

**Characterization of uncertainty
in Bayesian estimation using
Sequential Monte Carlo methods**

Composition of the Graduation Committee:

Prof. Dr.	A. Bagchi	University of Twente
Dr.	P. K. Mandal	University of Twente
Dr. Ir.	Y. Boers	Thales Nederland B. V.
Prof. Dr.	A. A. Stoorvogel	University of Twente
Prof. Dr. Ir.	R. N. J. Veldhuis	University of Twente
Prof. Dr.	M. G. S. Bruno	Instituto Tecnológico de Aeronáutica
Prof. Dr.-Ing.	U. D. Hanebeck	Karlsruhe Institute of Technology



CTIT

The research leading to this thesis has received funding from the EU's Seventh Framework Programme under grant agreement n° 238710. The research has been carried out in the MC IMPULSE project: <https://mcimpulse.isy.liu.se>.

CTIT Ph.D. Thesis Series No. 13-260
Centre for Telematics and Information Technology
P.O. Box 217, 7500 AE
Enschede, The Netherlands.

ISSN: 1381-3617 (CTIT Ph.D. thesis Series No. 13-260)

ISBN: 978-90-365-0526-0

DOI: 10.3990./1.9789036505260

<http://dx.doi.org/10.3990/1.9789036505260>

Typeset with \LaTeX . Printed by Wöhrmann Print Service.

All rights reserved. No part of this book may be reproduced or transmitted, in any form or by any means, electronic or mechanical, including photocopying, microfilming, and recording, or by any information storage or retrieval system, without the prior written permission of the author.

Characterization of uncertainty in Bayesian estimation using Sequential Monte Carlo methods

DISSERTATION

to obtain the degree of doctor
at the University of Twente, under the authority of
the rector magnificus, prof.dr. H. Brinksma,
on account of the decision of the graduation committee,
to be publicly defended
on Thursday 24 october 2013 at 14.45

by
Edson Hiroshi Aoki
born on 10 november 1981
in São Paulo, Brazil

This thesis has been approved by promotor prof. dr. A. Bagchi and assistant promotor dr. P. K. Mandal.

Copyright ©2013 Edson Hiroshi Aoki, Enschede, The Netherlands.

Acknowledgements

I would like to thank:

- My former teacher Marcelo Gomes da Silva Bruno, from Instituto Tecnológico de Aeronáutica, who was an enthusiastic and formidable teacher of stochastic processes and statistical signal processing;
- My co-supervisor Yvo Boers, from Thales Nederland B.V., who was my main supporter and motivator, as well as one of my best friends, during my three years in Enschede;
- My promotor Arun Bagchi, who accepted me as his PhD student and had all the patience that I did not;
- My co-supervisor Pranab Mandal, who at times gave all his time and energy to help me;
- My collaborator Norikazu Ikoma, from Kyushu Institute of Technology, who was the most warm and helpful host that one can have during my stay in Tokyo and Kitakyushu, and who taught me how to bake an okonomiyaki;
- My collaborators Lennart Svensson, from Chalmers University of Technology, Hans Driessen, from Thales Nederland B.V., and Fredrik Gustafsson, from Linköping University, who believed in my work and supported me as much as they could;
- My friend Fotios Katsilieris, with whom I had many fun moments and deep discussions in 8 countries and 4 continents;
- My “coffee time” mates Ove (most boring guy in the World), Svetlana (evil Russia/Apple agent), Elena (my antagonist), David

(Caribbean mercenary), Lulu (oooooohhh I am so tired), Gijs (child trapped in an 1.90 m body), Felix (the guy who constantly changes his looks), Shavarsh (most fashionable man), Lilya (most fashionable woman) and Michael (least fashionable man), with whom I had many interesting conversations and cups of terrible coffee;

- My family and my girlfriend Luciana, who gave me all the love and emotional support that are crucial when you move to a different place, with big challenges to face;
- All my friends from from MC IMPULSE, from the Netherlands, as well as my friends from Brazil who supported me during these three years. I am not going to name them all as I will unavoidably forget someone.

Contents

Contents	iii
List of Figures	ix
Nomenclature	xv
1 Introduction	1
1.1 Motivation	1
1.2 Overview of existing research for the problems considered in this thesis	2
1.2.1 Online joint state and parameter estimation	2
1.2.2 Multi-target tracking and labelling	3
1.2.3 Information-driven sensor management	4
1.3 Outline and contributions of this thesis	4
2 Mathematical background	6
2.1 Sequential Monte Carlo methods	6
2.1.1 The SIS particle filter	6
2.1.1.1 Derivation	6
2.1.1.2 SIS PF algorithm	9
2.1.1.3 SIS PF degeneracy	10
2.1.2 The SIR particle filter	11
2.1.2.1 Derivation	11
2.1.2.2 SIR PF algorithm	13
2.1.2.3 Systematic resampling algorithm for a time step k	13

2.1.2.4	SIR PF degeneracy	14
2.1.3	Particle filters applied to Partially Observed First Order Markov (POM1) processes	16
2.1.4	The Rao-Blackwellized Particle Filter (RBPF)	19
2.1.4.1	Mechanism	19
2.1.4.2	RBPF algorithm	20
2.1.4.3	Benefits and drawbacks	22
2.1.5	The Marginal Particle Filter (MPF)	22
2.1.5.1	Mechanism	22
2.1.5.2	MPF algorithm	25
2.1.5.3	Benefits and drawbacks	25
2.2	Joint state and parameter estimation	26
2.2.1	Mathematical formulation	26
2.2.2	Parameter estimation strategies	28
2.2.2.1	Maximum Likelihood approach	28
2.2.2.2	Decoupled Bayesian approach	29
2.2.2.3	Coupled Bayesian approach	29
2.3	Joint multi-target tracking	31
2.3.1	Finite Set Statistics	32
2.3.1.1	Multi-object calculus	32
2.3.1.2	Random Finite Sets	33
2.3.1.3	Estimation of Random Finite Sets	34
2.3.2	Poisson Point Process theory	36
2.3.3	FISST, PPP and multi-object statistical representations	37
2.3.4	The Multi-target Sequential Monte Carlo filter	39
2.3.4.1	System model	39
2.3.4.2	SIR M-SMC filter algorithm	40
2.4	Sensor management	41
2.4.1	Mathematical formulation	42
2.4.2	Sensor management criteria	44
2.4.2.1	Task-driven sensor management	44
2.4.2.2	Information-driven sensor management	45
2.4.3	Sensor management applied to POM1 processes	46

2.4.4	Sensor management using particle filters	47
3	Online Bayesian parameter estimation using the Rao-Blackwellized marginal particle filter	50
3.1	Introduction	51
3.2	State-of-the-art SMC methods applied to joint state and parameter estimation	52
3.2.1	The SIR PF	52
3.2.2	The Liu and West PF	55
3.2.3	The RBPF and the RBMPF	57
3.3	The Discrete RBMPF (D-RBMPF)	59
3.3.1	Obtaining the particle states/weights (MPF step)	60
3.3.2	Obtaining the parameter conditional probabilities (discrete step)	63
3.3.3	D-RBMPF algorithm	64
3.3.4	A special case	65
3.3.5	Computational complexity	65
3.4	The Monte Carlo RBMPF (MC-RBMPF)	66
3.4.1	Obtaining the particle states/weights (MPF step)	67
3.4.2	Obtaining the sub-particle states/weights (MC step)	68
3.4.3	MC-RBMPF algorithm	70
3.4.4	Computational complexity	72
3.5	Simulation - Turn rate estimation	74
3.5.1	Problem description	74
3.5.2	Simulation description	74
3.5.3	Results	77
3.5.4	Trade-off between computational cost and performance	79
3.6	Simulation - Stochastic volatility estimation	80
3.6.1	Problem description	80
3.6.2	Simulation description	82
3.6.3	Results	83
3.6.4	Trade-off between computational cost and performance	86
3.7	Conclusions and recommendations	86

4	The problem of optimal Bayesian track labelling in multi-target tracking	90
4.1	Introduction	91
4.2	Bayes formulation of the multi-target tracking and labelling (MTTL) problem	95
4.2.1	Physical interpretation of the Bayesian labelling problem	96
4.2.2	The prior pdf	99
4.2.3	The likelihood function	100
4.2.4	The state transition density	100
4.2.4.1	No target births or deaths	100
4.2.4.2	With target deaths, no target births	101
4.2.4.3	With target births and deaths	102
4.3	Statistics for Bayesian MTTL	104
4.3.1	The labelling probability	105
4.3.2	Track extraction methods for Bayesian MTTL	108
4.3.2.1	The MMOSPA-MLP estimate	108
4.3.2.2	The JoM-MLP estimate	109
4.3.2.3	Pure JoM estimate	110
4.3.2.4	Pure MMOSPA estimate	110
4.3.2.5	Which track extraction method to use?	111
4.3.3	Performance metrics for MTTL	111
4.4	Calculating labelling probabilities for existing MTT algorithms	113
4.4.1	Multi-target Sequential Monte Carlo (M-SMC) filter	113
4.4.2	Multiple Hypothesis Tracking (MHT)	114
4.4.3	M-SMC filter, MHT and degeneracy	115
4.4.4	A numerical example	116
4.5	A novel solution to the MTTL problem: The Labelling Uncertainty-Aware Particle Filter	118
4.5.1	Theoretical basis	120
4.5.2	Derivation of the LUA-PF	121
4.5.2.1	Calculation of particle states and weights (tracking step)	122

4.5.2.2	Calculation of particle labelling probabilities (labelling step)	122
4.5.3	LUA-PF algorithm	124
4.5.4	Computational aspects	126
4.6	Simulations	127
4.6.1	Which metric to use?	127
4.6.2	Scenarios	129
4.6.3	Results for Monte Carlo runs with varying sequence of measurements	131
4.6.4	Results for Monte Carlo runs with fixed sequence of measurements	134
4.7	Conclusions and recommendations	135
5	An analysis of information-driven sensor management criteria	137
5.1	Introduction	138
5.2	Information-theoretic sensor management	140
5.2.1	Shannon entropy	140
5.2.2	Kullback-Leibler (KL) divergence	141
5.2.3	Rényi entropy	141
5.2.4	Rényi divergence	142
5.3	A look at the Shannon entropy and KL divergence criteria	143
5.3.1	Entropy as an uncertainty measure	143
5.3.2	Relationship between the KL divergence and Shannon entropy criteria	146
5.4	On the near-universal proxy argument for theoretical justification of information-driven sensor management	149
5.4.1	The near-universal proxy argument	150
5.4.2	Rebuttal of the “near-universal” proxy argument	151
5.5	KL and Rényi divergences and “balancing estimation errors”	153
5.5.1	Rényi divergences and trade-off between uncertainty reduction and morphological changes	155
5.5.2	A practical example: bearings-only tracking	159
5.6	Conclusions and recommendations	162

6 Conclusions and recommendations	165
A The optimal proposal density of the MPF	168
B An in-depth analysis of the mixed labelling phenomenon in two-target tracking	170
B.1 The MTTL Bayesian recursion in the two-target case	170
B.2 Origin of mixed labelling	173
B.3 Persistence of mixed labelling	174
B.4 Mixed labelling and non-kinematic states	177
C The “one-sided” decoupling between tracking and labelling in Bayesian multi-target tracking	178
C.1 The tracking sub-problem	178
C.2 The labelling sub-problem	185
References	186

List of Figures

2.1	Illustration of a running SIR PF, showing the particles produced at each time step, after importance sampling and before resampling	15
2.2	Sensor management as a stochastic control problem	42
3.1	Target trajectory for the turn rate estimation scenario	75
3.2	Turn rate estimation – results	78
3.3	Turn rate estimation – ratio of outlier estimates	79
3.4	Turn rate estimation – results for the LWPF with 10 times more particles	81
3.5	Stochastic volatility estimation - RMSE results	84
3.6	Stochastic volatility estimation - NEES results	85
3.7	Stochastic volatility estimation - RMSE results for the LWPF with 40 times more particles	87
3.8	Stochastic volatility estimation - NEES results for the LWPF with 40 times more particles	88
4.1	Situation where assignment of labels to tracks is ambiguous	92
4.2	Well-posed (above) and ill-posed (below) Bayesian labelling problems	98
4.3	Particle representation of the multi-target distribution in a situation where mixed labelling occurs (source: Crouse et al. [2011a]). The squares and circles mark the possible locations of each target in terms of particles	105
4.4	Two-target scenario with mixed labelling	117
4.5	M-SMC filter labelling results	119
4.6	Multi-target simulation scenarios	130

LIST OF FIGURES

4.7	Simulation (varying Z^k) results for scenarios 1 and 2	132
4.8	Simulation (varying Z^k) results for scenarios 3 and 4	133
4.9	Simulation (fixed Z^k) results	135
5.1	Two Gaussian mixture pdfs, the first with variance 16.25 and entropy within the interval [1.07,1.23], and the second with variance 7.84 and entropy within [2.32,2.79]	145
5.2	Some bivariate Gaussian densities	154
5.3	Variance ratio $\frac{\sigma_y^2}{\sigma_x^2}$ according to the rotation angle θ	157
5.4	α -divergence trade-off test results	160
5.5	Results for bearings-only tracking and sensor management experiment	163

Nomenclature

Greek Symbols

β Belief mass function

Θ random parameter vector

$\gamma(\cdot)$ sensor management reward or risk function

$\delta_{\mathbf{x}'}(\mathbf{x})$ multi-object Dirac delta density evaluated at \mathbf{x} with argument \mathbf{x}'

σ_x standard deviation of a random variable X

σ_x^2 variance of a random variable X

θ parameter vector value

Superscripts

$\hat{}$ estimate (of some value)

T transpose

Other Symbols

\approx approximately equal to

$:=$ attribution operator

\mathbf{X}_k random finite set state at time k

\mathbf{Z}_k random finite set observation at time k

LIST OF FIGURES

$J_t(\cdot)$	Janossy measure
$P(\cdot)$	probability of an event
S_k	(part of) random vector state at time k
X_k	random vector state at time k
Z_k	random vector observation at time k
\mathbf{x}_k	finite set state value at time k
\mathbf{z}_k	finite set observation value at time k
$ \cdot $	cardinality operator
$D_\alpha(\cdot)$	Rényi divergence (α -divergence)
$D(\cdot)$	Kullback-Leibler divergence
$E[\cdot]$	expectation
$f(\cdot)$	Random Finite Set density
$\mathcal{N}(\hat{x}, P)$	multivariate Gaussian probability density mean \hat{x} and covariance P
$\mathcal{N}(x; \hat{x}, P)$	multivariate Gaussian probability density evaluated at x with mean \hat{x} and covariance P
$H(\cdot)$	Shannon entropy
$H_\alpha(\cdot)$	Rényi entropy
$I(X; Z)$	mutual information between state X and observation Z
$j_t(\cdot)$	Janossy density
N_P	number of particles
$p(\cdot)$	probability density function
P_k	covariance of a random variable X_k at time k

LIST OF FIGURES

$q(\cdot)$	proposal density
\mathbb{R}^n	n -Dimensional Euclidean space
s_k	(part of) vector value at time k
\sim	sampling (according to a probability distribution)
U_k	random control action at time k
u_k	control action value at time k
$\mathcal{U}(x; x_l, x_u)$	uniform probability density evaluated at x with lower bound x_l and upper bound x_u
$\mathcal{U}(x_l, x_u)$	uniform probability density lower bound x_l and upper bound x_u
w_k	particle weight at time k
x_k	vector state value at time k
z_k	vector observation value at time k
Z^k	all available information (observations and control actions) until and including time k

Acronyms

cdf	cumulative distribution function
CL	Closed-Loop
D-RBMPF	Discrete Rao-Blackwellized Marginal Particle Filter
EM	Expected Maximization
FISST	Finite Set Statistics
i.i.d.	independent and identically distributed
JoM	Joint Multi-target

LIST OF FIGURES

KL	Kullback-Leibler
LUA-PF	Labelling Uncertainty-Aware Particle filter
LWPF	Liu and West Particle Filter
MAP	Maximum a Posteriori
MCMC	Markov Chain Monte Carlo
MC-RBMPF	Monte Carlo Rao-Blackwellized Marginal Particle Filter
MHT	Multiple Hypothesis Tracking
ML	Maximum Likelihood
MLP	Maximum Labelling Probability
MMOSPA	Minimum Mean Optimal Subpattern Assignment
MMSE	Minimum Mean Square Error
MPF	Marginal Particle Filter
M-SMC	Multi-target Sequential Monte Carlo
MTTL	Multi-Target Tracking and Labelling
MTT	Multi-Target Tracking
OLF	Open-Loop Feedback
OL	Open-Loop
OSPA	Optimal Subpattern Assignment
pdf	probability density function
PF	Particle Filter
PHD	Probability Hypothesis Density
pmf	probability mass function

LIST OF FIGURES

POM1DF Partially Observed Markov-1 with Direct Feedthrough

POM1 Partially Observed Markov-1

PPP Poisson Point Processes

RBMPF Rao-Blackwellized Marginal Particle Filter

RBPF Rao-Blackwellized Particle Filter

RFS Random Finite Set

SIR Systematic Importance Resampling

SIS Sequential Importance Sampling

SMC Sequential Monte Carlo

UAV Unmanned Aerial Vehicle

w.r.t. with respect to

Chapter 1

Introduction

1.1 Motivation

The discrete-time Bayesian estimation problem may be generally described as the problem of estimating a random signal of interest (also referred to as the “state”) X_j , where j denotes the time index, from a sequence of random observations (Z_1, \dots, Z_k) . The cases where $j < k$, $j = k$ and $j > k$ correspond respectively to the *smoothing*, *filtering* and *prediction* variants of the estimation problem.

The observations, also called measurements, are treated as random because they are assumed to be corrupted by random errors, the so-called *observation noise*. In this case, we generally cannot estimate the exact value of X_j and the system composed by the state and the observations is called a *partially observed* system. The estimated value \hat{x}_j will hence contain errors with respect to the true value of the state X_j , which can be statistically quantified, for instance, by the covariance matrix or the entropy.

Quantifying estimation errors, or in other words, quantifying the *uncertainty* associated with the estimates, is an important problem for real-world applications, particularly when there is the possibility that such errors are high. This includes many special cases of the target tracking problem, including estimating the altitude and slant range error of a constant-velocity target using 2D radars (see Aoki [2010]), estimating sensor biases from targets of opportunity (see Syldatk et al. [2012]), and identifying targets that separate after moving in close proximity (see

Blom et al. [2008]). The last problem, plus the econometrics problem of stochastic volatility estimation (Aihara et al. [2009]), are considered in this thesis. An estimator that allows us to compute accurate measures of uncertainty is called a *statistically consistent* estimator (see [Bar-Shalom et al., 2001, Section 5.4]).

In some practical problems, there is also the possibility of redirecting sensors, or adjusting sensor properties, in order to reduce the estimation uncertainty. This problem is called *sensor management*, a special case of the stochastic control problem, where the feedback is directed to the observations, rather than to the state. Needless to say, a good characterization of estimation uncertainty is a prerequisite for good sensor management.

In summary, our problem of interest is how to *efficiently characterize estimation uncertainty*, and how to reduce it using sensor management techniques. To achieve these goals, we focus on Sequential Monte Carlo (SMC) methods, also known as particle filters (PF). As numerical, nonlinear estimators, particle filters do not rely on the principle of orthogonality ([Bar-Shalom et al., 2001, Section 3.3]) and are hence, at least in theory, able to perform Bayesian estimation in an approximately optimal manner even for systems with nonlinear structure and non-Gaussian process and observation noises.

1.2 Overview of existing research for the problems considered in this thesis

In this section, we will present a brief summary of the problems considered in this thesis, and the description of the state-of-the-art techniques available to deal with them. A more in-depth discussion of these problems and techniques, as well as proper literary reviews, will be presented in the corresponding chapters.

1.2.1 Online joint state and parameter estimation

In joint state and parameter estimation, for a partially observed system, part of the quantities to be estimated are time-invariant (which we refer to as “parameter”) and the another part is time-variant (which we refer to as “state”). It is

well-known that particle filters have difficulty dealing with such kind of problems, due to the so-called PF *degeneracy* phenomenon.

In *online* joint state and parameter estimation, both parameter and state estimates need to be produced as soon as a new measurement becomes available. For this class of problems, it is known that the particle filter degeneracy phenomenon becomes unavoidable regardless of the number of particles used. The usual approach for this problem is to modify the basic PF algorithm by attributing artificial dynamics to the parameters, such that they are also treated as time-varying states. This approach, however, leads to biases on the system model, making statistically consistent estimation harder to achieve. One popular algorithm based on this idea is the Liu and West Particle Filter (LWPF).

1.2.2 Multi-target tracking and labelling

The general Multi-Target Tracking (MTT) problem consists of estimating the locations of multiple targets (or, stating in more general form, estimating the states of multiple objects), where the number of targets itself may need to be estimated (i.e. it may be unknown and possibly time-varying). A more complex version of this problem is the Multi-Target Tracking and Labelling (MTTL), where the targets also need to be individually identified; in other words, we must assign a “label” to each estimated location, and ideally, this label should be consistently associated (across multiple time steps) with the same real-world target.

A challenging MTTL scenario is when targets move in close proximity for a while, and afterwards separate. In this case, it is intuitively clear that there will be uncertainty on labelling the individual targets, but how to mathematically model this uncertainty is not yet satisfactorily answered in literature. Some quantities have been proposed in the literature, but their physical interpretations are unclear, and none of them apply to the general case where the number of targets is unknown and/or time-varying. Disregarding the problem of finding an appropriate uncertainty measure, the problem of labelling itself is another concern as the use of particle filters to perform Bayesian labelling is known to lead to underestimation of uncertainty, due to the particle filter degeneracy phenomenon.

1.2.3 Information-driven sensor management

Sensor management is typically performed by choosing some control action that optimizes a criterion related to estimation quality, for instance, minimizing the determinant or trace of the covariance matrix of the state to be estimated. The so-called “information-driven sensor management” consists of using measures from information theory as sensor management criteria, like the Kullback-Leibler (KL) and Rényi divergences.

Such criteria are attractive as they can be applied in a relatively straightforward manner to complex problems (like multi-target tracking and hybrid continuous/discrete estimation), and have shown promising experimental results for some practical problems. However, the available theoretical arguments to justify their use remain unclear or debatable. One such argument is the “near-universal proxy” algorithm, which suggests that Rényi divergences can be used as substitutes for arbitrary risk-based sensor management criteria.

1.3 Outline and contributions of this thesis

Chapter 2 reviews the theoretical basis of this work, including SMC methods, Finite Set Statistics (FISST), Poisson Point Process (PPP) theory, parameter estimation strategies and basic sensor management concepts;

Chapter 3 proposes two new solutions for the problem of online joint state and parameter estimation, that addresses the limitations of state-of-the-art methods based on artificial dynamics. The proposed algorithms are novel implementations of the Rao-Blackwellized Marginal Particle Filter (RBMPF), a recently proposed PF variant designed to counter degeneracy. These methods are general, in the sense that they can be applied to nonlinear, non-Gaussian systems without any particular structure;

Chapter 4 presents a mathematical formulation of the Bayesian MTTL problem, based on FISST, culminating in the definition of a number of useful statistics, including the *labelling probability*, a measure of estimation uncertainty for this problem. Additionally, we propose a new algorithm, the Labelling Uncertainty-Aware Particle Filter (LUA-PF) filter, as a solution for the MTTL problem that

avoids the particle filter degeneracy phenomenon;

Chapter 5 presents a theoretical and empirical analysis of information-driven sensor management criteria, more specifically of previous arguments for (and against) using the Kullback-Leibler (KL) and Rényi divergence as sensor management criteria. The analysis leads to the conclusion that there is little support for the existing arguments for using the Rényi divergence, whereas there is some basis for using the KL divergence due to its relationship with the Shannon entropy.

Chapter 2

Mathematical background

2.1 Sequential Monte Carlo methods

As we mentioned in Section 1.1, SMC methods are techniques heavily emphasized in this work. The most well-known SMC method is the Sequential Importance Resampling (SIR) particle filter proposed by Gordon et al. [1993] and Kitagawa [1993], which is also the basis of many other SMC methods. In this section, we will look at the aspects of this technique relevant for our work, in particular, the PF degeneracy phenomenon. However, we will first take a look at the Sequential Importance Sampling (SIS) PF, which will help us later to understand the strengths and limitations of the SIR mechanism.

In addition, we also look at two PF variants that will be relevant later in this thesis: the Rao-Blackwellized Particle Filter (RBPF) and the Marginal Particle Filter (MPF).

2.1.1 The SIS particle filter

2.1.1.1 Derivation

We will review the derivation of the SIS particle filter for fully general stochastic state-space models, i.e. without any Markov assumptions, as shown e.g. in [de Freitas, 1999, Chapter 6]. Consider a stochastic process described by the sequence (X_k, Z_k) , where X_k and Z_k are random variables denoting the state and

2. MATHEMATICAL BACKGROUND

observation at time k , with realizations respectively given by x_k and z_k , Let Z^k (with k as superscript, not subscript) denote all available observations until and including time k ($Z^k = (z_1, \dots, z_k)$). At time k , the statistical information about the trajectory (X_0, \dots, X_k) given Z^k is summarized by the posterior probability density function (pdf) $p(x_0, \dots, x_k | Z^k)$. According to the Bayes theorem, the posterior is given by

$$\begin{aligned}
 p(x_0, \dots, x_k | Z^k) &= \frac{p(z_k | x_0, \dots, x_k, Z^{k-1}) p(x_k | x_0, \dots, x_{k-1}, Z^{k-1})}{p(z_k | Z^{k-1})} \\
 &\quad \times p(x_0, \dots, x_{k-1} | Z^{k-1}) \\
 &= p(x_0) \prod_{j=1}^k \frac{p(z_j | x_0, \dots, x_j, Z^{j-1}) p(x_j | x_0, \dots, x_{j-1}, Z^{j-1})}{p(z_j | Z^{j-1})}.
 \end{aligned} \tag{2.1}$$

In the filtering problem, one is interested in estimating some function $g(X_k)$ of the ‘‘current’’ state X_k . The expectation of this quantity, given the available observations Z^k , is given by

$$\begin{aligned}
 E[g(X_k) | Z^k] &= \underbrace{\int \dots \int}_{k+1} g(x_k) p(x_0, \dots, x_k | Z^k) dx_0 \dots dx_k \\
 &= \underbrace{\int \dots \int}_{k+1} \left(\prod_{j=1}^k \frac{p(z_j | x_0, \dots, x_j, Z^{j-1}) p(x_j | x_0, \dots, x_{j-1}, Z^{j-1})}{p(z_j | Z^{j-1})} \right) \\
 &\quad \times p(x_0) g(x_k) dx_0 \dots dx_k \\
 &= \underbrace{\int \dots \int}_{k+1} \left(\prod_{j=1}^k \frac{p(z_j | x_0, \dots, x_j, Z^{j-1}) p(x_j | x_0, \dots, x_{j-1}, Z^{j-1})}{p(z_j | Z^{j-1})} \right) \\
 &\quad \times \frac{p(x_0) g(x_k)}{q(x_0, \dots, x_k | Z^k)} q(x_0, \dots, x_k | Z^k) dx_0 \dots dx_k.
 \end{aligned} \tag{2.2}$$

where $q(x_0, \dots, x_k | Z^k)$ is some pdf of X_0, \dots, X_k parametrized on Z^k that we are able to sample from. If we obtain N_P independent, identically distributed (i.i.d.) samples $(x_0(i), \dots, x_k(i))$, $i = 1, \dots, N_P$, from this density, then according to the

2. MATHEMATICAL BACKGROUND

law of large numbers, expectation (2.2) can be approximated as

$$E [g(X_k) | Z^k] \approx \sum_{i=1}^{N_P} w_k(i) g(x_k(i)) \quad (2.3)$$

where, for $i = 1, \dots, N_P$

$$w_k(i) = \frac{1}{N_P} \left(\prod_{j=1}^k \frac{p(z_j | x_0(i), \dots, x_j(i), Z^{j-1}) p(x_j(i) | x_0(i), \dots, x_{j-1}(i), Z^{j-1})}{p(z_j | Z^{j-1})} \right) \times \frac{p(x_0(i))}{q(x_0(i), \dots, x_k(i) | Z^k)}. \quad (2.4)$$

Implicitly, we make the approximation

$$p(x_0, \dots, x_k | Z^k) \approx \sum_{i=1}^{N_P} w_k(i) \delta(x_0 - x_0(i)) \dots \delta(x_k - x_k(i)). \quad (2.5)$$

where we refer to the samples $(x_0(i), \dots, x_k(i))$, $i = 1, \dots, N_P$ as the *particles* and to $w_k(i)$, $i = 1, \dots, N_P$ as the *particle weights*. One interpretation of (2.5) is that each particle represents a hypothesis on the trajectory (X_0, \dots, X_k) with an attached weight. Now, let us assume that $q(x_0, \dots, x_k | Z^k)$ is structured as follows:

$$q(x_0, \dots, x_k | Z^k) \triangleq p(x_0) \prod_{j=1}^k q(x_j | x_0, \dots, x_{j-1}, Z^j) \quad (2.6)$$

where $q(x_j | x_0, \dots, x_{j-1}, Z^j)$ is some pdf of x_j parametrized on (x_0, \dots, x_{j-1}) , Z^j that we are able to sample from, which we refer to as *proposal density* or *importance sampling function*. Observe that we can obtain each particle $i = 1, \dots, N_P$ from $q(x_0, \dots, x_k | Z^k)$ by means of *sequential sampling*, i.e. we first obtain a sample $x_0(i)$ from $p(x_0)$, then use it to obtain a sample $x_1(i)$ from $q(x_1 | x_0(i), z_1)$, and so forth, until we obtain the complete sample $(x_0(i), \dots, x_k(i))$. In the particle filter algorithm, we refer to this procedure as *sequential importance sampling*.

With $q(x_j | x_0, \dots, x_{j-1}, Z^j)$ having the form (2.6), observe now that the ex-

2. MATHEMATICAL BACKGROUND

pression for $w_k(i)$ (2.4) can be rewritten as

$$w_k(i) = \frac{1}{N_P} \left(\prod_{j=1}^k \frac{p(z_j | x_0(i), \dots, x_j(i), Z^{j-1}) p(x_j(i) | x_0(i), \dots, x_{j-1}(i), Z^{j-1})}{p(z_j | Z^{j-1}) q(x_j(i) | x_0(i), \dots, x_{j-1}(i), Z^j)} \right) \quad (2.7)$$

which can be easily written in recursive form, as

$$w_j(i) \triangleq \frac{p(z_j | x_0(i), \dots, x_j(i), Z^{j-1}) p(x_j(i) | x_0(i), \dots, x_{j-1}(i), Z^{j-1})}{p(z_j | Z^{j-1}) q(x_j(i) | x_0(i), \dots, x_{j-1}(i), Z^j)} w_{j-1}(i) \quad (2.8)$$

for $j = 1, \dots, k$, and $w_0(i) \triangleq 1/N_P$.

Therefore, approximation (2.5) can be computed iteratively, at each time step j , on the basis of each new observation z_j . At each time step j , we sample $x_j(i)$, $i = 1, \dots, N_P$ from the proposal density $q(x_j(i) | x_0(i), \dots, x_{j-1}(i), Z^j)$, and we calculate the weights $w_j(i)$, $i = 1, \dots, N_P$ according to (2.8). The resulting iterative algorithm is called Sequential Importance Sampling Particle Filter (SIS PF).

In order for (2.3) to be a valid approximation of the true expectation, we must have $\sum_{i=1}^{N_P} w_k(i) = 1$. Hence, we do not need to compute the term $p(z_j | Z^{j-1})$ (that does not depend on i) explicitly. Instead, we can first compute the *unnormalized particle weights* by ignoring the term:

$$\bar{w}_j(i) = \frac{p(z_j | x_0(i), \dots, x_j(i), Z^{j-1}) p(x_j(i) | x_0(i), \dots, x_{j-1}(i), Z^{j-1})}{q(x_j(i) | x_0(i), \dots, x_{j-1}(i), Z^k)} w_{j-1}(i) \quad (2.9)$$

and the particle weights themselves can be computed using the normalization step $w_j(i) = \frac{\bar{w}_j(i)}{\sum_{j=1}^{N_P} \bar{w}_j(j)}$.

2.1.1.2 SIS PF algorithm

Initialization:

1. For each particle $i = 1, \dots, N_P$

2. MATHEMATICAL BACKGROUND

(a) Sample $x_0(i) \sim p(x_0)$

(b) Make $w_0(i) = \frac{1}{N_P}$

At every time step $k = 1, 2, \dots$:

1. For each particle $i = 1, \dots, N_P$

(a) Perform importance sampling by making

$$x_k(i) \sim q\left(x_k \mid x_0(i), \dots, x_{k-1}(i), Z^k\right)$$

where $q(x_k \mid x_0, \dots, x_{k-1}, Z^k)$ is a proposal density

(b) Calculate the unnormalized weight according to

$$\begin{aligned} \bar{w}_k(i) &= \frac{p(z_k \mid x_0(i), \dots, x_{k-1}(i), Z^{k-1}) p(x_k(i) \mid x_0(i), \dots, x_{k-1}(i), Z^{k-1})}{q(x_k(i) \mid x_0(i), \dots, x_{k-1}(i), Z^k)} \\ &\quad \times w_{k-1}(i) \end{aligned}$$

2. Normalize the particle weights according to

$$w_k(i) = \frac{\bar{w}_k(i)}{\sum_{j=1}^{N_P} \bar{w}_k(j)}, \quad i = 1, \dots, N_P$$

2.1.1.3 SIS PF degeneracy

The SIS PF mechanism relies on the law of large numbers, and therefore, approximation (2.3) is guaranteed to asymptotically converge to the true expectation. However, this convergence property does not guarantee that the necessary number of particles to obtain accurate estimates remains constant with time.

In practice, the algorithm is known to be ineffective except when the maximum considered time step k , if any, is very small. The reason is that as shown in [Doucet et al., 2001, Proposition 4], the variance of the weights (or more precisely, its expectation taken over all observations (Z_1, \dots, Z_k)) increases with time. Eventually, after a number of iterations, one of the weights will become 1, while all the remaining weights will become zero. This situation, where $p(x_0, \dots, x_k \mid Z^k)$ ends up being effectively represented by a single sample, is referred to as *degeneracy*.

2. MATHEMATICAL BACKGROUND

We might be able to postpone the time where degeneracy occurs by reducing the variance of the weights. The proposal density that minimizes the expectation of this variance is called the *optimal* proposal density, which is, as shown in [Doucet et al., 2001, Proposition 3], given by

$$q(x_k | x_0, \dots, x_{k-1}, Z^k) = p(x_k | x_0, \dots, x_{k-1}, Z^k) \quad (2.10)$$

Unfortunately, sampling directly from $p(x_k | x_0, \dots, x_{k-1}, Z^k)$ is often difficult, and even if possible, it would at most postpone degeneracy, meaning that the SIS PF would still become ineffective at some point (unless, naturally, if the maximum k is sufficiently small). This behavior has led to the development of the SIR PF.

2.1.2 The SIR particle filter

2.1.2.1 Derivation

Consider the SIS PF algorithm described in Section 2.1.1.2. We would like to prevent the variance of the particle weights from increasing at each time step. This can be accomplished by means of a *resampling* mechanism.

Let us take a look at the approximation (2.3) of $E[g(X_k) | Z^k]$. If we take again N_P samples of (X_0, \dots, X_k) according to the probability mass function (pmf) with probabilities $(w_k(i))_{i=1}^{N_P}$, we could approximate “again” the conditional expectation as

$$E[g(X_k) | Z^k] \approx \frac{1}{N_P} \sum_{i=1}^{N_P} g(\tilde{x}_k(i)) \quad (2.11)$$

where $(\tilde{x}_0(i), \dots, \tilde{x}_k(i))_{i=1}^{N_P}$ is the set of samples obtained by the resampling procedure. By making $x_k(i) := \tilde{x}_k(i)$ and $w_k(i) := 1/N_P$ for $i = 1, \dots, N_P$, we obtain a new set of particles with equal weights, that we can use in the next iteration of the algorithm. The modified particle filter algorithm is then called Sequential Importance Resampling Particle Filter (SIR PF).

2. MATHEMATICAL BACKGROUND

Note that, at the next iteration $k + 1$, when we sample from

$$q(x_{k+1} | x_0, \dots, x_k, Z^{k+1}),$$

the new samples $(x_0(i), \dots, x_{k+1}(i))_{i=1}^{N_P}$ cannot be assumed as independent, as some of them will share common past trajectories (x_0, \dots, x_k) . Therefore, convergence in expectation due to the law of large numbers is not guaranteed. However, asymptotic convergence results for the SIR PF, including convergence in distribution, in mean-square error, and in expectation (under different assumptions) have been provided by Crisan and Doucet [2002] and Hu et al. [2008].

Resampling has also some practical drawbacks. As observed by Liu and Chen [1998], in case all particles have already nearly all equal weights, resampling will merely reduce the number of distinct particles, hence impoverishing the particle approximation. Liu and Chen [1998]’s proposed solution is to perform resampling only after every couple of steps, when the variance of the weights exceeds some heuristic threshold. Another solution is to use the *systematic resampling* (also known as “stratified sampling”) scheme described by Kitagawa [1996], which guarantees that no particle is eliminated in case all particles have already equal weights. Alternatively, systematic resampling and threshold-based resampling can be combined, as suggested by Arulampalam et al. [2002], although they do not explicitly state the advantages of this approach.

Note that systematic resampling has other advantages. Compared to other resampling schemes, it has low computational cost (see Hol et al. [2006]) and it results in minimum variance in the selection of samples ([de Freitas, 1999, Chapter 6]). For the reader’s information, in all our SIR PF implementations, we use systematic resampling at every time step. The systematic resampling algorithm is presented in Section 2.1.2.3.

The computational complexity of the SIS PF is $O(N_P)$. According to Hol et al. [2006], all commonly used resampling schemes can also be implemented with $O(N_P)$ complexity; therefore, we can say that the SIR PF as a whole has $O(N_P)$ complexity.

2. MATHEMATICAL BACKGROUND

2.1.2.2 SIR PF algorithm

Initialization:

1. For each particle $i = 1, \dots, N_P$

- (a) Sample $x_0(i) \sim p(x_0)$

- (b) Make $w_0(i) = \frac{1}{N_P}$

At every time step $k = 1, 2, \dots$:

1. For each particle $i = 1, \dots, N_P$

- (a) Perform importance sampling by making

$$x_k(i) \sim q \left(x_k \mid x_0(i), \dots, x_{k-1}(i), Z^k \right)$$

where $q(x_k \mid x_0, \dots, x_{k-1}, Z^k)$ is a proposal density

- (b) Calculate the unnormalized weight according to

$$\begin{aligned} \bar{w}_k(i) &= \frac{p(z_k \mid x_0(i), \dots, x_k(i), Z^{k-1}) p(x_k(i) \mid x_0(i), \dots, x_{k-1}(i), Z^{k-1})}{q(x_k(i) \mid x_0(i), \dots, x_{k-1}(i), Z^k)} \\ &\quad \times w_{k-1}(i) \end{aligned}$$

2. Normalize the particle weights according to

$$w_k(i) = \frac{\bar{w}_k(i)}{\sum_{j=1}^{N_P} \bar{w}_k(j)}, \quad i = 1, \dots, N_P$$

3. Perform resampling by sampling N_P indexes $(\tilde{j}(i))_{i=1}^{N_P}$ according to the pmf $(w_k(j))_{j=1}^{N_P}$ and afterwards making

$$\begin{aligned} (x_0(i), \dots, x_k(i)) &:= (x_0(\tilde{j}(i)), \dots, x_k(\tilde{j}(i))) \\ w_k(i) &:= \frac{1}{N_P}, \quad i = 1, \dots, N_P \end{aligned}$$

2.1.2.3 Systematic resampling algorithm for a time step k

1. Generate a random number $u(1) \sim \mathcal{U}(0, 1/N_P)$
2. Make $F(1) := w_k(1)$

2. MATHEMATICAL BACKGROUND

3. For each particle $i = 2, \dots, N_P$, make $F(i) := F(i - 1) + w_k(i)$, such that $F(\cdot)$ corresponds to the cumulative distribution function (cdf) of the particles
4. Make $j := 1$
5. For each particle $i = 1, \dots, N_P$
 - (a) Make $u(i) := u(1) + (i - 1)/N_P$
 - (b) While $u(i) > F(j)$, make $j := j + 1$
 - (c) Make $\tilde{j}(i) := j$

2.1.2.4 SIR PF degeneracy

Although the SIR PF has been successfully employed in various estimation problems, it is known to lead to poor performance for certain classes of problems, including: joint state and parameter estimation (see Andrieu and Doucet [2002]; Kantas et al. [2009]), multi-modal estimation (Vermaak et al. [2003]), classification (Blom and Bloem [2004]), smoothing (Briers et al. [2010]) and track labelling (Boers et al. [2010]). This is also due to a type of degeneracy, which is however different in nature from the degeneracy that affects the SIS PF, and has different consequences.

Particle filters provide, in theory, an approximation of $p(x_0, \dots, x_k | Z^k)$, the posterior for the entire state trajectory. But since the dimensionality of the state trajectory increases with time, it is intuitive that no numerical approximation of the trajectory posterior based on a fixed number of samples could be effective – this is precisely the reason that the SIS PF is doomed to fail when time advances enough.

The SIR PF attempts to counter the problem with time-increasing dimension by progressively *sacrificing* information about past states, such that it is at least able to satisfactorily estimate the current state X_k . In other words, as stated in Doucet and Johansen [2011], the resampling mechanism causes the marginal density of the current state $p(x_k | Z^k)$ to be better approximated than the marginal densities of any “earlier” state, of the form $p(x_j | Z^k)$, $j \in [0, k)$.

This mechanism is illustrated in Fig. 2.1. In this example, at time $k = 1$, before the resampling step, there are $N_P = 6$ particles and hence 6 distinct

2. MATHEMATICAL BACKGROUND

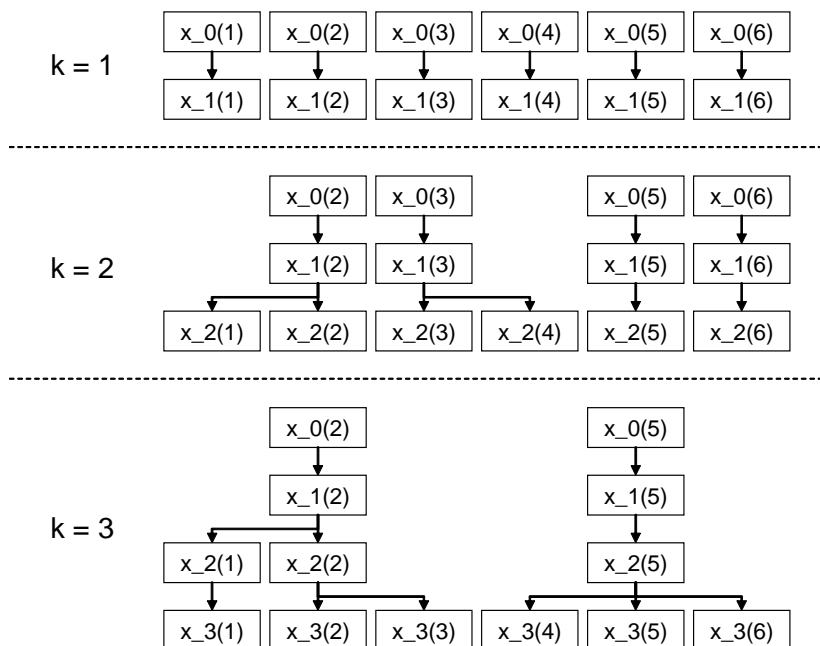


Figure 2.1: Illustration of a running SIR PF, showing the particles produced at each time step, after importance sampling and **before** resampling

hypotheses on the trajectory (X_0, X_1) . After the resampling step, only 4 distinct hypotheses remain. Importance sampling at time $k = 2$ generates again 6 distinct trajectories, but all of them contain one of the previous 4 distinct hypotheses for the (X_0, X_1) part of the trajectory. After another resampling, followed by importance sampling at time $k = 3$, only two distinct hypotheses for (X_0, X_1) , plus three distinct hypotheses for X_2 , remain.

After a number of time steps, the statistical information about the past trajectory (X_0, X_1) will collapse into a single hypothesis. Eventually the same will happen with X_2 , X_3 and so on. It is easy to see that this phenomenon will occur regardless the number of particles – after all, the number of particles representing the trajectory (X_0, \dots, X_j) , $j < k$, can only decrease when k increases. In other words, the SIR PF suffers from degeneracy on estimating past states, i.e. on performing smoothing.

This behavior may look acceptable if we are interested only in filtering, i.e. in estimating a function of the current state $g(X_k)$. The problem is that, if for

2. MATHEMATICAL BACKGROUND

some $j < k$, the statistical information about all past states including and until time j is represented by a single hypothesis (say (x_0^*, \dots, x_j^*)), then in practice, the particle approximation of the posterior $p(x_0, \dots, x_k | Z^k)$ can be considered *biased* towards

$$p(x_{j+1}, \dots, x_k | x_0^*, \dots, x_j^*, Z^k) \tag{2.12}$$

and therefore, unless filtering is indifferent to the aforementioned past hypothesis, i.e.

$$E[g(X_k) | Z^k] \approx E[g(X_k) | x_0^*, \dots, x_j^*, Z^k], \tag{2.13}$$

filtering is also going to be affected by degeneracy.

According to Crisan and Doucet [2002], a number of “mixing” properties ensures that a particular model will not suffer adverse consequences from degeneracy, more precisely, that the expected estimation errors have upper bounds that do not increase with time. As mentioned in the same work, these properties are highly restrictive, and although sufficient, they are perhaps not necessary. In practice, SIR particle filters have been successfully employed in several practical problems, which mostly do not have those mixing properties (e.g. Doucet et al. [2001]; Gustafsson and Saha [2010]; Kreucher et al. [2005]).

A common situation is that the “indifference property” (2.13) will hold only if the difference $k - j$ is large enough, i.e. if degeneracy takes a sufficiently long time to occur. In this case, we may attempt to “slow down” degeneracy to counter its adverse effects. This can be accomplished by increasing the number of particles or by decreasing the variance of the weights (which intuitively results in less distinct particles disappearing during resampling).

2.1.3 Particle filters applied to Partially Observed First Order Markov (POM1) processes

When designing discrete time models for describing real-world processes, a common assumption is that the behavior of the sequence of states and observations

2. MATHEMATICAL BACKGROUND

can be summarized by

$$X_{k+1} = f_k(X_k, M_k) \quad (2.14)$$

$$Z_k = h_k(X_k, N_k) \quad (2.15)$$

$$X_0 \sim p(x_0) \quad (2.16)$$

where $p(x_0)$ is referred to as the *prior* pdf, f_k and h_k are arbitrary functions, and $(M_k)_{k=0}^{\infty}$ and $(N_k)_{k=1}^{\infty}$ are all mutually independent random variables, also independent from $p(x_0)$. The stochastic process M_k is called the *process noise* and N_k the *measurement noise*. A particular case of this model is the linear-Gaussian model, where f_k and h_k have the form:

$$f_k(X_k, M_k) = F_k X_k + M_k \quad (2.17)$$

$$h_k(X_k, N_k) = H_k X_k + N_k \quad (2.18)$$

where F_k and H_k are linear transformations, and M_k and N_k are distributed according to

$$\begin{aligned} M_k &\sim \mathcal{N}(u_k, Q_k) \\ N_k &\sim \mathcal{N}(v_k, R_k) \end{aligned} \quad (2.19)$$

where u_k and v_k are the means of the noise terms, and Q_k and R_k their covariances.

In Bayesian filtering, the density $p(z_k | x_0, \dots, x_k, Z^{k-1})$ is commonly called the *likelihood function*, and the density $p(x_k | x_0, \dots, x_{k-1}, Z^{k-1})$ the *state transition density*. If the process/observation model is described by (2.14) and (2.15), these densities can be simplified to

$$p(z_k | x_0, \dots, x_k, Z^{k-1}) = p(z_k | x_k), \quad (2.20)$$

$$p(x_k | x_0, \dots, x_{k-1}, Z^{k-1}) = p(x_k | x_{k-1}) \quad (2.21)$$

i.e. the observation at time k , conditioned on the state at time k is independent of previous states and observations, and the state at time k , conditioned on the state at time $k-1$, is independent of states older than $k-1$ and of previous

2. MATHEMATICAL BACKGROUND

observations. A system with properties (2.20) and (2.21) is called a Partially Observed First Order Markov, or Partially Observed Markov-1 (POM1), process.

The main disadvantage of POM1 processes lies in the fact that real-world processes are more accurately modeled in continuous time, rather than in discrete time. As remarked by Gustafsson and Saha [2010], deriving a discrete time model from a continuous time model does not result in a model of the form (2.14), (2.15), unless unphysical assumptions, such as treating the process noise M_k as piecewise constant between two time steps, are made.

As we have seen in Section 2.1.1.3, the optimal proposal density is given by

$$q(x_k | x_0, \dots, x_{k-1}, Z^k) = p(x_k | x_0, \dots, x_{k-1}, Z^k)$$

and in the case of POM1 processes, observe that

$$\begin{aligned} p(x_k | x_0, \dots, x_{k-1}, Z^k) &= \frac{p(z_k | x_0, \dots, x_k, Z^{k-1}) p(x_k | x_0, \dots, x_{k-1}, Z^{k-1})}{p(z_k | x_0, \dots, x_{k-1}, Z^{k-1})} \\ &= \frac{p(z_k | x_k) p(x_k | x_{k-1})}{p(z_k | x_0, \dots, x_{k-1}, Z^{k-1})} \\ &= \frac{p(z_k | x_k) p(x_k | x_{k-1}) p(x_k | x_{k-1}, z_k)}{p(z_k | x_0, \dots, x_{k-1}, Z^{k-1}) p(x_k | x_{k-1}, z_k)} \\ &= \frac{p(z_k | x_k) p(x_k | x_{k-1}) p(x_k | x_{k-1}, z_k) p(z_k | x_{k-1})}{p(z_k | x_0, \dots, x_{k-1}, Z^{k-1}) p(z_k | x_k, x_{k-1}) p(x_k | x_{k-1})} \\ &= \frac{p(z_k | x_k) p(x_k | x_{k-1}, z_k) p(z_k | x_{k-1})}{p(z_k | x_0, \dots, x_{k-1}, Z^{k-1}) p(z_k | x_k)} \\ &= \frac{p(x_k | x_{k-1}, z_k) p(z_k | x_{k-1})}{p(z_k | x_0, \dots, x_{k-1}, Z^{k-1})} \end{aligned} \tag{2.22}$$

and by noticing that, for purposes of importance sampling, terms $p(z_k | x_{k-1})$ and $p(z_k | x_0, \dots, x_{k-1}, Z^{k-1})$ are irrelevant (as they do not depend on x_k), the optimal PF proposal density for POM1 processes is given by

$$\begin{aligned} q(x_k | x_0, \dots, x_{k-1}, Z^k) &= q(x_k | x_{k-1}, z_k) \\ &= p(x_k | x_{k-1}, z_k). \end{aligned} \tag{2.23}$$

For many problems, sampling directly from $p(x_k | x_{k-1}, z_k)$ is not possible.

2. MATHEMATICAL BACKGROUND

A popular alternative is instead to sample from the state transition density $p(x_k | x_{k-1})$, leading to a higher variance of the weights. Since sampling from the state transition density disregards the last observation z_k , it is commonly referred to in literature as *blind importance sampling*.

2.1.4 The Rao-Blackwellized Particle Filter (RBPF)

2.1.4.1 Mechanism

As we mentioned in Section 2.1.4, a low variance of weights is desirable to slow down degeneracy. For certain models, one way of achieving it is through the *Rao-Blackwellization* technique presented in Andrieu and Doucet [2002]. Consider a stochastic process with state vector of the form $X_k = [S_k^T, T_k^T]^T$, with realizations denoted by $x_k = [s_k^T, t_k^T]^T$, i.e. S_k and T_k are “sub-states”. Rao-Blackwellization consists of replacing the SMC mechanism by two *parallel interacting estimators*: part of the state X_k (say, S_k) is estimated using a SMC filter (a PF or one of its variants), and the another part (say, T_k) is estimated using a non-SMC method (typically an analytical method, like a Kalman filter). More specifically, the RBPF attempts to approximate the density

$$\begin{aligned} p(x_k, s_0, \dots, s_{k-1} | Z^k) &= p(s_0, \dots, s_k, t_k | Z^k) \\ &= p(s_0, \dots, s_k | Z^k) p(t_k | s_0, \dots, s_k, Z^k) \end{aligned}$$

where $p(s_0, \dots, s_k | Z^k)$ is calculated using an adapted PF and $p(t_k | s_0, \dots, s_k, Z^k)$ is calculated using a non-SMC filter. Since the PF is applied to the marginal $p(s_0, \dots, s_k | Z^k)$ of $p(x_k, s_0, \dots, s_{k-1} | Z^k)$, the RBPF is also called *marginalized particle filter*. The resulting approximated pdf is then given by

$$p(s_0, \dots, s_k, t_k | Z^k) \approx \sum_{i=1}^{N_P} w_k(i) \left(\prod_{j=0}^k \delta(s_j - s_j(i)) \right) p(t_k | s_0(i), \dots, s_k(i), Z^k) \quad (2.24)$$

where $(s_0(i), \dots, s_k(i))_{i=1}^{N_P}$ denote the particles states and $(w_k(i))_{i=1}^{N_P}$ the particle weights, with each particle being a weighted hypothesis on the trajectory

2. MATHEMATICAL BACKGROUND

(S_0, \dots, S_k) . Naturally, approximation (2.24) is only useful if we have means of calculating $p(t_k | s_0, \dots, s_k, Z^k)$ without resorting to SMC methods. One such situation is of a POM1 system which has the structure given by

$$S_{k+1} = f_k^s(S_k, M_k^s) \quad (2.25)$$

$$T_{k+1} = F_k^t(S_{k+1})T_k + M_k^t(S_{k+1}) \quad (2.26)$$

$$Z_k = H_k(S_k)T_k + N_k(S_k) \quad (2.27)$$

where f_k^s is a function, M_k^s is a random variable, and for fixed arguments ($S_{k+1} = s_{k+1}$, $S_k = s_k$), $F_k^t(s_{k+1})$ and $H_k(s_k)$ are linear transformations, and $M_k^t(s_{k+1})$ and $N_k(s_k)$ have Gaussian distribution, i.e. they can be written as

$$M_k^t(s_{k+1}) \sim \mathcal{N}(u_k^t(s_{k+1}), Q_k^t(s_{k+1})) \quad (2.28)$$

$$N_k(s_k) \sim \mathcal{N}(v_k(s_k), R_k(s_k)). \quad (2.29)$$

We also assume $(M_k^s)_{k=0}^\infty$, $(M_k^t(s_{k+1}))_{k=0}^\infty$ and $(N_k(s_k))_{k=1}^\infty$ to be all mutually independent and independent from $p(x_0)$. For this model, it is possible to show that assuming that $p(t_0 | s_0)$ is Gaussian, we have

$$p(t_k | s_0, \dots, s_k, Z^k) = \mathcal{N}(t_k; \hat{t}_k, P_k^t) \quad (2.30)$$

where \hat{t}_k and P_k^t depend both on s_0, \dots, s_k , and can be recursively calculated (from \hat{t}_{k-1} , P_{k-1}^t , s_k and z_k) using a Kalman filter.

