

Branko Ristic

Particle Filters for Random Set Models

Particle Filters for Random Set Models

Branko Ristic

Particle Filters for Random Set Models



Springer

Branko Ristic
DSTO
Port Melbourne
Australia

ISBN 978-1-4614-6315-3 ISBN 978-1-4614-6316-0 (eBook)
DOI 10.1007/978-1-4614-6316-0
Springer New York Heidelberg Dordrecht London

Library of Congress Control Number: 2013933594

© Springer Science+Business Media New York 2013

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law. The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

This book is about state estimation of stochastic dynamic systems (objects or phenomena) from noisy measurements, also known as *stochastic filtering*. The commonly accepted framework for stochastic filtering is the Bayesian probabilistic framework, founded around the concept of a random variable. The exact conceptual solution in this framework, referred to as the sequential Bayesian estimator, computes recursively the posterior density of the state of the dynamic system. This time-varying posterior density provides the complete probabilistic characterization of the dynamic system of interest over time.

In the general non-linear/non-Gaussian context, the sequential Bayesian estimator cannot be solved analytically in closed form and is therefore implemented by approximations. Particle filters are a class of Monte Carlo statistical simulation-based methods which can provide very accurate approximations of the sequential Bayesian estimator. In the last decade, particle filters have become exceptionally popular, mainly due to the ease of their implementation and ever increasing speed of computers. Applications of particle filters span many disciplines of human endeavor: radar/sonar, communications, navigation, bio-informatics, finance, and ecology to name a few. Not surprisingly, several good tutorials and books have been published in the last decade on the topic of particle filters.

While the formulation of the standard sequential Bayesian estimator is general in the sense that it provides the exact solution in the non-linear/non-Gaussian context, it is nevertheless limited in the following sense:

- it assumes a single dynamic system (object, phenomenon) which is permanently active (or present)
- it assumes perfect detection (i.e., no false or miss detections)
- it assumes precise measurements and measurement models.

The recent theoretical advances in sequential Bayesian estimation carried out in the framework of random set theory provide new opportunities which are not widely known. These recent developments, (due to Mahler, Goodman, Vo, and others) have overcome all the limitations listed above, and thus have dramatically widened the scope of applications of particle filters: from single to multiple systems; from permanently active systems (i.e., present objects) to on/off systems

(i.e., appearing/disappearing objects); from perfect to imperfect detection; from precise to imprecise measurements and measurement models.

This book attempts to review some of the theoretical aspects of sequential Bayesian estimation in the random set theoretic framework, describe the particle filter implementation of the most popular algorithms and illustrate the enormous practical importance of this framework for inference in applications. The book is divided into seven chapters. After the introductory chapter, [Chap. 2](#) reviews the theoretical foundations necessary for understanding the remaining five chapters. This includes non-standard measurement models, multi-object filtering, sensor control and standard particle filtering. [Chapter 3](#) is dedicated to applications of non-standard measurements, such as imprecise observations, imprecise measurement models and implication rules. [Chapter 4](#) presents particle filter implementation of multi-object filters such as the PHD/CPHD filter and the Bernoulli filter. Sensor control for random set-based particle filters is covered in [Chap. 5](#). Multi-target tracking, including algorithms, the axiomatic performance metric and the application to pedestrian tracking, is presented in [Chap. 6](#). [Chapter 7](#) is dedicated to more advanced topics of extended object tracking and calibration of multi-sensor tracking systems.

The book is the outcome of 3 years of active collaborative research funded by the DSTO fellowship programme. The help and guidance from the experts in the field, primarily D. E. Clark (Heriot Watt University), B.-N. Vo, B.-T. Vo (Curtin University), and R. P. S. Mahler (Lockheed Martin), was of crucial importance on this project and hence greatly appreciated. Other collaborators whose contributions are included in the book are: A. Bishop (NICTA), A. Gning and S. Julier (University College London), Y. Petetin (Telecom Sup Paris), L. Mihaylova (Lancaster University), T. Wood (The University of Oxford), M. Unay (Heriot Watt University), A. Farina (Selex SI), Á. F. García-Fernández (Chalmers University of Technology), S. Arulampalam, N. Gordon, J. Sherrah and A. Skvortsov (DSTO). The author is grateful to all of them.

Melbourne, October 2012

Branko Ristic

Contents

1	Introduction	1
	References	3
2	Background	5
2.1	A Brief Review of Particle Filters	5
2.2	Online Sensor Control	6
2.3	Non-standard Measurements	10
2.3.1	Imprecise Measurements	10
2.3.2	Imprecise Measurement Function	11
2.3.3	Uncertain Implication Rules	12
2.3.4	Particle Filter Implementation	13
2.3.5	Applications	14
2.4	Multiple Objects and Imperfect Detection	15
2.4.1	Random Finite Sets	15
2.4.2	Multi-Object Stochastic Filtering	19
2.4.3	OSPA Metric	20
2.5	Specialized Multi-Object Filters	22
2.5.1	Bernoulli Filter	22
2.5.2	PHD and CPHD Filter	24
	References	26
3	Applications Involving Non-standard Measurements	29
3.1	Estimation Using Imprecise Measurement Models	29
3.1.1	Localization Using the Received Signal Strength	29
3.1.2	Prediction of an Epidemic Using Syndromic Data	34
3.1.3	Summary	39
3.2	Fusion of Spatially Referring Natural Language Statements	39
3.2.1	Language, Space, and Modeling	40
3.2.2	An Illustrative Example	42
3.3	Classification Using Imprecise Likelihoods	44
3.3.1	Modeling	45
3.3.2	Classification Results	47
	References	50

4 Multi-Object Particle Filters	53
4.1 Bernoulli Particle Filters.	53
4.1.1 Standard Bernoulli Particle Filters.	53
4.1.2 Bernoulli Box-Particle Filter.	56
4.2 PHD/CPDH Particle Filters with Adaptive Birth Intensity	62
4.2.1 Extension of the PHD Filter.	63
4.2.2 Extension of the CPHD Filter.	66
4.2.3 Implementation.	67
4.2.4 A Numerical Study	72
4.2.5 State Estimation from PHD/CPHD Particle Filters	76
4.3 Particle Filter Approximation of the Exact Multi-Object Filter.	79
References	83
5 Sensor Control for Random Set Based Particle Filters	85
5.1 Bernoulli Particle Filter with Sensor Control.	85
5.1.1 The Reward Function	85
5.1.2 Bearings Only Tracking in Clutter with Observer Control	87
5.1.3 Target Tracking via Multi-Static Doppler Shifts	96
5.2 Sensor Control for PHD/CPHD Particle Filters	104
5.2.1 The Reward Function	105
5.2.2 A Numerical Study	108
5.3 Sensor Control for the Multi-Target State Particle Filter	112
5.3.1 Particle Approximation of the Reward Function	112
5.3.2 A Numerical Study	113
References	118
6 Multi-Target Tracking	121
6.1 OSPA-T: A Performance Metric for Multi-Target Tracking	122
6.1.1 The Problem and Its Conceptual Solution	123
6.1.2 The Base Distance and Labeling of Estimated Tracks	125
6.1.3 Numerical Examples	128
6.2 Trackers Based on Random Set Filters.	132
6.2.1 Multi-Target Trackers Based on the Bernoulli PF.	132
6.2.2 Multi-Target Trackers Based on the PHD Particle Filter	135
6.2.3 Error Performance Comparison Using the OSPA-T Error	136
6.3 Application: Pedestrian Tracking	137
6.3.1 Video Dataset and Detections.	138
6.3.2 Description of Algorithms	139
6.3.3 Numerical Results	141
References	142

Contents	ix
7 Advanced Topics	145
7.1 Bernoulli Filter for Extended Target Tracking.	145
7.1.1 Mathematical Models	146
7.1.2 Equations of the Bernoulli Filter for an Extended Target	148
7.1.3 Numerical Implementation	150
7.1.4 Simulation Results	152
7.1.5 Application to a Surveillance Video	155
7.2 Calibration of Tracking Systems	157
7.2.1 Background and Problem Formulation.	160
7.2.2 The Proposed Calibration Algorithm	161
7.2.3 Importance Sampling with Progressive Correction	163
7.2.4 Application to Sensor Bias Estimation.	165
References	171
Index	173

Acronyms

2DA	Two-dimensional assignment
BPF	Bernoulli particle filter
BPF-X	Bernoulli particle filter for extended target
CDF	Cumulative distribution function
CPHD	Cardinalized probability hypothesis density
CP	Constraints propagation
CRLB	Cramér-Rao lower bound
FISST	Finite set statistics
GLF	Generalized likelihood function
GM	Gaussian mixture
HOG	Histogram of oriented gradients
KDE	Kernel density estimation
KL	Kullback-Leibler
LM	Linear multitarget
MCMC	Markov chain Monte Carlo
MoE	Measure of effectiveness
NL	Natural language
OID	Optimal importance density
OSPA	Optimal sub-pattern assignment
OSPA-T	Optimal sub-pattern assignment metric for tracks
PDF	Probability density function
PF	Particle filter
PHD	Probability hypothesis density
POMDP	Partially observed Markov decision process
RFS	Random finite set
RSS	Received signal strength
SIR	Susceptible-infectious-recovered
SIS	Sequential importance sampling
SMC	Sequential Monte Carlo
TBM	Transferrable belief model
TPM	Transitional probability matrix

Notations

\mathcal{X}	State space
\mathbf{x}	The state of a dynamic object (modeled by a random variable)
\mathbf{X}	The state of multiple-dynamic objects (modeled as a RFS variable)
\mathcal{Z}	Measurement space
\mathbf{z}	Precise (random) measurement (modeled by a random variable)
ζ	Imprecise (random) measurement (modeled by a random closed set variable)
\mathbf{Z}	Measurement set (modeled by a RFS variable)
k	Discrete-time index
$p(\mathbf{x})$	Probability density function (PDF) of a random variable \mathbf{x}
$p_0(\mathbf{x})$	Initial (prior) density
$f(\mathbf{X})$	FISST PDF of a RFS variable \mathbf{X}
$\pi_{k k-1}(\mathbf{x} \mathbf{x}')$	Single object transitional PDF (from $k - 1$ to k)
$\Pi_{k k-1}(\mathbf{X} \mathbf{X}')$	Multi-object transitional FISST PDF
$g_k(\mathbf{z} \mathbf{x})$	Single-object likelihood function at time k
$\tilde{g}_k(\mathbf{z} \mathbf{x})$	Single-object <i>generalized</i> likelihood function at time k
$\varphi_k(\mathbf{Z} \mathbf{X})$	Multi-object likelihood function
N	Number of particles in the standard particle filter
\mathcal{J}	Information gain
\mathbf{u}, \mathbf{v}	Control vector (action)
\mathbb{U}_k	Admissible set of controls (actions) at discrete-time k
$D(\mathbf{x})$	Intensity function or probability hypothesis density (PHD)
d	Distance metric
$\rho(n)$	Cardinality distribution
λ	Mean number of false detections in a measurement set \mathbf{Z}_k
$c(\mathbf{z})$	Distribution of false detections over \mathcal{Z}
p_D	Probability of detection
p_S	Probability of survival
\mathbf{p}	Position vector
\mathbf{v}	Velocity vector

θ	Parameter vector
$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \mathbf{P})$	Gaussian PDF with mean $\boldsymbol{\mu}$ and covariance \mathbf{P}
$\mathcal{C}(\mathbf{x}; \boldsymbol{\mu}, \mathbf{P})$	Gaussian CDF with mean $\boldsymbol{\mu}$ and covariance \mathbf{P}
$\mathcal{U}_{[a,b]}(x)$	Uniform density over x with support $[a, b]$
\mathbb{R}	Set of real numbers
\mathbb{N}	Set of natural numbers
\mathbb{N}_0	Set of natural numbers including zero
$\langle a, b \rangle$	Inner product between $a(\mathbf{x})$ and $b(\mathbf{x})$
\otimes	Kronecker product

Chapter 1

Introduction

The book is about stochastic filtering in the framework of Bayesian probability theory. The problem is of paramount importance in many fields of science and engineering. It deals with estimation of dynamic stochastic systems (objects, phenomena) from noisy measurements collected by sensors.

The roots of stochastic filtering theory can be traced back to the early 1960s. Kalman and Bucy [1, 2] formulated the linear filtering theory, while Stratonovich [3] and Kusner [4] pioneered the development of the probabilistic approach to nonlinear filtering.

Discrete-time formulation of the stochastic filtering problem in the Bayesian framework is as follows. Suppose the state vector $\mathbf{x}_k \in \mathcal{X}$ provides the complete specification of the state of a dynamic system (object, phenomenon) at time t_k . Here $\mathcal{X} \subseteq \mathbb{R}^{n_x}$ is the state space, while k is the discrete-time index corresponding to t_k .

The stochastic dynamic system is described by two equations:

$$\mathbf{x}_k = \mathbf{f}_{k-1}(\mathbf{x}_{k-1}) + \mathbf{v}_{k-1}, \quad (1.1)$$

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{w}_k, \quad (1.2)$$

referred to as the *dynamics equation* and the *measurement equation*, respectively. Function $\mathbf{f}_{k-1} : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_x}$ is a nonlinear transition function defining the evolution of the state vector as a first-order Markov process. Random process $\mathbf{v}_k \in \mathbb{R}^{n_x}$ is independent identically distributed (IID) according to the probability density function (PDF) $p_{\mathbf{v}}$; \mathbf{v}_k is referred to as *process noise*, and its role is to model random disturbances during the state evolution. The dimension of the state vector (and the process noise vector) is $n_x \in \mathbb{N}$. Function $\mathbf{h}_k : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_z}$ defines the relationship between the state \mathbf{x}_k and the measurement $\mathbf{z}_k \in \mathcal{Z}$, where $\mathcal{Z} \subseteq \mathbb{R}^{n_z}$ is the measurement space. Random process $\mathbf{w}_k \in \mathbb{R}^{n_z}$, independent of \mathbf{v}_k , is also IID with PDF $p_{\mathbf{w}}$, and referred to as *measurement noise*; n_z is the dimensions of the measurement vector.

In the formulation specified by (1.1–1.2), the functions \mathbf{f}_k and \mathbf{h}_k , the PDFs $p_{\mathbf{v}}$ and $p_{\mathbf{w}}$, and the initial state PDF $p_0(\mathbf{x}_0)$, are assumed known. Equations

(1.1) and (1.2) effectively define two probability functions, the *transitional density* $\pi_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1}) = p_{\mathbf{v}}(\mathbf{x}_k - \mathbf{f}_{k-1}(\mathbf{x}_{k-1}))$ and the *likelihood function* $g_k(\mathbf{z}_k|\mathbf{x}_k) = p_{\mathbf{w}}(\mathbf{z}_k - \mathbf{h}_k(\mathbf{x}_k))$. The goal of stochastic filtering in the Bayesian framework is to estimate recursively the posterior PDF of the state, denoted $p(\mathbf{x}_k|\mathbf{z}_{1:k})$, with $\mathbf{z}_{1:k} \equiv \mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_k$.

The solution is usually presented as a two step procedure. Let $p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$ denote the posterior PDF at $k-1$. The first step *predicts* the density of the state to time k (when measurement \mathbf{z}_k is available) via the Chapman–Kolmogorov equation [5]:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int \pi_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) d\mathbf{x}_{k-1} \quad (1.3)$$

The second step applies the Bayes rule to *update* the predicted PDF using the measurement \mathbf{z}_k :

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \frac{g_k(\mathbf{z}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{z}_{1:k-1})}{\int g_k(\mathbf{z}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) d\mathbf{x}_k} \quad (1.4)$$

Knowing the posterior $p(\mathbf{x}_k|\mathbf{z}_{1:k})$, one can compute point estimates of the state, e.g. the expected a posterior (EAP) estimate or the maximum a posterior (MAP) estimate.

The closed-form analytic solution to (1.3–1.4) can be found only in some special cases. One important case is when \mathbf{f}_k and \mathbf{h}_k are linear functions and PDFs $p_{\mathbf{v}}$ and $p_{\mathbf{w}}$ are Gaussian; the solution in this case is the Kalman filter. In general, however, stochastic filtering via (1.3–1.4) can be solved only approximately. Many approximations have been proposed over decades of research, including analytic approximations (e.g. Extended Kalman filter and its variants), grid-based methods (where the posterior PDF is evaluated at a finite fixed set of points), Gaussian sum filters (where the posterior PDF is approximated by a Gaussian mixture), unscented transforms and particle filters [6, 7].

The particle filter (PF) is a sequential Monte Carlo method which provides a general framework for numerical (simulation based) solution of the stochastic Bayes filter specified by (1.3–1.4). Particle filters are attractive because, with sufficiently large samples, they approach the optimal estimate of the posterior PDF. Their drawback is the computational cost, which fortunately, with ever faster computer hardware, is becoming less relevant. Particle filters have become enormously popular and consequently several books and tutorials [6, 8–12] have been devoted to them. A brief overview of particle filters is also given in the next chapter.

The standard Bayes stochastic filter, typically implemented as a particle filter, despite its popularity and numerous applications in various fields of human endeavor (e.g., ecology, economics, robotics, navigation) is very restrictive. Here are some examples highlighting the limitations of the formulation of the Bayes filter (1.3–1.4) and its particle filter implementations. The formulation (1.3–1.4) does not cover situations where the dynamic system switches (in some probabilistic manner) *on* and *off*. Switching is very common; for example: moving objects enter and leave a surveillance area; an outbreak of an epidemic occurs at some point of time and

eventually disappears. The formulation above is also restricted to a single dynamic system or object. In many applications, however, multiple dynamic objects can appear and disappear during the observation period. Furthermore, the measurement model is very restrictive, because it assumes perfect detection (no false detections and no miss detections). The measurement model also assumes that the measurement function \mathbf{h}_k is precisely known and that measurements \mathbf{z}_k are precise (i.e., points in the measurement space). There are many applications where these two assumptions are not valid. The measurement function often features parameters which are not known precisely (e.g., sensor location, orientation, gain, environmental coefficients such as propagation loss, etc). Similarly, the measurements coming from some sources of information (e.g., natural language statements, sensors with bounded measurement errors) are better modeled by intervals (crisp or fuzzy), than by points. The measurements which are imprecise or which result from imprecise measurement functions are referred to as *non-standard measurements*.

Recently Goodman, Mahler and their co-workers [13, 14], formulated the Bayes filtering problem in the random set framework. This framework was proposed because it can cover, in an elegant but mathematically rigorous manner, all the situations that the standard Bayes filter (1.3–1.4) cannot, including multiple appearing/disappearing objects, imperfect detection and non-standard measurements. Over the last decade, several remarkable Bayes filters have emerged from the random set framework. In the most general setting, these random set based stochastic filters can be implemented only using the sequential Monte Carlo method. They include the *the Bernoulli filter* [14–17], *the probability hypothesis density (PHD) filter* [18–21], *the cardinalized PHD filter* [22, 23], *the multi-Bernoulli filter* [14, 24] and *the exact multi-object Bayes filter* [14, 19, 25–27]. Bayes filters using random set models have also been applied to non-standard measurements: the Bernoulli filter for interval measurements is presented in [28], localization using natural language statements is considered in [29] and imprecise measurement models in [30].

This monograph is dedicated to particle filter implementation of the most popular stochastic filters that emerged from the random set theoretical framework. Practical applications involving both the standard and non-standard measurement models are included when appropriate to illustrate the theoretical concepts.

References

1. R. E. Kalman, “A new approach to linear filtering and prediction problems”, *Trans. ASME — Journal of Basic Engineering*, vol. 82 (Series D), pp. 35–45, 1960.
2. R. E. Kalman and R. S. Bucy, “New results in linear filtering and prediction theory”, *Trans. ASME - Journ. Basic Engineering*, vol. 83 (Series D), pp. 95–108, 1961.
3. R. L. Stratonovich, “Conditional Markov processes”, *Theory Prob. Appl.*, vol. 5, no. 2, pp. 156–178, 1960.
4. H. J. Kushner, “On the dynamical equations of conditional probability density functions with applications to optimal stochastic control theory”, *Journ. Math. Anal.*, vol. 8, pp. 332–344, 1964.

5. A. H. Jazwinski, *Stochastic processes and filtering theory*. Academic press, 1970.
6. B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman filter: Particle filters for tracking applications*. Artech House, 2004.
7. S. Challa, M. R. Morelande, D. Muicki, and R. J. Evans, *Fundamentals of Object Tracking*. Cambridge Univ. Press, 2011.
8. A. Doucet, J. F. G. de Freitas, and N. J. Gordon, eds., *Sequential Monte Carlo Methods in Practice*. Springer, 2001.
9. M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for non-linear/non-Gaussian Bayesian tracking”, *IEEE Trans. Signal Processing*, vol. 50, pp. 174–188, Feb. 2002.
10. P. Djuric, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. Bugallo, and J. Miguez, “Particle filtering”, *IEEE Signal Processing Magazine*, pp. 19–38, Sept. 2003.
11. O. Cappé, S. J. Godsill, and E. Moulines, “An overview of existing methods and recent advances in sequential Monte Carlo”, *Proc. IEEE*, vol. 95, no. 5, pp. 899–924, 2007.
12. A. Doucet and A. M. Johansen, “A tutorial on particle filtering and smoothing: Fifteen years later”, tech. rep., Department of Statistics, University of British Columbia, Dec. 2008.
13. I. R. Goodman, R. P. S. Mahler, and H. T. Nguyen, *Mathematics of data fusion*. Springer, 1997.
14. R. Mahler, *Statistical Multisource Multitarget Information Fusion*. Artech House, 2007.
15. B.-T. Vo, D. Clark, B.-N. Vo, B. Ristic, “Bernoulli forward-backward smoothing for joint target detection and tracking”, *IEEE Trans. Signal Processing*, vol. 59, pp. 4473–4477, Sept. 2011.
16. B. Ristic and S. Arulampalam, “Bernoulli particle filter with observer control for bearings only tracking in clutter”, *IEEE Trans Aerospace and Electronic Systems*, vol. 48, July 2012.
17. B.-T. Vo, C.-M. See, N. Ma, and W. T. Ng, “Multi-sensor joint detection and tracking with the Bernoulli filter”, *IEEE Trans. Aerospace and Electronic Systems*, 2012.
18. R. P. S. Mahler, “Multi-target Bayes filtering via first-order multi-target moments”, *IEEE Trans. Aerospace & Electronic Systems*, vol. 39, no. 4, pp. 1152–1178, 2003.
19. B.-N. Vo, S. Singh, and A. Doucet, “Sequential Monte Carlo methods for multi-target filtering with random finite sets”, *IEEE Trans. Aerospace & Electronic Systems*, vol. 41, pp. 1224–1245, Oct. 2005.
20. N. P. Whiteley, S. S. Singh, and S. J. Godsill, “Auxiliary particle implementation of the probability hypothesis density filter”, *IEEE Trans. on Aerospace & Electronic Systems*, vol. 46, pp. 1437–1454, July 2010.
21. B. Ristic, D. Clark, B.-N. Vo, and B.-T. Vo, “Adaptive target birth intensity for PHD and CPHD filters”, *IEEE Trans. Aerospace and Electronic Systems*, vol. 48, pp. 1656–1668, April 2012.
22. R. P. S. Mahler, “PHD filters of higher order in target number”, *IEEE Trans. Aerospace & Electronic Systems*, vol. 43, no. 4, pp. 1523–1543, 2007.
23. B.-T. Vo, B.-N. Vo, and A. Cantoni, “Analytic implementations of the cardinalized probability hypothesis density filter”, *IEEE Trans. Signal Processing*, vol. 55, no. 7, pp. 3553–3567, 2007.
24. B.-T. Vo, B. Vo, and A. Cantoni, “The cardinality balanced multi-target multi-Bernoulli filter and its implementations”, *IEEE Trans. Signal Processing*, vol. 57, no. 2, pp. 409–423, 2009.
25. H. Sidenbladh and S. L. Wirkander, “Tracking random sets of vehicles in terrain”, in *Proc. 2nd IEEE Workshop on Multi-Object Tracking*, (Madison, WI, USA), June 2003.
26. T. Zajic and R. Mahler, “A particle-systems implementation of the PHD multitarget tracking filter”, in *Proc. SPIE*, vol. 5096, pp. 291–299, April 2003.
27. B.-T. Vo and B.-N. Vo, “A random finite set conjugate prior and application to multi-target tracking”, in *Proc. IEEE Conf. ISSNIP 2011*, (Adelaide, Australia), pp. 431–436, Dec. 2011.
28. A. Gning, B. Ristic, and L. Mihaylova, “Bernoulli particle/box-particle filters for detection and tracking in the presence of triple measurement uncertainty”, *IEEE Trans. Signal Processing*, vol. 60, pp. 2138–2151, May 2012.
29. A. Bishop and B. Ristic, “Fusion of natural language propositions: Bayesian random set framework”, in *Proc. 14th Int. Conf. Information Fusion*, (Chicago, USA), July 2011.
30. B. Ristic, “Bayesian estimation with imprecise likelihoods: Random set approach”, *IEEE Signal Processing Letters*, vol. 18, pp. 395–398, July 2011.

Chapter 2

Background

This chapter reviews the fundamentals of particle filtering with sensor control and introduces formally the most popular Bayes filters that emerged from the random set theoretical framework.

2.1 A Brief Review of Particle Filters

Detailed presentation on particle filters can be found in [1–6]. Here we review only the basic concepts. Suppose at time $k - 1$ the posterior density is approximated by a set of weighted particles $\{w_{k-1}^{(i)}, \mathbf{x}_{k-1}^{(i)}\}_{i=1}^N$, where $\mathbf{x}_{k-1}^{(i)}$ is the state of particle i and $w_{k-1}^{(i)}$ is its weight. The weights are normalized, i.e., $\sum_{i=1}^N w_{k-1}^{(i)} = 1$. This approximation can be expressed as:

$$p(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1}) \approx \sum_{i=1}^N w_{k-1}^{(i)} \delta_{\mathbf{x}_{k-1}^{(i)}}(\mathbf{x}_{k-1}) \quad (2.1)$$

where $\delta_{\mathbf{a}}(\mathbf{x})$ is the Dirac delta function concentrated at \mathbf{a} . The particle filter is based on the concept of importance sampling [7]. Suppose $q_k(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{z}_k)$ is a proposal or importance density whose support contains the support of the posterior PDF at time k . Then the posterior at k , $p(\mathbf{x}_k | \mathbf{z}_{1:k})$ can be approximated by a new set of weighted particles $\{w_k^{(i)}, \mathbf{x}_k^{(i)}\}_{i=1}^N$, that is

$$p(\mathbf{x}_k | \mathbf{z}_{1:k}) \approx \sum_{i=1}^N w_k^{(i)} \delta_{\tilde{\mathbf{x}}_k^{(i)}}(\mathbf{x}_k) \quad (2.2)$$

where

$$\tilde{\mathbf{x}}_k^{(i)} \sim q_k(\mathbf{x}_k | \mathbf{x}_{k-1}^{(i)}, \mathbf{z}_k) \quad (2.3)$$

$$\tilde{w}_k^{(i)} = w_{k-1}^{(i)} \frac{g_k(\mathbf{z}_k | \mathbf{x}_k^{(i)}) \pi_{k|k-1}(\mathbf{x}_k^{(i)} | \mathbf{x}_{k-1}^{(i)})}{q_k(\mathbf{x}_k^{(i)} | \mathbf{x}_{k-1}^{(i)}, \mathbf{z}_k)} \quad (2.4)$$

$$w_k^{(i)} = \frac{\tilde{w}_k^{(i)}}{\sum_{j=1}^N \tilde{w}_k^{(j)}} \quad (2.5)$$

The recursion is initialized by sampling N times from the initial PDF p_0 .

The procedure above, also known as sequential importance sampling (SIS), inevitably fails after a few iterations. This is because all particle weights, except for a few, become zero, and thus this leads to poor approximation of the posterior PDF. The collapse of SIS can be prevented by resampling the particles. The resampling step chooses N particles from $\{w_k^{(i)}, \tilde{\mathbf{x}}_k^{(i)}\}_{i=1}^N$, where the selection of particles is based on their weights: the probability of particle i being selected equals $w_k^{(i)}$. After resampling, all particle weights are equal to $1/N$. While resampling avoids degeneracy of particles, it leads to the loss of diversity among the particles, because the particles with large weights are selected (repeated) many times. In order to increase the particle diversity it is usually recommended to perform a Markov chain Monte Carlo (MCMC) move step after resampling.

The choice of the importance density q_k is also very important for efficient implementation of particle filters. The simplest choice is to select q_k as the transitional density, i.e., $q_k \equiv \pi_{k|k-1}$. This choice can lead to poor performance unless a massive number of particles are used. The problem is that many of the particles could be sampled from the region of the state space which is not in the support of the posterior. In general it is better to use the information contained in the latest measurement \mathbf{z}_k when you design the importance density. This can be achieved by the so-called optimal importance density (OID). The limitation is that the OID can be derived only for some special classes of model (1.1–1.2). Research into good importance densities has resulted in several versions of particle filters, such as the auxiliary PF and the unscented PF. Another very useful technique for the implementation of particle filters (and importance sampling in general) is the progressive correction or tempering [8] (see for more details Sect. 7.2.3). The bootstrap filter [9] is the simplest and possibly the most popular particle filter; the pseudocode of a single cycle of the bootstrap filter is given in Algorithm 1. The main features of the bootstrap filter are: (1) the importance density is the transitional PDF, i.e., $q_k \equiv \pi_{k|k-1}$ and (2) the resampling step is performed in every cycle $k = 1, 2, \dots$, and therefore there is no need to input/output particle weights.

2.2 Online Sensor Control

Modern sensors are capable of a variety of actions, such as looking in certain directions, moving to other locations, using different modes of operation (waveform, beam-pattern), etc. The objective of sensor control is the online selection of actions

Algorithm 1 Pseudocode of the bootstrap filter

```

1: function BOOTSTRAP FILTER
2:   Input:  $\{\mathbf{x}_{k-1}^{(i)}\}_{i=1}^N; \mathbf{z}_k$ 
3:   for  $i = 1, \dots, N$  do
4:     Draw a sample:  $\tilde{\mathbf{x}}_k^{(i)} \sim \pi_{k|k-1}(\mathbf{x}_k | \mathbf{x}_{k-1}^{(i)})$ 
5:     Calculate weight  $\tilde{w}_k^{(i)} = g_k(\mathbf{z}_k | \mathbf{x}_k^{(i)})$ 
6:   end for
7:    $w_k^{(i)} = \tilde{w}_k^{(i)} / \sum_{j=1}^N \tilde{w}_k^{(j)}$ , for  $i = 1, \dots, N$ 
8:   for  $i = 1, \dots, N$  do ▷ (Resampling)
9:     Select index  $j^i \in \{1, \dots, N\}$  with probability  $w_k^{(i)}$ 
10:     $\mathbf{x}_k^{(i)} = \tilde{\mathbf{x}}_k^{(j^i)}$ 
11:   end for
12:   Output:  $\{\mathbf{x}_k^{(i)}\}_{i=1}^N$ 
13: end function

```

to be taken by individual sensors, in order to maximize the overall utility of the surveillance system. Sensor control thus represents sequential decision making, where each decision generates measurements that provide additional information for estimation. With sensor control, sequential estimation (stochastic filtering) becomes a closed-loop system.

Decisions are made in the presence of uncertainty both in the state space and in the measurement space. The assumption is that past decisions and past observations are available when making the next decision. This class of problems has been studied in the framework of partially observed Markov decision processes (POMDPs) [10]. The elements of a POMDP include the (uncertain) current information state, a set of admissible sensor actions, and the reward function associated with each action. We adopt the information theoretic approach to sensor control, where the uncertain state is represented by the posterior PDF, while the reward function is a measure of the *information gain* assigned to each action.

Let $\mathbf{u}_{k-1} \in \mathbb{U}_{k-1}$ denote the control vector (i.e. action) applied to the sensor/observer at time t_{k-1} in order to collect the next measurement at t_k . Here \mathbb{U}_{k-1} represents the set of admissible control vectors at time t_{k-1} . The notation for the posterior PDF of the Bayes filter with sensor control typically is extended to include the sequence of applied control vectors, that is, the posterior at time $k - 1$ is $p(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1}, \mathbf{u}_{1:k-2})$, where action \mathbf{u}_{k-2} resulted in measurement \mathbf{z}_{k-1} . In general both the transitional density $\pi_{k|k-1}$ and the likelihood function g_k depend on action $\mathbf{u}_{k-1} \in \mathbb{U}_{k-1}$. The stochastic filtering Eqs. (1.3–1.4) can then be rewritten as follows:

$$p(\mathbf{x}_k | \mathbf{z}_{1:k-1}, \mathbf{u}_{0:k-1}) = \int \pi_{k|k-1}(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_{k-1}) p(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1}, \mathbf{u}_{0:k-2}) d\mathbf{x}_{k-1} \quad (2.6)$$

$$p(\mathbf{x}_k | \mathbf{z}_{1:k}, \mathbf{u}_{0:k-1}) = \frac{g_k(\mathbf{z}_k | \mathbf{x}_k, \mathbf{u}_{k-1}) p(\mathbf{x}_k | \mathbf{z}_{1:k-1}, \mathbf{u}_{0:k-1})}{\int g_k(\mathbf{z}_k | \mathbf{x}_k, \mathbf{u}_{k-1}) p(\mathbf{x}_k | \mathbf{z}_{1:k-1}, \mathbf{u}_{0:k-1}) d\mathbf{x}_k}. \quad (2.7)$$

The optimal one-step ahead control vector to be applied at time $k - 1$ (in order to collect the future measurement \mathbf{z}_k at t_k) is defined by:

$$\mathbf{u}_{k-1} = \arg \max_{\mathbf{v} \in \mathbb{U}_{k-1}} \mathbb{E} \{ \mathcal{J}(\mathbf{v}, p(\mathbf{x}_k | \mathbf{z}_{1:k-1}, \mathbf{u}_{1:k-2}), \mathbf{z}_k(\mathbf{v})) \} \quad (2.8)$$

where $\mathcal{J}(\mathbf{v}, p, \mathbf{z})$ is the real-valued reward function associated with the control \mathbf{v} , which will result in the predicted PDF p and the (future) measurement \mathbf{z} . The fact that the reward function \mathcal{J} depends on the future measurement is undesirable, since we want to decide on the best action without actually applying all of them before the decision is made. For this reason (2.8) includes the expectation operator \mathbb{E} , which is taken with respect to the prior measurement PDF $p(\mathbf{z}_k | \mathbf{z}_{1:k-1}, \mathbf{u}_{1:k-2}, \mathbf{v})$, where $\mathbf{v} \in \mathbb{U}_{k-1}$.

Sensor control via (2.8) maximizes the reward based on a single future step only (the so-called “myopic” policy). Sensor control policy may also look many steps ahead, in order to anticipate object motion and provide a long-term estimation gain. Due to the uncertainty both in the current knowledge of the state and in the state evolution model, looking too many steps ahead may become counterproductive. In addition, the computational cost grows exponentially with the number of steps ahead.

Reward functions typically measure the reduction in uncertainty or the information gain, in comparison with the current information state. The information gain can be formulated using various information measures, such as the Fisher information, entropy, Kullback-Leibler (KL) divergence, etc [11]. Throughout this monograph, the reward function based on the Rényi divergence will be primarily considered. The Rényi divergence between any two densities, $p_0(\mathbf{x})$ and $p_1(\mathbf{x})$, is defined as [11, 12]:

$$I_\alpha(p_1, p_0) = \frac{1}{\alpha - 1} \log \int p_1^\alpha(\mathbf{x}) p_0^{1-\alpha}(\mathbf{x}) d\mathbf{x} \quad (2.9)$$

where $\alpha \geq 0$ is a free parameter which determines how much we emphasize the tails of two distributions in the metric. In the special cases of $\alpha \rightarrow 1$ and $\alpha = 0.5$, $I_\alpha(p_1, p_0)$ reduces to the KL divergence and the Bhattacharyya distance, respectively [11].

The reward function $\mathcal{J}(\mathbf{u}_{k-1}, p, \mathbf{z})$ in (2.8) is adopted as the Rényi divergence between:

- the predicted PDF $p(\mathbf{x}_k | \mathbf{z}_{1:k-1}, \mathbf{u}_{0:k-1})$ of (2.6), where the prediction is based on \mathbf{u}_{k-1} , and
- the future posterior $p(\mathbf{x}_k | \mathbf{z}_{1:k}, \mathbf{u}_{0:k-1})$ of (2.7), obtained using the new measurement \mathbf{z}_k collected after the sensor has been controlled to take action \mathbf{u}_{k-1} .

According to (2.9) the reward function can be written as (in order to simplify notation we suppress the second and third arguments of \mathcal{J}):

$$\mathcal{J}(\mathbf{u}_{k-1}) = \frac{1}{\alpha - 1} \log \int [p(\mathbf{x}_k | \mathbf{z}_{1:k}, \mathbf{u}_{0:k-1})]^\alpha [p(\mathbf{x}_k | \mathbf{z}_{1:k-1}, \mathbf{u}_{0:k-2})]^{1-\alpha} d\mathbf{x}_k \quad (2.10)$$

Since the analytic solution to the computation of the expected reward function $\mathbb{E}[\mathcal{J}(\mathbf{u}_{k-1})]$ does not exist, we need to employ numerical approximations. For this purpose, particle filters become particularly useful. For a reasonably small cardinality of the set \mathbb{U}_{k-1} , and adopting the bootstrap filter, the optimal one-step ahead control vector selection can be implemented according to the pseudocode in Algorithm 2.

Algorithm 2 Pseudocode for the selection of the control vector

```

1: function CONTROL VECTOR SELECTION
2:   Input:  $\{\mathbf{x}_{k-1}^{(i)}\}_{i=1}^N; \mathbb{U}_{k-1}$ 
3:   for all  $\mathbf{v} \in \mathbb{U}_{k-1}$  do
4:      $\mathbf{x}_k^{(i)} \sim \pi_{k|k-1}(\mathbf{x}_k | \mathbf{x}_{k-1}^{(i)}, \mathbf{v})$ , for  $i = 1, \dots, N$ 
5:     for  $m = 1, \dots, M$  do ▷ Typically  $M \ll N$ 
6:       Select at random index  $j_m$  from  $\{1, \dots, N\}$ 
7:        $\mathbf{z}_k^{(m)} \sim g_k(\mathbf{z}_k | \mathbf{x}_k^{(j_m)}, \mathbf{v})$ 
8:        $\tilde{w}_k^{(i,m)} = g_k(\mathbf{z}_k^{(m)} | \mathbf{x}_k^{(i)}, \mathbf{v})$ , for  $i = 1, \dots, N$ 
9:        $w_k^{(i,m)} = \tilde{w}_k^{(i,m)} / \sum_{j=1}^N \tilde{w}_k^{(j,m)}$ , for  $i = 1, \dots, N$ 
10:      Gain  $\mathcal{J}^{(m)}(\mathbf{v}) \approx \frac{1}{\alpha-1} \log N^{\alpha-1} \sum_{i=1}^N [w_k^{(i,m)}]^\alpha$ 
11:    end for
12:     $\mathbb{E}[\mathcal{J}(\mathbf{v})] \approx \frac{1}{M} \sum_{m=1}^M \mathcal{J}^{(m)}(\mathbf{v})$ 
13:  end for
14:   $\mathbf{u}_{k-1} = \arg \max_{\mathbf{v} \in \mathbb{U}_{k-1}} \mathbb{E}\{\mathcal{J}(\mathbf{v})\}$  ▷ Eq.(2.8)
15:  Output:  $\mathbf{u}_{k-1}$ 
16: end function

```

The expected gain $\mathbb{E}[\mathcal{J}(\mathbf{v})]$ in Algorithm 2 is computed as the sample average over M realizations of the hypothetical future measurement caused by the action \mathbf{v} , see line 12. While the pseudocode in Algorithm 2 is self-explanatory, line 10 requires further explanation.

Let the predicted PDF $p(\mathbf{x}_k | \mathbf{z}_{1:k-1}, \mathbf{u}_{0:k-1})$ and the updated PDF $p(\mathbf{x}_k | \mathbf{z}_{1:k}, \mathbf{u}_{0:k-1})$, which both feature in (2.10), be approximated by particle systems $\{w_{k-1}^{(i)}, \tilde{\mathbf{x}}_k^{(i)}\}_{i=1}^N$ and $\{w_k^{(i)}, \tilde{\mathbf{x}}_k^{(i)}\}_{i=1}^N$, respectively. Computation of $\tilde{\mathbf{x}}_k^{(i)}$ and $w_k^{(i)}$ was explained by (2.3–2.5). Note that both particle systems have the same support (i.e., consist of identical particles); the only difference is in the values of their weights. Then we can approximate (2.10) by [13]:

$$\mathcal{J}(\mathbf{u}_{k-1}) \approx \frac{1}{\alpha-1} \log \int \left[\sum_{i=1}^N w_k^{(i)} \delta_{\tilde{\mathbf{x}}_k^{(i)}}(\mathbf{x}) \right]^\alpha \cdot \left[\sum_{i=1}^N w_{k-1}^{(i)} \delta_{\tilde{\mathbf{x}}_k^{(i)}}(\mathbf{x}) \right]^{1-\alpha} d\mathbf{x} \quad (2.11)$$

$$\approx \frac{1}{\alpha - 1} \log \int \left[\sum_{i=1}^N \left(w_k^{(i)} \right)^\alpha \delta_{\tilde{\mathbf{x}}_k^{(i)}}^\alpha(\mathbf{x}) \right] \left[\sum_{i=1}^N \left(w_{k-1}^{(i)} \right)^{1-\alpha} \delta_{\tilde{\mathbf{x}}_k^{(i)}}^{1-\alpha}(\mathbf{x}) \right] d\mathbf{x} \quad (2.12)$$

When the sum of delta functions in (2.11) is raised to an exponent, the “cross-terms” vanish due to the different support of delta functions, which leads to approximation (2.12). Furthermore, when two sums of delta functions are multiplied as in (2.12), the “cross-terms” vanish for the same reason. Thus the approximation further simplifies as follows:

$$\mathcal{J}(\mathbf{u}_{k-1}) \approx \frac{1}{\alpha - 1} \log \int \sum_{i=1}^N \left(w_k^{(i)} \right)^\alpha \left(w_{k-1}^{(i)} \right)^{1-\alpha} \delta_{\tilde{\mathbf{x}}_k^{(i)}}(\mathbf{x}) d\mathbf{x} \quad (2.13)$$

$$\approx \frac{1}{\alpha - 1} \log \sum_{i=1}^N \left(w_k^{(i)} \right)^\alpha \cdot \left(w_{k-1}^{(i)} \right)^{1-\alpha} \quad (2.14)$$

Recall that the predicted weights $w_{k-1}^{(i)}$ of the bootstrap filter are all equal to $1/N$. The expression in line 10 of Algorithm 2 then follows from (2.14).

2.3 Non-standard Measurements

Non-traditional measurements can be incorporated into the Bayesian estimation framework via the *generalized likelihood function* (GLF) [14].

2.3.1 Imprecise Measurements

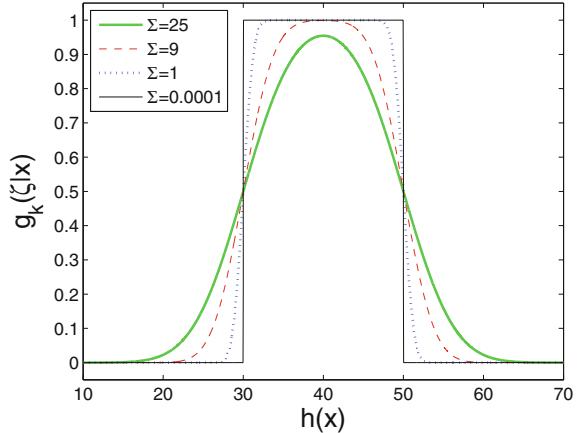
Let us first consider imprecise measurements. Imprecision is a form of uncertainty distinctly different from randomness, being a consequence of the *lack of knowledge* [15]. Recall that a standard point measurement is modeled by (1.2). An imprecise measurement, denoted by ζ , is modeled by a random closed set on \mathcal{X} [14], Chaps. 4–7. In general it can be an interval (crisp or fuzzy) or a weighted sum of intervals. The generalized likelihood of ζ_k is defined by [14, 16]:

$$\tilde{g}_k(\zeta_k | \mathbf{x}_k) = \Pr\{\mathbf{z}_k \in \zeta_k\} = \Pr\{\mathbf{h}_k(\mathbf{x}_k) + \mathbf{w}_k \in \zeta_k\} \quad (2.15)$$

In general the GLF does not integrate to 1 and can be represented by a weighted sum of the membership functions which describe the fuzzy sets on \mathcal{X} .

Example. Let $\mathcal{N}(\mathbf{y}; \boldsymbol{\mu}, \mathbf{P})$ denote a Gaussian PDF with mean $\boldsymbol{\mu}$ and covariance \mathbf{P} ; its cumulative distribution function (CDF) is $\mathcal{C}(\mathbf{y}; \boldsymbol{\mu}, \mathbf{P}) = \int_{-\infty}^{\mathbf{y}} \mathcal{N}(\mathbf{u}; \boldsymbol{\mu}, \mathbf{P}) d\mathbf{u}$.

Fig. 2.1 Illustration of the GLF (2.16) for 1D interval measurement $\zeta = [30, 50]$ affected by $w \sim \mathcal{N}(w; \mathbf{0}, \Sigma)$ with $\Sigma = 25, 9, 1, 0.0001$



Suppose that measurement noise \mathbf{w} in (1.12) is zero mean white Gaussian with covariance Σ , i.e. $p_{\mathbf{w}}(\mathbf{w}) = \mathcal{N}(\mathbf{w}; \mathbf{0}, \Sigma)$. If ζ is a crisp interval with lower and upper limits $\underline{\zeta}$ and $\bar{\zeta}$, respectively, then the generalized likelihood can be expressed analytically as [16]:

$$\begin{aligned}\tilde{g}_k(\zeta|\mathbf{x}) &= \int_{\underline{\zeta}}^{\bar{\zeta}} \mathcal{N}(\mathbf{z}; h(\mathbf{x}), \Sigma) d\mathbf{z} \\ &= \mathcal{C}(h(\mathbf{x}); \underline{\zeta}, \Sigma) - \mathcal{C}(h_k(\mathbf{x}); \bar{\zeta}, \Sigma).\end{aligned}\quad (2.16)$$

The shape of the GLF (2.16) is illustrated for a one-dimensional measurement space $\mathcal{Z} \subset \mathbb{R}$ in Fig. 2.1. In this example $\underline{\zeta} = 30$, $\bar{\zeta} = 50$ and Σ is 25, 9, 1, and 0.0001. Note that the GLF in Fig. 2.1 has the shape of a membership function representing a fuzzy set. \square

If the measurements are imprecise, the update equation of the stochastic filter (1.4) is modified in the sense that the standard likelihood $g_k(\mathbf{z}_k|\mathbf{x}_k)$ is replaced with the GLF $\tilde{g}_k(\zeta_k|\mathbf{x}_k)$. The theoretical justification of this claim can be found in [14, 17, 18].

2.3.2 Imprecise Measurement Function

Precise measurement $\mathbf{z} \in \mathcal{Z}$ which results from an imprecisely specified measurement function is also an instance of a non-standard measurement (sometimes this is referred to as the *imprecise likelihood* case). Suppose the measurement function is parameterized by a vector θ , and denoted $\mathbf{h}_k(\mathbf{x}; \theta)$. If θ is not known precisely, but only as an interval value, i.e., $\theta \in [\theta]$, then $\mathbf{h}_k(\mathbf{x}; [\theta]) + \mathbf{w}_k$ maps \mathbf{x} into a closed random set $\Sigma_{\mathbf{x}}$ of \mathcal{Z} . The GLF in this case is defined by:

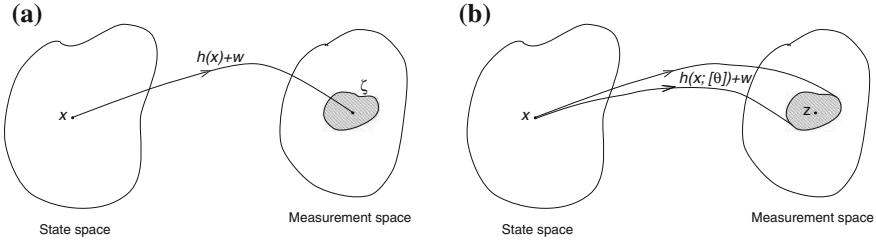


Fig. 2.2 Graphical representation of two types of non-standard measurements: **a** Imprecise measurement ξ , precise measurement function $\mathbf{h}(\mathbf{x})$; **b** precise measurement \mathbf{z} , imprecise measurement function $\mathbf{h}(\mathbf{x}, [\boldsymbol{\theta}])$

$$\tilde{g}_k(\mathbf{z}_k | \mathbf{x}_k) = \Pr\{\mathbf{z}_k \in \Sigma_{\mathbf{x}}\} = \Pr\{\mathbf{z}_k \in \mathbf{h}_k(\mathbf{x}_k; [\boldsymbol{\theta}]) + \mathbf{w}_k\} \quad (2.17)$$

If $p_{\mathbf{w}}(\mathbf{w}) = \mathcal{N}(\mathbf{w}; \mathbf{0}, \boldsymbol{\Sigma})$ then the GLF (2.17) has an analytic solution:

$$\tilde{g}_k(\mathbf{z}_k | \mathbf{x}_k) = \mathcal{C}(\mathbf{z}; \underline{\mathbf{h}}_x, \boldsymbol{\Sigma}) - \mathcal{C}(\mathbf{z}; \bar{\mathbf{h}}_x, \boldsymbol{\Sigma}) \quad (2.18)$$

where $\underline{\mathbf{h}}_x = \inf\{\mathbf{h}_k(\mathbf{x}_k; [\boldsymbol{\theta}])\}$ and $\bar{\mathbf{h}}_x = \sup\{\mathbf{h}_k(\mathbf{x}_k; [\boldsymbol{\theta}])\}$ are the limits of the closed set $\mathbf{h}_k(\mathbf{x}_k; [\boldsymbol{\theta}])$. The similarity between (2.18) and (2.16) is noteworthy, although one should be aware of the different context.

If the measurement function \mathbf{h}_k is imprecise, the update equation of the stochastic filter (1.4) is modified in the sense that the standard likelihood $g_k(\mathbf{z}_k | \mathbf{x}_k)$ is replaced with the GLF $\tilde{g}_k(\mathbf{z}_k | \mathbf{x}_k)$.

Imprecise measurement and imprecise measurement function are graphically illustrated in Fig. 2.2. The combination of an imprecise measurement with an imprecise measurement function naturally also results in a non-standard measurement, for details see [14].

2.3.3 Uncertain Implication Rules

Another instance of non-standard measurements are uncertain implication rules. The rule-based reasoning plays an important role in artificial intelligence for reasoning under uncertainty [19], but has been mainly ignored by the statistical signal processing community. Implication rules represent prior domain knowledge which can be used to reduce the uncertainty in the posterior PDF. Suppose we know that if condition $\mathbf{y} \in \mathbf{Y}$ is satisfied, then the state of dynamic system \mathbf{x} will be in region $\mathbf{X} \subset \mathcal{X}$. Here \mathbf{y} is typically an observation in a measurement space different from \mathcal{X} . For example, suppose the dynamic system of interest is a suspect in a murder case and the objective is to discover his/her whereabouts. The following piece of intelligence is available for estimation (localization): “The suspect is often in the *Corner hotel* between 10 pm and 11 pm”. If the time is indeed between 10 pm and 11 pm, then this

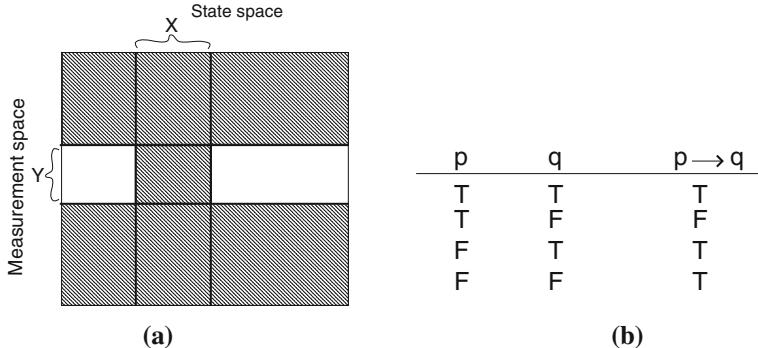


Fig. 2.3 A graphical interpretation of the definition of generalized likelihood in (2.19): **a** Product space $\mathcal{X} \times \mathcal{Y}$, with indicated subsets $\mathbf{X} \subset \mathcal{X}$ and $\mathbf{Y} \subset \mathcal{Y}$; the areas in gray are assigned confidence α ; the areas in white are assigned confidence $1 - \alpha$; **b** the truth table of $p \rightarrow q$

prior knowledge becomes very useful. The rules are in general uncertain, that is, they are trusted only with confidence $\alpha \in [0, 1]$.

Mathematically, the first-order rule R is expressed as: $\mathbf{Y} \Rightarrow \mathbf{X}$. The GLF of an uncertain implication rule is given by [20]:

$$\tilde{g}_R(\mathbf{y}|\mathbf{x}) = \begin{cases} \alpha, & \text{if } (\mathbf{y} \in \mathbf{Y} \wedge \mathbf{x} \in \mathbf{X}) \vee (\mathbf{y} \notin \mathbf{Y} \wedge \mathbf{x} \in \mathcal{X}) \\ 1 - \alpha, & \text{if } \mathbf{y} \in \mathbf{Y} \wedge \mathbf{x} \notin \mathbf{X}. \end{cases} \quad (2.19)$$

Definition (2.19) follows from the propositional logic, whereby an implication $p \rightarrow q$ is logically equivalent to ‘not- p or q ’. It can be verified that (2.19) complies with the truth table of $p \rightarrow q$. In our case the antecedent p is replaced with $\mathbf{y} \in \mathbf{Y}$, consequent q with $\mathbf{x} \in \mathbf{X}$, and confidence α by T (truth) in the truth table of $p \rightarrow q$ in Fig. 2.3b. The gray areas in Fig. 2.3a are assigned confidence α according to (2.19). The rule R is applied to update the posterior PDF only once, and that is when the condition $\mathbf{y} \in \mathbf{Y}$ is satisfied. For $\mathbf{y} \notin \mathbf{Y}$, the rule is uninformative and therefore its GLF does not provide any information about the state of the dynamic system.

2.3.4 Particle Filter Implementation

Particle filter implementation of the Bayes filter using non-standard measurements is effectively the same as when we deal with the standard measurements. The only difference is in the computation of the weights in Eq.(2.4), where $g_k(\mathbf{z}_k|\mathbf{x}_k^{(i)})$ is replaced with:

- $\tilde{g}_k(\xi_k|\mathbf{x}_k^{(i)})$, if the measurement is imprecise;
- $\tilde{g}_k(\mathbf{z}_k|\mathbf{x}_k^{(i)})$, if the measurement function \mathbf{h}_k is imprecise;

- $\tilde{g}_R(\mathbf{y} \in \mathbf{Y} | \mathbf{x}_k^{(i)})$, if antecedent of an implication rule, $\mathbf{y} \in \mathbf{Y}$, is satisfied; here

$$\tilde{g}_R(\mathbf{y} \in \mathbf{Y} | \mathbf{x}_k^{(i)}) = \begin{cases} \alpha, & \text{if } \mathbf{x}_k^{(i)} \in \mathbf{X} \\ 1 - \alpha, & \text{if } \mathbf{x}_k^{(i)} \notin \mathbf{X}. \end{cases} \quad (2.20)$$

Imprecision in general increases the uncertainty and consequently enlarges the support of the posterior PDF. This results in a requirement for more particles in the PF in order to approximate the posterior. In order to reduce the number of particles, and at the same time preserve the accuracy of the approximation, box particles have recently been proposed [21, 22]. A box particle is a sample from the posterior which occupies a small and controllable rectangular region of the state space. Box particle filtering is a synergy between interval analysis and sequential Monte Carlo estimation, and will be reviewed in Sect. 4.1.2.

2.3.5 Applications

Imprecise (fuzzy interval) measurements and uncertain implications rules were used in [20, 23] to express spatially referring natural language (NL) statements. The ultimate goal of this work was to localize an object (person, car) using NL statements spoken by observers; these statements relate the position of the object of interest to known anchors in the measurement space. The typical statements could be: “the object is near the tower”, or “the object is in front of the traffic light”. In these two examples, the anchors are “the tower” and “the traffic light”, while the spatial relationship is expressed by “near” and “in front of”. An automatic system for the computation of the posterior PDF is designed, which transforms the NL statements (by speech recognition and parsing) into the GLFs corresponding to imprecise spatial relationships. Prior knowledge in the form of implication rules can also be used in the update of the posterior PDF.

Imprecise measurement functions have been used for localization of objects in a sensor network measuring the received signal strength (RSS) [24, 25]. The network consisted of 12 sensor nodes, each characterized by partial knowledge of its own location and the propagation factor (due to multi-path) [25]. The parameter vector θ thus consisted of 36 components. It has been verified by simulations that using a sufficient number of particles, the true value of the state is guaranteed to be inside the support of the posterior PDF.

The concept of imprecise likelihood has been applied to model-based (target) classification [26, 27]. It has been demonstrated that when the likelihood functions for classification are imprecise, the Bayesian classifier in the random set framework (using the generalized likelihood functions) results in identical decisions as those produced by the classifier using the belief function theoretical framework. This is a significant result because the belief function theory operates on the space of subsets of the basic state space, and is therefore more computationally demanding than the

Bayesian classifier in the random set framework. Moreover, the Bayes approach is routinely used on both discrete and continuous state spaces, whereas the belief function theory can be used on discrete state space only.

Another application of imprecise measurement functions was presented in [28]. The problem was to detect and predict the size and peak of an epidemic using non-medical observations such as Internet searches (e.g. Google). An epidemic is a dynamic stochastic system governed by stochastic differential equations. The relationship between the number of Internet searches (measurement) and the number of infected people (a component of the state vector) has been established experimentally. The problem is that coefficients in this relationship (i.e. in the measurement function) can at best be represented as interval values.

Note that in some applications involving imprecise measurement functions, given enough measurements, a point estimate of the partially known parameter vector θ can be inferred. This process is called *automatic calibration* and will be discussed in detail in Sect. 7.2.

2.4 Multiple Objects and Imperfect Detection

2.4.1 Random Finite Sets

Suppose at time t_k , where $k = 0, 1, 2 \dots$, there are n_k objects (targets) with states $\mathbf{x}_{k,1}, \dots, \mathbf{x}_{k,n_k}$, taking values in the state space $\mathcal{X} \subseteq \mathbb{R}^{n_x}$. Both the number of targets n_k and their individual states in \mathcal{X} are random and time varying. The multi-target state at k , represented by a finite set

$$\mathbf{X}_k = \{\mathbf{x}_{k,1}, \dots, \mathbf{x}_{k,n_k}\} \in \mathcal{F}(\mathcal{X}), \quad (2.21)$$

can conveniently be modeled as a random finite set (RFS) on \mathcal{X} . Here $\mathcal{F}(\mathcal{X})$ is a set of finite subsets of \mathcal{X} .

The detection process is imperfect meaning that some of the targets in \mathbf{X}_k are detected, while the others are missed. In addition, the detector typically creates false detections. Suppose measurement sets are available at time $k = 1, 2, \dots$. A measurement set at time k contains m_k elements, each taking a value in the observation space $\mathcal{Z} \subseteq \mathbb{R}^{n_z}$. Then the multi-target observation set,

$$\mathbf{Z}_k = \{\mathbf{z}_{k,1}, \dots, \mathbf{z}_{k,m_k}\} \in \mathcal{F}(\mathcal{Z}), \quad (2.22)$$

whose cardinality and individual points in the measurement space are random, can also be modeled by an RFS. Its domain is $\mathcal{F}(\mathcal{Z})$, a set of finite subsets of \mathcal{Z} .

RFS variable is a random variable that takes values as unordered finite sets. The cardinality of an RFS variable \mathbf{X} is random and modeled by a discrete distribution $\rho(n) = \Pr\{|\mathbf{X}| = n\}$, where $n \in \mathbb{N}_0$. An RFS \mathbf{X} is completely specified

by its cardinality distribution $\rho(n)$ and a family of symmetric joint distributions¹ $f_n(\mathbf{x}_1, \dots, \mathbf{x}_n)$, $n \in \mathbb{N}$, that characterise the distribution of its elements over the state space, conditioned on cardinality n .

Since an RFS is nothing but a finite-set random variable, the usual probabilistic descriptors of a random variable, such as the PDF and its statistical moments, can be defined for it. Due to its simplicity, we adopt Mahler's [14] tools for mathematical representation and manipulation of random finite sets, referred to as *finite set statistics (FISST)*. The FISST PDF² of an RFS variable \mathbf{X} is denoted $f(\mathbf{X})$. It is uniquely defined by cardinality distribution $\rho(n)$ and symmetric joint distributions $f_n(\mathbf{x}_1, \dots, \mathbf{x}_n)$. The relationship is:

$$f(\{\mathbf{x}_1, \dots, \mathbf{x}_n\}) = n! \cdot \rho(n) \cdot f_n(\mathbf{x}_1, \dots, \mathbf{x}_n) \quad (2.23)$$

for $n \in \mathbb{N}_0$. Being a PDF, $f(\mathbf{X})$ integrates to one using the *set integral*. The set integral is defined as:

$$\int f(\mathbf{X}) \delta\mathbf{X} := f(\emptyset) + \sum_{n=1}^{\infty} \frac{1}{n!} \int f(\{\mathbf{x}_1, \dots, \mathbf{x}_n\}) d\mathbf{x}_1 \cdots d\mathbf{x}_n \quad (2.24)$$

Now it is straightforward to verify that indeed $f(\mathbf{X})$ integrates to 1:

$$\begin{aligned} \int f(\mathbf{X}) \delta\mathbf{X} &= \rho(0) + \sum_{n=1}^{\infty} \rho(n) \underbrace{\int f_n(\mathbf{x}_1, \dots, \mathbf{x}_n) d\mathbf{x}_1 \cdots d\mathbf{x}_n}_{=1, \text{ being a standard PDF}} \\ &= \sum_{n=0}^{\infty} \rho(n) = 1 \quad (\text{since } \rho(n) \text{ is a discrete distribution}) \end{aligned} \quad (2.25)$$

The cardinality distribution $\rho(n)$ of an RFS variable \mathbf{X} can be obtained from the FISST PDF $f(\mathbf{X})$ as:

$$\rho(n) = \frac{1}{n!} \int f(\{\mathbf{x}_1, \dots, \mathbf{x}_n\}) d\mathbf{x}_1, \dots, d\mathbf{x}_n \quad (2.26)$$

The *intensity function* (also known as the probability hypothesis density or PHD) of an RFS \mathbf{X} is defined as its first statistical moment:

$$D(\mathbf{x}) = \mathbb{E}\{\delta_{\mathbf{X}}(\mathbf{x})\} = \int \delta_{\mathbf{X}}(\mathbf{x}) f(\mathbf{X}) \delta\mathbf{X} \quad (2.27)$$

¹ A joint distribution function $f_n(\mathbf{x}_1, \dots, \mathbf{x}_n)$ is said to be symmetric if its value remains unchanged for all of the $n!$ possible permutations of its variables.

² While the FISST densities are not probability densities, they have been shown to be equivalent to probability densities on $\mathcal{F}(X)$ relative to some reference measure [29]. Subsequently, we do not distinguish between FISST densities and probability densities of random finite sets.

where $\delta_{\mathbf{X}}(\mathbf{x}) = \sum_{\mathbf{w} \in \mathbf{X}} \delta_{\mathbf{w}}(\mathbf{x})$ is the set Dirac delta function, while $\delta_{\mathbf{w}}(\mathbf{x})$ is the standard Dirac delta function concentrated at \mathbf{w} .

Some common RFS variables are the Bernoulli RFS, the IID cluster RFS, the Poisson RFS, the binomial RFS, and the multi-Bernoulli RFS.

Bernoulli RFS

This RFS can either be empty (with probability $1 - r$) or have one element (with probability r), spatially distributed over \mathcal{X} according to the (standard) PDF $p(\mathbf{x})$. The FISST PDF of the Bernoulli RFS \mathbf{X} is given by:

$$f(\mathbf{X}) = \begin{cases} 1 - r, & \text{if } \mathbf{X} = \emptyset \\ r \cdot p(\mathbf{x}), & \text{if } \mathbf{X} = \{\mathbf{x}\}. \end{cases} \quad (2.28)$$

The intensity function of the Bernoulli RFS \mathbf{X} is $D(\mathbf{x}) = r \cdot p(\mathbf{x})$.

IID Cluster RFS

Let the cardinality distribution of independent identically distributed (IID) cluster RFS \mathbf{X} be $\rho(n)$. For a given cardinality, the elements of \mathbf{X} are each IID random variables distributed according to the (standard) PDF $p(\mathbf{x})$ on \mathcal{X} . The FISST PDF of \mathbf{X} is:

$$f(\mathbf{X}) = |\mathbf{X}|! \cdot \rho(|\mathbf{X}|) \prod_{\mathbf{x} \in \mathbf{X}} p(\mathbf{x}) \quad (2.29)$$

The main simplification in comparison with (2.23) is that due to the IID property, the symmetric joint distribution is replaced by a product. The intensity function of an IID cluster RFS is:

$$D(\mathbf{x}) = p(\mathbf{x}) \sum_{n=1}^{\infty} n \rho(n) \quad (2.30)$$

Poisson RFS

A Poisson RFS \mathbf{X} is a special case of the IID cluster RFS, whose cardinality distribution is Poisson, that is:

$$\rho(n) = \frac{e^{-\lambda} \lambda^n}{n!}, \quad n = 0, 1, 2, \dots$$

Its FISST PDF follows from (2.29) and is given by:

$$f(\mathbf{X}) = e^{-\lambda} \prod_{\mathbf{x} \in \mathbf{X}} \lambda p(\mathbf{x})$$

while its intensity function is $D(\mathbf{x}) = \lambda p(\mathbf{x})$.

Binomial RFS

A Binomial RFS \mathbf{X} is also a special case of the IID cluster RFS, whose cardinality distribution is a binomial distribution with parameters L (number of binary experiments) and p_0 (the probability of success of each of the experiments):

$$\rho(n) = \binom{L}{n} p_0^n (1 - p_0)^{L-n}$$

Its FISST PDF follows from (2.29) and is given by:

$$f(\mathbf{X}) = \frac{L!}{(L - |\mathbf{X}|)!} p_0^{|\mathbf{X}|} (1 - p_0)^{L-|\mathbf{X}|} \prod_{\mathbf{x} \in \mathbf{X}} p(\mathbf{x}). \quad (2.31)$$

The intensity function is $D(\mathbf{x}) = L p_0 p(\mathbf{x})$.

Multi-Bernoulli RFS

A multi-Bernoulli RFS \mathbf{X} is a union of independent Bernoulli RFSs \mathbf{X}_i , each characterized by probability r_i and the spatial PDF $p_i(\mathbf{x})$, for $i = 1, \dots, M$:

$$\mathbf{X} = \bigcup_{i=1}^M \mathbf{X}_i.$$

The FISST PDF of \mathbf{X} is given by:

$$f(\mathbf{X}) = |\mathbf{X}|! \left[\prod_{j=1}^M (1 - r_j) \right] \sum_{1 \leq i_1 \leq \dots \leq i_{|\mathbf{X}|} \leq M} \prod_{j=1}^{|\mathbf{X}|} \frac{r_{i_j} \cdot p_{i_j}(\mathbf{x}_j)}{1 - r_{i_j}} \quad (2.32)$$

while the intensity function is $D(\mathbf{x}) = \sum_{i=1}^M r_i p_i(\mathbf{x})$.

2.4.2 Multi-Object Stochastic Filtering

The evolution of the multi-target state \mathbf{X}_k is a Markov process, characterized by its initial density $\mathbf{X}_0 \sim f_0(\mathbf{X}_0)$ and the multi-target transitional density $\Pi_{k|k-1}(\mathbf{X}_k|\mathbf{X}_{k-1})$. The multi-target state is not observed directly, but through the observation process. The observation process is assumed to be conditionally independent given the multi-target state process, and fully specified by the multi-target likelihood function $\varphi_k(\mathbf{Z}_k|\mathbf{X}_k)$. Standard cases of $\Pi_{k|k-1}(\mathbf{X}_k|\mathbf{X}_{k-1})$ and $\varphi_k(\mathbf{Z}_k|\mathbf{X}_k)$ have been derived in [14, Chaps. 12,13]. The standard multi-object transitional density is given by:

$$\Pi_{k|k-1}(\mathbf{X}|\mathbf{X}') = f_b(\mathbf{X})(1 - p_s)^{|\mathbf{X}'|} \sum_{\theta} \prod_{i:\theta(i)>0} \frac{p_s \cdot \pi_{k|k-1}(\mathbf{x}_{\theta(i)}|\mathbf{x}'_i)}{(1 - p_s) \cdot \mu_0 b(\mathbf{x}_{\theta(i)})} \quad (2.33)$$

where $\theta : \{1, \dots, |\mathbf{X}'|\} \rightarrow \{0, 1, \dots, |\mathbf{X}|\}$ represents a list of all associations of elements from set \mathbf{X}' to the elements of set \mathbf{X} (here the assumption is that $|\mathbf{X}'| > 0$ and $|\mathbf{X}| \geq 0$); p_s is the probability of survival from $k-1$ to k ; $f_b(\mathbf{X})$ is the object birth FISST PDF, which assuming Poisson birth process, with mean rate μ_0 and distribution $b(\mathbf{x})$, is given by:

$$f_b(\mathbf{X}) = e^{-\mu_0} \prod_{\mathbf{x} \in \mathbf{X}} \mu_0 b(\mathbf{x}).$$

The standard multi-object likelihood $\varphi_k(\mathbf{Z}_k|\mathbf{X}_k)$ is given by:

$$\varphi_k(\mathbf{Z}|\mathbf{X}) = f_c(\mathbf{Z})(1 - p_d)^{|\mathbf{X}|} \sum_{\theta} \prod_{i:\theta(i)>0} \frac{p_d \cdot g_k(\mathbf{z}_{\theta(i)}|\mathbf{x}_i)}{(1 - p_d) \cdot \lambda c(\mathbf{z}_{\theta(i)})} \quad (2.34)$$

where $\theta : \{1, \dots, |\mathbf{X}|\} \rightarrow \{0, 1, \dots, |\mathbf{Z}|\}$ represents a list of all associations of elements from set \mathbf{X} to the elements of set \mathbf{Z} (here the assumption is that if $\mathbf{x}_i \in \mathbf{X}$ is not detected, then $\theta(i) = 0$); p_d is the probability of detection; $f_c(\mathbf{Z})$ is the FISST PDF of clutter (or false detections), which is assumed to be a Poisson RFS with mean rate λ and distribution $c(\mathbf{z})$ over \mathcal{Z} . Hence the clutter FISST PDF is:

$$f_c(\mathbf{Z}) = e^{-\lambda} \prod_{\mathbf{z} \in \mathbf{Z}} \lambda c(\mathbf{z}).$$

The objective of the sequential Bayesian multi-object state estimator [14] is to determine at each time step k the posterior FISST PDF of the multi-object state $f_{k|k}(\mathbf{X}_k|\mathbf{Z}_{1:k})$, where $\mathbf{Z}_{1:k} \equiv \mathbf{Z}_1, \dots, \mathbf{Z}_k$ denotes the accumulated observation set sequence up to time k . The multi-object posterior can be computed sequentially via the prediction and the update steps. Suppose that $f_{k-1|k-1}(\mathbf{X}_{k-1}|\mathbf{Z}_{1:k-1})$ is known and that a new set of measurements \mathbf{Z}_k , corresponding to time t_k , has been received. Then the predicted and updated multi-object posterior densities are calculated as follows [14]:

$$f_{k|k-1}(\mathbf{X}_k | \mathbf{Z}_{1:k-1}) = \int \Pi_{k|k-1}(\mathbf{X}_k | \mathbf{X}_{k-1}) f_{k-1|k-1}(\mathbf{X}_{k-1} | \mathbf{Z}_{1:k-1}) \delta \mathbf{X}_{k-1} \quad (2.35)$$

$$f_{k|k}(\mathbf{X}_k | \mathbf{Z}_{1:k}) = \frac{\varphi_k(\mathbf{Z}_k | \mathbf{X}_k) f_{k|k-1}(\mathbf{X}_k | \mathbf{Z}_{1:k-1})}{\int \varphi_k(\mathbf{Z}_k | \mathbf{X}) f_{k|k-1}(\mathbf{X} | \mathbf{Z}_{1:k-1}) \delta \mathbf{X}}. \quad (2.36)$$

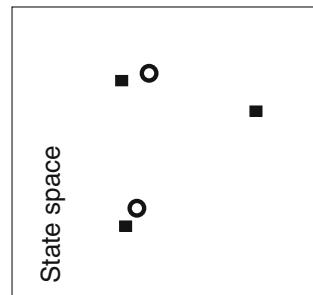
Equations (2.35) and (2.36) denote the prediction and update equations of the exact³ Bayes multi-object filter. Note that the two Eqs. (2.35) and (2.36), are given in the form that is very similar to the form of Eqs. (1.3) and (1.4), respectively. The recursion (2.35–2.36), however, is a non-trivial generalization of (1.3)–(1.4), because the integrals in (2.35–2.36) are set integrals and the expressions for $\Pi_{k|k-1}(\mathbf{X}_k | \mathbf{X}_{k-1})$ and $\varphi_k(\mathbf{Z}_k | \mathbf{X}_k)$ can be quite involved, see for example (2.33) and (2.34). It is no surprise than that the recursions (2.35–2.36) in general have no analytic closed-form solution. The most general implementation of the exact Bayes multi-object filter is based on the sequential Monte Carlo (SMC) method [29], where a set of weighted particles is recursively propagated as an approximation of the multi-object posterior density. Particle filter implementation of the exact Bayes multi-object filter has been considered in [14, 29–32]; more details can be found in Sect. 4.3. Sensor control for multi-object particle filter has been reported in [33].

2.4.3 OSPA Metric

In order to measure the error of a multi-object filter at each time step, we require a distance metric between two finite sets of objects: the set of true objects (ground truth) $\mathbf{X}_k = \{\mathbf{x}_{k,1}, \dots, \mathbf{x}_{k,n_k}\}$ and its estimate $\hat{\mathbf{X}}_k = \{\hat{\mathbf{x}}_{k,1}, \dots, \hat{\mathbf{x}}_{k,\hat{n}_k}\}$.

Example. Consider two sets in Fig. 2.4: the set of squares has three elements; the set of circles has two elements. There are two aspects of the distance metric between the set of squares and the set of circles: (1) the difference in the cardinality of the two sets; (2) the positional or spatial distance.

Fig. 2.4 What is the distance between the set of *squares* and the set of *circles*?



³ Some authors also call it the *full* Bayes multi-object filter.

There are only a few known metrics for measuring the distance between sets. The starting point is the Hausdorff metric, whose performance was found to be unsatisfactory due to its insensitivity to the difference in cardinality. Hoffman and Mahler [34] then proposed a new metric for sets, based on Wasserstein distance [35], which partly fixed the undesirable cardinality behavior of the Hausdorff metric. Schuhmacher et al. [36], however, subsequently demonstrated a number of shortcomings of the Hoffman-Mahler metric and proposed a new metric for sets, referred to as the *optimal subpattern assignment* (OSPA) metric. The OSPA metric is shown to eliminate all of the shortcomings of the Hoffman-Mahler distance and hence will be used extensively as a performance evaluation metric in this monograph.

Let \mathcal{X} be an arbitrary nonempty set. A function $d : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}_+ = [0, \infty)$ is called a metric on \mathcal{X} if it satisfies the following properties: identity, symmetry, and the triangle inequality [37].

Let the distance $d_b(\mathbf{x}, \mathbf{y})$ be referred to as the *base* distance on the single-object state space $\mathcal{X} \subseteq \mathbb{R}^N$. The base distance can be, for example, the Euclidian metric:

$$d_b(\mathbf{x}, \mathbf{y}) = \left(\sum_{\ell=1}^N |\mathbf{x}(\ell) - \mathbf{y}(\ell)|^p \right)^{1/p} \quad (2.37)$$

$\mathbf{x}, \mathbf{y} \in \mathbb{R}^N$, $1 \leq p \leq \infty$. Alternatively, it can be the Mahalanobis distance $d_b = \sqrt{(\mathbf{x} - \mathbf{y})^\top \mathbf{S}^{-1} (\mathbf{x} - \mathbf{y})}$, where \mathbf{S} is the covariance matrix of the difference $\mathbf{x} - \mathbf{y}$.

The cut-off base distance between vectors $\mathbf{x}, \mathbf{y} \in \mathcal{X}$ is defined as:

$$d_c(\mathbf{x}, \mathbf{y}) = \min\{c, d_b(\mathbf{x}, \mathbf{y})\}, \quad (2.38)$$

where $c > 0$ is a parameter referred to as the cut-off value.

Consider now two sets $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_m\}$ and $\mathbf{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_n\}$, where $n, m \in \mathbb{N}_0 = \{0, 1, \dots\}$. Vectors $\mathbf{x} \in \mathbf{X}$ and $\mathbf{y} \in \mathbf{Y}$ are taking values from the state space $\mathcal{X} \subseteq \mathbb{R}^N$, while sets $\mathbf{X}, \mathbf{Y} \in \mathcal{F}(\mathcal{X})$, where $\mathcal{F}(\mathcal{X})$ represents the set of all finite subsets of \mathcal{X} . The OSPA metric is next defined as a distance on $\mathcal{F}(\mathcal{X})$, between sets like \mathbf{X} and \mathbf{Y} .

The OSPA distance of order $1 \leq p \leq \infty$, with the cut-off parameter c , is defined for $m \leq n$ as [36]:

$$d_{p,c}(\mathbf{X}, \mathbf{Y}) = \left[\frac{1}{n} \left(\min_{\pi \in \Pi_n} \sum_{i=1}^m (d_c(\mathbf{x}_i, \mathbf{y}_{\pi(i)}))^p + (n-m) \cdot c^p \right) \right]^{1/p} \quad (2.39)$$

where Π_n represents the set of permutations of length m with elements taken from $\{1, 2, \dots, n\}$. For the case $m > n$, the definition is simply $d_{p,c}(\mathbf{X}, \mathbf{Y}) = d_{p,c}(\mathbf{Y}, \mathbf{X})$. The proof that the OSPA distance is indeed a metric on $\mathcal{F}(\mathcal{X})$ is given in [36].

Note that we can interpret the OSPA distance as a p th order “per-object” error, which comprises two components, each separately accounting for “localization” and “cardinality” errors. If $m \leq n$, these components are given as:

$$e_{p,c}^{\text{loc}}(\mathbf{X}, \mathbf{Y}) = \left[\frac{1}{n} \cdot \min_{\pi \in \Pi_n} \sum_{i=1}^m (d_c(\mathbf{x}_i, \mathbf{y}_{\pi(i)}))^p \right]^{1/p} \quad (2.40)$$

$$e_{p,c}^{\text{card}}(\mathbf{X}, \mathbf{Y}) = \left(\frac{(n-m)c^p}{n} \right)^{1/p}. \quad (2.41)$$

Otherwise (i.e. if $m > n$), we have $e_{p,c}^{\text{loc}}(\mathbf{X}, \mathbf{Y}) = e_{p,c}^{\text{loc}}(\mathbf{Y}, \mathbf{X})$ and $e_{p,c}^{\text{card}}(\mathbf{X}, \mathbf{Y}) = e_{p,c}^{\text{card}}(\mathbf{Y}, \mathbf{X})$. Only for $p = 1$ we can write $d_{p,c}(\mathbf{X}, \mathbf{Y}) = e_{p,c}^{\text{loc}}(\mathbf{X}, \mathbf{Y}) + e_{p,c}^{\text{card}}(\mathbf{X}, \mathbf{Y})$.

Note that the cut-off parameter c determines the relative weighting given to the cardinality error component against the localization error component. Smaller values of c tend to emphasize localization errors and vice versa. Note that c must be in the same units as the localization error.

2.5 Specialized Multi-Object Filters

The exact multi-object stochastic filter (2.35–2.36) is computationally demanding as its dimension of the state space is proportional to the number of objects whose states are to be sequentially estimated. All practical implementations of the multi-object stochastic filter (2.35–2.36) are hence restricted to a relatively small number of objects (maximum 10 was reported in [38]).

In order to overcome this limitation, Mahler [39] proposed several simplifications of the exact multi-object stochastic filter. This section will review only three of them: the Bernoulli filter, the probability density hypothesis (PHD) filter , and the Cardinalized PHD filter. The multi-Bernoulli filter, another approximation which has been proposed recently [14, 40], is not included in this monograph.

2.5.1 Bernoulli Filter

The Bernoulli filter is the exact Bayes filter for nonlinear/non-Gaussian estimation of a single dynamic system which randomly switches *on* and *off*. The binary switching is important: in surveillance applications, dynamic objects of interest *enter* and *leave* the surveillance volume; similarly, dynamic phenomena (e.g. epidemics, pollution), *appear* and eventually *disappear* from the community or environment.

Suppose the state of a dynamic object (phenomenon) at time k can be modeled by the Bernoulli RFS, whose FISST PDF was defined in (2.28). This PDF is uniquely specified by two quantities:

- the posterior probability of object existence $q_{k|k} = \Pr\{|\mathbf{X}_k| = 1 | \mathbf{Z}_{1:k}\}$, and
- the posterior spatial PDF of $\mathbf{X}_k = \{\mathbf{x}_k\}$ denoted $s_{k|k}(\mathbf{x}_k) = p(\mathbf{x}_k | \mathbf{Z}_{1:k})$.

The Bernoulli filter propagates these two quantities over time. Assume that the dynamics of the Bernoulli Markov process $\{\mathbf{X}_k\}_{k \geq 1}$ is characterized by the transitional FISST PDF $\Pi_{k|k-1}(\mathbf{X}|\mathbf{X}')$ specified by:

$$\Pi_{k|k-1}(\mathbf{X}|\emptyset) = \begin{cases} 1 - p_b & \text{if } \mathbf{X} = \emptyset \\ p_b \cdot b_{k|k-1}(\mathbf{x}) & \text{if } \mathbf{X} = \{\mathbf{x}\} \\ 0 & \text{if } |\mathbf{X}| \geq 2 \end{cases}, \quad (2.42)$$

and

$$\Pi_{k|k-1}(\mathbf{X}|\{\mathbf{x}'\}) = \begin{cases} 1 - p_s & \text{if } \mathbf{X} = \emptyset \\ p_s \cdot \pi_{k|k-1}(\mathbf{x}|\mathbf{x}') & \text{if } \mathbf{X} = \{\mathbf{x}\} \\ 0 & \text{if } |\mathbf{X}| \geq 2. \end{cases} \quad (2.43)$$

Here p_b and p_s are the probabilities of object birth and object survival, while $b_{k|k-1}(\mathbf{x})$ is the object birth density. If the posterior at time $k-1$ is specified by the pair $(q_{k-1|k-1}, s_{k-1|k-1}(\mathbf{x}))$, the prediction equation of the Bernoulli filter is then given by [14]:

$$q_{k|k-1} = p_b \cdot (1 - q_{k-1|k-1}) + p_s \cdot q_{k-1|k-1} \quad (2.44)$$

$$s_{k|k-1}(\mathbf{x}) = \frac{p_b(1 - q_{k-1|k-1})b_{k|k-1}(\mathbf{x}) + p_s q_{k-1|k-1} \int \pi_{k|k-1}(\mathbf{x}|\mathbf{x}') s_{k-1|k-1}(\mathbf{x}') d\mathbf{x}'}{p_b(1 - q_{k-1|k-1}) + p_s q_{k-1|k-1}} \quad (2.45)$$

It is straightforward to verify that if $p_s = 1$ and $q_{k-1|k-1} = 1$, Eqs. (2.44–2.45) reduce to (1.3).

Since the cardinality of \mathbf{X}_k equals zero or one, the expression for the likelihood function $\varphi_k(\mathbf{Z}|\mathbf{X})$ in (2.34) simplifies to:

$$\varphi_k(\mathbf{Z}|\mathbf{X}) = \begin{cases} f_c(\mathbf{Z}), & \text{if } \mathbf{X} = \emptyset, \\ f_c(\mathbf{Z}) \left[1 - p_{\text{D}} + p_{\text{D}} \sum_{\mathbf{z} \in \mathbf{Z}} \frac{g_k(\mathbf{z}|\mathbf{x})}{\lambda c(\mathbf{z})} \right], & \text{if } \mathbf{X} = \{\mathbf{x}\}. \end{cases} \quad (2.46)$$

The update equations of the Bernoulli filter can be derived from (2.36) as:

$$q_{k|k} = \frac{1 - \delta_k}{1 - \delta_k \cdot q_{k|k-1}} \cdot q_{k|k-1} \quad (2.47)$$

$$s_{k|k}(\mathbf{x}) = \frac{1 - p_{\text{D}} + p_{\text{D}} \sum_{\mathbf{z} \in \mathbf{Z}_k} \frac{g_k(\mathbf{z}|\mathbf{x})}{\lambda c(\mathbf{z})}}{1 - \delta_k} s_{k|k-1}(\mathbf{x}) \quad (2.48)$$

where

$$\delta_k = p_{\text{D}} \left(1 - \sum_{\mathbf{z} \in \mathbf{Z}_k} \frac{\int g_k(\mathbf{z}|\mathbf{x}) s_{k|k-1}(\mathbf{x}) d\mathbf{x}}{\lambda c(z)} \right). \quad (2.49)$$

If $p_{\text{D}} = 1$ and there are no false detections, then $\mathbf{Z}_k = \{\mathbf{z}\}$ and it is easy to verify that (2.48) reduces to (1.4).

2.5.2 PHD and CPHD Filter

The PHD filter is an approximation of the “full” multi-object stochastic filter (2.35–2.36). Instead of propagating the FISST PDF $f_{k|k}(\mathbf{X}|\mathbf{Z}_{1:k})$, the key idea is to propagate only its first-order statistical moment (PHD) $D_{k|k}(\mathbf{x}|\mathbf{Z}_{1:k})$, defined by (2.27). The PHD has the property that

$$\int_{\mathcal{X}} D_{k|k}(\mathbf{x}|\mathbf{Z}_{1:k}) d\mathbf{x} = v_{k|k}, \quad (2.50)$$

where $v_{k|k} \in \mathbb{R}$ is the expected number of objects in the state space \mathcal{X} . The Poisson RFS is the only RFS which is completely specified by its first-order moment. This means that knowing $D_{k|k}(\mathbf{x}|\mathbf{Z}_{1:k})$, one can easily determine the cardinality distribution as $\rho_{k|k}(n|\mathbf{Z}_{1:k}) = e^{-v_{k|k}} v_{k|k}^n / n!$, as well as the spatial distribution $p_{k|k}(\mathbf{x}|\mathbf{Z}_{1:k}) = D_{k|k}(\mathbf{x}|\mathbf{Z}_{1:k}) / v_{k|k}$.

Using abbreviation $D_{k|k}(\mathbf{x}|\mathbf{Z}_{1:k}) \stackrel{\text{abbr}}{=} D_{k|k}(\mathbf{x})$, the prediction equation of the PHD filter is given by [39]:

$$D_{k|k-1}(\mathbf{x}) = \gamma_{k|k-1}(\mathbf{x}) + p_s \int \pi_{k|k-1}(\mathbf{x}|\mathbf{x}') D_{k-1|k-1}(\mathbf{x}') d\mathbf{x}' \quad (2.51)$$

where $\gamma_{k|k-1}(\mathbf{x})$ is the PHD of the RFS of object births between time $k-1$ and k . Upon receiving the measurement set \mathbf{Z}_k at time k , the update step of the PHD filter is computed according to:

$$D_{k|k}(\mathbf{x}) = \left[1 - p_{\text{D}} + \sum_{\mathbf{z} \in \mathbf{Z}_k} \frac{p_{\text{D}} g_k(\mathbf{z}|\mathbf{x})}{\kappa_k(\mathbf{z}) + p_{\text{D}} \int g_k(\mathbf{z}|\mathbf{x}), D_{k|k-1}(\mathbf{x}) d\mathbf{x}} \right] D_{k|k-1}(\mathbf{x}) \quad (2.52)$$

where $\kappa_k(\mathbf{z})$ is the PHD of the clutter RFS at time k . Usually the clutter is modeled by a Poisson RFS whose PHD (intensity function) is $\kappa_k(\mathbf{z}) = \lambda c(\mathbf{z})$, where λ is the expected number of false detections at time k , while $c(\mathbf{z})$ is the spatial distribution of false detections in the measurement space. It can be easily verified that in the absence of clutter and target birth, with $p_s = p_{\text{D}} = 1$, the PHD filter equations (2.51–2.52) reduce to the standard equations (1.3–1.4).

Although the PHD filter is a very crude approximation of the exact multi-object filter, its elegance, computational simplicity, and ease of implementation have

contributed to its widespread use in applications. Noteworthy applications include the passive radar [41], sonar [42], video tracking [43], simultaneous localization and mapping (SLAM) [44], traffic monitoring [45], cell tracking [46], etc.

The estimate of the number of objects from the PHD filter, $\nu_{k|k}$ in (2.50), is very crude in the sense that it can vary significantly and rapidly over time. This was the motivation to develop another random set-based stochastic multi-object filter, which would involve a more accurate approximation of the FISST PDF $f_{k|k}(\mathbf{X}|\mathbf{Z}_{1:k})$ than just its first moment (i.e. PHD). Mahler introduced the Cardinalized PHD (CPHD) filter in [47], which propagates not only the PHD, but also the cardinality distribution of the multi-target PDF. The cardinality distribution of the CPHD can be arbitrary, not necessarily Poisson, as in the PHD filter. The estimate of the number of targets obtained from the CPHD filter is found to be more stable than that of the PHD filter. The CPHD filter has also attracted some interest and has been applied recently to GMTI tracking [48] and video tracking [49].

Let $\rho_{k|k}(n|\mathbf{Z}_{1:n}) \stackrel{\text{abbr}}{=} \rho_{k|k}(n)$ denote the posterior cardinality distribution at time k . Also, let $\rho_{\Gamma,k|k-1}(n) \stackrel{\text{abbr}}{=} \rho_{\Gamma}(n)$ denote the cardinality distribution of new targets at time k . The CPHD filter propagates both the cardinality distribution $\rho_{k|k}(n)$ and the PHD $D_{k|k}(\mathbf{x})$. The predicted cardinality distribution can be written as a convolution:

$$\rho_{k|k-1}(n) = \sum_{j=0}^n \rho_{s,k|k-1}(j) \rho_{\Gamma}(n-j) \quad (2.53)$$

where

$$\rho_{s,k|k-1}(j) = \sum_{\ell=j}^{\infty} \frac{\ell!}{j! (\ell-j)!} p_s^j (1-p_s)^{\ell-j} \rho_{k-1|k-1}(\ell) \quad (2.54)$$

is the predicted cardinality of survived targets. The predicted PHD is given by (2.51).

The CPHD filter update is given by:

$$\rho_{k|k}(n) = \frac{\Upsilon_k^0 D_{k|k-1}; \mathbf{Z}_k(n) \rho_{k|k-1}(n)}{\langle \Upsilon_k^0 D_{k|k-1}; \mathbf{Z}_k, \rho_{k|k-1} \rangle} \quad (2.55)$$

$$\begin{aligned} D_{k|k}(\mathbf{x}) &= (1 - p_d) \frac{\langle \Upsilon_k^1 D_{k|k-1}; \mathbf{Z}_k, \rho_{k|k-1} \rangle}{\langle \Upsilon_k^0 D_{k|k-1}; \mathbf{Z}_k, \rho_{k|k-1} \rangle} D_{k|k-1}(\mathbf{x}) \\ &\quad + \sum_{\mathbf{z} \in \mathbf{Z}_k} \frac{\langle \Upsilon_k^1 D_{k|k-1}; \mathbf{Z}_k \setminus \{\mathbf{z}\}, \rho_{k|k-1} \rangle}{\langle \Upsilon_k^0 D_{k|k-1}; \mathbf{Z}_k, \rho_{k|k-1} \rangle} \frac{p_d g_k(\mathbf{z}|\mathbf{x})}{c(\mathbf{z})} D_{k|k-1}(\mathbf{x}) \end{aligned} \quad (2.56)$$

where

- $\langle \rho_1, \rho_2 \rangle$ is the inner product between two real-valued sequences $\rho_1(n)$ and $\rho_2(n)$, i.e., $\langle \rho_1, \rho_2 \rangle = \sum_{n=0}^{\infty} \rho_1(n) \rho_2(n)$;

- the sequence $\Upsilon_{k,D,\mathbf{Z}}^u(n)$ is defined for $u \in \{0, 1\}$ as follows:

$$\Upsilon_{k,D,\mathbf{Z}}^u(n) = \sum_{j=0}^{\min(|\mathbf{Z}|, n)} (|\mathbf{Z}| - j)! \rho_{k,k}(|\mathbf{Z}| - j) P_{j+u}^n (1 - p_D) \left[\int D(\mathbf{x}) d\mathbf{x} \right]^{-(j+u)} \times e_j(\Xi_k(D, \mathbf{Z})) \quad (2.57)$$

The term $\rho_{k,k}(n)$ in (2.57) is the cardinality distribution of clutter at time k . The remaining terms are defined as:

$$\Xi_k(D, \mathbf{Z}) = \left\{ \frac{p_D \int D(\mathbf{x}) g_k(\mathbf{z}|\mathbf{x}) d\mathbf{x}}{c(\mathbf{z})} : \mathbf{z} \in \mathbf{Z} \right\} \quad (2.58)$$

$$e_j(\mathbf{Z}) = \sum_{\mathbf{W} \subseteq \mathbf{Z}, |\mathbf{W}|=j} \left(\prod_{\zeta \in \mathbf{W}} \zeta \right) \quad (2.59)$$

$$P_\ell^n = \frac{n!}{(n - \ell)!} \quad (2.60)$$

where $e_j(\mathbf{Z})$ denotes the elementary symmetric function (ESF) of order j for a finite set \mathbf{Z} [14], p. 369.

References

1. A. Doucet, J. F. G. de Freitas, and N. J. Gordon, eds., *Sequential Monte Carlo Methods in Practice*. Springer, 2001.
2. M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for non-linear/non-Gaussian Bayesian tracking”, *IEEE Trans. Signal Processing*, vol. 50, pp. 174–188, Feb. 2002.
3. P. Djuric, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. Bugallo, and J. Miguez, “Particle filtering”, *IEEE Signal Processing Magazine*, pp. 19–38, Sept. 2003.
4. O. Cappé, S. J. Godsill, and E. Moulines, “An overview of existing methods and recent advances in sequential Monte Carlo”, *Proc. IEEE*, vol. 95, no. 5, pp. 899–924, 2007.
5. A. Doucet and A. M. Johansen, “A tutorial on particle filtering and smoothing: Fifteen years later”, tech. rep., Department of Statistics, University of British Columbia, Dec. 2008.
6. B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman filter: Particle filters for tracking applications*. Artech House, 2004.
7. C. P. Robert and G. Casella, *Monte Carlo statistical methods*. Springer, 2nd ed., 2004.
8. M. R. Morelande and B. Ristic, “Radiological source detection and localisation using Bayesian techniques”, *IEEE Trans. Signal Processing*, vol. 57, no. 11, pp. 4220–4231, 2009.
9. N. J. Gordon, D. J. Salmond, and A. F. M. Smith, “Novel approach to nonlinear/non-Gaussian Bayesian state estimation”, *IEE Proc.-F*, vol. 140, no. 2, pp. 107–113, 1993.
10. D. A. Castanón and L. Carin, “Stochastic control theory for sensor management”, in *Foundations and Applications of Sensor Management* (A. O. Hero, D. A. Castanón, D. Cochran, and K. Kastella, eds.), ch. 2, pp. 7–32, Springer, 2008.

11. A. O. Hero, C. M. Kreucher, and D. Blatt, "Information theoretic approaches to sensor management", in *Foundations and applications of sensor management* (A. O. Hero, D. Castanòn, D. Cochran, and K. Kastella, eds.), ch. 3, pp. 33–57, Springer, 2008.
12. C. M. Kreucher, A. O. Hero, K. D. Kastella, and M. R. Morelande, "An information based approach to sensor management in large dynamic networks", *Proc. of the IEEE*, vol. 95, pp. 978–999, May 2007.
13. B. Ristic, B.-N. Vo, and D. Clark, "A note on the reward function for PHD filters with sensor control", *IEEE Trans. Aerospace & Electr. Systems*, vol. 47, no. 2, pp. 1521–1529, 2011.
14. R. Mahler, *Statistical Multisource Multitarget Information Fusion*. Artech House, 2007.
15. T. O'Hogan, "Dicing with the unknown", *Significance*, vol. 1, pp. 132–133, Sep. 2004.
16. A. Gning, B. Ristic, and L. Mihaylova, "Bernoulli particle/box-particle filters for detection and tracking in the presence of triple measurement uncertainty", *IEEE Trans. Signal Processing*, vol. 60, pp. 2138–2151, May 2012.
17. R. Mahler and A. El-Fallah, "The random set approach to nontraditional measurements is rigorously bayesian", in *Proc. SPIE*, vol. 8392 of *Signal Processing, Sensor Fusion and Target Recognition*, Apr. 2012.
18. I. R. Goodman, R. P. S. Mahler, and H. T. Nguyen, *Mathematics of data fusion*. Springer, 1997.
19. P. Smets and P. Magrez, "Implication in fuzzy logic", *Int. Journal Approx. Reasoning*, vol. 1, pp. 327–348, 1987.
20. A. N. Bishop and B. Ristic, "Information fusion with spatially referring natural language statements", *IEEE Trans. Aerospace and Electronic Systems*, 2012. (in print).
21. F. Abdallah, A. Gning, and P. Bonnifait, "Box particle filtering for nonlinear state estimation using interval analysis", *Automatica*, vol. 44, pp. 807–815, 2008.
22. A. Gning, B. Ristic, L. Mihaylova, A. Fahed, "Introduction to box particle filtering", *IEEE Signal Processing Magazine*, 2012. (in print).
23. A. Bishop and B. Ristic, "Fusion of natural language propositions: Bayesian random set framework", in *Proc. 14th Int. Conf. Information Fusion*, (Chicago, USA), July 2011.
24. B. Ristic, "Bayesian estimation with imprecise likelihoods: Random set approach", *IEEE Signal Processing Letters*, vol. 18, pp. 395–398, July 2011.
25. B. Ristic, "Bayesian estimation with imprecise likelihoods in the framework of random set theory", in *Proc. Australian Control Conference*, (Melbourne, Australia), pp. 481–486, Nov. 2011.
26. B. Ristic, "Target classification with imprecise likelihoods: Mahler's approach", *IEEE Transns. Aerospace and Electronic Systems*, vol. 47, no. 2, pp. 1530–1534, 2011.
27. A. Benavoli and B. Ristic, "Classification with imprecise likelihoods: A comparison of tbm, random set and imprecise probability approach", in *Proc. 14th Inter. Conf. Information Fusion*, (Chicago, USA), July 2011.
28. A. Skvortsov and B. Ristic, "Monitoring and prediction of an epidemic outbreak using syndromic observations", *Mathematical biosciences*, vol. 240, pp. 12–19, 2012.
29. B.-N. Vo, S. Singh, and A. Doucet, "Sequential Monte Carlo methods for multi-target filtering with random finite sets", *IEEE Trans. Aerospace & Electronic Systems*, vol. 41, pp. 1224–1245, Oct. 2005.
30. H. Sidenbladh and S. L. Wirkander, "Tracking random sets of vehicles in terrain", in *Proc. 2nd IEEE Workshop on Multi-Object Tracking*, (Madison, WI, USA), June 2003.
31. T. Zajic and R. Mahler, "A particle-systems implementation of the PHD multitarget tracking filter", in *Proc. SPIE*, vol. 5096, pp. 291–299, April 2003.
32. B.-T. Vo and B.-N. Vo, "A random finite set conjugate prior and application to multi-target tracking", in *Proc. IEEE Conf. ISSNIP* 2011, (Adelaide, Australia), pp. 431–436, Dec. 2011.
33. B. Ristic and B.-N. Vo, "Sensor control for multi-object state-space estimation using random finite sets", *Automatica*, vol. 46, pp. 1812–1818, 2010.
34. J. R. Hoffman and R. P. S. Mahler, "Multitarget miss distance via optimal assignment", *IEEE Trans. Systems, Man and Cybernetics - Part A*, vol. 34, pp. 327–336, May 2004.
35. L. Rueshendorff, "Wasserstein (Vasershtein) metric", in *Encyclopaedia of Mathematics* (M. Hazewinkel, ed.), Springer, 2001.

36. D. Schuhmacher, B.-T. Vo, and B.-N. Vo, “A consistent metric for performance evaluation of multi-object filters”, *IEEE Trans. Signal Processing*, vol. 56, pp. 3447–3457, Aug. 2008.
37. E. Kreyszig, *Introductory functional analysis with applications*. Wiley, 1989.
38. J. Czyz, B. Ristic, and B. Macq, “A particle filter for joint detection and tracking of color objects”, *Image and Vision Computing*, vol. 25, pp. 1271–1281, 2007.
39. R. P. S. Mahler, “Multi-target Bayes filtering via first-order multi-target moments”, *IEEE Trans. Aerospace & Electronic Systems*, vol. 39, no. 4, pp. 1152–1178, 2003.
40. B.-T. Vo, B. Vo, and A. Cantoni, “The cardinality balanced multi-target multi-Bernoulli filter and its implementations”, *IEEE Trans. Signal Processing*, vol. 57, no. 2, pp. 409–423, 2009.
41. M. Tobias and A. Lanterman, “Probability hypothesis density-based multitarget tracking with bistatic range and Doppler observations”, *IEE Proc.-Radar Sonar Navig*, vol. 152, no. 3, pp. 195–205, 2005.
42. D. Clark, I. T. Ruiz, Y. Petillot, and J. Bell, “Particle PHD filter multiple target tracking in sonar image”, *IEEE Trans. Aerospace & Electronic Systems*, vol. 43, no. 1, pp. 409–416, 2007.
43. E. Maggio, M. Taj, and A. Cavallaro, “Efficient multitarget visual tracking using random finite sets”, *IEEE Trans. Circuits & Systems for Video Technology*, vol. 18, no. 8, pp. 1016–1027, 2008.
44. J. Mullane, B.-N. Vo, M. Adams, and B.-T. Vo, “A random finite set approach to Bayesian SLAM”, *IEEE Trans. Robotics*, vol. 27, no. 2, pp. 268–282, 2011.
45. G. Battistelli, L. Chisci, S. Morrocchi, F. Papi, A. Benavoli, A. D. Lallo, A. Farina, and A. Graziano, “Traffic intensity estimation via PHD filtering”, in *Proc. 5th European Radar Conf.*, (Amsterdam, The Netherlands), pp. 340–343, Oct. 2008.
46. R. R. Juang, A. Levchenko, P. Burlina, “Tracking cell motion using GM-PHD”, in *Int. Symp. Biomedical, Imaging*, pp. 1154–1157, June/July 2009.
47. R. P. S. Mahler, “PHD filters of higher order in target number”, *IEEE Trans. Aerospace & Electronic Systems*, vol. 43, no. 4, pp. 1523–1543, 2007.
48. M. Ulmke, O. Erdinc, and P. Willett, “GMTI tracking via the Gaussian mixture cardinalized probability hypothesis density filter”, *IEEE Trans. Aerospace & Electronic Systems*, vol. 46, no. 4, pp. 1821–1833, 2010.
49. E. Pollard, A. Plyer, B. Pannetier, F. Champagnat, and G. L. Besnerais, “GM-PHD filters for multi-object tracking in uncalibrated aerial videos”, *Proc. 12th Int. Conf. Information Fusion*, pp. 1171–1178, July 2009.

Chapter 3

Applications Involving Non-standard Measurements

This chapter presents some practical estimation problems involving imprecision, including source localization, monitoring and prediction of an epidemic, localization using spatially referring natural language statements, and classification using the imprecise likelihood models.

3.1 Estimation Using Imprecise Measurement Models

Two applications involving precise measurements \mathbf{z}_k , but imprecise measurement functions, are presented in this section (see Fig. 2.2b for a graphical illustration of this case). The measurement function $\mathbf{h}_k(\mathbf{x}_k; [\theta])$ in both cases is affected by additive zero-mean Gaussian noise \mathbf{w}_k , with the generalized likelihood function given by (2.18).

3.1.1 Localization Using the Received Signal Strength

Received signal strength (RSS) is often used for localization of energy emitting sources. Since the emitting power of the source is in general unknown, it has to be included in the state vector. Let us consider a two-dimensional Cartesian coordinate system, so that the state vector of the emitting source that needs to be localized, is specified as $\mathbf{x} = [x_0 \ y_0 \ A_0]^\top$, where $(x_0, y_0) \in \mathbb{R}^2$ is the source position and $A_0 \in \mathbb{R}^+$ is the received power (in dB) at a short reference distance d_0 from the source. A measurement z_m of the RSS (in dB) at location (x_m, y_m) can be modeled as [1]:

$$z_m = A_0 - 10 \cdot \phi_m \cdot \log \frac{d_m}{d_0} + v_m \quad (3.1)$$

for $m = 1, \dots, M$, where

$$d_m = \sqrt{(x_m - x_0)^2 + (y_m - y_0)^2},$$

is the distance between the source and sensor m ; ϕ_m is the propagation loss from the source to sensor m , and v_m is zero-mean white Gaussian noise with a known standard deviation $\sigma = 2\text{dB}$. The propagation loss ϕ_m in free space is 2. In reality, however, due to multipath and shadowing [1], ϕ_m is only partially known: experimentally it has been found that it takes values from an interval, i.e., $\phi_m \in [2, 4]$. The location of sensor $m = 1, \dots, M$ also is only partially known in the sense that $x_m \in [\hat{x}_m - \varepsilon_x, \hat{x}_m + \varepsilon_x]$ and $y_m \in [\hat{y}_m - \varepsilon_y, \hat{y}_m + \varepsilon_y]$. Here \hat{x}_m and \hat{y}_m are the nominal sensor coordinates in x and y , with ε_x and ε_y being their confidence bounds. The parameter vector of the imprecise measurement function of sensor m is therefore three-dimensional: $\theta_m = [\phi_m \ x_m \ y_m]^\top$. Recall that the imprecise measurement function defines in this case a one-dimensional interval which, using interval arithmetic, can be expressed as:

$$h_m(\mathbf{x}_k; [\theta_m]) = A_0 - 10 \cdot [\phi_m] \log \frac{1}{d_0} \sqrt{([x_m] - x_0)^2 + ([y_m] - y_0)^2}. \quad (3.2)$$

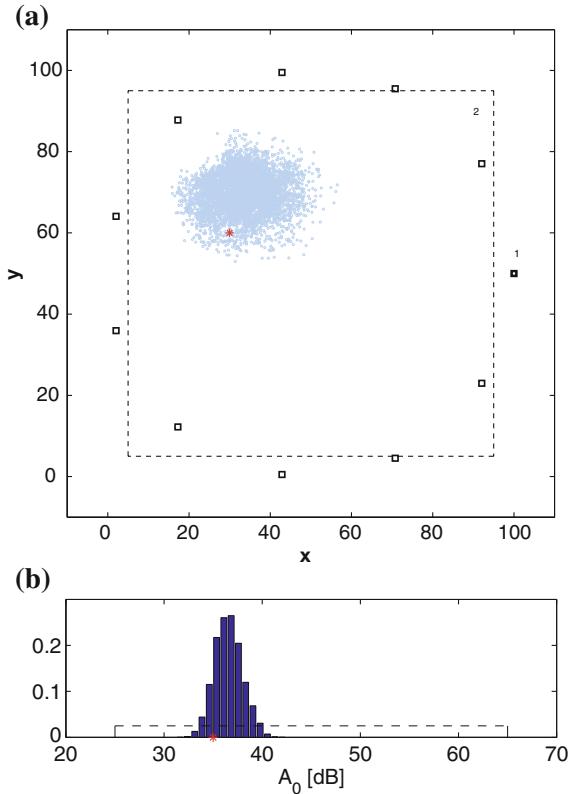
Its limits, according to (2.18), determine the generalized likelihood function of sensor m , i.e., $\tilde{g}(z_m|\mathbf{x})$.

3.1.1.1 Numerical Results

The scenario we consider is shown in Fig. 3.1a: there are $M = 12$ sensors placed (nominally) on a circle centered at $(50, 50)$, with a radius 50 (sensors 1 and 2 are labeled; anticlockwise labeling continues for the remaining sensors). The confidence bounds of sensor locations are $\varepsilon_x = \varepsilon_y = 0.2$. The unknown actual sensor locations are set to $x_m = \hat{x}_m - \varepsilon_x/2$ and $y_m = \hat{y}_m - \varepsilon_y/2$, for $m = 1, \dots, M$. The true source parameters (to be estimated) are $x_0 = 30$, $y_0 = 60$, and $A_0 = 35$, with a (known) reference distance $d_0 = 10$. The partially known propagation loss factors are set to $\phi_m = 2.3$ for sensors $m = 1, 2, \dots, 6$ and $\phi_m = 3.5$ for sensors $m = 7, 8, \dots, 12$. The prior PDF used in estimation is $p_0(\mathbf{x}) = \mathcal{U}_{[5, 95]}(x_0) \cdot \mathcal{U}_{[5, 95]}(y_0) \cdot \mathcal{U}_{[25, 65]}(A_0)$, where $\mathcal{U}_{[a,b]}(c)$ denotes the uniform density over c with the support $[a, b]$.

In the described setup, the source is static and therefore localization is based on the update equation of the Bayes stochastic filter, given by (1.4). Suppose that a single snapshot of the measurement vector $\mathbf{z} = [z_1 \ z_2 \ \dots \ z_M]^\top$ is available for estimation of the state vector \mathbf{x} (i.e., a single measurement from each sensor). The Bayes estimator (1.4) was implemented using a variant of the algorithm in Algorithm 1. A random sample is drawn from the initial (prior) PDF: $\mathbf{x}^{(i)} \sim p_0(\mathbf{x})$ for $i = 1 \dots, N_0$, as in line 4 of Algorithm 1. An importance weight $w^{(i)} \propto \prod_{m=1}^M \tilde{g}(z_m|\mathbf{x}^{(i)})$ is associated with each sample $\mathbf{x}^{(i)}$, where $\tilde{g}(z_m|\mathbf{x}^{(i)})$ is the generalized likelihood given by (2.18). This is done in accordance with line 5 of Algorithm 1, followed by normalization of weights in line 7 of Algorithm 1. The next step consists of resampling (lines 8–11 in

Fig. 3.1 The output of the Bayes estimator using the generalized likelihood function $\tilde{g}(\mathbf{z}|\mathbf{x}) = \prod_{m=1}^M \tilde{g}(z_m|\mathbf{x})$, where $M = 12$ and $\tilde{g}(z_m|\mathbf{x})$ is defined by (2.18) and (3.2). **a** The *cloud of dots* represents the random sample approximation of the posterior PDF $p(x_0, y_0|\mathbf{z})$; the true values are $x_0 = 30$ and $y_0 = 60$. **b** The histogram of random samples approximating the posterior PDF $p(A_0|\mathbf{z})$; the true value is $A_0 = 35$

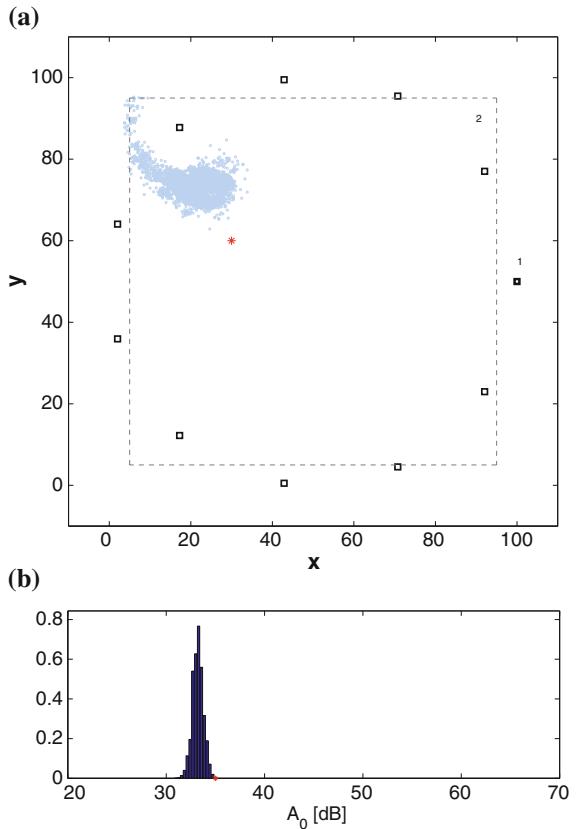


Algorithm 1) $N = N_0/10$ times from the weighted sample $\{w^{(i)}, \mathbf{x}^{(i)}\}_{i=1}^{N_0}$ to obtain N equally weighted samples $\{w^{(i)} = \frac{1}{N}, \mathbf{x}^{(i)*}\}_{i=1}^N$ which approximate the posterior PDF $p(\mathbf{x}|\mathbf{z})$. The last step is the MCMC move for sample diversification [6, p. 55]. The result is shown in Fig. 3.1a in (x_0, y_0) space and Fig. 3.1b in A_0 space, using $N = 5000$ samples.

Few observations can be made from this simple example. First, the Bayes estimator has reduced the uncertainty because the support of the posterior PDF is much smaller than the support of the initial PDF (indicated by dashed lines in Fig. 3.1). Second, the true parameter \mathbf{x} is included in the support of the posterior PDF. Third, the point Bayes estimates, such as the mean or the maximum of the posterior PDF, in general can be biased. Finally, note that an alternative method which carries out marginalization over nuisance parameter in this example would involve integration over a $3 \times M$ dimensional vector $\boldsymbol{\theta}$, i.e., $p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}) = \int p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}) p_0(\boldsymbol{\theta}) d\boldsymbol{\theta}$, where $\boldsymbol{\theta} = [\boldsymbol{\theta}_1^\top \cdots \boldsymbol{\theta}_M^\top]^\top$ and $p_0(\boldsymbol{\theta})$ is the prior PDF on the parameter space. In the random set-based approach, however, there was no need for integration since the expression for the generalized likelihood function has been worked out analytically.

Figure 3.2 illustrates what happens if we force a precise likelihood model in the setup described above. The posterior PDF shown in Fig. 3.2 was obtained by

Fig. 3.2 The output of the standard Bayes estimator using the precise likelihood functions $g_{\theta_m}(z_m|\mathbf{x}) = p_w(\mathbf{z}_m - h_m(\mathbf{x}; \boldsymbol{\theta}_m))$, with $\phi_m = 2.5$, $x_m = \hat{x}_m$, and $y_m = \hat{y}_m$ for $m = 1, \dots, 12$. **a** The *cloud of dots* represents the random sample approximation of the posterior PDF $p(x_0, y_0|\mathbf{z})$; the true values are $x_0 = 30$ and $y_0 = 60$. **b** The histogram of random samples approximating the posterior PDF $p(A_0|\mathbf{z})$; the true value is $A_0 = 35$



arbitrarily setting the values of: (1) propagation loss factors to $\phi_m = 2.5$; (2) sensor locations to nominal values, i.e., $x_m = \hat{x}_m$, $y_m = \hat{y}_m$, for $m = 1, \dots, 12$. This is followed by the standard Bayes estimator using the (precise) likelihood function $g_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) = \prod_{m=1}^M g_{\theta_m}(z_m|\mathbf{x})$, where $g_{\theta_m}(z_m|\mathbf{x}) = p_w(\mathbf{z}_m - h_m(\mathbf{x}; \boldsymbol{\theta}_m))$. While the resulting posterior PDF appears to be more concentrated (hence, less uncertain), it does not contain the true value of \mathbf{x} . This outcome is considered as a catastrophic failure of the estimation algorithm.

3.1.1.2 Inclusion Criterion

The Bayes optimality of the random-set-based estimator (using the generalized likelihood function) guarantees that the resulting posterior is the most concentrated PDF whose support contains the true value of \mathbf{x} . Next we verify this by simulations in the context of localization using the RSS measurements. For the purpose of verification, we need to introduce an assessment criteria referred as *inclusion*, which verifies that the support of the posterior includes the true value of \mathbf{x} .

In order to define inclusion, let us introduce a *credible set* [2] $\mathbf{C}_k(\alpha)$ associated with the posterior $p(\mathbf{x}|\mathbf{z})$. This set is defined implicitly as the smallest set $\mathbf{C}_k(\alpha) \subseteq \mathcal{X}$ such that the probability:

$$\Pr\{\mathbf{C}_k(\alpha)\} = \int_{\mathbf{C}_k(\alpha)} p(\mathbf{x}|\mathbf{z}) d\mathbf{x} = 1 - \alpha, \quad (3.3)$$

where typically $\alpha \ll 1$. A credible set at $\alpha \rightarrow 0$ represents the support of the posterior spatial PDF $s_{k|k}(\mathbf{x})$. The *inclusion criterion* ρ_k is defined as:

$$\rho_k = \begin{cases} 1, & \text{if the true state } \mathbf{x}_k \in \mathbf{C}_k(\alpha) \\ 0, & \text{otherwise.} \end{cases} \quad (3.4)$$

Our goal now is to establish the percentage of time the inclusion criterion is satisfied for the random-set-based estimator in the context of localization using the RSS measurements, described above. The simulation setup differs from the one described above only in one respect: the propagation loss factors and sensor locations are now random in each Monte Carlo run and for each sensor, that is:

$$\begin{aligned} \phi_m &\sim \mathcal{U}_{[2,4]}(\phi), \\ x_m &\sim \mathcal{U}_{[\hat{x}_m - \varepsilon_x, \hat{x}_m + \varepsilon_x]}(x) \quad (m = 1, \dots, M) \\ y_m &\sim \mathcal{U}_{[\hat{y}_m - \varepsilon_y, \hat{y}_m + \varepsilon_y]}(y). \end{aligned} \quad (3.5)$$

The kernel density estimation (KDE) method [3] is used to establish if the inclusion criterion is satisfied in each run. Assuming the posterior PDF is represented by equally weighted samples $\{w^{(i)} = \frac{1}{N}, \mathbf{x}^{(i)}\}_{i=1}^N$, the KDE approximation of the posterior PDF is [3]:

$$p(\mathbf{x}|\mathbf{z}) \approx \widehat{p}(\mathbf{x}) = \frac{1}{N \cdot W^{n_x}} \sum_{i=1}^N \varphi\left(\frac{\mathbf{x} - \mathbf{x}^{(i)}}{W}\right) \quad (3.6)$$

where $\varphi(\mathbf{x})$ is the kernel and W is the kernel width parameter. For convenience $\varphi(\mathbf{x})$ is adopted to be a Gaussian PDF with zero-mean and covariance matrix \mathbf{P} . The optimal fixed kernel width (under the assumption that the underlying PDF is Gaussian) for the Gaussian kernel $\varphi(\mathbf{x})$ is [3]: $W^* = A \cdot N^{-\frac{1}{n_x+4}}$, where $A = [4/(n_x + 2)]^{\frac{1}{n_x+4}}$. In this application $n_x = 3$. The covariance \mathbf{P} is estimated as a sample covariance. Using the KDE approximation (3.6) it would be possible to estimate the boundary of the credible set $C(\alpha)$. The computation involved, however, would be prohibitively expensive, and we adopt a simpler approximation of the inclusion criterion as follows:

$$\rho \approx \begin{cases} 1, & \text{if } \widehat{p}(\mathbf{x}) \geq \min_{i=1,\dots,N} \widehat{p}(\mathbf{x}^{(i)}) \\ 0, & \text{otherwise} \end{cases} \quad (3.7)$$

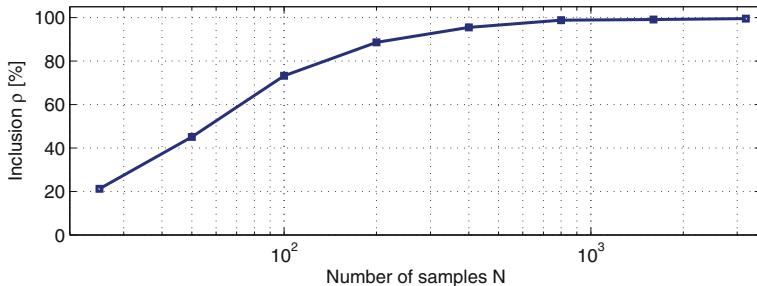


Fig. 3.3 The percentage of 1,000 Monte Carlo runs in which the inclusion was satisfied ($\rho = 1$) as a function of the number of samples N

where \mathbf{x} is the true value of the parameter vector and $\widehat{p}(\mathbf{x})$ was defined in (3.6). The value of $\min_{i=1,\dots,N} \widehat{p}(\mathbf{x}^{(i)})$ in (3.7) effectively approximates the boundary of the credible set $C(0)$.

Figure 3.3 shows the percentage of 1,000 Monte Carlo runs in which the inclusion was satisfied (i.e., $\rho = 1$), as a function of the number of samples N . Observe that as N is increased, and therefore the Monte Carlo approximation of the posterior PDF is more accurate, the inclusion percentage grows. For a sufficiently good approximation, i.e., for N above a certain threshold, the inclusion is satisfied 100 %.

3.1.2 Prediction of an Epidemic Using Syndromic Data

Epidemics is a dynamic phenomenon whose temporal progress in a large population can be fairly accurately modeled by so-called *compartmental models* [4]. The SIR compartmental model is a popular class of compartmental models, which divides the population, in relation to an infectious disease, into three categories: susceptible (S), infectious (I), and recovered or removed (R). Susceptible individuals have never been in contact with the disease. When infected, an individual from the compartment S moves to the compartment I. Infected individuals eventually either recover or die and in this way move into the compartment R. Assuming that the population size is constant and that population mixing is homogeneous (each susceptible person is equally likely to become the next victim), the dynamics of the compartments can be simply expressed by the Kermack-McKendrick SIR model [5]. Looking at the number of infected people over time in a large population with no prior exposure to the disease, one can observe three main phases. First is the establishment phase, where the number in compartment I grows very slowly. Second is the exponential-like growth of the compartment I. Finally in the endemicity phase, when the significant proportion of population is immune or dead, the incidence of new infections tends to decline. When the susceptible population is exhausted, the primary epidemic is over.

The SIR epidemic model is fully specified by two parameters. In order to relax the assumption of the SIR model that population mixing is homogeneous, a modification of the model was proposed in [6, 7], where the third parameter is included to account for mixing inhomogeneity. This new model, referred to as the mSIR, will be adopted here, due to its remarkably accurate agreement with real data and agent-based simulations [6, 8].

3.1.2.1 Dynamic Model

Let the number of susceptible, infectious, and recovered be denoted by S , I , and R , respectively, so that $S + I + R = P$, where $P = \text{const}$ is the total population size. The dynamic model of an epidemic progression in time can be expressed by two stochastic differential equations [5] and the “conservation” law for the population, as follows:

$$\frac{ds}{dt} = -q + \sigma_q \xi, \quad (3.8)$$

$$\frac{di}{dt} = q - \gamma i - \sigma_q \xi + \sigma_\gamma \zeta, \quad (3.9)$$

$$r = 1 - s - i. \quad (3.10)$$

The explanation of notation is as follows: $s = S/P$, $i = I/P$, $r = R/P$ are the normalized compartment sizes; ξ , ζ are two uncorrelated white Gaussian noise processes, both with zero mean and unity variance; γ is the recovery time, that is a reciprocal of the average infectious period for the disease; the term $q(i, s)$ describes the social interactions between the groups of population.

Following [6, 7] we adopt $q(i, s) = \beta i s^\nu$, where $\nu \geq 0$ is the ‘mixing’ intensity. The case $\nu = 1$ corresponds to the traditional homogeneous (or well-mixed) approximation, while for $\nu = 0$ we obtain a linear system without any social interactions (total self-isolation). The *contact rate* parameter β is a product $\beta = \rho \cdot \gamma$, with ρ being the *basic reproductive number*, which represents the average number of people infected by a direct contact with a sick person that would be expected if the population was entirely susceptible.

In general, the three parameters of the mSIR epidemic model, β , γ , and ν , can be assumed to be partially known as interval values. For a known disease, the prior knowledge of parameters is precise and the intervals are very tight. For an emerging disease, it is the opposite. The terms $\sigma_q \equiv \sigma_q(s, i)$ and $\sigma_\gamma \equiv \sigma_\gamma(s, i)$ in (3.8) and (3.9) are introduced to capture the demographic stochasticity, due to random variations in the contact rate β and in the recovery time γ between individuals in the community.

The mSIR model (3.8–3.10) is given in continuous time. For the purpose of computer implementation, we require its discrete-time approximation. The state vector, adopted as

$$\mathbf{x} = [i \ s \ \beta \ \gamma \ \nu]^T \quad (3.11)$$

includes the unknown disease parameters, β and γ , as well as the unknown population mixing parameter ν . Using the Euler approximation (valid for small integration interval) the epidemic dynamics in discrete-time can then be expressed as in (1.1):

$$\mathbf{x}_{k+1} \approx \mathbf{f}_k(\mathbf{x}_k) + \mathbf{v}_k \quad (3.12)$$

where

$$\mathbf{f}_k(\mathbf{x}_k) = \begin{bmatrix} \mathbf{x}_k[1] + \tau \mathbf{x}_k[1] (\mathbf{x}_k[3] \mathbf{x}_k[2]^{\mathbf{x}_k[5]} - \mathbf{x}_k[4]) \\ \mathbf{x}_k[2] - \tau \mathbf{x}_k[3] \mathbf{x}_k[1] \mathbf{x}_k[2]^{\mathbf{x}_k[5]} \\ \mathbf{x}_k[3] \\ \mathbf{x}_k[4] \\ \mathbf{x}_k[5] \end{bmatrix} \quad (3.13)$$

In this notation $\mathbf{x}_k[j]$ represents the j th component of vector \mathbf{x}_k . Process noise \mathbf{v}_k in (3.12) is assumed to be zero-mean white Gaussian with a diagonal covariance matrix \mathbf{Q} of size 5×5 . Its first two diagonal components can be expressed as [9]: $\mathbf{Q}(1, 1) \approx (\beta + \gamma)\tau^2/P^2$ and $\mathbf{Q}(2, 2) \approx \beta\tau^2/P^2$. The remaining components are set to zero based on the assumption that β, γ, ν are constant during the epidemic.

3.1.2.2 Measurement Model

Syndromic surveillance is referred to as systematic collection, analysis, and interpretation of public health data for the purpose of early (and often cost-effective) detection of an epidemic outbreak, its continuous monitoring, and timely response by public health agencies [10, 11]. The rationale for this method rests on a reasonable assumption that the spread of an infectious disease is usually associated with the measurable changes in social behavior.

We will assume here that measurements available for estimation of the state of an epidemics are non-medical syndromic data, such as the absenteeism from work/school, the pharmaceutical sales, Internet queries, Twitter messages, and alike. Recent studies [12–15] have demonstrated that this type of syndromic data contains enough information to draw important conclusions regarding the state of an epidemic in the community. The flu-related searches in Google [16] are a remarkable example of this approach. Following [16] we can assume that a power law relationship holds for odds-ratios of syndromic observations and the number of infected people:

$$\frac{z_m}{1 - z_m} \propto \left(\frac{i}{1 - i} \right)^{\varsigma_m}, \quad (3.14)$$

where z_m is the observable syndrome (e.g., a keyword in a Google search) with index $m = 1, \dots, M$ (normalized by population size P), ς_m is the power-law exponent (in general different for different syndromes). Since at the initial stage of epidemic $i \ll 1$, $z_m \ll 1$, Eq. (3.14) can be reduced to a simple power-law model

$$z_m = h_m(\mathbf{x}; b_m, \varsigma_m) + \tau_m, \quad (3.15)$$

where $h_m(\mathbf{x}; b_m, \varsigma_m) = b_m i^{\varsigma_m}$, and $b_m = \text{const} \geq 0$. Measurement noise term τ_m is added to simulate random nature of measurements, which is assumed to be uncorrelated to other syndromes and noises ξ and ζ . Since $z_m \geq 0$, the noise term τ_m associated with syndrome m should be modeled with a random variable that provides strictly non-negative realizations (e.g., Poisson, truncated Gaussian). We adopt a random variable that obeys the log-normal distribution, meaning that $\tau_m = \sigma_m \eta_m$, where η_m is the standard log-normal noise, $\eta_m \sim \log \mathcal{N}(0, 1)$.

Parameters of the observation model (3.15), $b_m, \sigma_m, \varsigma_m$, are estimated during the calibration phase and therefore are not included in the state vector. Parameter ς_j , however, is difficult to estimate accurately. For example, the linear regression fit reported in [12] suggests that ς_m is close to unity, but it is not precisely known because of the significant scattering of data points. Therefore ς_m will be assumed to be a point in an interval which includes 1, that is $\varsigma_m \in [\underline{\varsigma}_m, \bar{\varsigma}_m]$. Since [12, 14, 16], do not report any specific values of the fitting parameters, in our numerical studies we will synthesize the syndromic measurements using heuristic values for b_j, σ_j . The noise parameter σ_j is adopted as the scaling law $\sigma_j \propto (1/P)^{1/2}$.

Let all syndromic observations of (3.15), collected at time k , be denoted $z_{k,m}$, $m = 1, \dots, M$. They are all stacked into a measurement vector \mathbf{z}_k . The problem is to estimate the (normalized) number of infected i and susceptible s , using the syndromic observation vector sequence $\mathbf{z}_{1:k_0}$. Based on the estimate of the posterior $p(\mathbf{x}_{k_0} | \mathbf{z}_{1:k_0})$, the aim is then to predict the progress of the epidemic to some future time $k > k_0$, using the dynamic model (3.12).

The described problem can be solved using the particle filter (such as the one described in Algorithm 1), where the likelihood function $g_k(\mathbf{z}_k | \mathbf{x}_k)$, which features in line 5 of Algorithm 1, is replaced by the product of generalized likelihood functions:

$$\prod_{m=1}^M \tilde{g}_{k,m}(z_{k,m} | \mathbf{x}_k). \text{ Each GLF } \tilde{g}_{k,m}(z_{k,m} | \mathbf{x}_k) \text{ is of the form (2.18), with limits } \underline{h}_m = \inf\{h_m(\mathbf{x}_k; b_m, [\underline{\varsigma}_m, \bar{\varsigma}_m])\} \text{ and } \bar{h}_m = \sup\{h_m(\mathbf{x}_k; b_m, [\underline{\varsigma}_m, \bar{\varsigma}_m])\}.$$

3.1.2.3 Numerical Results

The numerical analysis is carried out using an experimental data set obtained from a large-scale agent-based simulation system called CROWD, developed by DSTO [17].

The following scenario was simulated [17]. A small Australian virtual town, population size $P = 5000$, was built according to the Census data from the Australian Bureau of Statistics. This virtual town has a typical age/gender breakdown with the population grouped in families and households. The census data were used to create typical daily activities within the town (workplaces, schools, etc). The inhabitants move around the town according to their daily habits and thus interact. A source of epidemic is injected in this virtual environment, with population interactions driving the spread of the epidemic in a probabilistic manner [17]. In line with the prediction of the mSIR model, the disease spreads quickly through the susceptible population. However, as more people become infected, the availability of susceptible people

Fig. 3.4 The *solid line*, representing the number of infected people over time, was obtained from the agent-based simulation population model [17]. The *dashed line* shows the modified SIR model fit

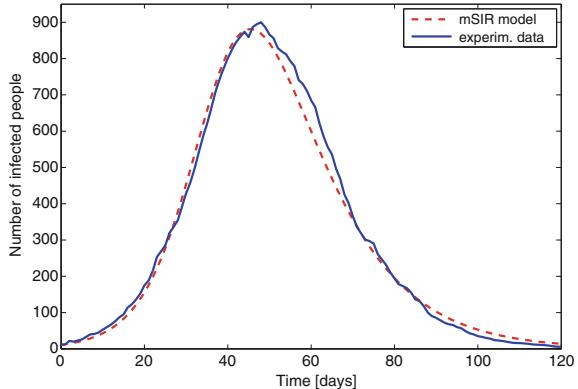


Table 3.1 Syndromic observation model parameters

m	b_m	ς_m	σ_m
1	0.25	1.07	0.0012
2	0.27	1.05	0.0008
3	0.23	1.01	0.001
4	0.29	0.98	0.0011

drops, making it less likely for those infected to pass the disease on. Thus the epidemic eventually dies out.

An example of the output of the agent-based simulations is presented in Fig. 3.4. The solid line shows the number of people in the town that are infected by a fictitious disease, reported once per day during a period of 154 days (only first 120 days shown). The dashed line represents the mSIR model fit (using the entire batch of 154 data points, and setting $v_k = 0$ in (3.12)), with estimated parameters $\hat{\beta} = 0.2399$, $\hat{\gamma} = 0.1066$, $\hat{v} = 1.2042$. The parameter estimates were obtained using importance sampling with progressive correction [8, Chap.14]. Figure 3.4 serves to verify that the mSIR model, although very simple and fast to run, is remarkably accurate in explaining the data obtained from a very complex agent-based simulation.

The syndromic-like measurements are generated synthetically in accordance with the model (3.15), where the “true” (normalized) number of infected people i is replaced by the output of the agent-based simulation at each time step (shown by the solid line in Fig. 3.4). The true parameters of the measurement model (3.15) are given in Table 3.1. The assumption is that the conditionally independent measurements of $M = 4$ syndromes are available on a daily basis during the epidemic.

The initial PDF for the state vector was chosen as

$$p_0(\mathbf{x}_0) = p_0(i_0)p_0(s_0)p_0(\beta_0)p_0(\gamma_0)p_0(v_0)$$

with $p_0(i_0) = \mathcal{N}_{[0,1]}(i_0; z_{1,1}/b_1, \sigma_1^2)$, $p_0(s_0) = \mathcal{N}_{[0,1]}(s_0; 1 - z_{1,1}/b_1, \sigma_1^2)$, $p_0(\beta_0) = \mathcal{U}_{[0.14, 0.5]}(\beta_0)$, $p_0(\gamma_0) = \mathcal{U}_{[0.09, 0.143]}(\gamma_0)$, $p_0(v_0) = \mathcal{U}_{[0.95, 1.3]}(v_0)$, where $\mathcal{N}_{[a,b]}$

(x, μ, P) denotes a truncated Gaussian distribution restricted at support $[a, b]$. The number of particles was set to $N = 10000$.

Numerical results are shown next. The algorithm assumed the correct knowledge of parameters b_m and σ_m in the measurement model, but only partial knowledge of ς_m , that is $\varsigma_m \in [0.85, 1.15]$, for all $m = 1, 2, 3, 4$. The prediction was carried out after day $k_0 = 25$.

Figure 3.5a shows the histograms of the particle filter estimated values of β , γ , and ν , after processing the first 25 days of measurements (i.e., in total 100 measurements, since $M = 4$). This figure reveals that the uncertainty in parameter β has been substantially reduced compared with the initial PDF $p_0(\beta_0) = \mathcal{U}_{[0.14, 0.50]}(\beta_0)$. The uncertainty in parameter γ has been also reduced, although not as dramatically. Finally, the uncertainty in ν , has not been reduced at all, indicating that this parameter is difficult to estimate in the early stages of an epidemic (ν can be estimated only when the measurements are available for the entire duration of the epidemic [8]). While this is unfortunate, it does not appear to be a serious problem since the prior on ν in practice is tight ($\nu \approx 1$). This is confirmed in Fig. 3.5b which shows a sample of 100 overlayed predicted epidemic curves (gray lines) based on the state estimate after 25 days. Figure 3.5b indicates the prediction performance: the timing of the peak of the epidemic appears to be fairly accurate, while the size of the peak is more uncertain.

It is noteworthy that even when our knowledge of the measurement model parameter ς is imprecise, the curve corresponding to what we consider as the “true” number of infected people (the solid line in Fig. 3.5) is always enveloped by predictions. More details about this application can be found in [9].

3.1.3 Summary

Imprecision of the measurement function can be dealt with in the Bayesian estimation framework using random set approach. The consequence of imprecision is higher uncertainty which results in a broader posterior density. If a sufficient amount of measurements over time is available for calibration, in some situations it may be possible to refine our knowledge of the imprecisely known parameter vector θ . This will be discussed in Sect. 7.2.

3.2 Fusion of Spatially Referring Natural Language Statements

This section, based on [18, 19], considers the problem of localization (positioning) of a certain object in a given space, via automatic fusion of spatially referring natural language (NL) statements. The NL statements are inherently imprecise and ambiguous as they depend greatly on the context and grounding of the referenced subjects [20, 21].

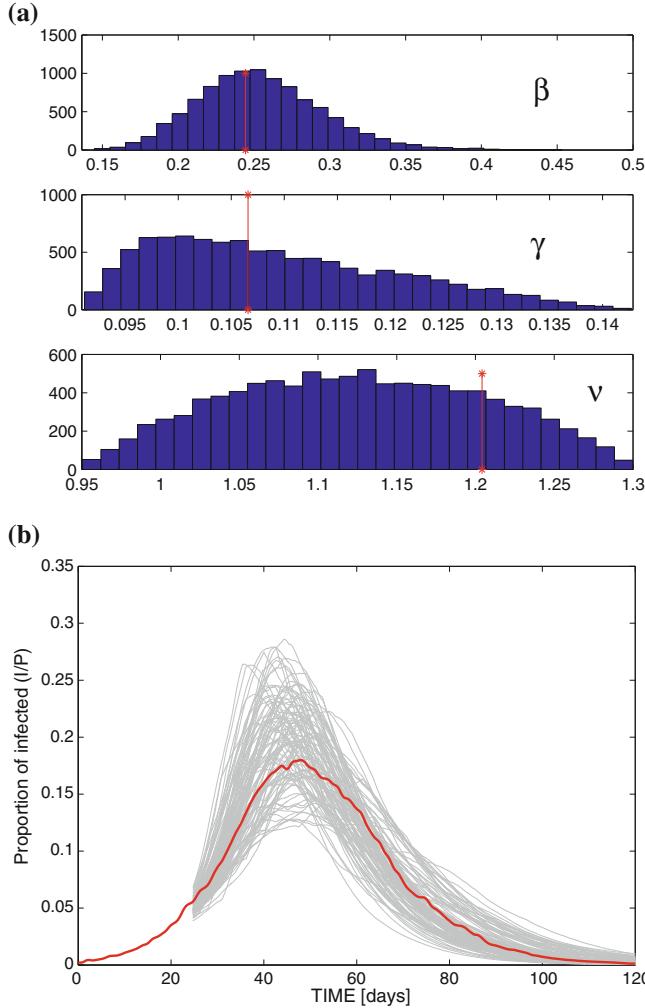


Fig. 3.5 Estimation/prediction results from the particle filter after processing measurements collected over 25 days of surveillance, assuming that measurement function parameters are imprecise, i.e., $\varsigma_j \in [0.85, 1.15]$ (the true values of ς_j given in Table 3.1). **a** The histograms of estimated parameters β , γ , and ν (the “true values” are $\hat{\beta} = 0.2399$, $\hat{\gamma} = 0.1066$, $\hat{\nu} = 1.2042$). **b** Prediction results for a random sample of 100 particles (gray lines); the solid black line is the experimental curve from Fig. 3.4

3.2.1 Language, Space, and Modeling

An element of the state space \mathcal{X} represents a complete specification of the system or subject of interest. A system or subject of interest is denoted by \mathfrak{S} , whereas the state of \mathfrak{S} , is denoted by $\mathbf{x} \in \mathcal{X}$. For example, in a radar scenario, we might have $\mathfrak{S} = \{\text{target}\}$ and $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^3$.

Let $\{\phi_m\}_{m=1}^M$ denote a set of spoken propositions or statements, concerning the value of the state of a subject. The normal form of a proposition is given by

$$\phi^* = \text{the } \textit{subject} \text{ is located with some spatial relationship to an anchor in the space} \quad (3.16)$$

Any ϕ which preserves exactly the semantics of ϕ^* is acceptable. Associated with each proposition ϕ is a mapping $\alpha : \phi \rightarrow [0, 1]$ which represents a probabilistic confidence of the proposition. If $\alpha = 0$ then ϕ can typically be neglected.

We consider a set of predefined *spatial relationships* denoted by $\{\mathcal{R}_j\}_{j=1}^r$ and a set of *anchors* $\{\mathcal{A}_k\}_{k=1}^a$ with known positions. An example of a specially referring proposition is:

$$\phi_m = \text{the train is near the bridge} \quad (3.17)$$

where $\mathcal{S} = \{\text{train}\}$, the spatial relationship is $\mathcal{R} = \{\text{near}\}$ and $\mathcal{A} = \{\text{bridge}\}$ is the anchor. A list of typical spatial relationships includes words such as *near*, *in front of*, *behind*, *inside*, *outside*, *close to*, etc.

The spatially referring propositions (3.16) are inherently imprecise and often ambiguous. Imprecision is due to the nature of spatial relationships (*near*, *close to*, *in between*), which refer to subsets (crisp or fuzzy) of the state space. Ambiguity could be the consequence of non-uniqueness of anchors. For example, in the proposition (3.17), the ambiguity could be caused by the fact that the set of anchors $\{\mathcal{A}_k\}_{k=1}^a$ contains more than one bridge. Alternatively, ambiguity could result from non-uniqueness of a spatial relationship itself. For example, a proposition “the target is *in front of* the red car” has two interpretations. The first is that the target is on the line between the speaker and the red car, but close to the car; the second is that the car has an orientation and its front is facing the target.

An automatic fusion of propositions ϕ_m , $m = 1, 2, \dots$ can be carried out in the Bayesian framework, as described in Sect. 2.3. Since a proposition of the form (3.16) is in general both imprecise and ambiguous, its likelihood can be modeled by a weighted sum of generalized likelihoods as follows:

$$\tilde{g}(\phi_m | \mathbf{x}) = (1 - \alpha_m) + \sum_{j \in \mathcal{H}_m} w_j \eta_j(\mathbf{x}; \phi_m) \quad (3.18)$$

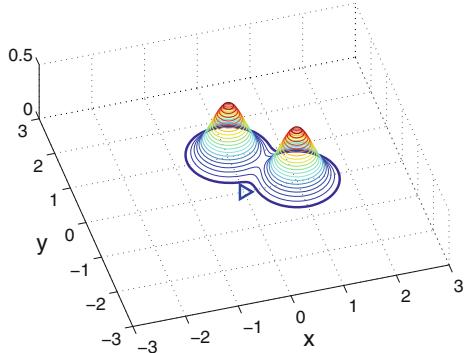
where α_m is the confidence assigned to proposition ϕ_m , $\sum_j w_j = \alpha_m$, and \mathcal{H}_m is a list of ambiguous hypotheses (a function of \mathcal{R}_j and \mathcal{A}_k). Function $\eta_j : \mathbf{x} \rightarrow [0, 1]$ for $\mathbf{x} \in \mathcal{X}$ is a membership function (function of belonging) which defines a fuzzy set on \mathcal{X} .

Example 3.1 We have seen that the proposition

$$\phi = \text{the target is in front of the red car} \quad (3.19)$$

is both imprecise and ambiguous. Suppose that the confidence in this proposition is $\alpha = 1$. Let $\mathcal{X} = \mathbb{R}^2$ and suppose the location of the car has a position (x, y) in \mathbb{R}^2

Fig. 3.6 An example of the GLF for the statement $\phi = \text{the target is in front of the red car}$. The speaker is located at $(0, 2.5)$ and the car is located at $(0, 0)$ facing in the positive x -direction



and an orientation ϑ . For this example, take $[x \ y \ \vartheta]^\top = [0 \ 0 \ 0]^\top$, such that the car is facing toward the positive x -axis. The speaker is at location $(0, 2.5)$. The generalized likelihood function of ϕ can be represented by a sum of two Gaussian membership functions, i.e.,

$$\tilde{g}(\phi|\mathbf{x}) = \frac{1}{2}\eta_1(\mathbf{x}; \phi) + \frac{1}{2}\eta_2(\mathbf{x}; \phi) \quad (3.20)$$

where

$$\eta_j(\mathbf{x}; \phi) = \exp\left\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_j)^\top \boldsymbol{\Omega}_j^{-1} (\mathbf{x} - \boldsymbol{\mu}_j)\right\}, \quad j = 1, 2. \quad (3.21)$$

The means $\boldsymbol{\mu}_j$ and covariances $\boldsymbol{\Omega}_j$ are the tuning parameters based on ϕ . An example of the GLF (3.20) is shown in Fig. 3.6 (the exact numbers defining the variance and mean are not important).

3.2.2 An Illustrative Example

The problem is to localize a static target using spatially referring spoken propositions. The Bayes estimator in this example is conveniently implemented as a particle filter. The scenario (the state space, anchors) is depicted in Fig. 3.7a.

Prior knowledge, assigned confidence $\alpha = 1$, is that the target is in the field. If we interpret this statement to mean the target is within the field proper, i.e., not in the building, garage, tower or the stable, then the prior PDF $p_0(\mathbf{x})$ can be approximated using particles as shown in Fig. 3.7b.

Domain knowledge is expressed in the form of an uncertain implication rule (see Sect. 2.3). Suppose the rule is:

$$R : \text{if the sun is shining then the target is near the stable or the garage} \quad (3.22)$$

and it is assigned confidence $\alpha = 0.7$. The observation space which triggers the implication rule is *weather*, and the proposition can be received from a sensor or

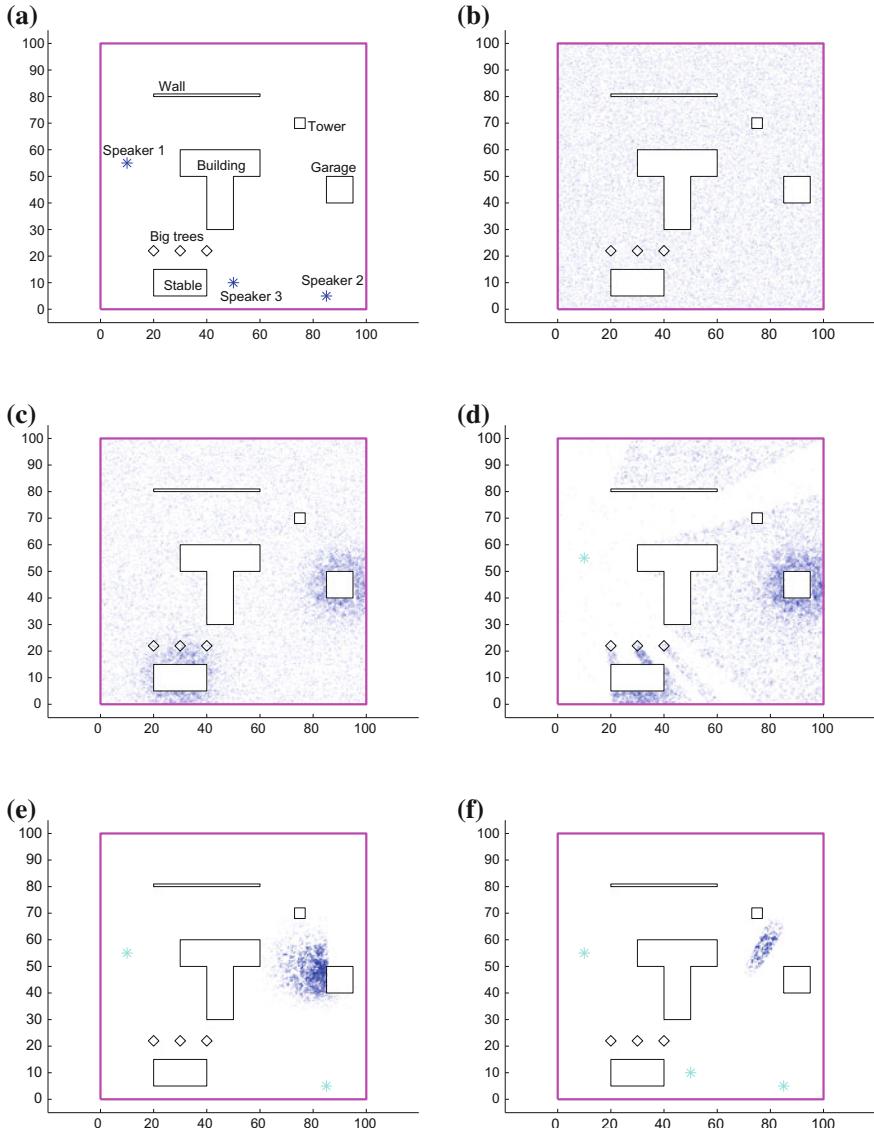


Fig. 3.7 Localization using spatially referring NL statements. **a** The field of interest in which three speakers and the target are located. The position of the anchors in the field, e.g., the building, tower, etc., are known. **b** Prior PDF $p_0(\mathbf{x})$, approximated by particles. **c** The posterior $p(\mathbf{x}|R)$ after the trigger of the uncertain implication rule R . **d-f** The particle approximation of the posterior PDF $p(\mathbf{x}|\phi_{1:m}, R)$ for $m = 1, 2, 3$

spoken by one of the observers as a simple statement *sun is shining* or *sun is not shining*. Suppose that the antecedent is true (i.e., the sun is shining). Then the prior PDF is updated with the GLF of the form in (2.20) and the resulting posterior is shown in Fig. 3.7c. Note how the posterior PDF around the stable and the garage has peaked, although some particles still cover the rest of the field because the confidence in the rule is only 0.7.

There are three speakers, denoted 1, 2, and 3, located at (10, 55), (85, 5) and (50, 10), respectively. Suppose the following propositions are received in a sequence:

$$\text{Speaker 1 : } \phi_1 = \text{the target is outside my visibility polygon} \quad (3.23)$$

$$\text{Speaker 2 : } \phi_2 = \text{the target is in front of the tower} \quad (3.24)$$

$$\text{Speaker 3 : } \phi_3 = \text{the target is at 1 o'clock} \quad (3.25)$$

with $\alpha_1 = \alpha_2 = \alpha_3 = 1$. A visibility polygon of Speaker 1 is a well-defined geometric structure which in this case is a star-shaped polygon and can be computed in linear time. The updated posterior $p(\mathbf{x}|\phi_1, R)$ appears as in Fig. 3.7d. Note that the particles in Fig. 3.7d are now evacuated from the visibility polygon of speaker 1, as expected. The updated posterior $p(\mathbf{x}|\phi_{1:2}, R)$ appears as in Fig. 3.7e. Note that there are no more particles concentrated near the stable, since Speaker 2 essentially negates this hypothesis. The GLF corresponding to statement (3.25) is a two-dimensional cone-based uniform distribution with a $\pm 3^\circ$ spread centered at *1 o'clock* (i.e., 30° clockwise from north) with an apex at Speaker 3. The updated posterior $p(\mathbf{x}|\phi_{1:3}, R)$ appears as in Fig. 3.7f. Note from this final posterior the relative accuracy we have achieved in locating the target given nothing but rather vague individual statements about its possible location.

Fusion of the three statements with prior knowledge and the implication rule, via the Bayes rule, has refined our knowledge about the position of the target. The key, as in all applications of this chapter, is the design of the generalized likelihoods from imprecise and possibly ambiguous propositions and rules.

3.3 Classification Using Imprecise Likelihoods

The problem is model-based classification (that is, estimation on a discrete state space) when likelihood models are specified only partially. Let us illustrate this problem by an example.

Example 3.2 Suppose there are three classes of aircraft, that is the state space is $\mathcal{X} = \{1, 2, 3\}$. Class 1 are commercial planes, class 2 are large military aircraft, and class 3 are light and agile military aircraft. The available target feature (i.e., observation, measurement) for classification is the maximum observed acceleration during an interval of time. This feature is related to the classes of aircraft as follows:

- targets from class 1 almost never fly with acceleration higher than $1g$ ($g = 9.81 \text{ m/s}^2$ is the acceleration due to gravity);
- targets from class 2 sometimes perform mild maneuvers, but their acceleration, due to their size, cannot be higher than $4g$;
- for targets from class 3 the acceleration can go up to $7g$;

In order to minimize the fuel consumption and stress for pilots, majority of the time all three classes of aircraft fly with nearly constant velocity. Thus the steady state for acceleration is approximately zero.

In this example, the expert (domain) knowledge is partially specified and as a result the likelihood functions for the three classes of aircraft can be modeled only imprecisely. The standard Bayes classifier can work only with precise likelihoods, thereby ignoring the uncertainty in mapping the state space to the feature space. The consequence is, as reported in [22], that the standard Bayes classifier behaves in a manner that is inconsistent with our intuition.

As a more appropriate alternative to the standard Bayes classifier, [22] proposed a classification method based on the transferrable belief model (TBM) [23], a variant of the Dempster-Shafer theory [24]. The main idea of this method was to treat the precise likelihoods, which are used by the standard Bayes classifier, as subjective, rather than true models. The precise likelihoods are therefore converted to the least-committed belief functions, followed by the application of the generalized Bayes transform [25] for classification. The final step involves the mapping from the belief domain into the probabilistic domain. In [22] this was done via the pignistic transform [26]. The numerical results in [22] verified that the TBM classifier indeed behaves in accordance with our intuition.

Mahler [27] argued that the Bayesian classifier combined with random set representation of imprecise likelihoods can produce identical results to those obtained using the TBM. This section verifies Mahler's approach using the described target classification example as a benchmark problem.

3.3.1 Modeling

In the target classification example we distinguish two spaces: $\mathcal{X} \subseteq \mathbb{N}$ is a discrete space of (target) classes, and \mathcal{L} is the target feature space (maximum acceleration in our case). We will consider two cases of the feature space:

- the continuous feature space, i.e., $\mathcal{L} = \mathbb{R}$
- the finite discrete feature space, $\mathcal{L} = \{z_1, z_2, z_3\}$, where z_1 corresponds to small acceleration, i.e., $z_1 = \{z : |z| \leq 1g\}$, $z_2 = \{z : 1g < |z| \leq 4g\}$ represents moderate acceleration and $z_3 = \{z : 4g < |z| \leq 7g\}$ high acceleration;

The inference is carried out on \mathcal{X} , which is hidden, being directly unobservable. The features are the measurements available for making inference; they are related to the hidden space of target classes via the likelihood function.

The likelihood of measurement $z \in \mathcal{Z}$ is referred to as imprecise because there is no unique mapping from \mathcal{X} to \mathcal{Z} . Take for example the discrete feature space $\mathcal{Z} = \{z_1, z_2, z_3\}$: class 2 maps into a set $\{z_1, z_2\}$, while class 3 maps into a set $\{z_1, z_2, z_3\}$. This is similar to the situation depicted in Fig. 2.2b.

Since there is no unique mapping from \mathcal{X} to \mathcal{Z} , it is impossible to write down z as a function $h(x) + w$. However, the correspondence between \mathcal{X} and \mathcal{Z} , in this case can be specified by an instance of a random set, referred to as the *basic belief assignment* (BBA), defined originally in the Dempster-Shafer theory.

Let Σ_x denote a random set representation of the feature on a finite discrete feature space $\mathcal{Z} = \{z_1, z_2, z_3\}$. Then Σ_x can be expressed as:

$$\Sigma_x = \{(m_j, A_j); j = 1, \dots, J\} \quad (3.26)$$

where

- $A_j \in 2^{\mathcal{Z}}$ and $2^{\mathcal{Z}}$ is the power set of \mathcal{Z}
- $m_j : 2^{\mathcal{Z}} \rightarrow [0, 1]$
- $\sum_{j=1}^J m_j = 1$.

Here $m_j(A_j) = Pr\{z \in A_j | x\}$ represents a BBA, defined as a measure of belief that is committed exactly to A [28]. A possible specification of Σ_x for each class in the example above is:

$$\Sigma_{x_1} = \{(1, \{z_1\})\} \quad (3.27)$$

$$\Sigma_{x_2} = \{(0.6, \{z_1\}), (0.4, \{z_1, z_2\})\} \quad (3.28)$$

$$\Sigma_{x_3} = \{(0.5, \{z_1\}), (0.3, \{z_1, z_2\}), (0.2, \{z_1, z_2, z_3\})\}. \quad (3.29)$$

The first BBA in (3.27) is *categorical*, with a singleton focal element; it describes a precise likelihood. The other two BBAs, however, are multi-focal and describe imprecise likelihoods.

For the continuous measurement space, a random set Σ_x can be specified by a fuzzy BBA as follows:

$$\Sigma_x = \{(m_j, \eta_j); j = 1, \dots, J\} \quad (3.30)$$

where

- $\eta_j : \mathcal{Z} \rightarrow [0, 1]$ is a membership function (function of belonging) which defines a fuzzy subset of \mathcal{Z} ;
- $m_j : \mathfrak{U}(\mathcal{Z}) \rightarrow [0, 1]$, where $\mathfrak{U}(\mathcal{Z})$ is a set of all fuzzy subsets of \mathcal{Z} ;
- $\sum_{j=1}^J m_j = 1$.

One possibility for the specification of Σ_x in accordance with the problem description is:

$$\Sigma_{x_i} = \{(w_i, \eta_i)\} \quad (3.31)$$

for $i = 1, 2, 3$, where

$$w_i = 1, \quad \eta_i(z) = \exp \left\{ -\frac{z^2}{2\sigma_i^2} \right\}, \quad (3.32)$$

with $\sigma_1 = 0.4$ g, $\sigma_2 = 1.6$ g, and $\sigma_3 = 2.8$ g. These values of standard deviations were adopted to ensure that $Pr\{|z| < \gamma\} = 0.99876$, for $\gamma = 1$ g, 4 g, and 7 g, representing the limits for class 1, 2, and 3, respectively. All three fuzzy BBAs of (3.31) have a single focal element. Other choices are possible too. For example, a fuzzy version of the BBA in (3.27–3.29) can be specified as:

$$\Sigma_{x_1} = \{(1, \eta_1)\}, \quad (3.33)$$

$$\Sigma_{x_2} = \{(0.6, \eta_1), (0.4, \eta_2)\} \quad (3.34)$$

$$\Sigma_{x_3} = \{(0.5, \eta_1), (0.3, \eta_2), (0.2, \eta_3)\} \quad (3.35)$$

where η_i were defined in (3.32).

3.3.2 Classification Results

In the random set framework, the Bayes classifier for imprecise likelihoods is given by:

$$Pr\{x|z\} = \frac{\tilde{g}(z|x) \cdot Pr\{x\}}{\sum_{x \in \mathcal{X}} \tilde{g}(z|x) \cdot Pr\{x\}} \quad (3.36)$$

where $Pr\{x\}$ is the prior class probability and $\tilde{g}(z|x)$ is the generalized likelihood function, defined as in (2.17), that is $\tilde{g}(z|x) = Pr\{z \in \Sigma_x\}$. This definition of the GLF provides an important link between the Dempster-Shafer reasoning and Bayes inference in the framework of random set theory: the GLF is identical to the plausibility function on singleton, a function which plays an important role in the Dempster-Shafer theory.

For the discrete feature space using (3.26) we have:

$$\begin{aligned} \tilde{g}(z|x) &= Pr\{z \in \Sigma_x\} = \sum_{j=1}^J Pr\{z \in A_j | x\} \\ &= \sum_{j=1}^J m_j \cdot \mathbb{I}_{A_j}(z) \end{aligned} \quad (3.37)$$

where $\mathbb{I}_A(z)$ is the indicator function, i.e., $\mathbb{I}_A(z) = 1$ if $z \in A$ and zero otherwise.

For the continuous feature space using (3.30) we have:

$$\tilde{g}(z|x) = Pr\{z \in \Sigma_x\} = \sum_{j=1}^J m_j \cdot \eta_j(z). \quad (3.38)$$

Next we present the numerical results.

Discrete feature space

The set of classes is $\mathcal{X} = \{1, 2, 3\}$, the feature space is $\mathcal{Z} = \{z_1, z_2, z_3\}$. The GLF $\tilde{g}(z_i|x = j)$, for $i, j = 1, 2, 3$, was specified by RFSs (3.27–3.29). Using (3.37) it follows that:

$$\tilde{g}(z_1|x = 1) = \tilde{g}(z_1|x = 2) = \tilde{g}(z_1|x = 3) = 1, \quad (3.39)$$

$$\tilde{g}(z_2|x = 1) = 0, \quad \tilde{g}(z_2|x = 2) = 0.4, \quad \tilde{g}(z_2|x = 3) = 0.5, \quad (3.40)$$

$$\tilde{g}(z_3|x = 1) = \tilde{g}(z_3|x = 2) = 0, \quad \tilde{g}(z_3|x = 3) = 0.2. \quad (3.41)$$

Assuming uniform priors on classes, i.e., $Pr\{x = 1\} = Pr\{x = 2\} = Pr\{x = 3\} = 1/3$, it follows from (3.36) that:

$$Pr\{x = 1|z_1\} = Pr\{x = 2|z_1\} = Pr\{x = 3|z_1\} = 1/3. \quad (3.42)$$

The conclusion is that the feature measurement z_1 (small acceleration) does not alter the priors (it is uninformative). This is perfectly in accordance with our intuition because all three classes of aircraft fly (most of the time) with small acceleration.

If the feature measurement is z_2 (medium acceleration), then:

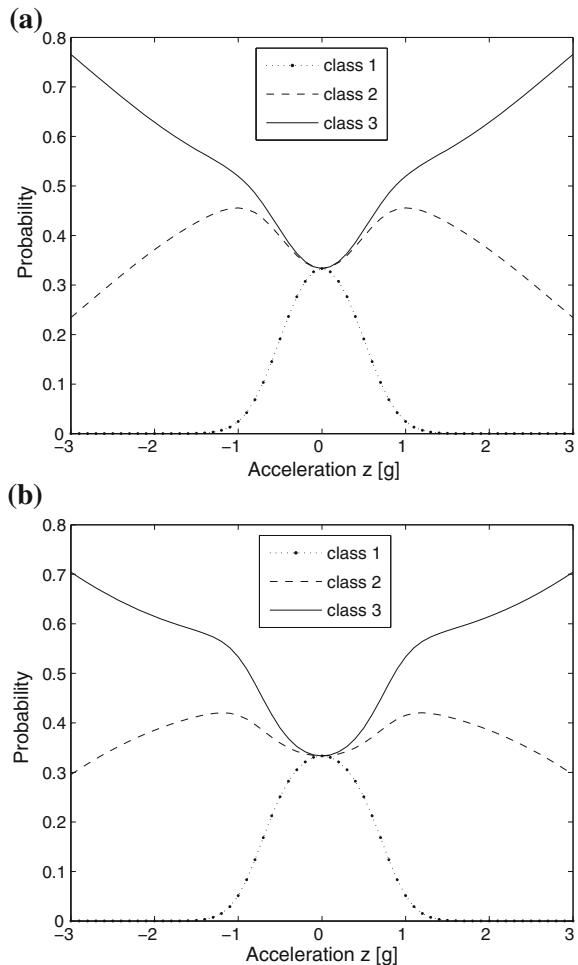
$$Pr\{x = 1|z_2\} = 0, \quad Pr\{x = 2|z_2\} = 4/9, \quad Pr\{x = 3|z_2\} = 5/9, \quad (3.43)$$

meaning that in this case class 1 can be eliminated, while classes 2 and 3 are characterized by similar probabilities. Finally, measurement z_3 results in classification probabilities:

$$Pr\{x = 1|z_3\} = Pr\{x = 2|z_3\} = 0, \quad Pr\{x = 3|z_3\} = 1. \quad (3.44)$$

Again, this is in accordance with our intuition. Note however that the TBM approach [22], which uses the pignistic transform to map belief functions to probabilities, results in slightly different numerical values. As Mahler [Sect. 5.3.4.2] pointed out, only using the Voorbraak approximation [29], instead of the pignistic transform, one can obtain identical numerical values.

Fig. 3.8 Posterior class probabilities $Pr\{x = j|z\}$, $j = 1, 2, 3$ for Σ_{x_j} defined by **a** Eqs. (3.31–3.32), **b** Eqs. (3.33–3.35)



Continuous feature space

Again $\mathcal{X} = \{1, 2, 3\}$, but $\mathcal{Z} = \mathbb{R}$. Let us first work out the case where Σ_{x_j} , $j = 1, 2, 3$ is specified by (3.31–3.32). According to (3.38), in this case the GLF is simply:

$$g(z|x = j) = \eta_j(z) \quad (3.45)$$

for $z \in \mathcal{Z}$ and $j = 1, 2, 3$. Application of (3.36) results in the classification probabilities depicted in Fig. 3.8a. This figure is almost identical to Figure 6 of [22], which was obtained using the TBM machinery on the continuous feature space. For very small values of acceleration z , all three classes are equally probable. It was argued in [22] that this is again perfectly in accordance with our intuition (as opposed to the classification results obtained by the standard Bayes filter).

Finally, for the case of imprecise likelihoods where the feature is specified by random set Σ_{x_j} , $j = 1, 2, 3$ in Eqs. (3.33–3.35), Fig. 3.8b depicts the classification probabilities. The result is very similar to that of Fig. 3.8a and the same arguments apply.

In summary, this section demonstrated that Bayesian inference, combined with random set representation of imprecise relationships between the classes and features, can produce identical results to those obtained using the TBM approach. The significance of this is remarkable. Not only that the random set Bayesian approach is much simpler (both theoretically and computationally), but also it is applicable to the continuous state space \mathcal{X} (as demonstrated for example in Sect. 3.2). The TBM and the Dempster-Shafer theory are at present restricted to the discrete state space.

References

1. N. Patwari, J. N. Ash, S. Kyperountas, R. L. M. A. O. Hero III, and N. S. Correal, “Locating the nodes”, *IEEE Signal Processing Magazine*, pp. 54–68, July 2005.
2. J. O. Berger, *Statistical Decision Theory and Bayesian Analysis*. New York: Springer Series in Statistics, 1985.
3. B. W. Silverman, *Density estimation for statistical and data analysis*. Chapman and Hall, 1986.
4. F. Brauer, P. V. D. Driessche, and J. Wu, eds., *Mathematical Epidemiology*. Lecture Notes in Mathematics, Springer, 2008.
5. R. M. Anderson and R. M. May, “Population biology of infectious diseases: Part 1”, *Nature*, vol. 280, pp. 361–367, 1979.
6. P. D. Stroud, S. J. Sydoriak, J. M. Riese, J. P. Smith, S. M. Mniszewski, and P. R. Romero, “Semi-empirical power-law scaling of new infection rate to model epidemic dynamics with inhomogeneous mixing”, *Mathematical Biosciences*, vol. 203, pp. 301–318, 2006.
7. A. S. Novozhilov, “On the spread of epidemics in a closed heterogeneous population”, *Mathematical Biosciences*, vol. 215, no. 2, p. 177185, 2008.
8. B. Ristic, A. Skvortsov, and M. Morelande, “Predicting the progress and the peak of an epidemic”, in *Proc. IEEE Intern. Conf. Acoustic Speech and Signal Proc. (ICASSP)*, (Taipei, Taiwan), pp. 513–516, 2009.
9. A. Skvortsov and B. Ristic, “Monitoring and prediction of an epidemic outbreak using syndromic observations”, *Mathematical biosciences*, vol. 240, pp. 12–19, 2012.
10. M. Wagner, A. Moore, and R. Aryel, *Handbook of Biosurveillance*. Elsevier, 2006.
11. A. Wilson, G. Wilson, and D. Olwell, *Statistical Methods in Counterterrorism: Game Theory, Modeling, Syndromic Surveillance, and Biometric Authentication*. Springer, 2006.
12. C. Chew and G. Eysenbach, “Pandemics in the age of Twitter: content analysis of tweets during the 2009 h1n1 outbreak”, *PLoS One*, vol. 5, no. 11, p. e14118, 2010.
13. N. M. Schuster, M. Rogers, and L. Jr, “Using search engine query data to track pharmaceutical utilization: A study of statin”, *The American Journal of Managed Care*, vol. 16, no. 8, pp. e215–e219, 2010.
14. A. Signorini, A. M. Segre, and P. M. Polgreen, “The use of Twitter to track levels of disease activity and public concern in the u.s. during the influenza a h1n1 pandemic”, *PLoS One*, vol. 6, no. 5, p. e19467, 2011.
15. A. Culotta, “Detecting influenza outbreaks by analyzing Twitter messages”, in *Proc. 2010 Conf. on Knowledge Discovery and Data Mining*, (Washington, D.C.), 2010.
16. J. Ginsberg, M. Mohebbi, R. S. Patel, L. Brammer, M. S. Smolinski, and L. Brilliant, “Detecting influenza epidemics using search engine query data”, *Nature*, vol. 457, pp. 1012–1015, 2009.

17. R. Connell, P. Dawson, and A. Skvortsov, “Comparison of an agent-based model of disease propagation with the generalised SIR epidemic model”, tech. rep., DSTO, 2009.
18. A. Bishop and B. Ristic, “Fusion of natural language propositions: Bayesian random set framework”, in *Proc. 14th Int. Conf. Information Fusion, (Chicago, USA), July 2011*.
19. A. N. Bishop and B. Ristic, “Information fusion with spatially referring natural language statements”, *IEEE Trans. Aerospace and Electronic Systems*, 2012. (in print).
20. W. Hayward and M. Tarr, “Spatial language and spatial representation”, *Cognition*, vol. 55, no. 1, pp. 39–84, 1995.
21. J. Bateman, J. Hois, R. Ross, and T. Tenbrink, “A linguistic ontology of space for natural language processing”, *Artificial Intelligence*, 2010.
22. B. Ristic and P. Smets, “Target classification approach based on the belief function theory”, *IEEE Trans. Aerospace and Electronic Systems*, vol. 41, pp. 574–583, April 2005.
23. P. Smets, “Data fusion in the transferable belief model”, in *Proc. 3rd Intern. Conf. Information Fusion (FUSION 2000), (Paris, France)*, pp. PS21-33, 2000.
24. R. Yager and L. Liu, eds., *Classic works of the DempsterShafer theory of belief functions. Studies in fuzziness and soft computing*, Springer, 2008.
25. P. Smets, “Belief functions: the disjunctive rule of combination and the generalized Bayesian theorem”, *International Journal of Approximate Reasoning*, vol. 9, pp. 1–35, 1993.
26. P. Smets, “Decision making in the TBM: The necessity of the pignistic transformation”, *International Journal of Approximate Reasoning*, vol. 38, pp. 133–147, 2005.
27. R. Mahler, *Statistical Multisource Multitarget Information Fusion*. Artech House, 2007.
28. G. Shafer, *A mathematical theory of evidence*. Princeton University Press, 1976.
29. F. Voorbraak, “A computationally efficient approximation of Dempster-Shafer theory”, *Int. J. Man-Machine Studies*, vol. 30, pp. 525–536, 1989.

Chapter 4

Multi-Object Particle Filters

This chapter reviews the particle filter implementations of the Bernoulli filter, the PHD/CPHD filter and the exact multi-object filter.

4.1 Bernoulli Particle Filters

The standard particle filter implementation of the Bernoulli filter is reviewed first, mainly building on the material presented in Sects. 2.1 and 2.5.1. Subsequently, a box-particle filter implementation is presented.

4.1.1 Standard Bernoulli Particle Filters

The Bernoulli filter was described in Sect. 2.5.1. In the general nonlinear/non-Gaussian case, the Bernoulli filter cannot be solved analytically and is therefore implemented approximately. The particle filter implementation of the Bernoulli filter has been discussed in [1–3]. The resulting Bernoulli-particle filter approximates the spatial PDF $s_{k|k}(\mathbf{x})$ by a particle system $\{w_k^{(i)}, \mathbf{x}_k^{(i)}\}_{i=1}^N$, where $\mathbf{x}_k^{(i)}$ is the state of particle i and $w_k^{(i)}$ is its weight. Since $s_{k|k}(\mathbf{x})$ is a conventional PDF, the weights are normalized, i.e. $\sum_{i=1}^N w_k^{(i)} = 1$.

Suppose at time $k-1$, the probability of existence is $q_{k-1|k-1}$ and the spatial PDF is approximated as

$$\widehat{s}_{k-1|k-1}(\mathbf{x}) = \sum_{i=1}^N w_{k-1}^{(i)} \delta_{\mathbf{x}_{k-1}^{(i)}}(\mathbf{x}). \quad (4.1)$$

The computation of the predicted probability of existence $q_{k|k-1}$ is straightforward, see (2.44). According to (2.45), the prediction step for the spatial PDF involves the sum of two terms. Hence the particle approximation of $s_{k|k-1}(\mathbf{x})$ can be written as

$$\widehat{s}_{k|k-1}(\mathbf{x}) = \sum_{i=1}^{N+J} w_{k|k-1}^{(i)} \delta_{\mathbf{x}_{k|k-1}^{(i)}}(\mathbf{x}) \quad (4.2)$$

where the particles are drawn from two importance (proposal) densities, ρ_k for *persisting* particles and β_k for *birth* particles [2]:

$$\mathbf{x}_{k|k-1}^{(i)} \sim \begin{cases} \rho_k(\mathbf{x}|\mathbf{x}_{k-1}^{(i)}, \mathbf{Z}_k) & i = 1, \dots, N \\ \beta_k(\mathbf{x}|\mathbf{Z}_k) & i = N+1, \dots, N+B \end{cases} \quad (4.3)$$

with weights

$$w_{k|k-1}^{(i)} = \begin{cases} \frac{p_s q_{k-1|k-1}}{q_{k|k-1}} \frac{\pi_{k|k-1}(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})}{\rho_k(\mathbf{x}_{k|k-1}^{(i)}|\mathbf{x}_{k-1}^{(i)}, \mathbf{Z}_k)} w_{k-1}^{(i)}, & i = 1, \dots, N \\ \frac{p_b (1 - q_{k-1|k-1})}{q_{k|k-1}} \frac{b_{k|k-1}(\mathbf{x}_{k|k-1}^{(i)})}{\beta_k(\mathbf{x}_{k|k-1}^{(i)}|\mathbf{Z}_k)} \frac{1}{B}, & i = N+1, \dots, N+B. \end{cases} \quad (4.4)$$

As discussed in Sect. 2.1, the simplest choice of importance density $\rho_k(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{Z}_k)$ is the transitional density $\pi_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1})$. For more efficient Bernoulli particle filters, however, better importance densities discussed in [4–9] are desirable.

In the absence of any prior knowledge on object birth, it is necessary to assume that the object of interest can appear anywhere in the state space \mathcal{X} . One may attempt in this case to model $b_{k|k-1}(\mathbf{x})$ by the uniform distribution over \mathcal{X} . The birth proposal β_k in (4.3) needs to have the same support as $b_{k|k-1}(\mathbf{x})$ (i.e. the entire \mathcal{X}) and consequently this approach would lead to a very inefficient algorithm which requires a massive number of birth particles.

A more efficient alternative is to use the measurements to build adaptively the birth density [3]. In this case the target birth density at k is represented by

$$b_{k|k-1}(\mathbf{x}) = \int \pi_{k|k-1}(\mathbf{x}|\mathbf{x}') \cdot b_{k-1}(\mathbf{x}'; \mathbf{Z}_{k-1}) d\mathbf{x}' \quad (4.5)$$

where $b_{k-1}(\mathbf{x}; \mathbf{Z}_{k-1})$ is the object birth density at $k-1$. This density can be constructed using the measurement set at $k-1$ and prior knowledge about targets (e.g. max/min speed). Let the particles $\mathbf{x}_{k-1}^{(i)} \sim b_{k-1}(\mathbf{x}; \mathbf{Z}_{k-1})$, for $i = N+1, \dots, N+B_{k-1}$, approximate the birth density at $k-1$. Here B_{k-1} is the number of newborn particles at $k-1$ and is typically selected as a multiple of the cardinality $|\mathbf{Z}_{k-1}|$. Then (4.3) and (4.4) can be replaced with:

$$\mathbf{x}_{k|k-1}^{(i)} \sim \rho_k(\mathbf{x}|\mathbf{x}_{k-1}^{(i)}, \mathbf{Z}_k), \quad i = 1, \dots, N, N+1, \dots, N+B_{k-1} \quad (4.6)$$

$$w_{k|k-1}^{(i)} = \begin{cases} \frac{p_s q_{k-1|k-1}}{q_{k|k-1}} \frac{\pi_{k|k-1}(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})}{\rho_k(\mathbf{x}_{k|k-1}^{(i)}|\mathbf{x}_{k-1}^{(i)}, \mathbf{Z}_k)} w_{k-1}^{(i)}, & i = 1, \dots, N \\ \frac{p_b(1-q_{k-1|k-1})}{q_{k|k-1}} \frac{\pi_{k|k-1}(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})}{\rho_k(\mathbf{x}_{k|k-1}^{(i)}|\mathbf{x}_{k-1}^{(i)}, \mathbf{Z}_k)} \frac{1}{B_{k-1}} & i = N+1, \dots, N+B_{k-1} \end{cases} \quad (4.7)$$

respectively.

The implementation of the update step of the Bernoulli filter follows from (2.47–2.49). First, for every $\mathbf{z} \in \mathbf{Z}_k$, the integral $I_k(\mathbf{z}) = \int g_k(\mathbf{z}|\mathbf{x}) s_{k|k-1}(\mathbf{x}) d\mathbf{x}$, which features in (2.49), is approximately computed as:

$$I_k(\mathbf{z}) \approx \sum_{i=1}^{N+B_{k-1}} w_{k|k-1}^{(i)} \cdot g_k(\mathbf{z}|\mathbf{x}_{k|k-1}^{(i)})$$

Then based on (2.49),

$$\delta_k \approx p_{\text{D}} \left(1 - \sum_{\mathbf{z} \in \mathbf{Z}_k} \frac{I_k(\mathbf{z})}{\lambda c(\mathbf{z})} \right).$$

Now the probability of existence is updated using (2.47) and weights are updated according to (2.48) as:

$$\tilde{w}_{k|k}^{(i)} \approx \left[1 - p_{\text{D}} + p_{\text{D}} \sum_{\mathbf{z} \in \mathbf{Z}_k} \frac{g_k(\mathbf{z}|\mathbf{x}_{k|k-1}^{(i)})}{\lambda c(\mathbf{z})} \right] \cdot w_{k|k-1}^{(i)} \quad (4.8)$$

These weights are then normalized to obtain

$$w_{k|k}^{(i)} = \frac{\tilde{w}_{k|k}^{(i)}}{\sum_{j=1}^{N+B_{k-1}} \tilde{w}_{k|k}^{(j)}}, \quad (4.9)$$

for $i = 1, \dots, N + B_{k-1}$. Finally, in order to avoid degeneracy of particles it is necessary to resample N times from $\{w_{k|k}^{(i)}, \mathbf{x}_{k|k-1}^{(i)}\}_{i=1}^{N+B_{k-1}}$, resulting in a particle approximation of $s_{k|k}(\mathbf{x})$.

The pseudo-code of the Bernoulli bootstrap-type particle filter is given in Algorithm 3. The resampling step in Algorithm 3 (lines 13–16) typically is followed by some form of particle diversification (e.g. MCMC move step). It can be easily verified that the predicted particle weights $w_{k|k-1}^{(i)}$, for $i = 1, \dots, N + B_{k-1}$, computed in lines 6 and 7 of Algorithm 3, are normalized.

Applications of the Bernoulli particle filter will be presented in Sect. 5.1.

Algorithm 3 Pseudo-code of the Bernoulli bootstrap filter

```

1: function BERNOLLII BOOTSTRAP FILTER
2:   Input:  $q_{k-1|k-1}$ ,  $\{\mathbf{x}_{k-1}^{(i)}\}_{i=1}^N$ ,  $\mathbf{Z}_{k-1}$ ,  $\mathbf{Z}_k$ 
3:   Draw:  $\mathbf{x}_{k|k-1}^{(i)} \sim b_{k-1}(\mathbf{x}|\mathbf{Z}_{k-1})$  for  $i = N+1, \dots, N+B_{k-1}$ 
4:    $q_{k|k-1} = p_b \cdot (1 - q_{k-1|k-1}) + p_s \cdot q_{k-1|k-1}$  ▷ Eq. (2.44)
5:   Draw a sample:  $\mathbf{x}_{k|k-1}^{(i)} \sim \pi_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1}^{(i)})$  for  $i = 1, \dots, N+B_{k-1}$ 
6:   Predicted weight:  $w_{k|k-1}^{(i)} = \frac{p_s q_{k-1|k-1}}{N q_{k|k-1}}$  for  $i = 1, \dots, N$  ▷ See Eq. (2.45)
7:   Predicted weight:  $w_{k|k-1}^{(i)} = \frac{p_b (1 - q_{k-1|k-1})}{B_{k-1} q_{k|k-1}}$  for  $i = N+1, \dots, N+B_{k-1}$  ▷ See Eq. (2.45)
8:   For every  $\mathbf{z} \in \mathbf{Z}_k$  compute  $I_k(\mathbf{z}) = \sum_{i=1}^{N+B_{k-1}} w_{k|k-1}^{(i)} \cdot g_k(\mathbf{z}|\mathbf{x}_{k|k-1}^{(i)})$ 
9:   Compute  $\delta_k \approx p_D \left(1 - \sum_{\mathbf{z} \in \mathbf{Z}_k} \frac{I_k(\mathbf{z})}{\lambda c(\mathbf{z})}\right)$ .
10:  Update existence:  $q_{k|k} = \frac{1 - \delta_k}{1 - \delta_k \cdot q_{k|k-1}} \cdot q_{k|k-1}$  ▷ Eq. (2.47)
11:  Update weights:  $\tilde{w}_{k|k}^{(i)} \approx \left[1 - p_D + p_D \sum_{\mathbf{z} \in \mathbf{Z}_k} \frac{g_k(\mathbf{z}|\mathbf{x}_{k|k-1}^{(i)})}{\lambda c(\mathbf{z})}\right] \cdot w_{k|k-1}^{(i)}$ , for  $i = 1, \dots, N+B_{k-1}$ 
12:  Normalise weights:  $w_{k|k}^{(i)} = \frac{\tilde{w}_{k|k}^{(i)}}{\sum_{j=1}^{N+B_{k-1}} \tilde{w}_{k|k}^{(j)}}$ , for  $i = 1, \dots, N+B_{k-1}$ 
13:  for  $i = 1, \dots, N$  do ▷ (Resampling)
14:    Select index  $j^i \in \{1, \dots, N\}$  with probability  $w_{k|k}^{(i)}$ 
15:     $\mathbf{x}_k^{(i)} = \mathbf{x}_{k|k-1}^{(j^i)}$ 
16:  end for
17:  Output:  $q_{k|k}$ ,  $\{\mathbf{x}_k^{(i)}\}_{i=1}^N$ 
18: end function

```

4.1.2 Bernoulli Box-Particle Filter

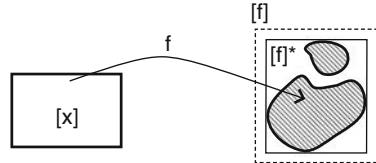
Box particle filtering is a recently emerged technique for numeric implementation of stochastic filters [10], resulting from the synergy between the sequential Monte Carlo (SMC) method [4] and interval analysis [11]. This technique is particularly appealing in situations involving imprecise stochastic measurements (e.g. intervals) that typically result in fairly broad posterior densities. It relies on the concept of *box particle* that occupies a small and controllable rectangular region of non-zero volume in the state space. The key advantage of the box particle filter (Box-PF) against the standard particle filter is the potential to reduce the number of particles required to approximate the posterior density.

This section first explains the bootstrap type box-particle filtering approach and then applies it to the Bernoulli filter for interval measurements. The section is based on the material presented in [13] and [14].

4.1.2.1 Box-Particle Filter

The key idea of the box-PF is to replace the Dirac delta function in approximation (2.1) with a box function defined as:

Fig. 4.1 Nonlinear transformation \mathbf{f} of a box $[\mathbf{x}]$ result is the *shaded image*; $[\mathbf{f}]$ is an inclusion of the image, $[\mathbf{f}]^*$ is the minimal inclusion of the image



$$\Delta_{[\mathbf{x}]}(\mathbf{x}) = \begin{cases} \frac{1}{|[\mathbf{x}]|}, & \text{if } \mathbf{x} \in [\mathbf{x}] \\ 0, & \text{otherwise} \end{cases} \quad (4.10)$$

where $[\mathbf{x}] \in \mathbb{IR}^{n_x} = \{[\mathbf{x}] \subset \mathbb{R}^{n_x}\}$ is a closed interval in \mathbb{R}^{n_x} and $|[\mathbf{x}]|$ is its volume. The box function $\Delta_{[\mathbf{x}]}(\mathbf{x})$ is effectively a uniform PDF with support $[\mathbf{x}]$. Suppose the measurements are crisp interval measurements (a special case of imprecise measurement ζ , discussed in Sect. 2.3.1), denoted by $[\mathbf{z}]$. Let the posterior PDF at $k - 1$ be approximated as

$$p(\mathbf{x}_{k-1}|[\mathbf{z}]_{1:k-1}) \approx \frac{1}{N} \sum_{i=1}^N \Delta_{[\mathbf{x}]_{k-1}^{(i)}}(\mathbf{x}_{k-1}) \quad (4.11)$$

where $[\mathbf{x}]_{k-1}^{(i)}$ is a box particle and all weights are $1/N$ as in the bootstrap filter. The prediction (1.3) is then:

$$p(\mathbf{x}_{k|k-1}|[\mathbf{z}]_{1:k-1}) \approx \frac{1}{N} \sum_{i=1}^N \frac{1}{|[\mathbf{x}]_{k-1}^{(i)}|} \int_{[\mathbf{x}]_{k-1}^{(i)}} \pi_{k|k-1}(\mathbf{x}_k | \mathbf{x}_{k-1}) d\mathbf{x}_{k-1}. \quad (4.12)$$

Using the dynamic model (1.1), where \mathbf{f}_{k-1} is a nonlinear function, we encounter the first problem with box-particles: *a nonlinear transformation of a box is not a box*. This is illustrated in Fig. 4.1. The problem is solved using the concept of *inclusion function*, borrowed from interval analysis [11].

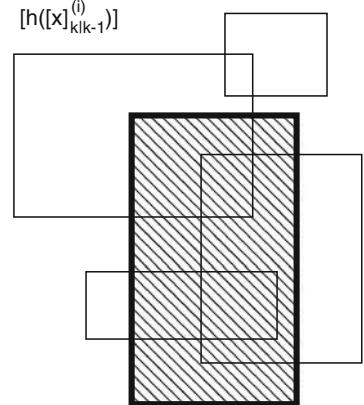
Definition

Let $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$. An “interval function” $[\mathbf{f}] : \mathbb{IR}^n \rightarrow \mathbb{IR}^m$ is said to be an inclusion function for \mathbf{f} if $\forall [\mathbf{x}] \in \mathbb{IR}^n: \mathbf{f}([\mathbf{x}]) \subseteq [\mathbf{f}([\mathbf{x}])]$.

A minimal inclusion function $[\mathbf{f}([\mathbf{x}])]^*$ is the smallest interval enclosing $\mathbf{f}([\mathbf{x}])$. Finding the smallest possible inclusion functions that can be evaluated in a convenient computational time is one of the main research topics of interval analysis [11]. Inclusion functions for elementary functions such as exp, ln, tan, cos and sin, are well studied and known.

In the context of the bootstrap filter, the predicted box particle is then an inclusion:

Fig. 4.2 The received measurement box $[\mathbf{z}]_k$ (shaded rectangle) and inclusion functions of box-particles projected into the measurement domain, $[\mathbf{h}_k([\mathbf{x}]_{k|k-1}^{(i)})]$, for $i = 1, \dots, N$



$$[\mathbf{x}]_{k|k-1}^{(i)} = [\mathbf{f}_{k-1}([\mathbf{x}]_{k-1}^{(i)}) + \mathbf{v}_{k-1}] \quad (4.13)$$

that is the expression $\int_{[\mathbf{x}]_{k-1}^{(i)}} \pi_{k|k-1}(\mathbf{x}_k | \mathbf{x}_{k-1}) d\mathbf{x}_{k-1} / |[\mathbf{x}]_{k-1}^{(i)}|$, which features in (4.12), is approximated by $\Delta_{[\mathbf{x}]_{k|k-1}^{(i)}}(\mathbf{x}_k)$, leading to:

$$p(\mathbf{x}_k | [\mathbf{z}]_{1:k-1}) \approx \frac{1}{N} \sum_{i=1}^N \Delta_{[\mathbf{x}]_{k|k-1}^{(i)}}(\mathbf{x}_k). \quad (4.14)$$

Note that by making the volume of the boxes smaller, the error in the approximation illustrated by Fig. 4.1 is reduced.

The update stage the bootstrap box-PF involves the contraction of predicted box-particles. Suppose the measurement function is given by (1.2). Figure 4.2 shows the received measurement box $[\mathbf{z}]_k$ (shaded rectangle) and inclusion functions of predicted box-particles projected into the measurement domain, i.e. $[\mathbf{h}_k([\mathbf{x}]_{k|k-1}^{(i)})]$, for $i = 1, \dots, N$. The goal of the contraction step is to reduce the size of predicted box-particles $[\mathbf{x}]_{k|k-1}^{(i)}$ so that their projected inclusions $[\mathbf{h}_k([\mathbf{x}]_{k|k-1}^{(i)})]$ overlap with the measurement box $[\mathbf{z}]_k$. Those predicted box-particles for which this operation results in non-zero volume box-particles are assigned weights equal to 1. Others, for which there is no overlap between $[\mathbf{z}]_k$ and $[\mathbf{h}_k([\mathbf{x}]_{k|k-1}^{(i)})]$ are assigned weight 0. The rationale is that the measurement box is guaranteed to *include* the *true* value $\mathbf{h}_k(\mathbf{x})$. Finally the box-particles are resampled so that again N of them are approximating the posterior PDF. In order to increase the diversity, if a box-particle has been selected during resampling n times, it is divided into n smaller box-particles which jointly have the same support as the original box-particle.

Several interval contractors are described in [11, Chap. 4], including the Gauss elimination, Gauss-Seidel algorithms and linear programming. An attractive contractor method is the so-called *Constraints Propagation* (CP) technique [11]. The main

advantage of the CP method is its efficiency in the presence of high redundancy of data and constraints. The CP method is also simple and, most importantly, independent of non-linearities. The CP algorithm is explained by an example. Suppose the state vector is $\mathbf{x} = [x \ \dot{x} \ y \ \dot{y}]^T$, where (x, y) is position and (\dot{x}, \dot{y}) velocity in Cartesian coordinates. Let the measurement vector consist of range r and azimuth θ , that is:

$$\mathbf{h}_k(\mathbf{x}) = \begin{bmatrix} \sqrt{x^2 + y^2} \\ \arctan\left(\frac{y}{x}\right) \end{bmatrix}. \quad (4.15)$$

The relationship in (4.15) is exploited as a constraint which results in a more precise state vector interval. The CP algorithm for this case is listed in Algorithm 4. Note that arithmetic operations in Algorithm 4 (addition, subtraction, multiplication, division, square root) are carried out on intervals, using interval arithmetic [11]. Interval analysis techniques can be conveniently implemented in MATLAB using INTLAB [12].

Algorithm 4 Constraints propagation algorithm

- 1: **Input:** $[\mathbf{x}] = [x] \times [\dot{x}] \times [y] \times [\dot{y}], [\mathbf{z}] = [r] \times [\theta]$
 - constraint 1**
 - 2: $[x] \leftarrow [x] \cap \sqrt{[r]^2 - [y]^2}$
 - 3: $[y] \leftarrow [y] \cap \sqrt{[r]^2 - [x]^2}$
 - 4: $[r] \leftarrow [r] \cap \sqrt{[x]^2 + [y]^2}$
 - constraint 2**
 - 5: $[x] \leftarrow [x] \cap \frac{[y]}{\tan(\theta)}$
 - 6: $[y] \leftarrow [y] \cap \tan(\theta) \cdot [x]$
 - 7: $[\theta] \leftarrow [\theta] \cap \arctan\left(\frac{[y]}{[x]}\right)$
 - 8: **Output:** $[\mathbf{x}] = [x] \times [\dot{x}] \times [y] \times [\dot{y}],$
-

4.1.2.2 Bernoulli Box-PF

The Bernoulli bootstrap box-PF is a modification of Algorithm 3, where “standard” particles are replaced with box-particles, following the ideas presented above. Let $[\mathbf{Z}]_k = \{[\mathbf{z}]_{k,1}, \dots, [\mathbf{z}]_{k,m_k}\}$ be the interval measurement set at k . The pseudo-code of the Bernoulli bootstrap box-PF is given in Algorithm 5. The prediction (lines 3–7) is straightforward. The most important step is in line 5, where inclusion function is applied to compute the predicted box-particles $[\mathbf{x}]_{k|k-1}^{(i)}$, for $i = 1, \dots, N + B_{k-1}$. The update of the Bernoulli box-particle filter is somewhat more involved. Lines 8–15 compute δ_k , which is required in the computation of the existence probability, see (2.47). Bearing in mind (2.48), the posterior PDF $s_{k|k}(\mathbf{x})$ is approximated by creating $m_k + 1$ sets of box particles: one set of $N + B_{k-1}$ box particles $[\tilde{\mathbf{x}}]_k^i$ with weights $(1 - p_D)w_{k|k-1}^i$ (lines 17–20) and m_k sets of $N + B_{k-1}$ box particles with weights $\tilde{w}_{k|k-1}^{i,j}$ obtained using the m_k measurements (lines 21–26). After normalization of

weights, the box-particles are resampled (line 28). Diversification of box-particles can be achieved by division into smaller box-particles which jointly have the same support as the original box-particle.

Algorithm 5 Pseudo-code of the Bernoulli bootstrap box-particle filter

```

1: function BERNOLLI BOOTSTRAP BOX- PF
2:   Input:  $q_{k-1|k-1}$ ,  $\{[\mathbf{x}]_{k-1}^{(i)}\}_{i=1}^N$ ,  $[\mathbf{Z}]_{k-1}$ ,  $[\mathbf{Z}]_k$ 
3:   Draw:  $[\mathbf{x}]_{k-1}^{(i)} \sim b_{k-1}([\mathbf{x}][\mathbf{Z}]_{k-1})$  for  $i = N+1, \dots, N+B_{k-1}$ 
4:    $q_{k|k-1} = p_b \cdot (1 - q_{k-1|k-1}) + p_s \cdot q_{k-1|k-1}$  ▷ Eq. (2.44)
5:   Predicted box-particles:  $[\mathbf{x}]_{k|k-1}^{(i)} = [\mathbf{f}_{k-1}([\mathbf{x}]_{k-1}^{(i)}) + \mathbf{v}_{k-1}]$  for  $i = 1, \dots, N+B_{k-1}$ 
6:   Predicted weight:  $w_{k|k-1}^{(i)} = \frac{p_s q_{k-1|k-1}}{N q_{k|k-1}}$  for  $i = 1, \dots, N$ 
7:   Predicted weight:  $w_{k|k-1}^{(i)} = \frac{p_b (1 - q_{k-1|k-1})}{B_{k-1} q_{k|k-1}}$ 
8:   for  $j = 1, \dots, m_k$  do
9:     for  $i = 1, \dots, N+B_{k-1}$  do
10:       if  $[\mathbf{x}]_{k|k-1}^{(i)} \cap [\mathbf{z}] \neq \emptyset$  then  $g_k([\mathbf{z}]_{k,j} | [\mathbf{x}]_{k|k-1}^{(i)}) = 1$ 
11:       else  $g_k([\mathbf{z}]_{k,j} | [\mathbf{x}]_{k|k-1}^{(i)}) = 0$ 
12:       end if
13:     end for
14:      $I_k([\mathbf{z}]_{k,j}) = \sum_{i=1}^{N+B_{k-1}} w_{k|k-1}^{(i)} \cdot g_k([\mathbf{z}]_{k,j} | [\mathbf{x}]_{k|k-1}^{(i)})$ 
15:   end for
16:   Compute  $\delta_k \approx p_D \left( 1 - \sum_{[\mathbf{z}] \in [\mathbf{Z}]_k} \frac{I_k([\mathbf{z}])}{\lambda c([\mathbf{z}])} \right)$ . for  $i = N+1, \dots, N+B_{k-1}$ 
17:   for  $i = 1, \dots, N+B_{k-1}$  do
18:      $[\tilde{\mathbf{x}}]_k^{(i,0)} = [\mathbf{x}]_{k|k-1}^{(i)}$ 
19:      $\tilde{w}_k^{(i,0)} = (1 - p_D) w_{k|k-1}^{(i)}$ 
20:   end for
21:   for  $j = 1, \dots, m_k$  do
22:     for  $i = 1, \dots, N+B_{k-1}$  do
23:       Apply contraction:  $[\tilde{\mathbf{x}}]_k^{(i,j)} = \text{Contract}([\mathbf{x}]_{k|k-1}^{(i)}, [\mathbf{z}]_{k,j})$ 
24:        $\tilde{w}_k^{(i,j)} = p_D \frac{I_k([\mathbf{z}]_{k,j})}{\lambda c([\mathbf{z}]_{k,j})} w_{k|k-1}^{(i)}$ 
25:     end for
26:   end for
27:   Normalise weights:  $w_k^{(i,j)} = \frac{\tilde{w}_k^{(i,j)}}{\sum_{\ell=1}^{N+B_{k-1}} \sum_{n=0}^{m_k} \tilde{w}_k^{(\ell,n)}}$ , for  $i = 1, \dots, N+B_{k-1}$  and  $j = 0, \dots, m_k$ 
28:   Resample  $N$  times from  $\{w_k^{(i,j)}, [\tilde{\mathbf{x}}]_k^{(i,j)}\}_{1 \leq i \leq N+B_{k-1}; 0 \leq j \leq m_k}$  to obtain  $\{[\mathbf{x}]_k^{(i)}\}_{i=1}^N$ 
29:   Output:  $\{[\mathbf{x}]_k^{(i)}\}_{i=1}^N$ 
30: end function

```

Next we present a numerical example where the problem is to track a moving target in a two-dimensional plane, using range, range-rate and azimuth measurements. The target state vector is $\mathbf{x} = [x \ \dot{x} \ y \ \dot{y}]^\top$, where (x, y) and (\dot{x}, \dot{y}) are the target position and velocity, respectively, in Cartesian coordinates. The target is moving according to the nearly constant velocity motion model with transitional density $\pi_{k|k-1}(\mathbf{x}|\mathbf{x}') = \mathcal{N}(\mathbf{x}; \mathbf{F}\mathbf{x}', \mathbf{Q})$. Here

$$\mathbf{F} = \mathbf{I}_2 \otimes \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \quad \mathbf{Q} = \mathbf{I}_2 \otimes \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} \\ \frac{T^2}{2} & T \end{bmatrix} \cdot \varpi, \quad (4.16)$$

\otimes is the Kronecker product, $T = t_{k+1} - t_k$ is the sampling interval and ϖ is intensity of process noise [15]. The target appears at scan $k = 3$ and disappears at scan $k = 54$. Initially (at $k = 0$) the target is located at (550, 300) m and is moving with velocity $(-5, -8.5)$ m/s. The sensor is static, located at the origin of the $x - y$ plane. Other values are adopted as $\varpi = 0.05$, $T = 1$ s, with the total observation interval of 60 s.

Measurement noise \mathbf{w}_k is zero mean white Gaussian with a covariance $\Sigma = \text{diag}[\sigma_r^2, \sigma_r^2, \sigma_\theta^2]$, where $\sigma_r = 2.5$ m, $\sigma_r = 0.01$ m/s and $\sigma_\theta = 0.25^\circ$. The sensor provides interval measurements, with interval length $\Delta = [\Delta r, \Delta \dot{r}, \Delta \theta]^\top$, where $\Delta r = 50$ m, $\Delta \dot{r} = 0.2$ m/s and $\Delta \theta = 4^\circ$ are the lengths of intervals in range, range-rate and azimuth, respectively. The sensor has a bias (systematic error) in the sense that the vector $h_k(\mathbf{x}) + \mathbf{w}_k$ is not in the middle of the measurement interval. The biased measurement at k is given by:

$$[\mathbf{z}]_k = [h_k(\mathbf{x}) + \mathbf{w}_k - \frac{3}{4}\Delta, h_k(\mathbf{x}) + \mathbf{w}_k + \frac{1}{4}\Delta]. \quad (4.17)$$

The Bernoulli box-PF filter is ignorant of the sensor bias.

The probability of detection is $p_D = 0.95$, the mean number of clutter detections per scan is $\lambda = 8$. The clutter detection spatial distribution $c([\mathbf{z}])$ is uniform across the range (mid intervals from 30 to 700 m), range-rate (mid intervals from -15 to $+15$ m/s) and azimuth (mid intervals from $-\pi/2$ to $\pi/2$ rad). The output of the Bernoulli filter is reported only if the existence probability $q_{k|k}$ is above the threshold of 0.5. p_D , clutter statistics λ and $c([\mathbf{z}])$, measurement function $h_k(\mathbf{x})$, covariance matrix Σ and the transitional density $\pi_{k|k-1}(\mathbf{x}|\mathbf{x}')$. The filter estimates the posterior at every k , using measurements $[\mathbf{Z}]_{1:k}$. The following parameters are used in the filter: $p_B = 0.01$, $p_s = 0.98$, $N = 10$ persistent box particles and $B_{k-1} = m_{k-1}$ birth box particles.

Figure 4.3 shows the results from a single run of the Bernoulli box-PF, described in Algorithm 5. Figure 4.3a displays the situation at $k = 45$: the interval measurements, the estimated trajectory and the current estimate of the posterior spatial PDF $s_{k|k}(\mathbf{x})$, represented by 10 box-particles. This estimate is zoomed-in in Fig. 4.3b, which also indicates the true target location (square) and the current point estimate (asterisk). Finally, Fig. 4.3b presents the estimate of the probability of target existence $q_{k|k}$ over time. Target presence is established at $k = 6$ with $q_{k|k}$ remaining close to 1.0 after that. Occasionally, when the target detection is missing from the measurement set $[\mathbf{Z}]_k$, existence probability drops below the value of 1.0. Overall Fig. 4.3 illustrates the successful operation of the Bernoulli box-PF: (1) the update step appears to correctly weight the relevant box particles and (2) the contraction steps combined with the resampling-subdivision steps are reducing the size of box particles over time.

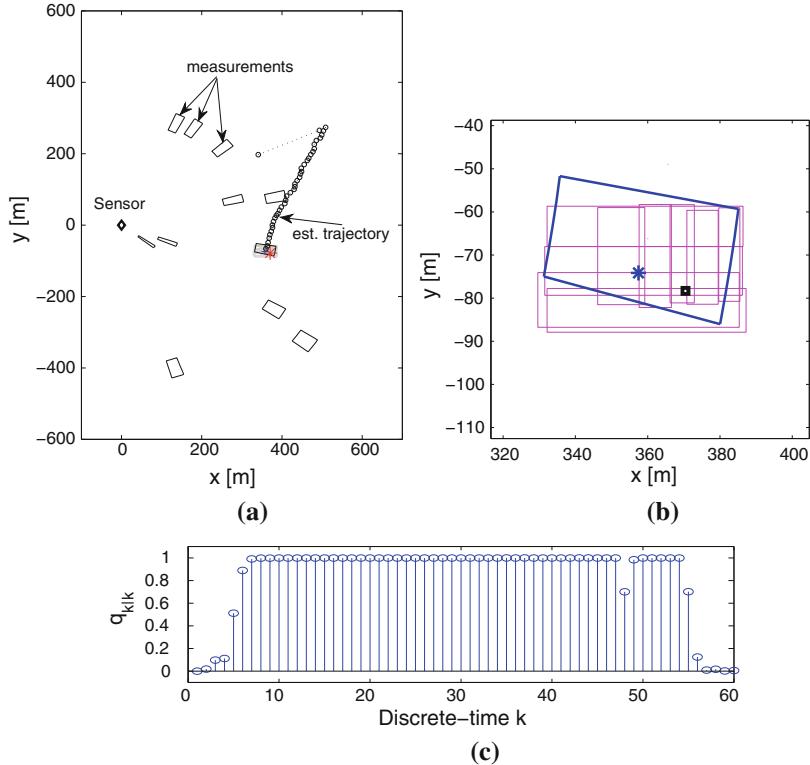


Fig. 4.3 A single run of the Bernoulli box-PF: **a** The situation at $k = 45$: the measurements, the estimated trajectory, and the approximation of the posterior $s_{k|k}(\mathbf{x})$; **b** Zoom-in, showing the 10 box particles, the target originated interval measurement (the quadrilateral drawn using thick lines), the true target location (\square) and the point estimate from the filter (*); (c) The estimate of the probability of target existence $q_{k|k}$ over time

The performance of the Bernoulli box-PF was compared with the standard Bernoulli particle filter in Gning et al. [13], using the described simulation setup. The results suggest that the Bernoulli box-PF is more cost efficient: it required twice less computational time and almost hundred times less particles (that is box-particles) to achieve similar performance. Its drawback was a slightly broader estimate of the posterior $s_{k|k}(\mathbf{x})$.

4.2 PHD/CPDH Particle Filters with Adaptive Birth Intensity

Standard formulations of PHD and CPHD filters assume that target birth intensity is known a priori. Typically the birth intensity has the majority of its mass distributed over small specific areas of the state space \mathcal{X} , which, for example in the context

of air surveillance, can be interpreted as the regions around airports [19, 20]. Note however, that if a target appears in a region that is not covered by the predefined birth intensity, the PHD/CPHD filter will be completely blind to its existence. Making the target birth intensity diffuse so that it covers the entire state space, typically results in a higher incidence of short-lived false tracks and longer confirmation times. The only way to overcome this drawback is to create at each processing step of the filter a massive number of potential (hypothesized) newborn targets covering the entire state-space, which is clearly inefficient. The described limitation affects both methods of PHD/CPHD filter implementation: the SMC method [21, 22] and the finite Gaussian mixtures (GM) [19, 20].

Starting from the standard equations of the PHD and the CPHD filter, this section derives novel extensions which distinguish, in both the prediction and the update step, between the persistent and the newborn targets. This approach allows the PHD/CPHD filter to adapt the target birth intensity at each processing step using the received measurements. The resulting measurement driven birth intensity thus removes the need for the prior specification of birth intensities and eliminates the restriction on target appearance volumes within \mathcal{X} . Measurement driven target birth intensity is complementary with the recent attempts to improve the efficiency of the SMC-PHD filter by pre-selecting particles for propagation (the so-called auxiliary particle PHD filter) presented in Whiteley et al. [22].

4.2.1 Extension of the PHD Filter

In the original formulation of the PHD filter, Sect. 2.5.2, new targets are “born” in the prediction step (2.51). The intensity function of newborn targets, $\gamma_{k|k-1}(\mathbf{x})$, is independent of measurements, and in the general case, where the targets can appear anywhere in the state space, it has to cover the entire state space \mathcal{X} . As it was argued above, this is inefficient and wasteful. Instead we propose to design a newborn target intensity in the region of the state-space $\mathbf{x} \in \mathcal{X}$ for which the likelihood $g_k(\mathbf{z}|\mathbf{x})$ will have high values.

It turns out, however, that if the birth intensity is adapted in accordance with the measurements, the PHD equations must be applied in a slightly different form [23]. Start from (2.51) and (2.52), where the state vector \mathbf{x} consists of the usual kinematic/feature component (position, velocity, amplitude, etc) which we denote by \mathbf{y} and a mark or a label β , which distinguishes a newborn target from the persistent target, i.e. $\mathbf{x} = (\mathbf{y}, \beta)$. Let us adopt the convention:

$$\beta = \begin{cases} 0 & \text{for a persistent target} \\ 1 & \text{for a newborn target} \end{cases} \quad (4.18)$$

and let $\mathbf{y} \in \mathcal{Y}$. The birth PHD is then:

$$\gamma_{k|k-1}(\mathbf{x}) = \gamma_{k|k-1}(\mathbf{y}, \beta) = \begin{cases} \gamma_{k|k-1}(\mathbf{y}), & \beta = 1 \\ 0, & \beta = 0. \end{cases} \quad (4.19)$$

Note a slight abuse of notation in using the same symbol $\gamma_{k|k-1}$ for both functions of \mathbf{x} and \mathbf{y} . Similar abuse will be used throughout this section, but the meaning should be clear from the context.

A newborn target becomes a persisting target at the next time, but a persisting target cannot become a newborn target. Thus the mark β can only change from 1 to 0 but not vice-versa. The transition model is then

$$\begin{aligned} \pi_{k|k-1}(\mathbf{x}|\mathbf{x}') &= \pi_{k|k-1}(\mathbf{y}, \beta|\mathbf{y}', \beta') \\ &= \pi_{k|k-1}(\mathbf{y}|\mathbf{y}')\pi_{k|k-1}(\beta|\beta') \end{aligned} \quad (4.20)$$

with

$$\pi_{k|k-1}(\beta|\beta') = \begin{cases} 0, & \beta = 1 \\ 1, & \beta = 0. \end{cases} \quad (4.21)$$

The probability of survival does not depend on β and hence

$$p_s(\mathbf{x}) = p_s(\mathbf{y}, \beta) = p_s(\mathbf{y}) \quad (4.22)$$

The PHD filter prediction Eq. (2.51) for the augmented state vector is given by:

$$\begin{aligned} D_{k|k-1}(\mathbf{y}, \beta) &= \gamma_{k|k-1}(\mathbf{y}, \beta) \\ &+ \sum_{\beta'=0}^1 \int D_{k-1|k-1}(\mathbf{y}', \beta') p_s(\mathbf{y}', \beta') \pi_{k|k-1}(\mathbf{y}, \beta|\mathbf{y}', \beta') d\mathbf{y}'. \end{aligned} \quad (4.23)$$

Upon the substitution of expressions (4.19–4.22) into (4.23) we obtain the new form of the PHD filter prediction:

$$\begin{aligned} D_{k|k-1}(\mathbf{y}, \beta) &= \begin{cases} \gamma_{k|k-1}(\mathbf{y}), & \beta = 1 \\ p_s \int \pi_{k|k-1}(\mathbf{y}|\mathbf{y}') \left[D_{k-1|k-1}(\mathbf{y}', 1) + D_{k-1|k-1}(\mathbf{y}', 0) \right] d\mathbf{y}', & \beta = 0. \end{cases} \end{aligned} \quad (4.24)$$

Now we carry out similar manipulations for the update step. First, note that since the new targets are created from measurements, it follows that newtargets are always

detected (as they always generate measurements), so we can write:

$$p_{\text{D}}(\mathbf{x}) = p_{\text{D}}(\mathbf{y}, \beta) = \begin{cases} 1, & \beta = 1 \\ p_{\text{D}}(\mathbf{y}), & \beta = 0 \end{cases} \quad (4.25)$$

The measurement does not depend on β , hence

$$g_k(\mathbf{z}|\mathbf{x}) = g_k(\mathbf{z}|\mathbf{y}, \beta) = g_k(\mathbf{z}|\mathbf{y}). \quad (4.26)$$

The PHD update Eq. (2.52) for the augmented state vector is given by:

$$\begin{aligned} D_{k|k}(\mathbf{y}, \beta) &= [1 - p_{\text{D}}(\mathbf{y}, \beta)]D_{k|k-1}(\mathbf{y}, \beta) \\ &+ \sum_{\mathbf{z} \in \mathbf{Z}_k} \frac{p_{\text{D}}(\mathbf{y}, \beta)g_k(\mathbf{z}|\mathbf{y}, \beta)D_{k|k-1}(\mathbf{y}, \beta)}{\kappa_k(\mathbf{z}) + \sum_{\beta=0}^1 \langle p_{\text{D}}(\cdot, \beta)g_k(\mathbf{z}|\cdot, \beta), D_{k|k-1}(\cdot, \beta) \rangle}, \end{aligned} \quad (4.27)$$

where $\langle f, g \rangle = \int f(\mathbf{a}) g(\mathbf{a}) d\mathbf{a}$ denotes the inner product between two real-valued functions f and g . Using (4.25) and (4.26), it follows that the update step for persisting targets ($\beta = 0$), can be expressed by:

$$\begin{aligned} D_{k|k}(\mathbf{y}, 0) &= [1 - p_{\text{D}}(\mathbf{y})]D_{k|k-1}(\mathbf{y}, 0) \\ &+ \sum_{\mathbf{z} \in \mathbf{Z}_k} \frac{p_{\text{D}}(\mathbf{y})g_k(\mathbf{z}|\mathbf{y})D_{k|k-1}(\mathbf{y}, 0)}{\kappa_k(\mathbf{z}) + \langle g_k(\mathbf{z}|\cdot), \gamma_{k|k-1} \rangle + \langle p_{\text{D}}g_k(\mathbf{z}|\cdot), D_{k|k-1}(\cdot, 0) \rangle} \end{aligned} \quad (4.28)$$

while the update step for newborn targets ($\beta = 1$) is given by:

$$D_{k|k}(\mathbf{y}, 1) = \sum_{\mathbf{z} \in \mathbf{Z}_k} \frac{g_k(\mathbf{z}|\mathbf{y}) \gamma_{k|k-1}(\mathbf{y})}{\kappa_k(\mathbf{z}) + \langle g_k(\mathbf{z}|\cdot), \gamma_{k|k-1} \rangle + \langle p_{\text{D}}g_k(\mathbf{z}|\cdot), D_{k|k-1}(\cdot, 0) \rangle}. \quad (4.29)$$

In summary, starting from the standard form of PHD filter equations, (2.51) and (2.52), we have derived a new form of PHD filter equations which are suitable for the measurement driven implementation of target birth intensity. The new equations are (4.24) for prediction and (4.28–4.29) for the update step.

Observe that both the prediction step and the update step are performed separately for newborn and persistent targets. The intensity functions corresponding to the two types of targets are added together and predicted jointly in the prediction step, Eq. (4.24), case $\beta = 0$. For reporting purposes, we are typically only interested in the updated intensity function of persistent targets (4.28). Note however, that (4.29) represents an explicit formula for the intensity function of the target-birth process.

4.2.2 Extension of the CPHD Filter

Using the same reasoning as in the previous section, the prediction step of the CPHD filter, in terms of the newborn and persistent targets, is given by Eqs. (2.53) and (4.24).

In order to express the CPHD filter update equations separately for the persistent and newborn targets, we will use the measurement model of (4.25) and (4.26). The update equation for cardinality distribution (2.55) preserves the original form, while (2.56) can be now written as:

$$\begin{aligned}
D_{k|k}(\mathbf{y}, \beta) &= (1 - p_{\text{D}}(\mathbf{y}, \beta)) \frac{\langle \Upsilon_k^1[D_{k|k-1}; \mathbf{Z}_k], \rho_{k|k-1} \rangle}{\langle \Upsilon_k^0[D_{k|k-1}; \mathbf{Z}_k], \rho_{k|k-1} \rangle} D_{k|k-1}(\mathbf{y}, \beta) \\
&+ \sum_{\mathbf{z} \in \mathbf{Z}_k} \frac{\langle \Upsilon_k^1[D_{k|k-1}; \mathbf{Z}_k \setminus \{\mathbf{z}\}], \rho_{k|k-1} \rangle}{\langle \Upsilon_k^0[D_{k|k-1}; \mathbf{Z}_k], \rho_{k|k-1} \rangle} \frac{\langle 1, \kappa_k \rangle}{\kappa_k(\mathbf{z})} p_{\text{D}}(\mathbf{y}, \beta) g_k(\mathbf{z}|\mathbf{y}, \beta) D_{k|k-1}(\mathbf{y}, \beta) \\
&= \begin{cases} \sum_{\mathbf{z} \in \mathbf{Z}_k} \frac{\langle 1, \kappa_k \rangle}{\kappa_k(\mathbf{z})} \frac{\langle \Upsilon_k^1[D_{k|k-1}; \mathbf{Z}_k \setminus \{\mathbf{z}\}], \rho_{k|k-1} \rangle}{\langle \Upsilon_k^0[D_{k|k-1}; \mathbf{Z}_k], \rho_{k|k-1} \rangle} g_k(\mathbf{z}|\mathbf{y}) \gamma_{k|k-1}(\mathbf{y}), & \beta = 1 \\ (1 - p_{\text{D}}(\mathbf{y})) \frac{\langle \Upsilon_k^1[D_{k|k-1}; \mathbf{Z}_k], \rho_{k|k-1} \rangle}{\langle \Upsilon_k^0[D_{k|k-1}; \mathbf{Z}_k], \rho_{k|k-1} \rangle} D_{k|k-1}(\mathbf{y}, 0) \\ + \sum_{\mathbf{z} \in \mathbf{Z}_k} \frac{\langle 1, \kappa_k \rangle}{\kappa_k(\mathbf{z})} \frac{\langle \Upsilon_k^1[D_{k|k-1}; \mathbf{Z}_k \setminus \{\mathbf{z}\}], \rho_{k|k-1} \rangle}{\langle \Upsilon_k^0[D_{k|k-1}; \mathbf{Z}_k], \rho_{k|k-1} \rangle} p_{\text{D}}(\mathbf{y}) g_k(\mathbf{z}|\mathbf{y}) D_{k|k-1}(\mathbf{y}, 0), & \beta = 0 \end{cases} \quad (4.30)
\end{aligned}$$

with

$$\begin{aligned}
\Upsilon_k^u[D_{k|k-1}, \mathbf{Z}](n) &= \sum_{j=0}^{\min(|\mathbf{Z}|, n)} (|\mathbf{Z}| - j)! \rho_{K,k}(|\mathbf{Z}| - j) P_{j+u}^n \\
&\times \frac{\langle 1 - p_{\text{D}}(\cdot), D_{k|k-1}(\cdot, 0) \rangle^{n-(j+u)}}{\langle 1, D_{k|k-1}(\cdot, 1) + D_{k|k-1}(\cdot, 0) \rangle^n} e_j(\Xi_k(D_{k|k-1}, \mathbf{Z})) \quad (4.31)
\end{aligned}$$

and

$$\Xi_k(D_{k|k-1}, \mathbf{Z}) = \left\{ \frac{\langle 1, \kappa_k \rangle}{\kappa_k(\mathbf{z})} \langle D_{k|k-1}(\cdot, 1) + p_{\text{D}}(\cdot) D_{k|k-1}(\cdot, 0), g(\mathbf{z}|\cdot) \rangle : \mathbf{z} \in \mathbf{Z} \right\} \quad (4.32)$$

The above two equations are obtained by evaluation of the following expressions that appear in (2.57) and (2.58):

$$\begin{aligned}\langle 1 - p_{\text{D}}, D \rangle &= \sum_{\beta=0}^1 \langle 1 - p_{\text{D}}(\cdot, \beta), D(\cdot, \beta) \rangle \\ &= \langle 1 - p_{\text{D}}(\cdot), D(\cdot, 0) \rangle\end{aligned}\quad (4.33)$$

$$\langle 1, D \rangle = \langle 1, D(\cdot, 0) + D(\cdot, 1) \rangle \quad (4.34)$$

$$\langle p_{\text{D}} D, g(\mathbf{z}|\cdot) \rangle = \langle D(\cdot, 1) + p_{\text{D}}(\cdot) D(\cdot, 0), g(\mathbf{z}|\cdot) \rangle \quad (4.35)$$

In a similar manner to Eqs. (4.28) and (4.29), Eq. (4.30) gives the updated intensities functions of the persisting targets and target-birth process separately.

4.2.3 Implementation

The section is devoted to the SMC implementation of the PHD and the CPHD filter, with measurement-driven target birth intensity. The pseudo-code of the resulting PHD particle filter is given in Algorithm 6. The pseudo-code of the CPHD particle filter is presented in Algorithm 7. Initially, at the discrete-time index $k = 0$, the assumption is that $\rho_{0|0}(n) = 1$ if $n = 0$, and zero for $n = 1, 2, \dots, v_{\max}$ (here v_{\max} represents the maximum anticipated number of targets), and $D_{0|0}(\mathbf{y}, 0) = D_{0|0}(\mathbf{y}, 1) = 0$. Random samples (particles) approximate the intensity functions; at time $k - 1$ the approximations can be expressed as:

$$D_{k-1|k-1}(\mathbf{y}, 0) \approx \sum_{i=1}^{N_{k-1}^p} w_{k-1,p}^{(i)} \delta_{\mathbf{y}_{k-1,p}^{(i)}}(\mathbf{y}) \quad (4.36)$$

$$D_{k-1|k-1}(\mathbf{y}, 1) \approx \sum_{i=1}^{N_{k-1}^b} w_{k-1,b}^{(i)} \delta_{\mathbf{y}_{k-1,b}^{(i)}}(\mathbf{y}) \quad (4.37)$$

where $\delta_{\mathbf{y}_0}(\mathbf{y})$ as usual is the Dirac delta function concentrated at \mathbf{y}_0 [1, p. 693], $\{w_{k-1,p}^{(i)}, \mathbf{y}_{k-1,p}^{(i)}\}_{i=1}^{N_{k-1}^p}$ and $\{w_{k-1,b}^{(i)}, \mathbf{y}_{k-1,b}^{(i)}\}_{i=1}^{N_{k-1}^b}$ are particle systems for persistent and newborn targets, respectively and N_{k-1}^p and N_{k-1}^b are the number of persistent target and newborn target particles, respectively. Before we apply the prediction step of the PHD or the CPHD filter, according to Eq. (4.24), case $\beta = 0$, we need to sum-up the two intensity functions, $D_{k-1|k-1}(\mathbf{y}, 0)$ and $D_{k-1|k-1}(\mathbf{y}, 1)$. This summation is carried out by taking the union of the two particle sets, that is

$$\{w_{k-1}^{(i)}, \mathbf{y}_{k-1}^{(i)}\}_{i=1}^{N_k} = \{w_{k-1,p}^{(i)}, \mathbf{y}_{k-1,p}^{(i)}\}_{i=1}^{N_{k-1}^p} \cup \{w_{k-1,b}^{(i)}, \mathbf{y}_{k-1,b}^{(i)}\}_{i=1}^{N_{k-1}^b} \quad (4.38)$$

is the particle system approximating the sum $D_{k-1|k-1}(\mathbf{y}, 0) + D_{k-1|k-1}(\mathbf{y}, 1)$. Then the predicted intensity function $D_{k|k-1}(\mathbf{y}, 0)$ can be expressed as:

$$D_{k|k-1}(\mathbf{y}, 0) \approx \sum_{i=1}^{N_k} w_{k|k-1,p}^{(i)} \delta_{\mathbf{y}_{k|k-1,p}^{(i)}}(\mathbf{y}), \quad (4.39)$$

where according to (4.24) for the $\beta = 0$ case:

$$\mathbf{y}_{k|k-1,p}^{(i)} \sim q_k(\cdot | \mathbf{y}_{k-1}^{(i)}, \mathbf{Z}_k) \quad (4.40)$$

$$w_{k|k-1,p}^{(i)} = \frac{p_S(\mathbf{y}_{k-1}^{(i)}) \pi_{k|k-1}(\mathbf{y}_{k|k-1,p}^{(i)} | \mathbf{y}_{k-1}^{(i)}) w_{k-1}^{(i)}}{q_k(\mathbf{y}_{k|k-1,p}^{(i)} | \mathbf{y}_{k-1}^{(i)}, \mathbf{Z}_k)} \quad (4.41)$$

with $q_k(\cdot | \mathbf{y}_{k-1}^{(i)}, \mathbf{Z}_k)$ being the importance density [7]. For simplicity of presentation we will sacrifice the efficiency of the SMC implementation and will adopt $q_k(\cdot | \mathbf{y}_{k-1}^{(i)}, \mathbf{Z}_k) = \pi_{k|k-1}(\cdot | \mathbf{y}_{k-1}^{(i)})$.

The $\beta = 1$ case in (4.24) is not implemented in a straightforward manner, because a massive number of particles would be required to approximate $D_{k|k-1}(\mathbf{y}, 1) = \gamma_{k|k-1}(\mathbf{y})$, of which the vast majority would be thrown away in the resampling step (to be carried out in the PHD/CPHD update). Instead, the current measurement set \mathbf{Z}_k is used to place the newborn particles in the region of the state space where the inner product $\langle g(\mathbf{z} | \cdot), \gamma_{k|k-1} \rangle$ will have non-zero values. Thus for each $\mathbf{z} \in \mathbf{Z}_k$, a set of M_b newborn particles $\mathbf{y}_{k|k-1,b}^{(i)}$ are generated in such a manner that \mathbf{z} can be considered as a random sample from the PDF $g_k(\cdot | \mathbf{y}_{k|k-1,b}^{(i)})$. This newborn target density, denoted by $b_k(\cdot | \mathbf{z})$ in line 14 of Algorithm 3, in practice can be approximated as follows. Suppose the target state vector \mathbf{y} consists of directly measured vector component \mathbf{p} and unobserved vector component \mathbf{d} , that is $\mathbf{y} = [\mathbf{p}^\top \ \mathbf{d}^\top]^\top$. Let the measurement equation be: $\mathbf{z} = \mathbf{h}(\mathbf{p}) + \mathbf{w}$, where function \mathbf{h} has the inverse \mathbf{h}^{-1} . Assume that $\mathbf{w} \sim \mathcal{N}(\mathbf{w}; \mathbf{0}, \mathbf{R})$ is zero-mean white Gaussian measurement noise with covariance \mathbf{R} . Then, particles $\mathbf{p}^{(i)}$, from the measured subspace of the target state space, can be drawn from $\mathcal{N}(\mathbf{y}; \mathbf{h}^{-1}(\mathbf{z}), \mathbf{H}_* \mathbf{R} \mathbf{H}_*^\top)$, where \mathbf{H}_* is the Jacobian of \mathbf{h}^{-1} . This, of course, is an approximation since \mathbf{h} is in general a nonlinear function [7]. The particles $\mathbf{d}^{(i)}$ from the unobserved subspace need to be drawn from the prior.

The total number of newborn target particles generated in this way is $N_k^b = M_b \cdot |\mathbf{Z}_k|$. The weights of the new target particles are uniform, i.e.

$$w_{k|k-1,b}^{(i)} = \frac{v_{k|k-1}^b}{N_k^b} \quad (4.42)$$

where $v_{k|k-1}^b$ is the predicted expected number of target births. The choice of this parameter of the PHD/CPHD filter is discussed later. In summary, the PHD filter birth intensity is modelled by an equally weighted mixture of birth densities $b_k(\cdot | \mathbf{z})$, for $\mathbf{z} \in \mathbf{Z}_k$, multiplied by the expected number of births $v_{k|k-1}^b$ (see lines 10 to 17 of Algorithm 6).

Consider next the update step of the PHD filter. Observe that upon the substitution of (4.39) into (4.28), $D_{k|k}(\mathbf{y}, 0)$ can be also written as the weighted sum of particles

$$D_{k|k}(\mathbf{y}, 0) \approx \sum_{i=1}^{N_k} w_{k|k,p}^{(i)} \delta_{\mathbf{y}_{k|k-1,p}^{(i)}}(\mathbf{y}), \quad (4.43)$$

where

$$w_{k|k,p}^{(i)} = (1 - p_{\text{D}}(\mathbf{y}_{k|k-1,p}^{(i)})) w_{k|k-1,p}^{(i)} + \sum_{\mathbf{z} \in \mathbf{Z}_k} \frac{p_{\text{D}}(\mathbf{y}_{k|k-1,p}^{(i)}) g_k(\mathbf{z} | \mathbf{y}_{k|k-1,p}^{(i)}) w_{k|k-1,p}^{(i)}}{\mathcal{L}(\mathbf{z})} \quad (4.44)$$

and

$$\mathcal{L}(\mathbf{z}) = \kappa_k(\mathbf{z}) + \sum_{i=1}^{N_k^b} w_{k|k-1,b}^{(i)} + \sum_{i=1}^{N_k} p_{\text{D}}(\mathbf{y}_{k|k-1,p}^{(i)}) g_k(\mathbf{z} | \mathbf{y}_{k|k-1,p}^{(i)}) w_{k|k-1,p}^{(i)} \quad (4.45)$$

Similarly it can be shown using (4.29) that $D_{k|k}(\mathbf{y}, 1)$ is approximated by the particle system $\{w_{k|k,b}^{(i)}, \mathbf{y}_{k|k-1,b}^{(i)}\}_{i=1}^{N_k^b}$ where:

$$w_{k|k,b}^{(i)} = \sum_{\mathbf{z} \in \mathbf{Z}_k} \frac{w_{k|k-1,b}^{(i)}}{\mathcal{L}(\mathbf{z})} \quad (4.46)$$

The particle set approximating the intensity function of persistent objects, $D_{k|k}(\mathbf{y}, 0)$, is next resampled N_k^p times in order to eliminate the samples with small weights and multiply the samples with large weights. The number of particles N_k^p is selected as:

$$N_k^p = \left\lfloor M_p \cdot \sum_{n=1}^{N_k} w_{k|k,p}^{(n)} \right\rfloor \quad (4.47)$$

where $\lfloor \cdot \rfloor$ denotes the nearest integer, M_p is the number of particles per persistent object (a parameter of the filter). Let us introduce $\hat{v}_k^p = \sum_{i=1}^{N_k} w_{k|k,p}^{(i)}$ which represents the posterior estimate of the number of persistent targets. After resampling the intensity function $D_{k|k}(\mathbf{y}, 0)$ is approximated by:

$$D_{k|k}(\mathbf{y}, 0) \approx \sum_{i=1}^{N_k^p} w_{k|k,p}^{(i)} \delta_{\mathbf{y}_{k,p}^{(i)}}(\mathbf{y}) \quad (4.48)$$

where $w_{k,p}^{(i)} = \hat{v}_k^p / N_k^p$. The PHD filter reports at time k only the intensity function of persistent targets $D_{k|k}(\mathbf{y}, 0)$.

The particle set approximating the intensity function of newborn objects, $D_{k|k}(\mathbf{y}, 1)$, can be also resampled, although this is not essential (line 23 in Algorithm 6). The value of $v_{k|k-1}^b$ in (4.42) has to be chosen in such a manner that the sum $\hat{v}_k^b = \sum_{i=1}^{N_k^b} w_{k|k,b}^{(i)}$ corresponds to the expected number of newborn objects at time k .

Algorithm 6 Pseudo-code of the PHD particle filter

- 1: **Input:**
 - 2: a) Particle representation of $D_{k-1|k-1}(\mathbf{y}, 0)$: $\{w_{k-1,p}^{(i)}, \mathbf{y}_{k-1,p}^{(i)}\}_{i=1}^{N_{k-1}^p}$ ▷ Eq. (4.36)
 - 3: b) Particle representation of $D_{k-1|k-1}(\mathbf{y}, 1)$: $\{w_{k-1,b}^{(i)}, \mathbf{y}_{k-1,b}^{(i)}\}_{i=1}^{N_{k-1}^b}$ ▷ Eq. (4.37)
 - 4: c) Measurement set at k : $\mathbf{Z}_k = \{\mathbf{z}_{k,1}, \dots, \mathbf{z}_{k,m_k}\}$; Note: $m_k = |\mathbf{Z}_k|$
 - 5: $\{w_{k-1}^{(i)}, \mathbf{y}_{k-1}^{(i)}\}_{i=1}^{N_k} = \{w_{k-1,p}^{(i)}, \mathbf{y}_{k-1,p}^{(i)}\}_{i=1}^{N_{k-1}^p} \cup \{w_{k-1,b}^{(i)}, \mathbf{y}_{k-1,b}^{(i)}\}_{i=1}^{N_{k-1}^b}$ ▷ Eq. (4.38)
 - 6: **for** $i = 1, \dots, N_k$ **do** ▷ Random sample approximation of $D_{k|k-1}(\mathbf{y}, 0)$
 - 7: Draw $\mathbf{y}_{k|k-1,p}^{(i)} \sim \pi_{k|k-1}(\cdot | \mathbf{y}_{k-1}^{(i)})$ ▷ Eq. (4.40)
 - 8: Compute weight $w_{k|k-1,p}^{(i)} = p_S(\mathbf{y}_{k-1}^{(i)}) w_{k-1}^{(i)}$ ▷ Eq. (4.41)
 - 9: **end for**
 - 10: $N_k^b = M_b \cdot m_k$ ▷ Parameter M_b : the number of particles per newborn target
 - 11: **for** $j = 1, \dots, m_k$ **do** ▷ Random sample approximation of $D_{k|k}(\mathbf{y}, 1)$
 - 12: **for** $\ell = 1, \dots, M_b$ **do**
 - 13: $n = \ell + (j - 1)M_b$
 - 14: Draw $\mathbf{y}_{k|k-1,b}^{(i)} \sim b_k(\cdot | \mathbf{z}_{k,j})$ ▷ b_k depends on $g_k(\mathbf{z}|\mathbf{y})$ and prior knowledge
 - 15: Compute weight $w_{k|k-1,b}^{(i)} = v_{k|k-1}^b / N_k^b$ ▷ Parameter $v_{k|k-1}^b$
 - 16: **end for**
 - 17: **end for**
 - 18: Compute $w_{k|k,p}^{(i)}$, $i = 1, \dots, N_{k-1}$ using Eq. (4.44)
 - 19: Compute $w_{k|k,b}^{(i)}$, $i = 1, \dots, N_k^b$ using Eq. (4.46)
 - 20: Compute $\hat{v}_k^p = \sum_{i=1}^{N_k} w_{k|k,p}^{(i)}$; $\hat{v}_k^b = \sum_{n=1}^{N_k^b} w_{k|k,b}^{(i)}$
 - 21: $N_k^p = [M_p \hat{v}_k^p]$ ▷ Eq. (4.47), Parameter M_p : the number of particles per persistent target
 - 22: Resample N_k^p times from $\{w_{k|k,p}^{(i)} / \hat{v}_k^p, \mathbf{y}_{k|k-1,p}^{(i)}\}_{i=1}^{N_k}$ to obtain $\{w_{k,p}^{(i)}, \mathbf{y}_{k,p}^{(i)}\}_{i=1}^{N_k^p}$, with $w_{k,p}^{(i)} = \hat{v}_k^p / N_k^p$.
 - 23: Resample N_k^b times from $\{w_{k|k,b}^{(i)} / \hat{v}_k^b, \mathbf{y}_{k|k-1,b}^{(i)}\}_{i=1}^{N_k^b}$ to obtain $\{w_{k,b}^{(i)}, \mathbf{y}_{k,b}^{(i)}\}_{i=1}^{N_k^b}$, with $w_{k,b}^{(i)} = \hat{v}_k^b / N_k^b$.
 - 24: **Report:** particle representation of $D_{k|k}(\mathbf{y}, 0)$: $\{w_{k,p}^{(i)}, \mathbf{y}_{k,p}^{(i)}\}_{i=1}^{N_k^p}$; cardinality estimate \hat{v}_k^p .
-

In the implementation of the CPHD particle filter it is necessary to propagate (predict and update) the cardinality distribution (see Algorithm 7 for details). The prediction step of the CPHD particle filter is straightforward. The update step first requires to compute three inner products in (4.32) and (4.31). Using the particle representation of intensity functions $D_{k|k-1}(\mathbf{y}, 0)$ and $D_{k|k-1}(\mathbf{y}, 1)$, we obtain the following approximations:

$$\langle 1 - p_D(\cdot), D_{k|k-1}(\cdot, 0) \rangle \approx \sum_{i=1}^{N_{k-1}} (1 - p_D(\mathbf{y}_{k|k-1,p}^{(i)})) w_{k|k-1,p}^{(i)} \quad (4.49)$$

$$\langle 1, D_{k|k-1}(\cdot, 1) + D_{k|k-1}(\cdot, 0) \rangle \approx \sum_{i=1}^{N_{k-1}} w_{k|k-1,p}^{(i)} + \sum_{i=1}^{N_k^b} w_{k|k-1,b}^{(i)} \quad (4.50)$$

$$\begin{aligned} & \langle D_{k|k-1}(\cdot, 1) + p_{\text{D}}(\cdot) D_{k|k-1}(\cdot, 0), g(\mathbf{z}|\cdot) \rangle \\ & \approx \sum_{i=1}^{N_{k-1}} p_{\text{D}}(\mathbf{y}_{k|k-1,p}^{(i)}) g_k(\mathbf{z}|\mathbf{y}_{k|k-1,p}^{(i)}) w_{k|k-1,p}^{(i)} + \sum_{i=1}^{N_k^b} w_{k|k-1,b}^{(i)} \end{aligned} \quad (4.51)$$

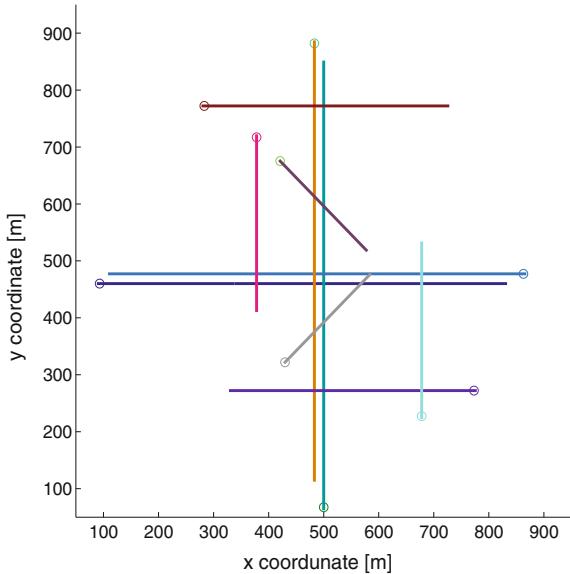
The computation of ESFs $e_j(\mathcal{E}_k(D_{k|k-1}, \mathbf{Z}))$, required in (4.31), is described in Vo et al. [20].

Algorithm 7 Pseudo-code of the CPHD particle filter

- 1: **Input:**
 - 2: a) Cardinality distribution $\rho_{k-1|k-1}(n)$
 - 3: b) particle representation of $D_{k-1|k-1}(\mathbf{y}, 0)$: $\{w_{k-1,p}^{(i)}, \mathbf{y}_{k-1,p}^{(i)}\}_{i=1}^{N_{k-1}^p}$
 - 4: c) Particle representation of $D_{k-1|k-1}(\mathbf{y}, 1)$: $\{w_{k-1,b}^{(i)}, \mathbf{y}_{k-1,b}^{(i)}\}_{i=1}^{N_{k-1}^b}$
 - 5: d) Measurement set at k : $\mathbf{Z}_k = \{\mathbf{z}_{k,1}, \dots, \mathbf{z}_{k,m_k}\}$;
 - 6: Prediction of cardinality distribution, Eq. (2.53);
 - 7: Union of input particle sets, Eq. (4.38)
 - 8: Prediction of persistent target particles (lines 6–9 in Alg.1)
 - 9: Creation of newborn target particles (lines 10–17 in Alg.1)
 - 10: Computation of elements in $\mathcal{E}_k(D_{k|k-1}, \mathbf{Z}_k)$, Eq. (4.32)
 - 11: Computation of ESFs, $e_j(\mathcal{E}_k(D_{k|k-1}, \mathbf{Z}_k))$
 - 12: Computation of $\Upsilon_k^u[D_{k|k-1}, \mathbf{Z}_k](n)$, Eq. (4.31), for $u = 1, 0$
 - 13: **for** every $\mathbf{z} \in \mathbf{Z}_k$ **do**
 - 14: Compute $\mathcal{E}_k(D_{k|k-1}, \mathbf{Z}_k \setminus \{\mathbf{z}\})$, $e_j(\mathcal{E}_k(D_{k|k-1}, \mathbf{Z}_k \setminus \{\mathbf{z}\}))$, $\Upsilon_k^1[D_{k|k-1}, \mathbf{Z}_k \setminus \{\mathbf{z}\}](n)$
 - 15: **end for**
 - 16: Update cardinality distribution, Eq. (2.55)
 - 17: Update weights $w_{k|k,p}^{(i)}$, $i = 1, \dots, N_{k-1}$ using Eq. (4.30), case $\beta = 0$
 - 18: Update weights $w_{k|k,b}^{(i)}$, $i = 1, \dots, N_k^b$ using Eq. (4.30), case $\beta = 1$
 - 19: Compute $\hat{v}_k^p = \sum_{i=1}^{N_{k-1}} w_{k|k,p}^{(i)}$; $\hat{v}_k^b = \sum_{i=1}^{N_k^b} w_{k|k,b}^{(i)}$
 - 20: $N_k^p = \lceil M_p \hat{v}_k^p \rceil$
 - 21: Resample N_k^p times from $\{w_{k|k,p}^{(i)} / \hat{v}_k^p, \mathbf{y}_{k|k-1,p}^{(i)}\}_{i=1}^{N_{k-1}}$ to obtain $\{w_{k,p}^{(i)}, \mathbf{y}_{k,p}^{(i)}\}_{i=1}^{N_k^p}$, with $w_{k,p}^{(i)} = \hat{v}_k^p / N_k^p$.
 - 22: Resample N_k^b times from $\{w_{k|k,b}^{(i)} / \hat{v}_k^b, \mathbf{y}_{k|k-1,b}^{(i)}\}_{i=1}^{N_k^b}$ to obtain $\{w_{k,b}^{(i)}, \mathbf{y}_{k,b}^{(i)}\}_{i=1}^{N_k^b}$, with $w_{k,b}^{(i)} = \hat{v}_k^b / N_k^b$.
 - 23: **Report:**
 - 24: a) Estimated cardinality distribution $\rho_{k|k}(n)$;
 - 25: b) particle representation of $D_{k|k}(\mathbf{y}, 0)$: $\{w_{k,p}^{(i)}, \mathbf{y}_{k,p}^{(i)}\}_{i=1}^{N_k^p}$
-

In summary, Algorithms 6 and 7 provide explicit Monte Carlo estimates of intensity functions for persistent targets and target-birth process separately. Algorithm 7, in addition computes an estimate of cardinality distribution.

Fig. 4.4 Target trajectories shown in $x - y$ plane. Starting points denoted by circle(\circ)



4.2.4 A Numerical Study

4.2.4.1 Simulation Setup

The performance of the described PHD and CPHD particle filters with measurement-driven target birth intensity is demonstrated using a scenario involving up to ten targets. The top down view of target trajectories is shown in Fig. 4.4. The starting point of each trajectory is indicated by sign \circ . The state vector of an individual target consists of a position and velocity in x and y coordinates, that is $\mathbf{y} = [x, \dot{x}, y, \dot{y}]^\top$. The transitional density is $\pi_{k|k-1}(\mathbf{y}|\mathbf{y}') = \mathcal{N}(\mathbf{y}; \mathbf{F}\mathbf{y}', \mathbf{Q})$. Matrices \mathbf{F} and \mathbf{Q} were specified in (4.16). In simulations the sampling period was set to $T = 1$ s, while the PHD/CPHD filters used parameters $\varpi = 0.03$ and $p_s = 0.98$.

The measurements of target range and bearing are collected by a sensor placed at $(x_s, y_s) = (-100 \text{ m}, -100 \text{ m})$. The measurement likelihood is then $g_k(\mathbf{z}|\mathbf{y}) = \mathcal{N}(\mathbf{z}; \mathbf{h}(\mathbf{y}), \mathbf{R})$, where

$$\mathbf{h}(\mathbf{y}) = \begin{pmatrix} \sqrt{(x - x_s)^2 + (y - y_s)^2} \\ \arctan \frac{x - x_s}{y - y_s} \end{pmatrix}. \quad (4.52)$$

and $\mathbf{R} = \text{diag}(\sigma_r^2, \sigma_b^2)$. The range measurements are very precise ($\sigma_r = 0.1 \text{ m}$) while the bearing measurements are fairly inaccurate (i.e. $\sigma_b = 2^\circ$). Consequently the measurement uncertainty regions are arcs with $\pm 3\sigma$ angular span of 12° . Kalman type filters, which assume the measurement uncertainty regions to be ellipsoids, would

be inappropriate in this case, thus justifying the need for an SMC implementation. The clutter is uniformly distributed along the range (from 0 to 1,300 m) and bearing ($\pm\pi/4$ rad with respect to the sensor pointing direction). The number of clutter points per scan is Poisson distributed with the mean value of $\lambda = 10$. The probability of detection is $p_d = 0.95$. For every measurement $\mathbf{z} \in \mathbf{Z}_k$, the newborn target particle are placed at:

$$\begin{aligned}x_k^{(i)} &= x_s + (\mathbf{z}[1] + \sigma_r v_1^{(i)}) \sin(\mathbf{z}[2] + \sigma_b v_2^{(i)}) \\y_k^{(i)} &= y_s + (\mathbf{z}[1] + \sigma_r v_1^{(i)}) \cos(\mathbf{z}[2] + \sigma_b v_2^{(i)})\end{aligned}$$

where $i = 1, \dots, N_k^b$, $\mathbf{z}[1]$ and $\mathbf{z}[2]$ are the measured range and bearing, respectively, and $v_1^{(i)}, v_2^{(i)} \sim \mathcal{N}(\cdot; 0, 1)$. The particle velocities are generated as $\dot{x}_k^{(i)}, \dot{y}_k^{(i)} \sim \mathcal{N}(\cdot; 0, \sigma_v^2)$, where $\sigma_v = 5$ m/s.

In both PHD and CPHD filters $M_b = M_p = 3,000$ particles. The parameter $v_{k|k-1}^b = 0.0001$ is selected so that the average number of newborn targets per scan is $\hat{v}_k^b \approx 0.25$. The cardinality distribution of newborn targets is assumed Poisson. Parameter v_{\max} in the CPHD filter is set to 30.

4.2.4.2 Error Performance Analysis

The performance of the PHD and CPHD particle filters is measured by two methods. The first method compares the true and the estimated cardinality value over time, that is n_k and \hat{n}_k^p , respectively. The second method measures the concentration of particles around the true target positions. In comparisons, we consider three contesting filters: (a) PHDF-M: the described PHD filter with the measurement driven birth intensity; (b) PHDF-U: the PHD filter which places M_b newborn target particles as if they result from fictitious measurements uniformly distributed across range (from 0 to 1,300 m) and bearing ($\pm\pi/4$ rad with respect to the sensor pointing direction); the velocities are generated as $\dot{x}_k^{(i)}, \dot{y}_k^{(i)} \sim \mathcal{N}(\cdot; 0, \sigma_v^2)$, with $\sigma_v = 5$ m/s; (c) CPHDF-M: the described CPHD filter with the measurement driven birth intensity. In order to make the comparison fair, for all three filters, an equal value of the average number of newborn targets per scan is adopted ($\hat{v}_k^b \approx 0.25$).

Cardinality Estimation

Figure 4.5 shows the true and estimated posterior cardinality value (the number of targets) over time obtained using the three contesting filters. The estimated cardinality curves were obtained by averaging over 100 Monte Carlo runs. The lines are as follows: the true number of targets is plotted by a thin black line, the mean estimated number of targets by a thick black line and the plus/minus one standard deviation of the estimated number of targets by gray lines. Observe that the two PHD filters

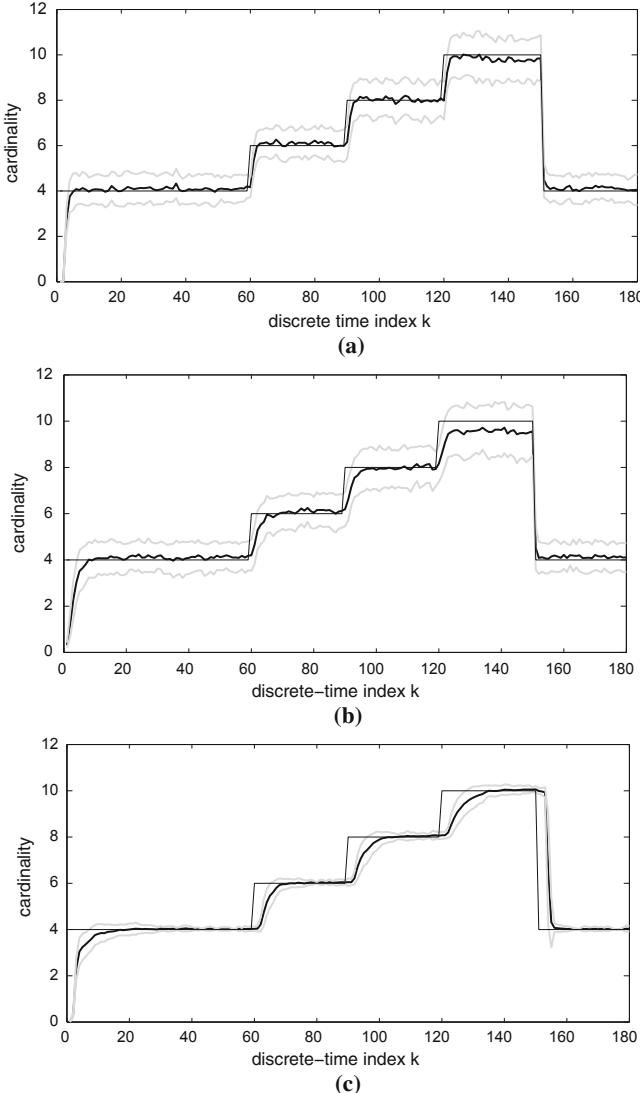


Fig. 4.5 True versus estimated cardinality over time: **a** PHDF-M; **b** PHDF-U; **c** CPHDF-M. *Explanation:* The thin black line is the true cardinality; thick black line is the average estimated cardinality; gray lines are ± 1 standard deviation of the estimate

(PHDF-M and PHDF-U) perform similarly with respect to the cardinality estimation. There are, however, two noticeable differences: (a) the PHDF-U underestimates cardinality in the presence of 10 targets and (b) the PHD-M responds slightly quicker to the changes in the target number. Regarding the comparison of the PHDF-M versus the CPHDF-M, the results are in a good agreement with those reported Vo

et al. in [20]: both filters produce unbiased cardinality estimates in the steady-state, but the PHD cardinality estimate is more responsive to the changes in cardinality. The CPHDF-M cardinality estimate, on the other hand, is characterized by a much smaller variance.

Spatial Distribution Estimation

Next we want to measure the concentration of particles around true target locations in the state-space. For this purpose we need to introduce the posterior spatial distribution of persistent targets, $s_{k|k}(\mathbf{y}, \beta = 0 | \mathbf{Z}_{1:k}) \stackrel{\text{abbr}}{=} s_{k|k}(\mathbf{y})$ defined as:

$$s_{k|k}(\mathbf{y}) = \frac{D_{k|k}(\mathbf{y}, 0)}{\int D_{k|k}(\mathbf{y}, 0) d\mathbf{y}} \quad (4.53)$$

The SMC estimate of the PHD of persisting targets, $D_{k|k}(\mathbf{y}, 0)$, as reported by the PHD or CPHD filter, was given in (4.48). The SMC estimate of $s_{k|k}(\mathbf{y})$ of (4.53) is then:

$$\hat{s}_{k|k}(\mathbf{y}) = \sum_{i=1}^{N_k^p} \tilde{w}_{k,p}^{(i)} \cdot \delta_{\mathbf{y}_{k,p}^{(i)}}(\mathbf{y}) \quad (4.54)$$

where

$$\tilde{w}_{k,p}^{(i)} = \frac{w_{k,p}^{(i)}}{\sum_{i=1}^{N_k^p} w_{k,p}^{(i)}} \quad (4.55)$$

are normalized weights [as opposed to PHD weights $w_{k,p}^{(i)}$ introduced in (4.48)].

The spatial distribution of the “ground truth” is defined as

$$s_k(\mathbf{y}) = \frac{1}{n_k} \sum_{i=1}^{n_k} \delta_{\mathbf{y}_{k,i}}(\mathbf{y}) \quad (4.56)$$

where $\{\mathbf{y}_{k,1}, \dots, \mathbf{y}_{k,n_k}\}$ is the true multi-target state at time k .

The similarity between the ground truth $s_k(\mathbf{y})$ of (4.56) and the PHD/CPHD filter estimate $\hat{s}_{k|k}(\mathbf{y})$ of (4.54), will be measured using the Bhattacharyya distance [16]:

$$B(s_k, \hat{s}_{k|k}) = -\ln \int \sqrt{s_k(\mathbf{y})} \sqrt{\hat{s}_{k|k}(\mathbf{y})} d\mathbf{y} \quad (4.57)$$

Its properties are as follows. Clearly $B(s, u) \geq 0$. Identity and symmetry properties are satisfied, i.e. $B(s, s) = 0$ and $B(s, u) = B(u, s)$, respectively. The triangle inequality, however, is not guaranteed.

In order to compute the integral in (4.57), it is first necessary to select a common set of support points for both s_k and $\hat{s}_{k|k}$. Let the common set of support points be the true multi-target state $\{\mathbf{y}_{k,1}, \dots, \mathbf{y}_{k,n_k}\}$. Then we need to determine the values of $\hat{s}_{k|k}(\mathbf{y})$ at points $\mathbf{y}_{k,i}$, for $i = 1, \dots, n_k$. Using the kernel density estimation (KDE) method [30], we have:

$$\hat{s}_{k|k}(\mathbf{y}_{k,i}) = Q_{k,i} \approx \frac{1}{W^{n_x}} \sum_{i=1}^{N_k^p} \tilde{w}_{k,p}^{(i)} \cdot \varphi\left(\frac{\mathbf{y}_{k,i} - \mathbf{y}_{k,p}^{(i)}}{W}\right) \quad (4.58)$$

where W is the kernel width, $n_x = 4$, and φ is the kernel function, adopted here to be Gaussian (the expression for the optimal W is given in Sect. 3.1.1.2, see also [30]).

Following the arguments presented in [17, Appendix], the substitution of $s_k(\mathbf{y})$ of (4.56) and

$$\hat{s}_{k|k}(\mathbf{y}) \approx \sum_{i=1}^{n_k} Q_{k,i} \delta_{\mathbf{y}_{k,i}}(\mathbf{y})$$

into (4.57) leads to expression:

$$B(s_k, \hat{s}_{k|k}) \approx -\ln\left(\sum_{i=1}^{n_k} \sqrt{\frac{Q_{k,i}}{n_k}}\right). \quad (4.59)$$

This distance measure will be used to evaluate the spatial distribution accuracy of PHD and CPHD particle filter implementations.

Figure 4.6 displays the Bhattacharyya distance (4.59) averaged over 100 Monte Carlo runs, for all three contesting filters: PHDF-M, PHDF-U and CPHDF-M. Observe first that the proposed PHD/CPHD filters with measurement driven birth intensity (PHDF-M and CPHDF-M) result in vastly more accurate approximations of the true spatial density than the PHDF-U. This indicates that PHDF-M and CPHDF-M position their particles much closer to the true target position in the state space. This result confirms the importance of the measurement driven approach to birth intensity in PHD/CPHD filters.

With regards to the comparison between the PHDF-M and CPHDF-M, note that in the steady-state the Bhattacharyya distance for the CPHD filter is slightly lower. This is a significant observation since it shows that the CPHD filter, in the steady state, not only provides more stable (i.e. less variable) cardinality estimates, but also more accurate spatial density estimates.

4.2.5 State Estimation from PHD/CPHD Particle Filters

In this section the problem we is how to compute the multi-target state estimate $\hat{\mathbf{y}}_{k,1}, \dots, \hat{\mathbf{y}}_{k,\hat{v}_k}$ from the PHD/CPHD particle filter. In order to illustrate the diffi-

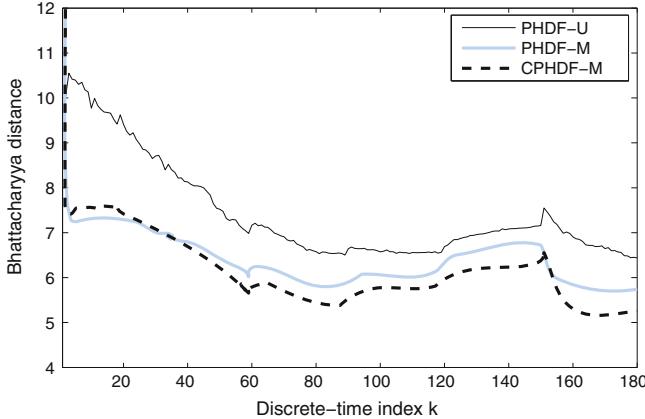


Fig. 4.6 Average Bhattacharyya distance between the true and the estimated spatial density over time: PHDF-M (gray thick line); PHDF-U (black thin line); CPHDF-M filter (black dashed line)

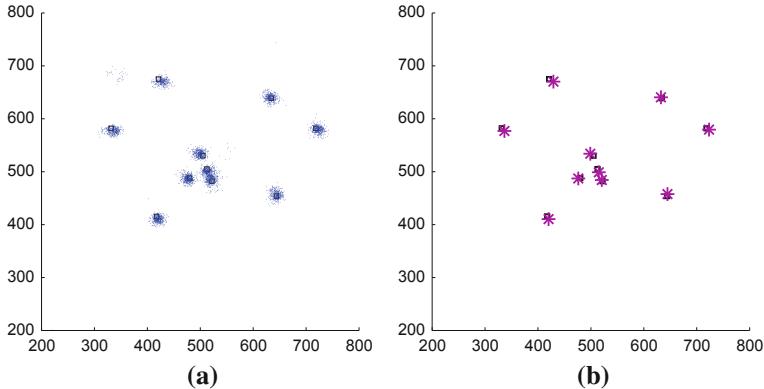


Fig. 4.7 **a** PHD particles (position only) corresponding to ten persistent objects at k ; **b** point estimates [denoted by asterisk(*)] and true object locations [denoted by square(\square)]

culty of this problem, Fig. 4.7a shows the particles corresponding to persistent objects (position only) of a PHD particle filter at time k . In this case ten targets are actually present, four of them in the close proximity to each other. The standard approach [1, 18, p. 623] is to first estimate the number of (persistent) targets as $\hat{v}_k^p = \sum_{i=1}^{N_k} w_{k|k,p}^{(i)}$ and then to group the particles corresponding to each individual target by a suitable data clustering algorithm. Finally, the mean value of each cluster of particles is an expected a posteriori estimate of each individual target. Various clustering algorithms have been attempted, such as hierarchical clustering, k-means and the EM algorithm.

Note that the clusters of particles corresponding to individual objects in general approximate non-Gaussian posterior PDFs. This means that clustering based on the

EM algorithm, for example (which assumes a model, typically Gaussian, of the underlying PDFs) is inappropriate in general. Other clustering techniques are ad-hoc and their use only spoils the elegance of the PHD filter for multi-object nonlinear filtering.

In this section we show that grouping of particles corresponding to individual objects can be done (in some cases) within the update step of the PHD/CPHD particle filter, thus eliminating the need for ad-hoc post-update particle clustering. The algorithm was first published in Ristic et al. [23].

A close inspection of (4.28) and (4.44) reveals that the update step can be interpreted as a weighted sum of $m_k + 1$ measurement hypotheses, where $m_k = |\mathbf{Z}_k|$. In the context of the PHD particle filter, the update step creates $m_k + 1$ replicas of the particle set $\{\mathbf{y}_{k|k-1}^{(i)}\}_{i=1}^{N_{k-1}}$, because the summation of $m_k + 1$ terms in (4.44) is identical to the union of replicated particle sets. Let us denote the weights corresponding to particle set replica $j = 0, 1, \dots, m_k$ by $\{w_{k|k,p}^{(i,j)}, i = 1, \dots, N_{k-1}\}$. Index $j > 0$ corresponds to the index of a measurement at time k as in $\mathbf{Z}_k = \{\mathbf{z}_{k,1}, \dots, \mathbf{z}_{k,m_k}\}$. The weights $w_{k|k,p}^{(i,j)}$, according to (4.44), can be computed as:

$$w_{k|k,p}^{(i,j)} = \begin{cases} \left[1 - p_D(\mathbf{y}_{k|k-1,p}^n)\right] \cdot w_{k|k-1,p}^{(i)}, & j = 0 \\ \frac{p_D(\mathbf{y}_{k|k-1,p}^{(i)}) \cdot g_k(\mathbf{z}_{k,j}|\mathbf{y}_{k|k-1,p}^{(i)}) \cdot w_{k|k-1,p}^{(i)}}{\mathcal{L}(\mathbf{z}_{k,j})}, & j = 1, \dots, m_k \end{cases} \quad (4.60)$$

with $\mathcal{L}(\mathbf{z}_{k,j})$ given by (4.45).

The sum $W_{k,p}^{(j)} = \sum_{i=1}^{N_{k-1}} w_{k|k,p}^{(i,j)}$ represents the total weight assigned to the particle set replica $j = 1, \dots, m_k$. Note from (4.28) that

$$W_{k,p}^{(j)} \approx \frac{\langle p_D g_k(\mathbf{z}_{k,j}|\cdot), D_{k|k-1}(\cdot, 0) \rangle}{\kappa_k(\mathbf{z}_{k,j}) + \langle g_k(\mathbf{z}_{k,j}|\cdot), \gamma_{k|k-1} \rangle + \langle p_D g_k(\mathbf{z}_{k,j}|\cdot), D_{k|k-1}(\cdot, 0) \rangle} \leq 1. \quad (4.61)$$

Hence $W_{k,p}^{(j)}$ can be interpreted as the probability that measurement $\mathbf{z}_{k,j}$ is due to a persistent target.

The state estimate and its covariance matrix at time k , that is $\hat{\mathbf{y}}_{k,j}$ and $\mathbf{P}_{k,j}$, respectively, can be computed as:

$$\hat{\mathbf{y}}_{k,j} = \sum_{i=1}^{N_{k-1}} w_{k|k,p}^{(i,j)} \mathbf{y}_{k|k-1,p}^{(i)} \quad (4.62)$$

$$\mathbf{P}_{k,j} = \sum_{i=1}^{N_{k-1}} w_{k|k,p}^{(i,j)} \left(\mathbf{y}_{k|k-1,p}^{(i)} - \hat{\mathbf{y}}_{k,j} \right) \left(\mathbf{y}_{k|k-1,p}^{(i)} - \hat{\mathbf{y}}_{k,j} \right)^T \quad (4.63)$$

for $j = 1, \dots, m_k$. This method does not require any clustering of the particles—the grouping of particles is implicitly taken care of by the weights. Each pair $(\hat{\mathbf{y}}_{k,j}, \mathbf{P}_{k,j})$

is assigned the probability $W_{k,p}^{(j)}$. Only if the value of $W_{k,p}^{(j)}$ is above a certain threshold value, the state estimate $\hat{\mathbf{y}}_{k,j}$ and its covariance $\mathbf{P}_{k,j}$ are reported. Figure 4.7b illustrates the state estimates obtained using the described method. The reporting threshold for the weights $W_{k,p}^{(j)}$ in this example was set to 0.5.

If we want to find the particle approximation of the posterior PDF for a particular target, we then need to resample the particles $\{\mathbf{y}_{k|k-1,p}^{(i)}; i = 1, \dots, N_{k-1}\}$ using the weights $\{w_{k|k,p}^{(i,j)}, i = 1, \dots, N_{k-1}\}$ computed using the measurement $\mathbf{z}_{k,j}$. This particle approximation of the posterior PDF may be important in the context of nonlinear/non-Gaussian estimation.

Two remarks are in order here in relation to the proposed state estimation scheme. First, note that only targets that have been detected in scan k have a chance to be reported at scan k . This is in agreement with the lack of “memory” or inertia of the PHD filter, a characteristic discussed in Refs. [1, (Sect. 16.6)], [24]. Second, the proposed scheme may not be applicable in all cases. Consider for example the bearing-only multi-target filtering and two targets on the same bearings angle. Using (4.62) the two target originated bearings measurement will produce two estimates that will appear midway between the two targets.

4.2.5.1 Comparison

The performance of the proposed method for state estimation is compared to the post-update step clustering method via k-means algorithm. The simulation setup was described in Sect. 4.2.4.1. The estimation error of the two methods is measured using the OSPA metric, discussed in Sect. 2.4.3, between the estimated and the true multi-target states, $\hat{\mathbf{Y}}_k$ and \mathbf{Y}_k . Recall that the parameters of OSPA metric are the *order* p (which determines the sensitivity to outliers) and the *cut-off* c (the relative weighting of the penalties assigned to cardinality and localization errors). The values adopted for OSPA parameters in this comparison are $p = 2$ and $c = 150$ m.

Figure 4.8 shows the OSPA metric for the two state estimation methods from PHD particle filters, averaged over 100 Monte Carlo runs. The scenario and the parameters were the same as in Fig. 4.4, however only the segment from $k = 60$ to $k = 150$ is considered. The results indicate that the proposed method is significantly more accurate. This improvement is mainly due to the reduction in the localization error, while cardinality errors are similar.

4.3 Particle Filter Approximation of the Exact Multi-Object Filter

The prediction and update equations of the exact multi-target Bayes stochastic filter were given in (2.35–2.36), with standard multi-object likelihood specified in (2.34) and standard multi-target transitional density in (2.33). The particle filter implemen-

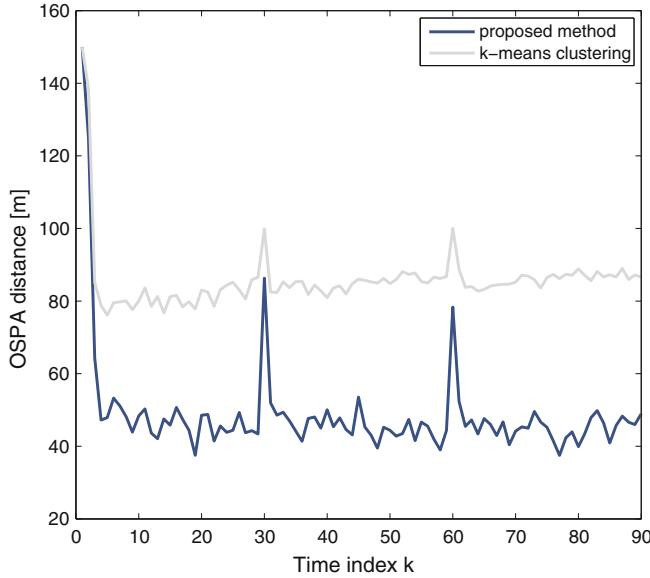


Fig. 4.8 OSPA distance (averaged over 100 Monte Carlo runs) for two methods of state estimation from the PHD particle filter

tation of this filter, referred in this monograph as the *multi-target state particle filter*, has been considered by a number of authors, e.g. [1, 21, 25, 26, 32–35]. The computational complexity of the multi-target Bayes filter grows exponentially with the number of targets and therefore all practical implementations are limited to a relatively small number of targets. An efficient importance density for persistent targets in multi-target state particle filters is proposed in Yi et al. [27] with an excellent review and comparison of existing approaches.

Let $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_m\}$ and $\mathbf{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_n\}$ be two finite sets. Then

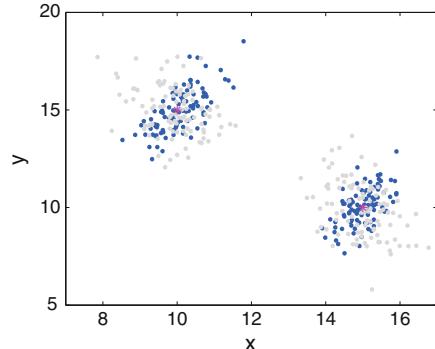
$$\delta_{\mathbf{Y}}(\mathbf{X}) = \begin{cases} 0 & m \neq n \\ 1 & m = n = 0 \\ \sum_{\pi \in \Pi_n} \delta(\mathbf{x}_1 - \mathbf{y}_{\pi(1)}) \dots \delta(\mathbf{x}_n - \mathbf{y}_{\pi(n)}) & n = m > 0 \end{cases}$$

is the multi-object Dirac delta function, where Π_n denotes the finite set of all permutations on $\{1, \dots, n\}$ [1].

The multi-target (FISST) posterior PDF $f_{k|k}(\mathbf{X}_k | \mathbf{Z}_{1:k})$ is approximated by particles as follows [1, Chap. 15]:

$$f_{k|k}(\mathbf{X}_k | \mathbf{Z}_{1:k}) \approx \sum_{i=1}^N w_k^{(i)} \delta_{\mathbf{X}_k^{(i)}}(\mathbf{X}_k) \quad (4.64)$$

Fig. 4.9 Illustration of the mixed labeling problem in multi-target particle filter. There are two targets, one is state $(10, 15)$, the other in $(15, 10)$. The *black* and *gray* dots show the first and the second target vector component (particle), respectively



where $\mathbf{X}_k^{(i)}$ is the state of the multi-target particle i and $w_k^{(i)}$ is its weight. The weights are normalized. Multi-target particle i has a variable dimension and can be expressed as:

$$\mathbf{X}_k^{(i)} = \begin{cases} [] & \text{if } |\mathbf{X}_k^{(i)}| = 0 \\ [\mathbf{x}_k^{(i)}] & \text{if } |\mathbf{X}_k^{(i)}| = 1 \\ [(\mathbf{x}_{k,1}^{(i)})^\top \ (\mathbf{x}_{k,2}^{(i)})^\top] & \text{if } |\mathbf{X}_k^{(i)}| = 2 \\ \dots & \\ [(\mathbf{x}_{k,1}^{(i)})^\top \ (\mathbf{x}_{k,2}^{(i)})^\top \ \dots \ (\mathbf{x}_{k,v_{\max}}^{(i)})^\top] & \text{if } |\mathbf{X}_k^{(i)}| = v_{\max} \end{cases} \quad (4.65)$$

where v_{\max} is a design parameter which denotes the maximum number of targets. Particle i in (4.65) is represented by a vector (for easier computer implementation), however, one has to keep in mind its permutation equivalence—the multi-target particle essentially represents a random finite set.

By ignoring the permutation equivalence the multi-target state particle filter can be affected by the *mixed labeling* problem [28]. This is illustrated in Fig. 4.9, which shows the particle approximation of the posterior $f_{k|k}(\mathbf{X}_k | \mathbf{Z}_{1:k})$ with $\mathbf{X}_k = \{\mathbf{x}_{k,1}, \mathbf{x}_{k,2}\}$. Each particle in (4.64) is a stacked vector $\mathbf{X}_k^{(i)} = [(\mathbf{x}_{k,1}^{(i)})^\top \ (\mathbf{x}_{k,2}^{(i)})^\top]$. The black and gray dots in Fig. 4.9 show the first and the second target vector component (particle), respectively. While the particle approximation of $f_{k|k}(\mathbf{X}_k | \mathbf{Z}_{1:k})$ is perfectly valid, ignoring the permutation equivalence can cause errors in computation of the point estimates. For example, the mean values in this example would appear in the middle of the two particle clouds.

A solution to the mixed labeling problem based on clustering of particles has been proposed in Kreucher et al. [29]. However, even if no action is made, it has been shown that the mixed labeling problem eventually resolves itself, thanks to the resampling of particles [28].

The pseudo-code of the Bootstrap type particle filter which implements the exact multi-target Bayes filter is given in Algorithm 8. The proposed multi-target particles at time k are constructed based on the transitional density, the survival probability and the birth distribution of new targets. The average number of newborn targets is μ_0 , with birth distribution $b_k(\mathbf{x})$ in line 14 typically designed using the measurements from time $k - 1$. The computation of the multi-object likelihood $\varphi_k(\mathbf{Z}_k | \tilde{\mathbf{X}}_k^{(i)})$ in line 17, defined in (2.34), requires to evaluate all association hypotheses θ between objects in $\tilde{\mathbf{X}}_k^{(i)}$ and detections in \mathbf{Z}_k . The number of these hypotheses grows exponentially with the number of targets. For example, if the cardinalities are: $|\tilde{\mathbf{X}}_k^{(i)}| = 2$ and $|\mathbf{Z}_k| = 3$, the number of association hypothesis is 13. This exponentially growing computational cost is the major limitation of the exact multi-target Bayes filter in practice.

Algorithm 8 Pseudo-code of the bootstrap type multi-target state particle filter

```

1: function MULTI-TARGET STATE BOOTSTRAP FILTER
2:   Input:  $\{\mathbf{X}_{k-1}^{(i)}\}_{i=1}^N, \mathbf{Z}_k$ 
3:   for  $i = 1, \dots, N$  do
4:      $\tilde{\mathbf{X}}_k^{(i)} = []$ 
5:     for  $j = 1, \dots, |\mathbf{X}_{k-1}^{(i)}|$  do
6:        $u_* \sim \mathcal{U}_{[0,1]}(u)$ 
7:       if  $u_* < p_s$  then
8:          $\mathbf{x}_* \sim \pi_{k|k-1}(\mathbf{x} | \mathbf{x}_{k,j}^{(i)})$ 
9:          $\tilde{\mathbf{X}}_k^{(i)} = [\left(\tilde{\mathbf{X}}_k^{(i)}\right)^\top \mathbf{x}_*^\top]^\top$ 
10:      end if
11:    end for
12:     $\mu \sim \text{Poisson}(\mu_0)$ 
13:    for  $j = 1, \dots, \mu$  do
14:       $\mathbf{x}_* \sim b_k(\mathbf{x})$ 
15:       $\tilde{\mathbf{X}}_k^{(i)} = [\left(\tilde{\mathbf{X}}_k^{(i)}\right)^\top \mathbf{x}_*^\top]^\top$ 
16:    end for
17:    Calculate weight:  $\tilde{w}_k^{(i)} = \varphi_k(\mathbf{Z}_k | \tilde{\mathbf{X}}_k^{(i)})$ 
18:  end for
19:   $w_k^{(i)} = \tilde{w}_k^{(i)} / \sum_{j=1}^N \tilde{w}_k^{(j)}$ , for  $i = 1, \dots, N$ 
20:  for  $i = 1, \dots, N$  do
21:    Select index  $j^i \in \{1, \dots, N\}$  with probability  $w_k^{(i)}$ 
22:     $\mathbf{X}_k^{(i)} = \tilde{\mathbf{X}}_k^{(j^i)}$ 
23:  end for
24:  Output:  $\{\mathbf{X}_k^{(i)}\}_{i=1}^N$ 
25: end function

```

Estimation of the multi-target state \mathbf{X}_k from the particle approximation of the multi-target posterior PDF $f_{k|k}(\mathbf{X}_k | \mathbf{Z}_{1:k})$ in (4.64) is discussed briefly next. The cardinality distribution can be estimated as:

$$\hat{p}_{k|k}(n|\mathbf{Z}_{1:k}) = \frac{1}{N} \sum_{i=1}^N \Delta[|\mathbf{X}_k^{(i)}|, n], \quad n = 0, 1, 2, \dots, v_{\max}, \quad (4.66)$$

where $\Delta[\ell, n]$ is the Kronecker delta function which equals 1 if $\ell = n$ and zero otherwise. Practically, for each $n = 0, 1, \dots, v_{\max}$, it is necessary to count the number of multi-target particles whose cardinality equals n , and then to divide this count with the total number of particles N . Then one can estimate the number of targets in \mathbf{X}_k , that is $\hat{n}_{k|k}$, either as the expected value or a maximum of posterior $\hat{p}_{k|k}(n|\mathbf{Z}_{1:k})$. Given $\hat{n}_{k|k}$, the multi-target state can be estimated from the particle system $\{w_k^{(i)}, \mathbf{X}_k^{(i)}\}_{i=1}^N$ as

$$\hat{X}_k = \frac{\sum_{i=1}^N w_k^{(i)} \mathbf{X}_k^{(i)} \Delta[|\mathbf{X}_k^{(i)}|, \hat{n}_k]}{\sum_{i=1}^N w_k^{(i)} \Delta[|\mathbf{X}_k^{(i)}|, \hat{n}_k]}, \quad (4.67)$$

which represents the mean of the multi-target particles characterized by cardinality $\hat{n}_{k|k}$.

The described particle filter implementation of the exact multi-target stochastic Bayes filter will be considered in Sect. 5.3, in the context of sensor control.

References

1. R. Mahler, *Statistical Multisource Multitarget Information Fusion*. Artech House, 2007.
2. B.-T. Vo, D. Clark, B.-N. Vo, B. Ristic, “Bernoulli forward-backward smoothing for joint target detection and tracking”, *IEEE Trans. Signal Processing*, vol. 59, pp. 4473–4477, Sept. 2011.
3. B. Ristic and S. Arulampalam, “Bernoulli particle filter with observer control for bearings only tracking in clutter”, *IEEE Trans Aerospace and Electronic Systems*, vol. 48, July 2012.
4. A. Doucet, J. F. G. de Freitas, and N. J. Gordon, eds., *Sequential Monte Carlo Methods in Practice*. Springer, 2001.
5. M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for non-linear/non-Gaussian Bayesian tracking”, *IEEE Trans. Signal Processing*, vol. 50, pp. 174–188, Feb. 2002.
6. P. Djuric, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. Bugallo, and J. Miguez, “Particle filtering”, *IEEE Signal Processing Magazine*, pp. 19–38, Sept. 2003.
7. B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman filter: Particle filters for tracking applications*. Artech House, 2004.
8. O. Cappé, S. J. Godsill, and E. Moulines, “An overview of existing methods and recent advances in sequential Monte Carlo”, *Proc. IEEE*, vol. 95, no. 5, pp. 899–924, 2007.
9. A. Doucet and A. M. Johansen, “A tutorial on particle filtering and smoothing: Fifteen years later”, tech. rep., Department of Statistics, University of British Columbia, Dec. 2008.
10. F. Abdallah, A. Gning, and P. Bonnifait, “Box particle filtering for nonlinear state estimation using interval analysis”, *Automatica*, vol. 44, pp. 807–815, 2008.
11. L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied Interval Analysis*. Springer, 2001.
12. G. I. Hargreaves, “Interval analysis in matlab”, tech. rep., The University of Manchester.
13. A. Gning, B. Ristic, and L. Mihaylova, “Bernoulli particle/box-particle filters for detection and tracking in the presence of triple measurement uncertainty”, *IEEE Trans. Signal Processing*, vol. 60, pp. 2138–2151, May 2012.

14. A. Gning, B. Ristic, L. Mihaylova, and A. Fahed, “Introduction to box particle filtering”, *IEEE Signal Processing Magazine*, 2012. (to appear).
15. Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, 2001.
16. T. Kailath, “The divergence and Bhattacharyya distance measures in signal selection”, *IEEE Trans. Communication technology*, vol. 15, pp. 52–60, Feb. 1967.
17. B. Ristic, B.-N. Vo, and D. Clark, “A note on the reward function for PHD filters with sensor control”, *IEEE Trans. Aerospace and Electronic Systems*, vol. 47, pp. 1521–1529, April 2011.
18. D. E. Clark, J. Bell, Y. de Saint-Pern, and Y. Petillot, “PHD filter for multi-target tracking in 3D sonar”, in *Proc. IEEE OCEANS-05-Europe*, (Brest, France), pp. 265–270, 2005.
19. B.-N. Vo and W.-K. Ma, “The Gaussian mixture probability hypothesis density filter”, *IEEE Trans. Signal Processing*, vol. 54, pp. 4091–4104, Nov. 2006.
20. B.-T. Vo, B.-N. Vo, and A. Cantoni, “Analytic implementations of the cardinalized probability hypothesis density filter”, *IEEE Trans. Signal Processing*, vol. 55, no. 7, pp. 3553–3567, 2007.
21. B.-N. Vo, S. Singh, and A. Doucet, “Sequential Monte Carlo methods for multi-target filtering with random finite sets”, *IEEE Trans. Aerospace & Electronic Systems*, vol. 41, pp. 1224–1245, Oct. 2005.
22. N. P. Whiteley, S. S. Singh, and S. J. Godsill, “Auxiliary particle implementation of the probability hypothesis density filter”, *IEEE Trans. on Aerospace & Electronic Systems*, vol. 46, pp. 1437–1454, July 2010.
23. B. Ristic, D. Clark, and B.-N. Vo, “Improved SMC implementation of the PHD filter”, in *Proc. 13th Int. Conf. Information Fusion*, (Edinburgh, UK), July 2010.
24. O. Erdinc, P. Willett, and Y. Bar-Shalom, “Probability hypothesis density filter for multitarget multisensor tracking”, in *Proc. Int. Conf. Information Fusion*, (Philadelphia, PA, USA), pp. 146–153, July 2005.
25. M. Vihola, “Rao-Blackwellised particle filtering in random set multitarget tracking”, *IEEE Trans. Aerospace & Electronic Systems*, vol. 43, no. 2, pp. 689–705, 2007.
26. S. Reuter and K. Dietmayer, “Pedestrian tracking using random finite sets”, in *Proc. Int. Conf. Information Fusion*, (Chicago, USA), July 2011.
27. W. Yi, M. Morelande, L. Kong, and J. Yang, “A computationally efficient particle filter for multi-target tracking using independence assumption”, *IEEE Trans. Signal Processing*, 2012. (In review).
28. Y. Boers, E. Sviestins, and H. Driessens, “Mixed labelling in multitarget particle filtering”, *IEEE Trans Aerospace and Electronic Systems*, vol. 46, no. 2, pp. 792–802, 2010.
29. C. Kreucher, K. Kastella, and A. O. Hero, “Multitarget tracking using the joint multi target probability density”, *IEEE Trans. on Aerospace and Electronic Systems*, vol. 41, no. 4, p. 1396–1414, 2005.
30. B. W. Silverman, *Density estimation for statistical and data analysis*. Chapman and Hall, 1986.
31. D. Schuhmacher, B.-T. Vo, and B.-N. Vo, “A consistent metric for performance evaluation of multi-object filters”, *IEEE Trans. Signal Processing*, vol. 56, pp. 3447–3457, Aug. 2008.
32. H. Sidenbladh and S. L. Wirkander, “Tracking random sets of vehicles in terrain”, in *Proc. 2nd IEEE Workshop on Multi-Object Tracking*, (Madison, WI, USA), June 2003.
33. T. Zajic and R. Mahler, “A particle-systems implementation of the PHD multitarget tracking filter”, in *Proc. SPIE*, vol. 5096, pp. 291–299, April 2003.
34. B. Ristic and B.-N. Vo, “Sensor control for multi-object state-space estimation using random finite sets”, *Automatica*, vol. 46, pp. 1812–1818, 2010.
35. B.-T. Vo and B.-N. Vo, “A random finite set conjugate prior and application to multi-target tracking”, in *Proc. IEEE Conf. ISSNIP 2011*, (Adelaide, Australia), pp. 431–436, Dec. 2011.

Chapter 5

Sensor Control for Random Set Based Particle Filters

The POMDP framework for sensor control has been introduced in Sect. 2.2. This framework is now applied for the purpose of sensor control when using random finite set stochastic filters and their sequential Monte Carlo implementations. The emphasis is on the computation of the reward function, which in the standard case has the form given by (2.10). In the context of random finite sets, the predicted PDF $p(\mathbf{x}_k | \mathbf{z}_{1:k-1}, \mathbf{u}_{0:k-2})$ and the updated PDF $p(\mathbf{x}_k | \mathbf{z}_{1:k}, \mathbf{u}_{0:k-1})$ in (2.10) need to be replaced by their FISST PDF counterparts. Accordingly, the integral in (2.10) must be replaced by the set integral defined in (2.24). Thus, the single-step ahead reward function for multi-target filtering in the context of random finite sets is adopted as follows:

$$\mathcal{J}(\mathbf{u}_{k-1}) = \frac{1}{\alpha - 1} \log \int [f_{k|k}(\mathbf{X}_k | \mathbf{Z}_{1:k}, \mathbf{u}_{0:k-1})]^\alpha [f_{k|k-1}(\mathbf{X}_k | \mathbf{Z}_{1:k-1}, \mathbf{u}_{0:k-2})]^{1-\alpha} \delta \mathbf{X} \quad (5.1)$$

Expression (5.1) will serve as the basis of the material presented in this chapter.

5.1 Bernoulli Particle Filter with Sensor Control

5.1.1 The Reward Function

We have seen that both the predicted and the updated FISST PDFs in the case of the Bernoulli filter are the Bernoulli FISST PDFs. Recall that this PDF is completely characterized by the probability of target existence and the spatial PDF. Let $f_{k|k-1}(\mathbf{X}_k | \mathbf{Z}_{1:k-1}, \mathbf{u}_{0:k-2})$ and $f_{k|k}(\mathbf{X}_k | \mathbf{Z}_{1:k}, \mathbf{u}_{0:k-1})$, which feature in (5.1), be specified by the pairs $(q_{k|k-1}, s_{k|k-1}(\mathbf{x}))$ and $(q_{k|k}, s_{k|k}(\mathbf{x}))$, respectively. Then according to (2.28) we can write:

$$f_{k|k-1}(\mathbf{X}_k | \mathbf{Z}_{1:k-1}, \mathbf{u}_{0:k-2}) = \begin{cases} 1 - q_{k|k-1} & \text{if } \mathbf{X}_k = \emptyset \\ q_{k|k-1} \cdot s_{k|k-1}(\mathbf{x}) & \text{if } \mathbf{X}_k = \{\mathbf{x}\}, \\ 0 & \text{if } |\mathbf{X}_k| \geq 2, \end{cases}$$

$$f_{k|k}(\mathbf{X}_k | \mathbf{Z}_{1:k}, \mathbf{u}_{0:k-1}) = \begin{cases} 1 - q_{k|k} & \text{if } \mathbf{X}_k = \emptyset \\ q_{k|k} \cdot s_{k|k}(\mathbf{x}) & \text{if } \mathbf{X}_k = \{\mathbf{x}\} \\ 0 & \text{if } |\mathbf{X}_k| \geq 2. \end{cases} \quad (5.2)$$

Using the definition of the set integral (2.24) it follows from (5.1) that the single-step reward function simplifies to:

$$\mathcal{J}(\mathbf{u}_{k-1}) = \frac{1}{\alpha - 1} \log \left\{ [1 - q_{k|k-1}]^{1-\alpha} [1 - q_{k|k}]^\alpha + [q_{k|k-1}]^{1-\alpha} q_{k|k}^\alpha \int [s_{k|k-1}(\mathbf{x})]^{1-\alpha} \cdot [s_{k|k}(\mathbf{x})]^\alpha d\mathbf{x} \right\} \quad (5.3)$$

According to (2.8), the optimal one-step ahead control vector is found as the expected value:

$$\mathbf{u}_{k-1} = \arg \max_{\mathbf{v} \in \mathbb{U}_{k-1}} \mathbb{E} \{ \mathcal{J}(\mathbf{v}) \} \quad (5.4)$$

Now we focus on the numerical implementation of (5.4). An obvious method for the computation of the expected value of $\mathcal{J}(\mathbf{v})$ would be to take the sample mean over the reward functions computed for an ensemble of future measurement sets sequences $Z_k(\mathbf{v})$, which would result from taking action $\mathbf{v} \in \mathbb{U}_{k-1}$. Each realization of $Z_k(\mathbf{v})$ could be generated based on the current estimate of the predicted PDF specified by $(q_{k|k-1}, s_{k|k-1}(\mathbf{x}))$ and the measurement model which includes both false and miss-detections.

In situations where the probability of detection p_D is high and the clutter is moderate, the quantities $q_{k|k}$, and $s_{k|k}(\mathbf{x})$, which feature in (5.3) can be approximated under the assumptions of the *predicted ideal measurement set* [1, Sect. 4.1]. This means that (in the absence of the actual future measurement set), we can assume ideal conditions in (2.44–2.45) and (2.47–2.48): $p_s = 1$, $p_b = 0$, $p_D = 1$, no clutter, no measurement noise, and no process noise. Then $Z_k(\mathbf{v})$ is a singleton whose only element is the target originated noiseless measurement.

The computation of integral

$$Q = \int [s_{k|k-1}(\mathbf{x})]^{1-\alpha} \cdot [s_{k|k}(\mathbf{x})]^\alpha d\mathbf{x} \quad (5.5)$$

which features in (5.3), in the context of the Bernoulli particle filter, follows the approach presented in Sect. 2.2. Let the $s_{k|k-1}(\mathbf{x})$ be approximated by the particle system $\{w_{k|k-1}^{(i)}, \mathbf{x}_{k|k-1}^{(i)}\}_{i=1}^{N+B_{k-1}}$, see for details Sect. 4.1.1 and Algorithm 3.

By taking action \mathbf{v} , one can generate a sample of a *future* measurement set $\mathbf{Z}_k(\mathbf{v})$. Given $\mathbf{Z}_k(\mathbf{v})$, density $s_{k|k}(\mathbf{x})$ is approximated by the particle system $\left\{ w_{k|k}^{(i)}, \mathbf{x}_{k|k-1}^{(i)} \right\}_{i=1}^{N+B_{k-1}}$, where $w_{k|k}^{(i)}$ are computed according to (4.8–4.9). Then according to the arguments presented in Sect. 2.2, integral (5.5) can be approximated by:

$$Q \approx \sum_{i=1}^{N+B_{k-1}} \left(w_{k|k-1}^{(i)} \right)^{1-\alpha} \cdot \left(w_{k|k}^{(i)} \right)^\alpha. \quad (5.6)$$

Integral Q depends on the measurement set $\mathbf{Z}_k(\mathbf{v})$ via $w_{k|k}^{(i)}$.

In summary, sensor control for the Bernoulli particle filter involves the computation of the reward assigned to every action $\mathbf{v} \in \mathbb{U}_{k-1}$:

$$\begin{aligned} \mathcal{J}(\mathbf{v}) = & \frac{1}{\alpha - 1} \log \left\{ [1 - q_{k|k-1}]^{1-\alpha} [1 - q_{k|k}]^\alpha + \right. \\ & \left. [q_{k|k-1}]^{1-\alpha} q_{k|k}^\alpha \sum_{i=1}^{N+B_{k-1}} \left(w_{k|k-1}^{(i)} \right)^{1-\alpha} \cdot \left(w_{k|k}^{(i)} \right)^\alpha \right\}, \end{aligned} \quad (5.7)$$

followed by the application of (5.4). The pseudo-code of the Bernoulli bootstrap-type particle filter (originally in Algorithm 3) is augmented with sensor control steps in Algorithm 9. The proposition of admissible actions (line 8) is application dependent (e.g., sensors can move to a new location or “look” in a particular direction). Lines 9 to 21 compute the expected reward for each action $\mathbf{v} \in \mathbb{U}_{k-1}$, followed by the selection of the optimal control vector in line 22. A sample of a future measurement set is drawn in line 12 from the likelihood function specified in (2.46).

5.1.2 Bearings Only Tracking in Clutter with Observer Control

The problem of autonomous bearings-only target tracking has been studied over many years due to its tremendous importance in passive surveillance. The basic problem is the sequential estimation of target location, speed, and heading, from noise corrupted measurements of target line-of-sight (LOS) bearing. Due to the nonlinear measurements, the optimal solution in the sequential Bayesian framework results in a non-Gaussian posterior PDF. The consequence of this nonlinear/non-Gaussian context is the absence of a closed form solution. Various approximate implementations of the sequential Bayesian estimator have been therefore proposed. Useful reviews, comparing various techniques for autonomous bearings-only tracking, can be found in [2–4].

In this chapter, we take a step further by considering a more realistic case of bearings-only tracking with imperfect target detection, resulting in false detections and possibly missed target detections. For this purpose we develop a dedicated Bernoulli particle filter. Moreover, in order to close the bearings-only surveillance

Algorithm 9 Pseudo-code of the Bernoulli bootstrap-type particle filter with sensor control

1: **function** BERNOUILLI BOOTSTRAP FILTER WITH SENSOR CONTROL

2: Input: $q_{k-1|k-1}$, $\{\mathbf{x}_{k-1}^{(i)}\}_{i=1}^N$, \mathbf{Z}_{k-1} , \mathbf{Z}_k

3: Draw: $\mathbf{x}_{k-1}^{(i)} \sim b_{k-1}(\mathbf{x}|\mathbf{Z}_{k-1})$ for $i = N+1, \dots, N+B_{k-1}$

4: $q_{k|k-1} = p_b \cdot (1 - q_{k-1|k-1}) + p_s \cdot q_{k-1|k-1}$ ▷ Eq. (2.44)

5: Draw a sample: $\mathbf{x}_{k|k-1}^{(i)} \sim \pi_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1}^{(i)})$ for $i = 1, \dots, N+B_{k-1}$

6: Predicted weight: $w_{k|k-1}^{(i)} = \frac{p_s q_{k-1|k-1}}{N q_{k|k-1}}$ for $i = 1, \dots, N$

7: Predicted weight: $w_{k|k-1}^{(i)} = \frac{p_b (1 - q_{k-1|k-1})}{B_{k-1} q_{k|k-1}}$ for $i = N+1, \dots, N+B_{k-1}$

8: Compute the set of admissible control vectors $\mathbb{U}_{k-1} = \{\mathbf{u}_1, \dots, \mathbf{u}_c\}$

9: **for** every $\mathbf{v} \in \mathbb{U}_{k-1}$ **do**

10: **for** $m = 1, \dots, M$ **do**

11: Select index $i_m \in \{1, \dots, N+B_{k-1}\}$ with probability $w_{k|k-1}^{(i)}$

12: Draw $\mathbf{Z}_k^{(m)} \sim \varphi_k(\mathbf{Z}|\mathbf{x}_{k|k-1}^{(i_m)})$ ▷ Eq. (2.46)

13: For every $\mathbf{z} \in \mathbf{Z}_k^{(m)}$ compute $I_k(\mathbf{z}) = \sum_{i=1}^{N+B_{k-1}} w_{k|k-1}^{(i)} \cdot g_k(\mathbf{z}|\mathbf{x}_{k|k-1}^{(i)})$

14: Compute $\delta_k \approx p_D \left(1 - \sum_{\mathbf{z} \in \mathbf{Z}_k^{(m)}} \frac{I_k(\mathbf{z})}{\lambda c(\mathbf{z})}\right)$.

15: Update existence: $q_{k|k} = \frac{1 - \delta_k}{1 - \delta_k \cdot q_{k|k-1}} \cdot q_{k|k-1}$ ▷ Eq. (2.47)

16: Update weights: $\tilde{w}_{k|k}^{(i)} \approx \left[1 - p_D + p_D \sum_{\mathbf{z} \in \mathbf{Z}_k^{(m)}} \frac{I_k(\mathbf{z})}{\lambda c(\mathbf{z})}\right] \cdot w_{k|k-1}^{(i)}$, for $i = 1, \dots, N+B_{k-1}$

17: Normalize weights: $w_{k|k}^{(i)} = \frac{\tilde{w}_{k|k}^{(i)}}{\sum_{j=1}^{N+B_{k-1}} \tilde{w}_{k|k}^{(j)}}$, for $i = 1, \dots, N+B_{k-1}$

18: Compute reward $\mathcal{J}^{(m)}(\mathbf{v})$ according to (5.7)

19: **end for**

20: Expected reward $\mathbb{E}\{\mathcal{J}(\mathbf{v})\} \approx \frac{1}{M} \sum_{m=1}^M \mathcal{J}^{(m)}(\mathbf{v})$

21: **end for**

22: Select control \mathbf{u}_{k-1} according to (5.4)

23: Apply control \mathbf{u}_{k-1} and collect the actual measurement set \mathbf{Z}_k

24: For every $\mathbf{z} \in \mathbf{Z}_k$ compute $I_k(\mathbf{z}) = \sum_{i=1}^{N+B_{k-1}} w_{k|k-1}^{(i)} \cdot g_k(\mathbf{z}|\mathbf{x}_{k|k-1}^{(i)})$

25: Compute $\delta_k \approx p_D \left(1 - \sum_{\mathbf{z} \in \mathbf{Z}_k} \frac{I_k(\mathbf{z})}{\lambda c(\mathbf{z})}\right)$.

26: Update existence: $q_{k|k} = \frac{1 - \delta_k}{1 - \delta_k \cdot q_{k|k-1}} \cdot q_{k|k-1}$ ▷ Eq. (2.47)

27: Update weights: $\tilde{w}_{k|k}^{(i)} \approx \left[1 - p_D + p_D \sum_{\mathbf{z} \in \mathbf{Z}_k} \frac{I_k(\mathbf{z})}{\lambda c(\mathbf{z})}\right] \cdot w_{k|k-1}^{(i)}$, for $i = 1, \dots, N+B_{k-1}$

28: Normalize weights: $w_{k|k}^{(i)} = \frac{\tilde{w}_{k|k}^{(i)}}{\sum_{j=1}^{N+B_{k-1}} \tilde{w}_{k|k}^{(j)}}$, for $i = 1, \dots, N+B_{k-1}$

29: **for** $i = 1, \dots, N$ **do** ▷ (Resampling)

30: Select index $j^i \in \{1, \dots, N\}$ with probability $w_{k|k}^{(i)}$

31: $\mathbf{x}_k^{(i)} = \mathbf{x}_{k|k-1}^{(j_i)}$

32: **end for**

33: Output: $q_{k|k}$, $\{\mathbf{x}_k^{(i)}\}_{i=1}^N$

34: **end function**

loop, the observer motion is controlled sequentially using the output of the Bernoulli particle filter. The observer motion is critical for autonomous bearings-only tracking, because the observer platform must perform a maneuver in order to be able to estimate its distance (range) to the target [5]. This need for the ownship maneuver and its impact on target state observability have been explored extensively, see for example [6, 7]. Observer path optimization, in the absence of detection uncertainty, has been studied by [8–10], typically using the Fisher information as a criterion. The problem with the Fisher information, however, is twofold. First, the target must be detected before one can compute the Fisher information (not defined otherwise). Second, it only concerns the second-order moment, which is not sufficient for non-Gaussian posterior densities. Instead, we use the criterion based on the information gain defined in (5.3): this criterion overcomes both aforementioned limitations of the Fisher information.

5.1.2.1 Problem Formulation

The target state vector is adopted as

$$\mathbf{x}_k^t = [x_k^t \dot{x}_k^t y_k^t \dot{y}_k^t]^\top \quad (5.8)$$

where (x_k^t, y_k^t) and $(\dot{x}_k^t, \dot{y}_k^t)$ denote the target position and velocity, respectively, in Cartesian coordinates. The ownship state vector \mathbf{x}_k^o , which is known, is similarly defined as $\mathbf{x}_k^o = [x_k^o \dot{x}_k^o y_k^o \dot{y}_k^o]^\top$. The motion model will be written for the relative state vector, defined as:

$$\mathbf{x}_k := \mathbf{x}_k^t - \mathbf{x}_k^o = [x_k \dot{x}_k y_k \dot{y}_k]^\top. \quad (5.9)$$

Target motion is modeled by a nearly constant velocity (CV) model:

$$\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k - \mathbf{S}_{k+1,k} + \mathbf{v}_k \quad (5.10)$$

where \mathbf{F}_k is the transition matrix, \mathbf{S}_k is a known deterministic matrix taking into account the effect of observer accelerations, and $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$ is white Gaussian process noise. We adopt:

$$\mathbf{F}_k = \mathbf{I}_2 \otimes \begin{bmatrix} 1 & T_k \\ 0 & 1 \end{bmatrix}, \quad \mathbf{Q}_k = \mathbf{I}_2 \otimes \varpi \begin{bmatrix} \frac{T_k^3}{3} & \frac{T_k^2}{2} \\ \frac{T_k^2}{2} & T_k \end{bmatrix}, \quad \mathbf{S}_{k+1,k} = \begin{bmatrix} x_{k+1}^o - x_k^o - T_k \dot{x}_k^o \\ \dot{x}_{k+1}^o - \dot{x}_k^o \\ y_{k+1}^o - y_k^o - T_k \dot{y}_k^o \\ \dot{y}_{k+1}^o - \dot{y}_k^o \end{bmatrix} \quad (5.11)$$

where \otimes is the Kronecker product, $T_k = t_{k+1} - t_k$ is the sampling interval, and ϖ is the intensity of process noise [11, p. 269]. By adopting $T_k = T = \text{const}$, notation simplifies to $\mathbf{F}_k = \mathbf{F}$ and $\mathbf{Q}_k = \mathbf{Q}$.

In order to model target appearance and disappearance during the observation period, we introduce a binary random variable $\varepsilon_k \in \{0, 1\}$ referred to as the *target existence* (the convention is that $\varepsilon_k = 1$ means that target exists at discrete-time k). Dynamics of ε_k is modeled by a two-state Markov chain with a transitional probability matrix (TPM) Π . The elements of the TPM are defined as $[\Pi]_{ij} = P\{\varepsilon_{k+1} = j - 1 | \varepsilon_k = i - 1\}$ for $i, j \in \{1, 2\}$. We adopt a TPM as follows:

$$\Pi = \begin{bmatrix} (1 - p_b) & p_b \\ (1 - p_s) & p_s \end{bmatrix} \quad (5.12)$$

where $p_b := P\{\varepsilon_{k+1} = 1 | \varepsilon_k = 0\}$ is the probability of target “birth” and $p_s := P\{\varepsilon_{k+1} = 1 | \varepsilon_k = 1\}$ the probability of target “survival”. These two probabilities together with the initial target existence probability $q_0 = P\{\varepsilon_0 = 1\}$ need to be specified in advance.

At each time t_k , a set of angular measurements reported by the passive sensor is available for processing. Let the set of bearings measurements, available at time t_k , be denoted $Z_k = \{z_{k,1}, \dots, z_{k,m_k}\}$. Both the number of measurements $m_k \in \mathbb{N}_0$ and the position of measurements in the measurement space $\mathcal{Z} = (-\pi, \pi)$ rad, are random. Due to the imperfections in the detection process, at most one measurement is from the target and the rest are false detections. If a measurement $z \in Z_k$ is from the target, then it is related to the target state as follows:

$$z = h(\mathbf{x}_k) + w_k \quad (5.13)$$

where w_k is a zero-mean independent Gaussian noise with variance σ_θ^2 and

$$h(\mathbf{x}_k) = \text{atan2}(x_k, y_k) \quad (5.14)$$

is the four-quadrant inverse tangent function, resulting in the true target bearing at time k , and taking values in \mathcal{Z} . The probability of target detection is p_D .

The false detections (or clutter) are modeled as follows: the spatial distribution of clutter over \mathcal{Z} is time invariant and denoted by $c(z)$; the number of clutter points per at time k is assumed to be Poisson distributed, with the constant mean value λ .

The problem is to estimate the posterior probability of target existence $q_{k|k} := P\{\varepsilon_k = 1 | Z_{1:k}\}$ and the posterior spatial PDF of the target, $s_{k|k}(\mathbf{x}) = p(\mathbf{x}_k | Z_{1:k})$. This task needs to be carried out by controlling sequentially the observer motion in such a manner that collected measurements are most informative. The observer motion decisions are made in the presence of uncertainty (especially at initial stages) using only the past observations. The action space (the space of admissible motion vectors) in this case is a continuous space of observer heading and speed values in the plane. Let $\mathbb{U}_k \subset \mathbb{R}^2$ denote the set of admissible observer control vectors at time k . The problem of observer control is to select at each k the most informative action $\mathbf{u} \in \mathbb{U}_k$, based on the current state of knowledge about the target.

5.1.2.2 Solution, Implementation, and the Testing Scenario

The adopted solution is the Bernoulli particle filter with sensor control, described in general terms in Algorithm 9. Some specific details of the implementation follow.

The observer typically moves in straight-line legs: observer control is applied at the end of each leg for the sake of choosing the heading and speed in the next leg. For this reason one step ahead observer control is not appropriate here; instead it is better to consider $H \geq 1$ steps ahead (H is referred to as *horizon*). Let us assume that the current time is t_k , that current posteriors are $q_{k|k}$ and $s_{k|k}(\mathbf{x})$ and that we want to find out the information gain of action $\mathbf{u} \in \mathbb{U}_k$ at time t_{k+H} which corresponds to the end of the next leg of the observer trajectory. Equation (5.3) then takes the form:

$$\begin{aligned} \mathcal{J}(\mathbf{u}) = & \frac{1}{\alpha - 1} \log \left\{ [1 - q_{k+H|k}]^{1-\alpha} [1 - q_{k+H|k+H}]^{\alpha} + \right. \\ & \left. [q_{k+H|k}]^{1-\alpha} q_{k+H|k+H}^{\alpha} \int [s_{k+H|k}(\mathbf{x})]^{1-\alpha} \cdot [s_{k+H|k+H}(\mathbf{x})]^{\alpha} d\mathbf{x} \right\} \end{aligned} \quad (5.15)$$

Furthermore, let us assume a high value of p_D and low to moderate clutter. Then the quantities $q_{k+H|k}$, $s_{k+H|k}(\mathbf{x})$, $q_{k+H|k+H}$, and $s_{k+H|k+H}(\mathbf{x})$, which feature in (5.15), can be found under the assumptions of the *predicted ideal measurement set*, i.e., for: $p_s = 1$, $p_b = 0$, $p_D = 1$, no clutter, no measurement noise, and no process noise. Under the adopted *ideal* conditions, the hypothetical measurement set at time $k + H$ is a singleton: $Z_{k+H} = \{z\}$. The likelihood function in (2.46) reduces to $g_k(z|\mathbf{x})$ and equations of the Bernoulli filter reduce to the equations of the standard Bayes filter. As a result we can write:

$$q_{k+H|k+H} = q_{k+H|k} = q_{k|k} \quad (5.16)$$

$$w_{k+H|k}^{(i)} = w_{k|k}^{(i)} \quad (5.17)$$

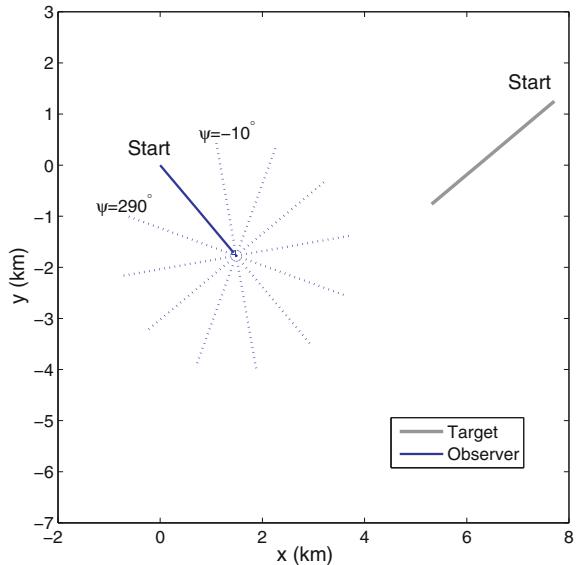
$$\mathbf{x}_{k+H|k}^{(i)} = \mathbf{F}^H \mathbf{x}_{k|k}^{(i)} - \mathbf{S}_{k+H,k} \quad (5.18)$$

$$w_{k+H|k+H}^{(i)} \propto g_{k+H}(z|\mathbf{x}_{k+H|k}^{(i)}) \cdot w_{k|k}^i \quad (5.19)$$

where $i = 1, \dots, N + B_{k-1}$. In order to compute the expected gain, an ensemble of future measurements z can be generated as in line 12 of Algorithm 9.

In the practical implementation of the algorithm, the space of admissible control vectors is discretized, i.e., $\mathbb{U}_k = \{\mathbf{u}_{k,\ell}; \ell = 1, \dots, L\}$. The observer controller has to choose between $L > 1$ turns at different headings. The testing scenario is shown in Fig. 5.1. Its total duration is 30 min, the sampling interval is $T = 30$ s (i.e., the total number of discrete-time steps is 60). The target, which is initially 8 km away from the observer, exists and is visible throughout the observation interval. It maintains the course of -130° and travels at a constant speed of 7 knots. Only the first half of its trajectory is shown in Fig. 5.1. The observer is traveling at the speed of 5 knots and its course during the first leg of its trajectory is 140° . At the end of the first leg

Fig. 5.1 Testing scenario, indicating the set of admissible future legs of the observer trajectory



(after $k = 30$), the observer has to choose a new course for the second leg of the journey (the observer speed remains unchanged). There are $L = 11$ options available in \mathbb{U}_k with horizon $H = 30$, all shown in Fig. 5.1 in dotted lines. The admissible observer trajectories in \mathbb{U}_k correspond to the observer course of $\psi = -10^\circ, 20^\circ, \dots, 290^\circ$.

For reporting and performance evaluation purposes, if the probability of target existence $q_{k|k}$ is above a certain threshold $\tau \in (0, 1)$, the expected a posteriori estimate of the target state is computed after line 28 of Algorithm 9 as:

$$\hat{\mathbf{x}}_{k|k} = \mathbb{E}[\mathbf{x}_k | Z_{1:k}] \approx \sum_{i=1}^{N+B_{k-1}} w_{k|k}^{(i)} \mathbf{x}_{k|k-1}^{(i)} \quad (5.20)$$

If required, one can also compute the sample covariance matrix associated with the estimate (5.20) [3].

5.1.2.3 Numerical Results

The measurement standard deviation is $\sigma_\theta = 1^\circ$, the probability of detection p_D and the average number of false detections per scan λ will be varied. The model of clutter spatial distribution is uniform, $c(z) = (2\pi)^{-1}$.

The parameters of the Bernoulli particle filter are as follows: the probability of survival $p_s = 0.98$, birth probability $p_b = 0.01$, process noise intensity $\varpi = 0.2$, number of particles $N = 5000$. The target birth density $\beta(\mathbf{x}|z)$ is designed based

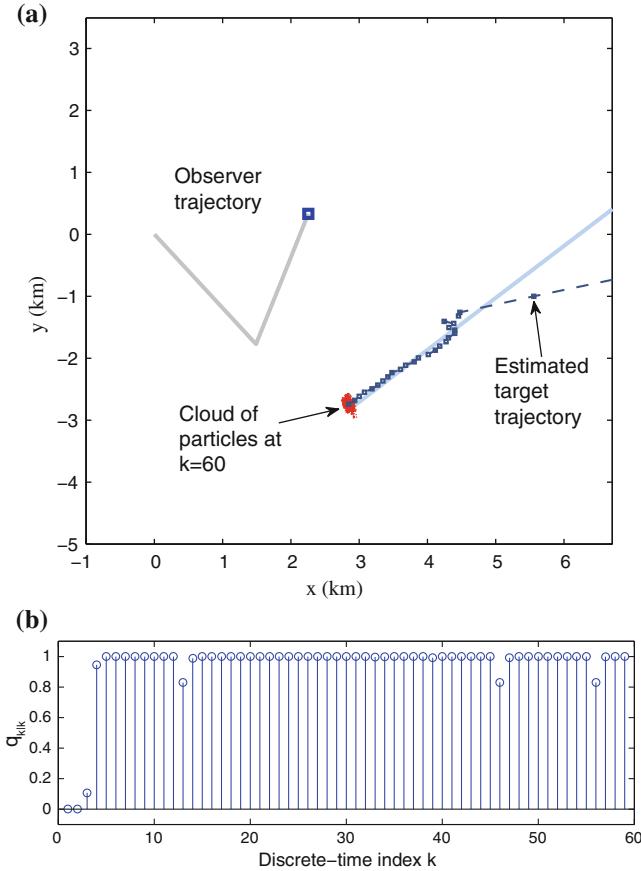


Fig. 5.2 A typical run of the Bernoulli particle filter with observer control: **a** observer motion and the estimated target trajectory at $k = 60$; **b** the probability of target existence $q_{k|k}$ versus k

on the anticipated target range $\bar{r} = 11$ km, with standard deviation $\sigma_{\bar{r}} = 3.5$ km; the standard deviation of velocity components of birth particles is $\sigma_v = 2.6$ m/s, see [12]. The parameters used in the reward function computation are: the ideal measurement sample size is $M = 500$ and the Rényi divergence parameter α will be varied. Figure 5.2 illustrates a typical result obtained by a single run of the algorithm, using $\alpha = 0.5$, $p_D = 0.9$ and $\lambda = 1.0$. The location of the observer at $k = 60$ is indicated by square in Fig. 5.2a. The course chosen for the second leg of the observer trajectory in this particular run was 20° . The estimated target trajectory, computed using (5.20), indicates a good agreement with the ground truth after the observer has maneuvered. The probability of existence, shown in Fig. 5.2b, grows to 1 after only a few discrete-time steps and remains high throughout the observation interval. Occasionally, when the target detection is missing, it drops to about 0.8. The occasional false detections do not seem to affect significantly the performance of the Bernoulli PF (BPF), developed according to Algorithm 9 for this application.

Two sets of Monte Carlo simulation results are presented next. The first deals with the error performance of the Bernoulli PF and considers a fixed scenario, where the observer heading during the second leg of its trajectory is fixed at 20° , as shown in Fig. 5.2a. The second investigates the best choice of the observer turn at the end of the first leg of its trajectory.

The performance measures in the first set of numerical results is the positional RMS error, estimated as:

$$\varepsilon_k = \sqrt{\frac{1}{P} \sum_{p=1}^P \left[(\hat{x}_{k|k}^{(p)} - x_k)^2 + (\hat{y}_{k|k}^{(p)} - y_k)^2 \right]} \quad (5.21)$$

where P is the total number of Monte Carlo runs, $(\hat{x}_{k|k}^{(p)}, \hat{y}_{k|k}^{(p)})$ denotes the estimated relative target state position at time k in the p th run, and (x_k, y_k) is the *ground truth*. The RMS errors will be compared against the square-root of the theoretical Cramér-Rao lower bound (CRLB) for a fixed target trajectory with $p_D < 1$ [13, 14]. The CRLB at time k is the inverse of the Fisher information matrix \mathbf{J}_k , which is computed recursively as follows [13]:

$$\mathbf{J}_{k+1} = (\mathbf{F}^{-1})^\top \mathbf{J}_k \mathbf{F}^{-1} + \frac{p_D}{\sigma_\theta^2} \mathbf{H}_{k+1}^\top \mathbf{H}_{k+1} \quad (5.22)$$

where \mathbf{H}_k is the Jacobian of the measurement function $h(\mathbf{x}_k)$ in (5.14), given by:

$$\mathbf{H}_k = \begin{bmatrix} \frac{y_k}{x_k^2 + y_k^2} & 0 \\ 0 & -\frac{x_k}{x_k^2 + y_k^2} \end{bmatrix}. \quad (5.23)$$

In the computation of the Fisher information (and the corresponding CRLB) it is necessary to know the ground truth, since it features in (5.23).

Figure 5.3a compares the Monte Carlo estimated positional RMS error ε_k with the theoretical $\sqrt{\text{CRLB}_k} = \sqrt{\mathbf{J}_k^{-1}[1, 1] + \mathbf{J}_k^{-1}[3, 3]}$. The two curves for ε_k were obtained by averaging over 100 Monte Carlo runs, for the case of $p_D = 0.9$ with the average number of false detections $\lambda = 0.5$ and 5. The RMS error curves are shown for the period after the maneuver (i.e. $k > 30$), when the state becomes observable. Figure 5.3b displays the probability of existence $q_{k|k}$, averaged over the same set of 100 Monte Carlo runs. In the preparation of Fig. 5.3, the reporting threshold τ of the Bernoulli PF was set to a very low value of 0.01.

Note from Fig. 5.3 that the RMS error of the Bernoulli PF at $\lambda = 0.5$ is only slightly worse than the theoretical bound (which is a conservative bound, being computed for $\lambda = 0$ case). For the higher value of λ , however, the estimation error increases. According to Fig. 5.3b, λ also affects the track initiation delay: on average the track is established quicker for a smaller value of λ . Once the track is established, λ does not seem to significantly affect the probability of existence.

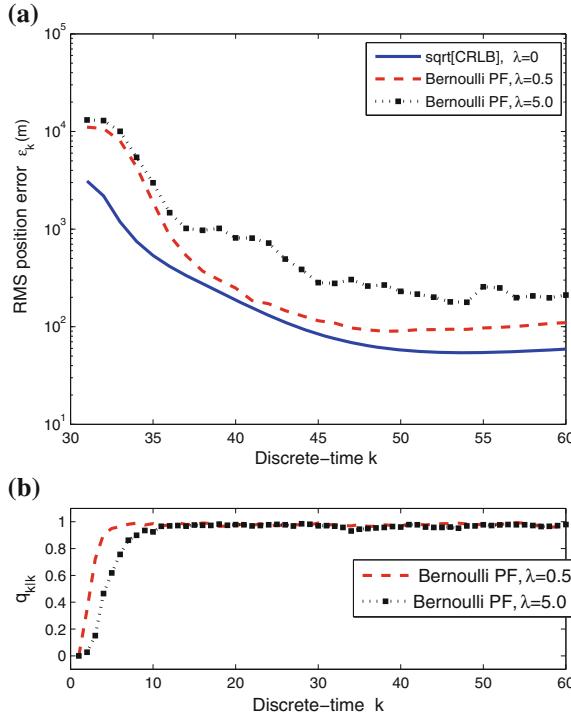


Fig. 5.3 Error performance of the Bernoulli PF (averaged over 100 Monte Carlo simulations) at $p_D = 0.9$, with $\lambda = 0.5$ and 5: **a** Positional RMS error ε_k and the theoretical CRLB for $p_D = 0.9$ and $\lambda = 0$, shown for $k > 30$; **b** The probability of existence $q_{k|k}$ versus k

Next we investigate the performance of the proposed reward function for observer control, using detection parameters $p_D = 0.9$ and $\lambda = 0.5$. The first question is: which of the $L = 11$ second-leg options for the observer heading is indeed the best. To answer this question, we have estimated the RMS position error at the last scan $k = 60$ for each admissible course $\psi = -10^\circ, 20^\circ, \dots, 290^\circ$. These RMS errors were obtained by fixing the value of ψ and conducting 100 Monte Carlo runs for each ψ to compute $\varepsilon_{k=60}$ according to (5.21). The result, plotted in Fig. 5.4, clearly indicates that the second-leg course decisions of $-10^\circ, 20^\circ$ or 50° are preferable in this scenario.

After establishing what is a good decision for the observer second-leg heading, we let the observer controller make its own decisions. Table 5.1 presents the number of times (out of 100 Monte Carlo runs) the observer control algorithm has opted for each particular course. The results, shown for different values of Rényi divergence ($\alpha = 0.1, 0.5$, and 0.99), indicate a strong preference for $\psi = 20^\circ$ course for all values of parameter α . Considering that the observer course decision was made at the time when the target range was unobservable (the initial target range was based

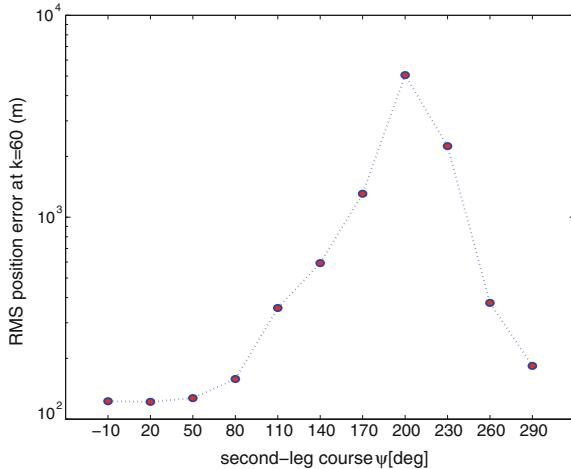


Fig. 5.4 RMS positional error at $k = 60$ for all admissible second-leg observer courses $\psi = -10^\circ, 20^\circ, \dots, 290^\circ$ (parameters $p_D = 0.9, \lambda = 0.5$). Every RMS error value is obtained by averaging over 100 Monte Carlo runs

Table 5.1 Number of times (out of 100 Monte Carlo runs) a particular second-leg course ψ has been chosen for different values of parameter α

$\alpha \setminus \psi$	-10°	20°	50°	80°	110°	140°	170°	200°	230°	260°	290°
0.10	15	71	14	0	0	0	0	0	0	0	0
0.50	17	75	8	0	0	0	0	0	0	0	0
0.99	13	76	11	0	0	0	0	0	0	0	0

on a guess with a significant bias), the controller decisions presented in Table 5.1 are remarkably good.

The Fig. 5.5, shows the RMS position errors (obtained by averaging over 100 Monte Carlo runs) for various values of the reward parameter α . It appears that in this application the choice of α has very little influence on the overall performance.

5.1.3 Target Tracking via Multi-Static Doppler Shifts

In this application the problem is to detect and estimate the position and velocity of a moving object, using measurements of Doppler-shift frequencies at several spread-out locations. The sensor control aspect is used here to select at each sampling time a subset of available Doppler-only sensors which is to report their measurements to the fusion center.

The problem, which can be cast in radar or sonar/acoustic context, has a long history [15–17]. The renewed interest is driven by applications, such as

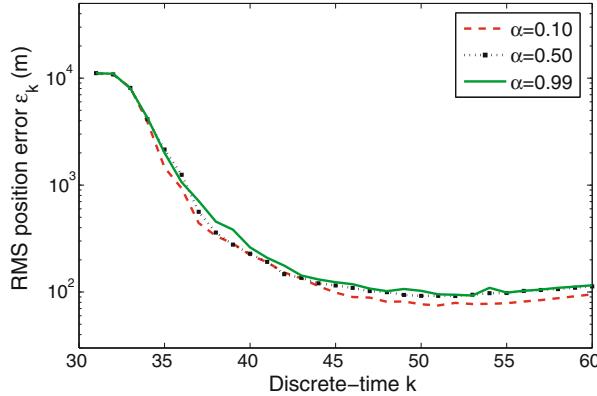


Fig. 5.5 RMS position error ε_k for $\alpha = 0.1, 0.5, 0.99$

passive surveillance, and the technological improvements in wireless sensor networks [18–20]. In the radar context, for example, the transmitters (illuminators) are typically the commercial digital audio/video broadcasters, FM radio transmitters, or GSM base stations, whose transmitting frequencies are known. The radar receivers can typically measure the multi-static range, angle, and Doppler-shift. The current trend in surveillance, however, is to use many low cost, low power sensors, connected in a network [21]. In line with this trend, our goal is to investigate the possibility of tracking a moving target using a network of low-cost receivers that measure Doppler frequencies only. The sonar/acoustic context has recently been investigated in [22], with experimental data showing promising performance.

Most of the earlier work is focused on *observability* of the target state from the Doppler-shift measurements [18, 19] and geometry-based *localization algorithms* [17, 20, 23]. We cast the problem in the stochastic filtering framework which includes the target appearance/disappearance model and realistic measurement characteristics.

5.1.3.1 Problem Formulation

The state of the moving object (target) in the two-dimensional surveillance area at time t_k is represented by the state vector

$$\mathbf{x}_k = [x_k \ \dot{x}_k \ y_k \ \dot{y}_k]^T, \quad (5.24)$$

where target position and velocity are denoted by $\mathbf{p}_k = [x_k, \ y_k]^T$ and $\mathbf{v}_k = [\dot{x}_k \ \dot{y}_k]^T$, respectively. Target motion is modeled by a nearly constant velocity (CV) model:

$$\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \mathbf{v}_k \quad (5.25)$$

where \mathbf{F}_k is the transition matrix and $\mathbf{v}_k \sim \mathcal{N}(\mathbf{u}; \mathbf{0}, \mathbf{Q}_k)$ is zero-mean white Gaussian process noise with covariance \mathbf{Q}_k . As in Sect. 5.1.2 we adopt:

$$\mathbf{F}_k = \mathbf{I}_2 \otimes \begin{bmatrix} 1 & T_k \\ 0 & 1 \end{bmatrix}, \quad \mathbf{Q}_k = \mathbf{I}_2 \otimes \varpi \begin{bmatrix} \frac{T_k^3}{3} & \frac{T_k^2}{2} \\ \frac{T_k^2}{2} & T_k \end{bmatrix}, \quad (5.26)$$

where as usual \otimes is the Kronecker product, $T_k = t_{k+1} - t_k$ is the sampling interval and ϖ is the intensity of process noise [11, p. 269].

Target appearance and disappearance are modeled as in Sect. 5.1.2: random variable $\varepsilon_k \in \{0, 1\}$ represents *target existence* and the convention is that $\varepsilon_k = 1$ means that target exists at time k . Dynamics of ε_k is modeled by a two-state Markov chain with a TPM Π specified in (5.12). Probabilities of birth and survival, p_b and p_s , respectively as well as the initial target existence probability $q_0 = P\{\varepsilon_0 = 1\}$ are assumed known.

Target Doppler-shift measurements are collected by spatially distributed sensors (e.g., multi-static Doppler-only receivers), as illustrated in Fig. 5.6. A transmitter \mathbf{T} at a known location $\mathbf{t} = [x_0 \ y_0]^\top$, illuminates the target at location \mathbf{p}_k by a sinusoidal waveform of a known carrier frequency f_c . In Fig. 5.6, the receivers are denoted by \mathbf{R}_s , $s = 1, \dots, S = 4$. If the target at time k is in the state \mathbf{x}_k , and is detected by a receiver $s \in \{1, \dots, S\}$ placed at a known location $\mathbf{r}_s = [x_s \ y_s]^\top$, then the receiver will report a Doppler-shift measurement modeled as follows:

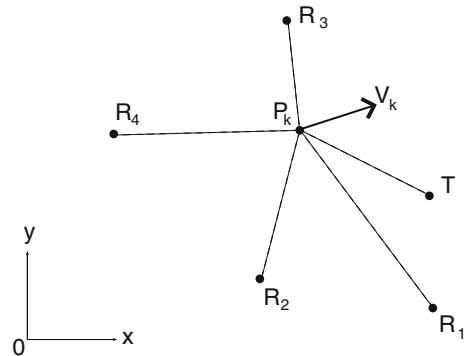
$$z_k^{(s)} = h_k^{(s)}(\mathbf{x}_k) + w_k^{(s)} \quad (5.27)$$

where

$$h_k^{(s)}(\mathbf{x}_k) = -\mathbf{v}_k^\top \left[\frac{\mathbf{p}_k - \mathbf{r}_s}{\| \mathbf{p}_k - \mathbf{r}_s \|} + \frac{\mathbf{p}_k - \mathbf{t}}{\| \mathbf{p}_k - \mathbf{t} \|} \right] \frac{f_c}{c} \quad (5.28)$$

is the true Doppler frequency shift, c is the speed of light and $w_k^{(s)}$ is measurement noise in receiver s . Measurement noise $w_k^{(s)}$ is modeled by white Gaussian noise

Fig. 5.6 Multi-static
Doppler-only surveillance
network in $x - y$ plane



with standard deviation $\sigma_w^{(s)}$; furthermore, it is independent between receivers and independent of process noise \mathbf{v}_k .

The Doppler-shift can be positive or negative. The measurement space is therefore an interval $\mathcal{Z} = [-f_0, +f_0]$, where f_0 is the maximal possible value of Doppler shift (a parameter assumed known).

Target originated Doppler shift measurement, as the one modeled by (5.27), is detected by receiver s with the probability of detection $p_D^{(s)}(\mathbf{x}_k) \leq 1$. This probability is typically a function of the distance from the transmitter to the target (in state \mathbf{x}_k) and from the target to the receiver s at location \mathbf{r}_s . Due to the imperfect detection of receivers, false detections are common. The distribution of false detections over the measurement space \mathcal{Z} is assumed time invariant and independent of the target state; it will be denoted by $c^{(s)}(z)$ for receiver s . The number of false detections in sensor s is assumed to be Poisson distributed, with the constant mean value $\lambda^{(s)}$ for all k .

A measurement set collected by sensor s at time t_k is denoted by $Z_k^{(s)} = \{z_{k,1}^{(s)}, z_{k,2}^{(s)}, \dots, z_{k,m_k^{(s)}}^{(s)}\}$. It is possible (and indeed desirable) that multiple receivers *simultaneously* collect measurements. This set of receivers is referred to as the set of *active receivers* at time t_k ; the set of active receiver indices at t_k is denoted $I_k \subseteq \{1, \dots, S\}$. The measurements from all active receivers are sent to the fusion center for processing as they become available, in the form of messages. A message referring to measurements collected at time t_k has the form:

$$\left(t_k, I_k, \bigcup_{s \in I_k} (s, Z_k^{(s)}) \right),$$

and will be denoted by $Z_k^{(I_k)}$.

The problem is to detect when a moving object appears in the surveillance area and, if present, to estimate sequentially its position and velocity vector.

5.1.3.2 Bernoulli Filter Update Equations

The problem is solved using the Bernoulli particle filter with receiver selection based on information gain specified by (5.3). The Bernoulli filter update equations for multiple simultaneous sensors and p_D dependent on the state, are slightly different from those given by (2.47–2.49).

The likelihood function of a target-originated measurement z from receiver $s \in I_k$ is denoted $g_k^{(s)}(z|\mathbf{x})$. According to (5.27) this likelihood is given by $g_k^{(s)}(z|\mathbf{x}) = \mathcal{N}(z; h_k^{(s)}(\mathbf{x}), \sigma_w^{(s)2})$. The update equation for the probability of existence is given by:

$$q_{k|k} = \frac{1 - \Delta_k}{1 - \Delta_k \cdot q_{k|k-1}} \cdot q_{k|k-1} \quad (5.29)$$

where

$$\Delta_k = 1 - \prod_{s \in I_k} \left[1 - \left\langle p_D^{(s)}, s_{k|k-1} \right\rangle + \sum_{z \in Z_k^{(s)}} \frac{\left\langle p_D^{(s)} g_k^{(s)}(z|\cdot), s_{k|k-1} \right\rangle}{\lambda^{(s)} c^{(s)}(z)} \right] \quad (5.30)$$

and, as usual, $\langle a, b \rangle = \int_{\mathcal{X}} a(\mathbf{x}) b(\mathbf{x}) d\mathbf{x}$ is the inner product between $a(\mathbf{x})$ and $b(\mathbf{x})$.

The spatial PDF is updated as follows:

$$s_{k|k}(\mathbf{x}) = \frac{\prod_{s \in I_k} \left[1 - p_D^{(s)}(\mathbf{x}) + p_D^{(s)}(\mathbf{x}) \sum_{z \in Z_k^{(s)}} \frac{g_k^{(s)}(z|\mathbf{x})}{\lambda^{(s)} c^{(s)}(z)} \right]}{1 - \Delta_k} \cdot s_{k|k-1}(\mathbf{x}) \quad (5.31)$$

It is straightforward to verify that if only one receiver is active at time k (i.e., if I_k is a singleton) and $p_D^{(s)}$ is not a function of the state \mathbf{x} , then the update equations simplify to those given by (2.47–2.49).

5.1.3.3 Implementation Remarks

We adopt receiver selection based on the reward function defined in (5.3), with parameter α fixed to 0.5, which is known as the Bhattacharyya distance [25]. The reward when using measurements $Z_k^{(I_k)}$ is:

$$\begin{aligned} \mathcal{J}(I, Z_k^{(I)}) &= -2 \log \left\{ \sqrt{[1 - q_{k|k-1}] \cdot [1 - q_{k|k}(Z_k^{(I)})]} \right. \\ &\quad \left. + \sqrt{q_{k|k-1} \cdot q_{k|k}(Z_k^{(I)})} \int \sqrt{s_{k|k-1}(\mathbf{x}) \cdot s_{k|k}(\mathbf{x}; Z_k^{(I)})} d\mathbf{x} \right\} \end{aligned} \quad (5.32)$$

where the integral is approximated using particles as in (5.6). The Bernoulli particle filter with receiver selection was implemented following the steps of Algorithm 9. Two noteworthy remarks are in order here.

First, suppose we can use maximum s_{\max} active receivers simultaneously. The set of admissible controls (actions) \mathbb{U}_{k-1} in this case is the set of subsets of $\{1, \dots, S\}$ with cardinality s_{\max} . In the computation of the information gain we make the following approximation. Initially, the expected reward $\mathbb{E}\{\mathcal{J}\}$ is computed for each receiver individually. Then the expected rewards are ranked. Finally, the set of indices I_k is chosen so that it includes s_{\max} receivers with highest ranked rewards.

Second, after resampling (lines 29–32 in Algorithm 9) the MCMC move step [3, p. 55] is necessary to improve the diversity of particles. The newborn particles

are drawn in line 3 from $b_{k-1} \left(\mathbf{x} | \mathbf{Z}_{k-1}^{(I_{k-1})} \right)$ using the Accept-Reject method [26]. Initially, we draw samples from a multivariate Gaussian $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \mathbf{C})$, where $\boldsymbol{\mu}$ and \mathbf{C} are the mean and covariance adopted so that the target birth density covers the entire state space \mathcal{X} . Then we accept only those samples whose velocity vector is compatible with the Doppler measurement $z \in \mathbf{Z}_{k-1}^{(I_{k-1})}$. The reporting threshold of the algorithm is $\tau = 0.5$. The point target state estimate is computed as in (5.20).

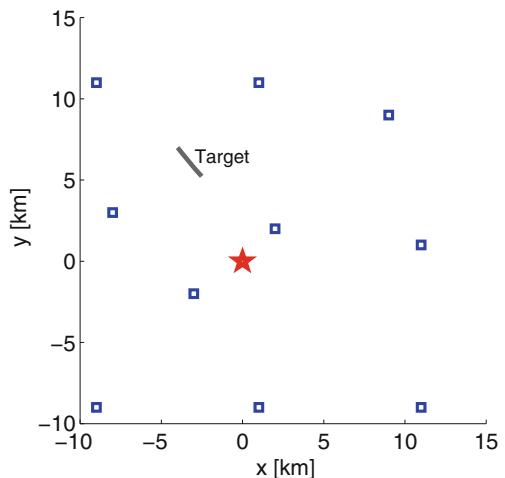
5.1.3.4 Numerical Results

The Bernoulli particle filter for multi-static Doppler-shift measurements is tested using a scenario with $S = 10$ receivers placed in the $x - y$ plane at locations shown in Fig. 5.7a. The emitter with transmitting frequency $f_c = 900$ MHz is placed at the origin of the $x - y$ plane. The sampling interval is fixed to $T = 2$ s. The target exists for discrete-time indices $k = 4, 5, \dots, 40$, with the true initial state vector $\mathbf{x}_4 = [-4 \text{ km} \quad 20 \text{ m/s} \quad 7 \text{ km} \quad -25 \text{ m/s}]^\top$ and process noise parameter $\varpi = 0.01$. All receivers $s = 1, \dots, S$ are characterized by identical measurement noise standard deviations $\sigma_w^{(s)} = \sigma_w$ Hz, and false detection (clutter) parameters $\lambda^{(s)} = 2$ and $c^{(s)}(z) = 1/400$ Hz for $z \in [-200 \text{ Hz}, 200 \text{ Hz}]$ and zero otherwise. The probability of detection is modeled as:

$$p_D^{(s)}(\mathbf{x}_k) = 1 - \mathcal{C}(d_{k,s}(\mathbf{x}_k); d_0, P) \quad (5.33)$$

where $d_{k,s}(\mathbf{x}_k) = \| \mathbf{t} - \mathbf{p}_k \| + \| \mathbf{p}_k - \mathbf{r}_s \|$ is the distance from the transmitter to the target and from the target to receiver s ; $\mathcal{C}(d; d_0, P) = \int_{-\infty}^d \mathcal{N}(v; d_0, P) dv$ is the Gaussian cumulative distribution function with $d_0 = 23$ km and $P = (3 \text{ km})^2$.

Fig. 5.7 The scenario considered in numerical simulations: placement of $S = 10$ receivers indicated by squares; the location of the the transmitter indicated by a star at $(0, 0)$; target trajectory is a solid straight line



According to model (5.33), if the distance $d_{k,s} \ll d_0$, then the term $\mathcal{C}(d; d_0, P) \approx 0$ and $p_D^{(s)} \approx 1$. Similarly, if $d_{k,s} \gg d_0$, then $p_D^{(s)} \approx 0$.

The Bernoulli particle filter was implemented using the birth parameters $\mu = [0 \ 0 \ 0 \ 0]^T$, $\mathbf{C} = \text{diag}[(4 \text{ km})^2 \ (30 \text{ m/s})^2 \ (4 \text{ km})^2 \ (30 \text{ m/s})^2]$, with $N = 10000$, $p_b = 0.02$, $p_s = 0.98$ and the initial probability of existence $q_0 = 0$. The average reward was computed using $M = 100$ fictitious future measurement samples (see the **for** loop in line 10 of Algorithm 9).

The results of a single run of the algorithm are illustrated in Fig. 5.8. This figure was obtained using $s_{\max} = 2$ active receivers at each time k , and for measurement noise standard deviation $\sigma_w = 2.5 \text{ Hz}$. The target actually appears at $k = 4$ and its existence is quickly detected as the probability of existence $q_{k|k}$ jumps to 0.8 at $k = 5$, see Fig. 5.8a. Once the target has been localized, the controller-selected active receivers are always those closest to the target. This is a perfectly reasonable strategy, because it ensures the probability of detection of active receivers to be close to 1. Consequently, the probability of existence remains very stable at the value of 1 throughout the observation interval. The target disappears at $k = 41$ and this is event is quickly detected since $q_{k|k}$ immediately drops to a small value close to zero. The target trajectory estimate, shown by a dotted line in Fig. 5.8c, approaches the ground truth (the solid straight line). The 2σ uncertainty ellipsoids (estimated from the particles) include at every discrete-time k the true target position.

The estimation accuracy of the Bernoulli particle filter is measured using the OSPA error metric, described in Sect. 2.4.3, with parameters $p = 1$ (so that the OSPA error represents the sum of the localization error and the cardinality error) and $c = 8 \text{ km}$ (the penalty assigned to the cardinality error). The OSPA error for the single-run of the algorithm is shown in Fig. 5.8b. Note that the OSPA error is very high for $4 \leq k \leq 10$; the source of error at $k = 4$ is the cardinality error; after that the OSPA is dominated by the localization error, which drops to small values for $k > 10$, as the Bernoulli PF converges (using $s_{\max} = 2$ the convergence takes six discrete-time steps).

Next we investigate the mean OSPA error (averaged over 20 Monte Carlo runs) for two comparative studies. In the first study, the influence of s_{\max} (the number of instantaneously active receivers) on tracking performance is analyzed. The resulting OSPA curves are shown in Fig. 5.9 for $s_{\max} = 1, 2, 4$ (note that the y-axis is in the log-scale and when the mean OSPA is zero, its values are not shown). Figure 5.9 demonstrates that, in accordance with our intuition, by using more active receivers (i.e., greater s_{\max}), the mean OSPA reduces quicker and to a lower value of the steady-state error. It is important to emphasize, however, that even for $s_{\max} = 1$, the algorithm performs reasonably well, and this case is arguably easier to implement. The jump in OSPA error at $k = 41$ and $k = 42$ is due to cardinality error: on some Monte Carlo runs the probability of existence $q_{k|k}$ was decreasing slowly after $k > 40$ and remained above the threshold 0.5.

The second study explored the significance of receiver selection by the controller. It compared the proposed information gain-based approach [Sect. 5.1.1, with the reward function in (5.32)], against a random selection of s_{\max} receivers

Fig. 5.8 Single run of the algorithm: **a** the posterior probability of existence $q_{k|k}$; **b** OSPA error over time; **c** The true and estimated target trajectory with overlayed 2σ uncertainty ellipsoids (the target is moving toward south-east)

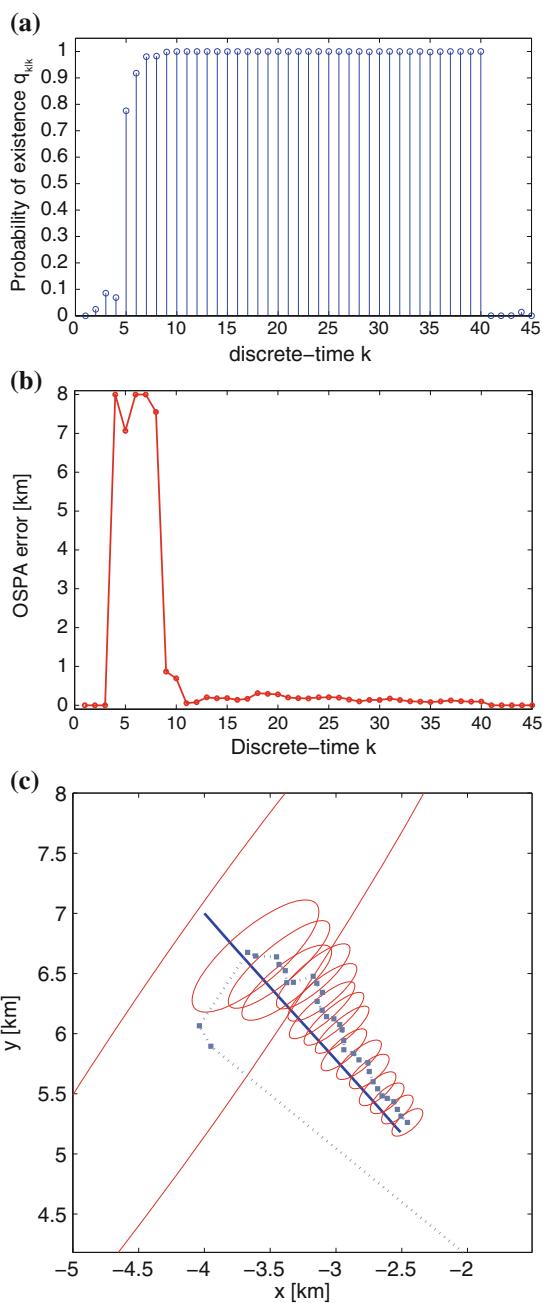


Fig. 5.9 The mean OSPA error for the number of active receivers $s_{\max} = 1, 2, 4$ ($\sigma_w = 2.5$ Hz)

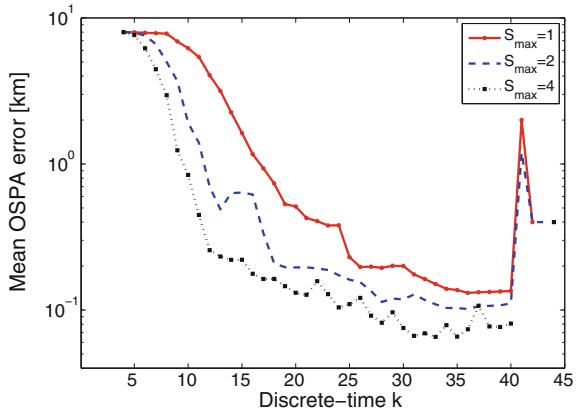
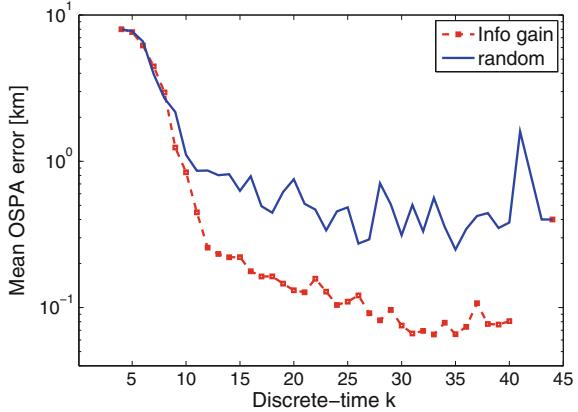


Fig. 5.10 The mean OSPA error using information gain-based selection versus a random selection of $s_{\max} = 4$ active receivers ($\sigma_w = 2.5$ Hz)



from $\{1, \dots, S\}$ at $k = 1, 2, \dots$. The results, using $s_{\max} = 4$, are shown in Fig. 5.10. The information gain-based selection achieves a significantly lower value of the steady-state error.

In summary, the Bernoulli PF solves effectively the problem of detection and tracking based on multi-static Doppler-shift measurements. The importance of information-gain sensor selection has been demonstrated, as well as the benefits of using synchronous measurements from multiple receivers.

5.2 Sensor Control for PHD/CPHD Particle Filters

PHD/CPHD filters (see Sects. 2.5.2 and 4.2) are computationally efficient yet principled approximations of the exact multi-object Bayes filter. This section derives analytical expressions for the Rényi divergence as the reward function for sensor control,

under the assumption of IID cluster RFS and Poisson RFS (defined in Sect. 2.4.1). The IID cluster RFS is the basis of the cardinalized PHD filter [27], while its special case, the Poisson RFS, is the basis of the PHD filter [28]. Both the PHD filter and its appropriate Rényi divergence for one-step ahead (myopic) sensor management are implemented using the sequential Monte Carlo method [29]. A numerical example involving multiple moving objects and a controllable moving range-only measuring sensor will be presented in this section in order to illustrate the utility of the derived reward function.

5.2.1 The Reward Function

The general multi-object formulation of the single-step ahead reward function was given by (5.1). Let us rewrite (5.1) using a simpler notation as follows:

$$\mathcal{J}(\mathbf{u}) = \frac{1}{\alpha - 1} \log \int f_1(\mathbf{X}; \mathbf{u})^\alpha f_0(\mathbf{X})^{1-\alpha} \delta \mathbf{X} \quad (5.34)$$

Suppose that RFS $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ and also that both the predicted and updated FISST PDFs in (5.34) are IID cluster PDFs (as in the CPHD filter). According to (2.29):

$$f_0(\mathbf{X}) = n! \rho_0(n) \prod_{\mathbf{x} \in \mathbf{X}} s_0(\mathbf{x}), \quad (5.35)$$

$$f_1(\mathbf{X}; \mathbf{u}) = n! \rho_1(n; \mathbf{u}) \prod_{\mathbf{x} \in \mathbf{X}} s_1(\mathbf{x}; \mathbf{u}). \quad (5.36)$$

The intensity function (or the PHD) of an IID cluster RFS \mathbf{X} was defined in (2.30) as:

$$D(\mathbf{x}) = s(\mathbf{x}) \sum_{n=1}^{\infty} n \cdot \rho(n). \quad (5.37)$$

Using the definition of the set-integral (2.24), note that

$$\int f_1(\mathbf{X}; \mathbf{u})^\alpha f_0(\mathbf{X})^{1-\alpha} \delta \mathbf{X} \quad (5.38)$$

$$= \sum_{n=0}^{\infty} \frac{1}{n!} \int [f_1(\{\mathbf{x}_1, \dots, \mathbf{x}_n\}; \mathbf{u})]^\alpha [f_0(\{\mathbf{x}_1, \dots, \mathbf{x}_n\})]^{1-\alpha} d\mathbf{x}_1 \cdots d\mathbf{x}_n \quad (5.39)$$

$$= \sum_{n=0}^{\infty} \frac{1}{n!} \int n! \rho_1(n; \mathbf{u})^\alpha \rho_0(n)^{1-\alpha} s_1(\mathbf{x}_1; \mathbf{u})^\alpha \cdots s_1(\mathbf{x}_n; \mathbf{u})^\alpha \quad (5.40)$$

$$\times s_0(\mathbf{x}_1)^{1-\alpha} \cdots s_0(\mathbf{x}_n)^{1-\alpha} d\mathbf{x}_1 \cdots d\mathbf{x}_n \quad (5.41)$$

$$= \sum_{n=0}^{\infty} \rho_1(n; \mathbf{u})^\alpha \rho_0(n)^{1-\alpha} \left[\int s_1(\mathbf{x}; \mathbf{u})^\alpha s_0(\mathbf{x})^{1-\alpha} d\mathbf{x} \right]^n. \quad (5.42)$$

The expression for Rényi divergence (5.34) in the case of CPHD filters then simplifies to:

$$\mathcal{J}(\mathbf{u}) = \frac{1}{\alpha - 1} \log \sum_{n=0}^{\infty} \rho_1(n; \mathbf{u})^\alpha \rho_0(n)^{1-\alpha} \left[\int s_1(\mathbf{x}; \mathbf{u})^\alpha s_0(\mathbf{x})^{1-\alpha} d\mathbf{x} \right]^n. \quad (5.43)$$

The sum in (5.43) is the probability generating function of the discrete distribution $p_\alpha(n; \mathbf{u}) = \rho_1(n; \mathbf{u})^\alpha \rho_0(n)^{1-\alpha}$ with parameter $z_\alpha(\mathbf{u}) = \int s_1(\mathbf{x}; \mathbf{u})^\alpha s_0(\mathbf{x})^{1-\alpha} d\mathbf{x}$.

Let us assume now that the predicted and the updated cardinality distributions are both Poisson, with means λ_0 and $\lambda_1(\mathbf{u})$, respectively, i.e.,

$$\rho_0(n) = \frac{e^{-\lambda_0} \lambda_0^n}{n!}, \quad \rho_1(n; \mathbf{u}) = \frac{e^{-\lambda_1(\mathbf{u})} \lambda_1(\mathbf{u})^n}{n!}. \quad (5.44)$$

This corresponds to the assumption that both the predicted and updated RFSs are Poisson RFSs, and applies to the recursions of the PHD filter.

Recall that the intensity function (i.e., PHD) characterizes completely a Poisson RFS. The integral of the PHD over the state space results in the expected number of objects, see (2.50). Then we can write:

$$\lambda_0 = \int D_0(\mathbf{x}) d\mathbf{x}, \quad \lambda_1(\mathbf{u}) = \int D_1(\mathbf{x}; \mathbf{u}) d\mathbf{x}, \quad (5.45)$$

which also follows from (5.37) since both $s_0(\mathbf{x})$ and $s_1(\mathbf{x}; \mathbf{u})$ are proper PDFs. The Rényi divergence (5.43) for Poisson RFSs is then:

$$\begin{aligned} \mathcal{J}(\mathbf{u}) &= \frac{1}{\alpha - 1} \log \left\{ e^{-\lambda_0(1-\alpha)} \cdot e^{-\lambda_1(\mathbf{u})\alpha} \right. \\ &\quad \left. \sum_{n=0}^{\infty} \frac{1}{n!} \left[\lambda_0^{1-\alpha} \lambda_1(\mathbf{u})^\alpha \int s_1(\mathbf{x}; \mathbf{u})^\alpha s_0(\mathbf{x})^{1-\alpha} d\mathbf{x} \right]^n \right\} \\ &= \frac{1}{\alpha - 1} \left\{ -\lambda_1(\mathbf{u})\alpha - \lambda_0(1 - \alpha) \right. \\ &\quad \left. + \log \sum_{n=0}^{\infty} \frac{1}{n!} \left[\lambda_1(\mathbf{u})^\alpha \lambda_0^{1-\alpha} \int s_1(\mathbf{x}; \mathbf{u})^\alpha s_0(\mathbf{x})^{1-\alpha} d\mathbf{x} \right]^n \right\} \\ &= \lambda_0 + \frac{\alpha}{1 - \alpha} \lambda_1(\mathbf{u}) + \frac{\lambda_1(\mathbf{u})^\alpha \lambda_0^{1-\alpha}}{\alpha - 1} \int s_1(\mathbf{x}; \mathbf{u})^\alpha s_0(\mathbf{x})^{1-\alpha} d\mathbf{x}. \quad (5.46) \end{aligned}$$

The last equation was obtained using identity: $e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$.

Let us now summarize the main steps of the PHD filter with sensor control. After performing the PHD prediction according to (2.51), the sensor control vector is selected according to (2.8), that is:

$$\mathbf{u}_{k-1} = \arg \max_{\mathbf{v} \in \mathbb{U}_{k-1}} \mathbb{E}[\mathcal{J}(\mathbf{v})] \quad (5.47)$$

where based on (5.46) and (5.45)

$$\begin{aligned} \mathcal{J}(\mathbf{v}) &= \int D_{k|k-1}(\mathbf{x}) d\mathbf{x} + \frac{\alpha}{1-\alpha} \int D_{k|k}(\mathbf{x}; \mathbf{v}) d\mathbf{x} \\ &\quad - \frac{1}{1-\alpha} \int D_{k|k}(\mathbf{x}; \mathbf{v})^\alpha D_{k|k-1}(\mathbf{x})^{1-\alpha} d\mathbf{x} \end{aligned} \quad (5.48)$$

and $D_{k|k}(\mathbf{x}; \mathbf{v})$ is a shortened notation for $D_{k|k}(\mathbf{x} | \mathbf{Z}_{1:k-1}, \mathbf{Z}_k(\mathbf{v}))$, computed via the PHD filter update Eq. (2.52). Note that the adopted framework for sensor control is directly applicable to multiple observers if the fusion architecture is centralized and the measurements from different sensors/observers are asynchronous.¹

Observe from (5.48) that if the predicted and updated intensity functions are equal, i.e., $D_{k|k-1}(\mathbf{x}) = D_{k|k}(\mathbf{x}; \mathbf{u})$, then $\mathcal{J}(\mathbf{v}) = 0$. For $0 < \alpha < 1$, the third integral on the RHS of (5.48) is limited as follows:

$$0 \leq \int D_{k|k}(\mathbf{x}; \mathbf{u})^\alpha D_{k|k-1}(\mathbf{x})^{1-\alpha} d\mathbf{x} \leq \left[\int D_{k|k}(\mathbf{x}; \mathbf{v}) d\mathbf{x} \right]^\alpha \left[\int D_{k|k-1}(\mathbf{x}) d\mathbf{x} \right]^{1-\alpha} \quad (5.49)$$

Hence, the Rényi divergence in (5.48) will have the maximum value if the support of the predicted PHD and the support of the updated PHD do not overlap, that is $\int D_{k|k}(\mathbf{x}; \mathbf{u})^\alpha D_{k|k-1}(\mathbf{x})^{1-\alpha} d\mathbf{x} \approx 0$.

Since there is no analytic solution to (5.48), the reward function is implemented numerically. In the context of PHD particle filters, the predicted and updated PHD particle approximations are given by (4.39) and (4.43), respectively. The key point to note in these two approximations is that the support (the actual particle sets) are identical; only the weights are different—the updated weights are given by (4.44). This is particularly convenient for the numerical approximation of the Rényi divergence in (5.48); its Monte Carlo approximation can then be written as:

$$\mathcal{J}(\mathbf{v}) \approx \sum_{i=1}^N w_{k|k-1,p}^{(i)} + \frac{\alpha}{1-\alpha} \sum_{i=1}^N w_{k|k,p}^{(i)} - \frac{1}{1-\alpha} \sum_{i=1}^N \left(w_{k|k,p}^{(i)} \right)^\alpha \left(w_{k|k-1,p}^{(i)} \right)^{1-\alpha}. \quad (5.50)$$

The third term on the RHS of (5.50) follows from the same arguments presented in relation to (2.14).

¹ For simultaneous multi-sensor measurements the PHD filter update has no closed form solution, see [30].

The expectation \mathbb{E} over the future measurement sets $\mathbf{Z}_k(\mathbf{v})$ in (5.47) again can be done by Monte Carlo approximation, by creating an ensemble of measurement sets $\mathbf{Z}_k(\mathbf{v})$ for given clutter intensity $\kappa_k(\mathbf{z})$, probability of detection p_D and the measurement likelihood $g_k(\mathbf{z}|\mathbf{x})$. A faster (but less accurate) simplification is to adopt the *predicted ideal measurement set* approach [1], discussed in Sect. 5.1.2.2. Then we can generate only one future measurement set for each action \mathbf{v} , the one which would have resulted under the ideal conditions of no measurement noise, no clutter and $p_D = 1$. Suppose a measurement due to a single object can be expressed as $\mathbf{z} = h(\hat{\mathbf{x}}) + \mathbf{w}$, where $\mathbf{z} \in \mathbf{Z}_k(\mathbf{v})$, and $\hat{\mathbf{x}} \in \hat{\mathbf{X}}_{k|k-1}$ is the predicted state of a detected object ($\hat{\mathbf{X}}_{k|k-1}$ is the predicted multi-object state). As usual, \mathbf{w} is measurement noise. Then for each action \mathbf{v} an ideal measurement set at time k is [1]:

$$\mathbf{Z}_k(\mathbf{v}) = \bigcup_{\hat{\mathbf{x}} \in \hat{\mathbf{X}}_{k|k-1}} \{h(\hat{\mathbf{x}})\} \quad (5.51)$$

Note that in order to apply this approach we need to estimate $\hat{\mathbf{X}}_{k|k-1}$, that is we need the estimate of the number of predicted objects and their state vectors. A fast and accurate method for multi-object estimation in the SMC framework was presented in Sect. 4.2.5.

5.2.2 A Numerical Study

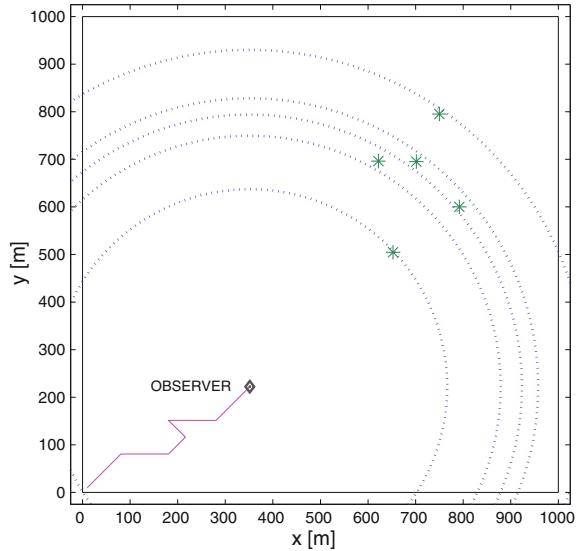
In order to demonstrate PHD particle filtering with sensor control, consider the following problem. A controllable fast moving observer, equipped with a range-only sensor, is tasked to estimate in a specified surveillance area the number of (slowly moving) objects and their positions and velocities. A single-object state is then a vector $\mathbf{x} = [\mathbf{p}^\top \ \mathbf{v}^\top]^\top$, where as usual $\mathbf{p} = [x \ y]^\top$ is the position vector, $\mathbf{v} = [\dot{x} \ \dot{y}]^\top$ is the velocity vector.

Suppose the position of the observer is $\mathbf{u} = [\chi \ \psi]^\top$. The sensor is capable of detecting an object at location \mathbf{p} with the probability:

$$p_D(\mathbf{p}) = \begin{cases} 1, & \text{if } \|\mathbf{p} - \mathbf{u}\| \leq R_0 \\ \max\{0, 1 - (\|\mathbf{p} - \mathbf{u}\| - R_0)\hbar\}, & \text{if } \|\mathbf{p} - \mathbf{u}\| > R_0 \end{cases} \quad (5.52)$$

where $\|\mathbf{p} - \mathbf{u}\| = \sqrt{(x - \chi)^2 + (y - \psi)^2}$ is the distance between the observer and an object at $\mathbf{p} = [x \ y]^\top$. In simulations we adopt a surveillance area to be a square of sides $a = 1000$ m, with p_D parameters $R_0 = 320$ m and $\hbar = 0.00025 \text{ m}^{-1}$. The range measurement originating from an object at location \mathbf{p} is then $z = \|\mathbf{p} - \mathbf{u}\| + w$, where w is zero-mean white Gaussian measurement noise, with standard deviation given by $\sigma_w = \sigma_0 + \beta \|\mathbf{p} - \mathbf{u}\|^2$. In simulations we adopt $\sigma_0 = 1$ m and $\beta = 5 \cdot 10^{-5} \text{ m}^{-1}$. Then the single-object likelihood function is $g_k(z|\mathbf{x}) = \mathcal{N}(z; \|\mathbf{p} - \mathbf{u}\|, \sigma_w^2)$.

Fig. 5.11 The location of five objects (asterisks) and the observer at time $k = 8$; current measurements are represented by arcs (dashed lines)



The sensor reports false range measurements, modeled by a Poisson RFS. The intensity function of false measurements $\kappa(z) = \mu \cdot c(z)$ is specified by uniform density from zero to $\sqrt{2}a$, i.e., $c(z) = \mathcal{U}_{[0, \sqrt{2}a]}(z)$, with mean $\mu = 0.5$. Overall, the sensor is more accurate (smaller σ_w) and more reliable (higher p_D) when it is closer to the object.

There are five moving objects in the surveillance area, positioned relatively close to each other. Their initial state vectors are (the units are omitted): $[800, 600, -1, 0]^T$, $[650, 500, 0.3, 0.6]^T$, $[620, 700, 0.25, -0.45]^T$, $[750, 800, 0, -0.6]^T$, and $[700, 700, 0.2, -0.6]^T$. The objects move according to the constant velocity model. The observer enters the surveillance area at position $(10, 10)$ m. Figure 5.11 shows the location of objects (indicated by asterisks) and the observer path, after $k = 8$ time steps. Current measurements are represented by arcs (dashed blue lines).

The described scenario for numerical simulations is designed for a good reason. In order to achieve a high level of estimation accuracy using the PHD filter, the observer (sensor) motion needs to comply with the following strategy: initially it must move quickly toward the objects (in order to achieve $p_D \approx 1$ and smaller σ_w) and then it has to zigzag in their vicinity (in order to estimate the states of individual objects from range-only measurements). A sensor control policy that does not follow this pattern will result in a poor PHD filter error performance.

The details of the PHD particle filter implementation are described in Sect. 4.2. The number of particles is 200 per estimated number of objects. The probability of survival is set to $p_s = 0.99$, and the transition density is $\pi_{k|k-1}(\mathbf{x}|\mathbf{x}') = \mathcal{N}(\mathbf{x}; \mathbf{F}\mathbf{x}', \mathbf{Q})$, where as in Sect. 5.1.2.1,

$$\mathbf{F} = \mathbf{I}_2 \otimes \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \quad \mathbf{Q} = \mathbf{I}_2 \otimes \varpi \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} \\ \frac{T^2}{2} & T \end{bmatrix} \quad (5.53)$$

with $\varpi = 0.05$ and $T = 1$ s. The object birth intensity is driven by measurements, meaning that target birth particles are placed along all measurement arcs at the previous time, with velocity vectors drawn from the prior on target velocity as $\mathbf{v}_k^{(i)} \sim \mathcal{N}(\mathbf{v}; 0, \mathbf{R})$ with $\mathbf{R} = \text{diag}[0.75^2 \ 0.75^2]$.

The set of admissible control vectors \mathbb{U}_k is computed as follows. If the current position of the observer is $\mathbf{u}_k = [\chi_k \ \psi_k]^\top$, its one-step ahead future admissible locations are:

$$\mathbb{U}_k = \{ (\chi_k + j \Delta_R \cdot \cos(\ell \Delta_\theta), \psi_k + j \Delta_R \cdot \sin(\ell \Delta_\theta)) ; j = 0, \dots, N_R; \ell = 1, \dots, N_\theta \} \quad (5.54)$$

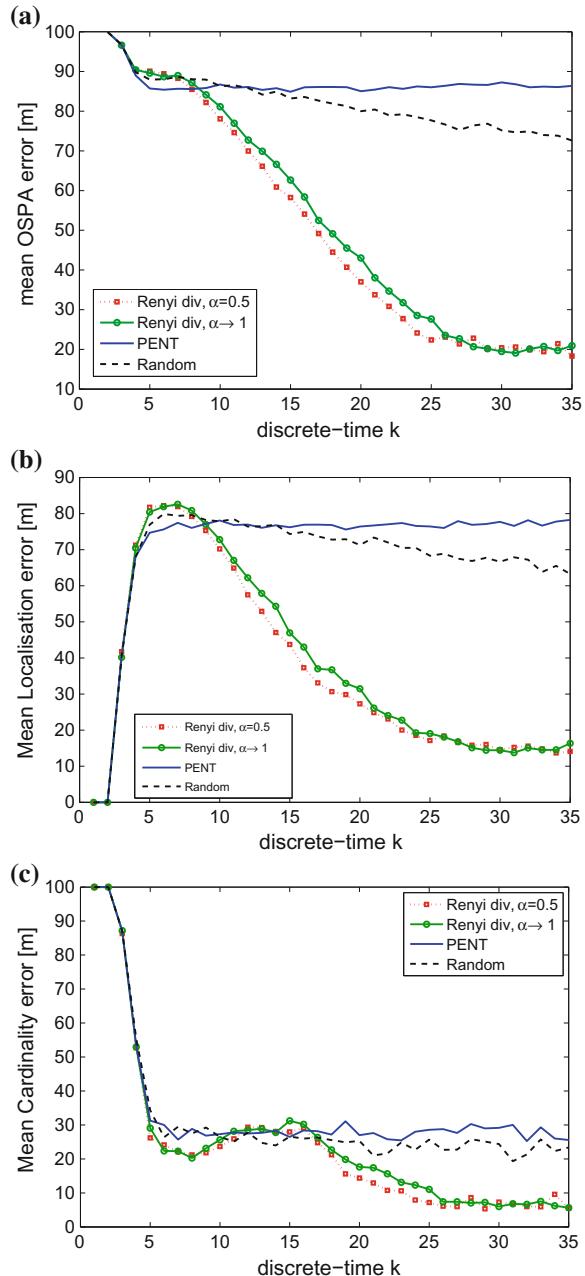
where $\Delta_\theta = 2\pi/N_\theta$ and Δ_R is a conveniently selected radial step size. In this way the observer can stay in its current position (case $j = 0$) or move radially in incremental steps. The following values were adopted in simulations: $N_R = 2$, $N_\theta = 8$, and $\Delta_R = 50$ m. In this way 17 control options are considered at each time step; for each of the admissible control vectors the reward function is computed as described in Sect. 5.2.1. If a control vector $\mathbf{u} \in \mathbb{U}_k$ is outside the specified surveillance area, its reward is set to $-\infty$; in this way the observer is always kept inside the area of interest.

The comparative error performance of the PHD particle filter using three different reward functions for observer control is discussed next. In addition to the Rényi divergence based reward, we considered:

- The uniformly random reward scheme: it selects at random a control vector from the set \mathbb{U}_k .
- The PENT criterion, introduced for control of sensors with a finite field-of-view (FoV) [1, p. 271]. Its goal is to maximize the number of targets seen by the sensors, by selected action $\mathbf{u} \in \mathbb{U}_k$ which maximises $\lambda_1(\mathbf{u}) = \int D_{k+1|k+1}(\mathbf{x}; \mathbf{u}) d\mathbf{x}$, see Eq. (5.45).

The error performance metric adopted for comparison is the OSPA metric, described in Sect. 2.4.3. OSPA metric measures the error between the estimated and the true multi-target states, $\hat{\mathbf{X}}_k$ and \mathbf{X}_k , respectively. Figure 5.12a shows the temporal evolution of the OSPA error (order parameter $p = 2$ and cutoff $c = 100$ m) averaged over 200 Monte Carlo runs for the three contesting reward functions. The corresponding mean localization error and mean cardinality errors are shown in Fig. 5.12b and c respectively. Initially the OSPA error is large (for all three reward functions) reflecting the initial uncertainty about the number of objects and their states. As the time progresses and measurements are collected, the PHD filter using the Rényi divergence as the reward function (with $\alpha = 0.5, \alpha \rightarrow 1$), performs superior to the other two schemes. This is because only the Rényi divergence controls the observer according to the strategy that we anticipated would be the best in this example: it immediately directs the observer toward the moving objects (because this results in higher p_D and lower σ_w) and then requests a zigzag motion

Fig. 5.12 Error performance of the contesting reward functions for PHD filter with sensor control: Rényi divergence with $\alpha = 0.5$, Rényi divergence with $\alpha \rightarrow 1$, PENT and random selection. **a** mean OSPA error (averaged over 200 Monte Carlo runs); **b** Localization component of the OSPA error; **c** Cardinality component of the OSPA error



in their vicinity (a pattern which is necessary for localization using range-only measurements). Comparing the two OSPA error curves for the Rényi divergence, note that the $\alpha = 0.5$ case converges somewhat quicker to the steady-state error (which is approximately attained for $k > 25$) than the $\alpha \rightarrow 1$ case. This observation is in agreement with analysis presented in [24].

In summary, this numerical example confirmed that the PHD particle filter with sensor control using the reward function based on the Rényi divergence performs remarkably well. Only the entry point for the observer was specified in this example; the remaining observer trajectory was determined sequentially by the controller based on received measurements and the current estimate of the posterior PDF. While on every Monte Carlo run the observer trajectory was different, the observer motion follows the general pattern of quick movement toward the objects succeeded by a zigzag motion in their vicinity.

5.3 Sensor Control for the Multi-Target State Particle Filter

5.3.1 Particle Approximation of the Reward Function

The last section of this chapter considers sensor control for the multi-target state particle filter, described in Sect. 4.3. The expression for the reward function, given by (5.1), has no analytic solution and as usual will be implemented numerically.

Note from (2.36) that

$$f_{k|k}(\mathbf{X}_k | \mathbf{Z}_{1:k}, \mathbf{u}_{0:k-1}) = \frac{\varphi_k(\mathbf{Z}_k(\mathbf{u}_{k-1}) | \mathbf{X}_k) f_{k|k-1}(\mathbf{X}_k | \mathbf{Z}_{1:k-1}, \mathbf{u}_{0:k-2})}{\ell_{k|k-1}(\mathbf{Z}_k(\mathbf{u}_{k-1}) | \mathbf{Z}_{1:k-1})} \quad (5.55)$$

where

$$\ell_{k|k-1}(\mathbf{Z}_k(\mathbf{u}_{k-1}) | \mathbf{Z}_{1:k-1}) = \int \varphi_k(\mathbf{Z}_k(\mathbf{u}_{k-1}) | \mathbf{X}) f_{k|k-1}(\mathbf{X} | \mathbf{Z}_{1:k-1}, \mathbf{u}_{0:k-2}) \delta \mathbf{X}. \quad (5.56)$$

Then (5.1) can be written as:

$$\mathcal{J}(\mathbf{u}_{k-1}) = \frac{1}{\alpha - 1} \log \left\{ \frac{\int \varphi_k(\mathbf{Z}_k(\mathbf{u}_{k-1}) | \mathbf{X}_k)^\alpha f_{k|k-1}(\mathbf{X}_k | \mathbf{Z}_{1:k-1}, \mathbf{u}_{0:k-2}) \delta \mathbf{X}_k}{\ell_{k|k-1}(\mathbf{Z}_k(\mathbf{u}_{k-1}) | \mathbf{Z}_{1:k-1})^\alpha} \right\} \quad (5.57)$$

Let us assume that a particle system $\{w_{k|k-1}^{(i)}, \mathbf{X}_{k|k-1}^{(i)}\}_{i=1}^N$ approximates the predicted FISST PDF $f_{k|k-1}(\mathbf{X}_k | \mathbf{Z}_{1:k-1}, \mathbf{u}_{0:k-2})$, see Sect. 4.3. Then the reward (5.57) can be approximated by:

$$\mathcal{J}(\mathbf{u}_{k-1}) \approx \frac{1}{\alpha - 1} \log \frac{\sum_{i=1}^N w_{k|k-1}^{(i)} \varphi_k(\mathbf{Z}_k(\mathbf{u}_{k-1}) | \mathbf{X}_{k|k-1}^{(i)})^\alpha}{\left[\sum_{i=1}^N w_{k|k-1}^{(i)} \varphi_k(\mathbf{Z}_k(\mathbf{u}_{k-1}) | \mathbf{X}_{k|k-1}^{(i)}) \right]^\alpha} \quad (5.58)$$

We need to compute the expectation of the reward (5.58) with respect to the prior measurement FISST PDF $\ell_{k|k-1}(\mathbf{Z}_k(\mathbf{u})|\mathbf{Z}_{1:k-1})$. As usual, we can use the Monte Carlo approximation for this purpose. Let us introduce notation:

$$\gamma_\alpha(\mathbf{Z}_k(\mathbf{u}_{k-1})) = \sum_{i=1}^N w_{k|k-1}^{(i)} \varphi_k(\mathbf{Z}_k(\mathbf{u}_{k-1})|\mathbf{X}_{k|k-1}^{(i)})^\alpha. \quad (5.59)$$

Note that for $\alpha = 1$, $\gamma_1(\mathbf{Z}_k(\mathbf{u}_{k-1}))$ is the Monte Carlo approximation of the prior measurement set FISST PDF $\ell_{k|k-1}(\mathbf{Z}_k(\mathbf{u}_{k-1})|\mathbf{Z}_{1:k-1})$, introduced in (5.56). Then we can approximate the expectation of the reward function as follows [31]:

$$\mathbb{E}[\mathcal{J}(\mathbf{u}_{k-1})] \approx \sum_{j=1}^M \gamma_1(\mathbf{Z}_k^{(j)}(\mathbf{u}_{k-1})) \mathcal{J}(\mathbf{u}_{k-1}) \quad (5.60)$$

$$= \frac{1}{\alpha - 1} \sum_{j=1}^M \gamma_1(\mathbf{Z}_k^{(j)}(\mathbf{u}_{k-1})) \log \frac{\gamma_\alpha(\mathbf{Z}_k^{(j)}(\mathbf{u}_{k-1}))}{\gamma_1(\mathbf{Z}_k^{(j)}(\mathbf{u}_k))^\alpha} \quad (5.61)$$

where $\mathbf{Z}_k^{(j)}(\mathbf{u})$, for $j = 1, \dots, M$, are random set samples from $\ell_{k|k-1}(\mathbf{Z}_k(\mathbf{u})|\mathbf{Z}_{1:k})$ approximated by $\gamma_1(\mathbf{Z}_k(\mathbf{u}))$. Expression (5.60) converges to the true expected value of the reward function as $M, N \rightarrow \infty$.

5.3.2 A Numerical Study

The numerical example is very similar to the one presented in Sect. 5.2.2. For simplicity we focus on the update step, because the sensor control is required only for the purpose of collecting the observation set used in the filter update. Thus we consider only a constant number of static objects, thereby ignoring the prediction step.

An unknown number of objects $n \geq 0$ are present in a specified surveillance area \mathcal{A} . The unknown state of each object $r = 1, \dots, n$ is fully specified by its location in the Cartesian coordinates (x, y) , i.e., the single-object state space $\mathcal{X} \equiv \mathcal{A} \subseteq \mathbb{R}^2$ is two-dimensional and the state vector of object r is $\mathbf{x}_r = [x_r \ y_r]^\top$. A controllable moving sensor (observer) provides noisy range-only scalar measurements $z \in \mathbb{R}$ with the probability of detection $p_D(\mathbf{x})$ and with a certain false alarm rate. The false alarms are modeled as a Poisson RFS: their cardinality distribution is Poisson with mean λ ; for a given cardinality, the false alarms are each IID with the spatial PDF $c(z|\mathbf{x})$. If object $r \in \{1, \dots, n\}$ is detected at time k by the sensor located at $\mathbf{u}_k = [\chi_k \ \psi_k]^\top$, the received measurement is modeled by:

$$z_k = \| \mathbf{x}_r - \mathbf{u}_k \| + w_k \quad (5.62)$$

where $\| \mathbf{x}_r - \mathbf{u}_k \| = \sqrt{(x_r - \chi_k)^2 + (y_r - \psi_k)^2}$ is the distance between the observer and object r , w_k is zero-mean white Gaussian measurement noise with variance σ_w^2 .

In simulations the specified area \mathcal{A} is a square $\mathcal{A} = [0, 1000] \times [0, 1000] \text{ m}^2$. The probability of detection is homogeneous and constant across \mathcal{A} , $p_D = 0.95$; the mean number of false detections are $\lambda = 0.8$; the spatial distribution of false detections is uniform $c(z|\mathbf{x}) = c(z) = \mathcal{U}_{[0, R_{\max}]}(z)$ with $R_{\max} = \sqrt{2} \times 1000 \text{ m}$. All these parameters are known to the estimation algorithm. Furthermore, we ‘place’ $n = 2$ objects in \mathcal{A} at $\mathbf{x}_1 = (300, 600) \text{ m}$ and $\mathbf{x}_2 = (750, 250) \text{ m}$. For the duration of the simulation, the number of objects and their locations remain constant. The algorithm will provide sequentially the multi-object state estimate $\hat{\mathbf{X}}_k$ (from which one can deduce \hat{n}_k , and individual states $\hat{\mathbf{x}}_{k,1}, \dots, \hat{\mathbf{x}}_{k,\hat{n}_k}$) as it receives measurements $Z_k = \{z_{k,1}, \dots, z_{k,m_k}\}$, $k = 1, 2, \dots$, from the observer. The standard deviation of range measurements depends on the distance between the observer located at \mathbf{u}_k and detected object r at \mathbf{x}_r as follows: $\sigma_w(\mathbf{x}_r) = 10 + 4 \times 10^{-5} \|\mathbf{x}_r - \mathbf{u}_k\|^2$.

The multi-target state particle filter is implemented as described in Sect. 4.3. Some useful remarks about the implementation are as follows: the maximum number of objects was fixed to $v_{\max} = 3$; the mixed labeling problem is not explicitly solved (left to be resolved by the particle filter); the major difficulty in the implementation is the enumeration of hypotheses θ required to compute the multi-object likelihood $\varphi_k(\mathbf{Z}_k|\mathbf{X}_k)$. We emphasize that although the objects were constantly present in the specified area \mathcal{A} , the particle filter dynamically at random added and removed object components in multi-object particles $\mathbf{X}_k^{(i)}$. The computation of the multi-object estimate $\hat{\mathbf{X}}_k$ involves the estimation of cardinality \hat{n}_k and individual object states $\hat{\mathbf{x}}_{k,1}, \dots, \hat{\mathbf{x}}_{k,\hat{n}_k}$. Cardinality is estimated using the MAP principle as:

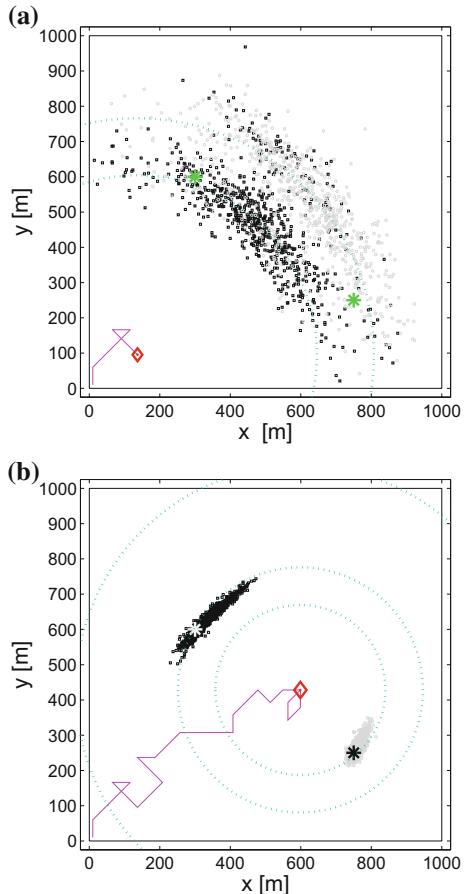
$$\hat{n}_k = \arg \max_{0 \leq j \leq v_{\max}} \left(\sum_{i=1}^N \Delta[j, |\mathbf{X}_k^{(i)}|] \right) \quad (5.63)$$

where $\Delta[i, j] = 1$ if $i = j$ and zero otherwise, is the Kronecker delta function. The individual object states are obtained by simply averaging the particles characterized by the cardinality estimate \hat{n}_k obtained from (5.63). The set of admissible control vectors \mathbb{U}_k is computed in the same manner as described in Sect. 5.2.2. The number of particles are $N = 750$ and the number of samples of the “future measurement set” $\mathbf{Z}_k^{(j)}$ are $M = 500$. The future measurement sets are created assuming $p_D = 1$ and zero false alarm rate. The parameter of Rényi divergence α was set to 0.5.

The error performance of the algorithm is measured using the average OSPA metric, with parameters $p = 2$ and $c = 100 \text{ m}$.

Figure 5.13 shows a single run of the proposed algorithm after (a) $k = 7$ and (b) $k = 24$ discrete-time steps. The zigzag solid line indicates the observer path; the current observer location is denoted by a diamond. The black and gray dots indicate the particles $\mathbf{X}_k^{(i)}$ characterized by cardinality $\hat{n}_k = 2$. The arcs/circles (dotted lines) correspond to the range measurements at time k . The true object locations are indicated with asterisks. The probability $Pr\{n = 2\}$ equals 0.93 and 0.95 at $k = 7$ and $k = 24$, respectively. In a typical run of the algorithm, the sensor is controlled at first toward the mid point between the sources, as illustrated in Fig. 5.13. Afterwards the

Fig. 5.13 Localization with sensor control: **a** after $k = 7$ time steps; **b** after $k = 24$ time steps. The *solid zigzag line* indicates the observer path; the current observer location is denoted by a *diamond* \diamond . The black and gray dots indicate the particles $\mathbf{X}_k^{(i)}$, $i = 1, \dots, N = 750$, characterized by cardinality $\hat{n}_k = 2$. The *arcs/circles (dashed line)* correspond to the range measurements. The true object locations are indicated by asterisks at $(300, 600)$ and $(750, 250)$. Notice the self-resolution of the permutation symmetry at $k = 24$



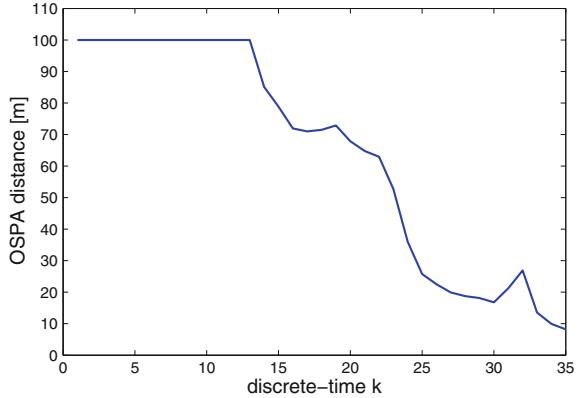
algorithm tends to favor observer paths which approach and circle around one of the objects, followed by a similar motion pattern around the other object.

The OSPA error for this particular run of $k = 1, \dots, 35$ is shown in Fig. 5.14: it reduces with time, as the sensor provides more measurements.

The performance of the proposed sensor control reward is compared next against three alternatives described below.

1. Uniformly random scheme, where action \mathbf{u}_k is chosen randomly from \mathbb{U}_k .
2. The approximate Fisher information gain, which is computed as follows. Suppose the estimate of the multi-object state at k is $\hat{\mathbf{X}}_k = \{\hat{\mathbf{x}}_{k,1}, \dots, \hat{\mathbf{x}}_{k,\hat{n}_k}\}$. Assuming ideal detection ($p_D = 1$ and no false detections), let the measurement set (after taking action \mathbf{u}_{k-1}) be denoted by $Z_k = \{z_{k,1}, \dots, z_{k,\hat{n}_k}\}$, with $z_{k,j} = h_k(\hat{\mathbf{x}}_{k,j}) = ||\hat{\mathbf{x}}_{k,j} - \mathbf{u}_k||$, for $j = 1, \dots, \hat{n}_k$. Furthermore, assume a significant separation of objects in the state space \mathcal{X} , so that one can approximate the multi-object likelihood function $\varphi_k(Z_k(\mathbf{u}_{k-1})|\mathbf{X}_k)$, defined by (2.34), with:

Fig. 5.14 OSPA metric versus time for the run illustrated in Fig. 5.13



$$\varphi_k(Z_k(\mathbf{u}_{k-1})|\hat{\mathbf{X}}_k) \approx \prod_{j=1}^{\hat{n}_k} g_k(z_{k,j}|\hat{\mathbf{x}}_{k,j}) \quad (5.64)$$

where the single-object likelihood $g_k(z|\mathbf{x})$, according to (5.62), is $\mathcal{N}(z; \|\mathbf{x} - \mathbf{u}_{k-1}\|, \sigma_w^2)$. The Fisher information gain is then given by:

$$\mathbf{J}(\mathbf{u}_{k-1}) \triangleq -\mathbb{E} \left[\nabla \nabla^\top \log \varphi_k(Z_k(\mathbf{u}_{k-1})|\hat{\mathbf{X}}_k) \right] \quad (5.65)$$

$$\approx \sum_{j=1}^{\hat{n}_k} \frac{1}{\sigma_k^2(\hat{\mathbf{x}}_{k,j})} \mathbf{H}_{k,j}^\top \mathbf{H}_{k,j} \quad (5.66)$$

where $\mathbf{H}_{k,j}$ is the Jacobian of nonlinear measurement function $h_k(\hat{\mathbf{x}}_{k,j}) = \|\mathbf{x}_{k,j} - \mathbf{u}_{k-1}\|$. The reward function in this scheme is the trace of the matrix $\mathbf{J}(\mathbf{u}_{k-1})$.

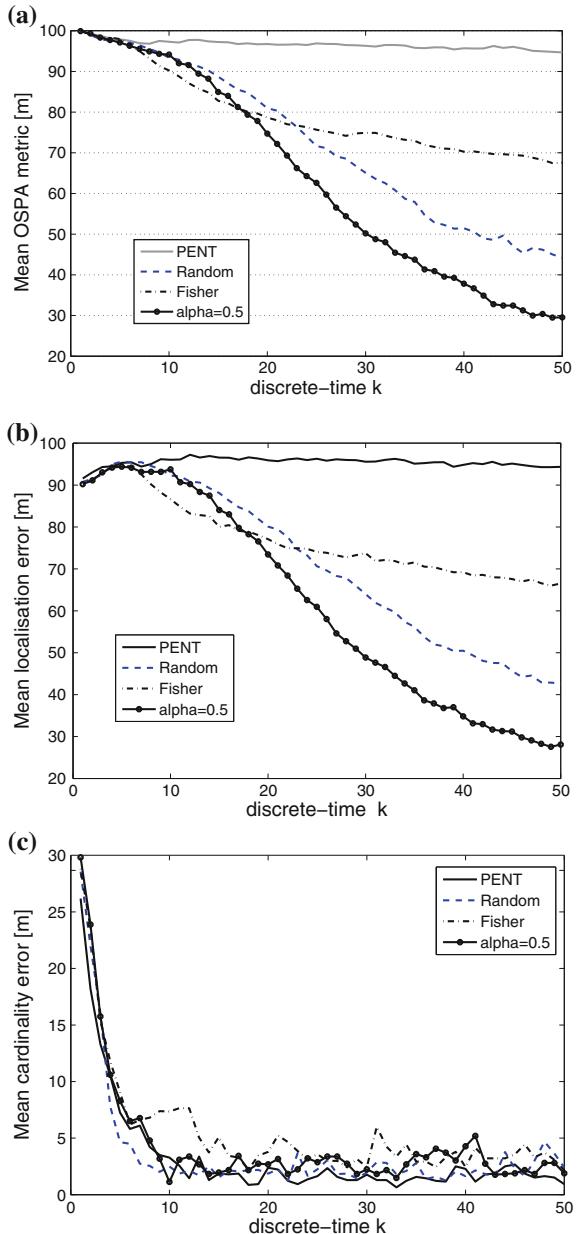
3. The PENT criterion was introduced in Sect. 5.2.2. In order to compute PENT we need first to approximate the PHD from the particle system $\{w_k^{(i)}, \mathbf{X}_k^{(i)}\}_{i=1}^N$ which approximates the posterior FISST PDF $f_{k|k}(\mathbf{X}_k|\mathbf{Z}_{1:k}, \mathbf{u}_{0:k-1})$. This can be done as follows:

$$D_{k|k}(\mathbf{x}; \mathbf{u}_{k-1}) \approx \sum_{i=1}^N w_k^{(i)} \sum_{\mathbf{y} \in \mathbf{X}_k^{(i)}} \delta_{\mathbf{y}}(\mathbf{x}) \quad (5.67)$$

The PENT criterion is then approximately:

$$\int D_{k|k}(\mathbf{x}; \mathbf{u}_{k-1}) d\mathbf{x} \approx \sum_{i=1}^N |\mathbf{X}_k^{(i)}| \cdot w_k^{(i)} \quad (5.68)$$

Fig. 5.15 Error performance (averaged over 200 Monte Carlo runs) of the four contesting reward functions for the particle filter implementation of the exact (full) Bayes multi-object filter: **a** mean OSPA error; **b** the localization component of the OSPA error; **c** the cardinality component of the OSPA error



The OSPA metrics, as well as the cardinality and localization errors, of all four sensor management approaches are averaged over 200 independent Monte Carlo runs, with results shown in Fig. 5.15. Note first from Fig. 5.15 that the main contributor to the OSPA error in this case study is the localization error (cardinality errors are comparatively much smaller). Second, the proposed Rényi-divergence

information-driven sensor control clearly performs the best: its OSPA error reduces over time to the lowest value. Random action selection performs quite well and even better than the Fisher-information and the PENT-based criteria. The Fisher information reduces OSPA quickly at initial stages (between $k = 7$ and $k = 18$ it actually performs the best), but subsequently it does not seem to provide a useful reward for sensor control. The PENT performs very poorly overall; this is not surprising because the PENT is not concerned with the localization uncertainty, which clearly dominates in this example due to the range-only measurements.

References

1. R. Mahler, “Multitarget sensor management of dispersed mobile sensors”, in Theory and algorithms for cooperative systems (D. Grundel, R. Murphrey, and P. Pardalos, eds.), ch. 12, pp. 239–310, World Scientific books, 2004.
2. A. Farina, “Target tracking with bearings only measurements”, *Signal Processing*, vol. 78, pp. 61–78, 1999.
3. B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman filter: Particle filters for tracking applications*. Artech House, 2004.
4. S. Arulampalam, B. Ristic, N. Gordon, and T. Mansell, “Bearings-only tracking of manoeuvring targets using particle filters”, *EURASIP Journal of Applied Signal Processing*, vol. 15, pp. 2351–2365, 2004.
5. S. C. Nardone and V. J. Aidala, “Observability criteria for bearings-only target motion analysis”, *IEEE Trans. Aerospace and Electronic Systems*, vol. 17, no. 2, pp. 162–166, 1981.
6. E. Fogel and M. Gavish, “Nth-order dynamics target observability from angle measurements”, *IEEE Trans. Aerospace and Electronic Systems*, vol. AES-24, pp. 305–308, May 1988.
7. T. L. Song, “Observability of target tracking with bearings-only measurements”, *IEEE Trans Aerospace & Electronic Systems*, vol. 32, no. 4, pp. 1468–1472, 1996.
8. J. M. Passerieux and D. V. Cappel, “Optimal observer maneuver for bearings-only tracking”, *IEEE Trans. Aerosp. Electr. Syst.*, vol. 34, no. 3, pp. 777–788, 1998.
9. J.-P. L. Cadre and S. Laurent-Michel, “Optimizing the receiver maneuvers for bearings-only tracking”, *Automatica*, vol. 35, no. 4, pp. 591–606, 1999.
10. Y. Oshman and P. Davidson, “Optimization of observer trajectories for bearings only target localization”, *IEEE Trans Aerospace and Electronic Systems*, vol. 35, no. 3, pp. 892–902, 1999.
11. Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, 2001.
12. B. Ristic and S. Arulampalam, “Bernoulli particle filter with observer control for bearings only tracking in clutter”, *IEEE Trans Aerospace and Electronic Systems*, vol. 48, July 2012.
13. M. Hernandez, B. Ristic, A. Farina, and L. Timmoneri, “Comparison of two Cramér-Rao bounds for non-linear filtering with $P_d < 1$ ”, *IEEE Trans. Signal Processing*, vol. 52, no. 9, pp. 2361–2370, 2004.
14. C. Jauffret and Y. Bar-Shalom, “Track formation with bearings and frequency measurements in clutter”, *IEEE Trans. Aerospace and Electronic Systems*, vol. 26, no. 6, pp. 999–1010, 1990.
15. A. M. Peterson, “Radio and radar tracking of the Russian Earth satellite”, *Proc. of the IRE*, vol. 45, no. 11, pp. 1553–1554, 1957.
16. S. N. Salinger and J. J. Brandstatter, “Application of recursive estimation and Kalman filtering to Doppler tracking”, *IEEE Trans. Aerospace and Electronic Systems*, vol. 4, no. 4, pp. 585–592, 1970.
17. Y.-T. Chan and F. Jardine, “Target localization and tracking from Doppler-shift measurements”, *IEEE Journal of Oceanic Engineering*, vol. 15, no. 3, pp. 251–257, 1990.

18. D. C. Torney, “Localization and observability of aircraft via Doppler-shifts”, IEEE Trans. Aerospace and Electronic Systems, vol. 43, no. 3, pp. 1163–1168, 2007.
19. Y.-C. Xiao, P. Wei, and T. Yuan, “Observability and performance analysis of bi/multi-static Doppler-only radar”, IEEE Trans. Aerospace and Electronic Systems, vol. 46, no. 4, pp. 1654–1667, 2010.
20. I. Shames, A. N. Bishop, M. Smith, and B. D. O. Anderson, “Analysis of target velocity and position estimation via Doppler-shift measurements”, in Proc. Australian Control Conference, (Melbourne, Australia), pp. 507–512, Nov. 2011.
21. W. Dargie and C. Poellabauer, Fundamentals of Wireless Sensor Networks: Theory and Practice. Wiley, 2010.
22. M. B. Guldogan, D. Lindgren, F. Gustafsson, H. Habberstad, and U. Orguner, “Multiple target tracking with gaussian mixture phd filter using passive acoustic doppler-only measurements”, in Proc. Int. Conf. Information Fusion, (Singapore), July 2012.
23. R. J. Webster, “An exact trajectory solution from Doppler shift measurements”, IEEE Trans. Aerospace and Electronic Systems, vol. 18, no. 2, pp. 249–252, 1982.
24. A. O. Hero, B. Ma, O. Michel, and J. Gorman, “Alpha-divergence for classification, indexing and retrieval”, Tech. Rep. CSPL-328, Comm. and Sig. Proc. Lab., Dept. EECS, The University of Michigan, 2002.
25. T. Kailath, “The divergence and Bhattacharyya distance measures in signal selection”, IEEE Trans. Communication technology, vol. 15, pp. 52–60, Feb. 1967.
26. C. P. Robert and G. Casella, Monte Carlo statistical methods. Springer, 2nd ed., 2004.
27. R. P. S. Mahler, “PHD filters of higher order in target number”, IEEE Trans. Aerospace & Electronic Systems, vol. 43, no. 4, pp. 1523–1543, 2007.
28. R. P. S. Mahler, “Multi-target Bayes filtering via first-order multi-target moments”, IEEE Trans. Aerospace & Electronic Systems vol. 39, no. 4, pp. 1152–1178, 2003.
29. B. Ristic, B. -N. Vo, and D. Clark, “A Note on the Reward Function for PHD Filters with Sensor Control”, IEEE Trans. Aerospace & Electronic Systems, vol. 47, no. 2, pp. 1521–1529, 2011.
30. R. Mahler, “The multisensor PHD filter II: Erroneous solution via “Poisson magic””, in, Proc. SPIE, vol. 7336, 2009.
31. B. Ristic and B. N. Vo, “Sensor control for multi-object state-space estimation using random finite sets”, Automatica, vol. 46, pp. 1812–1818, 2010.

Chapter 6

Multi-Target Tracking

All multi-target filters developed in the framework of random set theory (Sect. 2.4.2) estimate sequentially the number of targets and their individual states, in the form of a multi-object estimate $\hat{\mathbf{X}}_k = \{\hat{\mathbf{x}}_{k,1}, \hat{\mathbf{x}}_{k,2}, \dots, \hat{\mathbf{x}}_{k,n_k}\}$, for $k = 1, 2, \dots$.

Multi-target tracking refers to sequential estimation of the number of targets and their states (positions, velocities, etc.) tagged by a unique label. Hence the output of a tracking algorithm are *tracks*, where a track represents a labeled temporal sequence of state estimates, associated with the same target.

Since multi-target tracking has arguably more practical applications than multi-target filtering, numerous attempts have been made recently to develop multi-target trackers from multi-target random-set based filters. Most of the work was done in the context of the PHD filter. The early approach was to assign a track label to each particle in the PHD particle filter [1, 2], and to let resampling of particles resolve the track labels. Since this approach works well only for targets that are well separated, it became clear soon that inclusion of some sort of data association and track management in the PHD filter is preferable [3–6]. The same argument applies to the Cardinalized PHD filter based tracker, see Ref. [7].

Since data association is necessary for multi-target tracking and the Bernoulli filter is the exact single-target tracker for appearing/disappearing targets in clutter, one obvious solution is to build a multi-target tracker using a combination of data association with a Bernoulli filter [8, 9]. Various data association and track management schemes have been proposed for this purpose, such as the two-dimensional assignment (2DA), linear multi-target (LM) association and the loopy belief propagation.

Before describing how to build multi-target trackers from random-set based filters, this chapter will introduce a mathematically rigorous metric for measuring the distance between two sets of tracks. This metric plays an important role in error performance assessment and comparison of multi-target trackers. Finally, the chapter will present an application to pedestrian tracking using a surveillance video.

6.1 OSPA-T: A Performance Metric for Multi-Target Tracking

In evaluating the performance of a multi-target tracking algorithm the goal is to measure the distance between two sets of tracks: the set of ground truth tracks and the set of estimated tracks, produced by the tracking algorithm under evaluation. Performance evaluation of multi-target tracking algorithms is of great practical importance in the design of tracking systems, parameter tuning and tracker comparisons (for example in tender evaluations). Consequently the topic has been studied extensively, see for example [10, 11], [12, Chap. 13], [13, 14].

The general performance evaluation methodology includes three steps: (1) the creation of a scenario of a certain level of difficulty; (2) the execution of the multi-target tracking algorithm under assessment; (3) the assignment of a performance measure (score) as a measure of the distance between the ground truth and the tracker output. In order to estimate the expected performance, the score is typically averaged over independent Monte Carlo runs. The randomness in each run is caused by the measurements which feed the tracker (a consequence of measurement noise and imperfect detection).

In order to carry out the third step above (assessment), the standard approach to tracker performance evaluation is to first assign the tracker output tracks to the ground truth tracks [10]. Once the assignment is made, a list of operator defined measures of effectiveness (MoEs) are computed. These MoEs describe various aspects of tracking performance, such as [12, Chap. 13]: *timeliness* (e.g. track initiation delay, track overshoot), *track accuracy* (e.g. positional, heading, velocity error), *continuity* (e.g. track fragmentation, track labeling swaps), *false tracks* (their count and duration), to name a few.

The problem a practitioner faces is twofold. The first is how to choose the relevant MoEs and the second is how to combine them into a single score (which is typically required for parameter tuning or in tender evaluations). The choice of relevant MoEs is far from clear, since the traditional MoEs listed above are fairly arbitrary with various authors arguing in favour of different ones. Combination of MoEs is even more questionable from the theoretical point of view, since the typical MoEs are correlated. To illustrate this point, note that a reduction in the track initiation delay typically increases the number of false tracks (conversely, the reduction of the number of false tracks leads to an increase in the track initiation delay). Another important issue is the “transitive” property of the tracker score resulting from the combination of MoEs. This property can be explained as follows. Consider two trackers, A and B, and suppose that tracker A output is “close” (as indicated by the combined MoE) to the ground truth. If tracker B output is “close” to that of tracker A, can we then make the conclusion that tracker B output is also “close” to the ground truth? The answer is positive only if the combined MoE possesses the “transitive” property. It is far from clear if any combination of traditional MoEs would satisfy this property.

In order to overcome the aforementioned shortcomings of the traditional MoEs, it is necessary to define a mathematically rigorous metric for measuring the distance

between two sets of tracks. First of all, such a metric would naturally satisfy the transitive property discussed above, by the virtue of the *triangle inequality* axiom. Furthermore, a metric also has the necessary property that the estimated state is the same as the true state, if and only if the distance between them is zero. These properties are not guaranteed by other MoEs. The development of a metric is also important if we wish to determine the underlying theoretical performance of an algorithm, such as its asymptotic convergence and consistency properties, i.e. whether in the long run the algorithm does converge to the true state. Without a metric, these notions have no meaning.

We introduced in Sect. 2.4.3 the OSPA error as the distance measure between two sets of objects [16]. The OSPA metric has been demonstrated in the context of performance evaluation of multi-target *filtering algorithms*. For target tracking, however, we require a metric on the space of finite sets of tracks, where a track has been defined as a labeled temporal sequence. The tracks are typically of unequal length in time. This section, following [15], presents an adaptation of the standard OSPA metric for this purpose. The resulting metric measures the distance on the joint state-space—target-label domain. In addition, for the purpose of assigning the tracker output to the ground truth, the global OSPA distance between the pairs of tracks is minimized. The resulting metric is mathematically rigorous and consistent.

6.1.1 The Problem and Its Conceptual Solution

6.1.1.1 Problem Specification

Our goal is to define a metric on the space of finite sets of objects called *tracks*, defined over discrete-time support points $\tau = (t_1, t_2, \dots, t_K)$. A track X on τ is a labeled sequence of length K :

$$X = (X_1, X_2, \dots, X_K) \quad (6.1)$$

where $X_k, k = 1, \dots, K$, is either an empty set or a singleton whose element is (ℓ, \mathbf{x}_k) . Here $\ell \in \mathbb{N}$ is the track label, which does not change with time, while \mathbf{x}_k is the time evolving state vector, a point in an N -dimensional *state space* \mathbf{W} : $\mathbf{x}_k \in \mathbf{W} \subseteq \mathbb{R}^N$. The state space typically involves target position and velocity in the Cartesian coordinates, but may also include other target features, such as amplitude, size, shape, or similar. For convenience let us introduce an indicator e_k , which takes the value of one, if the track exists at time t_k , and zero otherwise. Then we can write:

$$X_k = \begin{cases} \emptyset, & \text{if } e_k = 0, \\ \{(\ell, \mathbf{x}_k)\}, & \text{if } e_k = 1. \end{cases} \quad (6.2)$$

Our aim is to define a metric at one particular time instant t_k , $k = 1, \dots, K$. Let the set of all tracks on τ at specific time t_k be denoted \mathbb{X}_k (i.e. $X_k \in \mathbb{X}_k$). Furthermore, let the set of finite subsets of \mathbb{X}_k be denoted by \mathcal{X}_k . We need to define a metric space (\mathcal{X}_k, D) , where function $d : \mathcal{X}_k \times \mathcal{X}_k \rightarrow \mathbb{R}_+ = [0, \infty)$ is a metric which satisfies the following three axioms for all $\mathfrak{X}_k, \mathfrak{Y}_k, \mathfrak{Z}_k \in \mathcal{X}_k$ [17]:

- *identity*: $d(\mathfrak{X}_k, \mathfrak{Y}_k) = 0$ if and only if $\mathfrak{X}_k = \mathfrak{Y}_k$,
- *symmetry*: $d(\mathfrak{X}_k, \mathfrak{Y}_k) = d(\mathfrak{Y}_k, \mathfrak{X}_k)$,
- *triangle inequality*: $d(\mathfrak{X}_k, \mathfrak{Y}_k) \leq d(\mathfrak{X}_k, \mathfrak{Z}_k) + d(\mathfrak{Z}_k, \mathfrak{Y}_k)$.

If, as a matter of convention, we denote by $\mathfrak{X}_k \in \mathcal{X}_k$ the set of tracks representing the ground truth at t_k , and by $\mathfrak{Y}_k \in \mathcal{X}_k$ the set of estimated tracks produced at t_k by the algorithm under assessment, metric $d(\mathfrak{X}, \mathfrak{Y}_k)$ should quantify the overall estimation error of the tracking algorithm at time t_k . In a mathematically rigorous manner, it should combine various aspects of the tracking error performance (e.g. timeliness, track accuracy, continuity, data association, false tracks, etc) into a single metric.

6.1.1.2 Conceptual Solution

The conceptual solution for proposed the multi-target tracking metric is based on the OSPA metric. The OSPA metric on \mathcal{X}_k [16] is a distance between any two sets of objects. In our case the objects are tracks at t_k , and the two sets are:

$$\mathfrak{X}_k = \{(\ell_1, \mathbf{x}_{k,1}), \dots, (\ell_m, \mathbf{x}_{k,m})\} \quad (6.3)$$

$$\mathfrak{Y}_k = \{(s_1, \mathbf{y}_{k,1}), \dots, (s_n, \mathbf{y}_{k,n})\}. \quad (6.4)$$

According to the convention, $\mathfrak{X}_k \in \mathcal{X}_k$ and $\mathfrak{Y}_k \in \mathcal{X}_k$ are the existing ground truth tracks and the tracker estimated tracks at t_k , respectively. The cardinalities of sets \mathfrak{X}_k and \mathfrak{Y}_k , m and n respectively, depend on k due to (6.2). For $m \leq n$, the OSPA distance between \mathfrak{X}_k and \mathfrak{Y}_k is defined as:

$$d_{p,c}(\mathfrak{X}_k, \mathfrak{Y}_k) = \left[\frac{1}{n} \left(\min_{\pi \in \Pi_n} \sum_{i=1}^m \left(d_c(\tilde{\mathbf{x}}_{k,i}, \tilde{\mathbf{y}}_{k,\pi(i)}) \right)^p + (n-m) \cdot c^p \right) \right]^{1/p} \quad (6.5)$$

where $\tilde{\mathbf{x}}_{k,i} \equiv (\ell_i, \mathbf{x}_{k,i})$, $\tilde{\mathbf{y}}_{k,\pi(i)} \equiv (s_{\pi(i)}, \mathbf{y}_{k,\pi(i)})$ and

- $d_c(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) = \min(c, d(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}))$ is the *cut-off distance* between two tracks at t_k , with $c > 0$ being the cut-off parameter;
- $d(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ is the *base distance* between two tracks at t_k ;
- Π_n represents the set of permutations of length m with elements taken from $\{1, 2, \dots, n\}$;
- $1 \leq p < \infty$ is the OSPA metric order parameter.

For the case $m > n$, the definition is $d_{p,c}(\mathfrak{X}, \mathfrak{Y}) = d_{p,c}(\mathfrak{Y}_k, \mathfrak{X}_k)$. If both \mathfrak{X}_k and \mathfrak{Y}_k are empty sets (i.e. $m = n = 0$), the distance is zero. The proof that OSPA distance is indeed a metric, including a number of arguments that it complies with intuition, is given in Schuhmacher et al. [16].

The choice of parameters c and p follows the guidelines of the original OSPA metric [16]. The cut-off parameter c determines the relative weighting given to the cardinality error component against the base distance error component. Larger values of c tend to emphasize cardinality errors and vice versa. Note that the cardinality error is a consequence of an unequal number of tracks in \mathfrak{X} and \mathfrak{Y} , meaning that c can be interpreted as a measure of penalty assigned to missed or false tracks. The order parameter, p , controls the penalty assigned to “outlier” estimates (that are not close to any of the ground truth tracks). A higher value of p increases sensitivity to outliers.

In order to apply the OSPA metric for tracks described above, it remains to define the base distance $d(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ between tracks $\tilde{\mathbf{x}} \equiv (\ell, \mathbf{x})$ and $\tilde{\mathbf{y}} \equiv (s, \mathbf{y})$. The base distance must be a proper metric (i.e. it needs to satisfy the three axioms of a metric) and will be discussed next.

6.1.2 The Base Distance and Labeling of Estimated Tracks

6.1.2.1 The Base Distance Between Two Labeled Vectors

The base distance between two labeled vectors $d(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ is a metric the on space $\mathbb{N} \times \mathbb{R}^N$ and is defined as:

$$d(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) = \left(d_b(\mathbf{x}, \mathbf{y})^{p'} + d_l(\ell, s)^{p'} \right)^{1/p'}, \quad (6.6)$$

where

- $1 \leq p' < \infty$ is the base distance order parameter;
- $d_b(\mathbf{x}, \mathbf{y})$ is the localization base distance, a metric on \mathbb{R}^N , typically adopted as the p' -norm: $d_b(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_{p'}$;
- $d_l(\ell, s)$ is the labeling error, a metric on \mathbb{N} , adopted as follows:

$$d_l(s, t) = \alpha \bar{\delta}[s, t] \quad (6.7)$$

where $\bar{\delta}[i, j]$ is the complement of the Kroneker delta, that is $\bar{\delta}[i, j] = 0$ if $i = j$, and $\bar{\delta}[i, j] = 1$ otherwise.

Parameter $\alpha \in [0, c]$ in (6.7) controls the penalty assigned to the labeling error $d_l(s, t)$ interpreted relative to the localization base distance $d_b(\mathbf{x}, \mathbf{y})$. The case $\alpha = 0$ assigns no penalty, and $\alpha = c$ assigns the maximum penalty.

The base distance between two labeled vectors $d(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$, defined by (6.6), satisfies the axioms of a metric. To prove identity and symmetry is trivial. The proof of the triangle inequality is presented next.

6.1.2.2 The Proof of Triangle Inequality

We want to prove that [20]

$$d(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) \leq d(\tilde{\mathbf{x}}, \tilde{\mathbf{z}}) + d(\tilde{\mathbf{z}}, \tilde{\mathbf{y}}) \quad (6.8)$$

where

$$d(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})^p = d_b(\mathbf{x}, \mathbf{y})^p + \alpha^p \bar{\delta}[l, s] \quad (6.9)$$

and $d_b(\cdot, \cdot)$ is any metric that does not account for the target labels, $p \geq 1$, $\alpha \in [0, c] : c > 0$ and $\bar{\delta}$ is the complement of the Kronecker delta.

As $d_b(\cdot, \cdot)$ is a metric, it meets the triangle inequality if labels are not considered

$$d_b(\mathbf{x}, \mathbf{y}) \leq d_b(\mathbf{x}, \mathbf{z}) + d_b(\mathbf{z}, \mathbf{y}) \quad (6.10)$$

As both sides of the inequality are positive numbers and $p \geq 1$

$$d_b(\mathbf{x}, \mathbf{y})^p \leq (d_b(\mathbf{x}, \mathbf{z}) + d_b(\mathbf{z}, \mathbf{y}))^p \quad (6.11)$$

We also have that

$$\alpha \bar{\delta}[l, s] \leq \alpha \bar{\delta}[l, u] + \alpha \bar{\delta}[u, s] \quad (6.12)$$

As both sides of the inequality are positive numbers and $p \geq 1$

$$(\alpha \bar{\delta}[l, s])^p \leq (\alpha \bar{\delta}[l, u] + \alpha \bar{\delta}[u, s])^p \quad (6.13)$$

Using (6.11) and (6.13)

$$d_b(\mathbf{x}, \mathbf{y})^p + (\alpha \bar{\delta}[l, s])^p \leq (d_b(\mathbf{x}, \mathbf{z}) + d_b(\mathbf{z}, \mathbf{y}))^p + (\alpha \bar{\delta}[l, u] + \alpha \bar{\delta}[u, s])^p \quad (6.14)$$

$$\sqrt[p]{d_b(\mathbf{x}, \mathbf{y})^p + (\alpha \bar{\delta}[l, s])^p} \leq \sqrt[p]{(d_b(\mathbf{x}, \mathbf{z}) + d_b(\mathbf{z}, \mathbf{y}))^p + (\alpha \bar{\delta}[l, u] + \alpha \bar{\delta}[u, s])^p} \quad (6.15)$$

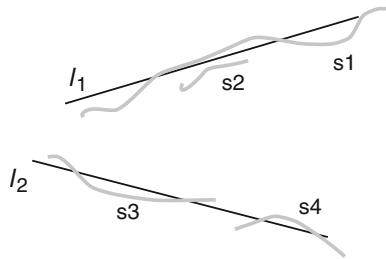
As $p \geq 1$, using the Minkowski integral inequality [20, p. 1917] on the right hand side of (6.15)

$$\begin{aligned} \sqrt[p]{(d_b(\mathbf{x}, \mathbf{z}) + d_b(\mathbf{z}, \mathbf{y}))^p + (\alpha \bar{\delta}[l, u] + \alpha \bar{\delta}[u, s])^p} &\leq \\ \sqrt[p]{d_b(\mathbf{x}, \mathbf{z})^p + \alpha^p \bar{\delta}[l, u]} + \sqrt[p]{d_b(\mathbf{z}, \mathbf{y})^p + \alpha^p \bar{\delta}[u, s]} &\quad (6.16) \end{aligned}$$

Finally, using (6.15) and (6.16), we get

$$\sqrt[p]{d_b(\mathbf{x}, \mathbf{y})^p + \alpha^p \bar{\delta}[l, s]} \leq \sqrt[p]{d_b(\mathbf{x}, \mathbf{z})^p + \alpha^p \bar{\delta}[l, u]} + \sqrt[p]{d_b(\mathbf{z}, \mathbf{y})^p + \alpha^p \bar{\delta}[u, s]} \quad (6.17)$$

Fig. 6.1 Example: two true tracks labeled by l_1 and l_2 (thin black lines) and four estimated tracks, labeled as s_1, \dots, s_4



Using (6.9), the proof is finished

$$d(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) \leq d(\tilde{\mathbf{x}}, \tilde{\mathbf{z}}) + d(\tilde{\mathbf{z}}, \tilde{\mathbf{y}}). \quad (6.18)$$

6.1.2.3 Labeling of Estimated Tracks

Having defined the base distance (6.6), we are still not in the position to apply the OSPA metric for tracks. What remains is to determine the labels of estimated tracks.

In order to explain why this step is necessary, consider an illustrative example shown in Fig. 6.1. There are two true tracks, labeled by l_1 and l_2 , (shown as thin straight lines) and four estimated tracks labeled by s_1, \dots, s_4 (thick solid lines). In order to apply the OSPA metric in the framework described above, it is necessary to assign the labels of true tracks to some of the estimated tracks (in this example the labels are l_1 and l_2) in a globally optimum manner. For the case shown in Fig. 6.1 it would be desirable to assign l_1 to track s_1 , because s_1 is the best approximation (among s_1, s_2, s_3, s_4) of l_1 . For the same reason we should assign l_2 to s_3 , and also make sure that the remaining two estimated tracks (i.e. s_2 and s_4) are assigned labels which are different from l_1 and l_2 . Hence, labeling of estimated tracks is based on the assignment of the tracker output to the ground truth, a preliminary step which is required in all standard tracking evaluation schemes [10].

A simple way to carry out the assignment task is by using one of the existing 2DA algorithms (Munkres, JVC, auction, etc) [12, p. 342]. Suppose the sets of ground truth tracks and estimated tracks are $\{X^{(1)}, \dots, X^{(L)}\}$ and $\{Y^{(1)}, \dots, Y^{(R)}\}$, respectively. Using the existence indicators for tracks, a track $X^{(\ell)}$ at time k , denoted $X_k^{(\ell)}$, is defined as in (6.2). An optimal global assignment λ^* can be determined as follows ($L \leq R$):

$$\begin{aligned} \lambda^* = \arg \min_{\lambda \in \Lambda_R} & \sum_{\ell=1}^L \sum_{k=1}^K \left[e_k^\ell e_k^{\lambda(\ell)} \min(\Delta, \| \mathbf{x}_k^\ell - \mathbf{y}_k^{\lambda(\ell)} \|_2 \right. \\ & \left. + (1 - e_k^\ell) e_k^{\lambda(\ell)} \Delta + e_k^\ell (1 - e_k^{\lambda(\ell)}) \Delta \right] \end{aligned} \quad (6.19)$$

where

- Λ_R is the set of permutations of length L with elements taken from $\{1, 2, \dots, R\}$;
- Δ is the penalty assigned to instantaneous miss or false track;
- e_k^ℓ and $e_k^{\lambda(\ell)}$ are existence indicators for true and λ -assigned estimated tracks.

The case $L > R$ is a trivial modification of (6.19).

A close inspection of (6.19) reveals that the term in the square brackets is in fact the OSPA distance between $X_k^{(\ell)}$ and $Y_k^{(\lambda(\ell))}$, with $p = 2$ and $c = \Delta$ [recall from (6.2) that these two sets are either empty or singletons, hence the form in (6.19)]. Thus the assignment of estimated tracks to ground-truth tracks is carried out by minimising the global OSPA distance between the pairs of tracks, accumulated over the discrete-time points τ . Parameter Δ controls the penalty assigned when at time t_k one of the tracks exists and the other does not. A high value of Δ will favour the assignment of longer duration estimated tracks to the true tracks. If an estimated track has been assigned by λ^* to a true track with label ℓ , then its label is set to ℓ too. Estimated tracks which remain unassigned according to λ^* , are given labels different from all the true track labels.

The basic steps in the computation of the proposed OSPA based metric for tracks (OSPA-T) are given in Algorithm 10. The MATLAB source code for the computation of OSPA metric for tracks is available at <http://randomsets.eps.hw.ac.uk/index.html>.

Algorithm 10 Computation Steps for OSPA-T metric

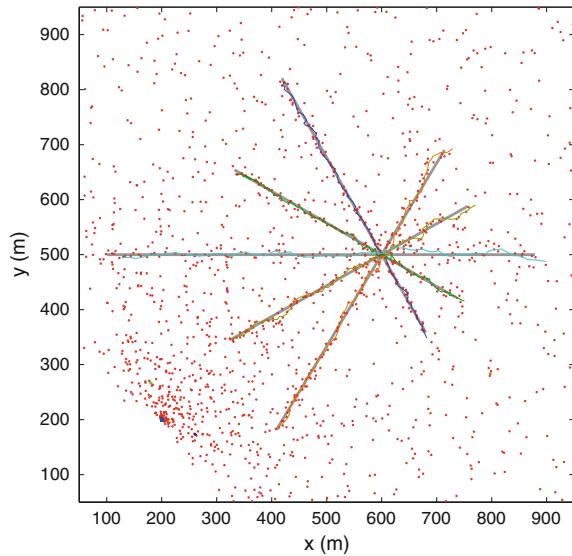
```

1: function OSPA-T( $\{X^{(1)}, \dots, X^{(L)}\}, \{Y^{(1)}, \dots, Y^{(R)}\}$ )
2:   % Label the estimated tracks
3:   For  $j = 1, \dots, R$ , Label[ $Y^{(j)}$ ] =  $I$  (initial value, different from all true track labels)
4:   Find  $\lambda^*$ , the globally optimum assignment of tracks  $\{X^{(1)}, \dots, X^{(L)}\}$  to  $\{Y^{(1)}, \dots, Y^{(R)}\}$ 
5:   For  $i = 1, \dots, L$ , Label[ $Y^{(\lambda^*(i))}$ ] = Label[ $X^{(i)}$ ]
6:   % Compute the distance
7:   For  $k = 1, \dots, K$ 
8:     Form the labeled sets  $\mathfrak{X}_k$  and  $\mathfrak{Y}_k$  at  $t_k$ , see eqs. (6.3) and (6.4)
9:     Compute the OSPA distance at  $t_k$  using (6.5) with the base distance (6.6)
10: end function
```

6.1.3 Numerical Examples

This sections presents a demonstration of the OSPA-T metric using a tracking scenario consisting of five crossing targets, shown in Fig. 6.2. The targets are moving in the $x - y$ plane, from left to right. The total observation interval is $K = 120$ time-steps long. The targets cross (effectively are on top of each other) at $k = 71$. Target durations varies: the start-time indices are $k = 3, 5, 7, 9, 10$; the termination-time indices are $k = 101, 103, 105, 108, 109$. The sensor is placed at (200 m, 200 m) and measures the range and bearing, with zero-mean white Gaussian noise. The measurement noise standard deviations are $\sigma_r = 3$ m and $\sigma_\theta = 1^\circ$. The multi-target tracking algorithm used in demonstration is based on the Bernoulli particle filter

Fig. 6.2 The testing scenario consisting of five crossing targets (crossing angle between the tracks is 30°)



(see Sects. 2.5.1, 4.1) with the LM data association [22] (more details about the algorithm in the next section).

The numerical simulations have been carried out for four cases, listed below. