol{z}}_{i,1,m}^{(k)}; \boldsymbol{0}, \boldsymbol{D}_{c_i^{(k-1)}}^{*(k-1)}\right)}{\prod_{m=1}^{M_{i,1}} \mathcal{N}\left(\boldsymbol{z}_{i,1,m}^{(k-1)}; \boldsymbol{0}, \boldsymbol{D}_{c_i^{(k-1)}}^{*(k-1)}\right)}\right\}. \tag{5.97}$$

Time Step N For the last time step N, we choose our proposal pdf as

$$g_{i,N}\left(\boldsymbol{x}_{i,N} \,\middle|\, \boldsymbol{x}_{i,1:N-1}^{(k)}, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}\right) = f_{\mathbf{x}_{i,N}|\mathbf{x}_{i,1:N-1},\mathbf{p}_{\mathbf{c}_i}^*}\left(\boldsymbol{x}_{i,N} \,\middle|\, \boldsymbol{x}_{i,1:N-1}^{(k)}, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}\right). \tag{5.98}$$

Using assumption A.2, we obtain

$$g_{i,N}\left(\boldsymbol{x}_{i,N} \,\middle|\, \boldsymbol{x}_{i,1:N-1}^{(k)}, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}\right) = f_{\mathbf{x}_{i,N}|\mathbf{x}_{i,N-1},\mathbf{p}_{\mathbf{c}_i}^*}\left(\boldsymbol{x}_{i,N} \,\middle|\, \boldsymbol{x}_{i,N-1}^{(k)}, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}\right), \tag{5.99}$$

which is

$$g_{i,N}\left(\boldsymbol{x}_{i,N} \,\middle|\, \boldsymbol{x}_{i,1:N-1}^{(k)}, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}\right) = \mathcal{N}\left(\boldsymbol{x}_{i,N}; \boldsymbol{F}\boldsymbol{x}_{i,N-1}^{(k-1)}, \boldsymbol{Q}_{c_i^{(k-1)}}^{*(k-1)}\right), \tag{5.100}$$

according to (5.6).

Here, the target pdf is

$$f_{\mathbf{x}_{i,N}|\mathbf{x}_{i,1:N-1},\mathbf{x}_{\neg i,1:N},\mathbf{p}_{\mathbf{c}_{1:I}}^*,\mathbf{c}_{1:I},\mathbf{y}_{1:I,1:N}}(\boldsymbol{x}_{i,N} \,|\, \boldsymbol{x}_{i,1:N-1}, \boldsymbol{x}_{\neg i,1:N}, \boldsymbol{p}_{\boldsymbol{c}_{1:I}}^*, \boldsymbol{c}_{1:I}, \boldsymbol{y}_{1:I,1:N}). \tag{5.101}$$

Using assumption A.1 and then assumptions A.2 and A.3, it can be shown that

$$\begin{aligned} &f_{\mathbf{x}_{i,N}|\mathbf{x}_{i,1:N-1},\mathbf{x}_{\neg i,1:N},\mathbf{p}_{\mathbf{c}_{1:I}}^*,\mathbf{c}_{1:I},\mathbf{y}_{1:I,1:N}}(\boldsymbol{x}_{i,N} \,|\, \boldsymbol{x}_{i,1:N-1}, \boldsymbol{x}_{\neg i,1:N}, \boldsymbol{p}_{\boldsymbol{c}_{1:I}}^*, \boldsymbol{c}_{1:I}, \boldsymbol{y}_{1:I,1:N}) \\ &\quad = f_{\mathbf{x}_{i,N}|\mathbf{x}_{i,1:N-1},\mathbf{p}_{\mathbf{c}_i}^*,\mathbf{y}_{i,1:N}}(\boldsymbol{x}_{i,N} \,|\, \boldsymbol{x}_{i,1:N-1}, \boldsymbol{p}_{c_i}^*, \boldsymbol{c}_{1:I}, \boldsymbol{y}_{i,1:N}) \\ &\quad = f_{\mathbf{x}_{i,N}|\mathbf{x}_{i,N-1},\mathbf{p}_{\mathbf{c}_i}^*,\mathbf{y}_{i,N}}(\boldsymbol{x}_{i,N} \,|\, \boldsymbol{x}_{i,N-1}, \boldsymbol{p}_{c_i}^*, \boldsymbol{y}_{i,N}), \end{aligned} \tag{5.102}$$

Therefore, a candidate $\bar{\boldsymbol{x}}_{i,N}^{(k)}$, sampled from $g_{i,N}\left(\bar{\boldsymbol{x}}_{i,N}^{(k)} \,\middle|\, \boldsymbol{x}_{i,1:N-1}^{(k)}, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}\right)$, is accepted $\boldsymbol{x}_{i,N}^{(k)} = \bar{\boldsymbol{x}}_{i,N}^{(k)}$ with probability

$$P_{\mathrm{a},i,N}^{(k)} = \min\left\{1, \rho_N\left(\bar{\boldsymbol{x}}_{i,N}^{(k)}, \boldsymbol{x}_{i,N}^{(k-1)}, \boldsymbol{x}_{i,N-1}^{(k)}, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}, \boldsymbol{y}_{i,N}\right)\right\}, \tag{5.103}$$

where

$$\begin{aligned} &\rho_N\left(\bar{\boldsymbol{x}}_{i,N}^{(k)}, \boldsymbol{x}_{i,N}^{(k-1)}, \boldsymbol{x}_{i,N-1}^{(k)}, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}, \boldsymbol{y}_{i,N}\right) \\ &\quad = \frac{f_{\mathbf{x}_{i,N}|\mathbf{x}_{i,N-1},\mathbf{p}_{\mathbf{c}_i}^*,\mathbf{y}_{i,N}}\left(\bar{\boldsymbol{x}}_{i,N}^{(k)} \,\middle|\, \boldsymbol{x}_{i,N-1}^{(k)}, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}, \boldsymbol{y}_{i,N}\right) g_{i,N}\left(\boldsymbol{x}_{i,N}^{(k-1)} \,\middle|\, \boldsymbol{x}_{i,1:N-1}^{(k)}, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}\right)}{f_{\mathbf{x}_{i,N}|\mathbf{x}_{i,N-1},\mathbf{p}_{\mathbf{c}_i}^*,\mathbf{y}_{i,N}}\left(\boldsymbol{x}_{i,N}^{(k-1)} \,\middle|\, \boldsymbol{x}_{i,N-1}^{(k)}, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}, \boldsymbol{y}_{i,N}\right) g_{i,N}\left(\bar{\boldsymbol{x}}_{i,N}^{(k)} \,\middle|\, \boldsymbol{x}_{i,1:N-1}^{(k)}, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}\right)}, \end{aligned} \tag{5.104}$$

or rejected $\boldsymbol{x}_{i,N}^{(k)} = \boldsymbol{x}_{i,N}^{(k-1)}$ with probability $1 - P_{\mathrm{a},i,N}^{(k)}$.

Using the same derivation as for (5.84) with $n = N$ and without the condition on $\mathsf{x}_{i,n+1:N}$ or $\mathsf{x}_{i,n+1}$ in all occurring pdfs, we can simplify (5.104) to

$$\rho_N\left(\bar{\boldsymbol{x}}_{i,N}^{(k)}, \boldsymbol{x}_{i,N}^{(k-1)}, \boldsymbol{x}_{i,N-1}^{(k)}, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}, \boldsymbol{y}_{i,N}\right) = \frac{\prod_{m=1}^{M_{i,N}} \mathcal{N}\left(\bar{\boldsymbol{z}}_{i,N,m}^{(k)}; \boldsymbol{0}, \boldsymbol{D}_{c_i^{(k-1)}}^{*(k-1)}\right)}{\prod_{m=1}^{M_{i,N}} \mathcal{N}\left(\boldsymbol{z}_{i,N,m}^{(k-1)}; \boldsymbol{0}, \boldsymbol{D}_{c_i^{(k-1)}}^{*(k-1)}\right)} \tag{5.105}$$

with $\bar{\boldsymbol{z}}_{i,N,m}^{(k)} = \boldsymbol{R}(\bar{\boldsymbol{x}}_{i,N}^{(k)})^{\mathrm{T}}(\boldsymbol{y}_{i,N,m} - \boldsymbol{H}\bar{\boldsymbol{x}}_{i,N}^{(k)})$ and $\boldsymbol{z}_{i,N,m}^{(k-1)} = \boldsymbol{R}(\boldsymbol{x}_{i,N}^{(k-1)})^{\mathrm{T}}(\boldsymbol{y}_{i,N,m} - \boldsymbol{H}\boldsymbol{x}_{i,N}^{(k-1)})$, and therefore have

$$P_{\mathrm{a},i,N}^{(k)} = \min\left\{1, \frac{\prod_{m=1}^{M_{i,N}} \mathcal{N}\left(\bar{\boldsymbol{z}}_{i,N,m}^{(k)}; \boldsymbol{0}, \boldsymbol{D}_{c_i^{(k-1)}}^{*(k-1)}\right)}{\prod_{m=1}^{M_{i,N}} \mathcal{N}\left(\boldsymbol{z}_{i,N,m}^{(k-1)}; \boldsymbol{0}, \boldsymbol{D}_{c_i^{(k-1)}}^{*(k-1)}\right)}\right\}. \tag{5.106}$$

5.2.2.2 Sampling the Indicator Variables and Parameters

As for the particle Markov chain Monte Carlo Algorithm outlined in Section 4.1.2, we use Algorithm 4.3 to sample the indicator variables $\mathsf{c}_{1:I}$ and parameter triples $\mathsf{p}_{\mathsf{c}_{1:I}}^*$. More specifically, we sample $\mathsf{c}_i = c \in \left\{c_1^{(k)}, \ldots, c_{i-1}^{(k)}, c_{i+1}^{(k-1)}, \ldots, c_I^{(k-1)}\right\}$ with the conditional probability (see (4.16))

$$b_{i,c}^{(k)} = \mathbb{P}\left(\mathsf{c}_i = c \,\middle|\, \mathsf{c}_{\neg i} = \boldsymbol{c}_{\neg i}^{(k)}, \mathsf{p}_{\mathsf{c}_{\neg i}}^* = \boldsymbol{p}_{\boldsymbol{c}_{\neg i}^{(k)}}^{*(k-1)}, \mathsf{x}_{1:I,1:N} = \boldsymbol{x}_{1:I,1:N}^{(k)}, \mathsf{y}_{1:I,1:N} = \boldsymbol{y}_{1:I,1:N}\right), \tag{5.107}$$

for which an expression is obtained by inserting (5.7), (5.8), and (5.21) into (4.19), that is,

$$\begin{aligned} b_{i,c}^{(k)} \propto{}& \frac{1}{\alpha + I - 1}\left(\sum_{i'=1}^{i-1} \mathbb{1}\left(c_{i'}^{(k)} = c\right) + \sum_{i'=i+1}^{I} \mathbb{1}\left(c_{i'}^{(k-1)} = c\right)\right) \\ &\times \left(\prod_{n=1}^{N} \mathrm{Pois}\left(M_{i,n}; \bar{M}_c^{*(k-1)}\right) \prod_{m=1}^{M_{i,n}} \mathcal{N}\left(\boldsymbol{y}_{i,n,m}; \boldsymbol{H}\boldsymbol{x}_{i,n}^{(k)}, \boldsymbol{R}\left(\boldsymbol{x}_{i,n}^{(k)}\right)\boldsymbol{D}_c^{*(k-1)}\boldsymbol{R}\left(\boldsymbol{x}_{i,n}^{(k)}\right)^{\mathrm{T}}\right)\right) \\ &\times \mathcal{N}\left(\boldsymbol{x}_{i,1}^{(k)}; \boldsymbol{0}, \boldsymbol{\Sigma}_{\mathsf{x}}\right) \prod_{n'=2}^{N} \mathcal{N}\left(\boldsymbol{x}_{i,n'}^{(k)}; \boldsymbol{F}\boldsymbol{x}_{i',n'-1}^{(k)}, \boldsymbol{Q}_c^{*(k-1)}\right). \end{aligned} \tag{5.108}$$

Furthermore, we sample c_i as being distinct from all the other indicator variables, that is, $\mathsf{c}_i \notin \left\{c_1^{(k)}, \ldots, c_{i-1}^{(k)}, c_{i+1}^{(k-1)}, \ldots, c_I^{(k-1)}\right\}$, with the conditional probability (see (4.17))

$$\begin{aligned} b_{i,0}^{(k)} = \mathbb{P}\left(\mathsf{c}_i \notin \left\{c_1^{(k)}, \ldots, c_{i-1}^{(k)}, c_{i+1}^{(k-1)}, \ldots, c_I^{(k-1)}\right\} \,\middle|\, \mathsf{c}_{\neg i} = \boldsymbol{c}_{\neg i}^{(k)}, \mathsf{p}_{\mathsf{c}_{\neg i}}^* = \boldsymbol{p}_{\boldsymbol{c}_{\neg i}^{(k)}}^{*(k-1)}, \mathsf{x}_{1:I,1:N} = \boldsymbol{x}_{1:I,1:N}^{(k)},\right. \\ \left.\mathsf{y}_{1:I,1:N} = \boldsymbol{y}_{1:I,1:N}\right), \end{aligned} \tag{5.109}$$

for which an expression is obtained by inserting (5.7), (5.8), (5.21), (5.29), and (5.56) into (4.20), that is,

$$
\begin{aligned}
b_{i,0}^{(k)} \propto{}& \frac{\alpha}{\alpha + I - 1} \\
&\times \frac{\Gamma^{-1}\left(\tilde{d}_1^2; \alpha_{\mathrm{d},1}, \beta_{\mathrm{d},1}\right)}{\Gamma^{-1}\left(\tilde{d}_1^2; \alpha_{\mathrm{d},1} + \frac{1}{2}\sum_{n=1}^{N} M_{i,n}, \beta_{\mathrm{d},1} + \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{M_{i,n}}\left(z_{i,n,m,1}^{(k)}\right)^2\right)} \\
&\times \frac{\Gamma^{-1}\left(\tilde{d}_2^2; \alpha_{\mathrm{d},2}, \beta_{\mathrm{d},2}\right)}{\Gamma^{-1}\left(\tilde{d}_2^2; \alpha_{\mathrm{d},2} + \frac{1}{2}\sum_{n=1}^{N} M_{i,n}, \beta_{\mathrm{d},2} + \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{M_{i,n}}\left(z_{i,n,m,2}^{(k)}\right)^2\right)} \\
&\times \frac{\Gamma\left(\tilde{\tilde{M}}; \alpha_{\mathrm{M}}, \beta_{\mathrm{M}}\right)}{\Gamma\left(\tilde{\tilde{M}}; \alpha_{\mathrm{M}} + \sum_{n=1}^{N} M_{i,n}, \beta_{\mathrm{M}} + N\right)} \\
&\times \frac{\Gamma^{-1}\left(\tilde{q}_1^2; \alpha_{\mathrm{v},1}, \beta_{\mathrm{v},1}\right)}{\Gamma^{-1}\left(\tilde{q}_1^2; \alpha_{\mathrm{v},1} + N - 1, \beta_{\mathrm{v},1} + \frac{1}{2}\sum_{n=2}^{N}\left(\left(v_{i,n,1}^{(k)}\right)^2 + \left(v_{i,n,2}^{(k)}\right)^2\right)\right)} \\
&\times \frac{\Gamma^{-1}\left(\tilde{q}_2^2; \alpha_{\mathrm{v},2}, \beta_{\mathrm{v},2}\right)}{\Gamma^{-1}\left(\tilde{q}_2^2; \alpha_{\mathrm{v},2} + N - 1, \beta_{\mathrm{v},2} + \frac{1}{2}\sum_{n=2}^{N}\left(\left(v_{i,n,3}^{(k)}\right)^2 + \left(v_{i,n,4}^{(k)}\right)^2\right)\right)} \\
&\times \left(\prod_{n=1}^{N} \mathrm{Pois}\left(M_{i,n}; \tilde{\tilde{M}}\right) \prod_{m=1}^{M_{i,n}} \mathcal{N}\left(\boldsymbol{y}_{i,n,m}; \boldsymbol{H}\boldsymbol{x}_{i,n}^{(k)}, \boldsymbol{R}\left(\boldsymbol{x}_{i,n}^{(k)}\right)\tilde{\boldsymbol{D}}\boldsymbol{R}\left(\boldsymbol{x}_{i,n}^{(k)}\right)^{\mathrm{T}}\right)\right) \\
&\times \mathcal{N}\left(\boldsymbol{x}_{i,1}^{(k)}; \boldsymbol{0}, \boldsymbol{\Sigma}_{\mathbf{x}}\right) \prod_{n'=2}^{N} \mathcal{N}\left(\boldsymbol{x}_{i,n'}^{(k)}; \boldsymbol{F}\boldsymbol{x}_{i,n'-1}^{(k)}, \tilde{\boldsymbol{Q}}\right)
\end{aligned} \tag{5.110}
$$

for any $\tilde{\boldsymbol{p}} = \left(\tilde{\boldsymbol{D}}, \tilde{\boldsymbol{Q}}, \tilde{\tilde{M}}\right)$ with

$$
\tilde{\boldsymbol{D}} = \begin{bmatrix} \tilde{d}_1^2 & 0 \\ 0 & \tilde{d}_2^2 \end{bmatrix}, \tag{5.111}
$$

$$
\tilde{\boldsymbol{Q}} = \begin{bmatrix} \tilde{q}_1^2 \mathbf{I}_2 & \mathbf{0}_{2\times 2} \\ \mathbf{0}_{2\times 2} & \tilde{q}_2^2 \mathbf{I}_2 \end{bmatrix}, \tag{5.112}
$$

and $f_{\mathbf{p}_i}(\tilde{\boldsymbol{p}}) > 0$ (where $f_{\mathbf{p}_i}(\tilde{\boldsymbol{p}})$ is given by (5.33) with $\boldsymbol{p}_i$ replaced by $\tilde{\boldsymbol{p}}$). The samples of the indicator variables $c_i^{(k)}$ that are distinct from the other indicator variables are assigned a new value $c_i^{(k)} = \max\left\{\max_{i' \in \{1,\ldots,i-1\}} c_{i'}^{(k)}, c_{\max}^{(k-1)}\right\} + 1$. Furthermore, a parameter triple $\boldsymbol{p}_{c_i^{(k)}}^{*(k-1)}$ has to be sampled, which will be discussed presently.

The parameter triple samples $\boldsymbol{p}_c^{*(k)}$ are generated from the conditional pdf (5.65), that

is,

$$
\begin{aligned}
&f_{\mathbf{p}_c^*|\mathbf{x}_{1:I,1:n},\mathbf{c}_{1:I},\mathbf{p}^*_{\neg c},\mathbf{y}_{1:I,1:n}}\left(\boldsymbol{p}_c^{*(k)} \,\middle|\, \boldsymbol{x}_{1:I,1:N}^{(k)}, \boldsymbol{c}_{1:I}^{(k)}, \boldsymbol{p}_{\neg c}^{*(k)}, \boldsymbol{y}_{1:I,1:N}\right) \\
&\quad = \Gamma^{-1}\left(d_{c,1}^{*2}; \alpha_{\mathrm{d},1} + \frac{1}{2}\sum_{i:c_i=c}\sum_{n=1}^{N} M_{i,n}, \beta_{\mathrm{d},1} + \frac{1}{2}\sum_{i:c_i=c}\sum_{n=1}^{N}\sum_{m=1}^{M_{i,n}}\left(z_{i,n,m,1}^{(k)}\right)^2\right) \\
&\qquad \times \Gamma^{-1}\left(d_{c,2}^{*2}; \alpha_{\mathrm{d},2} + \frac{1}{2}\sum_{i:c_i=c}\sum_{n=1}^{N} M_{i,n}, \beta_{\mathrm{d},2} + \frac{1}{2}\sum_{i:c_i=c}\sum_{n=1}^{N}\sum_{m=1}^{M_{i,n}}\left(z_{i,n,m,2}^{(k)}\right)^2\right) \\
&\qquad \times \Gamma\left(\bar{M}_c^*; \alpha_{\mathrm{M}} + \sum_{i:c_i=c}\sum_{n=1}^{N} M_{i,n}, \beta_{\mathrm{M}} + \sum_{i:c_i=c} N\right) \\
&\qquad \times \Gamma^{-1}\left(q_{c,1}^{*2}; \alpha_{\mathrm{v},1} + (N-1)\sum_{i:c_i=c} 1, \beta_{\mathrm{v},1} + \frac{1}{2}\sum_{i:c_i=c}\sum_{n=2}^{N}\left(\left(v_{i,n,1}^{(k)}\right)^2 + \left(v_{i,n,2}^{(k)}\right)^2\right)\right) \\
&\qquad \times \Gamma^{-1}\left(q_{c,2}^{*2}; \alpha_{\mathrm{v},2} + (N-1)\sum_{i:c_i=c} 1, \beta_{\mathrm{v},2} + \frac{1}{2}\sum_{i:c_i=c}\sum_{n=2}^{N}\left(\left(v_{i,n,3}^{(k)}\right)^2 + \left(v_{i,n,4}^{(k)}\right)^2\right)\right) \qquad (5.113)
\end{aligned}
$$

with (5.57)–(5.62) and the obvious changes to include the sample index k. Note that the parameter triple $\boldsymbol{p}_{c_i^{(k)}}^{*(k-1)}$ for a newly generated class, as discussed above, is also sampled from (5.113), but with the newest currently available samples of the other random variables.

5.2.2.3 Summary

The pseudo-code for one iteration of the complete algorithm is provided in Algorithm 5.1.

5.2.3 Particle Markov Chain Monte Carlo Algorithm

In this section, we discuss the Particle Markov Chain Monte Carlo (PMCMC) algorithm for inference in Bayesian nonparametric state-space models from Section 4.1.2 applied to the extended target tracking model of this chapter. The algorithm can be directly applied by using (5.7), (5.8), (5.21), the conditional pdfs presented in Section 5.2.1, and one of the following two proposal pdfs.

Bootstrap-Type Proposal pdf The first proposal pdfs that we consider are (see (5.1))

$$g_1(\boldsymbol{x}_{i,1} \,|\, \boldsymbol{p}_{c_i}^*, \boldsymbol{y}_{i,1}) = \mathcal{N}(\boldsymbol{x}_{i,1}; \mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{x}}), \qquad (5.114)$$

and (see (5.6))

$$g(\boldsymbol{x}_{i,n} \,|\, \boldsymbol{x}_{i,n-1}, \boldsymbol{p}_{c_i}^*, \boldsymbol{y}_{i,n}) = \mathcal{N}(\boldsymbol{x}_{i,n}; \boldsymbol{F}\boldsymbol{x}_{i,n-1}, \boldsymbol{Q}_{c_i}^*). \qquad (5.115)$$

Algorithm 5.1 Gibbs sampler for extended target tracking

Input: $\boldsymbol{x}_{1:I,1:N}^{(k-1)}, \boldsymbol{c}_{1:I}^{(k-1)}, c_{\max}^{(k-1)}, \boldsymbol{p}_{\boldsymbol{c}_{1:I}^{(k-1)}}^{*(k-1)}, \boldsymbol{y}_{1:I,1:N}$

for all $i = 1, \ldots, I$ **do**
  sample $\bar{\boldsymbol{x}}_{i,1}^{(k)}$ from $g_{i,1}\left(\bar{\boldsymbol{x}}_{i,1}^{(k)} \,\middle|\, \boldsymbol{x}_{i,2}^{(k-1)}, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}\right)$ (see (5.89))
  set $\boldsymbol{x}_{i,1}^{(k)} = \bar{\boldsymbol{x}}_{i,1}^{(k)}$ with probability $P_{\mathrm{a},i,1}^{(k)}$ (see (5.97))
   else set $\boldsymbol{x}_{i,1}^{(k)} = \boldsymbol{x}_{i,1}^{(k-1)}$
  for all $n = 2, \ldots, N-1$ **do**
    sample $\bar{\boldsymbol{x}}_{i,n}^{(k)}$ from $g_{i,n}\left(\bar{\boldsymbol{x}}_{i,n}^{(k)} \,\middle|\, \boldsymbol{x}_{i,n-1}^{(k)}, \boldsymbol{x}_{i,n+1}^{(k-1)}, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}\right)$ (see (5.73))
    set $\boldsymbol{x}_{i,n}^{(k)} = \bar{\boldsymbol{x}}_{i,n}^{(k)}$ with probability $P_{\mathrm{a},i,n}^{(k)}$ (see (5.85))
    else set $\boldsymbol{x}_{i,n}^{(k)} = \boldsymbol{x}_{i,n}^{(k-1)}$
  end for
  sample $\bar{\boldsymbol{x}}_{i,N}^{(k)}$ from $g_{i,N}\left(\bar{\boldsymbol{x}}_{i,N}^{(k)} \,\middle|\, \boldsymbol{x}_{i,N-1}^{(k)}, \boldsymbol{p}_{c_i^{(k-1)}}^{*(k-1)}\right)$ (see (5.100))
  set $\boldsymbol{x}_{i,N}^{(k)} = \bar{\boldsymbol{x}}_{i,N}^{(k)}$ with probability $P_{\mathrm{a},i,N}^{(k)}$ (see (5.106))
   else set $\boldsymbol{x}_{i,N}^{(k)} = \boldsymbol{x}_{i,N}^{(k-1)}$
end for
for all $i = 1, \ldots, I$ **do**
  sample $c_i^{(k)} = c$ with probability $b_{i,c}^{(k)}$ (see (5.108) and (5.110))
   for all $c \in \left\{0, c_1^{(k)}, \ldots, c_{i-1}^{(k)}, c_{i+1}^{(k-1)}, \ldots, c_I^{(k-1)}\right\}$
  if $c_i^{(k)} = 0$
    set $c_i^{(k)} = \max\left\{\max_{i' \in \{1,\ldots,i-1\}} c_{i'}^{(k)}, c_{\max}^{(k-1)}\right\} + 1$
    sample $\boldsymbol{p}_{c_i^{(k)}}^{*(k-1)}$ from (5.113)
end for
set $c_{\max}^{(k)} = \max\left\{\max_{i \in \{1,\ldots,I\}} c_i^{(k)}, c_{\max}^{(k-1)}\right\}$
for all $c \in \left\{c_1^{(k)}, \ldots, c_I^{(k)}\right\}$ **do**
  sample $\boldsymbol{p}_c^{*(k)}$ from (5.113)
end for

Output: $\boldsymbol{x}_{1:I,1:N}^{(k)}, \boldsymbol{c}_{1:I}^{(k)}, c_{\max}^{(k)}, \boldsymbol{p}_{\boldsymbol{c}_{1:I}^{(k)}}^{*(k)}$

with $\boldsymbol{p}_{c_i}^* = (\boldsymbol{D}_{c_i}^*, \boldsymbol{Q}_{c_i}^*, \bar{M}_{c_i}^*)$. Note that $g_1(\boldsymbol{x}_{i,1} \,|\, \boldsymbol{p}_{c_i}^*, \boldsymbol{y}_{i,1})$ and $g(\boldsymbol{x}_{i,n} \,|\, \boldsymbol{x}_{i,n-1}, \boldsymbol{p}_{c_i}^*, \boldsymbol{y}_{i,n})$ do not involve the measurements $\boldsymbol{y}_{i,n}$. A particle filter for a state-space model that uses the pdf of the state transition model as its proposal pdf, as we have here, is known as a bootstrap filter [43].

Auxiliary-Type Proposal pdf To refine the bootstrap-type proposal pdf, we can incorporate the measurements $\mathbf{y}_{i,n}$ of each time step in the proposal pdf. We do this by using the method of [44] while assuming a circular extent. The circular extent is assumed because the orientation of the true elliptical extent is unknown and depends on the state $\mathbf{x}_{i,n}$, which we are sampling. This leads us to defining the proposal pdf for the first state as [44]

$$g_1(\boldsymbol{x}_{i,1} \,|\, \boldsymbol{p}_{c_i}^*, \boldsymbol{y}_{i,1}) = \mathcal{N}(\boldsymbol{x}_{i,1}; \boldsymbol{\mu}_{g_1}, \boldsymbol{\Sigma}_{g_1}), \tag{5.116}$$

with $\boldsymbol{p}_{c_i}^* = (\boldsymbol{D}_{c_i}^*, \boldsymbol{Q}_{c_i}^*, \bar{M}_{c_i}^*)$,

$$\boldsymbol{\mu}_{g_1} = \boldsymbol{H}^T \frac{1}{M_{i,1}} \sum_{m=1}^{M_{i,1}} \boldsymbol{y}_{i,1,m}, \tag{5.117}$$

$$\boldsymbol{\Sigma}_{g_1} = \frac{d_{c_i,\text{avg}}^{*2}}{M_{i,1}} \begin{bmatrix} \mathbf{I}_2 & \mathbf{0}_{2,2} \\ \mathbf{0}_{2,2} & \mathbf{0}_{2,2} \end{bmatrix} + q_{c_i,2}^{*2} \begin{bmatrix} \mathbf{0}_{2,2} & \mathbf{0}_{2,2} \\ \mathbf{0}_{2,2} & \mathbf{I}_2 \end{bmatrix}, \tag{5.118}$$

and $d_{c_i,\text{avg}}^{*2} = d_{c_i,1}^* d_{c_i,2}^*$, where we have chosen $d_{c_i,\text{avg}}^{*2}$ such that a circle with radius $d_{c_i,\text{avg}}^*$ has the same area as an ellipse with half axis lengths $d_{c_i,1}^*$ and $d_{c_i,2}^*$. Further, the proposal pdf for every other state is defined as

$$g(\boldsymbol{x}_{i,n} \,|\, \boldsymbol{x}_{i,n-1}, \boldsymbol{p}_{c_i}^*, \boldsymbol{y}_{i,n}) = \mathcal{N}(\boldsymbol{x}_{i,n}; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) \tag{5.119}$$

with

$$\boldsymbol{\mu}_g = \boldsymbol{F}\boldsymbol{x}_{i,n-1} + \frac{q_{c_i,1}^{*2}}{q_{c_i,1}^{*2} + \frac{d_{c_i,\text{avg}}^{*2}}{M_{i,n}}} \boldsymbol{H}^T \left(\frac{1}{M_{i,1}} \sum_{m=1}^{M_{i,n}} \boldsymbol{y}_{i,n,m} - \boldsymbol{F}\boldsymbol{x}_{i,n-1} \right), \tag{5.120}$$

$$\boldsymbol{\Sigma}_g = \boldsymbol{Q}_{c_i}^* - q_{c_i,1}^{*2} \frac{q_{c_i,1}^{*2}}{q_{c_i,1}^{*2} + \frac{d_{c_i,\text{avg}}^{*2}}{M_{i,n}}} \begin{bmatrix} \mathbf{I}_2 & \mathbf{0}_{2,2} \\ \mathbf{0}_{2,2} & \mathbf{0}_{2,2} \end{bmatrix}. \tag{5.121}$$

Note that compared to the bootstrap-type proposal pdf, this proposal pdf has its mean shifted towards the mean of the measurements and its variance is reduced, which is especially beneficial in the case of a large number of measurements. A particle filter for a state-space model that uses the observations in the proposal pdf, as we have here, is known as an auxiliary particle filter [45].

5.2.4 Sequential Monte Carlo Algorithm

In this section, we discuss the SMC algorithm of Section 4.1.3 applied to the extended target tracking model of this chapter. That algorithm, too, can be directly applied using (5.7), (5.8), (5.21), the conditional pdfs presented in Section 5.2.1, and one of the proposal pdfs discussed in Section 5.2.3 to generate the intermediate state particles.

As discussed in Section 4.1.3.4, one drawback of directly using the pdfs presented in Section 5.2.1 is that we require data from the current and all previous time steps. That is, the required data and the computational complexity grow with n. However, for the statistical model of this chapter, it is possible to summarize the required data by a recursively calculated sufficient statistic with constant dimensionality, to be discussed presently, which results in a constant computational complexity at each time step n.

For this statistical model, it follows from (5.56) and (5.65) that we obtain the simplifications (4.40) and (4.41) with $\mathbf{s}_{i,n} = (\mathbf{s}_{i,n,1}, \mathbf{s}_{i,n,2}) \in \mathbb{R}^{6\times 6}$, where

$$\mathbf{s}_{i,n,1} = \begin{bmatrix} n \\ \sum_{n'=1}^{n} \mathsf{M}_{i,n'} \\ \sum_{n'=1}^{n} \sum_{m=1}^{\mathsf{M}_{i,n'}} \mathsf{z}_{i,n',m,1}^2 \\ \sum_{n'=1}^{n} \sum_{m=1}^{\mathsf{M}_{i,n'}} \mathsf{z}_{i,n',m,2}^2 \\ \sum_{n'=2}^{n} \left(\mathsf{v}_{i,n',1}^2 + \mathsf{v}_{i,n',2}^2 \right) \\ \sum_{n'=2}^{n} \left(\mathsf{v}_{i,n',3}^2 + \mathsf{v}_{i,n',4}^2 \right) \end{bmatrix} \tag{5.122}$$

and

$$\mathbf{s}_{i,n,2} = \mathbf{s}_{i,n,1}. \tag{5.123}$$

These sufficient statistics can be calculated recursively according to

$$\mathbf{s}_{i,n,1} = \begin{bmatrix} 1 \\ \mathsf{M}_{i,1} \\ \sum_{m=1}^{\mathsf{M}_{i,1}} \mathsf{z}_{i,1,m,1}^2 \\ \sum_{m=1}^{\mathsf{M}_{i,1}} \mathsf{z}_{i,1,m,2}^2 \\ 0 \\ 0 \end{bmatrix} + \sum_{n'=2}^{n} \begin{bmatrix} 1 \\ \mathsf{M}_{i,n'} \\ \sum_{m=1}^{\mathsf{M}_{i,n'}} \mathsf{z}_{i,n',m,1}^2 \\ \sum_{m=1}^{\mathsf{M}_{i,n'}} \mathsf{z}_{i,n',m,2}^2 \\ \mathsf{v}_{i,n',1}^2 + \mathsf{v}_{i,n',2}^2 \\ \mathsf{v}_{i,n',3}^2 + \mathsf{v}_{i,n',4}^2 \end{bmatrix} = \mathbf{s}_{i,1,1} + \sum_{n'=2}^{n} \mathbf{s}_{\Delta,i,n',1} = \mathbf{s}_{i,n-1,1} + \mathbf{s}_{\Delta,i,n,1} \tag{5.124}$$

for $n \geq 2$, where

$$\mathbf{s}_{i,1,1} = \begin{bmatrix} 1 \\ \mathsf{M}_{i,1} \\ \sum_{m=1}^{\mathsf{M}_{i,1}} \mathsf{z}_{i,1,m,1}^2 \\ \sum_{m=1}^{\mathsf{M}_{i,1}} \mathsf{z}_{i,1,m,2}^2 \\ 0 \\ 0 \end{bmatrix} \tag{5.125}$$

and

$$\mathbf{s}_{\Delta,i,n,1} = \begin{bmatrix} 1 \\ \mathsf{M}_{i,n} \\ \sum_{m=1}^{\mathsf{M}_{i,n}} \mathsf{z}_{i,n,m,1}^2 \\ \sum_{m=1}^{\mathsf{M}_{i,n}} \mathsf{z}_{i,n,m,2}^2 \\ \mathsf{v}_{i,n,1}^2 + \mathsf{v}_{i,n,2}^2 \\ \mathsf{v}_{i,n,3}^2 + \mathsf{v}_{i,n,4}^2 \end{bmatrix}. \tag{5.126}$$

Therefore, we define $u_1(\mathbf{x}_{i,1}, \mathbf{y}_{i,1})$ and $u(\mathbf{s}_{i,n-1}, \mathbf{x}_{i,n}, \mathbf{y}_{i,n})$ (see Section 4.1.3.4) such that

$$\mathbf{s}_{i,1} = u_1(\mathbf{x}_{i,1}, \mathbf{y}_{i,1}) = (\mathbf{s}_{i,1,1}, \mathbf{s}_{i,1,1}) \tag{5.127}$$

and

$$\mathbf{s}_{i,n} = u(\mathbf{s}_{i,n-1}, \mathbf{x}_{i,n}, \mathbf{y}_{i,n}) = (\mathbf{s}_{i,n-1,1} + \mathbf{s}_{\Delta,i,n,1}, \mathbf{s}_{i,n-1,1} + \mathbf{s}_{\Delta,i,n,1}) \tag{5.128}$$

for $n \geq 2$.

5.3 Performance Evaluation

In this section, we will investigate the performance of the MC algorithms outlined in Sections 5.2.2–5.2.4 for joint tracking and clustering based on the statistical model described in Section 5.1. In particular, in Section 5.3.1 we will discuss the convergence of the algorithms, and in Section 5.3.2 we will investigate the performance gain due to joint tracking and clustering.

5.3.1 Algorithm Convergence

We first investigate the convergence of the MC algorithms discussed in Sections 5.2.2–5.2.4.

5.3.1.1 Data Sets

We use $T = 10$ synthetic data sets sampled from the statistical model discussed in Section 5.1. The data sets are samples of the measurements $\mathbf{y}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}}$ for all targets $i \in \{1, \ldots, I\}$. Each data set is based on $I = 10$ targets that are observed over 5 time steps with starting time $N_{\mathrm{S},i} = 1$ and ending time $N_{\mathrm{E},i} = 5$ for all targets $i \in \{1, \ldots, I\}$. Furthermore, we generate at least one measurement at the starting time, that is, $M_{i,N_{\mathrm{S},i}}^{(t)} \geq 1$ for all targets $i \in \{1, \ldots, I\}$ and data sets $t \in \{1, \ldots, T\}$. The hyperparameters are chosen as $\alpha_{\mathrm{DP}} = 1$, $\alpha_{\mathrm{d},1} = \alpha_{\mathrm{d},2} = 3$, $\beta_{\mathrm{d},1} = \beta_{\mathrm{d},2} = 50$, $\alpha_{\mathrm{v},1} = \alpha_{\mathrm{v},2} = 3$, $\beta_{\mathrm{v},1} = \beta_{\mathrm{v},2} = 50$, $\alpha_{\mathrm{M}} = 3$, $\beta_{\mathrm{M}} = 0.6$, $\Delta T = 1$, and $\boldsymbol{\Sigma}_{\mathbf{x}} = \mathrm{diag}([100^2\ 100^2\ 10^2\ 10^2])$ for all data sets; they are assumed to be known by all algorithms. Note that[4] $\mathbb{E}(\mathbf{D}_c^*) = \mathrm{diag}([5^2\ 5^2])$, $\mathbb{E}(\mathbf{Q}_c^*) = \mathrm{diag}([5^2\ 5^2\ 5^2\ 5^2])$, and[5] $\mathbb{E}(\bar{\mathsf{M}}_c^*)) = 5$ for this hyperparameter choice. One of the data sets is illustrated in Figure 5.5, where we have plotted the measurements of all targets at all time steps.

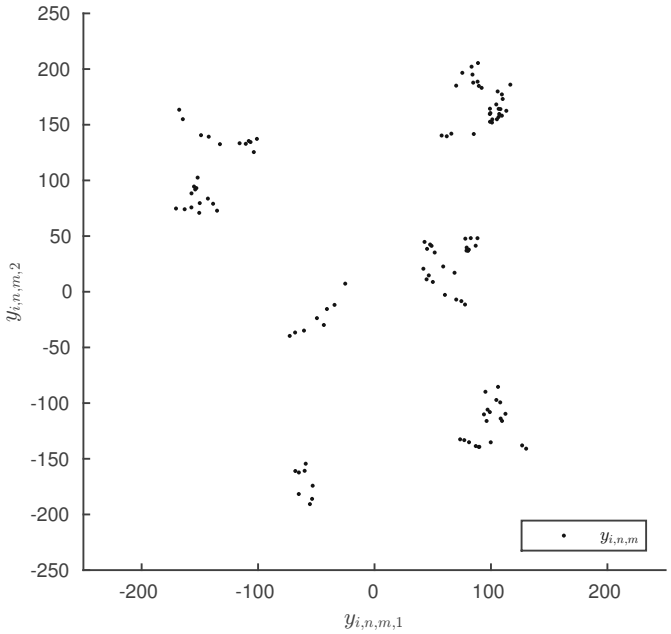

Fig. 5.5: *A data set for extended target tracking.*

5.3.1.2 Algorithms

The MC algorithms are applied to the data sets in order to generate K samples of the states $\mathbf{x}_{i,n}$ and parameter triple $\mathbf{p}_i = (\mathbf{D}_i, \mathbf{Q}_i, \bar{\mathsf{M}}_i)$ for all targets $i \in \{1, \ldots, I\}$, conditioned

[4] The expectation of the inverse gamma distribution $\Gamma^{-1}(\alpha, \beta)$ with $\alpha > 1$ is $\frac{\beta}{\alpha - 1}$.
[5] The expectation of the gamma distribution $\Gamma(\alpha, \beta)$ is $\frac{\alpha}{\beta}$.

on the measurements $\mathbf{y}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}}$ for all $i \in \{1,\ldots,I\}$.

We compare the reference algorithm discussed in Section 5.2.2, the PMCMC algorithm discussed in Section 5.2.3 with $L = 10$, and the SMC algorithm using the Metropolis-Hastings algorithm to sample the states discussed in Section 5.2.4. Note that we do not consider the SMC algorithm using the Gibbs sampling algorithm to sample the states as it is too computationally intensive for larger numbers of samples K. Furthermore, we do not consider the SMC algorithm using the bootstrap-type proposal pdf as it performs significantly worse than the other algorithms. This is mainly because the particles at the starting time of each target are sampled from (5.114) using the uninformative covariance matrix $\mathbf{\Sigma}_{\mathbf{x}}$, resulting in poor approximations. The PMCMC algorithm using the bootstrap-type proposal pdf does not suffer from this problem to a similar extent due to the deterministically set particle within the particle Gibbs sampler. We denote the reference algorithm as RA, the PMCMC algorithm using the bootstrap-type proposal pdf as PMCMC-BS, the PMCMC algorithm using the auxiliary-type proposal pdf as PMCMC-AUX, and the SMC algorithm using the auxiliary-type proposal pdf and the Metropolis-Hastings algorithm to sample the states as SMC-MH-AUX.

The MCMC algorithms, that is, the RA, PMCMC-BS, and PMCMC-AUX algorithms, are initialized as follows. For each data set $t \in \{1,\ldots,T\}$ and each target $i \in \{1,\ldots,I\}$, the target's position initialization $\boldsymbol{x}_{i,n,1:2}^{(t,1)}$ is chosen as the mean of the measurements $\boldsymbol{y}_{i,n}^{(t)}$ for all time steps $n \in \{N_{\mathrm{S},i},\ldots,N_{\mathrm{E},i}\}$ with a non-zero number of measurements, that is, $M_{i,n}^{(t)} \geq 1$, and chosen using linear inter- or extrapolation for all time steps $n \in \{N_{\mathrm{S},i},\ldots,N_{\mathrm{E},i}\}$ with $M_{i,n}^{(t)} = 0$. The target's velocity initialization is calculated according to $\boldsymbol{x}_{i,n,3:4}^{(t,1)} = \frac{1}{\Delta T}\left(\boldsymbol{x}_{i,n+1,1:2}^{(t,1)} - \boldsymbol{x}_{i,n,1:2}^{(t,1)}\right)$ for the time steps $n \in \{N_{\mathrm{S},i},\ldots,N_{\mathrm{E},i}-1\}$, and set to $\boldsymbol{x}_{i,n,3:4}^{(t,1)} = \boldsymbol{x}_{i,n-1,3:4}^{(t,1)}$ for the ending time $n = N_{\mathrm{E},i}$. We initialize the class indicator variables according to $c_i^{(t,1)} = i$ and the parameter triples according to $\boldsymbol{p}_{c^{(t,1)}}^{*(t,1)} = (\mathbb{E}(\mathbf{D}_c^*), \mathbb{E}(\mathbf{Q}_c^*), \mathbb{E}(\bar{\mathsf{M}}_c^*))$.

As the RA, PMCMC-BS, and PMCMC-AUX algorithms are MCMC algorithms, we execute them only once for each data set in order to generate the maximum number of samples $K_{\max}$ that we will require for our analysis. We can then evaluate the algorithm performance for different values of $K \in \{1,\ldots,K_{\max}\}$ by simply using the first K of our generated samples. However, as the SMC-MH-AUX algorithm is an SMC algorithm, we have to run it for each K separately.

5.3.1.3 Performance Metrics

For our performance evaluation, we will use, for all targets $i \in \{1, \dots, I\}$, the state estimates

$$\hat{\boldsymbol{x}}_{i,n}^{(t,K)} = \frac{1}{K} \sum_{k=1}^{K} \boldsymbol{x}_{i,n}^{(t,k)} \tag{5.129}$$

for all $n \in \{N_{\mathrm{S},i}, \dots, N_{\mathrm{E},i}\}$ and $i \in \{1, \dots, I\}$ and the parameter estimates

$$\hat{\boldsymbol{D}}_i^{(t,K)} = \frac{1}{K} \sum_{k=1}^{K} \boldsymbol{D}_i^{(t,k)}, \quad \hat{\boldsymbol{Q}}_i^{(t,K)} = \frac{1}{K} \sum_{k=1}^{K} \boldsymbol{Q}_i^{(t,k)}, \quad \hat{\bar{M}}_i^{(t,K)} = \frac{1}{K} \sum_{k=1}^{K} \bar{M}_i^{(t,k)}, \tag{5.130}$$

which are approximations of the respective posterior means. To evaluate algorithm convergence, we would ideally use the mean square errors (MSEs) of the state and parameter estimates (5.129) and (5.130) relative to their respective true posterior mean — subsequently called posterior mean MSEs — as a function of the number of samples K, that is,

$$\mathrm{MSE}_{\check{\boldsymbol{x}}_{1:2}}^{(K)} = \frac{1}{TI} \sum_{t=1}^{T} \sum_{i=1}^{I} \frac{1}{N_{\mathrm{TS},i}} \sum_{n=N_{\mathrm{S},i}}^{N_{\mathrm{E},i}} \left\| \hat{\boldsymbol{x}}_{i,n,1:2}^{(t,K)} - \check{\boldsymbol{x}}_{i,n,1:2}^{(t)} \right\|_2^2, \tag{5.131}$$

$$\mathrm{MSE}_{\check{\boldsymbol{x}}_{3:4}}^{(K)} = \frac{1}{TI} \sum_{t=1}^{T} \sum_{i=1}^{I} \frac{1}{N_{\mathrm{TS},i}} \sum_{n=N_{\mathrm{S},i}}^{N_{\mathrm{E},i}} \left\| \hat{\boldsymbol{x}}_{i,n,3:4}^{(t,K)} - \check{\boldsymbol{x}}_{i,n,3:4}^{(t)} \right\|_2^2, \tag{5.132}$$

and

$$\mathrm{MSE}_{\check{\boldsymbol{D}}}^{(K)} = \frac{1}{TI} \sum_{t=1}^{T} \sum_{i=1}^{I} \left\| \hat{\boldsymbol{D}}_i^{(t,K)} - \check{\boldsymbol{D}}_i^{(t)} \right\|_{\mathrm{F}}^2, \tag{5.133}$$

$$\mathrm{MSE}_{\check{\boldsymbol{Q}}}^{(K)} = \frac{1}{TI} \sum_{t=1}^{T} \sum_{i=1}^{I} \left\| \hat{\boldsymbol{Q}}_i^{(t,K)} - \check{\boldsymbol{Q}}_i^{(t)} \right\|_{\mathrm{F}}^2, \tag{5.134}$$

$$\mathrm{MSE}_{\check{\bar{M}}}^{(K)} = \frac{1}{TI} \sum_{t=1}^{T} \sum_{i=1}^{I} \left(\hat{\bar{M}}_i^{(t,K)} - \check{\bar{M}}_i^{(t)} \right)^2, \tag{5.135}$$

where $\|\cdot\|_2$ is the Euclidean norm,[6] $\|\cdot\|_F$ is the Frobenius norm,[7][8] $N_{\mathrm{TS},i} = N_{\mathrm{E},i} - N_{\mathrm{S},i} + 1$ is the number of time steps, and $\check{\boldsymbol{x}}_{i,n}^{(t)}$, $\check{\boldsymbol{D}}_i^{(t)}$, $\check{\boldsymbol{Q}}_i^{(t)}$, $\check{\bar{M}}_i^{(t)}$ are the true posterior means.

[6] The Euclidean norm of a vector $\boldsymbol{x} = [x_1 \ \dots \ x_M]^{\mathrm{T}} \in \mathbb{R}^M$ is $\|\boldsymbol{x}\|_2 = \left(\sum_{m=1}^{M} |x_m|^2 \right)^{\frac{1}{2}}$.

[7] The Frobenius norm of a matrix $\boldsymbol{A} = (a_{m,n})_{m \in \{1,\dots,M\}, n \in \{1,\dots,N\}} \in \mathbb{R}^{M \times N}$ is $\|\boldsymbol{A}\|_F = \left(\sum_{m=1}^{M} \sum_{n=1}^{N} |a_{m,n}|^2 \right)^{\frac{1}{2}}$.

[8] Calculating the MSEs of the complete matrices $\hat{\boldsymbol{D}}_i^{(t,K)}$ in (5.133) and $\hat{\boldsymbol{Q}}_i^{(t,K)}$ in (5.134) instead of the MSEs of their individual entries is reasonable because $\hat{\boldsymbol{D}}_i^{(t,K)}$ and $\hat{\boldsymbol{Q}}_i^{(t,K)}$ are diagonal matrices where, due to our hyperparameter choice, the diagonal entries are distributed according to the same distribution.

However, as the true posterior means are unknown, we will approximate them according to

$$\check{\boldsymbol{x}}_{i,n}^{(t)} \approx \hat{\boldsymbol{x}}_{i,n}^{(t,\check{K})}, \quad \check{\boldsymbol{D}}_i^{(t)} \approx \hat{\boldsymbol{D}}_i^{(t,\check{K})}, \quad \check{\boldsymbol{Q}}_i^{(t)} \approx \hat{\boldsymbol{Q}}_i^{(t,\check{K})}, \quad \check{\bar{M}}_i^{(t)} \approx \hat{\bar{M}}_i^{(t,\check{K})}, \tag{5.136}$$

where we use a sufficiently large number of samples $\check{K}$, and the samples generated with an MCMC algorithm that is proven to converge to the true posterior mean. We choose the number of samples $\check{K}$ and the algorithm generating these samples such that the MSEs relative to the respective ground truth $\boldsymbol{x}_{i,n}^{(t)}$, $\boldsymbol{D}_i^{(t)}$, $\boldsymbol{Q}_i^{(t)}$, $\bar{M}_i^{(t)}$ — subsequently called ground truth MSEs — seem to have converged. The ground truth MSEs are given by

$$\mathrm{MSE}_{\boldsymbol{x}_{1:2}}^{(K)} = \frac{1}{TI}\sum_{t=1}^{T}\sum_{i=1}^{I}\frac{1}{N_{\mathrm{TS},i}^{(t)}}\sum_{n=N_{\mathrm{S},i}^{(t)}}^{N_{\mathrm{E},i}^{(t)}}\left\|\hat{\boldsymbol{x}}_{i,n,1:2}^{(t,K)} - \boldsymbol{x}_{i,n,1:2}^{(t)}\right\|_2^2, \tag{5.137}$$

$$\mathrm{MSE}_{\boldsymbol{x}_{3:4}}^{(K)} = \frac{1}{TI}\sum_{t=1}^{T}\sum_{i=1}^{I}\frac{1}{N_{\mathrm{TS},i}^{(t)}}\sum_{n=N_{\mathrm{S},i}^{(t)}}^{N_{\mathrm{E},i}^{(t)}}\left\|\hat{\boldsymbol{x}}_{i,n,3:4}^{(t,K)} - \boldsymbol{x}_{i,n,3:4}^{(t)}\right\|_2^2, \tag{5.138}$$

and

$$\mathrm{MSE}_{\boldsymbol{D}}^{(K)} = \frac{1}{TI}\sum_{t=1}^{T}\sum_{i=1}^{I}\left\|\hat{\boldsymbol{D}}_i^{(t,K)} - \boldsymbol{D}_i^{(t)}\right\|_{\mathrm{F}}^2, \tag{5.139}$$

$$\mathrm{MSE}_{\boldsymbol{Q}}^{(K)} = \frac{1}{TI}\sum_{t=1}^{T}\sum_{i=1}^{I}\left\|\hat{\boldsymbol{Q}}_i^{(t,K)} - \boldsymbol{Q}_i^{(t)}\right\|_{\mathrm{F}}^2, \tag{5.140}$$

$$\mathrm{MSE}_{\bar{M}}^{(K)} = \frac{1}{TI}\sum_{t=1}^{T}\sum_{i=1}^{I}\left(\hat{\bar{M}}_i^{(t,K)} - \bar{M}_i^{(t)}\right)^2. \tag{5.141}$$

Note that the ground truth $\boldsymbol{x}_{i,n}^{(t)}$, $\boldsymbol{D}_i^{(t)}$, $\boldsymbol{Q}_i^{(t)}$, $\bar{M}_i^{(t)}$ is known as we are using synthetic data sets.

5.3.1.4 Results

In order to evaluate the convergence of our algorithms, we first evaluate the ground truth MSEs, defined in (5.137)–(5.141), as a function of the number of samples K. The results are illustrated in Figure 5.6 for the ground truth MSEs of the state estimates and in Figure 5.7 for the ground truth MSEs of the parameter estimates. While all algorithms seem to converge, the PMCMC-AUX algorithm tends to converge significantly faster than the other algorithms. Further, the PMCMC-AUX algorithm converges faster than the PMCMC-BS algorithm, which suggests that the auxiliary-type proposal pdf performs better than the bootstrap-type pdf. The superior performance of the auxiliary-type proposal pdf is not surprising, considering that the auxiliary-type proposal pdf uses the previous state and the measurements, instead of just the previous state.

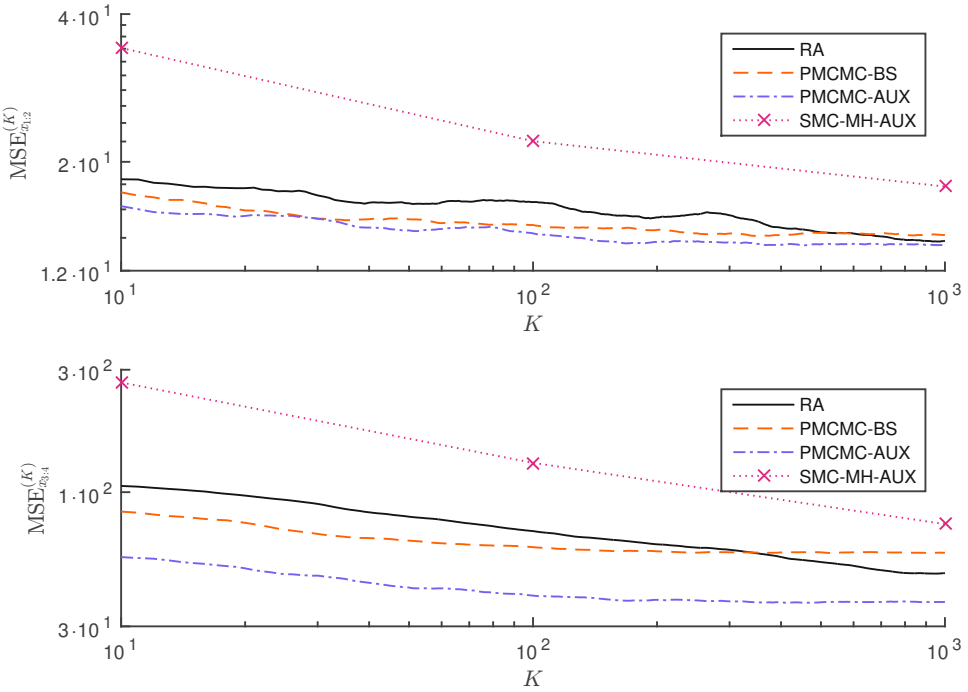

Fig. 5.6: *Ground truth MSEs of the state estimates as a function of the number of samples.*

While Figures 5.6 and 5.7 suggest that all algorithms converge in terms of the ground truth MSEs, we still need to investigate if all algorithms converge to the true posterior mean. Therefore, in Figures 5.8 and 5.9, we illustrate the (approximated) posterior mean MSEs, defined in (5.131)–(5.135), as a function of the number of samples K. Here, for the approximations (5.136), we have chosen $\check{K} = 10^4$ samples generated by the PMCMC-AUX algorithm. Note that the PMCMC-AUX algorithm is an MCMC algorithm that is proven to converge to the true posterior mean. The fact that the posterior mean MSEs in Figures 5.8 and 5.9 decrease with an increasing number of samples K suggests that all algorithms converge to the true posterior mean.

5.3.2 Tracking and Parameter Estimation Performance

In this section, we will evaluate the tracking and parameter estimation performance gain due to target clustering. We will consider cases where the targets belong to different classes, but the number of classes and the class parameters are unknown. In particular, we will compare inference based on the statistical model discussed in Section 5.1, which includes target clustering, to inference based on a simplified statistical model, which does not include target clustering and will be discussed in Section 5.3.2.1.

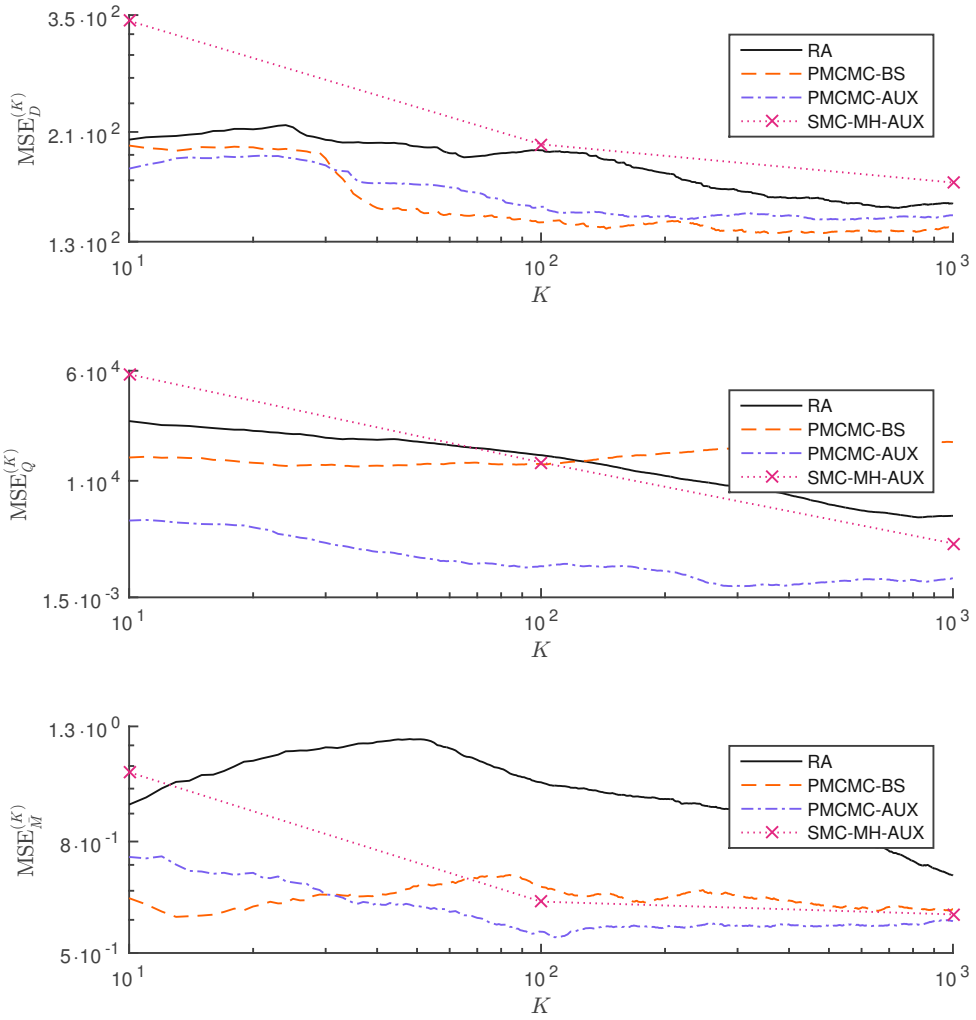

Fig. 5.7: *Ground truth MSEs of the parameter estimates as a function of the number of samples.*

5.3.2.1 Algorithms

Similarly to Section 5.3.1, we apply MC algorithms to different data sets in order to generate K samples of the states $\mathbf{x}_{i,n}$ and the parameter triple $\mathbf{p}_i = (\mathbf{D}_i, \mathbf{Q}_i, \bar{\mathsf{M}}_i)$ for all targets $i \in \{1, \ldots, I\}$, conditioned on the measurements $\mathbf{y}_{i,N_{\mathrm{S},i}:N_{\mathrm{E},i}}$ for all $i \in \{1, \ldots, I\}$.

For the case with clustering, we use the PMCMC algorithm discussed in Section 5.2.3 with the auxiliary-type proposal pdf, which is based on the statistical model discussed in Section 5.1, as it was the fastest converging algorithm in Section 5.3.1. We will use $K = 10^3$ and $L = 10$, as these parameters provided accurate results.

For the case without clustering, we use the same algorithm as for the case with clustering, but with small modifications. More specifically, we adapt the PMCMC algorithm discussed in Section 5.2.3 to a simplified version of the statistical model discussed in Section 5.1. The simplified statistical model uses the same motion model (see Section 5.1.1)

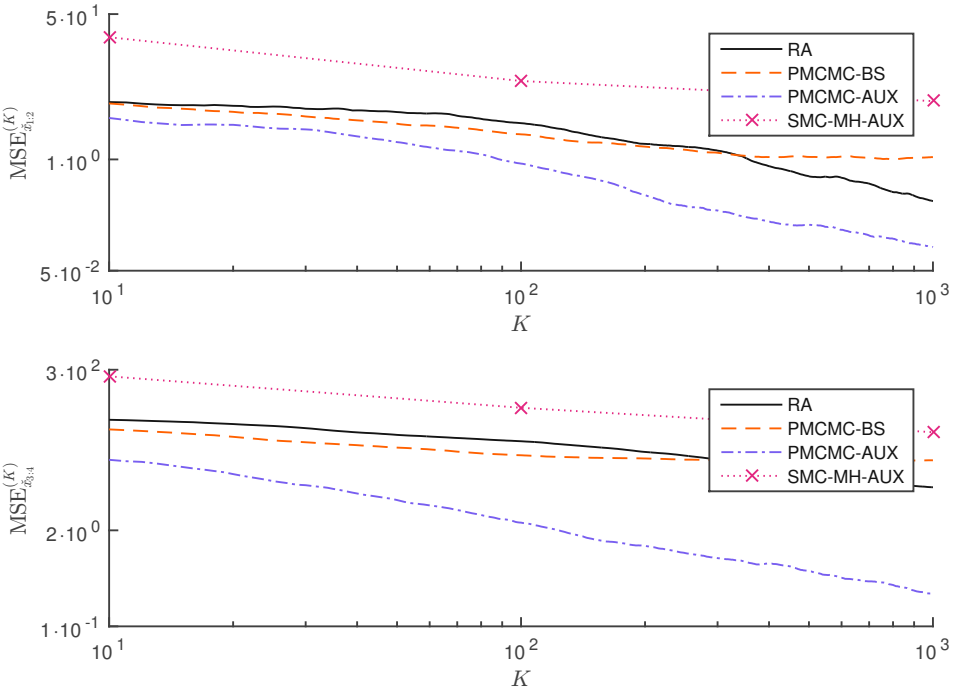

Fig. 5.8: *Posterior mean MSEs of the state estimates as a function of the number of samples.*

and measurement model (see Section 5.1.2), but the parameter triple $\mathbf{p}_i$ of the individual targets $i \in \{1, \ldots, I\}$ are i.i.d. according to the base pdf $f_G(\boldsymbol{p}_i)$ (see (5.33)) instead of distributed according to the the Dirichlet process (see (5.32)). In the statistical model discussed in Section 5.1, the class parameter triple $\mathbf{p}^*_c$ are i.i.d. according to the base pdf $f_G(\boldsymbol{p}^*_c)$ (see (3.13)), and therefore it follows that the simplified statistical model is equivalent to the statistical model discussed in Section 5.1 with each target deterministically assigned to a separate class, that is, $\mathsf{c}_i = i$ for all targets $i \in \{1, \ldots, I\}$. This allows us to easily adapt the PMCMC algorithm discussed in Section 5.2.3 to the simplified statistical model by deterministically setting $c_i^{(k)} = i$ for all targets $i \in \{1, \ldots, I\}$ instead of obtaining the $c_i^{(k)}$ by sampling c_i.

5.3.2.2 Performance Metrics

To evaluate the tracking and parameter estimation performance gain due to target clustering, we use the clustering gains (CGs) in decibels (dB)

$$\mathrm{CG}^{(K)}_{\boldsymbol{x}_{1:2}} = 10 \log_{10} \left(\frac{\mathrm{MSE}^{(K)}_{\boldsymbol{x}_{1:2},\mathrm{C}}}{\mathrm{MSE}^{(K)}_{\boldsymbol{x}_{1:2},\mathrm{NC}}} \right), \tag{5.142}$$

$$\mathrm{CG}^{(K)}_{\boldsymbol{x}_{3:4}} = 10 \log_{10} \left(\frac{\mathrm{MSE}^{(K)}_{\boldsymbol{x}_{3:4},\mathrm{C}}}{\mathrm{MSE}^{(K)}_{\boldsymbol{x}_{3:4},\mathrm{NC}}} \right), \tag{5.143}$$

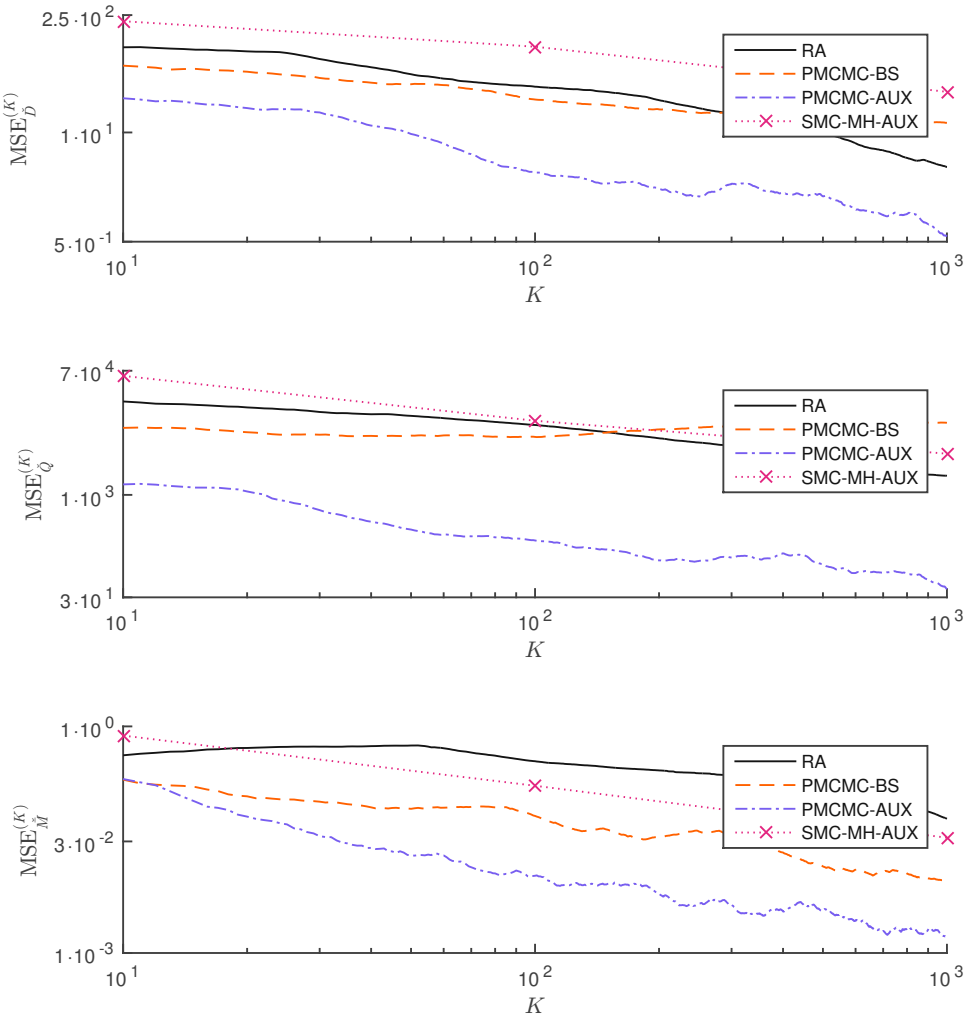

Fig. 5.9: *Posterior mean MSEs of the parameter estimates as a function of the number of samples.*

and

$$\mathrm{CG}_{\boldsymbol{D}}^{(K)} = 10\log_{10}\left(\frac{\mathrm{MSE}_{\boldsymbol{D},\mathrm{C}}^{(K)}}{\mathrm{MSE}_{\boldsymbol{D},\mathrm{NC}}^{(K)}}\right), \tag{5.144}$$

$$\mathrm{CG}_{\boldsymbol{Q}}^{(K)} = 10\log_{10}\left(\frac{\mathrm{MSE}_{\boldsymbol{Q},\mathrm{C}}^{(K)}}{\mathrm{MSE}_{\boldsymbol{Q},\mathrm{NC}}^{(K)}}\right), \tag{5.145}$$

$$\mathrm{CG}_{\bar{M}}^{(K)} = 10\log_{10}\left(\frac{\mathrm{MSE}_{\bar{M},\mathrm{C}}^{(K)}}{\mathrm{MSE}_{\bar{M},\mathrm{NC}}^{(K)}}\right), \tag{5.146}$$

where $\mathrm{MSE}_{\boldsymbol{x}_{1:2},\mathrm{C}}^{(K)}$, $\mathrm{MSE}_{\boldsymbol{x}_{3:4},\mathrm{C}}^{(K)}$, $\mathrm{MSE}_{\boldsymbol{D},\mathrm{C}}^{(K)}$, $\mathrm{MSE}_{\boldsymbol{Q},\mathrm{C}}^{(K)}$, $\mathrm{MSE}_{\bar{M},\mathrm{C}}$ are the ground truth MSEs using the samples generated with the PMCMC algorithm discussed in Section 5.2.3, that is, with clustering, and $\mathrm{MSE}_{\boldsymbol{x}_{1:2},\mathrm{NC}}^{(K)}$, $\mathrm{MSE}_{\boldsymbol{x}_{3:4},\mathrm{NC}}^{(K)}$, $\mathrm{MSE}_{\boldsymbol{D},\mathrm{NC}}^{(K)}$, $\mathrm{MSE}_{\boldsymbol{Q},\mathrm{NC}}^{(K)}$, $\mathrm{MSE}_{\bar{M},\mathrm{NC}}$ are the ground truth MSEs using the samples generated with the PMCMC algorithm adapted to the simplified statistical model discussed in Section 5.3.2.1, that is, without clustering. We

recall that the ground truth MSEs were defined in (5.137)–(5.140).

As parameter estimation is more challenging for targets that only exist for a small number of time steps $N_{\mathrm{TS},i}$, the benefit from target clustering is typically larger in that case. To illustrate this in the case of time series of different lengths, we additionally calculate the ground truth MSEs and CGs of all targets with a given number of time steps N_{TS}, that is, for all targets i with $N_{\mathrm{TS},i} = N_{\mathrm{TS}}$. More specifically, we calculate the ground truth MSEs of all targets with N_{TS} time steps according to (see (5.137)–(5.140))

$$\mathrm{MSE}_{\boldsymbol{x}_{1:2}}^{(N_{\mathrm{TS}},K)} = \frac{1}{TI}\sum_{t=1}^{T}\sum_{i:N_{\mathrm{TS},i}=N_{\mathrm{TS}}}\frac{1}{N_{\mathrm{TS}}}\sum_{n=N_{\mathrm{S},i}}^{N_{\mathrm{E},i}}\left\|\hat{\boldsymbol{x}}_{i,n,1:2}^{(t,K)} - \boldsymbol{x}_{i,n,1:2}^{(t)}\right\|_2^2, \tag{5.147}$$

$$\mathrm{MSE}_{\boldsymbol{x}_{3:4}}^{(N_{\mathrm{TS}},K)} = \frac{1}{TI}\sum_{t=1}^{T}\sum_{i:N_{\mathrm{TS},i}=N_{\mathrm{TS}}}\frac{1}{N_{\mathrm{TS}}}\sum_{n=N_{\mathrm{S},i}}^{N_{\mathrm{E},i}}\left\|\hat{\boldsymbol{x}}_{i,n,3:4}^{(t,K)} - \boldsymbol{x}_{i,n,3:4}^{(t)}\right\|_2^2, \tag{5.148}$$

and

$$\mathrm{MSE}_{\boldsymbol{D}}^{(N_{\mathrm{TS}},K)} = \frac{1}{TI}\sum_{t=1}^{T}\sum_{i:N_{\mathrm{TS},i}=N_{\mathrm{TS}}}\left\|\hat{\boldsymbol{D}}_i^{(t,K)} - \boldsymbol{D}_i^{(t)}\right\|_{\mathrm{F}}^2, \tag{5.149}$$

$$\mathrm{MSE}_{\boldsymbol{Q}}^{(N_{\mathrm{TS}},K)} = \frac{1}{TI}\sum_{t=1}^{T}\sum_{i:N_{\mathrm{TS},i}=N_{\mathrm{TS}}}\left\|\hat{\boldsymbol{Q}}_i^{(t,K)} - \boldsymbol{Q}_i^{(t)}\right\|_{\mathrm{F}}^2, \tag{5.150}$$

$$\mathrm{MSE}_{\bar{M}}^{(N_{\mathrm{TS}},K)} = \frac{1}{TI}\sum_{t=1}^{T}\sum_{i:N_{\mathrm{TS},i}=N_{\mathrm{TS}}}\left(\hat{\bar{M}}_i^{(t,K)} - \bar{M}_i^{(t)}\right)^2, \tag{5.151}$$

and the CGs of all targets with N_{TS} time steps (see (5.142)–(5.146))

$$\mathrm{CG}_{\boldsymbol{x}_{1:2}}^{(N_{\mathrm{TS}},K)} = 10\log_{10}\left(\frac{\mathrm{MSE}_{\boldsymbol{x}_{1:2},\mathrm{C}}^{(N_{\mathrm{TS}},K)}}{\mathrm{MSE}_{\boldsymbol{x}_{1:2},\mathrm{NC}}^{(N_{\mathrm{TS}},K)}}\right), \tag{5.152}$$

$$\mathrm{CG}_{\boldsymbol{x}_{3:4}}^{(N_{\mathrm{TS}},K)} = 10\log_{10}\left(\frac{\mathrm{MSE}_{\boldsymbol{x}_{3:4},\mathrm{C}}^{(N_{\mathrm{TS}},K)}}{\mathrm{MSE}_{\boldsymbol{x}_{3:4},\mathrm{NC}}^{(N_{\mathrm{TS}},K)}}\right), \tag{5.153}$$

and

$$\mathrm{CG}_{\boldsymbol{D}}^{(N_{\mathrm{TS}},K)} = 10\log_{10}\left(\frac{\mathrm{MSE}_{\boldsymbol{D},\mathrm{C}}^{(N_{\mathrm{TS}},K)}}{\mathrm{MSE}_{\boldsymbol{D},\mathrm{NC}}^{(N_{\mathrm{TS}},K)}}\right), \tag{5.154}$$

$$\mathrm{CG}_{\boldsymbol{Q}}^{(N_{\mathrm{TS}},K)} = 10\log_{10}\left(\frac{\mathrm{MSE}_{\boldsymbol{Q},\mathrm{C}}^{(N_{\mathrm{TS}},K)}}{\mathrm{MSE}_{\boldsymbol{Q},\mathrm{NC}}^{(N_{\mathrm{TS}},K)}}\right), \tag{5.155}$$

$$\mathrm{CG}_{\bar{M}}^{(N_{\mathrm{TS}},K)} = 10\log_{10}\left(\frac{\mathrm{MSE}_{\bar{M},\mathrm{C}}^{(N_{\mathrm{TS}},K)}}{\mathrm{MSE}_{\bar{M},\mathrm{NC}}^{(N_{\mathrm{TS}},K)}}\right). \tag{5.156}$$

5.3.2.3 Time Series of Equal Length

Data Sets As Section 5.3.1.1 already used data sets with time series of equal length, we will use these data sets again in this section. That is, we use $T = 10$ synthetic data sets generated as discussed in Section 5.3.1.1. Each data set includes $I = 10$ targets that are observed over 5 time steps with starting time $N_{\mathrm{S},i} = 1$ and ending time $N_{\mathrm{E},i} = 5$ for all targets $i \in \{1, \ldots, I\}$.

Results in Table 5.1, we summarize the inference results in terms of the ground truth MSEs, defined in (5.137)–(5.141), for the samples generated with clustering and without clustering, as well as in terms of the CGs, defined in (5.142)–(5.146). We can observe that clustering improves parameter estimation noticebly, whereas there is effectively no improvement of state estimation. The improved parameter estimation can be explained by the fact that the algorithm with clustering can use the measurements of all the targets in each class to estimate the class parameters instead of having to estimate the parameters for each target separately.

	C	NC
$\mathrm{MSE}^{(K)}_{\boldsymbol{x}_{1:2}}$	13.5	13.5
$\mathrm{MSE}^{(K)}_{\boldsymbol{x}_{3:4}}$	37.3	37.8
$\mathrm{MSE}^{(K)}_{\boldsymbol{D}}$	146	157
$\mathrm{MSE}^{(K)}_{\boldsymbol{Q}}$	2049	4809
$\mathrm{MSE}^{(K)}_{\bar{M}}$	0.6	0.7

$\mathrm{CG}^{(K)}_{\boldsymbol{x}_{1:2}}$	−0.02 dB
$\mathrm{CG}^{(K)}_{\boldsymbol{x}_{3:4}}$	0.06 dB
$\mathrm{CG}^{(K)}_{\boldsymbol{D}}$	0.3 dB
$\mathrm{CG}^{(K)}_{\boldsymbol{Q}}$	3.7 dB
$\mathrm{CG}^{(K)}_{\bar{M}}$	1 dB

Tab. 5.1: *Ground truth MSEs with and without clustering, denoted as C and NC, respectively, as well as the CGs.*

5.3.2.4 Time Series of Different Lengths

Data Sets Again similarly to Section 5.3.1.1, we generate $T = 10$ synthetic data sets that are sampled from the statistical model described in Section 5.1. However, these data sets now involve $I = 10$ targets that are observed over a varying number of time steps $N_{\mathrm{TS},i} = N_{\mathrm{E},i} - N_{\mathrm{S},i} + 1$. In particular, we use $N_{\mathrm{S},i} = \mathrm{ceil}(\frac{i}{2})$ and $N_{\mathrm{E},i} = 5$ for all targets $i \in \{1, \ldots, I\}$, where $\mathrm{ceil}(\cdot)$ is the ceiling function. That is, for example, targets

1 and 2 are observed over five time steps, and targets 9 and 10 are only observed over a single time step. Furthermore, we generate at least one measurement at the starting time, that is, $M_{i,N_{\mathrm{S},i}} \geq 1$ for all targets $i \in \{1,\dots,I\}$ and data sets $t \in \{1,\dots,T\}$. We use the hyperparameters defined in Section 5.3.1.1 for all data sets, and assume that these hyperparameters are known by all algorithms.

Results The inference results in terms of the ground truth MSEs, defined in (5.137)–(5.141), for the samples generated with clustering and without clustering, as well as in terms of the CGs, defined in (5.142)–(5.146), are summarized in Table 5.2. Further, in Figure 5.10 we plot the CGs separately calculated for targets with the same number of observed time steps $N_{\mathrm{TS},i}^{(t)} = N_{\mathrm{E},i}^{(t)} - N_{\mathrm{S},i}^{(t)} + 1$, defined in (5.152)–(5.156). We can again observe that clustering significantly improves parameter estimation, whereas it improves state estimation only slightly. In particular, we can see in Figure 5.10 that state and parameter estimation for targets that have only been observed for a low number of time steps (small $N_{\mathrm{TS},i}$) showed a larger improvement. As for the case of time series of equal length (see Section 5.3.2.3), the improved parameter estimation is probably due to the fact that the algorithm with clustering can use all the targets in each class for parameter estimation instead of estimating the parameters for each target separately. This improvement is especially noticable for targets that are observed over small number of time steps as they provide only a few data points that can be used for parameter estimation.

	C	NC
$\mathrm{MSE}_{\boldsymbol{x}_{1:2}}^{(K)}$	16.6	16.8
$\mathrm{MSE}_{\boldsymbol{x}_{3:4}}^{(K)}$	70.9	82
$\mathrm{MSE}_{\boldsymbol{D}}^{(K)}$	195	910
$\mathrm{MSE}_{\boldsymbol{Q}}^{(K)}$	1789	3292
$\mathrm{MSE}_{\bar{M}}^{(K)}$	1.8	2.5

$\mathrm{CG}_{\boldsymbol{x}_{1:2}}^{(K)}$	0.05 dB
$\mathrm{CG}_{\boldsymbol{x}_{3:4}}^{(K)}$	0.6 dB
$\mathrm{CG}_{\boldsymbol{D}}^{(K)}$	6.7 dB
$\mathrm{CG}_{\boldsymbol{Q}}^{(K)}$	2.6 dB
$\mathrm{CG}_{\bar{M}}^{(K)}$	1.4 dB

Tab. 5.2: *Ground truth MSEs with and without clustering, denoted as C and NC, respectively, as well as the CGs.*

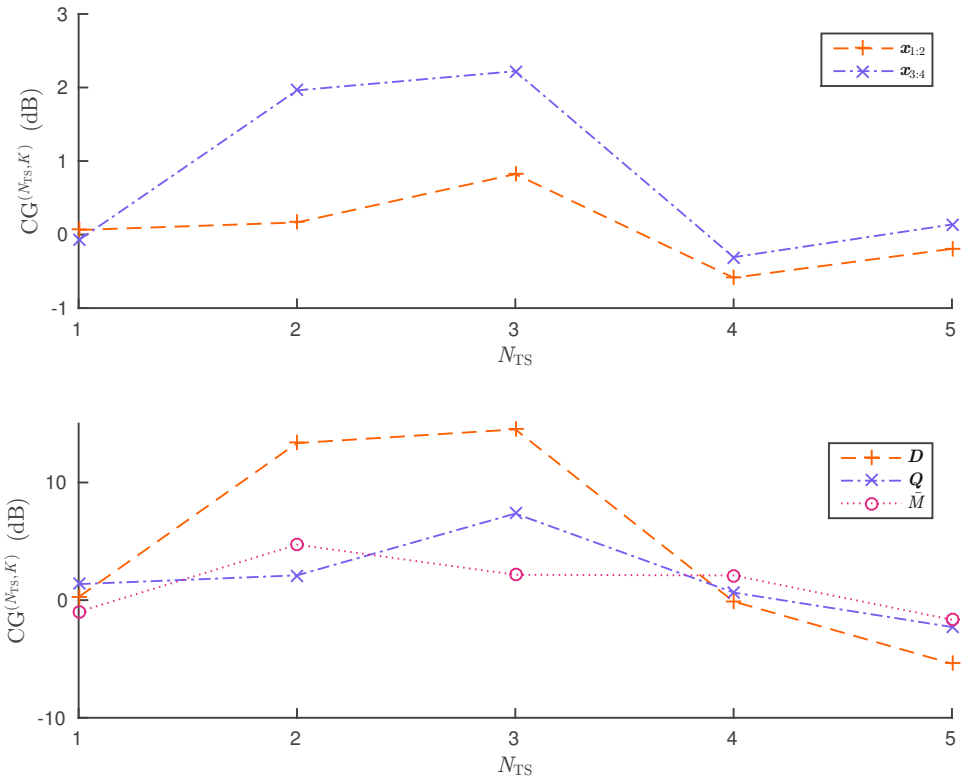

Fig. 5.10: *CGs for state and parameter estimation as a function of the number of observed time steps.*

Chapter 6

Conclusion

After giving an introduction to Monte Carlo methods and Bayesian nonparametrics, we discussed Bayesian nonparametric state-space models, that is, parameter dependent state-space models with a Dirichlet process prior. We developed two Monte Carlo based algorithms for inference in these models. As a specific application, we concretized the Bayesian nonparametric state-space models to an extended target tracking scenario and applied these algorithms to perform joint tracking and clustering. Finally, we evaluated the convergence of these algorithms and investigated the performance gain due to joint tracking and clustering.

Since the Dirichlet process prior of the parameters in our Bayesian nonparametric state-space models introduces a random class structure, we can use our algorithms not only to perform inference for the states and parameters, but also to cluster the targets. This is possible without prior knowledge of number of classes and the parameter values of each class. The first algorithm we proposed is based on particle Markov chain Monte Carlo and is suited to batch processing. The second algorithm is based on the resample-move particle filter and is suited to sequential processing.

Our performance evaluation showed that parameter estimation improved significantly due to the inherent clustering, especially for short time series. However, state estimation improved only slightly. This can be explained by the fact that our motion and measurement models allow precise state estimation even with imprecise parameter estimation.

Our statistical model for extended target tracking assumes that the number of targets is fixed and known, the measurements are already assigned to the individual targets, and there is no clutter. Thus, further research is necessary to extend our model and algorithms to the case where the number of targets is time-varying and unknown, some of the measurements may be clutter, and the assignment of the measurements to a target or to clutter is unknown. In addition, our model can be adapted to more sophisticated

motion and measurement models as well as to additional statistical dependencies between the targets. Finally, a performance evaluation using real world data sets besides synthetic data would be desirable.

Bibliography

[1] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking. Part I. Dynamic models," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1333–1364, 2003.

[2] S. S. Blackman, *Multiple-Target Tracking with Radar Applications*. Norwood, MA: Artech House, 1986.

[3] K. Granström, M. Baum, and S. Reuter, "Extended object tracking: Introduction, overview and applications," *J. Adv. Inf. Fusion*, vol. 12, no. 2, pp. 139–174, 2017.

[4] K. Granström, A. Natale, P. Braca, G. Ludeno, and F. Serafino, "Gamma Gaussian inverse Wishart probability hypothesis density for extended target tracking using X-band marine radar data," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 12, pp. 6617–6631, 2015.

[5] A. Scheel, K. Granström, D. Meissner, S. Reuter, and K. Dietmayer, "Tracking and data segmentation using a GGIW filter with mixture clustering," in *Proc. 17th Int. Conf. Inf. Fusion (FUSION)*, (Salamanca, Spain), pp. 1–8, 7 2014.

[6] J. S. Fowdur, M. Baum, and F. Heymann, "Tracking targets with known spatial extent using experimental marine radar data," in *Proc. 22th Int. Conf. Inf. Fusion (FUSION)*, (Ottawa, Canada), pp. 1–8, 7 2019.

[7] S. Challa and G. W. Pulford, "Joint target tracking and classification using radar and ESM sensors," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 37, no. 3, pp. 1039–1055, 2001.

[8] Y. Gao, Y. Liu, and X. R. Li, "Tracking-aided classification of targets using multi-hypothesis sequential probability ratio test," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 54, no. 1, pp. 233–245, 2017.

[9] X. He, R. Tharmarasa, T. Kirubarajan, A.-L. Jousselme, and P. Valin, "Joint class identification and target classification using multiple HMMs," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 50, no. 2, pp. 1269–1282, 2014.

[10] S. Ghosal and A. van der Vaart, *Fundamentals of Nonparametric Bayesian Inference*. Cambridge, United Kingdom: Cambridge University Press, 2017.

[11] T. S. Ferguson, "A Bayesian analysis of some nonparametric problems," *Ann. Stat.*, vol. 1, no. 2, pp. 209–230, 1973.

[12] T. S. Ferguson, "Bayesian density estimation by mixtures of normal distributions," in *Recent Advances in Statistics*, pp. 287–302, New York, NY: Academic Press, 1983.

[13] A. Lin, Y. Zhang, J. Heng, S. A. Allsop, K. M. Tye, P. E. Jacob, and D. Ba, "Clustering time series with nonlinear dynamics: A Bayesian non-parametric and particle-based approach," in *Proc. 22st Int. Conf. Artif. Intell. Stat. (AISTATS)*, (Naha, Japan), pp. 2476–2484, 4 2019.

[14] P. Smyth, "Clustering sequences with hidden Markov models," in *Proc. 9th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, (Denver, CO), pp. 648–654, 12 1996.

[15] K. P. Lennox, D. B. Dahl, M. Vannucci, R. Day, and J. W. Tsai, "A Dirichlet process mixture of hidden Markov models for protein structure prediction," *Ann. Appl. Stat.*, vol. 4, no. 2, pp. 916–942, 2010.

[16] V. Bastani, L. Marcenaro, and C. Regazzoni, "Unsupervised trajectory pattern classification using hierarchical Dirichlet process mixture hidden Markov model," in *Proc. IEEE Int. Workshop Mach. Learn. Signal Process. (MLSP)*, (Reims, France), pp. 1–6, 9 2014.

[17] N. Bathaee and H. Sheikhzadeh, "Non-parametric Bayesian inference for continuous density hidden Markov mixture model," *Stat. Methodol.*, vol. 33, pp. 256–275, 2016.

[18] S. Chiappa and D. Barber, "Dirichlet mixtures of Bayesian linear Gaussian state-space models: A variational approach," Tech. Rep. 161, Tübingen, Germany: Max Planck Institute for Biological Cybernetics, 2007.

[19] L. E. Nieto-Barajas, A. Contreras-Cristán, *et al.*, "A Bayesian nonparametric approach for time series clustering," *Bayesian Anal.*, vol. 9, no. 1, pp. 147–170, 2014.

[20] C. Andrieu, A. Doucet, and R. Holenstein, "Particle Markov chain Monte Carlo methods," *J. R. Stat. Soc. Series B*, vol. 72, no. 3, pp. 269–342, 2010.

[21] F. Lindsten, M. I. Jordan, and T. B. Schön, "Particle Gibbs with ancestor sampling," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 2145–2184, 2014.

[22] M. D. Escobar, "Estimating normal means with a Dirichlet process prior," *J. Am. Stat. Assoc.*, vol. 89, no. 425, pp. 268–277, 1994.

[23] S. N. MacEachern, "Estimating normal means with a conjugate style Dirichlet process prior," *Comm. Stat. Simul. Comput.*, vol. 23, no. 3, pp. 727–741, 1994.

[24] W. R. Gilks and C. Berzuini, "Following a moving target – Monte Carlo inference for dynamic Bayesian models," *J. R. Stat. Soc. Series B*, vol. 63, no. 1, pp. 127–146, 2001.

[25] G. Storvik, "Particle filters for state-space models with the presence of unknown static parameters," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 281–289, 2002.

[26] Y. B. Erol, Y. Wu, L. Li, and S. Russell, "Towards practical Bayesian parameter and state estimation," *arXiv preprint arXiv:1603.08988*, 2016.

[27] C. Berzuini, N. G. Best, W. R. Gilks, and C. Larizza, "Dynamic conditional independence models and Markov chain Monte Carlo methods," *J. Am. Stat. Assoc.*, vol. 92, no. 440, pp. 1403–1412, 1997.

[28] F. Septier, S. K. Pang, A. Carmi, and S. Godsill, "On MCMC-based particle methods for Bayesian filtering: Application to multitarget tracking," in *Proc. 3rd IEEE Int. Workshop Comput. Adv. Mult. Sensor Adapt. Process. (CAMSAP)*, (Aruba, Dutch Caribbean), pp. 360–363, 12 2009.

[29] A. Brockwell, P. Del Moral, A. Doucet, *et al.*, "Sequentially interacting Markov chain Monte Carlo methods," *Ann. Stat.*, vol. 38, no. 6, pp. 3387–3411, 2010.

[30] C. Robert and G. Casella, *Monte Carlo Statistical Methods*. New York, NY: Springer-Verlag, 2 ed., 2004.

[31] C. Andrieu, N. De Freitas, A. Doucet, and M. I. Jordan, "An introduction to MCMC for machine learning," *Mach. Learn.*, vol. 50, pp. 5–43, 2003.

[32] H. R. Künsch, "Recursive Monte Carlo filters: Algorithms and theoretical analysis," *Ann. Stat.*, vol. 33, no. 5, pp. 1983–2021, 2005.

[33] J. Sethuraman, "A constructive definition of Dirichlet priors," *Stat. Sin.*, vol. 4, no. 2, pp. 639–650, 1994.

[34] H. Ishwaran and L. F. James, "Gibbs sampling methods for stick-breaking priors," *J. Am. Stat. Assoc.*, vol. 96, no. 453, pp. 161–173, 2001.

[35] J. W. Miller and M. T. Harrison, "Inconsistency of Pitman-Yor process mixtures for the number of components," *J. Mach. Learn. Res.*, vol. 15, no. 96, pp. 3333–3370, 2014.

[36] J. W. Miller and M. T. Harrison, "Mixture models with a prior on the number of components," *J. Am. Stat. Assoc.*, vol. 113, no. 521, pp. 340–356, 2018.

[37] S. Aghabozorgi, A. S. Shirkhorshidi, and T. Y. Wah, "Time-series clustering – A decade review," *Inf. Syst.*, vol. 53, pp. 16–38, 2015.

[38] L. Mihaylova, A. Y. Carmi, F. Septier, A. Gning, S. K. Pang, and S. Godsill, "Overview of Bayesian sequential Monte Carlo methods for group and extended object tracking," *Digit. Signal Process.*, vol. 25, pp. 1–16, 2014.

[39] K. Gilholm, S. Godsill, S. Maskell, and D. Salmond, "Poisson models for extended target and group tracking," in *Proc. Signal Data Process. Small Targets*, vol. 5913, (San Diego, CA), pp. 230–241, 8 2005.

[40] K. Gilholm and D. Salmond, "Spatial distribution model for tracking extended objects," *IEE Radar, Sonar Navig.*, vol. 152, no. 5, pp. 364–371, 2005.

[41] S. Yang and M. Baum, "Extended Kalman filter for extended object tracking," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, (New Orleans, LA), pp. 4386–4390, 3 2017.

[42] R. P. Mahler, *Advances in Statistical Multisource-Multitarget Information Fusion*. Norwood, MA: Artech House, 2014.

[43] N. J. Gordon, D. J. Salmond, and A. F. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proc. F*, vol. 140, no. 2, pp. 107–113, 1993.

[44] M. Feldmann, D. Franken, and W. Koch, "Tracking of extended objects and group targets using random matrices," *IEEE Trans. Signal Process.*, vol. 59, no. 4, pp. 1409–1420, 2010.

[45] M. K. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," *J. Am. Stat. Assoc.*, vol. 94, no. 446, pp. 590–599, 1999.

Hiermit erkläre ich, dass die vorliegende Arbeit gemäß dem Code of Conduct, insbesondere ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel, angefertigt wurde. Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet.
Die Arbeit wurde bisher weder im In– noch im Ausland in gleicher oder in ähnlicher Form in anderen Prüfungsverfahren vorgelegt.

Wien, 13.01.2021 *Bernd Kreidl*