2.1.4.2 RBPF algorithm

For illustrative purposes, we present the RBPF algorithm for the structured model given by (2.25)–(2.27). Applications of the RBPF to more general structured system models can be found in Schön et al. [2005].

Initialization:

1. For each particle $i = 1, \dots, N_P$
 - (a) Sample $s_0(i) \sim p(s_0)$
 - (b) Initialize $\hat{t}_0(i)$, $P_0^t(i)$ (respectively the mean and covariance of $p(t_0 | s_0(i))$)

2. MATHEMATICAL BACKGROUND

(c) Make $w_0(i) = \frac{1}{N_P}$

At every time step $k = 1, 2, \dots$:

1. For each particle $i = 1, \dots, N_P$

(a) Perform importance sampling by making

$$s_k(i) \sim q\left(s_k \mid s_0(i), \dots, s_{k-1}(i), Z^k\right)$$

where $q(s_k \mid s_0, \dots, s_{k-1}, Z^k)$ is a proposal density

(b) Perform the Kalman filter steps

$$\begin{aligned} \hat{t}_{k|k-1}(i) &= F_{k-1}^t(s_k(i))\hat{t}_{k-1}(i) + u_{k-1}^t(s_k(i)) \\ P_{k|k-1}^t(i) &= F_{k-1}^t(s_k(i))P_{k-1}^t(i)F_{k-1}^t(s_k(i))^T + Q_{k-1}^t(s_k(i)) \\ \hat{z}_{k|k-1}(i) &= H_k(s_k(i))\hat{t}_{k|k-1}(i) + v_k(s_k(i)) \\ V_{k|k-1}(i) &= H_k(s_k(i))P_{k|k-1}^t(i)H_k(s_k(i))^T + R_k(s_k(i)) \\ K_k(i) &= P_{k|k-1}^t(i)H_k(s_k(i))^T V_{k|k-1}(i)^{-1} \\ \hat{t}_k(i) &= \hat{t}_{k|k-1}(i) + K_k(i)(z_k - \hat{z}_{k|k-1}(i)) \\ P_k^t(i) &= P_{k|k-1}^t(i) - K_k(i)H_k(s_k(i))P_{k|k-1}^t(i) \end{aligned}$$

(c) Calculate the unnormalized weight according to

$$\bar{w}_k(i) = \frac{\mathcal{N}(z_k; \hat{z}_{k|k-1}, V_{k|k-1}) p(s_k(i) \mid s_{k-1}(i))}{q(s_k(i) \mid s_0(i), \dots, s_{k-1}(i), Z^k)} w_{k-1}(i)$$

2. Normalize the particle weights according to

$$w_k(i) = \frac{\bar{w}_k(i)}{\sum_{j=1}^{N_P} \bar{w}_k(j)}, \quad i = 1, \dots, N_P$$

3. Perform resampling by sampling N_P indexes $(\tilde{j}(i))_{i=1}^{N_P}$ according to the pmf $(w_k(j))_{j=1}^{N_P}$ and afterwards making

$$\begin{aligned} (s_0(i), \dots, s_k(i), \hat{t}_k(i), P_k^t(i)) &:= (s_0(\tilde{j}(i)), \dots, s_k(\tilde{j}(i)), \hat{t}_k(\tilde{j}(i)), P_k^t(\tilde{j}(i))) \\ w_k(i) &:= \frac{1}{N_P}, \quad i = 1, \dots, N_P \end{aligned}$$

2.1.4.3 Benefits and drawbacks

As shown in Andrieu and Doucet [2002], Rao-Blackwellization results in a reduction of the variance of the weights. This result is intuitive, as we are effectively estimating at least part of the state (or more precisely, its conditional density resulting from the factorization of the posterior density) using an exact Bayes estimator (in the case of the algorithm in Section 2.1.4.2, a Kalman filter), rather than a Monte Carlo-based approximation.

Although Rao-Blackwellization requires extra processing steps, it also reduces the dimensionality of the state that needs to be estimated using a particle filter, hence allowing us to use smaller numbers of particles. For a given performance standard, the RBPF will typically have lower computational cost than the SIR PF applied to the entire state $X_k = [S_k^T, I_k^T]^T$.

Therefore, the only significant “drawback” of the RBPF is that it requires the system model to be structured in some manner, typically having some linear-Gaussian properties or containing discrete variables, such that $p(t_k | s_0, \dots, s_k, Z^k)$ can be computed analytically. If that is not the case, additional analytical or numerical approximations (such as using an Extended Kalman Filter instead of a Kalman filter) will be necessary for Rao-Blackwellization, which may naturally have a negative impact on performance.

2.1.5 The Marginal Particle Filter (MPF)

2.1.5.1 Mechanism

In a SIR PF, a particle i at time k is given by $(x_0(i), \dots, x_k(i))$, i.e. it is a hypothesis on the entire trajectory, since according to the sequential sampling mechanism, $x_k(i)$ is sampled assuming the entire past trajectory $(x_0(i), \dots, x_{k-1}(i))$ as the “true” past trajectory. As we have seen in Section 2.1.2.4, such behavior can be detrimental to performance, as the degeneracy phenomenon progressively reduces the diversity of past trajectories.

One may then have the following idea: to derive a Monte-Carlo based algorithm where each particle represents only the *current* state X_k , with each particle $x_k(i)$ being drawn without assuming any particular value for past states. Instead,

2. MATHEMATICAL BACKGROUND

each of the particles is drawn assuming only the posterior approximation obtained in the previous iteration. The result from this idea is the *Marginal Particle Filter* (MPF) algorithm presented by Klaas et al. [2005]. The algorithm has this name because it attempts to compute only the density $p(x_k | Z^k)$ obtained by the marginalization of $p(x_0, \dots, x_k | Z^k)$. It should not be confused with the Marginalized PF (which, as we mentioned, is just another name for the RBPF).

To derive the MPF, consider again that we would like to estimate some function $g(X_k)$ of the “current” state X_k . For the sake of simplicity, we will also assume a POM1 process. The expectation of this quantity, given the available observations Z^k , can be expressed as

$$\begin{aligned} E[g(X_k) | Z^k] &= \int g(x_k) p(x_k | Z^k) dx_k \\ &= \int g(x_k) \frac{p(z_k | x_k) \int p(x_k | x_{k-1}) p(x_{k-1} | Z^{k-1}) dx_{k-1}}{p(z_k | Z^{k-1})} dx_k. \end{aligned} \tag{2.31}$$

Assuming that $p(x_{k-1} | Z^{k-1})$ is approximated by a set of particles

$$(x_{k-1}(j), w_{k-1}(j))_{j=1}^{N_P}$$

(presumably generated during the previous iteration of the algorithm), (2.31) can be approximated as

$$\begin{aligned} E[g(X_k) | Z^k] &\approx \int g(x_k) \frac{p(z_k | x_k) \sum_{j=1}^{N_P} w_{k-1}(j) p(x_k | x_{k-1}(j))}{p(z_k | Z^{k-1})} dx_k \\ &= \int q(x_k | Z^k) g(x_k) \frac{p(z_k | x_k) \sum_{j=1}^{N_P} w_{k-1}(j) p(x_k | x_{k-1}(j))}{p(z_k | Z^{k-1}) q(x_k | Z^k)} dx_k \end{aligned} \tag{2.32}$$

where $q(x_k | Z^k)$ is a proposal density parametrized on the entire sequence of observations Z^k . By drawing N_P samples from $q(x_k | Z^k)$, we obtain the approx-

2. MATHEMATICAL BACKGROUND

imation

$$E [g(X_k) | Z^k] \approx \sum_{i=1}^{N_P} w_k(i) g(x_k(i)) \quad (2.33)$$

where

$$w_k(i) = \frac{p(z_k | x_k(i)) \sum_{j=1}^{N_P} w_{k-1}(j) p(x_k(i) | x_{k-1}(j))}{N_P p(z_k | Z^{k-1}) q(x_k(i) | Z^k)} \quad (2.34)$$

where the term $N_P p(z_k | Z^{k-1})$ does not depend on the sample and hence does not need to be calculated explicitly, as in the SIR/SIS PFs. Consequently, the current state posterior is approximated as

$$p(x_k | Z^k) \approx \sum_{i=1}^{N_P} w_k(i) \delta(x_k - x_k(i)). \quad (2.35)$$

Note that the MPF does not have a resampling step. However, sampling from the proposal density may involve a step similar to resampling. This is because a convenient form for the proposal density $q(x_k | Z^k)$ (in the sense that it takes advantage of the particle approximation of $p(x_{k-1} | Z^{k-1})$) is

$$q(x_k | Z^k) = \sum_{j=1}^{N_P} \lambda_k(j) q(x_k | x_{k-1}(j), z_k) \quad (2.36)$$

where $\sum_{j=1}^{N_P} \lambda_k(j) = 1$, and $\lambda_k(j)$ may depend on z_k , $x_{k-1}(j)$ and $w_{j-1}(j)$. In order to obtain a new set of particles by sampling from (2.36), we can use two steps:

1. Use a resampling algorithm (such as the systematic resampling scheme from Section 2.1.2.3) to obtain samples $\tilde{x}_{k-1}(i)$, $i = 1, \dots, N_P$ of X_{k-1} from the point-mass distribution with probabilities $(\lambda_k(j))_{j=1}^{N_P}$;
2. Sample $x_k(i)$, $i = 1, \dots, N_P$ according to $q(x_k | \tilde{x}_{k-1}(i), z_k)$.

The optimal proposal density of the MPF is not mentioned in Klaas et al. [2005], but for the interested reader, we have derived it and present it in Appendix

A.

2.1.5.2 MPF algorithm

We present below the MPF for POM1 processes. The algorithm can be easily extended to more general processes.

Initialization:

1. For each particle $i = 1, \dots, N_P$
 - (a) Sample $x_0(i) \sim p(x_0)$
 - (b) Make $w_0(i) = \frac{1}{N_P}$

At every time step $k = 1, 2, \dots$:

1. For each particle $i = 1, \dots, N_P$
 - (a) Perform importance sampling for the state vector by making

$$x_k(i) \sim q\left(x_k \mid Z^k\right)$$

where $q(x_k \mid Z^k)$ is a proposal density:

- (b) Calculate the unnormalized weight according to

$$\bar{w}_k(i) = \frac{p(z_k \mid x_k(i)) \sum_{j=1}^{N_P} w_{k-1}(j) p(x_k(i) \mid x_{k-1}(j))}{q(x_k(i) \mid Z^k)}$$

2. Normalize the particle weights according to

$$w_k(i) = \frac{\bar{w}_k(i)}{\sum_{j=1}^{N_P} \bar{w}_k(j)}, \quad i = 1, \dots, N_P$$

2.1.5.3 Benefits and drawbacks

The drawback of the MPF is evident, when one notes that each particle weight is calculated using all particles, rather than a single particle. This causes the algorithm to have an asymptotic complexity of $O(N_P^2)$ (holding all else constant) instead of $O(N_P)$ as in SIR PF. However, as shown in Klaas et al. [2005], using all particles to calculate each weight also has a “smoothing” effect on the weights, leading to a smaller, or at least equal, variance.

2. MATHEMATICAL BACKGROUND

It may be tempting, for the sake of ease of implementation, to use a proposal density that disregards the last measurement z_k . As in the SIR PF, such density is called a “blind proposal density”:

$$q(x_k | Z^k) = p(x_k | Z^{k-1}) \approx \sum_{j=1}^{N_P} w_{k-1}(j) p(x_k(i) | x_{k-1}(j)). \quad (2.37)$$

The problem is that as shown in Klaas et al. [2005], using (2.37) will result in the MPF being mathematically equivalent to the SIR PF with “blind proposal density”. Therefore, using a MPF with a blind proposal density does not provide any benefit in terms of reduction of variance; in fact, it also does not increase the computational complexity as the term $\sum_{j=1}^{N_P} w_{k-1}(j) p(x_k(i) | x_{k-1}(j))$ will be canceled in the calculation of the weight.

2.2 Joint state and parameter estimation

2.2.1 Mathematical formulation

In the parameter estimation problem (or more specifically, the *joint state and parameter estimation problem*), the system is a partially observed process consisting of (S_k, θ, Z_k) , where k is the time index, S_k is a random vector corresponding to the time-varying state (with the corresponding realization denoted by s_k), θ is a vector corresponding to the unknown parameter, and Z_k is again a random vector corresponding to the observation process (with the corresponding realization denoted by z_k). Our goal is to obtain estimates $\hat{\theta}$ and \hat{s}_k respectively of θ and S_k , given all available observations $Z^k = (z_1, \dots, z_k)$.

This process may also be modeled as a POM1 process, described as

$$\begin{aligned} S_{k+1} &= f_k(S_k, \theta, M_k) \\ Z_k &= h_k(S_k, \theta, N_k) \\ S_0 &\sim p(s_0) \end{aligned} \quad (2.38)$$

where f_k and h_k are arbitrary functions, and $(M_k)_{k=0}^{\infty}$ and $(N_k)_{k=1}^{\infty}$ are all mutually

2. MATHEMATICAL BACKGROUND

independent random variables, also independent from $p(s_0)$.

Alternatively, the unknown parameter may also be modeled as a random vector Θ (with realizations given by θ). In this case, for a POM1 process, we describe the system by

$$\begin{aligned} S_{k+1} &= f_k(S_k, \Theta, M_k) \\ Z_k &= h_k(S_k, \Theta, N_k) \\ \begin{bmatrix} S_0 \\ \Theta \end{bmatrix} &\sim p(s_0, \theta) \end{aligned} \tag{2.39}$$

where $(M_k)_{k=0}^\infty$ and $(N_k)_{k=1}^\infty$ are also independent from $p(s_0, \theta)$.

In practice, joint state and parameter estimation practical problems can be divided into two major groups:

1. In **offline estimation** problems, the parameter estimate $\hat{\theta}$ needs to be produced only after a finite sequence of measurements $Z^{k_{\max}}$ is available. There are no hard constraints on the processing time to obtain this estimate, although computational efficiency is often desirable. The estimate $\hat{\theta}$ may subsequently be assumed to be the “true” value of θ and used in pure state estimation problems (i.e. where only S_k is estimated);
2. In **online estimation** problems, parameter and state estimates need to be produced at recurring times, for instance, as soon as new measurements become available. Hence, there are hard constraints on the processing time, and estimation is typically implemented using some sort of recursion. It is also convenient to write each parameter estimate as $\hat{\theta}_k$, indicating that the estimate uses the observations available until and including time k . Online joint state and parameter estimation generally assumes no upper bound on the length of the sequence Z^k , which as we will see in Chapter 3, causes the problem to be tricky to solve using SMC methods (due to the degeneracy phenomenon).

2.2.2 Parameter estimation strategies

We look now at the three main strategies used for joint state and parameter estimation. Although all of them are suitable for offline and online estimation problems, we will focus on online estimation problems (and hence write the parameter estimate as $\hat{\theta}_k$).

2.2.2.1 Maximum Likelihood approach

The first strategy is the *Maximum Likelihood* (ML) approach, that treats the parameters as unknown, deterministic variables (i.e. we consider the model given by (2.38)), estimated using

$$\hat{\theta}_k = \arg \sup_{\theta} p(Z^k | \theta). \quad (2.40)$$

An example of nonlinear parameter estimation algorithm based on the ML strategy is the Expected Maximization (EM) algorithm (Dempster et al. [1977]) combined with SMC smoothing, used for instance in Olsson et al. [2008]; Schön et al. [2011].

The estimate $\hat{\theta}_k$ is afterwards treated as the true value of the parameter for the purpose of estimating the state S_k . Hence, in practice, the estimate \hat{s}_k (as well as other statistics of interest, such as the covariance of S_k) will be based on the conditional density $p(s_k | \hat{\theta}, Z^k)$. The estimate \hat{s}_k may be, for instance, the MAP estimate:

$$\hat{s}_k = \arg \sup_{s_k} p(s_k | \hat{\theta}_k, Z^k) \quad (2.41)$$

or, alternatively, the MMSE estimate:

$$\hat{s}_k = \int s_k p(s_k | \hat{\theta}_k, Z^k) ds_k. \quad (2.42)$$

2.2.2.2 Decoupled Bayesian approach

As mentioned in Syldatk et al. [2012], in case of low parameter identifiability (i.e. when the sequence of measurements Z^k may not be enough to obtain an accurate estimate $\hat{\theta}$), estimation can be improved by incorporation prior information about the parameter θ , if any. Doing so corresponds to the Bayesian strategy, where parameter vector is treated as a random variable, such that we consider the model given by (2.39). Estimation of states and parameters, in the Bayesian approach, can be either coupled or decoupled.

The *decoupled Bayesian* strategy basically consists of modifying the ML strategy to take the parameter prior information $p(\theta)$ into account, but still treating parameter and state estimation as decoupled problems. In this approach, the parameter vector is treated as a random variable (such that we consider the model given by (2.39)), and the estimate $\hat{\theta}$ is obtained as

$$\begin{aligned}\hat{\theta} &= \arg \sup_{\theta} p(\theta | Z^k) \\ &= \arg \sup_{\theta} p(Z^k | \theta) p(\theta)\end{aligned}\tag{2.43}$$

and we then replace Θ with $\hat{\theta}$ in order to estimate S_k , as in the ML approach. The decoupled Bayesian strategy is used for instance in the version of the EM/SMC smoothing algorithm that uses the parameter prior density (Syldatk et al. [2012]) and in some sensor bias estimation algorithms, like Lin et al. [2004].

2.2.2.3 Coupled Bayesian approach

For certain problems with low parameter identifiability, accurate parameter estimates may not be achievable even using the prior information, making $p(s_k | \hat{\theta}, Z^k)$ a poor approximation of $p(s_k | Z^k)$. As a simple example, let us suppose that S_k and Θ are both discrete random variables, where S_k can assume the values s_k^A and s_k^B , and Θ can assume the values θ^A and θ^B . Let us then assume that

$$\begin{aligned}P(s_k^A, \theta^A | Z^k) &= 0.45, & P(s_k^A, \theta^B | Z^k) &= 0, \\ P(s_k^B, \theta^A | Z^k) &= 0.10, & P(s_k^B, \theta^B | Z^k) &= 0.45.\end{aligned}$$

2. MATHEMATICAL BACKGROUND

In this case, the MAP estimate based on $p(\theta | Z^k)$ is $\hat{\theta}_k = \theta^A$. But then, we would have

$$P(s_k^A | \theta^A, Z^k) = 0.818 \text{ and } P(s_k^B | \theta^A, Z^k) = 0.182$$

whereas

$$P(s_k^A | Z^k) = 0.45 \text{ and } P(s_k^B | Z^k) = 0.55.$$

In order to avoid these discrepancies, an alternative is to tackle the problem using a *coupled Bayesian strategy*, where estimation is based on the *joint* posterior density $p(s_k, \theta | Z^k)$. Let us consider the augmented state $X_k = \begin{bmatrix} S_k \\ \Theta \end{bmatrix}$, with realizations denoted by x_k . In the coupled Bayesian approach, we could, for instance, obtain the joint MAP

$$\hat{x}_k = \arg \sup_{x_k} p(x_k | Z^k) \tag{2.44}$$

or the joint MMSE

$$\hat{x}_k = \int x_k p(x_k | Z^k) dx_k. \tag{2.45}$$

In line with our goal of characterization of uncertainty in estimation, the coupled Bayesian approach also allows us to calculate descriptive statistics about S_k and Θ that fully take the dependence between their errors into account, such as the joint covariance

$$P_k = \int (x_k - \hat{x}_k)(x_k - \hat{x}_k)^T p(x_k | Z^k) dx_k. \tag{2.46}$$

Examples of joint Bayesian nonlinear parameter estimation algorithms are the PF with artificial dynamics (Gordon et al. [1993], Higuchi [2001]) and the Liu and West Particle Filter (LWPF) (Liu and West [2001]).

2.3 Joint multi-target tracking

Many traditional estimation concepts such as probability densities, means, covariances, differential entropy, and so forth, rely on states and observations being represented as either scalars or vectors. For this reason, they are not directly applicable to the Multi-Target Tracking (MTT) problem, where we consider that the number of objects may need to be estimated itself, the number of observations may vary for a given state, and there may be uncertainty on which objects originated which observations.

Early MTT approaches consisted of treating the problem as a set of separate single-target tracking problems, each solved using a suitable Bayesian estimator, with the results of single-object estimation integrated by a separate process to form a global result. This led to development of algorithms such as the Nearest-Neighbor (NN), the Joint Probabilistic Data Association (JPDA), and the Multiple Hypothesis Tracking (MHT). These approaches were referred to by Mahler [2008] as “bottom-up” MTT, in the sense that they consisted in proposing a solution to a considerably simpler (i.e. “lower”) problem, and then using complementary steps to extend the solution to a more general (i.e. “higher”) problem.

The huge interest in a rigorous mathematical treatment of the MTT problem, as well as in approximately optimal Bayesian solutions for it, however, led to an alternative approach, based on the calculation of the posterior probability distribution that statistically represents the entire multi-target scenario. This approach was referred to by Mahler [2008] as “top-down” MTT, because it consists of first describing the most general and complex problem, and thereafter making successive approximations and simplifications until the problem becomes solvable. In order to extend traditional estimation concepts to multi-object scenarios, the Finite Set Statistics (FISST) and Poisson Point Process (PPP) may be used.

In the rest of this section, we are going to review some key concepts of FISST and PPP theory. This review will be restricted to the concepts that are effectively used throughout this thesis; for the interested reader, a detailed description of FISST can be found in Goodman et al. [1997]; Mahler [2007] and of PPP in Daley and Vere-Jones [2003].

2.3.1 Finite Set Statistics

2.3.1.1 Multi-object calculus

Let $\mathfrak{N} = \mathbb{R}^n \times \Pi$, where \mathbb{R}^n denotes the n -dimensional Euclidean space and Π is a discrete set of elements. Let also

- ρ be the product measure of the Lebesgue measure on \mathbb{R}^n with the counting measure;
- \mathfrak{A} be the hyperspace of all subsets A of \mathfrak{N} , i.e. containing every $A \subseteq \mathfrak{N}$, the null set \emptyset included;
- \mathfrak{X} be the hyperspace containing all *finite* subsets \mathbf{x} of \mathfrak{N} , i.e. containing all sets with the form $\mathbf{x} = \{x^{(1)}, \dots, x^{(t)}\}$, where t is the number of elements of the set and $x^{(i)} \in \mathfrak{N}$, $i \in \{1, \dots, t\}$, and also containing the null set \emptyset .

Consider a set function $\phi : \mathfrak{A} \rightarrow \mathbb{R}$ and a finite set function $f : \mathfrak{X} \rightarrow \mathbb{R}$. The *set integral* of $f(\mathbf{x})$ w.r.t. a subset A of \mathfrak{N} is defined as¹

$$\int_A f(\mathbf{x}) \delta \mathbf{x} \triangleq f(\emptyset) + \sum_{t=1}^{\infty} \frac{1}{t!} \int_{\underbrace{A \times \dots \times A}_t} f(\{x^{(1)}, \dots, x^{(t)}\}) \rho(dx^{(1)}) \dots \rho(dx^{(t)}) \quad (2.47)$$

and we also adopt the notation

$$\int f(\mathbf{x}) \delta \mathbf{x} \triangleq \int_{\mathfrak{N}} f(\mathbf{x}) \delta \mathbf{x}. \quad (2.48)$$

The *set derivative* of $\phi(A)$ w.r.t. a finite set \mathbf{x} is defined as

$$\frac{\delta \phi}{\delta \mathbf{x}}(A) \triangleq \lim_{\rho(\lambda_{\mathbf{x}}) \searrow 0} \frac{\phi(A \cup \lambda_{\mathbf{x}}) - \phi(A)}{\rho(\lambda_{\mathbf{x}})} \quad (2.49)$$

where $\lambda_{\mathbf{x}}$ denotes a neighborhood of \mathbf{x} . The set integral and derivative are established as inverse operations by the *fundamental theorem of multi-object calculus*,

¹A mathematically rigorous definition of the set integral requires additional explanations; see [Mahler, 2007, Section 11.3.3.1].

2. MATHEMATICAL BACKGROUND

which states that

$$\phi(A) = \int_A \frac{\delta\phi}{\delta\mathbf{x}}(\emptyset)\delta\mathbf{x} \quad (2.50)$$

$$\left[\frac{\delta}{\delta\mathbf{x}} \int_A f(\mathbf{w})\delta\mathbf{w} \right]_{A=\emptyset} = f(\mathbf{x}). \quad (2.51)$$

The *multi-object Dirac delta density* is defined as follows:

$$\delta_{\mathbf{x}'}(\mathbf{x}) \triangleq \begin{cases} 0, & \text{if } |\mathbf{x}| \neq |\mathbf{x}'| \\ 1, & \text{if } \mathbf{x} = \mathbf{x}' = \emptyset \\ \sum_{\pi \in \Pi^t} \prod_{i=1}^t \delta(x^{(i)} - x'^{(\pi(i))}) & \text{if } \begin{cases} \mathbf{x} = \{x^{(1)}, \dots, x^{(t)}\} \\ \mathbf{x}' = \{x'^{(1)}, \dots, x'^{(t)}\} \end{cases} \end{cases} \quad (2.52)$$

where Π^t is the set of all permutations of the sequence $(1, \dots, t)$, and $|\cdot|$ denotes the cardinality operator. It is possible to show [Mahler, 2007, Appendix B.2] that

$$\int \delta_{\mathbf{x}'}(\mathbf{x})f(\mathbf{x})\delta\mathbf{x} = f(\mathbf{x}'). \quad (2.53)$$

2.3.1.2 Random Finite Sets

A *Random Finite Set (RFS)* \mathbf{X} , defined for the space \mathfrak{N} , is a random variable $\mathbf{X} : \mathfrak{N} \rightarrow \mathfrak{X}$. It may be represented as $\mathbf{X} = \{X^{(1)}, \dots, X^{(T)}\}$, where the cardinality T of the RFS is also a random variable. A RFS is appropriate to represent the state of a multi-object scenario (such as a multi-target tracking scenario), where $X^{(1)}, \dots, X^{(T)}$ denote the state of individual objects.

Let P be the *probability function* (see e.g. Klenke [2008]) associated with the RFS \mathbf{X} . The *Belief-mass function* (associated with \mathbf{X}) of a subset A of \mathfrak{N} is defined as

$$\beta_{\mathbf{X}}(A) \triangleq P(\mathbf{X} \subseteq A) \quad (2.54)$$

and, denoting a realization of \mathbf{X} by \mathbf{x} , the *RFS density function* $f(\mathbf{x})$ is defined

2. MATHEMATICAL BACKGROUND

as

$$f(\mathbf{x}) \triangleq \left[\frac{\delta \beta_{\mathbf{X}}(A)}{\delta \mathbf{x}} \right]_{A=\emptyset} \quad (2.55)$$

which, from the fundamental theorem of multi-object calculus (2.51), satisfies

$$\int_A f(\mathbf{x}) \delta \mathbf{x} = \beta_{\mathbf{X}}(A). \quad (2.56)$$

Conditional Belief-mass functions and conditional RFS densities may be defined using conditional probabilities. Consider the joint probability function $P(\mathbf{X} \subseteq A, \Gamma = \gamma)$, where Γ and γ denote respectively a random variable (also possibly a RFS) and an arbitrary realization. The conditional probability $P(\mathbf{X} \subseteq A | \Gamma = \gamma)$ is given by $\frac{P(\mathbf{X} \subseteq A, \Gamma = \gamma)}{P(\Gamma = \gamma)}$. From this quantity, we can describe the conditional Belief-mass and conditional RFS density as follows:

$$\int_A f(\mathbf{x}|\gamma) \delta \mathbf{x} = \beta_{\mathbf{X}|\Gamma}(A|\gamma) = P(\mathbf{X} \subseteq A | \Gamma = \gamma). \quad (2.57)$$

2.3.1.3 Estimation of Random Finite Sets

It can be shown (see Vo et al. [2005]) that the Bayes rule also applies to RFS densities, i.e. given a second RFS \mathbf{Z} with corresponding realizations \mathbf{z} , we have

$$f(\mathbf{x}|\mathbf{z}) = \frac{f(\mathbf{z}|\mathbf{x})f(\mathbf{x})}{f(\mathbf{z})}. \quad (2.58)$$

When estimating a random variable X , we are frequently interested in extracting a point estimate \hat{x} from a pdf $p(x)$ (generally conditioned on some sequence of observations). \hat{x} may be for instance the Maximum a Posteriori (MAP) estimate, or the Minimum Mean Square Error (MMSE) estimate. The MAP and the MMSE cannot, however, be applied to RFS densities. Careless attempts to extend these estimates to the multi-object estimation problems may lead to lack of robustness (as shown by Mahler [1999]) or coalescence of estimates (by Blom et al. [2008]).

In order to obtain a point estimate $\hat{\mathbf{x}}$ of a RFS \mathbf{X} , from a RFS density $f(\mathbf{x})$,

2. MATHEMATICAL BACKGROUND

Mahler [1999] proposes the *Joint Multi-target* (JoM) estimate, given by

$$\hat{\mathbf{x}} = \arg \sup_{\{x^{(1)}, \dots, x^{(t)}\}} \frac{c^t}{t!} f(\{x^{(1)}, \dots, x^{(t)}\}) \quad (2.59)$$

where c is a parameter with the same units of measurement of a single-object state $x^{(i)}$ (which, in case one decides to scale $x^{(i)}$ by changing its units of measurements, should be scaled accordingly). The argument of the arg sup function in (2.59) can be interpreted as the probability mass of the hypervolumes around $\{x^{(1)}, \dots, x^{(t)}\}$, if $f(\{x^{(1)}, \dots, x^{(t)}\})$ was discretized with separation c in the single-object space. The parameter c is discussed with more detail in [Mahler, 2007, Remark 2.2]. The JoM estimate can be considered as a multi-object version of the MAP estimate, since it is mathematically equivalent to the MAP in the case of a single object (i.e. $\mathbf{X} = \{X^{(1)}\}$).

Another useful statistic related to RFS estimation is the *Optimal Subpattern Assignment Metric* (OSPA) defined in Schuhmacher et al. [2008]. It corresponds to a measure of distance between two finite sets, which can be used, for instance, to evaluate the accuracy of a multi-object estimate $\hat{\mathbf{x}}$. The OSPA metric between the finite sets $\mathbf{x} = \{x^{(1)}, \dots, x^{(t)}\}$ and $\bar{\mathbf{x}} = \{\bar{x}^{(1)}, \dots, \bar{x}^{(\bar{t})}\}$ (where the single-object states $x^{(i)}$ and $\bar{x}^{(i)}$ assume values in an Euclidean space \mathbb{R}^n) is defined as follows. Let $1 \leq p < \infty$ be an order parameter that emphasizes the Euclidean distances between the objects of the two finite sets, and $c > 0$ be a cut-off parameter that emphasizes the difference in the number of objects. Let also $\Pi_{\bar{t}}$ be the set of all permutations on $(1, \dots, \bar{t})$, and $d^{(c)}(a, b)$, $a, b \in \mathbb{R}^n$ be defined by

$$d^{(c)}(a, b) \triangleq \min(\|a - b\|, c). \quad (2.60)$$

The OSPA metric between \mathbf{x} and $\bar{\mathbf{x}}$, parametrized by p and c , is then defined by

$$\epsilon_p^{(c)}(\mathbf{x}, \bar{\mathbf{x}}) \triangleq \left(\frac{1}{\bar{t}} \left(\min_{\pi \in \Pi_{\bar{t}}} \sum_{j=1}^t d^{(c)}(x^{(j)}, \bar{x}^{(\pi(j))})^p + c^p(\bar{t} - t) \right) \right)^{\frac{1}{p}} \quad (2.61)$$

2. MATHEMATICAL BACKGROUND

for $t \leq \bar{t}$, and

$$\epsilon_p^{(c)}(\mathbf{x}, \bar{\mathbf{x}}) \triangleq \epsilon_p^{(c)}(\bar{\mathbf{x}}, \mathbf{x}) \quad (2.62)$$

otherwise. From the OSPA metric, Guerriero et al. [2010] have defined the *Minimum Mean OSPA* (MMOSPA) estimate, given by

$$\hat{\mathbf{x}} \triangleq \arg \inf_{\bar{\mathbf{x}}} \int (\epsilon_p^{(c)}(\mathbf{x}, \bar{\mathbf{x}}))^p f(\mathbf{x}) \delta \mathbf{x}. \quad (2.63)$$

If the JoM can be considered as a multi-object version of the MAP estimate, the MMOSPA can be considered as a modified MMSE estimate applied to the multi-object estimation problem, with the square error used in the MMSE estimate being replaced by the p -th power of the OSPA metric. For $\mathbf{X} = \{X^{(1)}\}$, the MMOSPA and MMSE are mathematically equivalent for $p = 2$.

2.3.2 Poisson Point Process theory

PPP theory (Daley and Vere-Jones [2003]) provides a way of representing multi-object probability distributions without resorting to multi-object calculus. Although we mainly use FISST in this thesis, two PPP concepts – Janossy measures and Janossy densities, will prove useful in our work.

Consider again the space \mathfrak{N} and the measure ρ described in Section 2.3.1.1. Let T be a discrete random variable that assumes values in \mathbb{N} . For a given realization t of T , consider t random variables (in \mathfrak{N}) $X^{(1)}, \dots, X^{(t)}$. We now make the important assumption that the joint probability distribution of $X^{(1)}, \dots, X^{(t)}$ conditioned on t is *permutation-symmetric* w.r.t. its arguments, i.e. that given a set of subsets $\{A^{(1)}, \dots, A^{(t)}\}$ of \mathfrak{N} , we have

$$\begin{aligned} & P(X^{(1)} \in A^{(1)}, \dots, X^{(t)} \in A^{(t)} | T = t) \\ &= P(X^{(1)} \in A^{(\pi(1))}, \dots, X^{(t)} \in A^{(\pi(t))} | T = t) \end{aligned} \quad (2.64)$$

for any permutation π of $(1, \dots, t)$. With this assumption, we can define the

2. MATHEMATICAL BACKGROUND

Janossy measure J_t on the Borel σ -algebra of $\underbrace{\mathfrak{N} \times \dots \times \mathfrak{N}}_t$ as

$$\begin{aligned} J_t (A^{(1)} \times \dots \times A^{(t)}) \\ \triangleq t! P(T = t) P (X^{(1)} \in A^{(1)}, \dots, X^{(t)} \in A^{(t)} | T = t) \end{aligned} \quad (2.65)$$

and the *Janossy density* j_t is defined as the Radon-Nikodym derivative

$$j_t (x^{(1)}, \dots, x^{(t)}) \triangleq \frac{dJ_t}{d\underbrace{\rho \times \dots \times \rho}_t} \quad (2.66)$$

such that the following relationship holds

$$\begin{aligned} P (X^{(1)} \in A^{(1)}, \dots, X^{(t)} \in A^{(t)} | T = t) \\ = \int_{A^{(1)} \times \dots \times A^{(t)}} \frac{1}{t! P(T = t)} j_t (x^{(1)}, \dots, x^{(t)}) \rho(dx^{(1)}) \dots \rho(dx^{(t)}). \end{aligned} \quad (2.67)$$

Note that both Janossy measures and Janossy densities are permutation-symmetric w.r.t. its arguments.

2.3.3 FISST, PPP and multi-object statistical representations

FISST or PPP theory are useful to statistically describe multi-object scenarios where

1. The number of targets may be both time-varying and unknown, and hence it is appropriate to treat it as a random variable;
2. The order of the individual object states has no relevance, and this should be reflected in the statistical representation.

To illustrate the second point, consider a two-target tracking scenario, where the state of each target has the form $X^{(i)} = [P_x^{(i)}, P_y^{(i)}]^T$, $i \in (1, 2)$ where P_x and P_y correspond to the position in two-dimensional Cartesian coordinates.

2. MATHEMATICAL BACKGROUND

Assuming that the number of targets is known, we could, in principle, represent the two-target scenario by a four-dimensional vector $X_k = \begin{bmatrix} X_k^{(1)} \\ X_k^{(2)} \end{bmatrix} = \left[P_x^{(1)}, P_y^{(1)}, P_x^{(2)}, P_y^{(2)} \right]^T$, and statistically describe this scenario by a pdf $p(x)$.

Let us then consider two realizations of this random variable, say, $x_k(1) = [4, 3, -1, 5]^T$ and $x_k(2) = [-1, 5, 4, 3]^T$. Clearly, the events $\{X_k = x_k(1)\}$ and $\{X_k = x_k(2)\}$ correspond exactly the same event; namely, the event that there is one target at position $(-1, 5)$ and another at $(4, 3)$. Therefore, we can say that the order of the entries $X_k^{(1)}$ and $X_k^{(2)}$ in X_k is irrelevant.

A more complex problem is the Multi-Target Tracking and Labelling (MTTL) problem, when we want also to uniquely identify objects across different time steps. Let X_1 and X_2 be the multi-target state at two different times 1 and 2. For two realizations $x_1 = [4, 3, -1, 5]^T$ and $x_2 = [3, -4, 2, -1]^T$, we would perhaps like to indicate that the target at $(3, -4)$ in time 2 is the same target which was at $(4, 3)$ at time 1.

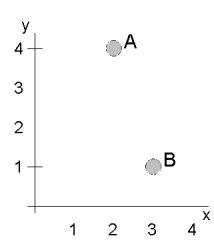
One possible solution is to treat the vector index as the object identifier, i.e. to assume that $x_k^{(i)}$, for a fixed i , corresponds always to the same object across multiple realizations and time steps. With this formulation, the order of the single-target vectors in the multi-target vector is thus relevant. This approach has been used in various works (e.g. Salmond et al. [1997], Crouse et al. [2011a], Blom and Bloem [2011], García-Fernández et al. [2011]), but cannot handle the situation where targets may appear and disappear.

A more general solution is to maintain the assumption that the targets are unordered, but add a identifier entry (label) to the single-target state vector. This approach has been mentioned in works like Ma et al. [2006], Vu et al. [2011], Zhu et al. [2011], and will be extensively discussed in Chapter 4.

The FISST and PPP multi-object statistical representations are equivalent, in the sense that both treat the multi-object states as unordered, and the number of targets as random. To see the relation between the two approaches, it can be shown (see Mori et al. [2009]) that for a RFS $\mathbf{X} = \{X^{(1)}, \dots, X^{(T)}\}$, alternatively

2. MATHEMATICAL BACKGROUND

Table 2.1: How the a labelled multi-target state may be mathematically represented

Physical state	Conventional notation	FISST	PPP
	$\begin{bmatrix} 2 \\ 4 \\ 3 \\ 1 \end{bmatrix}$, assuming $X_k^{(1)}$ to correspond to A, $X_k^{(2)}$ to B)	$\left\{ \begin{bmatrix} 2 \\ 4 \\ A \end{bmatrix}, \begin{bmatrix} 3 \\ 1 \\ B \end{bmatrix} \right\}$	$\left(\begin{bmatrix} 2 \\ 4 \\ A \end{bmatrix}, \begin{bmatrix} 3 \\ 1 \\ B \end{bmatrix} \right)$ or $\left(\begin{bmatrix} 3 \\ 1 \\ B \end{bmatrix}, \begin{bmatrix} 2 \\ 4 \\ A \end{bmatrix} \right)$

represented by the random variables $T, X^{(1)}, \dots, X^{(T)}$, we have

$$j_t(x^{(1)}, \dots, x^{(t)}) = f(\{x^{(1)}, \dots, x^{(t)}\}) \quad (2.68)$$

regardless, as expected, of the order of the arguments of $j_t(\cdot)$. Table 2.1 shows how a labelled multi-target state (i.e. a multi-target state where the objects are individually identified) may be represented using conventional statistics, FISST and PPP.

2.3.4 The Multi-target Sequential Monte Carlo filter

2.3.4.1 System model

The Multi-target Sequential Monte Carlo (M-SMC) filter, presented in Vo et al. [2005] and [Mahler, 2007, Chapter 15], is basically a multi-target version of the particle filter, based on FISST. It is derived by first assuming that the system described by $(\mathbf{X}_k, \mathbf{Z}_k)$ is a POM1 process. From Section 2.1.3, this corresponds to assuming that

$$f(\mathbf{z}_k | \mathbf{x}_0, \dots, \mathbf{x}_k, Z^{k-1}) = f(\mathbf{z}_k | \mathbf{x}_k), \quad (2.69)$$

$$f(\mathbf{x}_k | \mathbf{x}_0, \dots, \mathbf{x}_{k-1}, Z^{k-1}) = f(\mathbf{x}_k | \mathbf{x}_{k-1}). \quad (2.70)$$

2. MATHEMATICAL BACKGROUND

where $Z^k = (\mathbf{z}_1, \dots, \mathbf{z}_k)$. With these assumptions, the Bayesian recursion for the posterior RFS density $f(\mathbf{x}_k | Z^k)$ has the form

$$f(\mathbf{x}_k | Z^k) = \frac{f(\mathbf{z}_k | \mathbf{x}_k) f(\mathbf{x}_k | Z^{k-1})}{f(\mathbf{z}_k | Z^{k-1})} \quad (2.71)$$

where

$$f(\mathbf{x}_k | Z^{k-1}) = \int f(\mathbf{x}_k | \mathbf{x}_{k-1}) f(\mathbf{x}_{k-1} | Z^{k-1}) \delta \mathbf{x}_{k-1} \quad (2.72)$$

$$f(\mathbf{z}_k | Z^{k-1}) = \int f(\mathbf{z}_k | \mathbf{x}_k) f(\mathbf{x}_k | Z^{k-1}) \delta \mathbf{x}_k. \quad (2.73)$$

The functions $f(\mathbf{x}_0)$, $f(\mathbf{x}_k | \mathbf{x}_{k-1})$ and $f(\mathbf{z}_k | \mathbf{x}_k)$ correspond respectively to the multi-target prior, multi-target state transition and multi-target likelihood densities. Their exact formulas depend on the assumptions of the scenario. A list of multi-target filtering formulas for several multi-target scenarios is available in [Mahler, 2007, Chapters 12–14].

Assuming a POM1 system model, a multi-target version of the PF can be derived using relationships between RFS densities and conventional probability densities, as done by Vo et al. [2005]. The resulting algorithm is described in the next section.

2.3.4.2 SIR M-SMC filter algorithm

Initialization:

1. For each particle $i = 1, \dots, N_P$

- (a) Sample¹ $\mathbf{x}_0(i) \sim f(\mathbf{x}_0)$

- (b) Make $w_0(i) = \frac{1}{N_P}$

At every time step $k = 1, 2, \dots$:

1. For each particle $i = 1, \dots, N_P$

¹Sampling from a RFS density can be accomplished after expressing it in terms of conventional probability densities or mass functions, as shown in [Mahler, 2007, Chapters 12–14].

2. MATHEMATICAL BACKGROUND

- (a) Perform importance sampling by making

$$\mathbf{x}_k(i) \sim q(\mathbf{x}_k(i) | \mathbf{x}_{k-1}(i), \mathbf{z}_k)$$

where $q(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{z}_k)$ is a proposal RFS density

- (b) Calculate the unnormalized weight according to

$$\bar{w}_k(i) = \frac{f(\mathbf{z}_k | \mathbf{x}_k(i)) f(\mathbf{x}_k(i) | \mathbf{x}_{k-1}(i))}{q(\mathbf{x}_k(i) | \mathbf{x}_{k-1}(i), \mathbf{z}_k)} w_{k-1}(i)$$

2. Normalize the particle weights according to

$$w_k(i) = \frac{\bar{w}_k(i)}{\sum_{j=1}^{N_P} \bar{w}_k(j)}, \quad i = 1, \dots, N_P$$

3. Perform resampling by sampling N_P indexes $(\tilde{j}(i))_{i=1}^{N_P}$ according to the pmf $(w_k(j))_{j=1}^{N_P}$ and afterwards making

$$\begin{aligned} \mathbf{x}_k(i) &:= \mathbf{x}_k(\tilde{j}(i)) \\ w_k(i) &:= \frac{1}{N_P}, \quad i = 1, \dots, N_P \end{aligned}$$

2.4 Sensor management

The Bayesian sensor management problem, represented in Fig. 2.2, is basically a stochastic control extension of the Bayesian estimation problem that we described in Section 2.1.1. This means that besides the state and observation processes (denoted respectively by X_k and Z_k) we consider also a third random process U_k (with realizations denoted by u_k), corresponding to the control action, which in sensor management is also commonly referred to as the *sensing action*.

The sensing action is determined on the basis of statistics (ideally *sufficient* statistics) describing the states X_k , computed by the same estimating process that is used to compute the quantities of interest (mean, variance, etc.). The input to the estimator is the available information Z^k , composed by all observations z_1, \dots, z_k and sensing actions u_1, \dots, u_k known at time k . The control process that determines the sensing action is called the *sensor management process*.

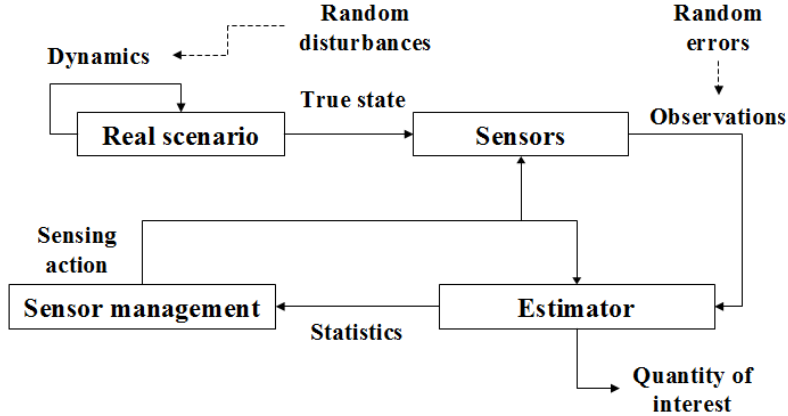


Figure 2.2: Sensor management as a stochastic control problem

As we can see from Figure 2.2, the difference between sensor management and the standard stochastic control problem is that, in sensor management, the feedback (i.e. the control decision U_k) targets the generation of observations Z_k (i.e. a *direct feedthrough* input), instead of targeting the state X_k . Typically, sensor management is performed in order to optimize the use of sensors, trying to achieve the best estimation performance with minimum sensing cost.

Some real-world examples of sensing actions include

1. Point the beam of a phased array radar to a particular direction (e.g. Burns and Blair [2004]);
2. Move a platform to which a sensor is attached (such as an Unmanned Aerial Vehicle (UAV)) (Skoglar et al. [2009]);
3. Activate or de-activate sensors in a sensor network (Boers et al. [2008]);
4. Changing the parameters of a radar transmitter or receiver (Katsilieris et al. [2012]).

2.4.1 Mathematical formulation

Let us suppose that the estimation quality is characterized by a reward (or risk) function $\gamma(X_k, Z_k, U_k)$, that we would like to maximize (or minimize). At a given

2. MATHEMATICAL BACKGROUND

time k , prior to the generation of the observation Z_k , let

$$Z^{k-1} = (z_1, \dots, z_{k-1}, u_1, \dots, u_{k-1})$$

denote all available information until and including time $k - 1$. Let us assume that the future sensing action U_j , $j \geq k$, has the form

$$U_j = \begin{cases} \eta_j(Z^{k-1}), & j = k \\ \eta_j(Z_k, \dots, Z_{j-1}, U_k, \dots, U_{j-1}, Z^{k-1}), & j > k \end{cases} \quad (2.74)$$

for some function η_j . We may then refer to the sequence of functions

$$(\eta_k, \dots, \eta_{k+h-1})$$

as a *control law* for time step k and time horizon h . Let us assume, without loss of generality, that γ is a reward function to be maximized. Then from [Bagchi, 1993, Chapter 2], the optimal *Closed-Loop* (CL) control law for time step k , with time horizon h and reward function γ is given by

$$\arg \max_{(\eta_k, \dots, \eta_{k+h-1})} E \left[\sum_{j=k}^{k+h-1} \gamma(X_j, Z_j, U_j) \middle| Z^{k-1} \right]. \quad (2.75)$$

Unfortunately, for general nonlinear, non-Gaussian systems, solving (2.75) is extremely difficult even with numerical approximations. Typically, suboptimal solutions (such as Kreucher and Hero [2005]; Kuwertz et al. [2010]) are used. The sensor management problem can be simplified by assuming that the control law does not depend on any information that is not yet available at time k , i.e. that the sensing action has the form $U_j = \eta_j(Z^{k-1})$ (see [Bagchi, 1993, Chapter 2]). Note that this assumption causes U_j to become deterministic, and hence it can be written as u_j . Also, we can perform the optimization directly over the space of sensing actions u_k, \dots, u_{k+h-1} , rather than over the space of functions $\eta_k, \dots, \eta_{k+h-1}$. The resulting solution corresponds to the optimal *Open-Loop* (OL)

2. MATHEMATICAL BACKGROUND

control law, given by

$$\arg \max_{(u_k, \dots, u_{k+h-1})} E \left[\sum_{j=k}^{k+h-1} \gamma(X_j, Z_j, u_j) \middle| Z^{k-1}, u_k, \dots, u_{k+h-1} \right]. \quad (2.76)$$

The optimal *Open-Loop Feedback* (OLF) solution is similar to the optimal OL solution, with the difference that at every subsequent time step $k+1, k+2, \dots$ we compute an entire new optimal OL solution, rather than applying sensing actions determined at previous time steps. This solution is used e.g. in Hanselmann et al. [2008]; Williams et al. [2007]. Optimal OL and OLF sensor management may, in theory, be performed by only using numerical approximations, although in practice, higher values of h will require additional approximations.

Finally, a simple but popular (Boers et al. [2008]; Kreucher et al. [2005]) approach to sensor management is to consider $h = 1$, i.e. to perform *short-term* sensor management. With this approach, the OL, CL and OLF solutions are all equivalent, and the optimal solution is given by

$$\arg \max_{u_k} E [\gamma(X_k, Z_k, u_k) | Z^{k-1}, u_k]. \quad (2.77)$$

2.4.2 Sensor management criteria

Besides deciding whether to perform CL, OL or OLF sensor management, it is necessary to choose the sensor management criterion, i.e. which reward/risk function γ should be maximized/minimized. There are two popular approaches to perform this choice: task-driven and information-driven sensor management.

2.4.2.1 Task-driven sensor management

A reasonable sensor management approach is to make γ correspond to a performance metric suitable for our problem of interest. For instance, we may have $\gamma(X_k, Z_k, U_k) = (X_k - \hat{x}_k(Z_k, U_k))^2$, i.e. the risk function corresponds to the squared error between the true state and the estimate (which depends on the observation Z_k and the sensing action U_k). This simple, performance-oriented approach to sensor management was referred to in Kreucher et al. [2005] as *task-driven* sensor management. While performing task-driven sensor management is

2. MATHEMATICAL BACKGROUND

intuitive and seemingly straightforward, it may be sometimes impractical.

As remarked by Kreucher et al. [2005], for complex problems (e.g. involving a large number of dimensions and/or state vector entries defined in different state spaces) it may be difficult to define a scalar metric that meaningfully measures the overall performance of our estimator. We may, of course, define a performance metric for such problem as a weighted sum of simpler performance metrics, but the selection of those weights may be highly subjective or prone to overfitting.

Task-driven sensor management may also be impractical when calculating the performance metric has a significant computational cost. In this case, even if we take the simplest sensor management approach (*short-term*), sensor management can be computationally expensive, since from (2.77), we also need to integrate the performance metric over both the state X_k and the observation Z_k , and we need to perform an optimization over the space of sensing actions u_k . Additionally, although we usually can evaluate a performance metric off-line, sensor management must be performed online.

2.4.2.2 Information-driven sensor management

One alternative to task-driven sensor management is to define a goal function $\gamma(Z_k, U_k)$ based on the posterior density $p(x_k | z_k, u_k, Z^{k-1})$, not on the estimate $\hat{x}_k(Z_k, U_k)$ or the true state X_k . In *information-driven sensor management* (as called by Kreucher et al. [2005]), one attempts to maximize the “information content” of the posterior, i.e. its capacity of yielding (in some sense) useful information to the user of the system, rather than attempting to directly optimize a performance metric.

There are basically two types of information-driven sensor management criteria:

1. When $\gamma(Z_k, U_k)$ represents the “information content” of the posterior density $p(x_k | Z^k) = p(x_k | z_k, u_k, Z^{k-1})$. Example are the Shannon entropy and the Rényi entropy criteria;
2. When $\gamma(Z_k, U_k)$ represents the “information gain” obtained by moving from the prior $p(x_k | Z^{k-1})$ to the posterior $p(x_k | z_k, u_k, Z^{k-1})$. Examples are the Kullback-Leibler (KL) divergence and Rényi divergence criteria.

2. MATHEMATICAL BACKGROUND

Information-driven sensor management are a popular alternative to task-driven sensor management, especially for complex problems where scalar performance measures are difficult to define or computationally expensive, such as the MTT problem. They have been used e.g. in Doucet et al. [2002]; Hanselmann et al. [2008]; Kreucher et al. [2005]; Williams et al. [2007].

However, whereas task-driven criteria are directly related to performance measures, having thus clear interpretation to the user, information-driven criteria (like the Shannon entropy and the KL divergence) represent more abstract concepts, raising the question of whether using this criteria can improve estimation quality in a way that conforms with the expectations of the user.

2.4.3 Sensor management applied to POM1 processes

POM1 processes, described in Section 2.1.3, can be easily adapted to include sensor management, leading to what we call Partially Observed Markov-1 processes with Direct Feedthrough (POM1DF), with the form

$$X_{k+1} = f_k(X_k, M_k) \tag{2.78}$$

$$Z_k = h_k(X_k, N_k, U_k) \tag{2.79}$$

$$U_k = \eta_k(Z_1 \dots, Z_{k-1}, U_1, \dots, U_{k-1}) \tag{2.80}$$

$$X_0 \sim p(x_0) \tag{2.81}$$

where $p(x_0)$ is the prior pdf, f_k , h_k and η_k are arbitrary functions, and $(M_k)_{k=0}^\infty$ and $(N_k)_{k=1}^\infty$ are all mutually independent random variables, also independent from $p(x_0)$. This model has the following properties:

$$p(z_k | x_0, \dots, x_k, Z^{k-1}, u_k) = p(z_k | x_k, u_k), \tag{2.82}$$

$$p(x_k | x_0, \dots, x_{k-1}, Z^{k-1}, u_k) = p(x_k | x_{k-1}), \tag{2.83}$$

$$p(x_0, \dots, x_{k-1} | Z^{k-1}, u_k) = p(x_0, \dots, x_{k-1} | Z^{k-1}). \tag{2.84}$$

Note than an important assumption of POM1DF processes is that the sensing action U_k does not affect the evolution of the state X_k . A particular sensor management problem that does not fit this assumption is the smart target track-

ing problem (considered e.g. in Kreucher et al. [2004]; Savage and Scala [2009]), where it is assumed that the target is aware of the sensing action and can respond to it (for instance, by attempting to evade being tracked).

2.4.4 Sensor management using particle filters

If a particle filter or similar method is used as estimator (see Fig. 2.2) in a sensor management problem, approximately optimal short-term sensor management, given by (2.77), can be implemented in a straightforward manner. We will show how this can be accomplished for POM1FD processes.

If sensor management is present, a SIS or SIR particle filter provides an approximation of $p(x_0, \dots, x_k | Z^k)$. The only difference is that Z^k includes not only the observations z_1, \dots, z_k but also the sensing actions u_1, \dots, u_k . For a POM1FD process, the optimal proposal density is given by

$$\begin{aligned} q(x_k | x_0, \dots, x_{k-1}, Z^k) &= q(x_k | x_{k-1}, z_k, u_k) \\ &= p(x_k | x_{k-1}, z_k, u_k). \end{aligned} \quad (2.85)$$

and for deterministic u_k (the case of short-term sensor management), the unnormalized weight update is given by

$$\bar{w}_k(i) = \frac{p(z_k | x_k(i), Z^{k-1}, u_k) p(x_k(i) | x_{k-1}(i))}{q(x_k(i) | x_{k-1}(i), z_k, u_k)} w_{k-1}(i).$$

Since all sensor management criteria implemented in this work belong to the information-driven category, we consider that reward function has the form $\gamma(Z_k, u_k)$. Therefore, the function to be maximized in (2.77), i.e. the expected reward, is given by

$$E[\gamma(Z_k, u_k) | Z^{k-1}, u_k] = \int \gamma(z_k, u_k) p(z_k | Z^{k-1}, u_k) dz_k. \quad (2.86)$$

2. MATHEMATICAL BACKGROUND

Observe now that the observation posterior is given by

$$\begin{aligned}
& p(z_k | Z^{k-1}, u_k) \\
&= \underbrace{\int \dots \int}_{k+1} p(z_k | x_0, \dots, x_k, Z^{k-1}, u_k) p(x_0, \dots, x_k | Z^{k-1}, u_k) dx_0 \dots dx_k \\
&= \underbrace{\int \dots \int}_{k+1} p(z_k | x_0, \dots, x_k, Z^{k-1}, u_k) p(x_k | x_0, \dots, x_{k-1}, Z^{k-1}, u_k) \\
&\quad \times p(x_0, \dots, x_{k-1} | Z^{k-1}, u_k) dx_0 \dots dx_k
\end{aligned} \tag{2.87}$$

and using POM1DF properties (2.82)–(2.84), (2.87) becomes

$$\begin{aligned}
& p(z_k | Z^{k-1}, u_k) \\
&= \int \int p(z_k | x_k, u_k) p(x_k | x_{k-1}) p(x_{k-1} | Z^{k-1}) dx_k dx_{k-1} \\
&\approx \sum_{i=1}^{N_P} w_{k-1}(i) \int p(z_k | x_k, u_k) p(x_k | x_{k-1}(i)) dx_k
\end{aligned} \tag{2.88}$$

where $\{x_{k-1}(i), w_{k-1}(i)\}_{i=1}^{N_P}$ is the set of particles obtained at the previous time step $k-1$.

We can then obtain a set of weighted samples $\{z_k(i), w_{k-1}(i)\}_{i=1}^{N_P}$ from the observation posterior $p(z_k | Z^{k-1}, u_k)$ by using sequential sampling; i.e. for each particle i , we sample $x_k(i) \sim p(x_k | x_{k-1}(i))$, and $z_k(i) \sim p(z_k | x_k(i), u_k)$. The expected reward (2.86) can then be approximated as

$$E[\gamma(Z_k, u_k) | Z^{k-1}] = \sum_{i=1}^{N_P} w_{k-1}(i) \gamma(z_k(i), u_k). \tag{2.89}$$

For some systems, it may be possible to provide better approximations of $E[\gamma(Z_k, u_k) | Z^{k-1}]$ by using more clever sampling schemes than the one described here; see Doucet et al. [2002] for details. Note also that the expected reward can be calculated in a similar manner for OL/OLF sensor management with an arbitrary time horizon h , by iteratively repeating the sequential sampling process. In practice, due to increase of dimensionality, the approximation of the expected

2. MATHEMATICAL BACKGROUND

reward is likely to be poor except for very small h . Another (and computationally expensive) approach is to increase the number of samples exponentially with h , as suggested by Kreucher and Hero [2005].

Chapter 3

Online Bayesian parameter estimation using the Rao-Blackwellized marginal particle filter

A well-known limitation of the standard particle filter is its inability to handle systems that include static variables (parameters) to be estimated together with the dynamic states. This limitation is due to the well-known degeneracy phenomenon, which is caused by the gradual loss of information that occurs during every iteration of the filter.

Given the practical interest in the problem of non-linear joint state and parameter estimation, various techniques have been proposed to adapt SMC methods to this problem. One of the best known techniques is the Liu and West Particle Filter (LWPF), which as most online Bayesian parameter estimation techniques, is based on the idea of adding artificial dynamics to the process model. While the LWPF has clear conceptual advantages over similar techniques, as we are going to see, it may still lead to biases in the approximation of the posterior probability distribution of states and parameters.

In this chapter, we provide two new methods to estimate the state and parameter jointly for a general state-space model, by first deriving new versions of

3. ONLINE BAYESIAN PARAMETER ESTIMATION USING THE RAO-BLACKWELLIZED MARGINAL PARTICLE FILTER

the recently proposed Rao-Blackwellized marginal particle filter (RBMPF). These novel algorithms combine a PF-like algorithm with a non-sequential Monte Carlo estimator to minimize the bias commonly produced by online Bayesian PF-based parameter estimation techniques. We illustrate the algorithms by applying them to two practical problems: the target tracking problem of estimating the turn rate of a constant turn maneuver, and the econometrics problem of stochastic volatility estimation from stock data (using the Heston model).

3.1 Introduction

As we mentioned in Section 2.1.2.4, the SIR degeneracy phenomenon makes the popular SIR particle filter an unsuitable solution to the problem of joint state and parameter estimation, where static parameters need to be estimated together with time-varying states. Despite this fact, the high interest in the non-linear parameter estimation problem has led to many developments on adapting SMC methods to this problem. As mentioned in Kantas et al. [2009], online Bayesian approaches typically use the idea of adding artificial dynamics to the system model. While this technique is theoretically able to prevent degeneracy from occurring, it has the collateral effect of leading to biases (generally difficult to quantify) in the particle approximation of the probability distribution of the states and parameters.

A popular and well-established method for online parameter estimation is the LWPF (Liu and West [2001]), used e.g. in Carvalho et al. [2010]; Liu and Niranjana [2012]; Nemeth et al. [2012]. The LWPF attempts to reduce the aforementioned bias in the posterior distribution by using a special model of parametrized artificial dynamics. As we are going to see in Section 3.2.2, however, the LWPF is also inherently sub-optimal in approximating the true posterior, and its performance may be sensitive to the parametrization of the artificial dynamics.

In this work, we propose two novel methods for online Bayesian joint state and parameter estimation. More specifically, we provide two new versions of the **Rao-Blackwellized marginal particle filter** (RBMPF), which has been recently proposed (Lindsten et al. [2012]) as a solution to the Bayesian joint state and parameter estimation problem. We emphasize here that the RBMPF

3. ONLINE BAYESIAN PARAMETER ESTIMATION USING THE RAO-BLACKWELLIZED MARGINAL PARTICLE FILTER

is not be confused with the far more well-known Rao-Blackwellized particle filter (RBPF) introduced in Section 2.1.4. The difference between the two techniques will be explained in Section 3.2.3.

Our novel algorithms, in comparison with the RBMPF presented in Lindsten et al. [2012], are more general, in the sense that they do not require the system model to contain any linear-Gaussian dependencies. In comparison with the LWPF, they use more elaborate (and non-parametric) approximations of the parameter conditional probabilities, which in principle, leads to a more accurate approximation of the posterior distribution (and therefore better characterization of estimation uncertainty), at the expense of additional computational cost.

This work is organized as follows. Section 3.2 discusses the theoretical advantages and issues of state-of-the-art online Bayesian PF-based methods, in the context of joint state and parameter estimation. Section 3.3 proposes the D-RBMPF, an implementation of the RBMPF for parameter estimation, directly applicable to discrete parameters and to discretized versions of continuous parameters with low dimension. Section 3.4 proposes the MC-RBMPF, an implementation of the RBMPF suitable for more general models. Section 3.5 presents simulation results for the D-RBMPF applied to the problem of turn rate estimation in target tracking. Section 3.6 presents simulation results for the MC-RBMPF applied to the problem of stochastic volatility estimation using the Heston model. Section 3.7 draws conclusions.

3.2 State-of-the-art SMC methods applied to joint state and parameter estimation

3.2.1 The SIR PF

In principle, any Bayesian recursion can be numerically approximated using a SIR PF, which means that we can apply the algorithm described in Section 2.1.2.2 to

3. ONLINE BAYESIAN PARAMETER ESTIMATION USING THE RAO-BLACKWELLIZED MARGINAL PARTICLE FILTER

the joint (Bayesian) state and parameter estimation model given by (2.39):

$$\begin{aligned} S_{k+1} &= f_k(S_k, \Theta, M_k) \\ Z_k &= h_k(S_k, \Theta, N_k) \\ \begin{bmatrix} S_0 \\ \Theta \end{bmatrix} &\sim p(s_0, \theta). \end{aligned} \tag{3.1}$$

The particle filter then approximates the joint posterior $p(s_k, \theta | Z^k)$ as

$$p(s_k, \theta | Z^k) \approx \sum_{i=1}^{N_P} w_k(i) \delta(s_k - s_k(i)) \delta(\theta - \theta_k(i)) \tag{3.2}$$

where N_P is the number of particles and $\{s_k(i), \theta_k(i), w_k(i)\}_{i=1}^{N_P}$ is the set of particles, with $s_k(i)$, $\theta_k(i)$ and $w_k(i)$ respectively being the state value, parameter value and weight corresponding to the i -th particle. The algorithm is as follows:

Initialization:

1. For each particle $i = 1, \dots, N_P$
 - (a) Sample $(s_0(i), \theta_0(i)) \sim p(s_0, \theta)$
 - (b) Make $w_0(i) = \frac{1}{N_P}$

At every time step $k = 1, 2, \dots$:

1. For each particle $i = 1, \dots, N_P$
 - (a) Perform importance sampling by making

$$s_k(i) \sim q(s_k | s_{k-1}(i), \theta_{k-1}(i), z_k) \tag{3.3}$$

$$\theta_k(i) = \theta_{k-1}(i) \tag{3.4}$$

where $q(s_k | s_{k-1}, \theta, z_k)$ is a proposal density

- (b) Calculate the unnormalized weight according to

$$\bar{w}_k(i) = \frac{p(z_k | s_k(i), \theta_{k-1}(i)) p(s_k(i) | s_{k-1}(i), \theta_k(i))}{q(s_k(i) | s_{k-1}(i), \theta_{k-1}(i), z_k)} \tag{3.5}$$

3. ONLINE BAYESIAN PARAMETER ESTIMATION USING THE RAO-BLACKWELLIZED MARGINAL PARTICLE FILTER

2. Normalize the particle weights according to

$$w_k(i) = \frac{\bar{w}_k(i)}{\sum_{j=1}^{N_P} \bar{w}_k(j)}, \quad i = 1, \dots, N_P \quad (3.6)$$

3. Perform resampling by sampling N_P indexes $(\tilde{j}(i))_{i=1}^{N_P}$ according to the pmf $(w_k(j))_{j=1}^{N_P}$ and afterwards making

$$\begin{aligned} (s_0(i), \dots, s_k(i), \theta_k(i)) &:= (s_0(\tilde{j}(i)), \dots, s_k(\tilde{j}(i)), \theta_k(\tilde{j}(i))) \\ w_k(i) &:= \frac{1}{N_P}, \quad i = 1, \dots, N_P \end{aligned} \quad (3.7)$$

In Section 2.1.2.4, we mentioned that due to the degeneracy phenomenon, the SIR PF is not an effective solution to the joint state and parameter estimation problem. For this problem, the presence of parameters causes the “indifference condition” (2.13) to be violated.

We can also see how the SIR PF fails at estimating static quantities by looking at approximation (3.2). Note that this approximation assumes that at time k , Θ takes one of the values $(\theta_k(i))_{i=1}^{N_P}$. Let then Λ_k be the support of Θ in the density approximation (3.2) (i.e. the set of unique values contained in $(\theta_k(i))_{i=1}^{N_P}$), and let Λ_{k+1} be defined analogously for $p(s_{k+1}, \theta | Z^{k+1})$. Then it is easily seen from the algorithm, in particular from (3.4) and (3.7), that

$$\Lambda_{k+1} \subseteq \Lambda_k \quad (3.8)$$

or, in other words, the diversity of realizations of Θ considered by the PF can only decrease with each time step. The speed of the decrease in cardinality of the sequence Λ_k , $k = 1, 2, \dots$, is directly related to the variance of the weights; the higher this variance, the less is the number of distinct particles that will survive during the resampling step (3.7) of the filter.

Therefore, if the considered time period (i.e. the maximum value of k) is not very large (which may be the case in offline parameter estimation), the number of particles is sufficiently large, and the variance of the weights is sufficiently small, the PF may still lead to good joint state and parameter estimation performance. However, for online parameter estimation, where typically one considers an un-

3. ONLINE BAYESIAN PARAMETER ESTIMATION USING THE RAO-BLACKWELLIZED MARGINAL PARTICLE FILTER

constrained time period, the cardinality of Λ_k will drop until it becomes 1, i.e. until the PF will assume that there is a single possible value for Θ , a situation that is referred to in literature as “degeneracy” or “self-resolving”. Needless to say, deterioration of performance of the estimates may occur long before the PF achieves the self-resolving situation.

The simplest approach to deal with degeneracy, as suggested in Gordon et al. [1993] and applied e.g. in Higuchi [2001], is to attribute “artificial dynamics” to the parameters, i.e. to assume that Θ is a time-varying state that evolves according to

$$\Theta_{k+1} = f^\theta(\Theta_k, M_k^\theta) \quad (3.9)$$

where f^θ is a function and M_k^θ is an artificial noise. This approach ensures maintenance of diversity in the considered values of Θ , but without any guidance on how to choose f^θ and M_k^θ , it may easily lead to biases in the approximation of the posterior distribution the joint state and parameter, as the mathematical model of the system is being deviated from its underlying physical model.

3.2.2 The Liu and West PF

The LWPF (Liu and West [2001]) is a more elaborate PF variant designed to prevent degeneracy in joint state and parameter estimation problem, based on the idea of artificial dynamics. Recall that (3.2), together with (3.4) and (3.7), causes degeneracy. The LWPF replaces approximation (3.2) with

$$p(s_k, \theta | Z^k) \approx \sum_{i=1}^{N_P} w_k(i) \delta(s_k - s_k(i)) \mathcal{N}(\theta; m_k(i, h), h^2 V_k) \quad (3.10)$$

3. ONLINE BAYESIAN PARAMETER ESTIMATION USING THE RAO-BLACKWELLIZED MARGINAL PARTICLE FILTER

where $h \in (0, 1]$ is a smoothing parameter and

$$m_k(i, h) = \sqrt{1 - h^2} \theta_k(i) + \left(1 - \sqrt{1 - h^2}\right) \hat{\theta}_k \quad (3.11)$$

$$V_k = \sum_{j=1}^{N_P} w_k(j) \left(\hat{\theta}_k - \theta_k(j)\right) \left(\hat{\theta}_k - \theta_k(j)\right)^T \quad (3.12)$$

$$\hat{\theta}_k = \sum_{j=1}^{N_P} w_k(j) \theta_k(j). \quad (3.13)$$

The original version presented in Liu and West [2001] is based on the Auxiliary Particle Filter proposed by Pitt and Shephard [1999]. In practice, approximation (3.10) can be easily adapted to be used with the SIR PF and nearly all variants of the PF algorithm. For the SIR PF described in Section 3.2.1, this can be done by replacing the step given by eq. (3.4) with

$$\theta_k(i) \sim \mathcal{N}(\theta; m_{k-1}(i, h), h^2 V_{k-1}) \quad (3.14)$$

i.e. $\mathcal{N}(\theta; m_{k-1}(i, h), h^2 V_{k-1})$ represent the “artificial dynamics” of the state Θ_k . In order to understand the strengths and limitations of the LWPF method, we need, naturally, to evaluate how reasonable it is to replace approximation (3.2) with (3.10). As shown in Liu and West [2001], the mean and the covariance of Θ (w.r.t. $p(\theta | Z^k)$) calculated by marginalizing (3.10) will be given respectively by (3.13) and (3.12), i.e. it will be exactly the same as if they were calculated by marginalizing (3.2) instead. Therefore, we can say that the LWPF provides a reasonable approximation of the marginal density $p(\theta | Z^k)$, in the sense that its first two moments are correctly calculated based on the particle approximation.

Note, however, that $p(s_k, \theta | Z^k) = p(s_k | Z^k) p(\theta | s_k, Z^k)$, i.e. it is not simply the product of the marginals $p(s_k | Z^k)$ and $p(\theta | Z^k)$. Since $p(s_k | Z^k)$ is being approximated according to

$$p(s_k | Z^k) \approx \sum_{i=1}^{N_P} w_k(i) \delta(s_k - s_k(i)) \quad (3.15)$$

the LWPF approximation (3.10) becomes reasonable if we additionally assume

3. ONLINE BAYESIAN PARAMETER ESTIMATION USING THE RAO-BLACKWELLIZED MARGINAL PARTICLE FILTER

that

$$p(\theta | s_k(i), Z^k) \approx \mathcal{N}(\theta; m_k(i, h), h^2 V_k). \quad (3.16)$$

We can see that approximation (3.16) has a number of limitations:

1. The covariance of θ conditioned on $s_k(i)$ and Z^k is assumed to be the same regardless of the particle i ;
2. It is based on only the approximation of the marginal density $p(\theta | Z^k)$ and the fact that $s_k(i)$ and $\theta(i)$ correspond to the same particle;
3. It is sensitive to the choice of the parameter h . A very low value of h is undesirable because it causes the algorithm to behave as a regular PF, being thus susceptible to degeneracy. But a large h has a collateral effect: it causes (3.16) to depend less on $\theta(i)$, hence diluting the association between same-particle values $s_k(i)$ and $\theta(i)$. Therefore, by having h too large, we will cause the existing information about the dependency between S_k and Θ to be neglected;

Therefore, in principle, the LWPF may still lead to significant biases on the approximation on the joint posterior density $p(s_k, \theta | Z^k)$, in particular, if an inappropriate value for the parameter h is chosen for the problem of interest.

3.2.3 The RBPF and the RBMPF

The two Rao-Blackwellized filters (RBPF and RBMPF) share a common principle, namely, using two *parallel interacting estimators*. In other words, part of the state X_k (say, S_k) is estimated using a SMC filter (a PF or one of its variants), and the another part (say, T_k) is estimated using a non-SMC method (like a Kalman filter-like or a grid-based estimator).

As we have seen in Section 2.1.4.1, the RBPF approximates the density $p(s_0, \dots, s_k, t_k | Z^k)$ as

$$p(s_0, \dots, s_k, t_k | Z^k) \approx \sum_{i=1}^{N_P} w_k(i) \left(\prod_{j=0}^k \delta(s_j - s_j(i)) \right) p(t_k | s_0(i), \dots, s_k(i), Z^k)$$

3. ONLINE BAYESIAN PARAMETER ESTIMATION USING THE RAO-BLACKWELLIZED MARGINAL PARTICLE FILTER

where $p(t_k | s_0(i), \dots, s_k(i), Z^k)$ is supposed to be approximated or calculated using an appropriate non-SMC method, and $\{s_0(i), \dots, s_k(i), w_k(i)\}_{i=1}^{N_P}$ is a particle approximation of $p(s_0, \dots, s_k | Z^k)$ (obtained using a SMC-like algorithm). Using marginalization, we can easily see that the filtering density may then be approximated as

$$p(s_k, t_k | Z^k) \approx \sum_{i=1}^{N_P} w_k(i) \delta(s_k - s_k(i)) p(t_k | s_0(i), \dots, s_k(i), Z^k). \quad (3.17)$$

The RBMPF, proposed by Lindsten et al. [2012] as a solution to the joint state and parameter estimation problem, can be considered as a combination of the RBPF with the MPF described in Section 2.1.5. Like the MPF, the RBMPF attempts to compute only the filtering density $p(s_k, t_k | Z^k)$, which is approximated as

$$p(s_k, t_k | Z^k) \approx \sum_{i=1}^{N_P} w_k(i) \delta(s_k - s_k(i)) p(t_k | s_k(i), Z^k) \quad (3.18)$$

where $p(t_k | s_k(i), Z^k)$, $i = 1, \dots, N_P$ is supposed to be approximated or calculated using a non-SMC method, and $\{s_k(i), w_k(i)\}_{i=1}^{N_P}$ is a particle approximation of $p(s_k | Z^k)$ obtained using a SMC-like algorithm.

Rao-Blackwellization seems an intuitive strategy to deal with the degeneracy problem in joint state and parameter estimation. By replacing T_k with the parameter Θ in (3.17), (3.18), we obtain an alternative approximation of the posterior density corresponding to (3.2). Furthermore, the posterior of Θ does not have a time-decreasing support (see Section 3.2.1), due to the non-SMC nature of its estimator.

However, as remarked by Lindsten et al. [2012], the RBPF is ineffective as a solution to parameter estimation. The reason is as follows. By replacing T_k with Θ in (3.17), we obtain

$$p(s_k, \theta | Z^k) \approx \sum_{i=1}^{N_P} w_k(i) \delta(s_k - s_k(i)) p(\theta | s_0(i), \dots, s_k(i), Z^k). \quad (3.19)$$

3. ONLINE BAYESIAN PARAMETER ESTIMATION USING THE RAO-BLACKWELLIZED MARGINAL PARTICLE FILTER

The problem of approximation (3.19) is that it explicitly depends on the particle representation of the past trajectory (S_0, \dots, S_{k-1}) . Assuming that this particle representation is obtained using a SIR PF, we know that these past states cannot be effectively estimated due to the degeneracy phenomenon described in Section 2.1.2.4. In fact, with the passage of time, the PF framework will cause states far in the past to be represented by a single particle, i.e. they will also suffer from degeneracy in the same way as parameters do. As a result, $p(\theta | s_0(i), \dots, s_k(i), Z^k)$ will have bias and subsequently, the joint posterior $p(s_k, \theta | Z^k)$ will be poorly approximated.

This problem is avoided by using a RBMPF instead of a RBPF. By replacing T_k with Θ in (3.18), we obtain the approximation

$$p(s_k, \theta | Z^k) \approx \sum_{i=1}^{N_P} w_k(i) \delta(s_k - s_k(i)) p(\theta | s_k(i), Z^k) \quad (3.20)$$

which depends only on the particle-approximated filtering density

$$p(s_k | Z^k) \approx \sum_{i=1}^{N_P} w_k(i) \delta(s_k - s_k(i)) \quad (3.21)$$

instead of the particle approximation of $p(s_0, \dots, s_k | Z^k)$. In order to devise a practical implementation of the RBMPF, we must be able to:

1. Approximate $p(s_k | Z^k)$ using a SMC method;
2. Calculate or approximate $p(\theta | s_k(i), Z^k)$ using a non-SMC method.

This task is definitely not trivial. Lindsten et al. [2012] proposed an approximation of $p(\theta | s_k(i), Z^k)$, which however could only be applied to structured models where Θ has a linear-Gaussian relationship with S_k .

3.3 The Discrete RBMPF (D-RBMPF)

Our first proposed version of the RBMPF assumes that the parameter Θ is discrete. If that is not the case, we can naturally perform discretization of the parameter space, although for computational reasons, this will limit the usefulness

3. ONLINE BAYESIAN PARAMETER ESTIMATION USING THE RAO-BLACKWELLIZED MARGINAL PARTICLE FILTER

of the method to low dimensionality parameters. We call this method Discrete RBMPF (D-RBMPF).

Let us assume that Θ assumes values in a set of discrete points Π . With $p(s_k | Z^k)$ approximated by the set of particles $\{s_k(i), w_k(i)\}_{i=1}^{N_P}$, an expectation of the form $E[g(S_k, \Theta) | Z^k]$, for any function g of S_k and Θ , would be given by

$$E[g(S_k, \Theta) | Z^k] \approx \sum_{i=1}^{N_P} w_k(i) \sum_{\theta \in \Pi} p(\theta | s_k(i), Z^k) g(s_k(i), \theta) \quad (3.22)$$

and the output of the filter at each time step k consists of

$$\left\{ s_k(i), w_k(i), \{p(\theta | s_k(i), Z^k)\}_{\theta \in \Pi} \right\}_{i=1}^{N_P}. \quad (3.23)$$

As we see from (3.23), each iteration of the filter must have two steps: one to obtain $\{s_k(i), w_k(i)\}_{i=1}^{N_P}$, and another one to obtain $\left\{ \{p(\theta | s_k(i), Z^k)\}_{\theta \in \Pi} \right\}_{i=1}^{N_P}$. We will now derive each of these two steps.

3.3.1 Obtaining the particle states/weights (MPF step)

In order to obtain $\{s_k(i), w_k(i)\}_{i=1}^{N_P}$, we will derive a modified version of the MPF (just as the RBPF uses a modified PF). Observe first that

$$\begin{aligned} p(s_k | Z^k) &= \int \int p(s_k, s_{k-1}, \theta | Z^k) ds_{k-1} d\theta \\ &= \int \int \frac{p(z_k | s_k, s_{k-1}, \theta, Z^{k-1})}{p(z_k | Z^{k-1})} p(s_k, s_{k-1}, \theta | Z^{k-1}) ds_{k-1} d\theta \\ &= \int \int \frac{p(z_k | s_k, s_{k-1}, \theta, Z^{k-1}) p(s_k | s_{k-1}, \theta, Z^{k-1})}{p(z_k | Z^{k-1})} p(s_{k-1}, \theta | Z^{k-1}) ds_{k-1} d\theta. \end{aligned} \quad (3.24)$$

3. ONLINE BAYESIAN PARAMETER ESTIMATION USING THE RAO-BLACKWELLIZED MARGINAL PARTICLE FILTER

From (3.1), we have

$$p(z_k | s_k, s_{k-1}, \theta, Z^{k-1}) = p(z_k | s_k, \theta) \quad (3.25)$$

$$p(s_k | s_{k-1}, \theta, Z^{k-1}) = p(s_k | s_{k-1}, \theta) \quad (3.26)$$

and therefore from (3.24)

$$p(s_k | Z^k) = \int \int \frac{p(z_k | s_k, \theta) p(s_k | s_{k-1}, \theta)}{p(z_k | Z^{k-1})} p(s_{k-1}, \theta | Z^{k-1}) ds_{k-1} d\theta. \quad (3.27)$$

By defining

$$K(s_k, s_{k-1}, \theta, z_k) \triangleq p(z_k | s_k, \theta) p(s_k | s_{k-1}, \theta) \quad (3.28)$$

we have

$$\begin{aligned} p(s_k | Z^k) &= \int \int \frac{K(s_k, s_{k-1}, \theta, z_k) p(s_{k-1}, \theta | Z^{k-1})}{p(z_k | Z^{k-1})} ds_{k-1} d\theta \\ &= \frac{E[K(s_k, S_{k-1}, \Theta, z_k) | Z^{k-1}]}{p(z_k | Z^{k-1})}. \end{aligned} \quad (3.29)$$

Now, observe that a conditional expectation of the form $E[g(S_k) | Z^k]$ (where g is an arbitrary function) is given by

$$\begin{aligned} E[g(S_k) | Z^k] &= \int g(s_k) p(s_k | Z^k) ds_k \\ &= \int g(s_k) \frac{E[K(s_k, S_{k-1}, \Theta, z_k) | Z^{k-1}]}{p(z_k | Z^{k-1})} ds_k \\ &= \int g(s_k) \frac{E[K(s_k, S_{k-1}, \Theta, z_k) | Z^{k-1}]}{p(z_k | Z^{k-1}) q(s_k | Z^k)} q(s_k | Z^k) ds_k \end{aligned} \quad (3.30)$$

where $q(s_k | Z^k)$ is an appropriate proposal density for S_k . If we generate N_P i.i.d. particle states $s_k(i)$ by sampling from $q(s_k | Z^k)$, $E[g(S_k) | Z^k]$ may be

3. ONLINE BAYESIAN PARAMETER ESTIMATION USING THE RAO-BLACKWELLIZED MARGINAL PARTICLE FILTER

approximated by

$$E [g(S_k) | Z^k] \approx \sum_{i=1}^{N_P} g(s_k(i)) w_k(i) \quad (3.31)$$

where the particle weights are given by

$$w_k(i) = \frac{E [K(s_k(i), S_{k-1}, \Theta, z_k) | Z^{k-1}]}{N_{PP}(z_k | Z^{k-1}) q(s_k(i) | Z^k)}. \quad (3.32)$$

To calculate (3.32), we approximate the numerator of (3.32) using (3.22), i.e. we use the output (3.23) produced at the previous iteration $k - 1$:

$$\begin{aligned} & E [K(s_k(i), S_{k-1}, \Theta, z_k) | Z^{k-1}] \\ & \approx \sum_{j=1}^{N_P} w_{k-1}(j) \sum_{\theta \in \Pi} p(\theta | s_{k-1}(j), Z^{k-1}) K(s_k(i), s_{k-1}(j), \theta, z_k). \end{aligned} \quad (3.33)$$

Observe that the term $N_{PP}(z_k | Z^{k-1})$ in (3.32) becomes irrelevant as it does not depend on the particle state $s_k(i)$, and hence can be taken in account by normalizing the weights. Note also that unlike the SIR particle filter, the MPF does not contain a resampling step. A proposal density that is particularly easy to implement is the “blind proposal density”, i.e. to perform importance sampling disregarding the last observation z_k . This can be accomplished by making

$$\begin{aligned} q(s_k | Z^k) &= p(s_k | Z^{k-1}) \\ &= E [p(s_k | S_{k-1}, \Theta) | Z^{k-1}] \end{aligned} \quad (3.34)$$

which, from (3.22), can be approximated as

$$q(s_k | Z^k) \approx \sum_{j=1}^{N_P} w_{k-1}(j) \sum_{\theta \in \Pi} p(\theta | s_{k-1}(j), Z^{k-1}) p(s_k | s_{k-1}(j), \theta). \quad (3.35)$$

For a given particle i , in order to sample from (3.35), one can thus take the following steps:

1. Obtain a sample $s_{k-1}(i)$ from the pmf $(w_{k-1}(j))_{j=1}^{N_P}$. Using a resampling

3. ONLINE BAYESIAN PARAMETER ESTIMATION USING THE RAO-BLACKWELLIZED MARGINAL PARTICLE FILTER

algorithm, such as the systematic resampling algorithm described in Section 2.1.2.3, it is possible to obtain N_P samples, one for each $i = 1, \dots, N_P$;

2. Obtain a sample $\theta_k(i)$ from the pmf $(p(\theta | s_{k-1}(i), Z^{k-1}))_{\theta \in \Pi}$;
3. Finally, obtain a sample $s_k(i)$ from the pdf $p(s_k | s_{k-1}(i), \theta(i))$.

For alternative importance sampling strategies applicable to a MPF, refer to Klaas et al. [2005]. It is maybe possible to replace the MPF with a regular SIR PF in this step, although the consequences of this replacement in the mathematical validity of the algorithm require further investigation.

3.3.2 Obtaining the parameter conditional probabilities (discrete step)

Now we describe how to obtain $\left\{ \left\{ p(\theta | s_k(i), Z^k) \right\}_{\theta \in \Pi} \right\}_{i=1}^{N_P}$ given the particle cloud $\{s_k(i), w_k(i)\}_{i=1}^{N_P}$. First, observe that, for a given particle i

$$\begin{aligned}
 p(\theta | s_k(i), Z^k) &= \frac{p(\theta, s_k(i) | Z^k)}{p(s_k(i) | Z^k)} \\
 &= \frac{\int p(\theta, s_k(i), s_{k-1} | Z^k) ds_{k-1}}{p(s_k(i) | Z^k)} \\
 &= \int \frac{p(z_k | s_k(i), \theta) p(s_k(i) | s_{k-1}, \theta)}{p(z_k | Z^{k-1}) p(s_k(i) | Z^k)} p(s_{k-1}, \theta | Z^{k-1}) ds_{k-1} \\
 &= \int \frac{K(s_k(i), s_{k-1}, \theta, z_k) p(\theta | s_{k-1}, Z^{k-1})}{p(z_k | Z^{k-1}) p(s_k(i) | Z^k)} p(s_{k-1} | Z^{k-1}) ds_{k-1}
 \end{aligned} \tag{3.36}$$

where $K(s_k, s_{k-1}, \theta, z_k)$ is defined by (3.28). By using the algorithm output obtained at the previous iteration $k - 1$, we can approximate (3.36) by

$$p(\theta | s_k(i), Z^k) \approx \sum_{j=1}^{N_P} \frac{w_{k-1}(j) K(s_k(i), s_{k-1}(j), \theta, z_k)}{p(z_k | Z^{k-1}) p(s_k(i) | Z^k)} p(\theta | s_{k-1}(j), Z^{k-1}) \tag{3.37}$$

where we note that the term $p(z_k | Z^{k-1}) p(s_k(i) | Z^k)$ is irrelevant as it does not depend on θ , and hence can be taken into account by normalizing the parameter

3. ONLINE BAYESIAN PARAMETER ESTIMATION USING THE RAO-BLACKWELLIZED MARGINAL PARTICLE FILTER

conditional probabilities.

3.3.3 D-RBMPF algorithm

Initialization:

1. For each particle $i = 1, \dots, N_P$
 - (a) Sample $s_0(i) \sim p(s_0)$
 - (b) Make $w_0(i) = \frac{1}{N_P}$
 - (c) For each $\theta \in \Pi$, set $p(\theta | s_0(i))$

At every time step $k = 1, 2, \dots$:

1. For each particle $i = 1, \dots, N_P$
 - (a) Perform particle importance sampling by drawing

$$s_k(i) \sim q\left(s_k \mid Z^k\right) \quad (3.38)$$

where $q(s_k | Z^k)$ is a proposal density

- (b) Calculate the unnormalized particle weight according to

$$\bar{w}_k(i) = \frac{E\left[K(s_k(i), S_{k-1}, \Theta, z_k) \mid Z^{k-1}]\right]}{q(s_k(i) | Z^k)} \quad (3.39)$$

where the expectation in the numerator is calculated using (3.33)

- (c) For each $\theta \in \Pi$, compute the unnormalized parameter conditional probability according to

$$\bar{p}\left(\theta \mid s_k(i), Z^k\right) = \sum_{j=1}^{N_P} w_{k-1}(j) K(s_k(i), s_{k-1}(j), \theta, z_k) p\left(\theta \mid s_{k-1}(j), Z^{k-1}\right) \quad (3.40)$$

- (d) Normalize the conditional parameter probabilities according to

$$p\left(\theta \mid s_k(i), Z^k\right) = \frac{\bar{p}\left(\theta \mid s_k(i), Z^k\right)}{\sum_{\theta^* \in \Pi} \bar{p}\left(\theta^* \mid s_k(i), Z^k\right)} \quad (3.41)$$

3. ONLINE BAYESIAN PARAMETER ESTIMATION USING THE RAO-BLACKWELLIZED MARGINAL PARTICLE FILTER

2. Normalize the particle weights according to

$$w_k(i) = \frac{\bar{w}_k(i)}{\sum_{j=1}^{N_P} \bar{w}_k(j)}, \quad i = 1, \dots, N_P \quad (3.42)$$

3.3.4 A special case

If $p(z_k | s_k, \theta) = p(z_k | s_k)$, the algorithm described in Section 3.3.3 can be simplified. In this case, (3.39) becomes

$$\bar{w}_k(i) = \frac{p(z_k | s_k(i)) E [p(s_k | S_{k-1}, \Theta) | Z^{k-1}]}{q(s_k(i) | Z^k)} \quad (3.43)$$

and hence, if we use blind importance sampling (3.34) for the particles, we have

$$\bar{w}_k(i) = p(z_k | s_k(i)) \quad (3.44)$$

i.e. the weights are simply proportional to the likelihoods (as in a SIR PF using blind importance sampling). An example of this special case is the problem of target tracking with target type identification (where the parameter Θ is the target type) using sensors that measure only the target's position and other kinematic properties.

3.3.5 Computational complexity

The blind importance sampling scheme (3.34) has a computational complexity of $O(N_P |\Pi|)$. The type of resampling used in blind importance sampling does not affect complexity, as all well-known resampling methods have a computational complexity of $O(N_P)$ (see Hol et al. [2006]). The calculation of weights and the calculation of parameter conditional probabilities have both a complexity of $O(N_P^2 |\Pi|)$. Hence the overall complexity of the D-RBMPF is $O(N_P^2 |\Pi|)$.

In the special case where $p(z_k | s_k, \theta) = p(z_k | s_k)$, the complexity of the calculation of weights goes down to $O(N_P)$. However, since the complexity of the calculation of parameter conditionals is not changed, the overall complexity of the D-RBMPF is still $O(N_P^2 |\Pi|)$.

The main loop of the algorithm (steps (3.38)–(3.41)) may be parallelized, by

3. ONLINE BAYESIAN PARAMETER ESTIMATION USING THE RAO-BLACKWELLIZED MARGINAL PARTICLE FILTER

dividing the particles among the processing nodes. With each processing node processing at most one particle, the maximum complexity per node would be $O(N_P|\Pi|)$.

3.4 The Monte Carlo RBMPF (MC-RBMPF)

Our second proposed algorithm based on the RBMPF does not depend on a discretization of the parameter space, and therefore, can be useful for problems with high dimensional parameters. The algorithm consists of using a MPF to estimate $p(s_k | Z^k)$, and a (non-Sequential) Monte Carlo (MC) estimator for $p(\theta | s_k(i), Z^k)$. We call this method Monte Carlo RBMPF (MC-RBMPF). Though we use, as in the LWPF, a Gaussian density to approximate the parameter conditional $p(\theta | s_k(i), Z^k)$, we shall see later that the proposed method is intrinsically different from the LWPF.

In the MC-RBMPF, we assume that expectations taken over $p(\theta | s_k(i), Z^k)$ have no closed form expressions. In particular, we use Monte Carlo sampling, i.e. we generate N_S weighted samples $\{\theta_k(i, j), v_k(i, j)\}_{j=1}^{N_S}$ from $p(\theta | s_k(i), Z^k)$, where $v_k(i, j)$ is the sample weight and $\theta_k(i, j)$ is the sample value. An expectation of the form $E[g(S_k, \Theta) | Z^k]$, for any function g of S_k and Θ , may then be approximated as

$$E[g(S_k, \Theta) | Z^k] \approx \sum_{i=1}^{N_P} w_k(i) \sum_{m=1}^{N_S} v_k(i, m) g(s_k(i), \theta_k(i, m)) \quad (3.45)$$

and the output of the filter at each time step k consists of

$$\left\{ s_k(i), w_k(i), \{\theta_k(i, m), v_k(i, m)\}_{m=1}^{N_S} \right\}_{i=1}^{N_P}. \quad (3.46)$$

We will henceforth refer to $\{\theta_k(i, m), v_k(i, m)\}_{m=1}^{N_S}$ as the “sub-particles” of each particle $(s_k(i), w_k(i))$. This is a convenient abuse of notation; it should be clear to the reader that they are not akin to particles produced by a particle filter. This is because they are generated by a Monte Carlo-based and not by a Sequential Monte Carlo method, as in our scheme, the samples $\{\theta_k(i, m), v_k(i, m)\}_{m=1}^{N_S}$

3. ONLINE BAYESIAN PARAMETER ESTIMATION USING THE RAO-BLACKWELLIZED MARGINAL PARTICLE FILTER

are *not* propagated to the next iteration of the algorithm.

As we see from (3.46), each iteration of the filter must have two steps: one to obtain $\{s_k(i), w_k(i)\}_{i=1}^{N_P}$, and another one to obtain $\left\{ \left\{ \theta_k(i, m), v_k(i, m) \right\}_{m=1}^{N_S} \right\}_{i=1}^{N_P}$. We will now derive each of these two steps.

3.4.1 Obtaining the particle states/weights (MPF step)

The derivation of the MPF step of the MC-RBMPF is similar to the derivation of the MPF step of the D-RBMPF. Let us look at (3.32), the expression that we obtained for the particle weight:

$$w_k(i) = \frac{E \left[K(s_k(i), S_{k-1}, \Theta, z_k) \mid Z^{k-1} \right]}{N_P p(z_k \mid Z^{k-1}) q(s_k(i) \mid Z^k)}. \quad (3.47)$$

In the MC-RBMPF, the numerator is approximated using (3.45). By using the output (3.46) produced at the previous iteration $k - 1$, we have

$$\begin{aligned} & E \left[K(s_k(i), S_{k-1}, \Theta, z_k) \mid Z^{k-1} \right] \\ & \approx \sum_{j=1}^{N_P} w_{k-1}(j) \sum_{m=1}^{N_S} v_{k-1}(j, m) K(s_k(i), s_{k-1}(j), \theta_{k-1}(j, m), z_k). \end{aligned} \quad (3.48)$$

The blind proposal density (3.34), using (3.45), can be approximated as

$$q(s_k \mid Z^k) \approx \sum_{j=1}^{N_P} w_{k-1}(j) \sum_{m=1}^{N_S} v_{k-1}(j, m) p(s_k \mid s_{k-1}(j), \theta_{k-1}(j, m)). \quad (3.49)$$

For a given particle i , in order to sample from (3.49), one can thus take the following steps:

1. Obtain a sample $s_{k-1}(i)$ from the pmf $(w_{k-1}(j))_{j=1}^{N_P}$. As for the D-RBMPF, a resampling algorithm can be used in this step;
2. Obtain a sample $\theta(i)$ from the pmf $(v_{k-1}(i, m))_{m=1}^{N_S}$;
3. Finally, obtain a sample $s_k(i)$ from the pdf $p(s_k \mid s_{k-1}(i), \theta(i))$.

3. ONLINE BAYESIAN PARAMETER ESTIMATION USING THE RAO-BLACKWELLIZED MARGINAL PARTICLE FILTER

3.4.2 Obtaining the sub-particle states/weights (MC step)

Now we describe how to obtain $\left\{ \{\theta_k(i, m), v_k(i, m)\}_{m=1}^{N_S} \right\}_{i=1}^{N_P}$ given the particle cloud $\{s_k(i), w_k(i)\}_{i=1}^{N_P}$. Recall that

$$\{\theta_k(i, m), v_k(i, m)\}_{m=1}^{N_S}, i = 1, \dots, N_P$$

is assumed to be a set of samples from $p(\theta | s_k(i), Z^k)$. In order to obtain these samples, let us consider again approximation (3.37) of $p(\theta | s_k(i), Z^k)$, that we obtained when deriving the D-RBMPF:

$$p(\theta | s_k(i), Z^k) \approx \sum_{j=1}^{N_P} \frac{w_{k-1}(j) K(s_k(i), s_{k-1}(j), \theta, z_k)}{p(z_k | Z^{k-1}) p(s_k(i) | Z^k)} p(\theta | s_{k-1}(j), Z^{k-1}). \quad (3.50)$$

How to approximate the term $p(\theta | s_{k-1}(j), Z^{k-1})$? We know that each set of sub-particles produced at the previous iteration, i.e.

$$\{\theta_{k-1}(j, m), v_{k-1}(j, m)\}_{m=1}^{N_S}, j = 1, \dots, N_P,$$

consists of a set of samples of $p(\theta | s_{k-1}(j), Z^{k-1})$. A “naive” approximation of this term would then be the *Sequential* Monte Carlo approximation, i.e.

$$\sum_{m=1}^{N_S} v_{k-1}(j, m) \delta(\theta - \theta_{k-1}(j, m))$$

which we can promptly recognize as a bad choice. It would cause the support of the approximated $p(\theta | Z^{k-1})$ to be restricted to the sub-particle values $\left\{ \{\theta_{k-1}(j, m)\}_{m=1}^{N_S} \right\}_{j=1}^{N_P}$. The support of the approximated $p(\theta | Z^k)$ would hence be also restricted to the same set of sub-particle values, leading to the same degeneracy phenomenon described in Section 3.2.1.

We propose instead to fit an empirical, “easy-to-sample” probability density to the set of points given by $\{\theta_{k-1}(j, m), v_{k-1}(j, m)\}_{m=1}^{N_S}$. A practical (but not the

3. ONLINE BAYESIAN PARAMETER ESTIMATION USING THE RAO-BLACKWELLIZED MARGINAL PARTICLE FILTER

only) choice is a Gaussian probability density

$$\mathcal{N}\left(\theta; \hat{\theta}_{k-1}(j), P_{k-1}(j)\right),$$

with first and second moments matching those of the sub-particle approximation, i.e. with mean and covariance given by

$$\hat{\theta}_{k-1}(j) = \sum_{m=1}^{N_S} v_{k-1}(j, m) \theta_{k-1}(j, m), \quad (3.51)$$

$$P_{k-1}(j) = \sum_{m=1}^{N_S} v_{k-1}(j, m) \left(\theta_{k-1}(j, m) - \hat{\theta}_{k-1}(j) \right) \left(\theta_{k-1}(j, m) - \hat{\theta}_{k-1}(j) \right)^T. \quad (3.52)$$

Using the aforementioned empirical density, (3.50) becomes

$$p(\theta | s_k(i), Z^k) \approx \sum_{j=1}^{N_P} \frac{w_{k-1}(j) K(s_k(i), s_{k-1}(j), \theta, z_k) \mathcal{N}\left(\theta; \hat{\theta}_{k-1}(j), P_{k-1}(j)\right)}{p(z_k | Z^{k-1}) p(s_k(i) | Z^k)}. \quad (3.53)$$

We now need, for each particle i , to obtain a set of weighted samples (“sub-particles”) $\{\theta_k(i, m), v_k(i, m)\}_{m=1}^{N_S}$ from $p(\theta | s_k(i), Z^k)$ approximated as (3.53). This is accomplished, for instance, by importance sampling. First, we sample $(\theta_k(i, m))_{m=1}^{N_S}$ from some appropriate proposal density $q(\theta | s_k(i), Z^k)$. We then calculate the unnormalized sub-particle weights using the proportionality

$$v_k(i, m) \propto \sum_{j=1}^{N_P} \frac{w_{k-1}(j) K(s_k(i), s_{k-1}(j), \theta_k(i, m), z_k)}{q(\theta_k(i, m) | s_k(i), Z^k)} \mathcal{N}\left(\theta_k(i, m); \hat{\theta}_{k-1}(j), P_{k-1}(j)\right) \quad (3.54)$$

and then normalize them. One proposal density that is easy to implement is given by

$$q(\theta | s_k(i), Z^k) = \sum_{j=1}^{N_P} w_{k-1}(j) \mathcal{N}\left(\theta; \hat{\theta}_{k-1}(j), P_{k-1}(j)\right) \quad (3.55)$$

3. ONLINE BAYESIAN PARAMETER ESTIMATION USING THE RAO-BLACKWELLIZED MARGINAL PARTICLE FILTER

which does not consider, however, neither the particle value $s_k(i)$ nor the last observation z_k . To sample from this proposal density, we may:

1. Obtain a sample $\left(\hat{\theta}_{k-1}(i), P_{k-1}(i)\right)$ from the pmf $(w_{k-1}(j))_{j=1}^{N_P}$. For all N_P particles and $N_S \times N_P$ sub-particles, we may accomplish step this by performing N_S resampling steps. Here it is better to avoid the systematic resampling algorithm, by using, for instance, the multinomial resampling algorithm (see Hol et al. [2006]). The reason is that systematic resampling will result in similar choices of particles for each $m = 1, \dots, N_S$;
2. Sample $\theta(i, m)$ from $\mathcal{N}\left(\theta; \hat{\theta}_{k-1}(i), P_{k-1}(i)\right)$.

Similar to what happens in the importance sampling step of a particle filter, if the proposal $q(\theta | s_k(i), Z^k)$ is far more heavy-tailed than the target density $p(\theta | s_k(i), Z^k)$, importance sampling may result in sub-particle weights with large variance. In this situation, many sub-particles will have weights close to zero, such that they will be effectively “wasted”. A solution to this problem is to use proposal density closer to $p(\theta | s_k(i), Z^k)$. If this is not possible, an alternative solution, also considered in PF literature (see Doucet et al. [2001]; Gilks and Berzuini [2001]), is to employ MCMC methods (such as the Metropolis-Hastings algorithm) as a replacement or in addition to importance sampling.

The reader may have noticed that in the RBMPF, obtaining samples of Θ from (3.53) (which approximates the conditional density $p(\theta | s_k(i), Z^k)$) is similar to obtaining samples as in (3.14) when one uses a LWPF. Although in the RBMPF, we do not explicitly add artificial dynamics to the system model, the method relies on Gaussian probability densities, and therefore, (3.53) may be considered some sort of “artificial dynamics” for Θ . However, in comparison with the LWPF, the RBMPF approximation of $p(\theta | s_k(i), Z^k)$ is non-parametrized and numerical, i.e. based on multiple samples of Θ associated with each particle i , and the covariance of Θ given $s_k(i)$ and Z_k is allowed to be different for each particle.

3.4.3 MC-RBMPF algorithm

Initialization:

1. For each particle $i = 1, \dots, N_P$

3. ONLINE BAYESIAN PARAMETER ESTIMATION USING THE RAO-BLACKWELLIZED MARGINAL PARTICLE FILTER

- (a) Sample $s_0(i) \sim p(s_0)$
- (b) Make $w_0(i) = \frac{1}{N_P}$
- (c) For each sub-particle $m = 1, \dots, N_S$
 - i. Sample $\theta_0(i, m) \sim p(\theta | s_0(i))$
 - ii. Make $v_0(i, m) = \frac{1}{N_S}$

At every time step $k = 1, 2, \dots$:

1. For each particle $i = 1, \dots, N_P$

- (a) Perform particle importance sampling by drawing

$$s_k(i) \sim q\left(s_k \mid Z^k\right) \quad (3.56)$$

where $q(s_k \mid Z^k)$ is a proposal density

- (b) Calculate the unnormalized particle weight according to

$$\bar{w}_k(i) = \frac{E\left[K(s_k(i), S_{k-1}, \Theta, z_k) \mid Z^{k-1}\right]}{q(s_k(i) \mid Z^k)} \quad (3.57)$$

where the expectation in the numerator is calculated using (3.48)

- (c) Calculate

$$\hat{\theta}_{k-1}(i) = \sum_{m=1}^{N_S} v_{k-1}(i, m) \theta_{k-1}(i, m) \quad (3.58)$$

$$P_{k-1}(i) = \sum_{m=1}^{N_S} v_{k-1}(i, m) \left(\theta_{k-1}(i, m) - \hat{\theta}_{k-1}(i)\right) \left(\theta_{k-1}(i, m) - \hat{\theta}_{k-1}(i)\right)^T \quad (3.59)$$

2. Normalize the particle weights according to

$$w_k(i) = \frac{\bar{w}_k(i)}{\sum_{j=1}^{N_P} \bar{w}_k(j)}, \quad i = 1, \dots, N_P \quad (3.60)$$

3. For each particle $i = 1, \dots, N_P$

- (a) For each sub-particle $m = 1, \dots, N_S$

3. ONLINE BAYESIAN PARAMETER ESTIMATION USING THE RAO-BLACKWELLIZED MARGINAL PARTICLE FILTER

- i. Perform sub-particle importance sampling by drawing

$$\theta_k(i, m) \sim q\left(\theta \mid s_k(i), Z^k\right) \quad (3.61)$$

where $q(\theta \mid s_k, Z^k)$ is a proposal density

- ii. Calculate the unnormalized sub-particle weight according to

$$\begin{aligned} \bar{v}_k(i, m) &= \sum_{j=1}^{N_P} \frac{w_{k-1}(j)}{q(\theta_k(i, m) \mid s_k(i), Z^k)} K(s_k(i), s_{k-1}(j), \theta_k(i, m), z_k) \\ &\quad \times \mathcal{N}\left(\theta_k(i, m); \hat{\theta}_{k-1}(j), P_{k-1}(j)\right) \end{aligned} \quad (3.62)$$

- (b) Normalize the subparticle weights according to

$$v_k(i, m) = \frac{\bar{v}_k(i, m)}{\sum_{n=1}^{N_S} \bar{v}_k(i, n)}, \quad m = 1, \dots, N_S \quad (3.63)$$

3.4.4 Computational complexity

The computational complexity of the MC-RBMPF is obtained similarly to the D-RBMPF, replacing the cardinality of the set of discrete points ($|\Pi|$) with the number of sub-particles N_S . Like in the D-RBMPF, the special case where $p(z_k \mid s_k, \theta) = p(z_k \mid s_k)$ simplifies the algorithm, but does not reduce overall complexity.

Table 3.1 shows a comparison of computational complexity between the SIR PF and the MC-RBMPF. The D-RBMPF is not shown in the table since as already mentioned, its only replaces N_S with $|\Pi|$ in complexity. Likewise, the SIR LWPF mentioned in Section 3.2.2 has the same complexity as the SIR PF. On obtaining these complexities, blind importance sampling is assumed to be used whenever applicable, and the proposal given by (3.55) is used for the sub-particle importance sampling.

We can see that the D-RBMPF and the MC-RBMPF have higher computational cost than the SIR PF and the SIR LWPF. This is both because the RBMPF is based on the marginal particle filter (which as shown in Klaas et al. [2005], has complexity quadratic on the number of particles), and because both the D-RBMPF and the MC-RBMPF associate multiple values of Θ with each par-

3. ONLINE BAYESIAN PARAMETER ESTIMATION USING THE RAO-BLACKWELLIZED MARGINAL PARTICLE FILTER

Table 3.1: Comparison of computational complexity between algorithms

Step	SIR PF	MC-RBMPF	SIR PF (parallelized)	MC-RBMPF (parallelized)
Particle importance sampling	$O(N_P)$	$O(N_P N_S)$	$O(1)$	$O(N_P N_S)$
Particle weight calculation	$O(N_P)$	$O(N_P^2 N_S)$	$O(1)$	$O(N_P N_S)$
Sub-particle importance sampling	—	$O(N_P N_S)$	—	$O(N_P N_S)$
Sub-particle weight calculation	—	$O(N_P^2 N_S)$	—	$O(N_P N_S)$
Resampling	$O(N_P)$	—	$O(N_P)$	—
Overall	$O(N_P)$	$O(N_P^2 N_S)$	$O(N_P)$	$O(N_P N_S)$

ticle i , in order to provide a numerical approximation of the conditional density $p(\theta | s_k(i), Z^k)$.

However, like the D-RBMPF, the MC-RBMPF has nice parallelization properties. The last two columns of Table 3.1 show the complexity of the parallelized versions of the SIR PF and the MC-RBMPF. It corresponds to the complexity for a single processing node, in case N_P processors were available. Since resampling cannot be easily made concurrent (see Bolić et al. [2005]), we assume that there is no reduction of complexity for the resampling procedures involved in all algorithms. As one can see, this causes the parallelized SIR PF (and the SIR LWPF) to have the same overall complexity as their non-parallelized counterparts, whereas with the parallelized MC-RBMPF, it is possible to reduce the overall complexity by assigning each particle to each processing node during the algorithm main loop.

3. ONLINE BAYESIAN PARAMETER ESTIMATION USING THE RAO-BLACKWELLIZED MARGINAL PARTICLE FILTER

3.5 Simulation - Turn rate estimation

3.5.1 Problem description

In the first simulation scenario, we will consider the D-RBMPF applied to the target tracking problem of estimating the turn rate of a target performing a constant turn. In this problem, we consider a moving target, that moves according to the constant turn model described in Rong Li and Jilkov [2003]. The time-varying state of the target is given by $S_k = [P_k^x, P_k^y, V_k^x, V_k^y]^T$, where x and y denote the Cartesian coordinates, (P_k^x, P_k^y) corresponds to the position (with realizations denoted by (p_k^x, p_k^y)) and (V_k^x, V_k^y) corresponds to the velocity (with realizations denoted by (v_k^x, v_k^y)). The parameter is the turn rate Θ , which will always be expressed in $^\circ/\text{second}$. The target is assumed to be observed by a sensor, that produces measurements according to the following observation model:

$$p(z_k | x_k, \theta) = p(z_k | x_k) = \mathcal{N} \left(z_k; \begin{bmatrix} p_k^x \\ p_k^y \end{bmatrix}, \begin{bmatrix} 2.25 & 0 \\ 0 & 2.25 \end{bmatrix} \right) \quad (3.64)$$

i.e. we have the special case described in Section 3.3.4.

3.5.2 Simulation description

For the given problem, we compare the D-RBMPF with the SIR LWPF described in Section 3.2.2, tuned using different values of the smoothing parameter h . Since both are coupled Bayesian techniques, they can be used to obtain both estimates and variances of Θ and S_k as they approximate the joint posterior $p(s_k, \theta | Z^k)$.

In the scenario, the true value of Θ is $\theta = 5$ and the power spectral density of the process noise (see Rong Li and Jilkov [2003]) is given by $\sigma_w^2 = 36$. The target trajectory is shown in Fig. 3.1. We perform a Monte Carlo simulation with 25 runs, generating a new sequence of observations at each run.

The compared algorithms are the D-RBMPF (with $N_P = 1,000$ and $\Pi = \{-9, -7, \dots, 7, 9\}$) and the SIR LWPF with $N_P = 1,000$, using the values $h = 0.02, 0.1$ and 0.5 for the smoothing parameter. We assume the following prior

3. ONLINE BAYESIAN PARAMETER ESTIMATION USING THE RAO-BLACKWELLIZED MARGINAL PARTICLE FILTER

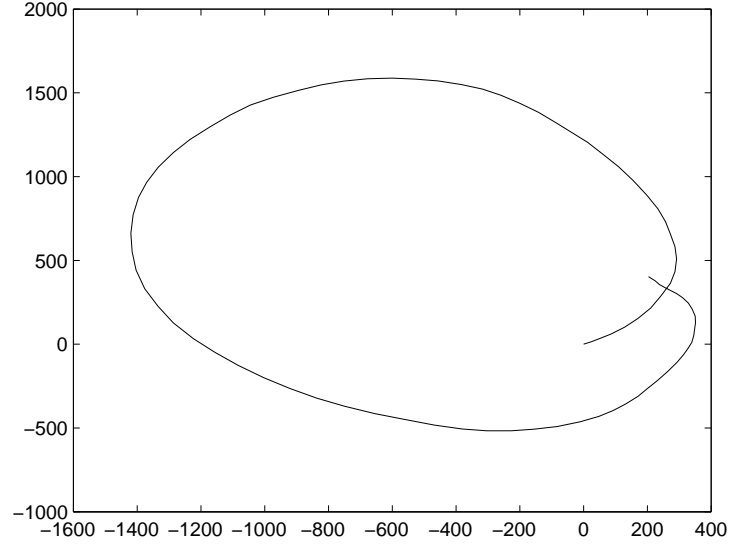


Figure 3.1: Target trajectory for the turn rate estimation scenario

densities for S_k and Θ :

$$p(s_0) = p(s_0|\theta) = \mathcal{N} \left(s_0; \begin{bmatrix} p_0^x \\ p_0^y \\ v_0^x \\ v_0^y \end{bmatrix}, \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 25 & 0 \\ 0 & 0 & 0 & 25 \end{bmatrix} \right),$$

$$p(\theta) = p(\theta|s_0) = \mathcal{N}(\theta; 0.01, 36)$$

where for the D-RBMPF, we consider a discretized version of $p(\theta)$, obtained by constraining the support of Θ to the set of discrete points Π and normalizing the values of the pdf.

For all implemented filters, blind importance sampling is used. We evaluate performance using two criteria: the Root Mean Square Errors (RMSE) of the Minimum Mean Square Error (MMSE) estimates and the Normalized Estimation Errors Squared (NEES) (Bar-Shalom et al. [2001]) obtained from the MMSE estimates and the variances. The RMSE of some scalar quantity Q_k (e.g. an

3. ONLINE BAYESIAN PARAMETER ESTIMATION USING THE RAO-BLACKWELLIZED MARGINAL PARTICLE FILTER

entry of the state or parameter vectors) are given by

$$\epsilon_k^{\text{RMSE}} = \sqrt{\frac{1}{N_R} \sum_{m=1}^{N_R} (\hat{q}_k(m) - q_k(m))^2} \quad (3.65)$$

where N_R is the number of Monte Carlo runs, and for the m -th run, $q_k(m)$ is the true value of the quantity and $\hat{q}_k(m)$ is its MMSE estimate calculated using the output of the filter. The NEES of the same quantity are given by

$$\epsilon_k^{\text{NEES}} = \frac{1}{N_R} \sum_{m=1}^{N_R} \frac{(\hat{q}_k(m) - q_k(m))^2}{\sigma_{q_k}^2(m)} \quad (3.66)$$

where $\sigma_{q_k}^2(m)$ is the variance of Q_k (also calculated using the output of the filter). Note that for the D-RBMPF, both the MMSE estimate and the variance can be calculated in a straightforward manner by using (3.22). The NEES are a performance measure of the filter's statistical consistency, i.e. the consistency between filter-calculated variances and actual estimation errors. In accordance to this thesis' goal of characterization of uncertainty in Bayesian estimation, it is interesting for us to use the NEES as a performance criterion in addition to the RMSE.

Generally, the NEES are compared with the bounds of an acceptance region (see Bar-Shalom et al. [2001]). For instance, a 95% acceptance region corresponds to the region that has a 95% chance of containing the NEES of some estimator, in the assumption that the estimator is consistent and that the underlying distribution (in our case, the marginal distributions of the entries of the state vector and of the parameter) is Gaussian. In theory, NEES higher than the upper bound indicate that the variances are underestimated (i.e. actual errors are higher than indicated by the variances), and when lower than the lower bound, indicate that the variances are overestimated. However, since these bounds are valid only in the Gaussian assumption, which is clearly not our case, they are more useful to have a notion of statistical consistency, rather than to make hard conclusions about it.

3. ONLINE BAYESIAN PARAMETER ESTIMATION USING THE RAO-BLACKWELLIZED MARGINAL PARTICLE FILTER

3.5.3 Results

Although we have calculated the RMSE and NEES for both state and parameter, we only show the performance metrics for the parameter, as in this example, state estimation performance is much less affected by the choice of algorithm. Let us then first take a look at the RMSE of the turn rate estimates, shown in Fig. 3.2(a). The D-RBMPF obtained by far the best results, followed by the LWPFs with $h = 0.1$, $h = 0.5$ and $h = 0.02$ respectively. The NEES of the turn rate estimate are shown in Fig. 3.2(b), together with the upper and lower bounds of the 95% acceptance region. As we can see, for all filters, the NEES were most of the time above the upper bounds, which indicates that filter-calculated variances tended to underestimate the actual estimation errors. The D-RBMPF, however, has lead to much lower NEES than the LWPFs.

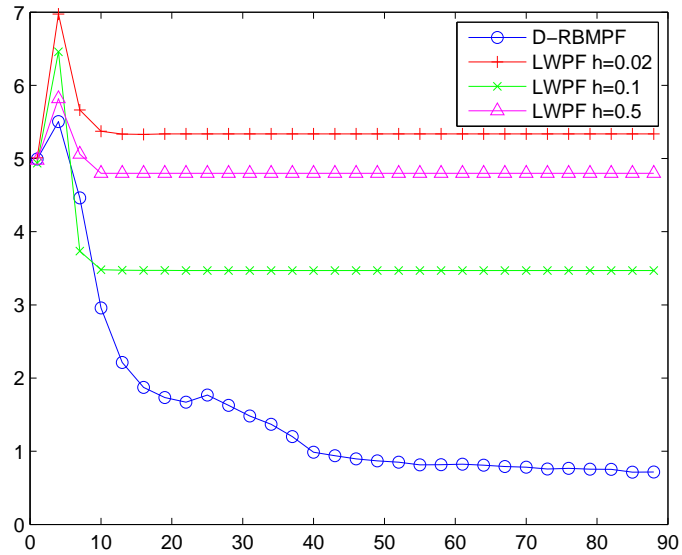
In order to find the cause of the high NEES, we give a look at the number of “outlier” estimates produced by each of the algorithms. We classify an estimate $\hat{\theta}_k$ as an “outlier” if its error w.r.t. the true value of θ is more than $3\sqrt{\sigma_{\theta_k}^2}$ (where $\sigma_{\theta_k}^2$ is the filter-calculated variance of Θ). The rate of outliers is given by the number of outliers divided by the number of Monte Carlo runs, and shown in Fig. 3.3.

From Fig. 3.3, we can see that almost all estimates produced by the LWPFs were outlier estimates. In the D-RBMPF, this happened only for a small percentage of the runs. For this subset of runs, however, we verified that the NEES were exceptionally high, leading to the result in Fig. 3.2(b).

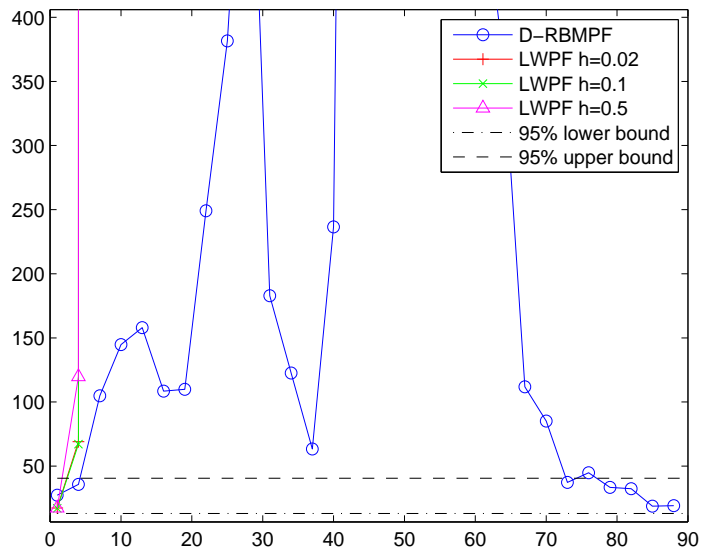
Overall, for this problem, the D-RBMPF has been shown successful on reducing the estimation bias caused by methods based on artificial dynamics, leading to both smaller estimation errors and more statistically consistent estimates (i.e. better characterization of estimation uncertainty) than the LWPF for different values of the smoothing parameter h . The results also illustrate the advantage of using a non-parametric numerical approximation for the conditional density $p(\theta | s_k(i), Z^k)$, as the performance of the LWPF in terms of RMSE seems to be very sensitive to the choice of the smoothing parameter, with $h = 0.1$ being the choice that has lead to best performance for this problem.

Yet, the performance of the D-RBMPF in terms of statistical consistency

3. ONLINE BAYESIAN PARAMETER ESTIMATION USING THE RAO-BLACKWELLIZED MARGINAL PARTICLE FILTER



(a) Turn rate RMSE



(b) Turn rate NEES

Figure 3.2: Turn rate estimation – results

3. ONLINE BAYESIAN PARAMETER ESTIMATION USING THE RAO-BLACKWELLIZED MARGINAL PARTICLE FILTER

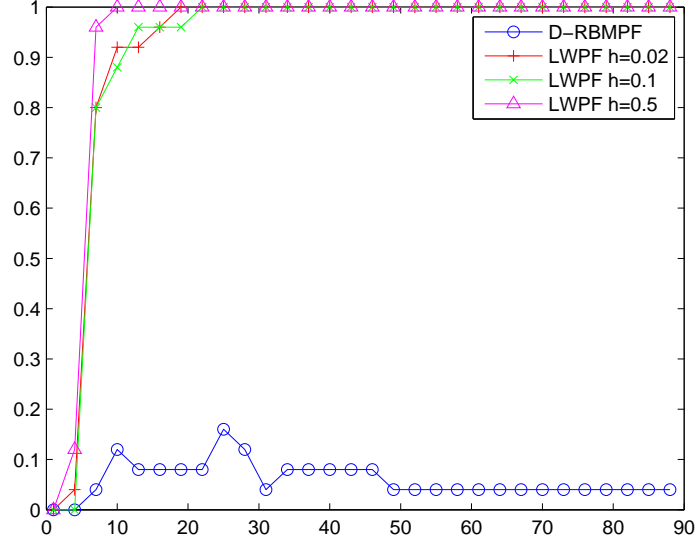


Figure 3.3: Turn rate estimation – ratio of outlier estimates

(NEES) was not completely satisfactory, with a small subset of the runs leading to highly underestimated variances. This suggests that a larger number of particles and/or a larger number of discrete points is needed, or perhaps other improvements in the algorithm.

3.5.4 Trade-off between computational cost and performance

In the previous experiment, we did not consider the difference in computational complexity between the D-RBMPF and the LWPF. However, from Section 3.3.5, we know that the D-RBMPF is computationally more expensive than the LWPF. Even if parallelization is possible, the D-RBMPF is at least $|\mathbb{I}|$ times more costly than the LWPF. Therefore, it is worthwhile to verify which algorithm offers a better trade-off between computational cost and performance, by using a LWPF with a larger number of particles.

We then repeat the same simulation described in Section 3.5.2, but we test only the D-RBMPF and the SIR LWPF with $h = 0.1$ (the value of h which lead

3. ONLINE BAYESIAN PARAMETER ESTIMATION USING THE RAO-BLACKWELLIZED MARGINAL PARTICLE FILTER

to best results in the previous experiment), with the LWPF (only) using $|\Pi| = 10$ times more particles, i.e. 10,000 particles. The results, in terms of RMSE and ratio of outliers, are shown in Fig. 3.4.

We can see that using more particles in the LWPF led to a significant improvement in terms of RMSE, albeit its performance was still inferior to the D-RBMPF. In terms of ratio of outliers, increasing the number of particles of the LWPF led to some improvement, but value of the metric was still much higher than for the D-RBMPF. Hence, in terms of characterization of uncertainty, the D-RBMPF produced seems to fare better than just using a LWPF with a large number of particles.

3.6 Simulation - Stochastic volatility estimation

3.6.1 Problem description

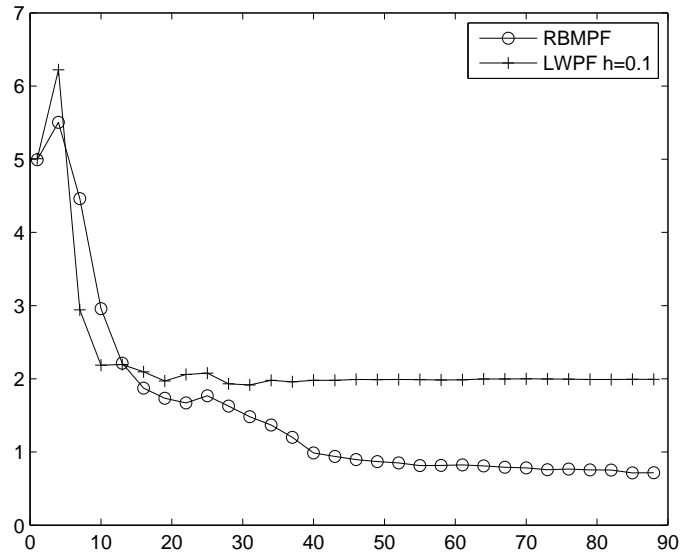
In the second simulation scenario, we apply the MC-RBMPF to the problem of stochastic volatility estimation from stock data using the Heston model (Heston [1993]). More precisely, we consider a discretized version of this model (using the Euler scheme), described in Aihara et al. [2009]:

$$S_k = \left(1 + \frac{1}{2}\xi\rho\Delta t\right)^{-1} \left\{ S_{k-1} + \kappa \left(\frac{\kappa'}{\kappa} - S_{k-1}\right) \Delta t + \xi\rho\frac{1}{2}S_{k-1}\Delta t + \xi\sqrt{S_{k-1}}\sqrt{1-\rho^2}\Delta\tilde{Z}_k + \xi\rho\sqrt{S_{k-1}}\Delta B_k \right\}, \quad (3.67)$$

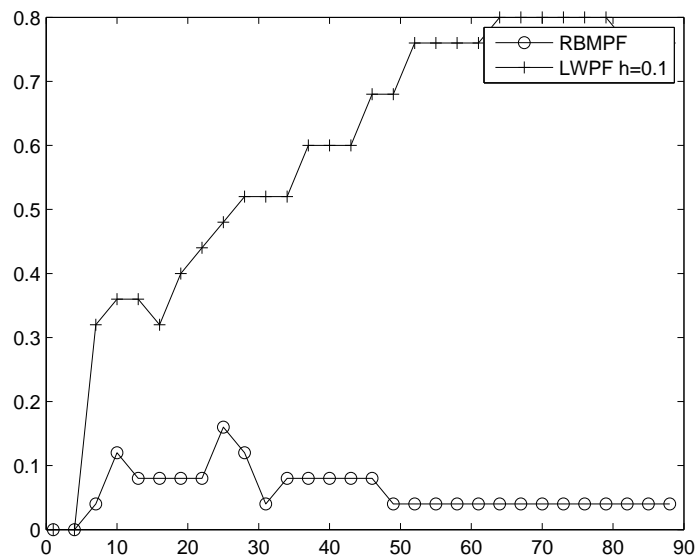
$$Z_k = Z_{k-1} + \left(\mu_S - \frac{1}{2}S_k\right) \Delta t + \sqrt{S_{k-1}}\Delta B_k \quad (3.68)$$

where Δt is the time interval between $k - 1$ and k , and $\Delta\tilde{Z}_k$ and ΔB_k are i.i.d. sequences of standard normal variables and the sequences themselves are independent. In this model the state to be estimated is the stochastic volatility process S_k , based on the observations Z_k (computed from the observed stock prices). The unknown parameter Θ is the vector $\Theta = [\kappa, \kappa', \xi, \mu_S, \rho]^T$, where, for the sake of simplicity, we use the same notations for the individual entries of

3. ONLINE BAYESIAN PARAMETER ESTIMATION USING THE RAO-BLACKWELLIZED MARGINAL PARTICLE FILTER



(a) Turn rate RMSE



(b) Ratio of turn rate outlier estimates

Figure 3.4: Turn rate estimation – results for the LWPF with 10 times more particles

3. ONLINE BAYESIAN PARAMETER ESTIMATION USING THE RAO-BLACKWELLIZED MARGINAL PARTICLE FILTER

vector Θ and their realizations.

Note that this system model is more general than the more commonly used model given by (3.1), but it can be straightforwardly shown that the MC-RBMPF can be applied to this model as well, by using:

- $p(s_k|s_{k-1}, \theta, z_{k-1})$ instead of $p(s_k|s_{k-1}, \theta)$ as the state transition density;
- $p(z_k|s_k, s_{k-1}, \theta, z_{k-1})$ instead of $p(z_k|s_k, \theta)$ as the likelihood function.

These densities are given by

$$p(s_k|s_{k-1}, \theta, z_{k-1}) = \mathcal{N}(s_k; m_{s_k}(s_{k-1}, \theta), \sigma_{s_k}^2(s_{k-1}, \theta)) \quad (3.69)$$

$$p(z_k|s_k, s_{k-1}, \theta, z_{k-1}) = \mathcal{N}(z_k; m_{z_k}(s_{k-1}, \theta, z_{k-1}), \sigma_{z_k}^2(s_{k-1})) \quad (3.70)$$

where

$$m_{s_k}(s_{k-1}, \theta) = \left(1 + \frac{1}{2}\xi\rho\Delta t\right)^{-1} \left(s_{k-1} + \kappa \left(\frac{\kappa'}{\kappa} - s_{k-1}\right) \Delta t + \frac{\xi\rho}{2}s_{k-1}\Delta t\right)$$

$$\sigma_{s_k}^2(s_{k-1}, \theta) = \left(1 + \frac{1}{2}\xi\rho\Delta t\right)^{-2} \xi^2 s_{k-1} \Delta t$$

$$m_{z_k}(s_{k-1}, \theta, z_{k-1}) = z_{k-1} + \left(\mu_S - \frac{1}{2}s_k\right) \Delta t$$

$$\sigma_{z_k}^2(s_{k-1}) = s_{k-1} \Delta t.$$

3.6.2 Simulation description

For this problem, we compare the MC-RBMPF with the SIR LWPF, again tuned using different values of h . In our scenario, the true values of the parameters are $\kappa = 3$, $\kappa' = 0.3$, $\mu_S = 0.1$, $\rho = -0.2$, $\xi = 0.4$, and the true initial state and the initial observation are respectively given by $s_0 = 0.25$ and $z_0 = 0$. We perform a Monte Carlo simulation with 25 runs, generating a new stochastic volatility trajectory (s_0, \dots, s_f) and sequence of observations Z^f at each run, with $f = 252$ and $\Delta t = 1/252$ (measured in years).

The compared algorithms are the MC-RBMPF (with $N_P = 400$ and $N_S = 40$) and the SIR LWPF with $N_P = 400$, using the values $h = 0.02, 0.1$ and 0.5 for

3. ONLINE BAYESIAN PARAMETER ESTIMATION USING THE RAO-BLACKWELLIZED MARGINAL PARTICLE FILTER

the smoothing parameter. The filters assume that in the absence of observations, the state and parameters are mutually independent, with priors given by

$$\begin{aligned}
 p(\kappa) &= \mathcal{U}(1, 9), & p(\kappa') &= \mathcal{U}(0.05, 0.4), \\
 p(\mu_S) &= \mathcal{U}(0.05, 0.5), & p(\rho) &= \mathcal{U}(-0.5, 0), \\
 p(\xi) &= \mathcal{U}(0.01, 0.91), & p(s_0) &= \mathcal{N}(0.25, 4 \cdot 10^{-4}).
 \end{aligned}$$

For all implemented filters, blind importance sampling is used (the optimal importance sampling scheme for this problem, described in Aihara et al. [2009], does not result in significant improvement). We evaluate performance again using the RMSE and the NEES, this time for both the state and the individual entries of the parameter vector.

3.6.3 Results

First, let us take a look at the MMSE estimates of the states and parameters, or more precisely at their RMSE, shown in Figs. 3.5. We can see that overall, the RBMPF obtained the best parameter estimation performance in terms of RMSE, leading to the smallest errors for the parameters κ , κ' and ρ . It was followed by the LWPF with $h = 0.1$, which had RMSE close to the RBMPF for the parameters κ , μ_S and ξ . The LWPF with $h = 0.02$ and $h = 0.5$ had good performance respectively for the parameters μ_S and ξ , but overall their performance was worse than the other two filters. For the RMSE of the stochastic volatility s_k , the difference in the results is somehow difficult to observe (possibly due to low parameter identifiability), but with a small margin of difference, the RBMPF seems to have the smallest RMSE and the LWPF with $h = 0.5$ seems to have the largest.

The NEES of the estimated state and parameters are shown in Figs. 3.6, together with the upper and lower bounds of the 95% acceptance region. We can see that the RBMPF also leads to the best results in terms of NEES, with the metric approximately inside the acceptance region for three of the parameters, and also for the stochastic volatility s_k . All the other filters have state and parameter estimates quite above the upper bound of the acceptance region, indicating that

3. ONLINE BAYESIAN PARAMETER ESTIMATION USING THE RAO-BLACKWELLIZED MARGINAL PARTICLE FILTER

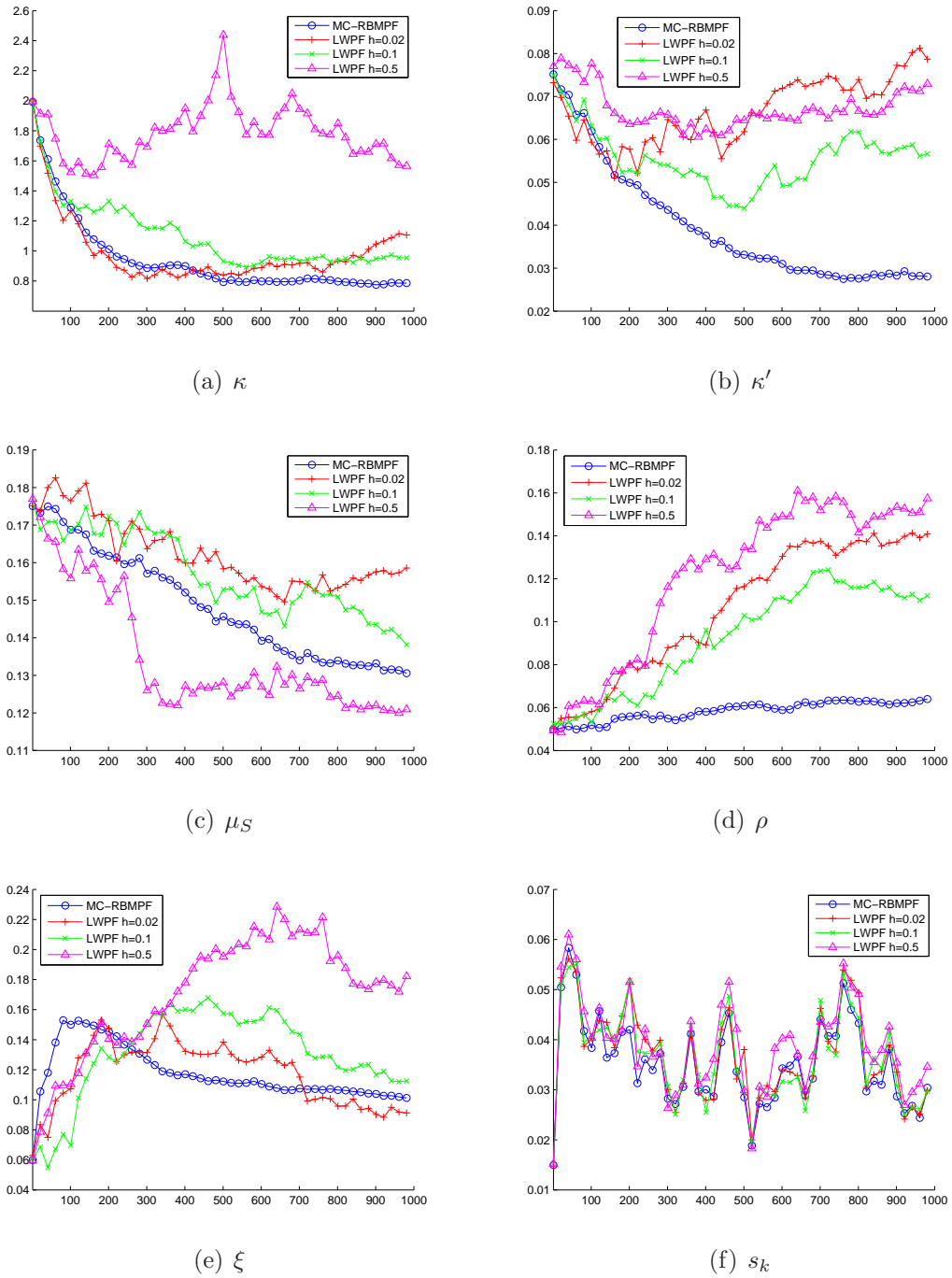


Figure 3.5: Stochastic volatility estimation - RMSE results

3. ONLINE BAYESIAN PARAMETER ESTIMATION USING THE RAO-BLACKWELLIZED MARGINAL PARTICLE FILTER

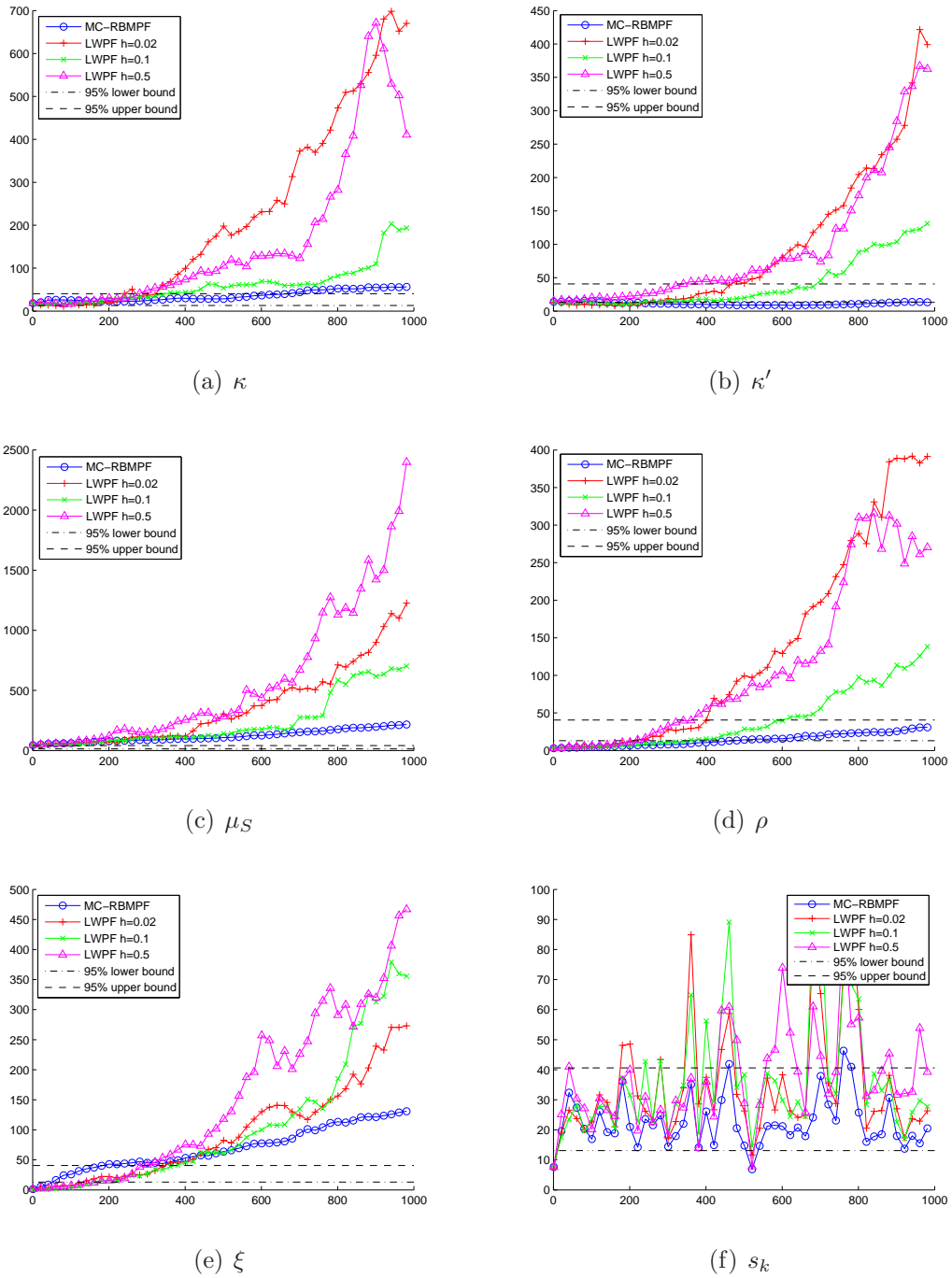


Figure 3.6: Stochastic volatility estimation - NEEs results

3. ONLINE BAYESIAN PARAMETER ESTIMATION USING THE RAO-BLACKWELLIZED MARGINAL PARTICLE FILTER

the filter-calculated variances tend to underestimate the actual estimation errors. However, the LWPF with $h = 0.1$ has lead to noticeably less deviation from the acceptance region that the LWPF with $h = 0.02$ or $h = 0.5$.

Again, the RBMPF has been shown to produce more accurate estimates and better characterization of uncertainty than all LWPFs, and the performance of the LWPF in terms of both RMSE and NEES has been shown to be sensitive to the choice of the smoothing parameter. Again, $h = 0.1$ was the best choice of parameter for the LWPF.

3.6.4 Trade-off between computational cost and performance

As we did for the turn rate estimation problem, we will run another experiment to analyze the trade-off between computational complexity and performance of the tested algorithms. We then repeat the same simulation described in Section 3.6.2, but we test only the MC-RBMPF and the SIR LWPF with $h = 0.1$, with the LWPF (only) using $N_S = 40$ times more particles, i.e. 16,000 particles. This gives the SIR LWPF the same complexity of the (fully parallelized) MC-RBMPF with $N_P = 400$ and $N_S = 40$.

The results, in terms of RMSE and NEES, are respectively shown in Fig. 3.7 and 3.8. For this experiment, the LWPF with $h = 0.1$ outperforms the MC-RBMPF in terms of parameter estimation, leading to smaller RMSE for three of the five parameters, and overall lower NEES (although the NEES was lower than the lower bounds for some parameters, suggesting an overestimation of variance). In terms of state estimation, the performance of the two algorithms was quite similar. We can then say that for this scenario, the SIR LWPF with $h = 0.1$ offers overall a better trade-off between computational complexity and performance than the MC-RBMPF.

3.7 Conclusions and recommendations

In this chapter, we have proposed two novel joint and state parameter estimation algorithms, more specifically, two new versions of the RBMPF, a technique that

3. ONLINE BAYESIAN PARAMETER ESTIMATION USING THE RAO-BLACKWELLIZED MARGINAL PARTICLE FILTER

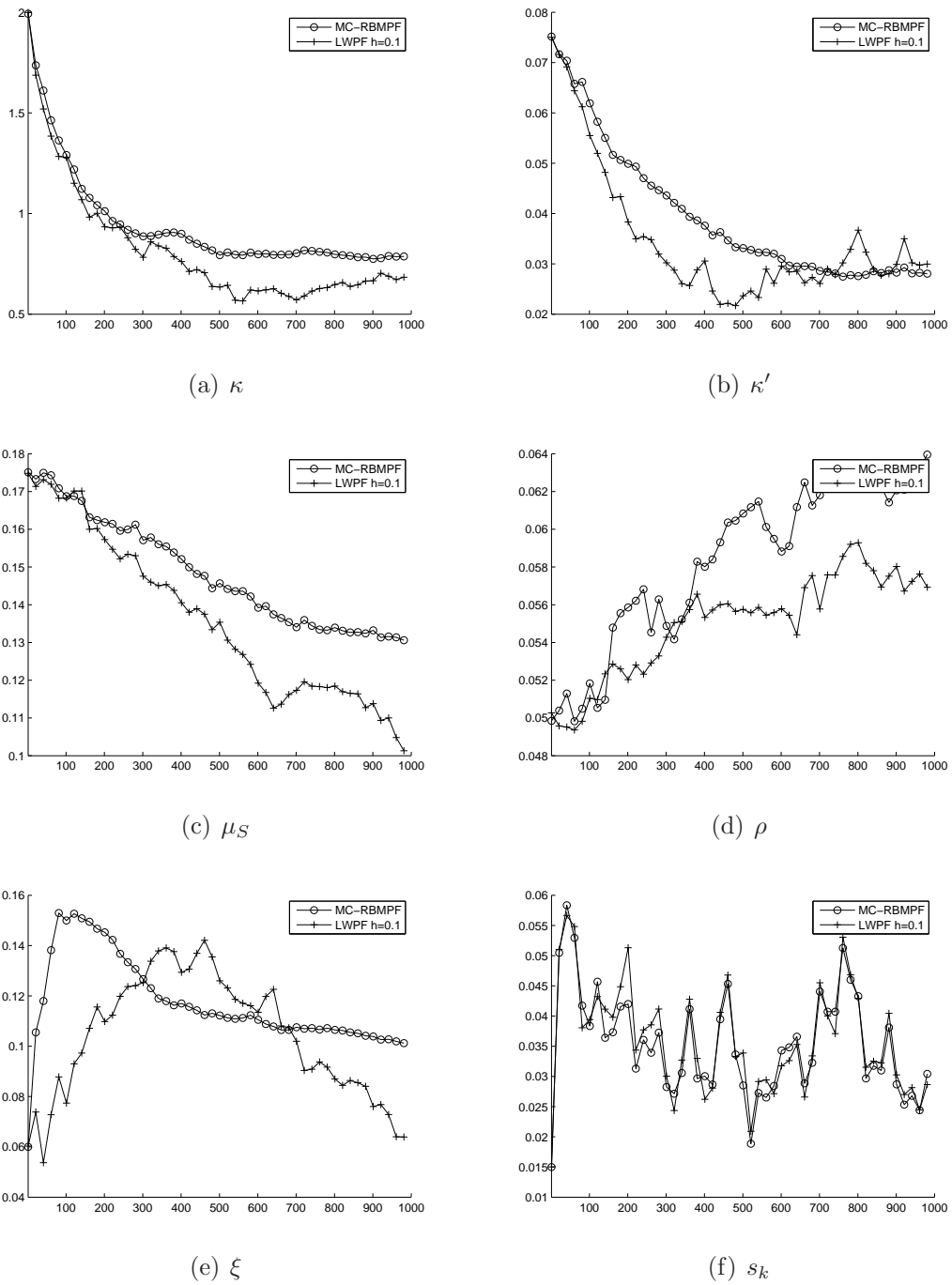
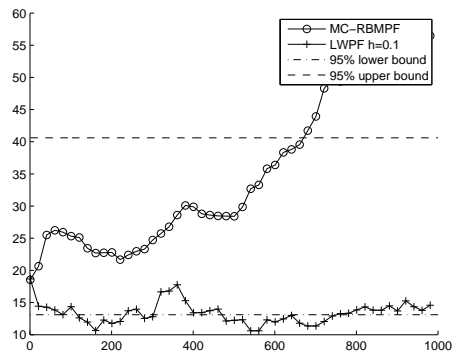
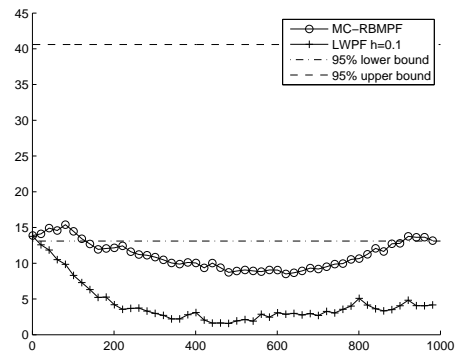


Figure 3.7: Stochastic volatility estimation - RMSE results for the LWPF with 40 times more particles

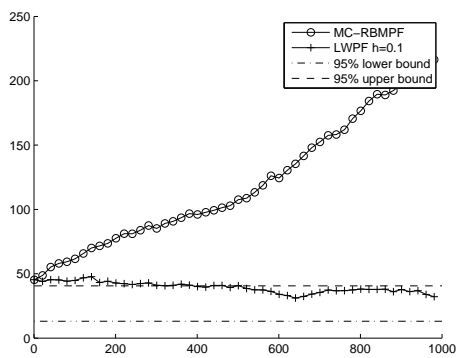
3. ONLINE BAYESIAN PARAMETER ESTIMATION USING THE RAO-BLACKWELLIZED MARGINAL PARTICLE FILTER



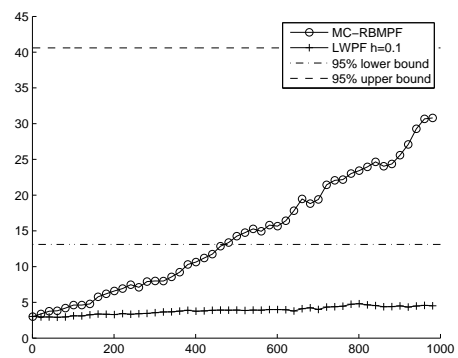
(a) κ



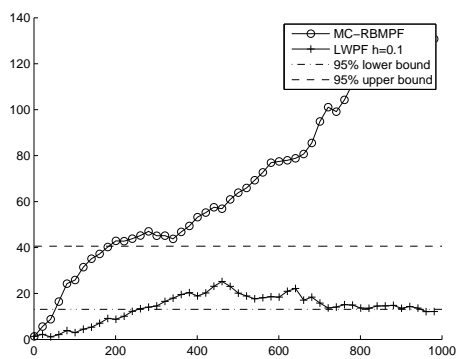
(b) κ'



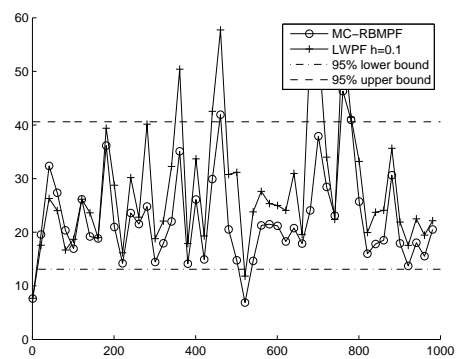
(c) μ_S



(d) ρ



(e) ξ



(f) s_k

Figure 3.8: Stochastic volatility estimation - NEES results for the LWPF with 40 times more particles

3. ONLINE BAYESIAN PARAMETER ESTIMATION USING THE RAO-BLACKWELLIZED MARGINAL PARTICLE FILTER

combines a SMC filter with a non-sequential MC estimator. The proposed methods are general, in the sense that they do not require the underlying process and observation model to have partly linear-Gaussian structure, and are designed to mitigate the bias caused by the introduction of artificial dynamics, typical of online Bayesian PF-based parameter estimation methods. The proposed algorithms have also nice parallelization properties. By examining their performance in two practical problems, we have seen that they can be an effective solution to online joint state and parameter estimation, in particular when good characterization of uncertainty is needed.

In comparison with the SIR LWPF, however, the proposed RBMPF algorithms have higher computational cost. We have verified that depending on the application, it might be more cost-effective to use a LWPF with a carefully chosen h parameter and a large number of particles. It is therefore highly desirable to find a RBMPF implementation with lower computational cost but still applicable to highly general models. One idea is to replace the Monte Carlo-based estimator used to approximate the conditional density $p(\theta | s_k(i), Z^k)$ in the MC-RBMPF with a bank of parallel deterministic estimators (for instance, Extended or Unscented Kalman Filters).

In addition, although we have tested the proposed algorithms only in online parameter estimation context, there is no reason to believe that they cannot be applied to off-line parameter estimation problems. For instance, to adapt the MC-RBMPF for off-line processing, we could use higher numbers of particles/sub-particles, MCMC steps to reduce the variance of sub-particle weights, and alternative empirical densities. As an example, if lower/upper bounds on the parameter values are known, a truncated Gaussian empirical density might be a better choice of empirical density than a Gaussian. Such adaptations are also interesting subjects for future work. Finally, it is worthwhile investigating whether (and when) the MPF step in the RBMPF can be replaced by a step of a SIR PF, which is computationally cheaper. In the track labelling problem, which we will examine in Chapter 4, we will see that this is certainly possible.

Chapter 4

The problem of optimal Bayesian track labelling in multi-target tracking

In Multi-Target Tracking (MTT), the problem of assigning labels to tracks (track labelling) is vastly covered in literature, but its exact mathematical formulation, in terms of Bayesian statistics, has not been yet looked at in detail. Doing so would, however, help us to better understand and tackle Bayes-optimal track labelling, as well as some practical difficulties associated with the MTT problem, in particular, the situation where targets move in close proximity with each other and thereafter separate. In such situation, it is well-known that there is confusion on target identities, also known as “mixed labelling”.

In this chapter, we rigorously formulate the probabilistic track labelling problem using Finite Set Statistics (FISST), showing how the multi-target posterior density considering labels can be recursively calculated in a Bayesian manner. On the basis of this density, we derive statistics associated with track labelling, with sound physical interpretation, that can be used to quantify labelling uncertainty, find the optimal track label assignment, and evaluate track labelling performance.

Computation of these labelling-related statistics is possible, in principle, using state-of-the-art multi-target tracking algorithms, including the Multi-target Sequential Monte Carlo (M-SMC) and the Multiple Hypothesis Tracking (MHT)

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING

filters. In practice, however, it is well-known that these methods suffer from the degeneracy phenomenon, causing them to underestimate uncertainty-related quantities, like the probability of labelling error. To deal with this this problem, we also propose, in this chapter, a new Sequential Monte Carlo algorithm for the multi-target tracking and labelling problem – the Labelling Uncertainty-Aware Particle Filter (LUA-PF) – and validate it using simulation.

4.1 Introduction

The track labelling problem is perhaps just as old as the multi-target tracking problem itself. In the display of a radar operator, it is often necessary not only to display the estimated position of the multiple objects (i.e. the tracks), but also attribute a unique label to each track. Ideally, this track label should consistently be associated with the same real-world object, enhancing the situational awareness of the operator.

In practice, the feasibility of maintaining this label-to-target consistency depends on observability conditions. One situation where this consistency is frequently lost is after targets move in close proximity to each other. In this case, after the separation, the measurements and initial information may not allow us to precisely determine which target is which. Therefore, if required to make a hard decision to assign labels to tracks, the tracking system will frequently make wrong choices. This situation (with two targets as example) is illustrated in Fig. 4.1.

This situation where the available information allows more than one labelling possibility is referred to as “mixed labelling” by Boers, Sviestins and Driessen Boers et al. [2010]. In this situation, two questions – which form the main motivation of this work – are highly relevant:

- **Question 1:** *How does one optimally assign labels $T1$ and $T2$ to the two tracks?*
- **Question 2:** *What is the probability that the assignment is incorrect, i.e. that track swap has occurred?* This probability may be useful to the operator; for instance, when a decision is only acceptable if we have high

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING

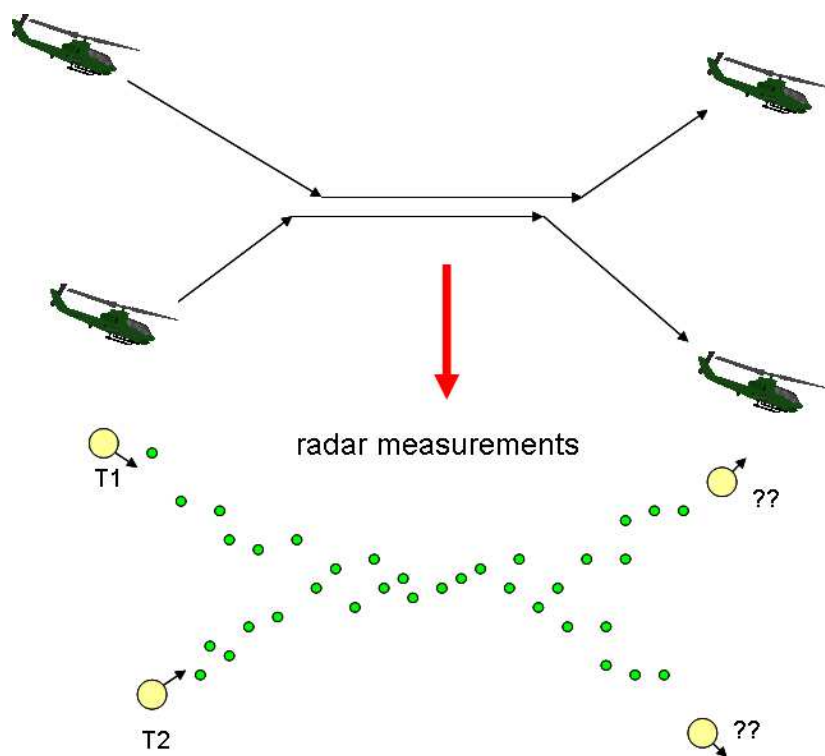


Figure 4.1: Situation where assignment of labels to tracks is ambiguous

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING

confidence that a target is who it seems to be.

These two questions can be straightforwardly extended for situations involving three or more targets. The problem, however, is that the questions are not perfectly clear. For instance, what do we exactly mean by “probability of incorrect labelling”? After all, tracks are only estimates of the true target states, and in probabilistic sense, they almost never coincide. If the tracks are themselves not “correct”, what shall we understand by “correct labelling”? To add even more complexity, what would be the “correct labelling” if we assume that we do not know precisely where the targets initially are, or if we assume that other targets may appear later?

This idea of obtaining target identities using a probabilistic approach is known for some time in the literature, e.g. in Salmond et al. [1997]. More general approaches, considering arbitrary number of targets and possibility of target birth and death, have appeared in García-Fernández and Grajal [2009]; García-Fernández et al. [2012]; Ma et al. [2006]; Morelande et al. [2007]; Vo and Vo [2011], but the problem of labelling in situations such as Fig. 4.1 has not been considered. Recent works Blom and Bloem [2009, 2011]; Crouse et al. [2011a]; García-Fernández et al. [2011]; Georgescu et al. [2012] propose quantities associated with labelling uncertainty that could be used to answer **Questions 1 and 2**, but the physical interpretation of these quantities is not clear from their mathematical description. They also have restrictive assumptions such as considering the target dynamics as linear-Gaussian, or being defined for only two targets, or assuming the number of targets to be known and time-invariant.

The difficulty to even properly formulate the questions urges us to look at the Multi-Target Tracking and Labelling (MTTL) problem from a more fundamental perspective. We do so by resorting to a rigorous formulation and analysis of the problem of MTTL in a Bayesian framework. In this chapter, we delve into **Questions 1 and 2** in a more general set-up including target birth and death. First, we focus on providing the *correct questions*, i.e. giving a proper mathematical formulation to **Questions 1 and 2**, and then we proceed to giving *correct answers* to these questions (i.e. presenting practical filtering algorithms for the MTTL problem).

The contributions of this chapter are:

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING

1. We provide a mathematical description of the general MTTL problem using FISST, showing how the *labelled multi-target posterior density* can be recursively calculated;
2. We provide a *statistical description of the labelling error* with clear physical interpretation, based on the labelled multi-target posterior density: the *labelling probability*. This gives a proper formulation for **Question 2**;
3. We propose four conceptual *track extraction schemes* for the Bayesian MTTL problem, which describe how the optimal assignment of labels should be performed, i.e., we give proper formulations for **Question 1**;
4. We propose two metrics to evaluate labelling performance for MTTL systems;
5. We provide methods to calculate the labelling probability for two well-known MTT algorithms: the MHT and the M-SMC filter, i.e., we give also *answers* to **Questions 1 and 2**, although we note that these methods suffer from the degeneracy phenomenon;
6. Last but not least, we present a MTTL algorithm that avoids degeneracy and is applicable to general multi-target scenarios with time-varying number of targets, the LUA-PF.

The organization of this chapter is as follows. In Section 4.2, we show how the labelled multi-target posterior can be recursively calculated in a Bayesian manner. In Section 4.3, we propose a number of useful statistics for Bayesian labelling, including the labelling probability. In Section 4.4, we provide methods to calculate the labelling probability for the MHT and M-SMC filter algorithms, and make a comment on the degeneracy phenomenon. In Section 4.5, we introduce the LUA-PF method for the labelling problem and discuss various practical aspects. In Section 4.6, we present simulation results for the LUA-PF applied to the problem of tracking closely spaced targets. In Section 4.7, we draw conclusions.

4.2 Bayes formulation of the multi-target tracking and labelling (MTTL) problem

In this section, we will show how the labelled multi-target posterior density (the labelled posterior RFS density when the FISST formulation is used) can be recursively calculated in a Bayesian manner. This posterior density will be necessary to calculate useful statistics for the Bayesian labelling problem in Section 4.3.

Let us assume that the single-target state vector (composed of entries such as position, velocity, etc., which we will henceforth refer to simply as *location*) assumes values in \mathbb{R}^n , and that a label to be assigned to a track may assume values in a discrete set Π . We then define the *labelled multi-target state* at time k as the random finite set

$$\mathbf{X}_k = \left\{ X_k^{(1)}, \dots, X_k^{(T_k)} \right\}$$

where $X_k^{(i)} = \left[S_k^{(i)}, L_k^{(i)} \right]$ with $S_k^{(i)} \in \mathbb{R}^n$ (corresponding to the the location) and $L_k^{(i)} \in \Pi$ (corresponding to the label)¹. Since, in order to make labels useful as target identifiers, no two single-target states can have the same label, a RFS density function associated with \mathbf{X}_k (referred to as a *labelled RFS density*), must satisfy

$$\begin{aligned} f \left(\left\{ \left[S_k^{(1)}, l_k^{(1)} \right], \dots, \left[S_k^{(t_k)}, l_k^{(t_k)} \right] \right\} \right) &= 0, \\ \text{if } \exists i, j \in \{1, \dots, t_k\} \text{ s.t. } i \neq j, l_k^{(i)} &= l_k^{(j)}. \end{aligned} \quad (4.1)$$

Examples of closed-form RFS densities that satisfy (4.1) are the labelled Poisson RFS density and the labelled multi-Bernoulli RFS density described by Vo and Vo [2011].

Let us denote the corresponding observation as \mathbf{Z}_k (i.e. also a RFS), and the sequence of all observations available until and including time k by Z^k . Note that there is no loss of generality on treating observations as finite sets, as an observation modeled as a vector can also be modeled as a RFS with a single

¹In this chapter, for convenience, we will write all vectors as row vectors.

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING

vector element.

As we have seen in Section 2.3.4, in order to recursively compute the multi-target posterior RFS density $f(\mathbf{x}_k | Z^k)$, we need to be able to calculate the prior $f(\mathbf{x}_0)$, the likelihood $f(\mathbf{z}_k | \mathbf{x}_k)$ and for the state transition density $f(\mathbf{x}_k | \mathbf{x}_{k-1})$. Before delving into these densities, we will first take a look at the physical interpretation of the Bayesian labelling problem and delineate its scope. This will be important to derive formulas that make sense from a physical perspective.

4.2.1 Physical interpretation of the Bayesian labelling problem

From our discussion in Section 2.3.3, it becomes clear that a label is not a physical quantity in the traditional sense, as it does not describe any real-world property of targets. A label is instead an “imaginary stamp” that is a placeholder for the identity of a target¹. This causes labels to have physical interpretation only when one considers multi-target *trajectories*, and not the individual multi-target states at a particular time instant.

As an example, let us suppose that the single-target state $X_k^{(i)}$ is described by $S_k^{(i)} \in \mathbb{R}^2$ and $L_k^{(i)} \in \{A, B\}$. If one considers solely the realization of the multi-target labelled state \mathbf{X}_k given by

$$\mathbf{x}_k = \{[4, -2, A], [3, 7, B]\}$$

then the labels A and B do not have any practical meaning. But if one considers the multi-target trajectory given by $(\mathbf{x}_k, \mathbf{x}_{k+1})$, where we also have

$$\mathbf{x}_{k+1} = \{[3.4, -2.7, A], [3.4, 6.2, B]\}$$

then labels A and B have clear physical meaning: they establish, for instance, that location $(4, -2)$ at time k corresponds to the same target as location $(3.4, -2.7)$ at time $k + 1$.

¹Although there are other less common interpretations of a label, like a sequence of measurement-to-target associations (e.g. Horridge and Maskell [2009]), if one assumes one-to-one correspondence between non-clutter measurements and targets.

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING

We can then define the Bayesian labelling problem by attributing the labels (i.e. the imaginary stamps) to the locations at time $k = 0$ (or the desired start of labelled tracking). If the initial locations are not assumed to be exactly known, we may instead assign labels to the initial (prior) distributions of the locations. For the following time steps ($k > 0$), the Bayesian labelling problem becomes that of attributing the “imaginary stamps”, in a probabilistic manner, to the locations. Therefore, if we do not take target births into account, labelling is essentially a problem of “dot connecting”, i.e. connecting locations at a given time step (say k) with the initial location distributions at time 0.

Note that if the initial location distributions have large overlap, from the probabilistic point of view, the targets are already (moving) in close proximity. Hence we have confusion about the target identities from the very beginning of the tracks. In the extreme case, if all targets have the same initial distribution, then naturally all possible labels (dotted connections) will be equivalent, i.e., have equal probabilities. We call this an *ill-posed* Bayesian labelling problem, since one cannot extract any meaningful results as far as the labels are concerned. On the other hand, if the initial targets are separated (or equivalently, the location distributions are distinct, having very little overlap), we have a chance to draw meaningful conclusions and we consider this as a *well-posed* labelling problem. For these reasons, the assumption of well-separated initial target distributions is common in the Bayesian labelling literature (e.g. Blom and Bloem [2011]; Crouse et al. [2011a]; García-Fernández et al. [2011, 2012]). Fig. 4.2 illustrates examples of well-posed and ill-posed Bayesian labelling problems.

When one has no idea at all about the initial target states and the best location prior is the uniform distribution over the entire surveillance area, Bayesian labeling can be more practically applied once the locations of the targets are precisely identified. A possibility is to have the track initiation performed by a separate process and to include the tracks in the Bayesian (labelling) recursion only after having a high probability of existence with relatively small overlap among the location distributions. A Bayesian MTTL implementation based on this idea has been recently provided in García-Fernández et al. [2012].

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING

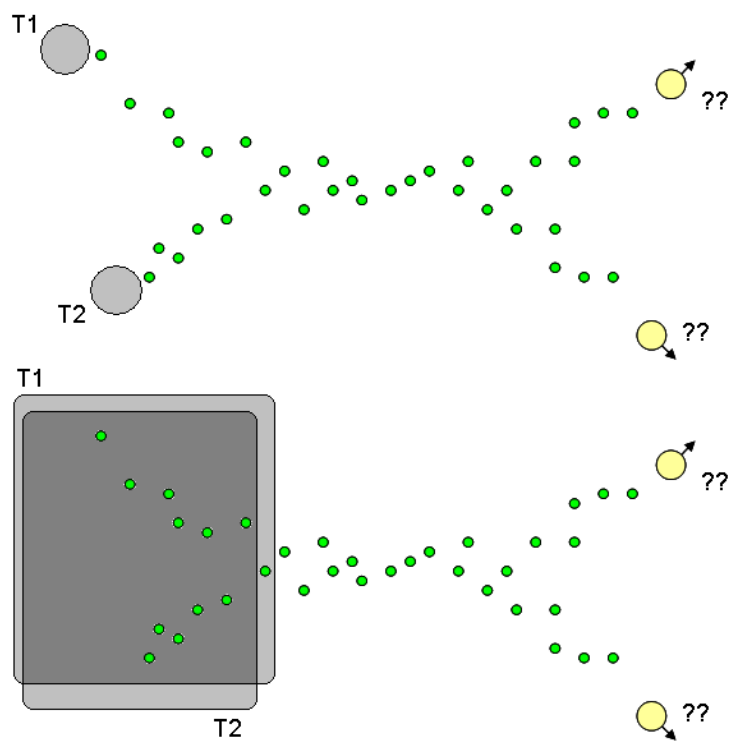


Figure 4.2: Well-posed (above) and ill-posed (below) Bayesian labelling problems

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING

4.2.2 The prior pdf

From our discussion in Section 4.2.1, we can formulate a well-posed Bayesian labelling problem when we assume that the targets' (location) initial distributions are reasonably distinct. It is also convenient to assume that they are independent, as it will make easier to write down the multi-target prior as a closed-form expression. Both the labelled Poisson and multi-Bernoulli RFS densities presented in Vo and Vo [2011] treat the initial distributions as independent, but only the multi-Bernoulli RFS allows them to be distinct, being thus an appropriate choice for the labelled multi-target prior.

With the initial state \mathbf{X}_0 being a labelled multi-Bernoulli RFS, the initial targets have distinct labels, contained in a set, say, \mathbb{L}_0 , where each label $l \in \mathbb{L}_0$ is associated with one target with (location) initial distribution $p^{(l)}(\cdot)$ and with a probability of existence $r^{(l)}$. The multi-target prior is then given by

$$\begin{aligned} f(\mathbf{x}_0) &= f\left(\left\{\left[s_0^{(1)}, l_0^{(1)}\right], \dots, \left[s_0^{(t_0)}, l_0^{(t_0)}\right]\right\}\right) \\ &= \delta_{t_0}\left(\left|\left\{l_0^{(1)}, \dots, l_0^{(t_0)}\right\}\right|\right) \mathbf{1}_{\mathbb{L}_0}\left(\left\{l_0^{(1)}, \dots, l_0^{(t_0)}\right\}\right) \\ &\quad \times \prod_{l \in \mathbb{L}_0} (1 - r^{(l)}) \prod_{i=1}^{t_0} \frac{r^{(l_0^{(i)})} p^{(l_0^{(i)})}(s_0^{(i)})}{\left(1 - r^{(l_0^{(i)})}\right)} \end{aligned} \quad (4.2)$$

where the term $\delta_{t_0}\left(\left|\left\{l_0^{(1)}, \dots, l_0^{(t_0)}\right\}\right|\right)$ ensures that $f(\mathbf{x}_0)$ is nonzero only if the labels are distinct. A special case of the multi-Bernoulli RFS is when the initial number of targets is known. In this case, $r^{(l)} = 1$ for every $l \in \mathbb{L}_0$, and thus (4.2) becomes

$$f(\mathbf{x}_0) = \delta_{\mathbb{L}_0}\left(\left\{l_0^{(1)}, \dots, l_0^{(t_0)}\right\}\right) \prod_{i=1}^{t_0} p^{(l_0^{(i)})}(s_0^{(i)}) \quad (4.3)$$

where $\delta_{\mathbb{L}_0}$ is a multi-target Dirac delta density (see (2.52)).

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING

4.2.3 The likelihood function

Let $\mathbf{S}_k = \{S_k^{(1)}, \dots, S_k^{(T_k)}\}$ correspond to the unlabelled multi-target state. We assume that the observations, conditioned on the rest of the state, are not affected by labels, i.e.,

$$f(\mathbf{z}_k | \mathbf{x}_k) = f(\mathbf{z}_k | \mathbf{s}_k) \quad (4.4)$$

where formulas for $f(\mathbf{z}_k | \mathbf{s}_k)$ can be found in [Mahler, 2007, Chapter 12].

This assumption conforms to the discussion in Section 4.2.1 that labels do not have any physical meaning when considered at a single time step. If some (partial) information is observable associated with the target identity (non-kinematic data such as the target's identification friend-or-foe (IFF) code, which may appear in messages sent by an aircraft's transponder), one can simply include such information in the location S_k , rather than in the label component L_k , in order to satisfy (4.4). Note that in this case, the labelling problem only becomes easier. See for example Appendix B.4, that shows how mixed labelling may not appear at all, if the single-target states contain certain non-kinematic quantities that we are able to estimate.

4.2.4 The state transition density

We will present formulas for the state transition density $f(\mathbf{x}_k | \mathbf{x}_{k-1})$ for three different cases, based on the guidelines in [Mahler, 2007, Chapter 13]. In each case, we need to ensure that $f(\mathbf{x}_k | \mathbf{x}_{k-1})$ is a proper labelled RFS density, i.e. that it fulfills condition (4.1), or in other words, that the density is nonzero only if the labels of \mathbf{x}_k are distinct. In practice, we will already have distinct labels in \mathbf{x}_{k-1} (from recursion). Hence it is sufficient to ensure that $f(\mathbf{x}_k | \mathbf{x}_{k-1})$ is zero when the labels of \mathbf{x}_k are not distinct.

4.2.4.1 No target births or deaths

Let $p\left(s_k^{(i)} \middle| s_{k-1}^{(j)}\right)$ be the single-target state transition density, i.e. the motion model that describes the transition from location $s_{k-1}^{(j)}$ to $s_k^{(i)}$. Assuming that single-target dynamics are decoupled, i.e., $f(\mathbf{x}_k | \mathbf{x}_{k-1})$ can be factorized into

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING

single-target densities, from [Mahler, 2007, Section 13.2.2], we have

$$\begin{aligned} f(\mathbf{x}_k | \mathbf{x}_{k-1}) &= f\left(\left\{x_k^{(1)}, \dots, x_k^{(t_k)}\right\} \middle| \left\{x_{k-1}^{(1)}, \dots, x_{k-1}^{(t_{k-1})}\right\}\right) \\ &= \sum_{\theta \in \Theta_{t_k}} \prod_{i=1}^{t_k} p\left(x_k^{(i)} \middle| x_{k-1}^{(\theta(i))}\right) \end{aligned} \quad (4.5)$$

where Θ_{t_k} is the set of all permutations on $(1, \dots, t_k)$. Note that with no target births or deaths in the model, $t_k = t_{k-1}$, and

$$p\left(s_k^{(i)}, l_k^{(i)} \middle| s_{k-1}^{(j)}, l_{k-1}^{(j)}\right) = p\left(s_k^{(i)} \middle| s_{k-1}^{(j)}\right) \delta_{l_k^{(i)} l_{k-1}^{(j)}} \quad (4.6)$$

since a target cannot change its label. Therefore, from (4.5), we obtain

$$f(\mathbf{x}_k | \mathbf{x}_{k-1}) = \sum_{\theta \in \Theta_{t_k}} \prod_{i=1}^{t_k} p\left(s_k^{(i)} \middle| s_{k-1}^{(\theta(i))}\right) \delta_{l_k^{(i)} l_{k-1}^{(\theta(i))}}. \quad (4.7)$$

Observe that since the labels of \mathbf{x}_{k-1} are distinct, $f(\mathbf{x}_k | \mathbf{x}_{k-1})$ is nonzero only if the labels of \mathbf{x}_k are also distinct. Otherwise, if, for example, $l_k^{(1)} = l_k^{(2)}$, then for every θ , either $l_k^{(1)} \neq l_{k-1}^{(\theta(1))}$ or $l_k^{(2)} \neq l_{k-1}^{(\theta(2))}$, because the labels $l_{k-1}^{(i)}$ are all distinct.

4.2.4.2 With target deaths, no target births

According to [Mahler, 2007, Section 13.2.2], the general formula for $f(\mathbf{x}_k | \mathbf{x}_{k-1})$ in the case when targets may die but no new targets may appear, is given by

$$f(\mathbf{x}_k | \mathbf{x}_{k-1}) = \prod_{j=1}^{t_{k-1}} \left(1 - P_S\left(x_{k-1}^{(j)}\right)\right) \sum_{\theta \in \Theta_{t_k t_{k-1}}} \prod_{i=1}^{t_k} \frac{P_S\left(x_{k-1}^{(\theta(i))}\right) p\left(x_k^{(i)} \middle| x_{k-1}^{(\theta(i))}\right)}{1 - P_S\left(x_{k-1}^{(\theta(i))}\right)} \quad (4.8)$$

where $P_S\left(x_{k-1}^{(i)}\right)$ denotes the survival probability, i.e. the probability that a target survives from time step $k-1$ to time k , and $\Theta_{t_k t_{k-1}}$ is the set of all mappings $\theta: \{1, \dots, t_k\} \rightarrow \{1, \dots, t_{k-1}\}$, subject to $\theta(i) \neq \theta(j)$ for $i \neq j$.

It is reasonable to assume that, due to the lack of physical interpretation of labels for a single time step (see Section 4.2.1), the survival probability does not

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING

depend on the label, i.e. $P_S(x_{k-1}^{(i)}) = P_S(s_{k-1}^{(i)})$. It then follows from (4.6) that

$$f(\mathbf{x}_k | \mathbf{x}_{k-1}) = \prod_{j=1}^{t_{k-1}} \left(1 - P_S(s_{k-1}^{(j)})\right) \sum_{\theta \in \Theta_{t_k t_{k-1}}} \prod_{i=1}^{t_k} \frac{P_S(s_{k-1}^{(\theta(i))}) p(s_k^{(i)} | s_{k-1}^{(\theta(i))}) \delta_{l_k^{(i)} l_{k-1}^{(\theta(i))}}}{1 - P_S(s_{k-1}^{(\theta(i))})}. \quad (4.9)$$

By the same line of argument of Section 4.2.4.1, since the labels of \mathbf{x}_{k-1} are distinct, (4.9) implies that the labels of \mathbf{x}_k must also be distinct.

4.2.4.3 With target births and deaths

In order to discuss the case where new targets may appear, let us recall our discussion in Section 4.2.1. We have described the label as an imaginary stamp attributed to a particular target initial distribution. Similarly, we can assign arbitrarily chosen labels to the distributions of the appearing targets, i.e. the initial distributions of targets that appear after time 0. This can be accomplished if we assume, similar to what we have done in Section 4.2.2, that the initial distributions of the appearing targets are also mutually distinct (in order to make the Bayesian labelling problem well-posed), and for convenience, that they are also mutually independent and independent from the states of existing targets.

Let again $p(s_k^{(i)} | s_{k-1}^{(j)})$ denote the single-target state transition density and $P_S(s_{k-1}^{(i)})$ the survival probability. Let us assume that the targets appearing at time k have distinct labels, contained in a set \mathbb{L}_k , subject to

$$\mathbb{L}_k \cap (\mathbb{L}_0 \cup \dots \cup \mathbb{L}_{k-1}) = \emptyset, \quad (4.10)$$

with each label $l \in \mathbb{L}_k$ being associated with a new target with (location) initial distribution $p^{(l)}(\cdot)$ and probability of appearance $r^{(l)}$.

With the aforementioned independence assumptions, $f(\mathbf{x}_k | \mathbf{x}_{k-1})$ is given by (see [Mahler, 2007, Appendix G.18])

$$f(\mathbf{x}_k | \mathbf{x}_{k-1}) = \sum_{\mathbf{x}_k^* \subseteq \mathbf{x}_k} f_{\text{nb}}(\mathbf{x}_k^* | \mathbf{x}_{k-1}) f(\mathbf{x}_k \setminus \mathbf{x}_k^* | \emptyset) \quad (4.11)$$

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING

where $f_{\text{nb}}(\mathbf{x}_k^* | \mathbf{x}_{k-1})$ is the state transition density $f(\mathbf{x}_k^* | \mathbf{x}_{k-1})$ with the additional assumption of no possibility of target births (with its expression being thus given by (4.9)) and $f(\mathbf{x}_k \setminus \mathbf{x}_k^* | \emptyset)$ represents the distribution of the newly appearing targets at time k . Since the assumptions for the distributions of appearing targets are the same as the assumptions for the initial distributions at time 0, we can use an expression analogous to (4.2) for $f(\mathbf{x}_k \setminus \mathbf{x}_k^* | \emptyset)$, i.e.

$$\begin{aligned}
& f(\mathbf{x}_k \setminus \mathbf{x}_k^* | \emptyset) \\
&= f\left(\left\{\left[s_k^{(1)}, l_k^{(1)}\right], \dots, \left[s_k^{(b_k)}, l_k^{(b_k)}\right]\right\}\right) \\
&= \delta_{b_k}\left(\left|\left\{l_k^{(1)}, \dots, l_k^{(b_k)}\right\}\right|\right) \mathbf{1}_{\mathbb{L}_k}\left(\left\{l_k^{(1)}, \dots, l_k^{(b_k)}\right\}\right) \\
&\quad \times \prod_{l \in \mathbb{L}_k} (1 - r^{(l)}) \prod_{i=1}^{b_k} \frac{r^{(l_k^{(i)})} p^{(l_k^{(i)})} \left(s_k^{(i)}\right)}{\left(1 - r^{(l_k^{(i)})}\right)} \tag{4.12}
\end{aligned}$$

where $b_k = |\mathbf{x}_k \setminus \mathbf{x}_k^*|$. Now, let \mathbf{I}_k^* be the set of labels of \mathbf{x}_k^* , and $\bar{\mathbf{I}}_k^*$ be the set of labels of $\mathbf{x}_k \setminus \mathbf{x}_k^*$. Substituting (4.12) and (4.9) into (4.11), we obtain

$$\begin{aligned}
& f(\mathbf{x}_k | \mathbf{x}_{k-1}) \\
&= \sum_{\mathbf{x}_k^* \subseteq \mathbf{x}_k} \left(\delta_{|\bar{\Phi}|} \left(|\bar{\mathbf{I}}_k^*| \right) \mathbf{1}_{\mathbb{L}_k} \left(\bar{\mathbf{I}}_k^* \right) \prod_{l \in \mathbb{L}_k} (1 - r^{(l)}) \prod_{m \in \bar{\Phi}} \frac{r^{(l_k^{(m)})} p^{(l_k^{(m)})} \left(s_k^{(m)}\right)}{\left(1 - r^{(l_k^{(m)})}\right)} \right. \\
&\quad \times \left. \prod_{j=1}^{t_{k-1}} \left(1 - P_S \left(s_{k-1}^{(j)}\right)\right) \sum_{\theta \in \Theta_{\Phi t_{k-1}}} \prod_{n \in \Phi} \frac{P_S \left(s_{k-1}^{(\theta(n))}\right) p \left(s_k^{(n)} \mid s_{k-1}^{(\theta(n))}\right) \delta_{l_k^{(n)} l_{k-1}^{(\theta(n))}}{1 - P_S \left(s_{k-1}^{(\theta(n))}\right)} \right) \tag{4.13}
\end{aligned}$$

where

$$\begin{aligned}
\Phi &= \left\{ i \mid i \in \{1, \dots, t_k\}, x_k^{(i)} \in \mathbf{x}_k^* \right\} \\
\bar{\Phi} &= \left\{ i \mid i \in \{1, \dots, t_k\}, x_k^{(i)} \in \mathbf{x}_k \setminus \mathbf{x}_k^* \right\}, \\
\Theta_{\Phi t_{k-1}} &= \left\{ \theta : \Phi \rightarrow \{1, \dots, t_{k-1}\} \mid \theta(i) \neq \theta(j) \text{ if } i \neq j \right\}.
\end{aligned}$$

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING

Observe that

1. $f_{\text{nb}}(\mathbf{x}_k^* | \mathbf{x}_{k-1})$ is nonzero only if $\mathbf{I}_k^* \subseteq \mathbb{L}_0 \cup \dots \cup \mathbb{L}_{k-1}$;
2. $f(\mathbf{x}_k \setminus \mathbf{x}_k^* | \emptyset)$ is nonzero only if $\bar{\mathbf{I}}_k^* \subseteq \mathbb{L}_k$;

and from assumption (4.10), we conclude that there is only one choice of $\mathbf{x}_k^* \subseteq \mathbf{x}_k$ that may lead to both $f_{\text{nb}}(\mathbf{x}_k^* | \mathbf{x}_{k-1})$ and $f(\mathbf{x}_k \setminus \mathbf{x}_k^* | \emptyset)$ being nonzero, namely, the choice where all labels of $\mathbf{x}_k \setminus \mathbf{x}_k^*$ are contained in \mathbb{L}_k and all labels of \mathbf{x}_k^* are not. This implies that $f(\mathbf{x}_k | \mathbf{x}_{k-1})$ is nonzero only if $\mathbf{I}_k^* \cap \bar{\mathbf{I}}_k^* = \emptyset$, and since the labels of \mathbf{x}_{k-1} are distinct, it also implies that $f(\mathbf{x}_k | \mathbf{x}_{k-1})$ is nonzero only if all labels of \mathbf{x}_k are distinct.

Remark 4.2.1 *Our discussion in Section 4.2.1 about “well-posed” and “ill-posed” Bayesian labelling problems in terms of initial targets (priors) applies to the newly appearing targets as well. The problem becomes ill-posed unless we assume that the appearing target distributions are reasonably distinct (either via time or location). Even when we assume that at most one target can appear during each time interval $[k - 1, k]$, the labelling problem may still be ill-posed if appearing target distributions at consecutive time intervals are similar and there is uncertainty on determining which time interval a target has been born (for instance, in observations models with the possibility of missed detections).*

4.3 Statistics for Bayesian MTTL

To handle the Bayesian MTTL problem, we need more than just calculating the labelled multi-target posterior density $f(\mathbf{x}_k | Z^k)$. We need to extract practical information from this posterior density, that allows us, for instance, to answer **Questions 1 and 2** from our motivational example in Section 4.1. This practical information, i.e. useful statistics for the MTTL problem, will be described in this section.

In Bayesian MTTL, extracting practical information from the posterior is not straightforward due to “mixed labelling”, a situation where there is ambiguity in the assignment of labels to locations. The occurrence of mixed labelling

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING

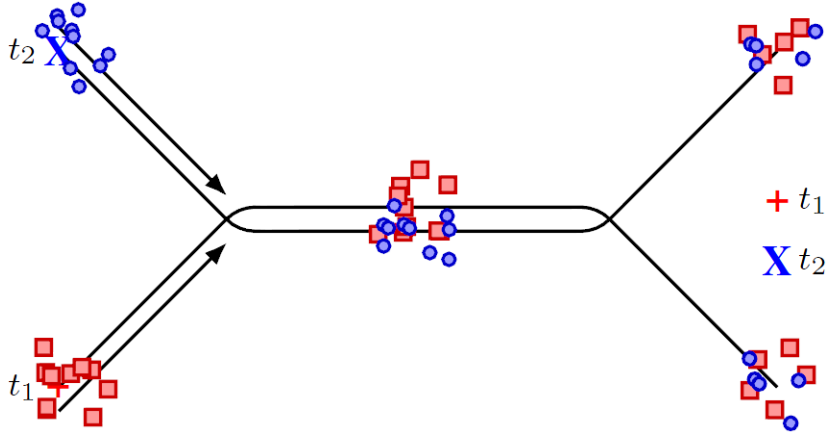


Figure 4.3: Particle representation of the multi-target distribution in a situation where mixed labelling occurs (source: Crouse et al. [2011a]). The squares and circles mark the possible locations of each target in terms of particles

when targets separate after moving in close proximity to each other has been observed empirically (as e.g. in Boers et al. [2010]) or theoretically by analyzing the Bayesian recursion (see Appendix B.2). When the multi-target Bayes recursion is implemented by a particle filter (i.e. the M-SMC filter described in Section 2.3.4), mixed labelling manifests itself by particle clouds corresponding to each target intersecting each other, as shown in Fig. 4.3.

4.3.1 The labelling probability

We are now ready to propose a *mathematical formulation* for **Question 2** of Section 4.1, i.e. for the probability of labelling error.

Definition 4.3.1 Consider a RFS \mathbf{X} described as in Section 4.2. We define the **labelling probability** associated with a realization of \mathbf{X} (a finite set of labelled target states $\mathbf{x} = \{[s^{(1)}, l^{(1)}], \dots, [s^{(t)}, l^{(t)}]\}$) as

$$p_l(\mathbf{x}|\mathbf{s}) \triangleq \frac{f(\mathbf{x})}{f(\mathbf{s})} \quad (4.14)$$

where $\mathbf{s} = \{s^{(1)}, \dots, s^{(t)}\}$. Note that, if the denominator is zero, the numerator will necessarily be zero as well. As a result, since realizations of \mathbf{X} can only come

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING

from the support points of the RFS, then the denominator is always positive. ■

To put simply, the labelling probability is the ratio between the a “labelled” multi-target density and its corresponding “unlabelled” multi-target density. In order to understand what the labelling probability means from a physical point of view, we use relationships (2.67) and (2.68), which combined and applied to our problem of interest, lead to

$$\begin{aligned}
 & P(X^{(1)} \in A^{(1)}, \dots, X^{(t)} \in A^{(t)} | T = t) \\
 &= \sum_{l^{(1)} \in A_L^{(1)}} \dots \sum_{l^{(t)} \in A_L^{(t)}} \int_{A_S^{(t)} \times \dots \times A_S^{(1)}} \frac{1}{t!P(T = t)} f(\{x^{(1)}, \dots, x^{(t)}\}) ds^{(1)} \dots ds^{(t)}
 \end{aligned} \tag{4.15}$$

where for $i = 1, \dots, t$, $A^{(i)} = A_S^{(i)} \times A_L^{(i)}$ (where $A_S^{(i)} \subseteq \mathbb{R}^n$ and $A_L^{(i)} \subseteq \Pi$). The derivative of $P(X^{(1)} \in A^{(1)}, \dots, X^{(t)} \in A^{(t)} | T = t)$ w.r.t. the product measure $(ds^{(1)} \times \mu) \times \dots \times (ds^{(t)} \times \mu)$ (where μ denotes the counting measure) is a conditional probability density¹, given by

$$p(x^{(1)}, \dots, x^{(t)} | t) = \frac{1}{t!P(T = t)} f(\{x^{(1)}, \dots, x^{(t)}\}). \tag{4.16}$$

Observe now that from (4.14) and (4.16), we get

$$\begin{aligned}
 p_l(\{x^{(1)}, \dots, x^{(t)}\} | \{s^{(1)}, \dots, s^{(t)}\}) &= \frac{p(x^{(1)}, \dots, x^{(t)} | t)}{p(s^{(1)}, \dots, s^{(t)} | t)} \\
 &= \frac{p(s^{(1)}, l^{(1)}, \dots, s^{(t)}, l^{(t)} | t)}{p(s^{(1)}, \dots, s^{(t)} | t)} \\
 &= p(l^{(1)}, \dots, l^{(t)} | s^{(1)}, \dots, s^{(t)}, t). \tag{4.17}
 \end{aligned}$$

At first glance, eq. (4.17) may seem to have an inconsistency, as its left-hand side involves a set valued variable wherein the order of the elements does not play any role, whereas the right-hand side involves vector variables wherein the order of the elements is important. A closer look, however, reveals that in (4.17), if we

¹Here we consider probability densities in general sense, when the corresponding probability function may be differentiable w.r.t. a measure other than the Lebesgue measure.

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING

write $\{x^{(1)}, \dots, x^{(t)}\}$ in a different order, say as $\{x^{(\theta(1))}, \dots, x^{(\theta(t))}\}$ (where θ is a permutation map), then the orders of the $[l^{(1)}, \dots, l^{(t)}]$ and $[s^{(1)}, \dots, s^{(t)}]$ vectors will also change, but without affecting the association between the indexes of these two vectors. We can then rewrite (4.17) as

$$\begin{aligned} p_l(\{x^{(1)}, \dots, x^{(t)}\} | \{s^{(1)}, \dots, s^{(t)}\}) \\ = p(l^{(\theta(1))}, \dots, l^{(\theta(t))} | s^{(\theta(1))}, \dots, s^{(\theta(t))}, t) \end{aligned} \quad (4.18)$$

where θ can be any permutation of $(1, \dots, t)$. This leads to a practical interpretation of the labelling probability: it corresponds to the (conditional) joint probability mass of the set of labels $\{l^{(1)}, \dots, l^{(t)}\}$ and the assignment of these labels to a given set of target locations $\{s^{(1)}, \dots, s^{(t)}\}$ that results in $\{x^{(1)}, \dots, x^{(t)}\}$. The conditioning on $T = t$ is irrelevant since the given set of locations $\{s^{(1)}, \dots, s^{(t)}\}$ already determines the cardinality.

For Bayesian labelling purposes, we are interested in the posterior version of the labelling probability, i.e. considering a time-varying multi-target state conditioned on all observations up to and including time k , given by

$$p_l(\mathbf{x}_k | \mathbf{s}_k, Z^k) = \frac{f(\mathbf{x}_k | Z^k)}{f(\mathbf{s}_k | Z^k)} \quad (4.19)$$

which, as a conditional probability mass, has the property

$$\sum_{\mathbf{x}_k \in \Pi_k(\mathbf{s}_k)} p_l(\mathbf{x}_k | \mathbf{s}_k, Z^k) = 1 \quad (4.20)$$

where

$$\Pi_k\left(\left\{s_k^{(1)}, \dots, s_k^{(t_k)}\right\}\right) \triangleq \left\{ \mathbf{x}_k \mid \mathbf{x}_k = \left\{ \begin{bmatrix} s_k^{(1)} \\ l_k^{(1)} \end{bmatrix}, \dots, \begin{bmatrix} s_k^{(t_k)} \\ l_k^{(t_k)} \end{bmatrix} \right\}, f(\mathbf{x}_k | Z^k) > 0 \right\}. \quad (4.21)$$

The labelled multi-target posterior $f(\mathbf{x}_k | Z^k)$ can be calculated using the formulas described in Section 4.2. Formulas to calculate the *unlabelled* multi-target posterior $f(\mathbf{s}_k | Z^k)$ can be found in [Mahler, 2007, Chapters 12–14].

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING

We can readily use (4.19) to mathematically formulate **Question 2** of Section 4.1. For a set of *labelled tracks* $\hat{\mathbf{x}}_k = \{\hat{x}_k^{(1)}, \dots, \hat{x}_k^{(t)}\}$ and the corresponding *unlabelled tracks* $\hat{\mathbf{s}}_k = \{\hat{s}_k^{(1)}, \dots, \hat{s}_k^{(t)}\}$, the *probability of labelling error* can be described by $1 - p_1(\hat{\mathbf{x}}_k | \hat{\mathbf{s}}_k)$. This corresponds to the posterior probability of error in the assignment of labels to tracks (which also includes errors in the choice of the set of labels, in case target birth and/or deaths are allowed), given the set of locations $\hat{\mathbf{s}}_k$.

If we need to quantify the “amount of mixed labelling” associated with a set of locations \mathbf{s}_k , we may use the Shannon entropy of the labelling probability, given by

$$- \sum_{\mathbf{x}_k \in \Pi(\mathbf{s}_k)} p_1(\mathbf{x}_k | \mathbf{s}_k, Z^k) \log p_1(\mathbf{x}_k | \mathbf{s}_k, Z^k) \quad (4.22)$$

since the Shannon entropy, in Bayesian context, is an uncertainty measure for a posterior distribution. The situation where all labelling probabilities for a given \mathbf{s}_k are equal (and thus entropy is maximized) would correspond to the situation of “total mixed labelling”, where there is total ambiguity in the assignment of labels to locations. The situation where one of the labelling probabilities has a value of one (and thus entropy is minimum and has a value of zero) corresponds to the situation of “no mixed labelling”, where there is only one reasonable assignment.

4.3.2 Track extraction methods for Bayesian MTTL

We are now ready to propose some conceptual track extraction schemes especially suited for the Bayesian MTTL problem. These schemes also give a proper formulation to **Question 1** proposed in Section 4.1, i.e. they give a formal interpretation to the notion of “optimal assignment of labels”.

4.3.2.1 The MMOSPA-MLP estimate

Let $\mathbf{S}_k = \{S_k^{(1)}, \dots, S_k^{(t_k)}\}$ denote the RFS corresponding to the locations. In the MMOSPA-MLP scheme, the set of labelled tracks $\hat{\mathbf{x}}_k$ is the solution of the

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING

problem given by

$$\hat{\mathbf{s}}_k = \arg \inf_{\bar{\mathbf{s}}_k} \int (\epsilon_p^{(c)}(\mathbf{s}_k, \bar{\mathbf{s}}_k))^p f(\mathbf{s}_k | Z^k) \delta \mathbf{s}_k \quad (4.23)$$

$$\hat{\mathbf{x}}_k = \arg \max_{\mathbf{x}_k} p_1(\mathbf{x}_k | \hat{\mathbf{s}}_k, Z^k) \quad (4.24)$$

where $\epsilon_p^{(c)}$ is the OSPA metric given by (2.61) and (2.62). The rationale of the estimate given by (4.23)–(4.24) is quite simple. In the first step (4.23), that we call MMOSPA step, we obtain the unlabelled tracks, according to the MMOSPA estimate given by (2.63). That corresponds, hence, to an optimal choice (in Mean OSPA sense) of unlabelled tracks.

In the second step (4.24), that we refer as MLP (Maximum Labelling Probability) step, the labelled tracks are obtained by using the previously obtained MMOSPA estimate and choosing the assignment of labels that maximizes the labelling probability according to Definition 4.3.1. The MLP step also gives a mathematical formulation to **Question 1** proposed in Section 4.1; i.e. we treat the “optimal assignment of labels” as the assignment of labels that gives the maximum labelling probability.

4.3.2.2 The JoM-MLP estimate

For the purpose of obtaining the unlabelled tracks, an alternative to the MMOSPA estimate is the JoM estimate given by (2.59). In this case, the set of unlabelled tracks is obtained by

$$\hat{\mathbf{s}}_k = \arg \sup_{\{s_k^{(1)}, \dots, s_k^{(t_k)}\}} \frac{c^{t_k}}{t_k!} f\left(\left\{s_k^{(1)}, \dots, s_k^{(t_k)}\right\} \middle| Z^k\right) \quad (4.25)$$

and from $\hat{\mathbf{s}}_k$, the set of labelled tracks can be then straightforwardly obtained using the MLP step (4.24), such that we may refer to the combined estimate as JoM-MLP estimate.

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING

4.3.2.3 Pure JoM estimate

Instead of first obtaining the set of unlabelled tracks, and then using the result to obtain a set of labelled tracks, one may ask: why not simply obtain a set of labelled tracks in a single step?

This can be accomplished if we apply the JoM estimate directly to $f(\mathbf{x}_k | Z^k)$ instead of $f(\mathbf{s}_k | Z^k)$. The set of labelled tracks is then obtained by

$$\hat{\mathbf{x}}_k = \arg \sup_{\{x_k^{(1)}, \dots, x_k^{(t_k)}\}} \frac{c^{t_k}}{t_k!} f\left(\left\{x_k^{(1)}, \dots, x_k^{(t_k)}\right\} \middle| Z^k\right). \quad (4.26)$$

The solution given by (4.26), however, does not precisely give a mathematical formulation for **Question 1** of Section 4.1, since it consists of simultaneously obtaining the tracks and their labels, rather than assigning labels to tracks. For the scenario in Fig. 4.1, it gives instead a mathematical formulation to the question: *Where (in the entire surveillance area) is the track with label T1 and the track with label T2?*

4.3.2.4 Pure MMOSPA estimate

Similarly to the “pure JoM” estimate, we can propose a “pure MMOSPA” estimate, that consists of obtaining a set of labelled tracks in a single step. But the OSPA metric given by (2.61) and (2.62) cannot be applied to the labelled multi-target state \mathbf{X}_k , only to the unlabelled multi-target state \mathbf{S}_k , as it requires that the single-target states assume values in an Euclidean space. We then require a modified version of the OSPA metric that can account for labelling errors. Such metric was presented by Ristic et al. [2011b], with its expression being almost identical to (2.61) and (2.62), with only $d^{(c)}(x^{(j)}, \bar{x}^{(\pi(j))})$ being replaced by $d_{p'}^{(c, \alpha)}(x^{(j)}, \bar{x}^{(\pi(j))})$, where

$$d_{p'}^{(c, \alpha)}\left(\begin{bmatrix} s^{(1)} \\ l^{(1)} \end{bmatrix}, \begin{bmatrix} s^{(2)} \\ l^{(2)} \end{bmatrix}\right) \triangleq \min\left(\|s^{(1)} - s^{(2)}\|_{p'}^{p'} + \alpha^{p'} \bar{\delta}_{l^{(1)}}(l^{(2)}), c\right) \quad (4.27)$$

where $p' \geq 1$, $\alpha \in [0, c]$ and $\bar{\delta}$ denotes the complement of the Kronecker delta, i.e. $\bar{\delta}_{l^{(1)}}(l^{(2)}) = 0$ if $l^{(1)} = l^{(2)}$, and $\bar{\delta}_{l^{(1)}}(l^{(2)}) = 1$ otherwise. The parameter p'

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING

emphasizes the Euclidean distance between the two objects, whereas α emphasizes their difference in labelling. Finally, the pure MMOSPA estimate $\hat{\mathbf{x}}_k$ is obtained by

$$\hat{\mathbf{x}}_k = \arg \inf_{\bar{\mathbf{x}}_k} \int \left(\epsilon_{p,p'}^{(c,\alpha)}(\mathbf{x}_k, \bar{\mathbf{x}}_k) \right)^p f(\mathbf{x}_k | Z^k) \delta \mathbf{x}_k \quad (4.28)$$

where $\epsilon_{p,p'}^{(c,\alpha)}$ is the modified OSPA metric.

4.3.2.5 Which track extraction method to use?

Each track extraction method has a distinct emphasis, making it difficult to declare one as the “best” one. Provided below are some reasonable guidelines:

1. If accurate location/cardinality estimation is far more important than accurate labelling, then it is logical to use the MMOSPA-MLP or the JoM-MLP as they obviously give priority to optimal location estimation. If they have similar importance, the pure JoM seems more adequate. If one would like to fine-tune the relative importance of location estimation and labelling, this is possible using the pure MMOSPA;
2. If in the single-target tracking case, the MMSE would be a better choice than the MAP, then it makes sense to use the MMOSPA-MLP or pure MMOSPA schemes (which in the single-target case, are equivalent to the MMSE for $p = 2$). On the other hand, if the MAP would be a better choice than the MMSE (due to truncation, multi-modality, etc.), the JoM-MLP or pure JoM make more sense as they are equivalent to the MAP in the single-target case.

4.3.3 Performance metrics for MTTL

For a given MTTL algorithm (Bayesian or not), if we wish to measure the performance of tracking (in the sense of estimating locations) together with labelling, we may use the modified OSPA metric described in Section 4.3.2.4. Sometimes, however, we may wish to evaluate the labelling performance of an algorithm separately from its tracking performance.

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING

This is certainly possible if the MTTL algorithm obtains the unlabelled tracks at one step and assigns the labels at another step via some labelling procedure. For a Bayesian MTTL algorithm, that would be the case if the MMOSPA-MLP or JoM-MLP track extraction schemes are used (but not the pure JoM or pure MMOSPA).

Assuming that the ground truth trajectories of targets are known, the first step is to assign labels to them consistently with the way we assigned labels to the initial target distributions (see Section 4.2.2) and to the appearing target distributions (see Section 4.2.4.3). Let \mathbf{s}_k and \mathbf{x}_k correspond to the unlabelled and labelled ground truth multi-target states at time k . We may then, solely for performance evaluation purposes, apply the evaluated algorithm's labelling procedure to the ground truth \mathbf{s}_k to obtain $\hat{\mathbf{x}}_k$, rather than to the set of unlabelled tracks $\hat{\mathbf{s}}_k$. In order to evaluate $\hat{\mathbf{x}}_k$, we define the following simple hit-or-miss metric:

$$\begin{aligned} \epsilon(\hat{\mathbf{x}}_k, \mathbf{x}_k) &= \epsilon \left(\left\{ \left[\hat{s}_k^{(1)}, \hat{l}_k^{(1)} \right], \dots, \left[\hat{s}_k^{(t_k)}, \hat{l}_k^{(t_k)} \right] \right\}, \left\{ \left[s_k^{(1)}, l_k^{(1)} \right], \dots, \left[s_k^{(t_k)}, l_k^{(t_k)} \right] \right\} \right) \\ &\triangleq \begin{cases} 1, & \exists i, j \text{ s.t. } \hat{s}_k^{(i)} = s_k^{(j)}, \hat{l}_k^{(i)} \neq l_k^{(j)} \\ 0, & \text{otherwise} \end{cases} \end{aligned} \quad (4.29)$$

i.e. the value of the metric is 1 if there is at least one incorrectly assigned label, and 0 otherwise. Naturally, the metric is only statistically relevant if averaged over a sufficient number of Monte Carlo runs.

This simple hit-or-miss metric may, however, not be appropriate if there is mixed labelling involving a large number of targets, such that the chance of all labels being correctly assigned is very small. For these situations, we propose the following ‘‘per-target’’ hit-or-miss metric:

$$\epsilon(\hat{\mathbf{x}}_k, \mathbf{x}_k) \triangleq \frac{1}{t_k} \left| \left\{ i \in \{1, \dots, t_k\} \mid \exists j \text{ s.t. } \hat{s}_k^{(i)} = s_k^{(j)}, \hat{l}_k^{(i)} \neq l_k^{(j)} \right\} \right| \quad (4.30)$$

i.e. the value of the metric is equal to the ratio of incorrect label-to-location assignments.

4.4 Calculating labelling probabilities for existing MTT algorithms

In this section we look into the calculation of the labelling probabilities described in Section 4.3.1 for two existing MTT algorithms: the M-SMC filter described in Section 2.3.4, and the MHT algorithm (Reid [1979]). This corresponds to answering **Question 2** proposed in Section 4.1, and using the MLP step (4.24) to perform label assignment, it also corresponds to answering **Question 1**.

Calculation of the MMOSPA and JoM steps (necessary for the MMOSPA-MLP and JoM-MLP estimates), and of the “Pure MMOSPA” and the “Pure JoM” estimates, is not discussed in this work. They can, in principle, be computed using optimization methods combined with numerical approximation of the quantities to be maximized/minimized. For the MMOSPA estimate with assumption of no target births/deaths, some efficient calculation methods have been presented in Crouse et al. [2011a,b].

4.4.1 Multi-target Sequential Monte Carlo (M-SMC) filter

By applying the M-SMC filter from Section 2.3.4 to the Bayesian MTTL problem described in Section 2.3.4, we are, in principle, able to compute the labelling probability and other statistics described in Section 4.3.

In this algorithm, the multi-target density $f(\mathbf{x}_k | Z^k)$ is represented by a set of particles $\{\mathbf{x}_k(i), w_k(i)\}_{i=1}^{N_P}$, where $\mathbf{x}_k(i)$ denotes a realization of multi-target state, $w_k(i)$ the particle weight, and N_P the number of particles.

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING

Note that, from Definition 4.3.1 and given assumption (4.4), we have

$$\begin{aligned}
 p_l(\mathbf{x}_k | \mathbf{s}_k, Z^k) &= \frac{f(\mathbf{x}_k | Z^k)}{f(\mathbf{s}_k | Z^k)} \\
 &= \frac{f(\mathbf{z}_k | \mathbf{s}_k) f(\mathbf{x}_k | Z^{k-1})}{f(\mathbf{z}_k | Z^{k-1}) f(\mathbf{s}_k | Z^k)} \\
 &= \frac{f(\mathbf{x}_k | Z^{k-1})}{f(\mathbf{s}_k | Z^{k-1})} \\
 &= \frac{\int f(\mathbf{x}_k | \mathbf{x}_{k-1}) f(\mathbf{x}_{k-1} | Z^{k-1}) \delta \mathbf{x}_{k-1}}{f(\mathbf{s}_k | Z^{k-1})} \tag{4.31}
 \end{aligned}$$

and hence, the labelling probabilities can then be calculated by straightforward particle approximation of (4.31), i.e.

$$p_l(\mathbf{x}_k | \mathbf{s}_k, Z^k) \propto \sum_{j=1}^{N_P} w_{k-1}(j) f(\mathbf{x}_k | \mathbf{x}_{k-1}(j)) \tag{4.32}$$

where due to (4.20), the proportionality can be turned into an equality by normalization over all $\mathbf{x}_k \in \Pi_k(\mathbf{s}_k)$.

4.4.2 Multiple Hypothesis Tracking (MHT)

In the MHT algorithm (Reid [1979]), the multi-target density $f(\mathbf{x}_k | Z^k)$ is represented by a set of hypotheses $\{\mathbf{h}_k(i), w_k(i)\}_{i=1}^{N_H}$ where $\mathbf{h}_k(i)$ denotes a hypothesis on the multi-target state, $w_k(i)$ the hypothesis weight, and N_H the number of hypotheses. According to [Mahler, 2007, Section 10.5], we can model each hypothesis as $\mathbf{h}_k(i) = \{h_k^{(1)}(i), \dots, h_k^{(t_k(i))}(i)\}$, where the single-target hypothesis $h_k^{(j)}(i)$ is given by a triple:

$$\left(\hat{s}_k^{(j)}(i), l_k^{(j)}(i), P_k^{(j)}(i) \right) \tag{4.33}$$

where $\hat{s}_k^{(j)}(i)$ and $P_k^{(j)}(i)$ are respectively hypotheses on the mean and the covariance of the location $S_k^{(j)}$, and $l_k^{(j)}(i)$ is a hypothesis on the corresponding label.

We can use the following procedure to approximate the labelling probabilities for the MHT. At every time step (starting from $k = 0$), we produce a number

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING

of samples, say N_P , by sampling from the set of hypotheses $\{\mathbf{h}_k(i), w_k(i)\}_{i=1}^{N_H}$. In other words, for samples $m = 1, \dots, N_P$, we do the following:

1. Choose a hypothesis index i_m by sampling from the pmf $(w_k(i))_{i=1}^{N_H}$
2. For $j = 1, \dots, t_k(i_m)$, sample

$$s_k^{(j)}(i_m) \sim \mathcal{N}\left(\hat{s}_k^{(j)}(i_m); P_k^{(j)}(i_m)\right) \quad (4.34)$$

3. Make

$$\mathbf{x}_k(m) = \left\{ \left[s_k^{(1)}(i_m), l_k^{(1)}(i_m) \right], \dots, \left[s_k^{(t_k(i_m))}(i_m), l_k^{(t_k(i_m))}(i_m) \right] \right\} \quad (4.35)$$

As a result of this sampling process, we obtain N_P samples of $f(\mathbf{x}_k | Z^k)$, as if they were same-weight particles produced by a M-SMC filter that approximates $f(\mathbf{x}_k | Z^k)$. Consequently, at the next time step $k + 1$, we can calculate the labelling probabilities as in (4.32).

4.4.3 M-SMC filter, MHT and degeneracy

The M-SMC filter is basically a SIR PF applied to the multi-target tracking problem, and therefore is subject to the degeneracy phenomenon described in Section 2.1.2.4. The question is whether degeneracy has impact on tracking, which would be the case if the ‘‘indifference condition’’ (2.13) is not fulfilled. Using RFS notation, we rewrite (2.13) as

$$E[g(\mathbf{X}_k) | Z^k] \approx E[g(\mathbf{X}_k) | \mathbf{x}_0^*, \dots, \mathbf{x}_j^*, Z^k] \quad (4.36)$$

where $(\mathbf{x}_0^*, \dots, \mathbf{x}_j^*)$ is some hypothesis on the target trajectories until time j , a time where we expect the information about the trajectories $(\mathbf{X}_0, \dots, \mathbf{X}_j)$ to have collapsed into a single particle due to degeneracy.

From our analysis on the persistence of the mixed labelling in Appendix B.3, we know that if mixed labelling existed at some time $j' < k$, it may also be present in $f(\mathbf{x}_k | Z^k)$, regardless of the observations Z^k . However, if we assume labelled multi-target target trajectories $(\mathbf{x}_0^*, \dots, \mathbf{x}_j^*)$, we are also assuming that there is

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING

no mixed labelling until and including time j . As consequence, if mixed labelling happened before time j , i.e. if $j' \leq j$, we may have that $f(\mathbf{x}_k | Z^k)$ contains mixed labelling, but $f(\mathbf{x}_k | \mathbf{x}_0^*, \dots, \mathbf{x}_j^*, Z^k)$ (the statistic effectively approximated by the M-SMC filter) does not. In this case, condition (4.36) is likely going to fail.

Note that this situation can happen no matter how far in the past index j is, meaning that we cannot prevent degeneracy from affecting tracking performance by “slowing it down” as described in Section 2.1.2.4. The M-SMC filter has therefore a tendency of “forgetting” the mixed labelling that exists in the true posterior density, underestimating the probability of labelling errors. This situation has been also empirically observed by Boers et al. [2010].

It is easy to see that the MHT algorithm suffers from a similar problem. In the MHT, each hypothesis on the multi-target state at time k implicitly assumes hypotheses on the multi-target state at past times $0, \dots, k-1$. Since low probability hypotheses are periodically pruned, we have a degeneracy phenomenon similar to the SIR PF degeneracy described in Section 2.1.2.4.

4.4.4 A numerical example

We will show a simple example of the application of the M-SMC filter to the MTTL problem. We consider the scenario shown in Fig. 4.4, where two targets start off separately, approach each other (around time step $k = 23$) and some time after (around $k = 37$) separate again.

The location has the form $S_k^{(i)} = [P_x^{(i)}, P_y^{(i)}, V_x^{(i)}, V_y^{(i)}]$, where x and y denote the Cartesian coordinates, $(P_x^{(i)}, P_y^{(i)})$ corresponds to the position (with realizations denoted by $(p_x^{(i)}, p_y^{(i)})$) and $(V_x^{(i)}, V_y^{(i)})$ corresponds to the velocity (with realizations denoted by $(v_x^{(i)}, v_y^{(i)})$).

The multi-target measurement model $f(\mathbf{z}_k | \mathbf{s}_k)$ corresponds to the detection-type measurement model with no missed detections or false alarms (described in [Mahler, 2007, Section 12.3.4]). The single-measurement, single-target likelihood function is given by

$$p\left(z_k^{(i)} \mid s_k^{(j)}\right) = \mathcal{N}\left(z_k^{(i)}; [p_x^{(j)}, p_y^{(j)}], \begin{bmatrix} 676 & 0 \\ 0 & 676 \end{bmatrix}\right). \quad (4.37)$$

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING

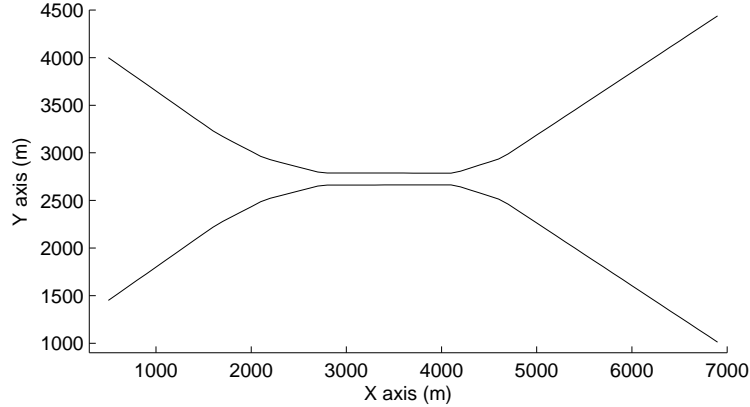


Figure 4.4: Two-target scenario with mixed labelling

We consider that the number of targets is known by the tracker, i.e. that the prior is given by (4.3) and the state transition density by (4.7). The single-target state transition model corresponds to the discretized White Noise acceleration model described in Bar-Shalom et al. [2001], given by

$$p\left(x_k^{(i)} \mid x_{k-1}^{(j)}\right) = \mathcal{N}\left(x_k^{(i)}; \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x_{k-1}^{(j)}, \sigma^2 \begin{bmatrix} T^3/3 & 0 & T^2/2 & 0 \\ 0 & T^3/3 & 0 & T^2/2 \\ T^2/2 & 0 & T & 0 \\ 0 & T^2/2 & 0 & T \end{bmatrix}\right) \quad (4.38)$$

where $T = 2$ is the interval between observations and $\sigma^2 = 676$ is the power spectral density of the process noise. We use $N_P = 20,000$ particles, initiated near the true locations of targets; this high number of particles is chosen to somewhat compensate the degeneracy phenomenon mentioned in Section 4.4.3. We perform blind importance sampling, i.e. we use the state transition density $f(\mathbf{x}_k | \mathbf{x}_{k-1})$ as proposal density for the M-SMC filter.

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING

We perform track extraction using the MMOSPA-MLP scheme described in Section 4.3.2.1, with parameter $p = 2$ (the parameter c is irrelevant since the number of targets is fixed). The MMOSPA estimate (4.23) is calculated using a simple “brute force” optimization (i.e. a search over all particles), with the quantity being minimized approximated by

$$\sum_{i=1}^{N_P} w_k(i) (\epsilon_p^{(c)}(\mathbf{s}_k(i), \bar{\mathbf{s}}_k))^p$$

and the labelling probabilities (also used in the MLP step (4.24)) are calculated using (4.32).

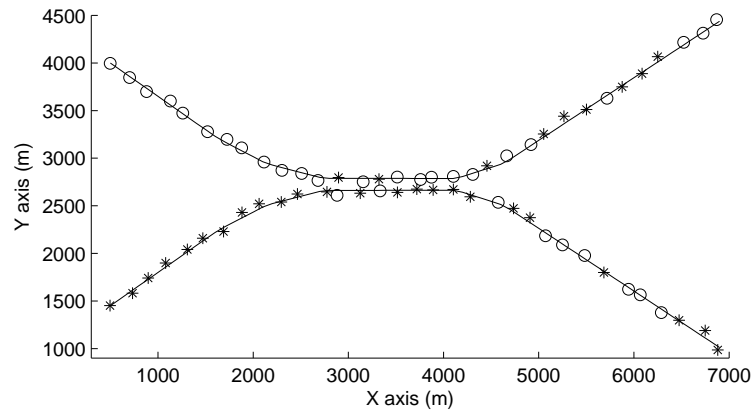
Fig. 4.5(a) shows the MMOSPA-MLP tracks calculated by the filter in a typical run for scenario, and Fig. 4.5(b) shows the calculated labelling probability associated with the tracks at each time step. As expected, the labelling probability decreases as the two targets mutually approach, indicating a partial mixed labelling situation. However, after the targets come closer together ($k = 23$), the behavior of the computed statistic becomes somewhat erratic (Fig. 4.5(b)).

From the theoretical analysis in Appendix B.3, after target separation ($k = 37$), we expect the labelling probability to become approximately constant. Since we cannot see this happening in Fig. 4.5(b), the results produced by the M-SMC filter should be considered questionable. These results suggest that the M-SMC filter, applied to the MTTL problem, is not only impaired by the degeneracy phenomenon as seen in Boers et al. [2010], but has poor performance in general in terms of quantifying the labelling uncertainty. This gives us extra motivation to look for a novel solution to the MTTL problem.

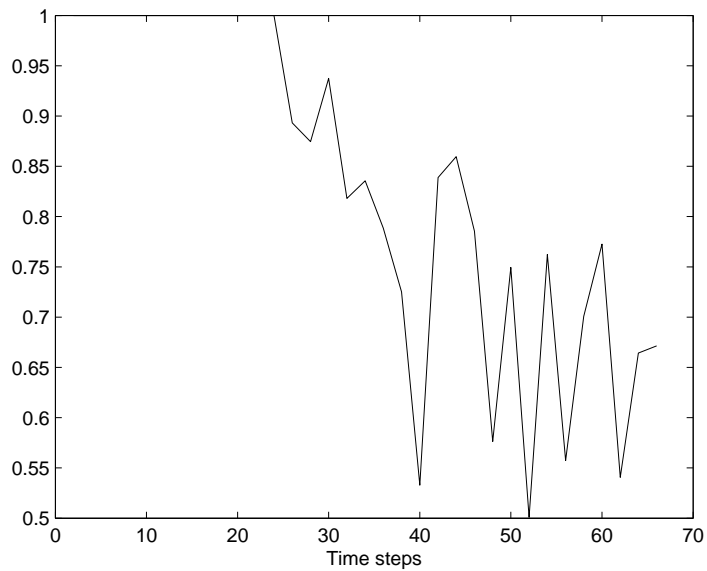
4.5 A novel solution to the MTTL problem: The Labelling Uncertainty-Aware Particle Filter

We are now ready to propose a new algorithm for Bayesian MTTL that counters the degeneracy phenomenon. The algorithm, that we call Labelling Uncertainty-Aware Particle Filter (LUA-PF), is based on the the RBMPF, conceptually described in Section 3.2.3. In Section 4.5.1, we describe the theoretical basis of the

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING



(a) MMOSPA-MLP estimate calculated using M-SMC filter



(b) Labelling probability calculated using M-SMC filter

Figure 4.5: M-SMC filter labelling results

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING

algorithm, which is derived in detail in Section 4.5.2, and summarized in Section 4.5.3. In Section 4.5.4, we discuss computational performance aspects.

4.5.1 Theoretical basis

In Section 4.5.2, we have described the RBMPF technique and explained why it is effective against the SMC degeneracy phenomenon in the context of joint state and parameter estimation. The same technique can be applied to other problems where the SIR PF is impaired by degeneracy. The idea is to estimate part of the state vector (namely, the part that causes the “indifference condition” (2.13) to be violated) using a non-SMC estimator. More precisely, the non-SMC estimator is used to calculate the *conditional* probability distribution of this part of the state, conditioned on the rest of the state and the available observations.

In Chapter 3, the non-SMC estimator is used for the static components of the state vector, i.e. parameters. In the MTTL problem, as we have seen in Section 4.4.3, the indifference condition may be violated whenever the posterior density $f(\mathbf{x}_k | Z^k)$ contains mixed labelling. Hence, an intuitive solution to degeneracy is to apply the RBMPF to the MTTL problem and have the labels $\{L_k^{(1)}, \dots, L_k^{(T_k)}\}$ estimated by a non-SMC estimator. This can be accomplished by recalling, from (4.19), that $f(\mathbf{x}_k | Z^k)$ can be decomposed as

$$f(\mathbf{x}_k | Z^k) = f(\mathbf{s}_k | Z^k) p_l(\mathbf{x}_k | \mathbf{s}_k, Z^k). \quad (4.39)$$

The RBMPF solution to the MTTL problem, which we name LUA-PF, would then consist of using a SMC algorithm to approximate $f(\mathbf{s}_k | Z^k)$ (by a set of particles $\{\mathbf{s}_k(i), w_k(i)\}_{i=1}^{N_P}$), and a non-SMC estimator to calculate the labelling probabilities

$$\begin{aligned} & p_l \left(\left\{ x_k^{(1)}, \dots, x_k^{(t_k)} \right\} \middle| \left\{ s_k^{(1)}, \dots, s_k^{(t_k)} \right\}, Z^k \right) \\ &= p_l \left(\left\{ \left[s_k^{(1)}, l_k^{(1)} \right], \dots, \left[s_k^{(t_k)}, l_k^{(t_k)} \right] \right\} \middle| \left\{ s_k^{(1)}, \dots, s_k^{(t_k)} \right\}, Z^k \right), \end{aligned}$$

which, as explained in Section 4.3.1, correspond to the conditional joint probability mass of the set of labels $\{l_k^{(1)}, \dots, l_k^{(t_k)}\}$ and the assignment of these labels

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING

to a given set of target locations $\{s_k^{(1)}, \dots, s_k^{(t_k)}\}$ that results in $\{x_k^{(1)}, \dots, x_k^{(t_k)}\}$, given also the available observations Z^k . The resulting approximation is then given by

$$f(\mathbf{x}_k | Z^k) \approx \sum_{i=1}^{N_P} w_k(i) \delta_{\mathbf{s}_k(i)}(\mathbf{s}_k) p_l(\mathbf{x}_k | \mathbf{s}_k, Z^k). \quad (4.40)$$

Remark 4.5.1 *The theoretical justification that we have just presented is valid in the assumption that the posterior distribution of the locations, i.e. $f(\mathbf{s}_k | Z^k)$ can be effectively approximated using methods like the M-SMC filter or the MHT. Naturally, this assumption may not hold; for instance, if the location $S_k^{(i)}$ contains parameters, the indifference condition (2.13) will be violated, even without considering labels. Some adjustments to the algorithm (such as estimating $f(\mathbf{s}_k | Z^k)$ using also a RBMPF-like algorithm) are needed in these cases.*

4.5.2 Derivation of the LUA-PF

From (4.40), the expectation of an arbitrary function g of the labelled multi-target state \mathbf{X}_k , given the available observations, is approximated as

$$E[g(\mathbf{X}_k) | Z^k] \approx \sum_{i=1}^{N_P} w_k(i) \sum_{\mathbf{x}_k \in \Pi_k(\mathbf{s}_k(i))} g(\mathbf{x}_k) p_l(\mathbf{x}_k | \mathbf{s}_k(i), Z^k) \quad (4.41)$$

where Π_k is defined by (4.21).

To be able to use approximation (4.41), the LUA-PF must produce, at each time step k , the following output

$$\left\{ \mathbf{s}_k(i), w_k(i), \left\{ p_l(\mathbf{x}_k | \mathbf{s}_k(i), Z^k) \right\}_{\mathbf{x}_k \in \Pi_k(\mathbf{s}_k(i))} \right\}_{i=1}^{N_P} \quad (4.42)$$

and in order to calculate these quantities, we are going to resort to an useful property of the Bayesian MTTL problem: the “one-sided” decoupling between tracking and labelling (see Appendix C). This property states that given nonrestrictive assumptions, tracking (in the sense of estimating locations) does not depend on labelling, although labelling depends on the results of tracking. In other

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING

words, we can iteratively obtain the set of particles $\{\mathbf{s}_k(i), w_k(i)\}_{i=1}^{N_P}$ without any need to concern ourselves about labelling, which can be done at a complementary step.

4.5.2.1 Calculation of particle states and weights (tracking step)

From Corollary C.1.5 (Appendix C), we have

$$f(\mathbf{s}_k|Z^k) = \frac{f(\mathbf{z}_k|\mathbf{s}_k)f(\mathbf{s}_k|Z^{k-1})}{f(\mathbf{z}_k|Z^{k-1})} \quad (4.43)$$

where

$$f(\mathbf{s}_k|Z^{k-1}) = \int f(\mathbf{s}_k|\mathbf{s}_{k-1})f(\mathbf{s}_{k-1}|Z^{k-1})\delta\mathbf{s}_{k-1}. \quad (4.44)$$

This recursion can be implemented using a M-SMC filter or any other multi-target tracking algorithm that is able to estimate the unlabelled multi-target posterior distribution (such as the MHT). These algorithms requires us to specify $f(\mathbf{z}_k|\mathbf{s}_k)$, $f(\mathbf{s}_0)$ and $f(\mathbf{s}_k|\mathbf{s}_{k-1})$; formulas for these densities for various multi-target models can be found in [Mahler, 2007, Chapters 12–14].

Note that although the LUA-PF is based on the RBMPF, the algorithm used to estimate $f(\mathbf{s}_k|Z^k)$ does not need to be a marginal particle filter. The reason is that due to the “one-sided” decoupling between tracking and labelling, we are free to use any algorithm capable of estimating $f(\mathbf{s}_k|Z^k)$ for the tracking step.

4.5.2.2 Calculation of particle labelling probabilities (labelling step)

To calculate the labelling probabilities, let us recall (4.31)

$$p_l(\mathbf{x}_k|\mathbf{s}_k, Z^k) = \frac{\int f(\mathbf{x}_k|\mathbf{x}_{k-1})f(\mathbf{x}_{k-1}|Z^{k-1})\delta\mathbf{x}_{k-1}}{f(\mathbf{s}_k|Z^{k-1})}$$

and since the denominator is constant for a given set of locations \mathbf{s}_k , it does not need to be explicitly calculated; it may be taken into account by normalizing the labelling probabilities for the given \mathbf{s}_k , according to (4.20). We therefore only need to look at the numerator. Assuming that $f(\mathbf{x}_{k-1}|Z^{k-1})$ is approximated

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING

by the set of particles

$$\left\{ \mathbf{s}_{k-1}(j), w_{k-1}(j), \left\{ p_1(\mathbf{x}_{k-1} | \mathbf{s}_{k-1}(j), Z^{k-1}) \right\}_{\mathbf{x}_{k-1} \in \Pi_{k-1}(\mathbf{s}_{k-1}(j))} \right\}_{j=1}^{N_P} \quad (4.45)$$

where $\Pi_k(\cdot)$ is defined by (4.21), and we can apply (4.41) to make the approximation

$$\begin{aligned} & \int f(\mathbf{x}_k | \mathbf{x}_{k-1}) f(\mathbf{x}_{k-1} | Z^{k-1}) \delta \mathbf{x}_{k-1} \\ & \approx \sum_{j=1}^{N_P} w_{k-1}(j) \sum_{\mathbf{x}_{k-1} \in \Pi_{k-1}(\mathbf{s}_{k-1}(j))} f(\mathbf{x}_k | \mathbf{x}_{k-1}) p_1(\mathbf{x}_{k-1} | \mathbf{s}_{k-1}(j), Z^{k-1}). \end{aligned} \quad (4.46)$$

where we have presented formulas for the labelled multi-target state transition density $f(\mathbf{x}_k | \mathbf{x}_{k-1})$ in Section 4.2.4. Finally, to initialize the recursion for $p_1(\mathbf{x}_k | \mathbf{s}_k, Z^k)$, we have to set

$$p_1(\mathbf{x}_0 | \mathbf{s}_0) = f(\mathbf{x}_0) / f(\mathbf{s}_0).$$

where formulas for $f(\mathbf{x}_0)$ are presented in Section 4.2.2.

From (3.46), we need to calculate the labelling probability of the each element of the set of possible labellings $\Pi_k(\mathbf{s}_k)$. From (4.21) and (4.46), $\Pi_k(\mathbf{s}_k)$ may be approximated as

$$\begin{aligned} & \Pi_k \left(\left\{ s_k^{(1)}, \dots, s_k^{(t_k)} \right\} \right) \\ & \approx \left\{ \mathbf{x}_k | \mathbf{x}_k = \left\{ \left[\begin{array}{c} s_k^{(1)} \\ l_k^{(1)} \end{array} \right], \dots, \left[\begin{array}{c} s_k^{(t_k)} \\ l_k^{(t_k)} \end{array} \right] \right\} \text{ and} \right. \\ & \left. \exists j, \mathbf{x}_{k-1} \in \Pi_{k-1}(\mathbf{s}_{k-1}(j)) \text{ s.t. } f(\mathbf{x}_k | \mathbf{x}_{k-1}) p_1(\mathbf{x}_{k-1} | \mathbf{s}_{k-1}(j), Z^{k-1}) > 0 \right\}. \end{aligned} \quad (4.47)$$

Depending on the number of targets and particles, $\Pi_k(\mathbf{s}_k)$ may contain labellings with extremely low probability, resulting in waste of computational processing. A solution is to replace the condition

$$f(\mathbf{x}_k | \mathbf{x}_{k-1}) p_1(\mathbf{x}_{k-1} | \mathbf{s}_{k-1}(j), Z^{k-1}) > 0$$

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING

with a stronger condition, for instance, replacing 0 with a probability threshold τ^{t_k} instead. If u denotes the units of measurement of the single-target location $s_k^{(i)}$, τ must have units of measurement u^{-1} . Naturally, the threshold should be small enough to prevent labelling hypotheses from prematurely disappearing, which would incur a sort of degeneracy.

4.5.3 LUA-PF algorithm

The algorithm presented below uses the M-SMC filter described in Section 2.3.4 for the tracking step. As we mentioned in Section 4.5.2.1, other Bayesian multi-target tracking algorithms can be used for this step.

Initialization: For each particle $i = 1, \dots, N_P$

1. Sample $\mathbf{s}_0(i) \sim f(\mathbf{s}_0)$ (refer to [Mahler, 2007, Chapter 14] for formulas for $f(\mathbf{s}_0)$)
2. Make $w_0(i) = \frac{1}{N_P}$
3. For each $\mathbf{x}_0 \in \Pi_0(\mathbf{s}_0(i))$, set $p_1(\mathbf{x}_0|\mathbf{s}_0(i)) = f(\mathbf{x}_0)/f(\mathbf{s}_0)$ (refer to Section 4.2.2 for formulas for $f(\mathbf{x}_0)$)

At every time step k :

1. For each particle $i = 1, \dots, N_P$
 - (a) Sample $\mathbf{s}_k(i) \sim q(\mathbf{s}_k|\mathbf{s}_{k-1}(i), \mathbf{z}_k)$, where $q(\mathbf{s}_k|\mathbf{s}_{k-1}, \mathbf{z}_k)$ is a proposal density
 - (b) Calculate the unnormalized weight according to

$$\bar{w}_k(i) = \frac{f(\mathbf{z}_k|\mathbf{s}_k(i))f(\mathbf{s}_k(i)|\mathbf{s}_{k-1}(i))}{q(\mathbf{s}_k|\mathbf{s}_{k-1}(i), \mathbf{z}_k)} \quad (4.48)$$

(refer to [Mahler, 2007, Chapters 12, 13] for formulas for $f(\mathbf{z}_k|\mathbf{s}_k)$ and $f(\mathbf{s}_k|\mathbf{s}_{k-1})$)

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING

(c) Obtain the set $\Pi_k(\mathbf{s}_k(i))$ according to

$$\begin{aligned} & \Pi_k \left(\left\{ s_k^{(1)}(i), \dots, s_k^{(t_k)}(i) \right\} \right) \\ & \approx \left\{ \mathbf{x}_k \mid \mathbf{x}_k = \left\{ \begin{bmatrix} s_k^{(1)}(i) \\ l_k^{(1)} \end{bmatrix}, \dots, \begin{bmatrix} s_k^{(t_k)}(i) \\ l_k^{(t_k)} \end{bmatrix} \right\} \text{ and } \exists j, \mathbf{x}_{k-1} \in \Pi_{k-1}(\mathbf{s}_{k-1}(j)) \right. \\ & \left. \text{s.t. } f(\mathbf{x}_k | \mathbf{x}_{k-1}) p_1(\mathbf{x}_{k-1} | \mathbf{s}_{k-1}(j), Z^{k-1}) > \tau^{t_k(i)} \right\}. \end{aligned} \quad (4.49)$$

where the parameter τ is discussed in Section 4.5.2.2 (refer to Section 4.2.4 for formulas for $f(\mathbf{x}_k | \mathbf{x}_{k-1})$)

(d) For each $\mathbf{x}_k \in \Pi_k(\mathbf{s}_k(i))$, compute the unnormalized particle labelling probability according to

$$\begin{aligned} \bar{p}_1(\mathbf{x}_k | \mathbf{s}_k(i), Z^k) &= \sum_{j=1}^{N_P} w_{k-1}(j) \\ &\times \sum_{\mathbf{x}_{k-1} \in \Pi_{k-1}(\mathbf{s}_{k-1}(j))} f(\mathbf{x}_k | \mathbf{x}_{k-1}) p_1(\mathbf{x}_{k-1} | \mathbf{s}_{k-1}(j), Z^{k-1}) \end{aligned} \quad (4.50)$$

(e) Normalize the particle labelling probabilities according to

$$p_1(\mathbf{x}_k | \mathbf{s}_k(i), Z^k) = \frac{\bar{p}_1(\mathbf{x}_k | \mathbf{s}_k(i), Z^k)}{\sum_{\tilde{\mathbf{x}}_k \in \Pi_k(\mathbf{s}_k(i))} \bar{p}_1(\tilde{\mathbf{x}}_k | \mathbf{s}_k(i), Z^k)} \quad (4.51)$$

2. Normalize the particle weights according to

$$w_k(i) = \frac{\bar{w}_k(i)}{\sum_{j=1}^{N_P} \bar{w}_k(j)} \quad (4.52)$$

3. Perform resampling by sampling N_P indexes $(\tilde{j}(i))_{i=1}^{N_P}$ according to the pmf $(w_k(j))_{j=1}^{N_P}$ and afterwards making, for $i = 1, \dots, N_P$

$$\begin{aligned} \mathbf{s}_k(i) &:= \mathbf{s}_k(\tilde{j}(i)) \\ p_1(\mathbf{x}_k | \mathbf{s}_k(i), Z^k) &:= p_1(\mathbf{x}_k | \mathbf{s}_k(\tilde{j}(i)), Z^k), \quad \forall \mathbf{x}_k \in \Pi_k(\mathbf{s}_k(\tilde{j}(i))) \\ w_k(i) &:= \frac{1}{N_P} \end{aligned}$$

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING

4.5.4 Computational aspects

On analyzing the computational complexity of LUA-PF, we focus on the labelling step, i.e. computing the particle labelling probabilities $p_1(\mathbf{x}_k | \mathbf{s}_k(i), Z^k)$. For the tracking step, as we mentioned, we can use any Bayesian multi-target tracking algorithm, and therefore any technique designed to reduce computational complexity of such algorithms (such as the Independent Partitions and Coupled Partitions methods proposed by Kreucher et al. [2005], or hypothesis pruning techniques applicable to the MHT). That does not mean, of course, that obtaining a good computational performance in Bayesian multi-target tracking is a simple task, but it is extensively covered in the literature and hence not the main focus of this work.

If we consider a constant number of targets t , there are t possible labels, and hence $|\Pi_k(\mathbf{s}_k(i))|$ can be as high as $t!$. This means that from (4.50), the worst-case complexity of calculating a single labelling probability for a single particle $\mathbf{s}_k(i)$ is $O(N_P t!)$, and to compute all labelling probabilities for all particles, the worst-case complexity is $O(N_P^2 (t!)^2)$. Needless to say, this computational cost can be prohibitive even if we consider a problem of labelling, let us say, seven targets.

If we assume that target births and deaths may occur, $\Pi_k(\mathbf{s}_k)$ may also grow with time. The explanation is the following. In our Bayesian formulation of the MTTL problem in Section 4.2, in order to make the problem well-posed from a physical perspective, we assume the number of appearing targets at each time step (and their corresponding labels) to be known. However, we are not required to assume that the labels of the disappearing targets, neither the number of disappearing targets, are also known. So, in principle, the algorithm will keep taking a particular label in consideration until there is very high probability that it has disappeared. This may never happen due to lack of observability.

For these reasons, the LUA-PF is unsuitable for tracking a large number of targets, and if we consider the possibility of recurrent target births/deaths, it is also unsuitable for long time horizons. However, for the problem focused in this Chapter, illustrated in Fig. 4.1 (i.e. tracking a small group of targets and individually identifying the targets of the group after separation), the LUA-PF filter may be a feasible solution. For more complex scenarios, we may resort

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING

to additional approximations, or to more pragmatic solutions, such as pruning all but one labelling hypothesis after the user becomes aware of the existence of mixed labelling (due to receiving some warning message).

The algorithm has also good parallelization properties: steps (4.49)–(4.51) can be fully parallelized, by making each computation node process a single labelling hypothesis $\mathbf{x}_k(i)$. In this way, the computational complexity of each node can be reduced down to $O(N_P t!)$.

4.6 Simulations

4.6.1 Which metric to use?

To decide how to empirically evaluate an arbitrary MTTL algorithm, it is only natural that we return to the two questions we proposed in Section 4.1, i.e. that we evaluate how does the algorithm fare on attempting to answer the two questions.

Question 1 (How does one optimally assign labels to a set of tracks?) may be evaluated using the hit-or-miss metric that we proposed in Section 4.3.3. We take the true target locations as the unlabelled tracks, find the optimal assignment of labels to these tracks (using the MLP step described in Section 4.3.2.1) and see how often this labelling matches the “true labelling” across a series of Monte Carlo runs.

The “true labelling” is the assignment of labels to the true locations consistent with the assignment of labels to the initial distributions (in $f(\mathbf{x}_0)$) and to the distributions of appearing targets (in $f(\mathbf{x}_k|\mathbf{x}_{k-1})$). If the Bayesian labelling problem is well-posed (in the sense of the initial/appearing target distributions being well-separated in either space or time, as described in Section 4.2.1), this labelling choice should be unique.

The *rate of labelling errors* for a MTTL algorithm, for a given time k , is then given by

$$\varepsilon_k^{\text{true}} = 1 - \frac{1}{N_R} \sum_{i=1}^{N_R} \epsilon(\hat{\mathbf{x}}_k(i), \mathbf{x}_k) \quad (4.53)$$

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING

where ϵ is the hit-or-miss metric given by (4.29), N_R is the number of Monte Carlo runs, \mathbf{x}_k is the true multi-target state (with the “true labelling”), and $\hat{\mathbf{x}}_k(i)$ is the set of tracks computed by the algorithm being tested in the i th-run.

The reader may wonder why do we base performance evaluation on the ground truth locations, rather than using the MMOSPA or JoM methods (described in Section 4.3.2) to obtain a set of unlabelled tracks. The reason is that the novel features of the LUA-PF filter affect only labelling, not tracking (and the choice of the tracking algorithm is flexible), such that it makes sense to evaluate only the labelling performance. In case we wanted to jointly evaluate tracking and labelling, we could use the “labelling-aware” modified OSPA metric presented by Ristic et al. [2011a].

Question 2 (What is the probability that the assignment of labels is incorrect?) may be evaluated by looking at the probability of labelling error calculated using the MTTL algorithm being tested, given by $1 - p_1(\hat{\mathbf{x}}_k | \mathbf{s}_k, Z^k)$ (where \mathbf{s}_k denotes the set of true target locations), and comparing it with the actual rate of labelling errors, given by (4.53). A large difference between these quantities implies that the given algorithm tends to under- or overestimate its own labelling errors.

To perform this evaluation, we can compare the rate of labelling errors with the algorithm-calculated probability of labelling error averaged over a series of Monte Carlo runs, given by

$$\epsilon_k^{\text{calc}} = 1 - \frac{1}{N_R} \sum_{i=1}^{N_R} p_1(\hat{\mathbf{x}}_k(i) | \mathbf{s}_k, Z^k). \quad (4.54)$$

We must remind, however, that ϵ_k^{calc} does not say anything about the variance of $p_1(\hat{\mathbf{x}}_k(i) | \mathbf{s}_k, Z^k)$ across the Monte Carlo runs. This variance is caused both by the variance of the sequence of observations Z^k , and by the Monte Carlo variance inherent to SMC methods (i.e. the variance due to importance sampling and resampling, when applicable).

The second type of variance is problematic, as it may indicate that the calculated $p_1(\hat{\mathbf{x}}_k(i) | \mathbf{s}_k, Z^k)$ is unreliable. In order to observe the Monte Carlo variance, we perform a second analysis, this time using a Monte Carlo simulation with a

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING

fixed sequence of measurements Z^k (such that the variance in the results is entirely due to the Monte Carlo variance). In this analysis, we look at the standard deviation of the calculated probability of labelling error $1 - p_1(\hat{\mathbf{x}}_k | \mathbf{s}_k)$ across the multiple runs, given by

$$\sigma_k^\varepsilon = \sqrt{\frac{1}{N_R} \sum_{i=1}^{N_R} (\varepsilon_k^{\text{calc}} - (1 - p_1(\hat{\mathbf{x}}_k(i) | \mathbf{s}_k, Z^k)))^2}. \quad (4.55)$$

4.6.2 Scenarios

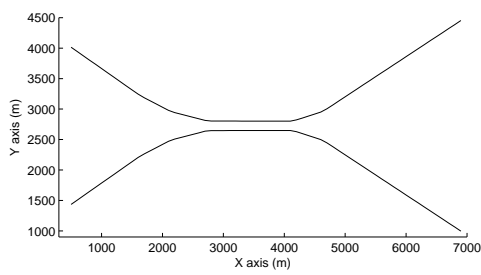
The following scenarios, shown in Fig. 4.6, are analyzed:

1. Two targets approach each other and separate after some time;
2. Two targets approach each other and separate after some time (with swapped routes);
3. Two targets approach each other (with smaller target separation) and separate after some time;
4. Two targets approach each other and separate after some time, with one of the targets appearing some instants before the approximation, and another target disappearing some instants after the separation.

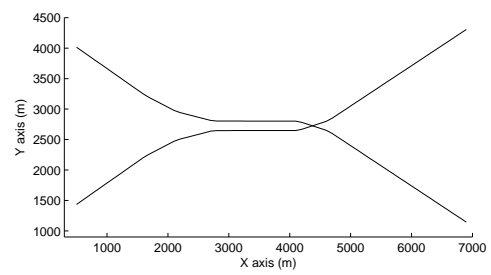
The multi-target state transition density assumed by the tracker is given by (4.7) for Scenarios 1–3, and by (4.13) for Scenario 4, with $P_S \left(s_{k-1}^{(j)} \right)$ constant and equal to 0.95. We assume that the number of initial targets, as well as the number of appearing targets at each time k , is known. This implies that the multi-target prior is given by (4.3), and for Scenario 4, that $r^{(l)} = 1$ for $l \in \mathbb{L}_k$. The location initial distributions are simply the true target initial states, with similar considerations for the appearing target. The location vector and the single-target state transition model are the same as in the numerical example in Section 4.4.4.

The multi-target measurement model $f(\mathbf{z}_k | \mathbf{s}_k)$ corresponds to the detection-type measurement model (described in [Mahler, 2007, Section 12.3]). Missed detections and false alarms are only considered in the last scenario (with target

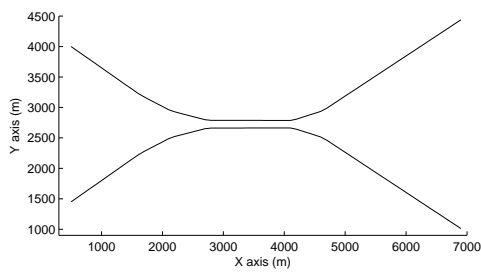
4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING



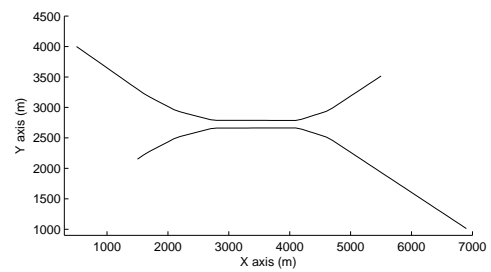
(a) Scenario 1



(b) Scenario 2



(c) Scenario 3



(d) Scenario 4

Figure 4.6: Multi-target simulation scenarios

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING

birth and death), with probability of detection 0.95 and uniform clutter density of $2 \cdot 10^{-7}$ per unit of area. The single-measurement, single-target likelihood function is also the same as in Section 4.4.4.

For all scenarios, we evaluate both the (SIR) M-SMC filter described in Section 2.3.4 and the LUA-PF algorithms. For both the M-SMC filter and the LUA-PF, we use 2,000 particles for the scenarios 1, 2 and 3, and 4,000 particles for scenario 4. For both filters, we use blind importance sampling, i.e. we use $f(\mathbf{x}_k|\mathbf{x}_{k-1})$ as proposal density for the M-SMC filter, and $f(\mathbf{s}_k|\mathbf{s}_{k-1})$ for the LUA-PF.

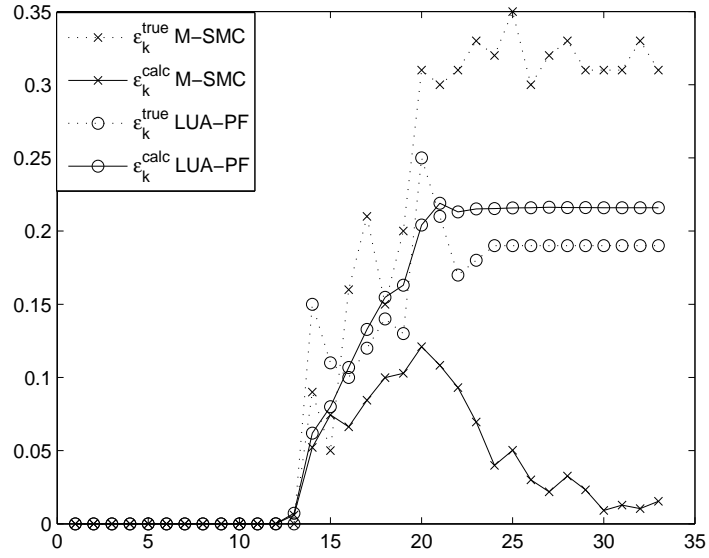
The reader may wonder why did we not consider other MTTL algorithms proposed in the literature, such as the ones in Blom and Bloem [2011]; Crouse et al. [2011a]. The reason is simply because these algorithms are based on alternate mathematical definitions of “labelling probability” (with unclear physical interpretation, as we mentioned in Section 4.1) and hence do not fit the analysis framework described in Section 4.6.1.

4.6.3 Results for Monte Carlo runs with varying sequence of measurements

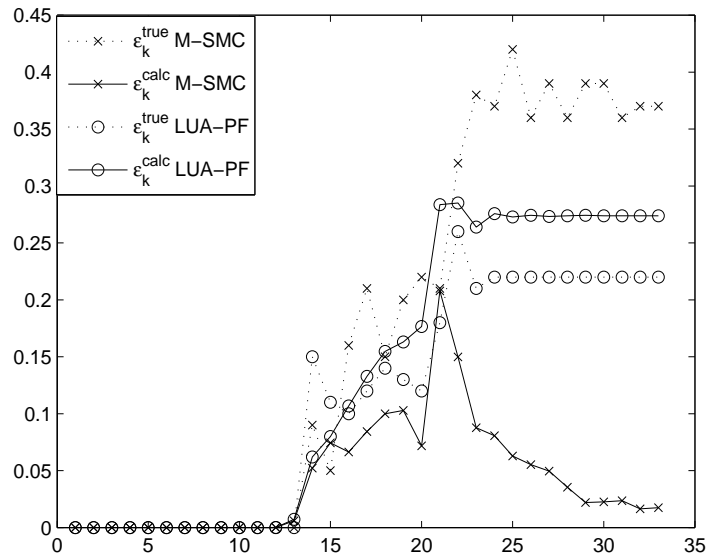
The results for the Monte Carlo simulation with Z^k being resampled at each run (as usual), for both the M-SMC filter and LUA-PF, are shown in Figs. 4.7 and 4.8, which show both the rate of labelling errors $\varepsilon_k^{\text{true}}$ and the average calculated probability of labelling error $\varepsilon_k^{\text{calc}}$. In terms of $\varepsilon_k^{\text{true}}$, we see that the LUA-PF provided a lower rate of labelling errors $\varepsilon_k^{\text{calc}}$ for all scenarios, i.e. it was clearly a superior solution w.r.t. answering **Question 1**. The improvement of using the LUA-PF was much more significant in Scenarios 1 and 2, where the separation between the targets was larger (and hence ambiguity in label-to-location association was lower).

In terms of $\varepsilon_k^{\text{calc}}$, we can see that, as expected, the M-SMC filter is impaired by degeneracy; for all scenarios, $\varepsilon_k^{\text{calc}}$ is severely underestimated, tending to zero with time. The LUA-PF, on the other hand, exhibits far more consistency between actual and calculated errors, clearly avoiding the degeneracy phenomenon since after target separation as $\varepsilon_k^{\text{calc}}$ does not decrease with time. Moreover, $\varepsilon_k^{\text{calc}}$ remains constant with time. This is consistent with the theoretical behavior of

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING



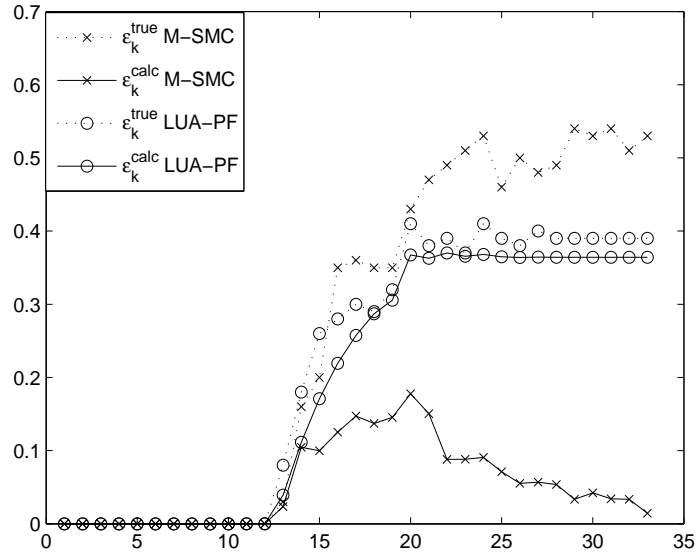
(a) Scenario 1



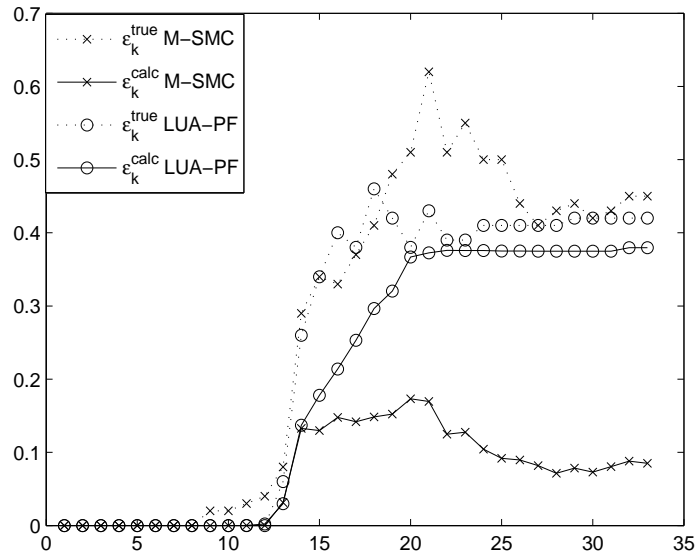
(b) Scenario 2

Figure 4.7: Simulation (varying Z^k) results for scenarios 1 and 2

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING



(a) Scenario 3



(b) Scenario 4

Figure 4.8: Simulation (varying Z^k) results for scenarios 3 and 4

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING

$f(\mathbf{x}_k|Z^k)$ for this type of scenario, that we have described in Appendix B.3. The difference between $\varepsilon_k^{\text{calc}}$ and $\varepsilon_k^{\text{true}}$ is somewhat higher, however, for Scenarios 1 and 2. This will be further discussed in the next analysis.

As we know, the LUA-PF is computationally more expensive than the M-SMC filter. If we increase the number of particles of the M-SMC filter such that it has computational cost comparable to the LUA-PF, then depending on the number of targets, the M-SMC filter might be able to outperform the LUA-PF in terms of rate of labelling errors. However, the M-SMC filter would lead to degeneracy in the calculation of the probability of labelling error regardless of the number of particles.

4.6.4 Results for Monte Carlo runs with fixed sequence of measurements

The results for the Monte Carlo simulation with fixed Z^k , for the LUA-PF, are shown in Fig. 4.9. Only scenarios 1 and 3 are considered in this simulation, in order to observe the effect of increased target separation on σ_k^ε , the standard deviation of the calculated probability of labelling error. Since we have seen that the M-SMC filter is affected by degeneracy (and hence completely incorrect $\varepsilon_k^{\text{calc}}$), there is no point on considering it for this analysis.

The variation in $\varepsilon_k^{\text{calc}}$ is caused solely by the Monte Carlo variance (as we have fixed the measurements), and hence, a large σ_k^ε indicates low reliability of the particle filter algorithm. The results show that σ_k^ε is significant, and even more significant for scenario 1, with larger target separation. Together with the results in Figs. 4.7, 4.8, we notice that it is more difficult to accurately calculate the labelling probabilities for scenarios involving closed spaced targets which never become very close to each other (in comparison to scenarios where the targets become almost adjacent).

If a more accurate calculation of the probability of labelling error (i.e. a more accurate answer to **Question 2**) is required, we need to reduce the Monte Carlo variance, for instance by increasing the number of particles, using a better proposal density, or employing MCMC methods as a replacement or complement to importance sampling.

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING

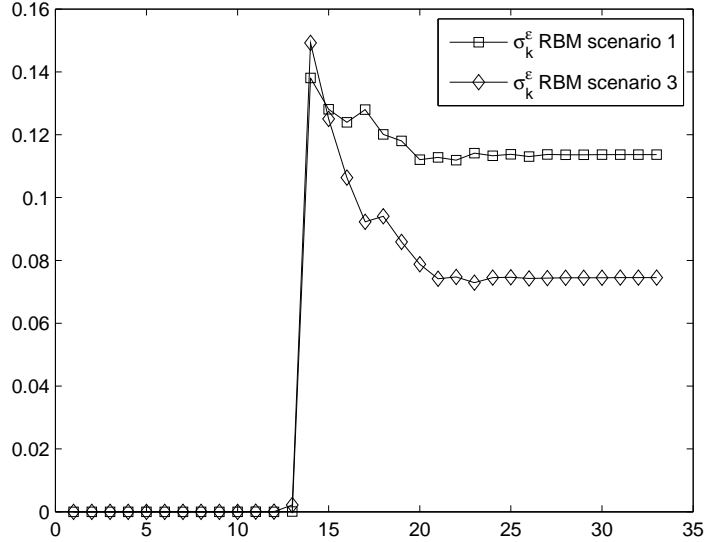


Figure 4.9: Simulation (fixed Z^k) results

4.7 Conclusions and recommendations

In this chapter, we presented an in-depth analysis of the Bayesian labelling problem, focusing on practical aspects, such as how to perform optimal labelling (in probabilistic sense) and how to characterize the probability of labelling error, corresponding respectively to **Question 1** and **Question 2** that we have proposed for the motivational scenario described in Section 4.1. A recurring concern of this work was to define statistics with clear physical interpretation (i.e. that are meaningful to the user of the system), rather than only abstract mathematical quantities.

We believe that the contents of this work provide a solid framework for the development of new multi-target target tracking algorithms with labelling capability, for better understanding and adapting existing algorithms, and for evaluating multi-target tracking and labelling performance. We also remark that since the statistics proposed in this chapter have clear physical interpretation, one can obtain analogous versions of them for multi-target statistical representations other than FISST (such as Janossy densities and families of conventional

4. THE PROBLEM OF OPTIMAL BAYESIAN TRACK LABELLING IN MULTI-TARGET TRACKING

probability densities, as described in Kastella [1997b]), making the contents of this work highly general.

Moreover, we have used the results of our theoretical analysis to propose a PF-like algorithm, the LUA-PF, that implements the Bayesian recursion described in Section 4.2 and avoids the degeneracy phenomenon described in Section 4.4.3. The experimental results show that the proposed algorithm is indeed far more suitable to answering the two questions that we proposed in Section 4.1 than the “plain vanilla” particle filter implementation of the MTTL problem, i.e. the SIR M-SMC filter.

Some care, however, is necessary when using the proposed algorithm to answer **Question 2**, as our results indicate that the Monte Carlo variance of the calculated labelling probabilities may still be significant. If that happens, we may attempt to use well-established SMC techniques to reduce this variance, or to increase the number of particles, although the latter may be unfeasible due to the high computational cost of the algorithm (discussed in Section 4.5.4).

Interesting future works include comparing the track extraction methods described in Section 4.3.2 (and possibly propose new methods), studying effective ways of integrating Bayesian labelling with non-Bayesian labelling (for large scale scenarios where we cannot formulate a well-posed Bayesian labelling problem, or where the LUA-PF is computationally unfeasible), and for the LUA-PF, finding more computationally efficient ways to calculate labelling probabilities, allowing us to handle more complex scenarios. Naturally, it would also be interesting to try the proposed algorithm with more complex observation models, such as the track-before-detect observation model.

Chapter 5

An analysis of information-driven sensor management criteria

In sensor management, the usefulness of information theoretic measures seems to be validated by a large number of empirical studies, but theoretical justification presented until so far, both for selection of the measure and for the use of information-driven sensor management itself, still seems unclear, inconsistent or debatable.

In this chapter, we argue that information-driven sensor management may be justified on the basis of uncertainty reduction rather than information gain, by noting the equivalence between the Kullback-Leibler and the Shannon entropy sensor management criteria. This is also used to demonstrate that, unlike previous claims, the asymmetry of the KL divergence is not relevant to the sensor management problem.

Moreover, we provide a rebuttal to one strong argument for using the Rényi divergence as a sensor management criterion, namely the “near-universal proxy” argument. Finally, by using some simple but representative numerical examples, we analyze the practical effectiveness of the KL and Rényi divergences to problems where we would like to balance estimation errors in different dimensions.

5.1 Introduction

As we mentioned in Section 2.4.2.2, information-driven sensor management consists of performing sensor management by attempting to maximize the “information content” of the posterior, i.e. its capacity of yielding useful information to the user. The information content is typically quantified using measures employed in information theory; for this reason, information-driven sensor management is also commonly referred as *information-theoretic* sensor management.

Hintz and McVey [1991] were the first to suggest using information theory in a problem related to sensor management and state estimation, followed by Manyika and Durrant-Whyte [1994], who considered expected information gain in sensor management and data fusion problems. The idea of using the KL divergence for sensor management appeared in the works of Schmaedeke and Kastella [1994], Kastella [1997a], and Mahler [1996]. Doucet et al. [2002] provided a particle filter implementation of sensor management based on the KL divergence, which could be used for general non-linear systems.

The idea of using the more general Rényi divergence (or α -divergence), instead of the KL divergence, was introduced by Kreucher et al. [2003a] and demonstrated for a multi-target tracking problem. In this initial work, the authors did not provide a particular reason for using the α -divergence instead of the KL divergence, except that it gives an extra freedom of choosing the parameter α , which could, in principle, emphasize certain parts of the distribution functions.

Later, Kreucher et al. [2003b] suggest using either $\alpha = 0.5$ (which corresponds to the Hellinger affinity) or $\alpha = 1$ (which corresponds to the KL divergence), based on Hero et al. [2002], where an empirical study on resolution of clusters for a georegistration problem was done. The authors also suggest the use of the value $\alpha = 0.5$ when the prior and posterior densities are similar. This is based on an asymptotic analysis, provided in Hero et al. [2001], of the Chernoff exponent, used in hypothesis testing between two probability densities.

However, Aughenbaugh and La Cour [2008] showed that the supposed superior discrimination capability of the α -divergence with $\alpha = 0.5$ does not necessarily correspond to our intuitive interpretation of information gain. Through analysis of a few examples, they observed that using $\alpha = 0.5$ for sensor management, as

5. AN ANALYSIS OF INFORMATION-DRIVEN SENSOR MANAGEMENT CRITERIA

opposed to using the KL divergence, seems to result in actions that emphasize morphological changes on the distribution (such as rotation, or translation of modes for multi-modal distributions). Such kind of changes may not necessarily be desirable from an operational point of view.

Hanselmann et al. [2008] supported the use of the α -divergence instead of the KL divergence, using the asymmetry of the KL divergence (as opposed to the symmetry of the α -divergence for $\alpha = 0.5$) as argument. Hero et al. [2007]; Kreucher et al. [2005] made a strong theoretical argument in favor of information-driven sensor management, by claiming that the expected value of arbitrary risk functions is sandwiched between functions of two marginalized Rényi divergences; this would make a criterion based on the α -divergences a “near-universal” proxy for task-driven sensor management.

Aughenbaugh and La Cour [2011] also compares information-theoretic measures for sensor management applied to the problem of tracking a maneuvering target observed by multi-static sensors. In this work they state the well-known relationship (Cover and Thomas [1991]; Williams [2007]) between Shannon entropy, mutual information and expected KL divergence, but the comparison between criteria is done empirically. Finally, Ristic et al. [2011a] describe the implementation of α -divergences for the Probability Hypothesis Density (PHD) filter, and also present an empirical comparison between those and other criteria.

In this chapter, we will make a re-evaluation of previous arguments for using (or not) KL/Rényi divergences as information-theoretic sensor management criteria. The organization and contributions of this chapter are as follows:

- Section 5.2 reviews the KL Divergence, Shannon entropy, α -divergence and Rényi entropy criteria;
- Section 5.3 looks at the properties of the Shannon entropy and presents a theoretical justification of the KL divergence sensor management criterion based on its relationship with the Shannon entropy;
- Section 5.4 explains the previous “near-universal proxy” argument in favor of α -divergences, and presents a rebuttal to this argument;
- Section 5.5 provides an empirical analysis of the effectiveness of the KL

5. AN ANALYSIS OF INFORMATION-DRIVEN SENSOR MANAGEMENT CRITERIA

and α -divergence sensor management criteria to the problem of balancing estimation errors in different dimensions. Although we confirm the result of Aughenbaugh and La Cour [2008], which suggests that using lower values of α favors distribution rotations over reduction of uncertainty, this property does not lead to any practical benefit in our considered problem of balancing estimation errors;

- Section 5.6 draws conclusions.

5.2 Information-theoretic sensor management

Let us consider again the mathematical formulation of the sensor management problem provided in Section 2.4.1. As we have seen in Section 2.4.2.2, in the case of information-driven sensor management, the reward/risk function at time k has form $\gamma(Z_k, U_k)$, i.e. it is a function of the observation and of the sensing action, not of the true state. In this section, we take a closer look at four information-theoretic sensor management criteria.

5.2.1 Shannon entropy

In this criterion, the risk function is the Shannon entropy (also known simply as entropy) of the posterior distribution. Entropy is considered to be a measure of “uncertainty” of a distribution, being given by

$$H(X_k | Z^k) \triangleq - \int p(x_k | Z^k) \log p(x_k | Z^k) dx_k \quad (5.1)$$

where without loss of generality, we are assuming that X_k is a continuous random variable (in this case, the Shannon entropy is also known as *differential entropy*). The entropy, as well as other information-driven criteria discussed in this section, can be similarly defined for discrete and hybrid continuous-discrete random variables.

5.2.2 Kullback-Leibler (KL) divergence

The KL divergence, also known as relative Shannon entropy, is a measure of difference between two distributions. The KL divergence, as a sensor management criterion, measures the “information gain” obtained by moving from the prior $p(x_k | Z^{k-1})$ to the posterior $p(x_k | Z^k)$, making it a reward function given by

$$D(p(X_k | Z^k) || p(X_k | Z^{k-1})) \triangleq \int p(x_k | Z^k) \log \frac{p(x_k | Z^k)}{p(x_k | Z^{k-1})} dx_k \quad (5.2)$$

where we apply the conventions

$$\begin{aligned} \log \frac{p(x_k | Z^k)}{p(x_k | Z^{k-1})} &= 0, \text{ for } p(x_k | Z^k) = p(x_k | Z^{k-1}) = 0, \\ p(x_k | Z^k) / 0 &= \infty, \text{ for } p(x_k | Z^k) > 0. \end{aligned}$$

Since the KL divergence is asymmetric, one can, using analogous conventions, define the following alternate reward function:

$$D(p(X_k | Z^{k-1}) || p(X_k | Z^k)) \triangleq \int p(x_k | Z^{k-1}) \log \frac{p(x_k | Z^{k-1})}{p(x_k | Z^k)} dx_k \quad (5.3)$$

and in fact, Doucet et al. [2002], suggests, an alternative, using the average of reward functions (5.2) and (5.3).

5.2.3 Rényi entropy

By taking the same Fadeev’s postulates used to characterize the Shannon entropy, one can define a generalization of the Shannon entropy (Rényi [1961]), which corresponds to the Rényi entropies

$$H_\alpha(X_k | Z^k) \triangleq \frac{1}{1-\alpha} \log \int p(x_k | Z^k)^\alpha dx_k \quad (5.4)$$

where the parameter α may be used to give more or less emphasis to low probability regions (“tails”) of the distribution. The Rényi entropy is considered as a

5. AN ANALYSIS OF INFORMATION-DRIVEN SENSOR MANAGEMENT CRITERIA

generalization of the Shannon entropy due to the relationship

$$\lim_{\alpha \rightarrow 1} H_\alpha (X_k | Z^k) = H (X_k | Z^k) \quad (5.5)$$

i.e. the Shannon entropy corresponds to the Rényi entropy for $\alpha \rightarrow 1$.

5.2.4 Rényi divergence

The Rényi divergence or α -divergence is a generalization of the KL divergence, also in the sense that it satisfies a set of postulates that characterize the KL divergence. The α -divergence is given by

$$D_\alpha (p (X_k | Z^k) || p (X_k | Z^{k-1})) \triangleq \frac{1}{\alpha - 1} \log \int p (x_k | Z^k)^\alpha p (x_k | Z^{k-1})^{1-\alpha} dx_k \quad (5.6)$$

where we apply the conventions

$$\begin{aligned} p (x_k | Z^k)^\alpha p (x_k | Z^{k-1})^{1-\alpha} &= 0, \text{ for } p (x_k | Z^k) = p (x_k | Z^{k-1}) = 0, \\ p (x_k | Z^k) / 0 &= \infty, \text{ for } p (x_k | Z^k) > 0 \end{aligned}$$

and D_0 and D_1 are defined using the limits from right and left respectively, which makes D_1 the same as the KL divergence. Similarly to the KL divergence, we can define the alternate reward function

$$D_\alpha (p (X_k | Z^{k-1}) || p (X_k | Z^k)) \triangleq \frac{1}{\alpha - 1} \log \int p (x_k | Z^{k-1})^\alpha p (x_k | Z^k)^{1-\alpha} dx_k. \quad (5.7)$$

A special case of the α -divergence is $D_{0.5}$, which is a *true metric*, in the sense that it obeys the triangle inequality and it is symmetric, i.e.

$$D_{0.5} (p (X_k | Z^k) || p (X_k | Z^{k-1})) = D_{0.5} (p (X_k | Z^{k-1}) || p (X_k | Z^k)). \quad (5.8)$$

5.3 A look at the Shannon entropy and KL divergence criteria

5.3.1 Entropy as an uncertainty measure

For a sequence of i.i.d. discrete random variables, the Shannon source coding theorem relates entropy to the minimum necessary number of bits for lossless data compression of the sequence. In other words, entropy corresponds to a notion of the amount of “uncertainty” contained in a distribution, since the less predictable the symbols of the sequence are, the greater the risk is of data loss by not completely encoding every symbol. Some relatively similar results can be obtained for continuous random variables (see Cover and Thomas [1991]).

In order to justify the use of entropy as a sensor management criterion, it is desirable to analyze how the “uncertainty” represented by the measure is related to estimation performance. In the case of a Gaussian posterior distribution, we have the following well-known relationship between Shannon entropy and covariance

$$H(X_k | Z^k) = \frac{n}{2} \log 2\pi e + \frac{1}{2} \log \det P_k. \quad (5.9)$$

where n is the number of dimensions of X_k and P_k is the covariance of X_k given all available information Z^k . Hence, in the Gaussian case, the entropy is monotonically increasing with the covariance determinant. This behavior is appropriate as the covariance gives us an intuitive notion of uncertainty for Gaussian distributions.

Let us now consider a Gaussian mixture posterior distribution, i.e.

$$p(x_k | Z^k) = \sum_{i=1}^{N_M} w_k^{(i)} \mathcal{N}(x_k; \hat{x}_k^{(i)}, P_k^{(i)})$$

where N_M is the number of components of the mixture and $w_k^{(i)}$, $\hat{x}_k^{(i)}$ and $P_k^{(i)}$, $i = 1, \dots, N_M$ denote respectively the weight, mean and covariance of each com-

5. AN ANALYSIS OF INFORMATION-DRIVEN SENSOR MANAGEMENT CRITERIA

ponent. Then the following relationship (see Huber et al. [2008]) holds:

$$\begin{aligned}
 & - \sum_{i=1}^{N_M} w_k^{(i)} \log \left(\sum_{j=1}^{N_M} w_k^{(j)} \mathcal{N} \left(\hat{x}_k^{(i)}; \hat{x}_k^{(j)}, P_k^{(i)} + P_k^{(j)} \right) \right) \\
 & \leq H(X_k | Z^k) \leq \\
 & \sum_{i=1}^{N_M} w_k^{(i)} \left(-\log w_k^{(i)} + \frac{n}{2} \log 2\pi e + \frac{1}{2} \log \det P_k^{(i)} \right). \tag{5.10}
 \end{aligned}$$

Now, recall that the covariance of a Gaussian mixture with mean \hat{x}_k is given by

$$P_k = \sum_{i=1}^{N_M} w_k^{(i)} \left(P_k^{(i)} + \left(\hat{x}_k^{(i)} - \hat{x}_k \right) \left(\hat{x}_k^{(i)} - \hat{x}_k \right)^T \right) \tag{5.11}$$

where \hat{x}_k is the mean of the entire mixture (i.e. $\hat{x}_k = \sum_{i=1}^{N_M} w_k^{(i)} \hat{x}_k^{(i)}$). The consequence is that if we arbitrarily increase the distance between the means of the components of the mixture, we are able to arbitrarily increase P_k . On the other hand, by looking at (5.10), we see that the lower bound of $H(X_k | Z^k)$ depends on both means and covariances of individual components of the mixture, but the upper bound depends only on covariances. So, if we arbitrarily increase the distance between modes, we can at most make the entropy closer to the upper bound.

This means that for a Gaussian mixture with sufficiently high separation between the modes, entropy practically depends only on the amount of uncertainty around each component, without considering how dispersed these components are. This behavior is illustrated in Fig. 5.1 for a scalar (i.e. single-dimensional X_k) problem. Distribution B has smaller variance, but higher entropy than distribution A.

In this situation, entropy is in some sense a more intuitive measure of uncertainty than variance. This is because by looking at the distribution A, we know that the state is likely to be very close to point $x = -5$, and if that is not the case, to point $x = 5$. In contrast, by looking at the distribution B we just have a vague idea of where the real state is! That does not mean, however, that entropy is always a better sensor management criterion than variance for a scalar problem;

5. AN ANALYSIS OF INFORMATION-DRIVEN SENSOR MANAGEMENT CRITERIA

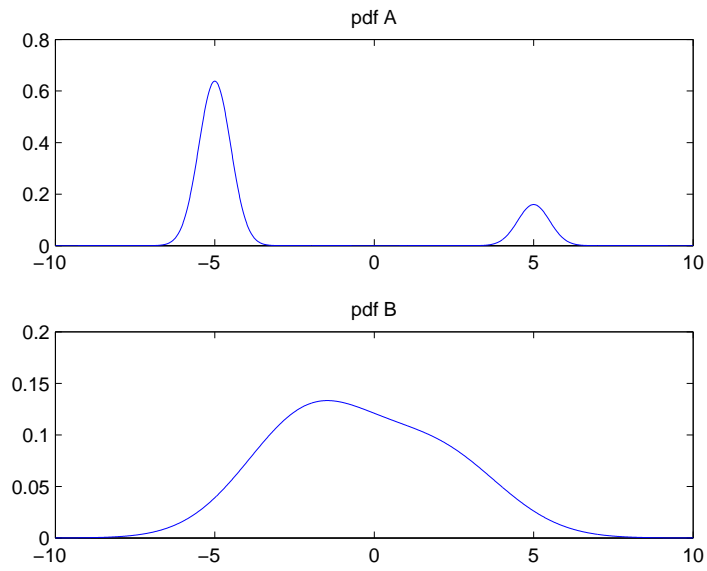


Figure 5.1: Two Gaussian mixture pdfs, the first with variance 16.25 and entropy within the interval $[1.07, 1.23]$, and the second with variance 7.84 and entropy within $[2.32, 2.79]$

5. AN ANALYSIS OF INFORMATION-DRIVEN SENSOR MANAGEMENT CRITERIA

for instance, if it is desirable to minimize the squared error w.r.t. to the mean \hat{x}_k , the variance criterion would still, by construction, provide the optimal results. This is because entropy can only measure the “overall uncertainty” contained in a distribution, not the errors relative to a particular estimate.

5.3.2 Relationship between the KL divergence and Shannon entropy criteria

Let us assume that the system is a POM1DF (hence with the properties described in Section 2.4.3). Observe that, for this type of system, we have

$$\begin{aligned}
 & p(x_k | Z^{k-1}, u_k) \\
 &= \underbrace{\int \dots \int}_k p(x_k | x_0, \dots, x_{k-1}, Z^{k-1}, u_k) p(x_0, \dots, x_{k-1} | Z^{k-1}, u_k) dx_0 \dots dx_{k-1} \\
 &= \underbrace{\int \dots \int}_k p(x_k | x_{k-1}) p(x_0, \dots, x_{k-1} | Z^{k-1}) dx_0 \dots dx_{k-1} \tag{5.12}
 \end{aligned}$$

which does not depend on u_k , and hence, a POM1DF has the property

$$p(x_k | Z^{k-1}, u_k) = p(x_k | Z^{k-1}). \tag{5.13}$$

The *mutual information* between the state X_k and the observation Z_k , con-

5. AN ANALYSIS OF INFORMATION-DRIVEN SENSOR MANAGEMENT CRITERIA

ditioned on Z^{k-1} and on the sensing action u_k , is defined as

$$\begin{aligned}
& I(X_k; Z_k | Z^{k-1}, u_k) \\
& \triangleq \int p(x_k, z_k | Z^{k-1}, u_k) \log \frac{p(x_k, z_k | Z^{k-1}, u_k)}{p(x_k | Z^{k-1}, u_k) p(z_k | Z^{k-1}, u_k)} dx_k dz_k \\
& = \int p(z_k | Z^{k-1}, u_k) \int p(x_k | Z^k) \log \frac{p(x_k | Z^k)}{p(x_k | Z^{k-1}, u_k)} dx_k dz_k \\
& = \int p(z_k | Z^{k-1}, u_k) \int p(x_k | Z^k) \log \frac{p(x_k | Z^k)}{p(x_k | Z^{k-1})} dx_k dz_k \\
& = \int p(z_k | Z^{k-1}, u_k) D(p(X_k | Z^k) || p(X_k | Z^{k-1})) dz_k \\
& = E [D(p(X_k | Z_k, u_k, Z^{k-1}) || p(X_k | Z^{k-1})) | Z^{k-1}, u_k] \tag{5.14}
\end{aligned}$$

i.e. the mutual information, for a POM1DF system, corresponds to the expected KL divergence, conditioned on the available information Z^{k-1} and the sensing action u_k , with the expectation taken over Z_k .

Moreover, the mutual information can be rewritten as

$$\begin{aligned}
& I(X_k; Z_k | Z^{k-1}, u_k) \\
& = \int p(z_k | Z^{k-1}, u_k) \int p(x_k | Z^k) \log p(x_k | Z^k) dx_k dz_k \\
& \quad - \int p(z_k | Z^{k-1}, u_k) \int p(x_k | Z^k) \log p(x_k | Z^{k-1}) dx_k dz_k \\
& = - \int p(z_k | Z^{k-1}, u_k) H(X_k | Z^k) dz_k \\
& \quad - \int \int p(x_k, z_k | Z^{k-1}, u_k) \log p(x_k | Z^{k-1}) dx_k dz_k \\
& = - \int p(z_k | Z^{k-1}, u_k) H(X_k | Z^k) dz_k \\
& \quad - \int p(x_k | Z^{k-1}) \log p(x_k | Z^{k-1}) dx_k \\
& = -E [H(X_k | Z_k, u_k, Z^{k-1}) | Z^{k-1}, u_k] + H(X_k | Z^{k-1}) \tag{5.15}
\end{aligned}$$

and by noting that the term $H(X_k | Z^{k-1})$ does not depend on u_k , we can see

5. AN ANALYSIS OF INFORMATION-DRIVEN SENSOR MANAGEMENT CRITERIA

that maximizing

$$E \left[D \left(p \left(X_k \mid Z_k, u_k, Z^{k-1} \right) \parallel p \left(X_k \mid Z^{k-1} \right) \right) \mid Z^{k-1}, u_k \right]$$

over u_k is equivalent to maximizing

$$-E \left[H \left(X_k \mid Z_k, u_k, Z^{k-1} \right) \mid Z^{k-1}, u_k \right]. \quad (5.16)$$

In other words, for short-term sensor management (performed according to (2.77)) and for POM1DF systems, *minimizing the expected entropy and maximizing the expected KL divergence (with order of arguments as in (5.2)) leads to exactly the same sensing actions*. Therefore, we can say that in this case, the KL divergence and Shannon entropy criteria are equivalent¹.

In contrast, to the best of our knowledge, the KL divergence criterion with switched order of arguments (i.e. (5.3)), does not have this property of equivalence with entropy. Therefore, if, following our discussion in Section 5.3.1, we consider that minimization of entropy is desirable, the *asymmetry of the KL divergence with respect to its arguments is irrelevant* because there is a “correct” order of arguments to be used!

One may naturally ask whether we can obtain a similar relationship for the α -divergence and the Rényi entropy. Let $H_\alpha(X_k \mid Z^{k-1})$ and $H_\alpha(X_k \mid Z^k)$ be respectively the Rényi entropies of $p(x_k \mid Z^{k-1})$ and $p(x_k \mid Z^k)$. Let us now define

$$I_\alpha(X_k; Z_k \mid Z^{k-1}, u_k) \triangleq H_\alpha(X_k \mid Z^{k-1}) - E \left[H_\alpha(X_k \mid Z_k, u_k, Z^{k-1}) \mid Z^{k-1}, u_k \right]. \quad (5.17)$$

¹To the best of our knowledge, this equivalence between both criteria has not been explicitly stated elsewhere, although relations (5.14) and (5.15) have been known for quite a while, see e.g. Aughenbaugh and La Cour [2011]; Cover and Thomas [1991]; Williams [2007]. The equivalence is ignored in some works such as Zhao et al. [2002], where the two criteria are empirically compared (and even different results are obtained, due to the use of suboptimal heuristics)

5. AN ANALYSIS OF INFORMATION-DRIVEN SENSOR MANAGEMENT CRITERIA

Observe that

$$\begin{aligned}
 & I_\alpha (X_k; Z_k | Z^{k-1}, u_k) \\
 &= -\frac{1}{\alpha - 1} \log \left(\int p(x_k | Z^{k-1})^\alpha dx_k \right) + \frac{1}{\alpha - 1} \int p(z_k | Z^{k-1}, u_k) \\
 & \quad \times \log \left(\int p(x_k | Z^k)^\alpha dx_k \right) dz_k \tag{5.18}
 \end{aligned}$$

but the expected value of the α -divergence

$$D(p(X_k | Z^k) || p(X_k | Z^{k-1}))$$

is given by

$$\begin{aligned}
 & E [D(p(X_k | Z_k, u_k, Z^{k-1}) || p(X_k | Z^{k-1})) | Z^{k-1}, u_k] \\
 &= \frac{1}{\alpha - 1} \int p(z_k | Z^{k-1}, u_k) \log \left(\int p(x_k | Z^k)^\alpha p(x_k | Z^{k-1})^{1-\alpha} dx_k \right) dz_k. \tag{5.19}
 \end{aligned}$$

It is easy to verify empirically that (5.18) and (5.19) generally do not lead to the same results, except for $\alpha \rightarrow 1$ (in which case the Rényi entropy and the α -divergence become the Shannon entropy and the KL divergence respectively). Therefore, at least to the best of our knowledge, there is no equivalence between the maximum α -divergence and minimum Rényi entropy criteria for POM1DF systems in short-term sensor management. Some empirical results in literature Hero et al. [2007], however, indicate that both criteria may behave similarly from a practical point of view.

5.4 On the near-universal proxy argument for theoretical justification of information-driven sensor management

One argument in favor of information-driven sensor management is that the Rényi divergence is a “near-universal” proxy for arbitrary task-driven risk functions, implying that these could be replaced by a Rényi divergence-based criterion, and

5. AN ANALYSIS OF INFORMATION-DRIVEN SENSOR MANAGEMENT CRITERIA

this would likely result in similar performance. It is perhaps the strongest theoretical justification until so far for the use of information-driven sensor management, as it directly links it to task-driven sensor management. We will present, in this section, the mathematical formulation of the argument (from Hero et al. [2007]; Kreucher et al. [2005]) and its rebuttal.

5.4.1 The near-universal proxy argument

Let us consider a *task-driven sensor management* goal $\gamma(X_k, Z_k, U_k)$ (as in Section 2.4.2.1), and consider also the Rényi divergence

$$\mathcal{D}_\alpha \triangleq D_\alpha(p(X_k | Z^k) \| p(X_k | Z^{k-1}, u_k)) \quad (5.20)$$

where for a POM1DF process, from (5.13), we have

$$\mathcal{D}_\alpha = D_\alpha(p(X_k | Z^k) \| p(X_k | Z^{k-1})). \quad (5.21)$$

The near-universal proxy argument claims that the expectation of γ taken over the posterior $p(x_k | Z^k)$ admits the following “sandwich-inequality”:

$$\gamma_l \exp\left(-\frac{1-\alpha_2}{\alpha_2} \mathcal{D}_{\alpha_2}\right) \leq E[\gamma(X_k, z_k, u_k) | Z^k] \leq \gamma_u \exp\left(-\frac{1-\alpha_1}{\alpha_1} \mathcal{D}_{\alpha_1}\right) \quad (5.22)$$

where γ_l and γ_u are respectively the lower bound and upper bound of γ . There is a contradiction between Kreucher et al. [2005] and Hero et al. [2007] about the values of α_1 and α_2 ; we show below that Hero et al. [2007] has the correct values: $\alpha_1 > 1$ and $\alpha_2 \in [0, 1)$.

According to Hero et al. [2007], this “sandwich-inequality” would imply that the Rényi divergence is a “near-universal” proxy that performs nearly as well as task-specific optimal policies for a wide range of tasks. By maximizing an α -divergence with $\alpha < 1$, we would automatically minimize a lower bound on the value of arbitrary error functions. Therefore we could replace possibly complex and computationally expensive task-driven risk functions by a criterion based on α -divergences and expect satisfactory performance most of the time.

5.4.2 Rebuttal of the “near-universal” proxy argument

The “sandwich-inequality” (5.22) was used to justify the use of the Rényi divergence in a series of other works (Chong et al. [2009]; Hero and Kreucher [2007]; Hero et al. [2007]; Kreucher et al. [2007]). Since the “near-universal” proxy argument seems very strong given the properties of information-driven measures that we have been able to identify until so far, we will attempt to repeat the same derivations done in Kreucher et al. [2005] and check the validity of these results.

We start by assuming that the task-driven risk function $\gamma(X_k, Z_k, U_k)$ is non-negative, with an upper bound $\gamma_u < \infty$ and a lower bound $\gamma_l > 0$. Note that this assumption can already be quite restrictive; for instance, if $p(x_k | Z^k)$ is Gaussian and the risk function is the sum of the square errors with respect to some estimate (say, the mean), rigorously we do not have $\gamma_u < \infty$ as the support of the Gaussian distribution is unbounded.

Now, assuming that the system is a POM1DF process (and hence with property (5.13)), and by noting that the support of $p(x_k | Z^k)$ is contained in the support of $p(x_k | Z^{k-1})$, we have

$$\begin{aligned} E[\gamma(X_k, z_k, u_k) | Z^k] &= \int \gamma(x_k, z_k, u_k) p(x_k | Z^k) dx_k \\ &= \int \gamma(x_k, z_k, u_k) \frac{p(x_k | Z^k)}{p(x_k | Z^{k-1})} p(x_k | Z^{k-1}) dx_k \\ &= E \left[\gamma(X_k, z_k, u_k) \frac{p(X_k | Z^k)}{p(X_k | Z^{k-1})} \middle| Z^{k-1} \right]. \end{aligned} \quad (5.23)$$

Let us assume that $\alpha \in [0, 1)$. For any $a > 0$, a^α is a concave function, and

5. AN ANALYSIS OF INFORMATION-DRIVEN SENSOR MANAGEMENT CRITERIA

by applying Jensen's inequality to (5.23), we have

$$\begin{aligned}
E [\gamma(X_k, z_k, u_k) | Z^k] &\geq \left(E \left[\gamma(X_k, z_k, u_k)^\alpha \frac{p(X_k | Z^k)^\alpha}{p(X_k | Z^{k-1})^\alpha} \middle| Z^{k-1} \right] \right)^{\frac{1}{\alpha}} \\
&= \left(\int \gamma(x_k, z_k, u_k)^\alpha p(x_k | Z^k)^\alpha p(x_k | Z^{k-1})^{1-\alpha} dx_k \right)^{\frac{1}{\alpha}} \\
&\geq \gamma_1 \left(\int p(x_k | Z^k)^\alpha p(x_k | Z^{k-1})^{1-\alpha} dx_k \right)^{\frac{1}{\alpha}} \\
&= \gamma_1 \exp \left(\frac{1}{\alpha} \log \int p(x_k | Z^k)^\alpha p(x_k | Z^{k-1})^{1-\alpha} dx_k \right) \\
&= \gamma_1 \exp \left(\frac{\alpha - 1}{\alpha} \mathcal{D}_\alpha \right) \tag{5.24}
\end{aligned}$$

i.e. we have derived the left side of the sandwich inequality (5.22), and hence the correct values of α in the inequality are the ones presented in Hero et al. [2007], not in Kreucher et al. [2005]. However, because $D_\alpha \geq 0$ (α -divergences are nonnegative for $\alpha > 0$) and $\frac{\alpha-1}{\alpha} < 0$, we have

$$\gamma_1 \exp \left(\frac{\alpha - 1}{\alpha} \mathcal{D}_\alpha \right) \leq \gamma_1 \tag{5.25}$$

and because of the trivial relation

$$E [\gamma(X_k, z_k, u_k) | Z^k] \geq \gamma_1 \tag{5.26}$$

we observe that the bound (5.24) has no real significance, since regardless of the value of the Rényi divergence, it is always more loose than the trivial lower bound γ_1 . The same holds for $\alpha > 1$ and the upper bound in (5.22).

Hence, it is evident that the described bounds do not give any performance guarantees with respect to using α -divergences as replacement for task-driven sensor management criteria, and thus we cannot consider these bounds as justification for the near-universal proxy argument.

5.5 KL and Rényi divergences and “balancing estimation errors”

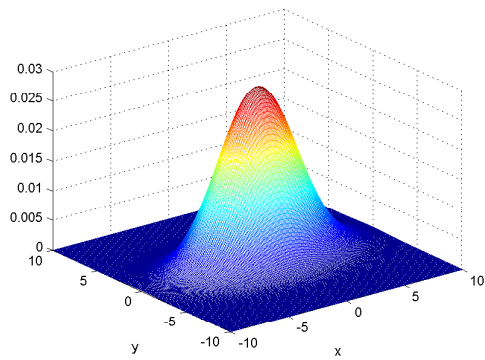
In sensor management context, a suggested reason (Aughenbaugh and La Cour [2008]) for using α -divergences with $\alpha < 1$ (i.e. *to not* use the KL divergence) is the apparent superior capability of these measures, in comparison to the KL divergence, of discriminating *morphological* differences between two distributions. More specifically, differences in aspects other than “overall uncertainty” (i.e. the Shannon entropy), like rotation around some axis, or relative distance between the modes of a multi-modal distribution.

As an example, Aughenbaugh and La Cour [2008] have empirically observed that in sensor management, using lower values of α seems to favor sensing actions that lead to rotation of the covariance matrix (with the prior $p(x_k | Z^{k-1})$ and the posterior $p(x_k | Z^k)$ assumed to be Gaussian), in comparison with actions that reduce the determinant (which as we have seen in Section (5.9), is proportional to the entropy). This property may be useful in practice. One example is the problem of tracking a target using bearing-only sensors, or any other sensor that has dramatically different resolution in different directions. For this type of problem, we might want to use sensor management not only to reduce average estimation errors, but also to avoid a large disparity between tracking errors in different directions, i.e. to “balance” tracking errors.

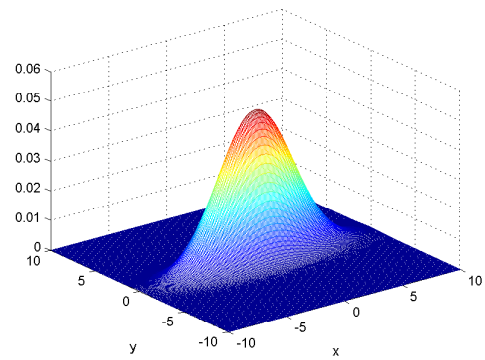
For this problem, sensor management criteria like minimum entropy or covariance determinant, although useful to reduce estimation errors in general, would not, in principle, guarantee this balance. As an example, suppose that the prior is the Gaussian distribution shown in Fig. 5.2(a). Observe that this distribution has variance in the x-axis much higher than in the y-axis. Now, consider that the Gaussian distributions shown in Figs. 5.2(b), 5.2(c) are alternative posteriors resulting from different sensing actions and observations. Both distributions have the same entropy and covariance determinant, but the one in Fig. 5.2(b) still has highly unbalanced errors; in fact, the variance in the x-axis is as high as before!

Therefore, for this problem it might be worthwhile considering to use a sensor management criterion that, whenever sensor management is performed, causes the posterior to be rotated in relation with the prior, in addition to reducing

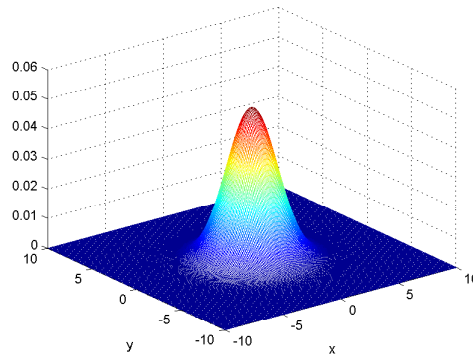
5. AN ANALYSIS OF INFORMATION-DRIVEN SENSOR MANAGEMENT CRITERIA



(a) Prior with unbalanced errors



(b) Posterior with unbalanced errors



(c) Posterior with balanced errors

Figure 5.2: Some bivariate Gaussian densities

5. AN ANALYSIS OF INFORMATION-DRIVEN SENSOR MANAGEMENT CRITERIA

overall uncertainty. Intuitively, this rotation would cause high errors in the prior, in some particular direction, to be considerably lower in the posterior, favoring the distribution in Fig. 5.2(c). For this purpose, the suitability of the KL divergence as a sensor management criterion is indeed questionable, since, as we mentioned in Section 5.3.2, the KL divergence sensor management criterion, under some common assumptions, behaves identically to the Shannon entropy sensor management criterion.

In the remainder of this section, we will conduct two empirical analyses to verify if α -divergences with $\alpha < 1$ are indeed useful to “balance” estimation errors. The first analysis will be an extension of the empirical analysis in Aughenbaugh and La Cour [2008], in order to obtain more solid conclusions about the ability of low α Rényi divergences to favor distribution rotations over uncertainty reduction. The second experiment will be a more practical experiment, where we will evaluate the performance of the α -divergences in the problem of tracking a target in two-dimensions using a mobile bearings-only sensor.

5.5.1 Rényi divergences and trade-off between uncertainty reduction and morphological changes

The empirical analysis in Aughenbaugh and La Cour [2008] was conducted by taking one Gaussian distribution (representing the prior $p(x_k | Z^{k-1})$), other five Gaussian distributions (representing the likelihood $p(z_k | x_k, u_k)$, each supposed to result from some sensing action u_k), and analyzing the effect of the α parameter on the expected α -divergence between the prior and the posteriors resulting from each sensing action. Due to the small number of likelihood functions considered, this analysis does not allow us to clearly identify the tradeoff between uncertainty reduction and morphological changes associated with the α -divergences. Therefore we will perform a complementary test, where we will only consider three different values of α but a much larger number of likelihood functions. As in Aughenbaugh and La Cour [2008], our analysis will be based on two-dimensional states.

Let us assume that the prior has the form $p(x_k | Z^{k-1}) = \mathcal{N}(x_k; \hat{x}_{k|k-1}, P_{k|k-1})$, and that the likelihood has the form $p(z_k | x_k, u_k) = \mathcal{N}(z_k; x_k, R_k)$. We consider

5. AN ANALYSIS OF INFORMATION-DRIVEN SENSOR MANAGEMENT CRITERIA

the likelihood to be a “scaled down” version of the prior when

$$R_k = cP_{k|k-1} \quad (5.27)$$

where $c \in (0, 1)$. In this case, from the Kalman filter formula, the covariance of the posterior will be given by

$$\begin{aligned} P_k &= P_{k|k-1} - P_{k|k-1} (P_{k|k-1} + cP_{k|k-1})^{-1} P_{k|k-1} \\ &= P_{k|k-1} - \frac{1}{1+c} P_{k|k-1} \\ &= \frac{c}{1+c} P_{k|k-1} \end{aligned} \quad (5.28)$$

i.e. such likelihood function will result in the posterior covariance having a smaller determinant than the prior (since $\frac{c}{1+c} < 1$). However, the balance of errors in different dimensions will be the same. For instance, if $P_{k|k-1}$ and P_k are given by

$$P_{k|k-1} = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix}, \quad P_k = \begin{bmatrix} \sigma_x'^2 & \sigma_{xy}' \\ \sigma_{xy}' & \sigma_y'^2 \end{bmatrix} \quad (5.29)$$

then for a scaled down likelihood, we will have $\frac{\sigma_y'^2}{\sigma_x'^2} = \frac{\sigma_y^2}{\sigma_x^2}$, and $\sigma_{xy}' = 0$.

We consider the likelihood to be a “rotated” version of the prior when

$$R_k = MP_{k|k-1}M^T \quad (5.30)$$

where M is an orthogonal matrix with $\det M = +1$. For the case of two-dimensional states, M has the form

$$M = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \quad (5.31)$$

where θ is the rotation angle. The covariance of the posterior will then be given by

$$P_k = P_{k|k-1} - P_{k|k-1} (P_{k|k-1} + MP_{k|k-1}M^T)^{-1} P_{k|k-1} \quad (5.32)$$

5. AN ANALYSIS OF INFORMATION-DRIVEN SENSOR MANAGEMENT CRITERIA

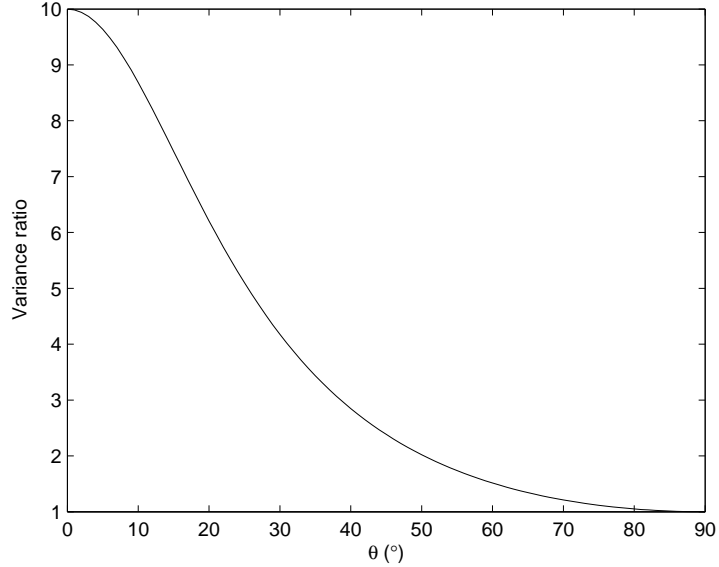


Figure 5.3: Variance ratio $\frac{\sigma_y'^2}{\sigma_x'^2}$ according to the rotation angle θ

and for $P_{k|k-1}$ and P_k given by (5.29), we have

$$\frac{\sigma_y'^2}{\sigma_x'^2} = \frac{2\frac{\sigma_y^2}{\sigma_x^2} - \frac{\sigma_y^2}{\sigma_x^2} \sin(\theta)^2 + \sin(\theta)^2}{\frac{\sigma_y^2}{\sigma_x^2} \sin(\theta)^2 - \sin(\theta)^2 + 2}. \quad (5.33)$$

Fig. 5.3 shows $\frac{\sigma_y'^2}{\sigma_x'^2}$ for $\frac{\sigma_y^2}{\sigma_x^2} = 10$ and varying values of the rotation angle θ . As we can clearly see, rotating the likelihood covariances leads to an increasingly high balance of errors in different dimensions. Hence, in principle, choosing sensing actions that favor rotated likelihoods over scaled down likelihoods would be beneficial in terms of balancing estimation errors.

We conduct two experiments, where in each experiment we compute the expected α -divergences (of the form $E[H_\alpha(X_k | Z^k) | Z^{k-1}, u_k]$) for two hypothetical sensing actions $u_k(1)$ and $u_k(2)$. We consider only the first form of the α -divergence (given by (5.6)), since given the discussion in Section 5.3.2, we would like to have equivalence of the KL divergence ($\alpha \rightarrow 1$) sensor management criterion with the Shannon entropy. Observe that assuming a POM1DF system, the

5. AN ANALYSIS OF INFORMATION-DRIVEN SENSOR MANAGEMENT CRITERIA

expression of the α -divergence (5.6) can be rewritten as

$$\begin{aligned}
 H_\alpha (X_k | Z^k) &= \frac{1}{\alpha - 1} \log \int p(x_k | Z^k)^\alpha p(x_k | Z^{k-1})^{1-\alpha} dx_k \\
 &= \frac{1}{\alpha - 1} \log \int \frac{p(z_k | x_k, u_k)^\alpha}{p(z_k | Z^{k-1}, u_k)^\alpha} p(x_k | Z^{k-1}) dx_k \\
 &= \frac{1}{\alpha - 1} \log \frac{\int p(z_k | x_k, u_k)^\alpha p(x_k | Z^{k-1}) dx_k}{\left(\int p(z_k | x_k, u_k) p(x_k | Z^{k-1}) dx_k\right)^\alpha}. \tag{5.34}
 \end{aligned}$$

which is a convenient expression as it does not require explicit calculation of the posterior density. The expected α -divergence may be then be easily approximated using Monte Carlo sampling. For each sensing action, we first generate N_S samples $x_k(i)$, $i = 1, \dots, N_S$ of the state X_k according to $p(x_k | Z^{k-1})$. For each $i = 1, \dots, N_S$, we then generate a sample $z_k(i)$ of Z_k according to $p(z_k | x_k(i), u_k)$. Observe that this procedure corresponds to using sequential sampling to generate N_S samples of Z_k according to $p(z_k | Z^{k-1}, u_k)$, since

$$p(z_k | Z^{k-1}, u_k) = \int p(z_k | x_k, u_k) p(x_k | Z^{k-1}) dx_k \tag{5.35}$$

and the expected α -divergence can then be evaluated using

$$\begin{aligned}
 &E [H_\alpha (X_k | Z^k) | Z^{k-1}, u_k] \\
 &= \int \frac{1}{\alpha - 1} \log \frac{\int p(z_k | x_k, u_k)^\alpha p(x_k | Z^{k-1}) dx_k}{\left(\int p(z_k | x_k, u_k) p(x_k | Z^{k-1}) dx_k\right)^\alpha} p(z_k | Z^{k-1}, u_k) dz_k \\
 &\approx \frac{1}{N_S} \sum_{i=1}^{N_S} \frac{1}{\alpha - 1} \log \frac{\sum_{j=1}^{N_S} p(z_k(i) | x_k(j), u_k)^\alpha}{\left(\sum_{j=1}^{N_S} p(z_k(i) | x_k(j), u_k)\right)^\alpha}. \tag{5.36}
 \end{aligned}$$

In the first experiment, we assume that sensing action $u_k(1)$ results in a rotated likelihood with $\theta = 90^\circ$, and that $u_k(2)$ results in a scaled likelihood where we vary the scaling factor c between 0.3 and 0.8. We then verify, for the values of α of 0.1, 0.5, 0.999 (where for the last value, the α -divergence corresponds practically to the KL divergence), which sensing action is preferred, i.e. leads to the largest value of the expected α -divergence. We use $N_S = 5,000$ samples to approximate the expected α -divergences, and the results, in terms of whether decision 1 or 2

5. AN ANALYSIS OF INFORMATION-DRIVEN SENSOR MANAGEMENT CRITERIA

is taken, are shown in Fig. 5.4(a). We can see, as intuitive, that for very low c all α -divergence criteria favor decision 2 (corresponding to scaling down the covariance), and for c close to 1 all criteria favor decision 1 (corresponding to the rotation). However, it is also clear that lower values of α lead to the rotation having a higher preference, as they require lower values of c in order to favor the scaling.

The second experiment is basically the opposite: $u_k(1)$ results in a scaled down likelihood with $c = 0.75$, and $u_k(2)$ results in a rotated likelihood where we vary the rotation angle θ between $\theta = 0$ and $\theta = 90^\circ$. The results are shown in Fig. 5.4(b). The behavior of the α -divergences is quite similar to the one verified in the first experiment: for θ close to zero, all criteria favor the scaling, whereas for θ close to 90° , all criteria favor the rotation, with lower values of α giving preference to the rotation. There are, however, some discontinuities in the decision curve, which indicates that the sensitivity of the expected α -divergence to θ is not particularly strong (recalling that the expected α -divergence is calculated using a numerical approximation).

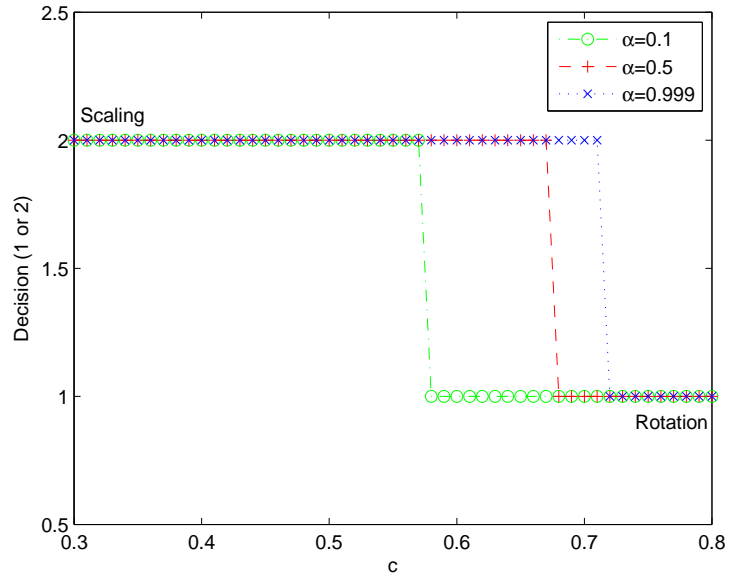
Regardless, the results of both experiments seem to confirm the suggestion in Aughenbaugh and La Cour [2008] that using lower values of α gives preference to sensing actions that lead to morphological changes in the posterior, over sensing actions that lead solely to reduction of overall uncertainty. Therefore, *in theory*, using a α -divergence with $\alpha < 1$ instead of the KL divergence might be advantageous for the aforementioned problem of balancing estimation errors in different dimensions. This is what we will try to verify in the next section.

5.5.2 A practical example: bearings-only tracking

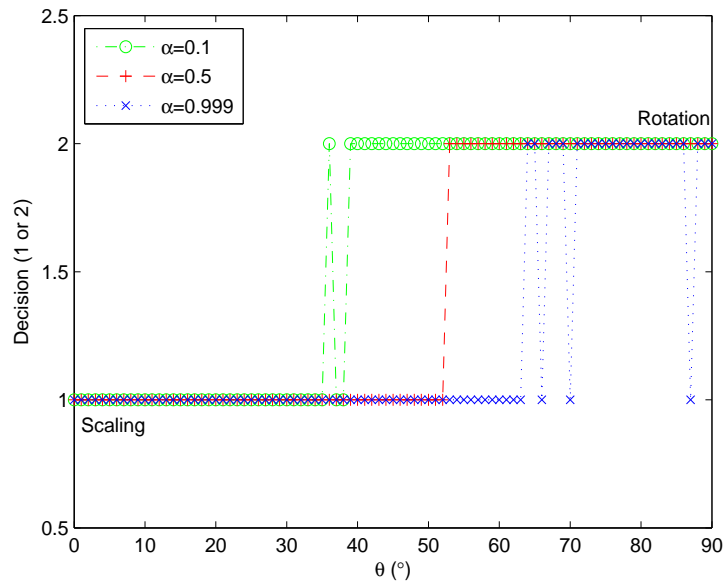
We now consider the problem of tracking using bearings-only sensors, where we will check whether the choice of the α parameter in the α -divergence has effect on sensor management, regarding the balancing of the estimation errors in different dimensions.

We consider a scenario where a moving target is observed by a Unmanned Aerial Vehicle (UAV) which attempts to maintain a fixed distance $d = 1000$ from the target. More precisely, it does so by maintaining a fixed distance to

5. AN ANALYSIS OF INFORMATION-DRIVEN SENSOR MANAGEMENT CRITERIA



(a) Experiment 1 (fixed rotation angle, varying scaling factor) results



(b) Experiment 2 (fixed scaling factor, varying rotation angle) results

Figure 5.4: α -divergence trade-off test results

5. AN ANALYSIS OF INFORMATION-DRIVEN SENSOR MANAGEMENT CRITERIA

the MMSE estimate of the target position calculated using the predictive density $p(x_k | Z^{k-1})$. We assume that, at each time step k , there are three sensing actions available:

1. To move the UAV such that its relative angle (to the MMSE estimate) is the same of the previous step $k - 1$;
2. To move the UAV such that its relative angle moves 5° clockwise from $k - 1$;
3. To move the UAV such that its relative angle moves 5° counterclockwise from $k - 1$.

The target motion model is the simple discretized white noise acceleration model described in Bar-Shalom et al. [2001], with the state vector given by $X_k = [P_k^x, P_k^y, V_k^x, V_k^y]^T$, where x and y denote the Cartesian coordinates, (P_x, P_y) corresponds to the position (with realizations denoted by (p_x, p_y)) and (V_x, V_y) corresponds to the velocity (with realizations denoted by (v_x, v_y)). The state transition density is given by (4.38), where we consider $T = 1$ as the interval between two observations and $\sigma^2 = 4900$ as the power spectral density of the process noise. The observations are the observed relative angles of the UAV w.r.t. to the targets, and they are modeled as

$$p(z_k | x_k, u_k) = \mathcal{N}\left(z_k; \arctan \frac{\tilde{p}_k^x - p_k^x}{p_k^y - \tilde{p}_k^y}, \sigma_b^2\right) \quad (5.37)$$

where $(\tilde{p}_k^x, \tilde{p}_k^y)$ is the true position of the sensor in Cartesian coordinates and $\sigma_b^2 = (1^\circ)^2$ is the variance of the measurement noise.

The target is tracked using a SIR PF with $N_P = 1,000$ particles, with the initial state of the target assumed to be known by the tracker. The target trajectory is a simple straight line, constant velocity trajectory with $x_0 = [0, 5000, 250, 0]^T$ and the sensor initial position is $(\tilde{p}_k^x, \tilde{p}_k^y) = (0, 4000)$. The expected sensor management reward, i.e. the expected α -divergence is calculated using sequential sampling based on the particles produced at the previous time step $k - 1$, as described in Section 2.4.4.

By using the α -divergence with $\alpha = 0.999$ as sensor management criterion, we obtain the results as given in Fig. 5.5. Fig. 5.5(a) shows both the trajectory

5. AN ANALYSIS OF INFORMATION-DRIVEN SENSOR MANAGEMENT CRITERIA

estimated by the tracker (using the MMSE) and the trajectory of the UAV determined on basis of sensor management. Fig. 5.5(b) shows the absolute position errors in the two dimensions. There is a noticeable unbalance of errors in the two dimensions, especially during the initial time steps. This is caused by the relative position of the bearings-only sensor to the target in those time steps.

What we would like to know is whether using lower values of α would give us a better balance of these errors, as implied by our analysis in Section 5.5.1. Unfortunately, for this experiment, the answer is no. Using $\alpha = 0.1$ or $\alpha = 0.5$ results in exactly the same sensing actions being chosen at all time steps as $\alpha = 0.999$. Hence, obviously, tuning the α parameter does not produce better results in terms of balancing estimation errors.

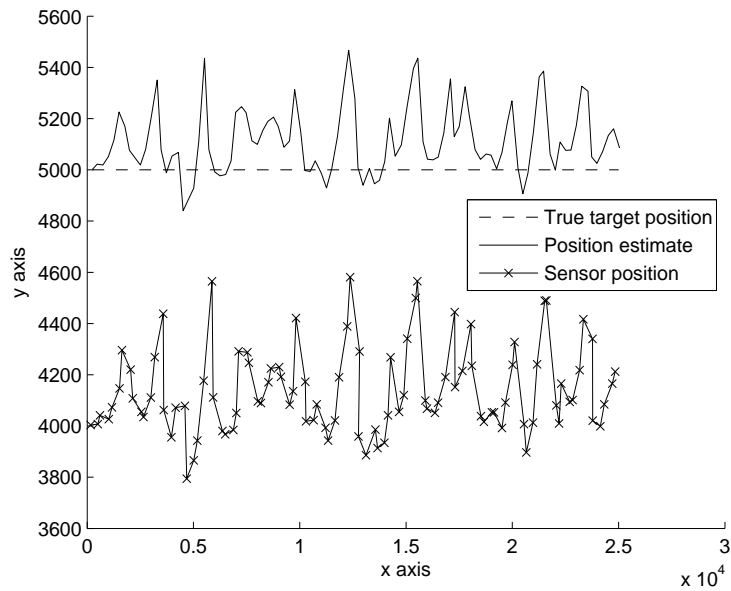
Therefore, although the results in Section 5.5.1 shows that using a lower value of α leads to a somewhat higher preference to distribution rotations, the sensitivity of the decision curve to the α parameter may not be strong enough to result in different performance on balancing estimation errors for practical applications (as also suggested by Fig. 5.4(b)).

5.6 Conclusions and recommendations

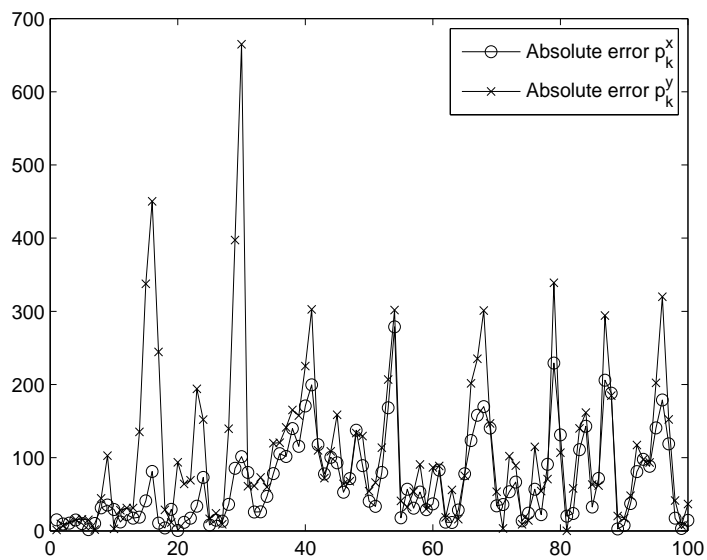
In this chapter, we examined several arguments for the theoretical justification of information-driven sensor management criteria. We have rebutted the strongest argument in favor of information-driven sensor management, namely, the “near-universal proxy” argument, meaning that the relationship between information-driven sensor management and task-driven performance is not simple as previously thought. However, we verified that the KL divergence and the Shannon entropy criteria are mutually linked, with the latter being an intuitive representation of the “overall uncertainty” contained in a distribution, at least for Gaussian and Gaussian mixture probability densities.

Possibly the main conclusion of our work is that the benefits of using the general Rényi divergences instead of the KL divergence for sensor management are not entirely clear. From empirical analyses, it seems that α -divergences with different values of α could be useful for problems where a better balance of estimation errors in multiple dimensions is desirable, but in practice, we were not

5. AN ANALYSIS OF INFORMATION-DRIVEN SENSOR MANAGEMENT CRITERIA



(a) Target, estimate and UAV position



(b) Position absolute errors in x and y coordinates

Figure 5.5: Results for bearings-only tracking and sensor management experiment

5. AN ANALYSIS OF INFORMATION-DRIVEN SENSOR MANAGEMENT CRITERIA

able to verify any difference in performance in a practical setting.

A possible subject of future work would be to analyze the properties of the KL divergence criterion for long-term sensor management (OL, OLF or CL), where the criterion is not, to the best of our knowledge, equivalent to the Shannon entropy criterion, as well as to analyze possible advantages of using general α -divergences instead of the KL divergence for such kind of problem.

Chapter 6

Conclusions and recommendations

In this thesis, we have studied two classes of estimation problems where characterization of uncertainty, using Sequential Monte Carlo (SMC) methods, is a difficult problem: joint state and parameter estimation, and Multi-Target Tracking and Labelling (MTTL). In addition, we have studied the problem of reducing estimation uncertainty through the use of information-driven sensor management.

For the joint state and parameter estimation problem, we have proposed two novel algorithms, more specifically, two new versions of the Rao-Blackwellized Marginal Particle Filter (RBMPF). The two proposed methods, the Discrete RBMPF (D-RBMPF) and the Monte Carlo RBMPF (MC-RBMPF) are applicable to general models and are designed to mitigate the bias caused by the introduction of artificial dynamics, typical of online Bayesian PF-based parameter estimation methods. The proposed algorithms have also nice parallelization properties. By examining their performance in two practical problems, we have seen that they can be an effective solution to online joint state and parameter estimation, in particular when good characterization of uncertainty is needed. In comparison with state-of-the-art techniques, however, these algorithms have higher computational cost. It is therefore highly desirable to find a RBMPF implementation with lower computational cost but still applicable to highly general models. One idea is to combine a SMC filter with a bank of parallel deterministic estimators (for instance, Extended or Unscented Kalman Filters). Other interesting topics of future research include the adaption of the proposed joint

state and parameter estimation algorithms for off-line processing, and studying the possibility of replacing the MPF needed by these algorithms by a regular SIR PF.

For the MTTL problem, we have theoretically shown how to perform optimal labelling and how to characterize the uncertainty associated with labelling in Bayesian context. These are particularly useful for the situation targets move in close proximity for a while, and afterwards separate. A recurring concern of this work was to define statistics with clear physical interpretation (i.e. that are meaningful to the user of the system), rather than only abstract mathematical quantities. Moreover, we have used the results of our theoretical analysis to propose a PF-like algorithm, the Labelling Uncertainty Aware PF (LUA-PF), that avoids the particle filter degeneracy phenomenon. The experimental results show that the proposed algorithm has superior labelling performance and is indeed far more suitable for characterizing labelling uncertainty than the “plain vanilla” particle filter implementation of the MTTL problem, i.e. the SIR M-SMC filter. Interesting future works include a more in-depth analysis of track extraction methods, integrating Bayesian labelling with non-Bayesian labelling (for large scale scenarios where we cannot formulate a well-posed Bayesian labelling problem, or where the LUA-PF is computationally unfeasible), and for the LUA-PF, finding more computationally efficient ways to calculate labelling probabilities, as well as test the algorithm with more complex observation models.

Finally, in the topic of sensor management, we have examined several arguments for the theoretical justification of information-driven sensor management criteria. We have rebutted the strongest argument in favor of information-driven sensor management, namely, the “near-universal proxy” argument, meaning that the relationship between information-driven sensor management and task-driven performance is not simple as previously thought. However, we verified that the KL divergence and the Shannon entropy criteria are mutually linked, with the latter being an intuitive representation of the “overall uncertainty” contained in a distribution. Possibly the main conclusion of our work is that the benefits of using the general Rényi divergences instead of the KL divergence for sensor management are not entirely clear. From empirical analyses, it seems that α -divergences with different values of α could be useful for problems where a better

balance of estimation errors in multiple dimensions is desirable, but in practice, we were not able to verify any difference in performance in a practical setting. A possible subject of future work would be to analyze the properties of the KL divergence criterion for long-term sensor management, where the criterion is not, to the best of our knowledge, equivalent to the Shannon entropy criterion, as well as to analyze possible advantages of using general α -divergences instead of the KL divergence for such kind of problem.

Appendix A

The optimal proposal density of the MPF

In Section 2.1.5 we have presented the Marginal Particle Filter (MPF) algorithm and explained, in the context of the algorithm, how to sample from the blind proposal density, which disregards the last observation. In this appendix, we will derive the optimal proposal density for the MPF, i.e. the proposal density that minimizes the variance of the weights, for a POM1 process.

From (2.31), the minimum variance of the weights (more precisely zero) would be achieved if we could sample directly from $p(x_k | Z^k)$. While this cannot be

accomplished for obvious reasons, we can approximate $p(x_k | Z^k)$ as

$$\begin{aligned}
p(x_k | Z^k) &\approx \frac{p(z_k | x_k) \sum_{j=1}^{N_P} w_{k-1}(j) p(x_k | x_{k-1}(j))}{p(z_k | Z^{k-1})} \\
&= \sum_{j=1}^{N_P} \frac{w_{k-1}(j) p(z_k | x_k, x_{k-1}(j)) p(x_k | x_{k-1}(j))}{p(z_k | Z^{k-1})} \\
&= \sum_{j=1}^{N_P} \frac{w_{k-1}(j) p(z_k | x_{k-1}(j)) p(x_k | x_{k-1}(j), z_k)}{p(z_k | Z^{k-1})} \\
&= \sum_{j=1}^{N_P} \lambda_k(j) p(x_k | x_{k-1}(j), z_k) \tag{A.1}
\end{aligned}$$

where

$$\lambda_k(j) = \frac{w_{k-1}(j) p(z_k | x_{k-1}(j))}{p(z_k | Z^{k-1})}. \tag{A.2}$$

Now, observe that

$$\begin{aligned}
p(z_k | x_{k-1}(j)) &= \int p(z_k | x_k, x_{k-1}(j)) p(x_k | x_{k-1}(j)) dx_k \\
&= \int p(z_k | x_k) p(x_k | x_{k-1}(j)) dx_k \tag{A.3}
\end{aligned}$$

such that we can compute $p(z_k | x_{k-1}(j))$ by generating some number (say N_S) of samples $x_k(j')$, $j' = 1, \dots, N_S$ according to $p(x_k | x_{k-1}(j))$, and performing the approximation

$$p(z_k | x_{k-1}(j)) \approx \frac{1}{N_S} \sum_{j'=1}^{N_S} p(z_k | x_k(j')). \tag{A.4}$$

Therefore, the most significant challenge of performing optimal importance sampling in a MPF is to sample from $p(x_k | x_{k-1}, z_k)$, which as we have seen in Section 2.1.3, is also the optimal proposal density of the SIR PF.

Appendix B

An in-depth analysis of the mixed labelling phenomenon in two-target tracking

In this appendix, we will use the formulas and statistics for the Bayesian Multi-Target Tracking and Labelling (MTTL) problem (described in Sections 4.2 and 4.3) to analyze the mixed labelling phenomenon in a simple scenario, namely the case where there are two targets, with no possibility of targets births or deaths. In particular, we will show, for this scenario, how the mixed labelling phenomenon arises in the situation of closely spaced targets, and how (and when) it persists after target separation.

B.1 The MTTL Bayesian recursion in the two-target case

We will first derive some useful formulas to describe the MTTL Bayesian recursion in the two-target scenario. Consider the mathematical formulation of

B. AN IN-DEPTH ANALYSIS OF THE MIXED LABELLING PHENOMENON IN TWO-TARGET TRACKING

the Bayesian MTTL problem in Section 4.2. Assuming that there are two targets, that the number of targets is known, and that there are no target births or deaths, then $\mathbf{X}_k = \{X_k^{(1)}, X_k^{(2)}\}$ is the RFS describing the labelled states, $\mathbf{S}_k = \{S_k^{(1)}, S_k^{(2)}\}$ is the RFS describing the unlabelled states (locations), and $\mathbf{L}_k = \{L_k^{(1)}, L_k^{(2)}\}$ is the RFS describing the labels. Without loss of generality, we also assume that a target's label is either A or B .

First, let us define

$$p_{AB} \left(s_k^{(1)}, s_k^{(2)} \right) \triangleq p_l \left(\left(\left\{ \begin{bmatrix} s_k^{(1)} \\ A \end{bmatrix}, \begin{bmatrix} s_k^{(2)} \\ B \end{bmatrix} \right\} \middle| \left\{ s_k^{(1)}, s_k^{(2)} \right\}, Z^k \right), \quad (\text{B.1})$$

$$p_{BA} \left(s_k^{(1)}, s_k^{(2)} \right) \triangleq p_l \left(\left(\left\{ \begin{bmatrix} s_k^{(1)} \\ B \end{bmatrix}, \begin{bmatrix} s_k^{(2)} \\ A \end{bmatrix} \right\} \middle| \left\{ s_k^{(1)}, s_k^{(2)} \right\}, Z^k \right), \quad (\text{B.2})$$

$$p_{AB} \left(s_{k-1}^{(1)}, s_{k-1}^{(2)} \right) \triangleq p_l \left(\left(\left\{ \begin{bmatrix} s_{k-1}^{(1)} \\ A \end{bmatrix}, \begin{bmatrix} s_{k-1}^{(2)} \\ B \end{bmatrix} \right\} \middle| \left\{ s_{k-1}^{(1)}, s_{k-1}^{(2)} \right\}, Z^k \right), \quad (\text{B.3})$$

$$p_{BA} \left(s_{k-1}^{(1)}, s_{k-1}^{(2)} \right) \triangleq p_l \left(\left(\left\{ \begin{bmatrix} s_{k-1}^{(1)} \\ B \end{bmatrix}, \begin{bmatrix} s_{k-1}^{(2)} \\ A \end{bmatrix} \right\} \middle| \left\{ s_{k-1}^{(1)}, s_{k-1}^{(2)} \right\}, Z^k \right) \quad (\text{B.4})$$

where p_l denotes the labelling probability (as defined in Definition 4.3.1). Using (4.31), we have

$$\begin{aligned} & p_{AB} \left(s_k^{(1)}, s_k^{(2)} \right) \\ &= \frac{1}{f \left(\left\{ s_k^{(1)}, s_k^{(2)} \right\} \middle| Z^{k-1} \right)} \int f \left(\left(\left\{ \begin{bmatrix} s_k^{(1)} \\ A \end{bmatrix}, \begin{bmatrix} s_k^{(2)} \\ B \end{bmatrix} \right\} \middle| \mathbf{x}_{k-1} \right) f \left(\mathbf{x}_{k-1} \middle| Z^{k-1} \right) \delta \mathbf{x}_{k-1} \\ &= \frac{1}{f \left(\left\{ s_k^{(1)}, s_k^{(2)} \right\} \middle| Z^{k-1} \right)} \left(p_{AB} \left(s_k^{(1)}, s_k^{(2)} \right) + p_{BA} \left(s_k^{(1)}, s_k^{(2)} \right) \right) \end{aligned} \quad (\text{B.5})$$

where, assuming without loss of generality that location $S_k^{(i)}$ assumes values in

B. AN IN-DEPTH ANALYSIS OF THE MIXED LABELLING PHENOMENON IN TWO-TARGET TRACKING

an Euclidean space \mathbb{R}^n , we have

$$\begin{aligned}
 p_{AB}^{AB} \left(s_k^{(1)}, s_k^{(2)} \right) &= \int \int f \left(\left\{ \begin{bmatrix} s_k^{(1)} \\ A \end{bmatrix}, \begin{bmatrix} s_k^{(2)} \\ B \end{bmatrix} \right\} \middle| \left\{ \begin{bmatrix} s_{k-1}^{(1)} \\ A \end{bmatrix}, \begin{bmatrix} s_{k-1}^{(2)} \\ B \end{bmatrix} \right\} \right) \\
 &\quad \times f \left(\left\{ \begin{bmatrix} s_{k-1}^{(1)} \\ A \end{bmatrix}, \begin{bmatrix} s_{k-1}^{(2)} \\ B \end{bmatrix} \right\} \middle| Z^{k-1} \right) ds_{k-1}^{(1)} ds_{k-1}^{(2)}, \quad (\text{B.6})
 \end{aligned}$$

$$\begin{aligned}
 p_{AB}^{BA} \left(s_k^{(1)}, s_k^{(2)} \right) &= \int \int f \left(\left\{ \begin{bmatrix} s_k^{(1)} \\ A \end{bmatrix}, \begin{bmatrix} s_k^{(2)} \\ B \end{bmatrix} \right\} \middle| \left\{ \begin{bmatrix} s_{k-1}^{(1)} \\ B \end{bmatrix}, \begin{bmatrix} s_{k-1}^{(2)} \\ A \end{bmatrix} \right\} \right) \\
 &\quad \times f \left(\left\{ \begin{bmatrix} s_{k-1}^{(1)} \\ B \end{bmatrix}, \begin{bmatrix} s_{k-1}^{(2)} \\ A \end{bmatrix} \right\} \middle| Z^{k-1} \right) ds_{k-1}^{(1)} ds_{k-1}^{(2)}. \quad (\text{B.7})
 \end{aligned}$$

From Definition 4.3.1, we have

$$f \left(\left\{ \begin{bmatrix} s_{k-1}^{(1)} \\ A \end{bmatrix}, \begin{bmatrix} s_{k-1}^{(2)} \\ B \end{bmatrix} \right\} \middle| Z^{k-1} \right) = p_{AB} \left(s_{k-1}^{(1)}, s_{k-1}^{(2)} \right) f \left(\left\{ s_{k-1}^{(1)}, s_{k-1}^{(2)} \right\} \middle| Z^{k-1} \right), \quad (\text{B.8})$$

$$f \left(\left\{ \begin{bmatrix} s_{k-1}^{(1)} \\ B \end{bmatrix}, \begin{bmatrix} s_{k-1}^{(2)} \\ A \end{bmatrix} \right\} \middle| Z^{k-1} \right) = p_{BA} \left(s_{k-1}^{(1)}, s_{k-1}^{(2)} \right) f \left(\left\{ s_{k-1}^{(1)}, s_{k-1}^{(2)} \right\} \middle| Z^{k-1} \right) \quad (\text{B.9})$$

and substituting (4.7), (B.8), (B.9) into (B.6), (B.7), we obtain

$$\begin{aligned}
 p_{AB}^{AB} \left(s_k^{(1)}, s_k^{(2)} \right) &= \int \int p \left(s_k^{(1)} \middle| s_{k-1}^{(1)} \right) p \left(s_k^{(2)} \middle| s_{k-1}^{(2)} \right) p_{AB} \left(s_{k-1}^{(1)}, s_{k-1}^{(2)} \right) \\
 &\quad \times f \left(\left\{ s_{k-1}^{(1)}, s_{k-1}^{(2)} \right\} \middle| Z^{k-1} \right) ds_{k-1}^{(1)} ds_{k-1}^{(2)} \quad (\text{B.10})
 \end{aligned}$$

$$\begin{aligned}
 p_{AB}^{BA} \left(s_k^{(1)}, s_k^{(2)} \right) &= \int \int p \left(s_k^{(1)} \middle| s_{k-1}^{(2)} \right) p \left(s_k^{(2)} \middle| s_{k-1}^{(1)} \right) p_{BA} \left(s_{k-1}^{(1)}, s_{k-1}^{(2)} \right) \\
 &\quad \times f \left(\left\{ s_{k-1}^{(1)}, s_{k-1}^{(2)} \right\} \middle| Z^{k-1} \right) ds_{k-1}^{(1)} ds_{k-1}^{(2)}. \quad (\text{B.11})
 \end{aligned}$$

B. AN IN-DEPTH ANALYSIS OF THE MIXED LABELLING PHENOMENON IN TWO-TARGET TRACKING

Analogously, we can show that

$$p_{BA} \left(s_k^{(1)}, s_k^{(2)} \right) = \frac{1}{f \left(\left\{ s_k^{(1)}, s_k^{(2)} \right\} \middle| Z^{k-1} \right)} \left(p_{BA}^{AB} \left(s_k^{(1)}, s_k^{(2)} \right) + p_{BA}^{BA} \left(s_k^{(1)}, s_k^{(2)} \right) \right) \quad (\text{B.12})$$

where

$$p_{BA}^{AB} \left(s_k^{(1)}, s_k^{(2)} \right) = \int \int p \left(s_k^{(1)} \middle| s_{k-1}^{(2)} \right) p \left(s_k^{(2)} \middle| s_{k-1}^{(1)} \right) p_{AB} \left(s_{k-1}^{(1)}, s_{k-1}^{(2)} \right) \times f \left(\left\{ s_{k-1}^{(1)}, s_{k-1}^{(2)} \right\} \middle| Z^{k-1} \right) ds_{k-1}^{(1)} ds_{k-1}^{(2)} \quad (\text{B.13})$$

$$p_{BA}^{BA} \left(s_k^{(1)}, s_k^{(2)} \right) = \int \int p \left(s_k^{(1)} \middle| s_{k-1}^{(1)} \right) p \left(s_k^{(2)} \middle| s_{k-1}^{(2)} \right) p_{BA} \left(s_{k-1}^{(1)}, s_{k-1}^{(2)} \right) \times f \left(\left\{ s_{k-1}^{(1)}, s_{k-1}^{(2)} \right\} \middle| Z^{k-1} \right) ds_{k-1}^{(1)} ds_{k-1}^{(2)}. \quad (\text{B.14})$$

B.2 Origin of mixed labelling

In multi-target tracking, if \mathfrak{N}_0^S denotes the single-target location state space (for instance, \mathbb{R}^4 if the single-target state corresponds to the position and the velocity in Cartesian coordinates), given a sufficiently large sequence of measurements Z^{k-1} , the Belief mass associated with $f \left(\left\{ s_{k-1}^{(1)}, s_{k-1}^{(2)} \right\} \middle| Z^{k-1} \right)$ will be mostly contained in a small subset of \mathfrak{N}_0^S , say \mathfrak{N}_*^S . Effectively, the double integrals in (B.10), (B.11), (B.13) and (B.14) are all taken over $\mathfrak{N}_*^S \times \mathfrak{N}_*^S$, and given good observability conditions, \mathfrak{N}_*^S will be formed by the regions surrounding the true target states.

But if the targets are moving in close proximity with each other, we will have $s_{k-1}^{(1)} \approx s_{k-1}^{(2)}$ for $s_{k-1}^{(1)}, s_{k-1}^{(2)} \in \mathfrak{N}_*^S$, and hence, within \mathfrak{N}_*^S , we will have

$$p \left(s_k^{(1)} \middle| s_{k-1}^{(1)} \right) p \left(s_k^{(2)} \middle| s_{k-1}^{(2)} \right) \approx p \left(s_k^{(1)} \middle| s_{k-1}^{(2)} \right) p \left(s_k^{(2)} \middle| s_{k-1}^{(1)} \right). \quad (\text{B.15})$$

B. AN IN-DEPTH ANALYSIS OF THE MIXED LABELLING PHENOMENON IN TWO-TARGET TRACKING

As a consequence, from (B.10), (B.11), (B.13) and (B.14), we have

$$\begin{aligned} p_{AB}^{AB} \left(s_k^{(1)}, s_k^{(2)} \right) &\approx p_{BA}^{AB} \left(s_k^{(1)}, s_k^{(2)} \right) \\ p_{AB}^{BA} \left(s_k^{(1)}, s_k^{(2)} \right) &\approx p_{BA}^{BA} \left(s_k^{(1)}, s_k^{(2)} \right) \end{aligned}$$

and from (B.5), (B.12)

$$p_{AB} \left(s_k^{(1)}, s_k^{(2)} \right) \approx p_{BA} \left(s_k^{(1)}, s_k^{(2)} \right) \quad (\text{B.16})$$

i.e. there will be “total mixed labelling” as described in Section 4.3.1. Interestingly, (B.16) will hold regardless of the values of $s_k^{(1)}, s_k^{(2)}$, implying that total mixed labelling will affect the entire state space of $\mathbf{S}_k = \left\{ s_k^{(1)}, s_k^{(2)} \right\}$.

Note that, if the targets are reasonably close to each other, but not that much given observability conditions, the most likely result will be some degree of “partial mixed labelling” (as described in Section 4.3.1) instead.

B.3 Persistence of mixed labelling

If “total mixed labelling” (i.e. the situation given by (B.16)) affects the entire space of \mathbf{S}_k , it can be shown that the situation will persist indefinitely, even after the targets separate from each other. In order to see that, let us suppose that we are at time $k - 1$, and we have $p_{AB} \left(s_{k-1}^{(1)}, s_{k-1}^{(2)} \right) \approx p_{BA} \left(s_{k-1}^{(1)}, s_{k-1}^{(2)} \right)$ for $s_{k-1}^{(1)}, s_{k-1}^{(2)} \in \mathfrak{N}_*^S$.

For the next time step (k), from (B.10), (B.11), (B.13), (B.14) we will have $p_{AB}^{AB} \left(s_k^{(1)}, s_k^{(2)} \right) \approx p_{BA}^{BA} \left(s_k^{(1)}, s_k^{(2)} \right)$ and $p_{AB}^{BA} \left(s_k^{(1)}, s_k^{(2)} \right) \approx p_{BA}^{AB} \left(s_k^{(1)}, s_k^{(2)} \right)$, and from (B.5), (B.12), it is easy to see that (B.16) will hold at time k . As result, total mixed labelling propagates from time $k - 1$ to time k , regardless of the observation \mathbf{z}_k .

If we have instead “partial mixed labelling”, it is possible that mixed labelling disappears with time. An interesting question, however, is whether partial mixed labelling may disappear after the targets become well-separated again. To illus-

B. AN IN-DEPTH ANALYSIS OF THE MIXED LABELLING PHENOMENON IN TWO-TARGET TRACKING

trate this situation, let us assume that at time $k - 1$, we have $\mathfrak{R}_*^S = \Omega^{(1)} \cup \Omega^{(2)}$, with $\Omega^{(1)} \cap \Omega^{(2)} = \emptyset$, which would be the case if the targets are well-separated and $\Omega^{(1)}, \Omega^{(2)}$ are the regions surrounding each of the true target states.

We also assume that $\Omega^{(1)}$ and $\Omega^{(2)}$ are small enough such that the probability (conditioned on Z^{k-1}) that a certain element of $\Omega^{(1)}$ corresponds to A and that a certain element of $\Omega^{(2)}$ corresponds to B is more-or-less constant and equal to P_{k-1} . Conversely, P_{k-1}^* would be the probability that an element of $\Omega^{(2)}$ corresponds to A and an element of $\Omega^{(1)}$ corresponds to B .

Since the targets are well-separated, clearly (B.15) does not hold. Instead, a more reasonable assumption is that, without loss of generality, for given $s_k^{(1)}, s_k^{(2)}$

$$p\left(s_k^{(1)} \mid s_{k-1}^{(1)}\right) p\left(s_k^{(2)} \mid s_{k-1}^{(2)}\right) \gg p\left(s_k^{(1)} \mid s_{k-1}^{(2)}\right) p\left(s_k^{(2)} \mid s_{k-1}^{(1)}\right)$$

if $s_{k-1}^{(1)} \in \Omega^{(1)}$ and $s_{k-1}^{(2)} \in \Omega^{(2)}$, and

$$p\left(s_k^{(1)} \mid s_{k-1}^{(1)}\right) p\left(s_k^{(2)} \mid s_{k-1}^{(2)}\right) \ll p\left(s_k^{(1)} \mid s_{k-1}^{(2)}\right) p\left(s_k^{(2)} \mid s_{k-1}^{(1)}\right) \quad (\text{B.17})$$

if $s_{k-1}^{(1)} \in \Omega^{(2)}$ and $s_{k-1}^{(2)} \in \Omega^{(1)}$. From (B.10), (B.11), (B.13) and (B.14), we then

B. AN IN-DEPTH ANALYSIS OF THE MIXED LABELLING PHENOMENON IN TWO-TARGET TRACKING

have

$$\begin{aligned}
 p_{AB}^{AB} \left(s_k^{(1)}, s_k^{(2)} \right) &= \int_{\Omega^{(2)}} \int_{\Omega^{(1)}} p \left(s_k^{(1)} \mid s_{k-1}^{(1)} \right) p \left(s_k^{(2)} \mid s_{k-1}^{(2)} \right) p_{AB} \left(s_{k-1}^{(1)}, s_{k-1}^{(2)} \right) \\
 &\quad \times f \left(\left\{ s_{k-1}^{(1)}, s_{k-1}^{(2)} \right\} \mid Z^{k-1} \right) ds_{k-1}^{(1)} ds_{k-1}^{(2)}, \tag{B.18}
 \end{aligned}$$

$$\begin{aligned}
 p_{AB}^{BA} \left(s_k^{(1)}, s_k^{(2)} \right) &= \int_{\Omega^{(1)}} \int_{\Omega^{(2)}} p \left(s_k^{(1)} \mid s_{k-1}^{(2)} \right) p \left(s_k^{(2)} \mid s_{k-1}^{(1)} \right) p_{BA} \left(s_{k-1}^{(1)}, s_{k-1}^{(2)} \right) \\
 &\quad \times f \left(\left\{ s_{k-1}^{(1)}, s_{k-1}^{(2)} \right\} \mid Z^{k-1} \right) ds_{k-1}^{(1)} ds_{k-1}^{(2)}, \tag{B.19}
 \end{aligned}$$

$$\begin{aligned}
 p_{BA}^{AB} \left(s_k^{(1)}, s_k^{(2)} \right) &= \int_{\Omega^{(1)}} \int_{\Omega^{(2)}} p \left(s_k^{(1)} \mid s_{k-1}^{(2)} \right) p \left(s_k^{(2)} \mid s_{k-1}^{(1)} \right) p_{AB} \left(s_{k-1}^{(1)}, s_{k-1}^{(2)} \right) \\
 &\quad \times f \left(\left\{ s_{k-1}^{(1)}, s_{k-1}^{(2)} \right\} \mid Z^{k-1} \right) ds_{k-1}^{(1)} ds_{k-1}^{(2)}, \tag{B.20}
 \end{aligned}$$

$$\begin{aligned}
 p_{BA}^{BA} \left(s_k^{(1)}, s_k^{(2)} \right) &= \int_{\Omega^{(2)}} \int_{\Omega^{(1)}} p \left(s_k^{(1)} \mid s_{k-1}^{(1)} \right) p \left(s_k^{(2)} \mid s_{k-1}^{(2)} \right) p_{BA} \left(s_{k-1}^{(1)}, s_{k-1}^{(2)} \right) \\
 &\quad \times f \left(\left\{ s_{k-1}^{(1)}, s_{k-1}^{(2)} \right\} \mid Z^{k-1} \right) ds_{k-1}^{(1)} ds_{k-1}^{(2)}. \tag{B.21}
 \end{aligned}$$

Using the probabilities P_{k-1} and P_{k-1}^* that we have defined, we obtain

$$\begin{aligned}
 p_{AB}^{AB} \left(s_k^{(1)}, s_k^{(2)} \right) &\approx P_{k-1} \int_{\Omega^{(2)}} \int_{\Omega^{(1)}} p \left(s_k^{(1)} \mid s_{k-1}^{(1)} \right) p \left(s_k^{(2)} \mid s_{k-1}^{(2)} \right) \\
 &\quad \times f \left(\left\{ s_{k-1}^{(1)}, s_{k-1}^{(2)} \right\} \mid Z^{k-1} \right) ds_{k-1}^{(1)} ds_{k-1}^{(2)} \tag{B.22}
 \end{aligned}$$

$$\begin{aligned}
 p_{AB}^{BA} \left(s_k^{(1)}, s_k^{(2)} \right) &\approx P_{k-1} \int_{\Omega^{(1)}} \int_{\Omega^{(2)}} p \left(s_k^{(1)} \mid s_{k-1}^{(2)} \right) p \left(s_k^{(2)} \mid s_{k-1}^{(1)} \right) \\
 &\quad \times f \left(\left\{ s_{k-1}^{(1)}, s_{k-1}^{(2)} \right\} \mid Z^{k-1} \right) ds_{k-1}^{(1)} ds_{k-1}^{(2)} \tag{B.23}
 \end{aligned}$$

$$\begin{aligned}
 p_{BA}^{AB} \left(s_k^{(1)}, s_k^{(2)} \right) &\approx P_{k-1}^* \int_{\Omega^{(1)}} \int_{\Omega^{(2)}} p \left(s_k^{(1)} \mid s_{k-1}^{(2)} \right) p \left(s_k^{(2)} \mid s_{k-1}^{(1)} \right) \\
 &\quad \times f \left(\left\{ s_{k-1}^{(1)}, s_{k-1}^{(2)} \right\} \mid Z^{k-1} \right) ds_{k-1}^{(1)} ds_{k-1}^{(2)} \tag{B.24}
 \end{aligned}$$

$$\begin{aligned}
 p_{BA}^{BA} \left(s_k^{(1)}, s_k^{(2)} \right) &\approx P_{k-1}^* \int_{\Omega^{(2)}} \int_{\Omega^{(1)}} p \left(s_k^{(1)} \mid s_{k-1}^{(1)} \right) p \left(s_k^{(2)} \mid s_{k-1}^{(2)} \right) \\
 &\quad \times f \left(\left\{ s_{k-1}^{(1)}, s_{k-1}^{(2)} \right\} \mid Z^{k-1} \right) ds_{k-1}^{(1)} ds_{k-1}^{(2)} \tag{B.25}
 \end{aligned}$$

B. AN IN-DEPTH ANALYSIS OF THE MIXED LABELLING PHENOMENON IN TWO-TARGET TRACKING

and finally

$$p_{AB} \left(s_k^{(1)}, s_k^{(2)} \right) \propto P_{k-1}, \quad (\text{B.26})$$

$$p_{BA} \left(s_k^{(1)}, s_k^{(2)} \right) \propto P_{k-1}^* \quad (\text{B.27})$$

also regardless of the values of $s_k^{(1)}$ and $s_k^{(2)}$. Therefore, although partial mixed labelling may disappear with time, this will generally not happen after the targets are separated enough. Rather, the labelling probabilities will converge to constant values.

B.4 Mixed labelling and non-kinematic states

It is possible that we have a situation of closely spaced targets, but mixed labelling does not arise. This may happen when the location $S_k^{(i)}$ contains entries corresponding to non-kinematic quantities, such as the target's classification, or the target's Identification Friend-or-Foe (IFF) code, or the callsign attributed by the Air Traffic Control. Let $N_k^{(i)}$ be this non-kinematic quantity. Typically, $N_k^{(i)}$ has a very small (or zero) probability of changing between two subsequent time steps.

As consequence, the condition for the appearance of mixed labelling (B.15) may only hold if $s_k^{(1)}$ and $s_k^{(2)}$ contain the same value of $N_k^{(i)}$. If two targets do not share the same value for $N_k^{(i)}$, and we can effectively estimate this quantity (for instance, when the targets' callsigns are provided in the observations), then a pair of locations in a high probability area (say, the unlabelled tracks $\{\hat{s}_k^{(1)}, \hat{s}_k^{(2)}\}$) should also contain different values of $N_k^{(i)}$. As a consequence, there will be no mixed labelling associated with $\{\hat{s}_k^{(1)}, \hat{s}_k^{(2)}\}$.

Appendix C

The “one-sided” decoupling between tracking and labelling in Bayesian multi-target tracking

In this appendix, we will show that in Bayesian MTTL, the concept of labelling probabilities allows us to partially decouple the tracking sub-problem (in the sense of estimating the set of locations) from the labelling sub-problem. Specifically, the decoupling is “one-sided”: tracking does not depend on the labelling, but labelling depends on the results of tracking.

C.1 The tracking sub-problem

Consider the mathematical formulation of the Bayesian MTTL problem in Section 4.2. We are going to show that, given some nonrestrictive assumptions, the tracking sub-problem (i.e. the problem of estimating the set of locations $\mathbf{S}_k = \{S_k^{(1)}, \dots, S_k^{(t)}\}$ from a sequence of observations $Z^k = (\mathbf{z}_1, \dots, \mathbf{z}_k)$) is completely independent from the labelling sub-problem.

C. THE “ONE-SIDED” DECOUPLING BETWEEN TRACKING AND LABELLING IN BAYESIAN MULTI-TARGET TRACKING

Definition C.1.1 Consider two identical size vectors $s_k = [s_k^{(1)}, \dots, s_k^{(t_k)}]'$ and $l_k = [l_k^{(1)}, \dots, l_k^{(t_k)}]'$. The **set-composition** of vectors s_k and l_k is defined as

$$h_{S^{(\cdot)}, L^{(\cdot)}}(s_k, l_k) \triangleq \left\{ \left[\begin{array}{c} s_k^{(1)} \\ l_k^{(1)} \end{array} \right], \dots, \left[\begin{array}{c} s_k^{(t_k)} \\ l_k^{(t_k)} \end{array} \right] \right\}, \quad (\text{C.1})$$

i.e. $h_{S^{(\cdot)}, L^{(\cdot)}}$ is a special function that maps a pair of vectors to a finite set. Note that (C.1) preserves the relative order between the elements of both vectors (but not their absolute order). ■

Lemma C.1.2 Consider a generic RFS \mathbf{Z} . Observe that

$$p(z||\mathbf{z}) = \frac{1}{|\mathbf{z}|! P(|\mathbf{Z}| = |\mathbf{z}|)} f(\mathbf{z}) \quad (\text{C.2})$$

the vector z is obtained by (arbitrarily) ordering the set \mathbf{z} , and $p(z||\mathbf{z})$ is a short notation for $p(z || \mathbf{Z} = |\mathbf{z}|)$. Note that this lemma relates a RFS density, evaluated at \mathbf{z} , with a family of $|\mathbf{z}|!$ (non-RFS) conditional pdfs, each evaluated at one permutation z of the elements of \mathbf{z} . Each permutation z represents the same physical event as the set \mathbf{z} (see Section 2.3.3).

Proof See (4.16). ■

Lemma C.1.3 For $f(\mathbf{x}_k|Z^k)$ given by (2.71) and given assumption $f(\mathbf{z}_k|\mathbf{x}_k) = f(\mathbf{z}_k|\mathbf{s}_k)$ (see (4.4)), we have

$$f(\mathbf{z}_k | \mathbf{s}_k, Z^{k-1}) = f(\mathbf{z}_k | \mathbf{s}_k) \quad (\text{C.3})$$

C. THE “ONE-SIDED” DECOUPLING BETWEEN TRACKING AND LABELLING IN BAYESIAN MULTI-TARGET TRACKING

Proof First, using Lemma C.1.2, observe that

$$p(z_k | s_k, Z^{k-1}, |\mathbf{z}_k|) = \frac{1}{K_z} f(\mathbf{z}_k | \mathbf{s}_k, Z^{k-1}). \quad (\text{C.4})$$

where the vectors z_k and s_k are obtained by (arbitrarily) ordering the sets \mathbf{z}_k and \mathbf{s}_k respectively and

$$K_z = |\mathbf{z}_k|! P(|\mathbf{Z}_k| = |\mathbf{z}_k| | s_k, Z^{k-1}).$$

Note that we can replace the set \mathbf{s}_k by the vector s_k in the right side of the conditional probability, since as we mentioned, they represent the same physical event. Observe now that

$$\begin{aligned} f(\mathbf{z}_k | \mathbf{s}_k, Z^{k-1}) &= K_z p(z_k | s_k, Z^{k-1}, |\mathbf{z}_k|) \\ &= K_z \sum_{l_k \in \Omega_{k-1}(s_k)} p(z_k | s_k, l_k, Z^{k-1}, |\mathbf{z}_k|) p(l_k | s_k, Z^{k-1}, |\mathbf{z}_k|) \end{aligned} \quad (\text{C.5})$$

where

$$\Omega_{k-1}(s_k) = \left\{ l_k \mid p(l_k | s_k, Z^{k-1}, |\mathbf{z}_k|) > 0 \right\}. \quad (\text{C.6})$$

Now, let us consider the pdf

$$p(z_k | s_k, l_k, Z^{k-1}, |\mathbf{z}_k|) = \frac{f(\mathbf{z}_k | \mathbf{x}_k, Z^{k-1})}{|\mathbf{z}_k|! P(|\mathbf{Z}_k| = |\mathbf{z}_k| | s_k, l_k, Z^{k-1})} \quad (\text{C.7})$$

(for $\mathbf{x}_k = h_{S(\cdot), L(\cdot)}(s_k, l_k)$), and let us assume that

$$P(|\mathbf{Z}_k| = |\mathbf{z}_k| | s_k, l_k, Z^{k-1}) = P(|\mathbf{Z}_k| = |\mathbf{z}_k| | s_k, Z^{k-1}),$$

which is a reasonable assumption, given that labels have no physical interpretation

C. THE “ONE-SIDED” DECOUPLING BETWEEN TRACKING AND LABELLING IN BAYESIAN MULTI-TARGET TRACKING

when considered in a single time step (see Section 4.2.1). The assumption implies that

$$p(z_k | s_k, l_k, Z^{k-1}, |\mathbf{z}_k|) = \frac{1}{K_z} f(\mathbf{z}_k | \mathbf{x}_k, Z^{k-1})$$

for $\mathbf{x}_k = h_{S^{(\cdot)}, L^{(\cdot)}}(s_k, l_k)$, and finally, by substituting in (C.5):

$$\begin{aligned} f(\mathbf{z}_k | \mathbf{s}_k, Z^{k-1}) &= \sum_{l_k \in \Omega_{k-1}(s_k)} f(\mathbf{z}_k | \mathbf{x}_k, Z^{k-1}) p(l_k | s_k, Z^{k-1}, |\mathbf{z}_k|) \\ &= \sum_{l_k \in \Omega_{k-1}(s_k)} f(\mathbf{z}_k | \mathbf{x}_k) p(l_k | s_k, Z^{k-1}, |\mathbf{z}_k|) \\ &= \sum_{l_k \in \Omega_{k-1}(s_k)} f(\mathbf{z}_k | \mathbf{s}_k) p(l_k | s_k, Z^{k-1}, |\mathbf{z}_k|) \\ &= f(\mathbf{z}_k | \mathbf{s}_k). \end{aligned} \tag{C.8}$$

■

Lemma C.1.4 Consider $f(\mathbf{x}_k | Z^k)$ given by (2.71), and the additional assumption

$$f(\mathbf{s}_k | \mathbf{x}_{k-1}) = f(\mathbf{s}_k | \mathbf{s}_{k-1}) \tag{C.9}$$

(which is also reasonable given the discussion in Section 4.2.1). We then have

$$f(\mathbf{s}_k | Z^{k-1}) = \int f(\mathbf{s}_k | \mathbf{s}_{k-1}) f(\mathbf{s}_{k-1} | Z^{k-1}) \delta_{\mathbf{s}_{k-1}}. \tag{C.10}$$

Proof Given $f(\mathbf{s}_k | Z^{k-1})$, $f(\mathbf{x}_k | Z^{k-1})$ and Lemma C.1.2, let us consider the probability densities

$$p(s_k | Z^{k-1}, t_k) = \frac{1}{K_s} f(\mathbf{s}_k | Z^{k-1}), \quad p(x_k | Z^{k-1}, t_k) = \frac{1}{K_s} f(\mathbf{x}_k | Z^{k-1})$$

where s_k and x_k are obtained by (arbitrarily) ordering \mathbf{s}_k and \mathbf{x}_k respectively,

C. THE “ONE-SIDED” DECOUPLING BETWEEN TRACKING AND LABELLING IN BAYESIAN MULTI-TARGET TRACKING

$K_s = t_k!P(T_k = t_k | Z^{k-1})$, and t_k is the cardinality of \mathbf{s}_k and \mathbf{x}_k (which is obviously the same if \mathbf{s}_k is assumed to be the unlabelled counterpart of \mathbf{x}_k). This leads to

$$\begin{aligned}
f(\mathbf{s}_k | Z^{k-1}) &= K_s p(s_k | Z^{k-1}, t_k) \\
&= K_s \sum_{l_k \in \Omega_{k-1}(s_k)} p(s_k, l_k | Z^{k-1}, t_k) \\
&= \sum_{l_k \in \Omega_{k-1}(s_k)} f(\mathbf{x}_k | Z^{k-1}) \\
&= \sum_{l_k \in \Omega_{k-1}(s_k)} \int f(\mathbf{x}_k | \mathbf{x}_{k-1}) f(\mathbf{x}_{k-1} | Z^{k-1}) \delta \mathbf{x}_{k-1} \\
&= \int \sum_{l_k \in \Omega_{k-1}(s_k)} f(\mathbf{x}_k | \mathbf{x}_{k-1}) f(\mathbf{x}_{k-1} | Z^{k-1}) \delta \mathbf{x}_{k-1} \tag{C.11}
\end{aligned}$$

where $\Omega_{k-1}(s_k)$ is defined by (C.6), $\mathbf{x}_k = h_{S(\cdot), L(\cdot)}(s_k, l_k)$, and the last line is obtained non-rigorously as we have not checked the conditions for switching the summation with the integral.

Now, let us consider the densities

$$\begin{aligned}
p(s_k, l_k | x_{k-1}, t_k) &= \frac{1}{K_{s|x}} f(\mathbf{x}_k | \mathbf{x}_{k-1}), \\
p(s_k | x_{k-1}, t_k) &= \frac{1}{K_{s|x}} f(\mathbf{s}_k | \mathbf{x}_{k-1})
\end{aligned}$$

where $K_{s|x} = t_k!P(T_k = t_k | \mathbf{X}_{k-1} = \mathbf{x}_{k-1})$. From (C.11), we then have

$$\begin{aligned}
f(\mathbf{s}_k | Z^{k-1}) &= \int K_{s|x} \left(\sum_{l_k \in \Omega_{k-1}(s_k)} p(s_k, l_k | x_{k-1}, t_k) \right) f(\mathbf{x}_{k-1} | Z^{k-1}) \delta \mathbf{x}_{k-1} \\
&= \int K_{s|x} p(s_k | x_{k-1}, t_k) f(\mathbf{x}_{k-1} | Z^{k-1}) \delta \mathbf{x}_{k-1} \\
&= \int f(\mathbf{s}_k | \mathbf{x}_{k-1}) f(\mathbf{x}_{k-1} | Z^{k-1}) \delta \mathbf{x}_{k-1} \tag{C.12}
\end{aligned}$$

C. THE “ONE-SIDED” DECOUPLING BETWEEN TRACKING AND LABELLING IN BAYESIAN MULTI-TARGET TRACKING

and let us then assume, without loss of generality, that each location $S_k^{(i)}$ is a continuous random variable, i.e. it assumes values in \mathbb{R}^n . By expanding the set integral in (C.12), we obtain

$$f(\mathbf{s}_k | Z^{k-1}) = \sum_{t_{k-1}=0}^{\infty} \frac{1}{t_{k-1}!} \int \sum_{l_{k-1} \in \Omega_{k-1}(s_{k-1})} f(\mathbf{s}_k | \mathbf{x}_{k-1}) f(\mathbf{x}_{k-1} | Z^{k-1}) ds_{k-1} \quad (\text{C.13})$$

where $\mathbf{x}_{k-1} = h_{S^{(\cdot)}, L^{(\cdot)}}(s_{k-1}, l_{k-1})$. Applying now assumption (C.9), we have

$$f(\mathbf{s}_k | Z^{k-1}) = \sum_{t_{k-1}=0}^{\infty} \frac{1}{t_{k-1}!} \int \sum_{l_{k-1} \in \Omega_{k-1}(s_{k-1})} f(\mathbf{s}_k | \mathbf{s}_{k-1}) f(\mathbf{x}_{k-1} | Z^{k-1}) ds_{k-1} \quad (\text{C.14})$$

where \mathbf{s}_{k-1} denotes a finite set whose elements are the entries of vector s_{k-1} . Let us now consider the probability densities

$$\begin{aligned} p(s_{k-1}, l_{k-1} | Z^{k-1}, t_{k-1}) &= \frac{1}{K_{s-1}} f(\mathbf{x}_{k-1} | Z^{k-1}), \\ p(s_{k-1} | Z^{k-1}, t_{k-1}) &= \frac{1}{K_{s-1}} f(\mathbf{s}_{k-1} | Z^{k-1}) \end{aligned}$$

where

$$K_{s-1} = t_{k-1}! P(T_{k-1} = t_{k-1} | Z^{k-1}).$$

C. THE “ONE-SIDED” DECOUPLING BETWEEN TRACKING AND LABELLING IN BAYESIAN MULTI-TARGET TRACKING

From (C.14), we have then

$$\begin{aligned}
& f(\mathbf{s}_k | Z^{k-1}) \\
&= \sum_{t_{k-1}=0}^{\infty} \frac{K_{s-1}}{t_{k-1}!} \int \sum_{l_{k-1} \in \Omega_{k-1}(s_{k-1})} f(\mathbf{s}_k | \mathbf{s}_{k-1}) p(s_{k-1}, l_{k-1} | Z^{k-1}, t_{k-1}) ds_{k-1} \\
&= \sum_{t_{k-1}=0}^{\infty} \frac{K_{s-1}}{t_{k-1}!} \int \sum_{l_{k-1} \in \Omega_{k-1}(s_{k-1})} f(\mathbf{s}_k | \mathbf{s}_{k-1}) p(s_{k-1} | Z^{k-1}, t_{k-1}) \\
&\quad \times p(l_{k-1} | s_{k-1}, Z^{k-1}, t_{k-1}) ds_{k-1} \\
&= \sum_{t_{k-1}=0}^{\infty} \frac{K_{s-1}}{t_{k-1}!} \int f(\mathbf{s}_k | \mathbf{s}_{k-1}) p(s_{k-1} | Z^{k-1}, t_{k-1}) ds_{k-1} \\
&= \sum_{t_{k-1}=0}^{\infty} \frac{1}{t_{k-1}!} \int f(\mathbf{s}_k | \mathbf{s}_{k-1}) f(s_{k-1} | Z^{k-1}) ds_{k-1} \\
&= \int f(\mathbf{s}_k | \mathbf{s}_{k-1}) f(s_{k-1} | Z^{k-1}) \delta \mathbf{s}_{k-1}. \tag{C.15}
\end{aligned}$$

■

Corollary C.1.5 For $f(\mathbf{x}_k | Z^k)$ given by (2.71), plus assumptions $f(\mathbf{z}_k | \mathbf{x}_k) = f(\mathbf{z}_k | \mathbf{s}_k)$ and (C.9), $(\mathbf{S}_k, \mathbf{Z}_k)$ consists of a first-order partially observed Markov process, i.e.

$$f(\mathbf{s}_k | Z^k) = \frac{f(\mathbf{z}_k | \mathbf{s}_k) f(\mathbf{s}_k | Z^{k-1})}{f(\mathbf{z}_k | Z^{k-1})} \tag{C.16}$$

where

$$f(\mathbf{s}_k | Z^{k-1}) = \int f(\mathbf{s}_k | \mathbf{s}_{k-1}) f(\mathbf{s}_{k-1} | Z^{k-1}) \delta \mathbf{s}_{k-1}. \tag{C.17}$$

■

Clearly, the locations-only Bayesian recursion (C.16) does not involve any probability distribution of labels or labelled multi-target states, and therefore, we can say that Bayesian tracking does not depend on Bayesian labelling. The recursion given by (C.16) may be implemented using any multi-target tracking

C. THE “ONE-SIDED” DECOUPLING BETWEEN TRACKING AND LABELLING IN BAYESIAN MULTI-TARGET TRACKING

algorithm that estimates the entire multi-target posterior (such as the M-SMC filter, the PHD filter, or the MHT).

C.2 The labelling sub-problem

If we use some algorithm to approximate only $f(\mathbf{s}_k | Z^k)$, how do we obtain, from the algorithm's results, the labelled multi-target density $f(\mathbf{x}_k | Z^k)$? Observe now that, from Definition 4.3.1

$$f(\mathbf{x}_k | Z^k) = p_l(\mathbf{x}_k | \mathbf{s}_k, Z^k) f(\mathbf{s}_k | Z^k) \quad (\text{C.18})$$

and therefore, the only extra necessary step is to calculate the labelling probabilities $p_l(\mathbf{x}_k | \mathbf{s}_k, Z^k)$. Given property (4.31) and since

$$f(\mathbf{x}_{k-1} | Z^{k-1}) = p_l(\mathbf{x}_{k-1} | \mathbf{s}_{k-1}, Z^{k-1}) f(\mathbf{s}_{k-1} | Z^{k-1}),$$

labelling probabilities can be calculated recursively given $f(\mathbf{s}_k | Z^k)$ across different time steps and appropriate initial conditions. In other words, Bayesian labelling depends on the results of Bayesian tracking.

References

- S. I. Aihara, A. Bagchi, and S. Saha. On parameter estimation of stochastic volatility models from stock data using particle filter – Application to AES index –. *International Journal of Innovative Computing, Information and Control*, 5(1):17–27, 2009. 2, 80, 83
- C. Andrieu and A. Doucet. Particle filtering for partially observed Gaussian state space models. *J. Royal Stat. Soc. B*, 64:827–836, 2002. 14, 19, 22
- E. H. Aoki. A general approach for altitude estimation and mitigation of slant range errors on target tracking using 2D radars. In *Proc. 13th International Conference on Information Fusion*, Edinburgh, UK, June 26–29, 2010. 1
- M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Process.*, 50(2):174–188, February 2002. 12
- J. M. Aughenbaugh and B. R. La Cour. Metric selection for information theoretic sensor management. In *Proc. 11th International Conference on Information Fusion*, Cologne, Germany, June 30, July 1–3, 2008. 138, 140, 153, 155, 159
- J. M. Aughenbaugh and B. R. La Cour. Sensor management for particle filter tracking. *IEEE Trans. Aerosp. Electron. Syst.*, 47(1):503–523, 2011. 139, 148
- A. Bagchi. *Optimal Control of Stochastic Systems*. Prentice-Hall, Upper Saddle River, NJ, 1993. 43

REFERENCES

- Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with applications to tracking and navigation*. John Wiley & Sons, New York, NY, 2001. 2, 75, 76, 117, 161
- H. A. P. Blom and E. A. Bloem. Particle filtering for stochastic hybrid systems. In *Proc. 43rd IEEE Conf. Decision and Control*, Atlantis, Bahamas, December 14–17, 2004. 14
- H. A. P. Blom and E. A. Bloem. Permutation invariance in bayesian estimation of two targets that maneuver in and out formation flight. In *Proc. 12th International Conference of Information Fusion*, pages 1296 –1303, july 2009. 93
- H. A. P. Blom and E. A. Bloem. Decomposed particle filtering and track swap estimation in tracking two closely spaced targets. In *Proc. 14th International Conference of Information Fusion*, Chicago, IL, July 5–8, 2011. 38, 93, 97, 131
- H. A. P. Blom, E. A. Bloem, Y. Boers, and J. N. Driessen. Tracking closely spaced targets: Bayes outperformed by an approximation? In *Proc. 11th International Conference on Information Fusion*, Cologne, Germany, June 30, July 1–3, 2008. 2, 34
- Y. Boers, J. N. Driessen, and L. Schipper. Particle filter based sensor selection in binary sensor networks. In *Proc. 11th International Conference on Information Fusion*, pages 389–395, Cologne, Germany, June 30, July 1–3, 2008. 42, 44
- Y. Boers, E. Sviestins, and J. N. Driessen. Mixed labelling in multitarget particle filtering. *IEEE Trans. Aerosp. Electron. Syst.*, 46(2):792–802, 2010. 14, 91, 105, 116, 118
- M. Bolić, P. M. Djurić, and S. Hong. Resampling algorithms and architectures for distributed particle filters. *IEEE Trans. Signal Process.*, 53(7):2442–2450, 2005. 73
- M. Briers, A. Doucet, and S. Maskell. Smoothing algorithms for state-space model. *Annals Institute Statistical Mathematics*, 62(1):61–89, 2010. 14

REFERENCES

- P. D. Burns and W. D. Blair. Optimal phased array radar beam pointing for mtt. In *Proc. 2004 IEEE Aerospace Conference*, volume 3, March 2004. 42
- C. M. Carvalho, M. S. Johannes, H. F. Lopes, and N. G. Polson. Particle learning and smoothing. *Statistical Science*, 25(1):88–106, 2010. 51
- E. K. P. Chong, C. Kreucher, and A. O. Hero. Partially observable Markov decision process approximations for adaptive sensing. *Discrete Event Dynamic Systems*, 19(3):377–422, September 2009. 151
- T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley–Interscience, 1991. 139, 143, 148
- D. Crisan and A. Doucet. A survey of convergence results on particle filtering methods for practitioners. *IEEE Trans. Signal Process.*, 50(3):736–746, March 2002. 12, 16
- D. Crouse, P. Willett, L. Svensson, D. Svensson, and M. Guerriero. The set MHT. In *Proc. 14th International Conference of Information Fusion*, Chicago, IL, July 5–8, 2011a. ix, 38, 93, 97, 105, 113, 131
- D. F. Crouse, P. Willett, and Y. Bar-Shalom. Developing a real-time track display that operators do not hate. *IEEE Trans. Aerosp. Electron. Syst.*, 59(7):3441–3447, July 2011b. 113
- D. J. Daley and D. Vere-Jones. *An Introduction to the Theory of Point Processes*, volume I: Elementary Theory and Methods. Springer, second edition, 2003. 31, 36
- J. F. G. de Freitas. *Bayesian Methods for Neural Networks*. PhD thesis, University of Cambridge, Cambridge, MA, 1999. 6, 12
- N. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Stat. Soc. B*, 39:1–38, 1977. 28

-
- A. Doucet and A. M. Johansen. Tutorial on particle filtering and smoothing: Fifteen years later. In Dan Crisan and Boris Rozovskii, editors, *The Oxford Handbook of Nonlinear Filtering*. Oxford University Press, 2011. 14
- A. Doucet, N. J. Gordon, and V. Krishnamurthy. Particle filters for state estimation of jump Markov linear systems. *IEEE Trans. Signal Process.*, 49(3): 613–624, 2001. 10, 11, 16, 70
- A. Doucet, B.-N. Vo, C. Andrieu, and M. D. Thomas. Particle filtering for multi-target tracking and sensor management. In *Proc. 5th International Conference on Information Fusion*, pages 474–481, Annapolis, MD, July 7–11, 2002. 46, 48, 138, 141
- Á. García-Fernández and J. Grajal. Multitarget tracking using the Joint Multi-track Probability Density. In *Proc. 12th International Conference on Information Fusion*, Seattle, WA, 2009. 93
- Á. García-Fernández, M. Morelande, and J. Grajal. Particle filter for extracting target label information when targets move in close proximity. In *Proc. 14th International Conference of Information Fusion*, Chicago, IL, July 5–8, 2011. 38, 93, 97
- Á. F. García-Fernández, J. Grajal, and M. R. Morelande. Two-layer particle filter for multiple target detection and tracking. *IEEE Trans. Aerosp. Electron. Syst.*, accepted for publication, 2012. 93, 97
- R. Georgescu, P. Willett, L. Svensson, and M. Morelande. Two linear complexity particle filters capable of maintaining target label probabilities for targets in close proximity. In *Proc. 15th International Conference on Information Fusion*, pages 2370 –2377, july 2012. 93
- W. R. Gilks and C. Berzuini. Following a moving target - Monte Carlo inference for dynamic Bayesian models. *J. Royal Stat. Soc. B*, 63:127–146, 2001. 70
- I. R. Goodman, R. Mahler, and H. T. Nguyen. *Mathematics of data fusion*. Kluwer Academic Publishers, Norwell, MA, 1997. 31

REFERENCES

- N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to non-linear/non-Gaussian Bayesian state estimation. *IEE Proceedings-F*, 140:107–113, 1993. 6, 30, 55
- M. Guerriero, L. Svensson, D. Svensson, and P. Willett. Shooting two birds with two bullets: how to find Minimum Mean OSPA estimates. In *Proc. 13th International Conference on Information Fusion*, Edinburgh, UK, 2010. 36
- F. Gustafsson and S. Saha. Particle filtering with dependent noise. In *Proc. 13th International Conference on Information Fusion*, Edinburgh, UK, June 26–29, 2010. 16, 18
- T. Hanselmann, M. Morelande, B. Moran, and P. Sarunic. Sensor scheduling for multiple target tracking and detection using passive measurements. In *Proc. 11th International Conference on Information Fusion*, pages 1528–1535, Cologne, Germany, June 30, July 1–3, 2008. 44, 46, 139
- A. O. Hero and C. Kreucher. Network sensor management for tracking and localization. In *Proc. 10th International Conference on Information Fusion*, Québec, Canada, July 9–12, 2007. 151
- A. O. Hero, B. Ma, O. Michel, and J. D. Gorman. Alpha-divergence for classification, indexing and retrieval. Technical Report 328, Univ. Michigan, Ann Arbor, MI, May 2001. 138
- A. O. Hero, B. Ma, O. Michel, and J. D. Gorman. Applications of entropic spanning graphs. *IEEE Signal Process. Mag.*, 19(5):85–95, 2002. 138
- A. O. Hero, C. Kreucher, and D. Blatt. Information theoretic approaches to sensor management. In A. O. Hero, D. Castañón, D. Cochran, and K. Kastella, editors, *Foundations and Applications of Sensor Management*, chapter 3, pages 33–57. Springer, 2007. 139, 149, 150, 151, 152
- S. L. Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Rev. Financial Stud.*, 6(2):327–343, 1993. 80

-
- T. Higuchi. Self-organizing time series model. In A. Doucet, J. F. G. De Freitas, and N. J. Gordon, editors, *Sequential Monte Carlo Methods in Practice*. Springer, 2001. 30, 55
- K. J. Hintz and E. S. McVey. Multi-process constrained estimation. *IEEE Trans. Syst., Man, Cybern.*, 21(1):237–244, 1991. 138
- J. D. Hol, T. B. Schön, and F. Gustafsson. On resampling algorithms for particle filters. In *Nonlinear Statistical Signal Processing Workshop*, Cambridge, UK, September 2006. 12, 65, 70
- P. Horridge and S. Maskell. A scalable method of tracking targets with dependent distributions. In *Proc. 12th International Conference on Information Fusion*, Seattle, WA, 2009. 96
- X.-L. Hu, T. B. Schön, and L. Ljung. A basic convergence result for particle filtering. *IEEE Trans. Signal Process.*, 56(4):1337–1348, April 2008. 12
- M. Huber, T. Bailey, H. Durrant-Whyte, and U. Hanebeck. On entropy approximation for gaussian mixture random vectors. In *Proc. IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 181–188, August 20–22, 2008. 144
- N. Kantas, A. Doucet, S. S. Singh, and J. M. Maciejowski. An overview of Sequential Monte Carlo methods for parameter estimation on general state space models. In *Proc. 15th IFAC Symp. System Identification (SYSID)*, Saint-Malo, France, 2009. 14, 51
- K. Kastella. Discrimination gain to optimize classification. *IEEE Trans. Syst., Man, Cybern. A*, 27(1):112–116, 1997a. 138
- K. Kastella. Joint multitarget probabilities for detection and tracking. In *Proc. SPIE Acquisition, Tracking, and Pointing XI*, volume 3086, pages 122–128, 1997b. 136
- F. Katsilieris, Y. Boers, and H. Driessen. Sensor management for PRF selection in the track-before-detect context. In *Proc. 2012 IEEE Radar Conference*, pages 360–365, Atlanta, GA, May 2012. 42

REFERENCES

- G. Kitagawa. A Monte Carlo filtering and smoothing method for non-Gaussian nonlinear state space models. In *Proc. 2nd US-Japan Joint Seminar on Statistical Time Series Analysis*, pages 110–131, Honolulu, HI, 1993. 6
- G. Kitagawa. Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *J. Comput. Graph. Statist.*, 5(1):1–25, 1996. 12
- M. Klaas, N. de Freitas, and A. Doucet. Toward practical N^2 Monte Carlo: the marginal particle filter. In *Proc. 21th Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-05)*, pages 308–315, Arlington, Virginia, 2005. AUAI Press. 23, 24, 25, 26, 63, 72
- A. Klenke. *Probability Theory – A Comprehensive Course*. Springer, London, UK, 2008. 33
- C. Kreucher and Alfred O. Hero. Non-myopic approaches to scheduling agile sensors for multitarget detection, tracking, and identification. In *Proc. IEEE Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 885–888, Philadelphia, PA, March 18–23, 2005. 43, 49
- C. Kreucher, K. Kastella, and A. O. Hero. Multi-target sensor management using alpha-divergence measures. In *Lecture Notes in Computer Science, Proc. of the 2nd International Conference on Information Processing in Sensor Networks*, pages 209–222, 2003a. 138
- C. Kreucher, K. Kastella, and A. O. Hero. Information based sensor management for multitarget tracking. In *Proc. SPIE*, volume 5204, pages 480–489, August 3–8, 2003b. 138
- C. Kreucher, D. Blatt, A. Hero, and K. Kastella. Adaptive multi-modality sensor scheduling for detection and tracking of smart targets. In *Proc. Defense Applications of Signal Processing Workshop*, October 31, November 1–5, 2004. 47
- C. Kreucher, A. O. Hero, and K. Kastella. A comparison of task driven and information driven sensor management for target tracking. In *Proc. IEEE 44th*

-
- Conference on Decision and Control*, pages 4004–4009, December 12–15, 2005. 16, 44, 45, 46, 126, 139, 150, 151, 152
- C. Kreucher, A. O. Hero, K. Kastella, and M. R. Morelande. An information-based approach to sensor management in large dynamic networks. In *Proc. IEEE*, pages 978–999, May 2007. 151
- A. Kuwertz, M. F. Huber, and F. Sawo. Multi-step sensor management for localizing movable sources of spatially distributed phenomena. In *Proc. 13th International Conference on Information Fusion*, Edinburgh, UK, July 26–29, 2010. 43
- X. Lin, Y. Bar-Shalom, and T. Kirubarajan. Multisensor-multitarget bias estimation for asynchronous sensors. In *Proc. SPIE Signal processing, sensor fusion, and target recognition XIII*, volume 5429, pages 105–116, Orlando, FL, April 12–14, 2004. 29
- F. Lindsten, T. B. Schön, and L. Svensson. A non-degenerate Rao-Blackwellised particle filter for estimating static parameters in dynamical models. In *Proc. 16th IFAC Symposium on System Identification (SYSID)*, 2012. 51, 52, 58, 59
- J. Liu and M. West. Combined parameters and state estimation in simulation-based filtering. In A. Doucet, N. de Freitas, and N. Gordon, editors, *Sequential Monte Carlo Methods in Practice*, pages 197–223. New York: Springer-Verlag, 2001. 30, 51, 55, 56
- J. S. Liu and R. Chen. Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93:1032–1044, 1998. 12
- X. Liu and M. Niranjan. State and parameter estimation of the heat shock response system using Kalman and particle filters. *Bioinformatics*, 28(3):366–372, 2012. 51
- W.-K. Ma, B.-N. Vo, S. Singh, and A. Baddeley. Tracking an unknown time-varying number of speakers using TDOA measurements: A random finite set approach. *IEEE Trans. Signal Process.*, 54(9):3291–3304, 2006. 38, 93

REFERENCES

- R. Mahler. Global optimal sensor allocation. In *Proc. 9th National Symposium on Sensor Fusion*, pages 167–172, Monterey, CA, 1996. 138
- R. Mahler. Why multi-source, multi-target data fusion is tricky. In *Proc. 1999 IRIS Nat'l Symp. on Sensor and Data Fusion*, pages 135–154, Laurel, MD, May 25–27, 1999. 34, 35
- R. Mahler. *Statistical Multisource-Multitarget Information Fusion*. Artech House, Norwood, MA, 2007. 31, 32, 33, 35, 39, 40, 100, 101, 102, 107, 114, 116, 122, 124, 129
- R. Mahler. Implementation and application of PHD and CPHD filters. In *The 11th International Conference on Information Fusion, Tutorial Session 10*, Cologne, Germany, June 30, July 1–3, 2008. 31
- J. Manyika and H. Durrant-Whyte. *Data Fusion and Sensor Management: a decentralized information-theoretic approach*. Ellis Horwood, New York, NY, 1994. 138
- M. Morelande, C. Kreucher, and K. Kastella. A Bayesian approach to multiple target detection and tracking. *IEEE Trans. Aerosp. Electron. Syst.*, 55(5): 1589–1604, 2007. 93
- S. Mori, C.-Y. Chong, and K. C. Chang. Track association and fusion using Janossy measure density functions. In *Proc. 12th International Conference on Information Fusion*, Seattle, WA, 2009. 38
- C. Nemeth, P. Fearnhead, L. Mihaylova, and D. Vorley. Particle learning methods for state and parameter estimation. In *Proc. DF&TT'12*, London, UK, May 16–17, 2012. 51
- J. Olsson, O. Cappé, R. Douc, and É. Moulines. Sequential Monte Carlo smoothing with application to parameter estimation in nonlinear state space models. *Bernoulli*, 14(1):155–179, 2008. 28
- M. Pitt and N. Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94:590–599, 1999. 56

REFERENCES

- D. B. Reid. An algorithm for tracking multiple targets. *IEEE Trans. Automat. Contr.*, AC-24(6), December 1979. 113, 114
- A. Rényi. On measures of information and entropy. In *Proc. 4th Berkeley Symposium on Mathematics, Statistics and Probability*, volume 1, pages 547–561, 1961. 141
- B. Ristic, B.-N. Vo, and D. Clark. A note on the reward function for PHD filters with sensor control. *IEEE Trans. Aerosp. Electron. Syst.*, 47(2):1521–1529, 2011a. 128, 139
- B. Ristic, Ba-Ngu Vo, Daniel Clark, and Ba-Tuong Vo. A metric for performance evaluation of multi-target tracking algorithms. *IEEE Trans. Aerosp. Electron. Syst.*, 59(7):3452–3457, 2011b. 110
- X. Rong Li and V. P. Jilkov. Survey of maneuvering target tracking. Part I: Dynamic models. *IEEE Trans. Aerosp. Electron. Syst.*, 39(4):1333–1364, 2003. 74
- D. J. Salmond, D. Fisher, and N. J. Gordon. Tracking and identification for closely spaced objects in clutter. In *Proc. European Control Conf.*, 1997. 38, 93
- C. O. Savage and B. F. La Scala. Sensor management for tracking smart targets. *Digit. Signal Process.*, 19(6):968–977, December 2009. 47
- W. Schmaedeke and K. Kastella. Event-averaged maximum likelihood estimation and information-based sensor management. In *Proc. SPIE*, volume 2232, pages 91–96, 1994. 138
- T. Schön, F. Gustafsson, and P.-J. Nordlund. Marginalized particle filters for mixed linear nonlinear state-space models. *IEEE Trans. Signal Process.*, 53: 2279–2289, 2005. 20
- T. B. Schön, A. Wills, and B. Ninness. System identification of nonlinear state-space models. *Automatica*, 47(1):39–49, 2011. 28

REFERENCES

- D. Schuhmacher, B.-T. Vo, and B.-N. Vo. A consistent metric for performance evaluation of multi-object filters. *IEEE Trans. Signal Process.*, 56(8):3447–3457, 2008. 35
- P. Skoglar, U. Orguner, and F. Gustafsson. On information measures based on particle mixture for optimal bearings-only tracking. In *Proc. 2009 IEEE Aerospace Conference*, pages 1–14, March 2009. 42
- M. Syldatk, E. Sviestins, and F. Gustafsson. Expectation maximization algorithm for calibration of ground sensor networks using a road constrained particle filter. In *Proc. 15th International Conference of Information Fusion*, Singapore, July 9–12, 2012. 1, 29
- J. Vermaak, A. Doucet, and P. Pérez. Maintaining multi-modality through mixture tracking. In *Proc. 9th IEEE International Conference on Computer Vision*, volume 2, pages 1110–1116, Nice, France, 2003. 14
- B.-N. Vo, S. Singh, and A. Doucet. Sequential Monte Carlo methods for multi-target filtering with random finite sets. *IEEE Trans. Aerosp. Electron. Syst.*, 41(4):1224–1245, 2005. 34, 39, 40
- B.-T. Vo and B.-N. Vo. A random finite set conjugate prior and application to multi-target tracking. In *Proc. 7th International Conference on Intelligent Sensors, Sensor Networks and Information*, pages 431–436, 2011. 93, 95, 99
- A.-T. Vu, B.-N. Vo, and R. Evans. Particle Markov Chain Monte Carlo for bayesian multi-target tracking. In *Proc. 14th International Conference of Information Fusion*, Chicago, IL, July 5–8, 2011. 38
- J. L. Williams. *Information Theoretic Sensor Management*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, February 2007. 139, 148
- J. L. Williams, J. W. Fisher, and A. S. Willsky. Approximate dynamic programming for communication-constrained sensor network management. *IEEE Trans. Signal Process.*, 55(8):4300–4311, 2007. 44, 46
- F. Zhao, J. Shin, and J. Reich. Information-driven dynamic sensor collaboration. *IEEE Signal Process. Mag.*, 19(2):61–72, 2002. 148

REFERENCES

- H. Zhu, C. Han, and Y. Lin. Particle labeling PHD filter for multi-target track-valued estimates. In *Proc. 14th International Conference of Information Fusion*, Chicago, IL, July 5–8, 2011. 38

Abstract

In estimation problems, accuracy of the estimates of the quantities of interest cannot be taken for granted. This means that estimation errors are expected, and a good estimation algorithm should be able not only to compute estimates that are optimal in some sense, but also provide meaningful measures of uncertainty associated with those estimates. In some situations, we might also be able to reduce estimation uncertainty through the use of feedback on observations, an approach referred to as sensor management.

Characterization of estimation uncertainty, as well as sensor management, are certainly difficult tasks for general partially observed processes, which might be non-linear, non-Gaussian, and/or have dependent process and observation noises. Sequential Monte Carlo (SMC) methods, also known as particle filters, are numerical Bayesian estimators which are, in principle, able to handle highly general estimation problems. However, SMC methods are known to suffer from a phenomenon called degeneracy, or self-resolving, which greatly impairs their usefulness against certain classes of problems.

One of such classes, that we address in the first part of this thesis, is the joint state and parameter estimation problem, where there are unknown parameters to be estimated together with the time-varying state. Some SMC variants have been proposed to counter the degeneracy phenomenon for this problem, but these state-of-the-art techniques are either non-Bayesian or introduce biases on the system model, which might not be appropriate if proper characterization of estimation uncertainty is required. For this type of scenario, we propose using the Rao-Blackwellized Marginal Particle Filter (RBMPF),

a combination of two SMC algorithm variants: the Rao-Blackwellized Particle Filter (RBPF) and the Marginal Particle Filter (MPF). We derive two new versions of the RBMPF: one for models with low dimensional parameter vectors, and another for more general models. We apply the proposed methods to two practical problems: the target tracking problem of turn rate estimation for a constant turn maneuver, and the econometrics problem of stochastic volatility estimation. Our proposed methods are shown to be effective solutions, both in terms of estimation accuracy and statistical consistency, i.e. characterization of estimation uncertainty.

Another problem where standard particle filters suffer from degeneracy, addressed in the second part of this thesis, is the joint multi-target tracking and labelling problem. In comparison with the joint state and parameter estimation problem, this problem poses an additional challenge, namely, the fact that it has not been properly mathematically formulated in previous literature. Using Finite Set Statistics (FISST), we provide a sound theoretical formulation for the problem, and in order to actually solve the problem, we propose a novel Bayesian algorithm, the Labelling Uncertainty-Aware Particle Filter (LUA-PF) filter, essentially a combination of the RBMPF and the Multi-target Sequential Monte Carlo (M-SMC) filter techniques. We show that the new algorithm achieves significant improvements on both finding the correct track labelling and providing a meaningful measure of labelling uncertainty.

In the last part of this thesis, we address the sensor management problem. Although we apply particle filters to the problem, they are not the main focus of this part of the work. Instead, we concentrate on a more fundamental question, namely, which sensor management criterion should be used in order to obtain the best results in terms of information gain and/or reduction of uncertainty. In order to answer this question, we perform an in-depth theoretical and empirical analysis on two popular sensor management criteria based on information

theory – the Kullback-Leibler and Rényi divergences. On the basis of this analysis, we are able to either confirm or reject some previous arguments used as theoretical justification for these two criteria.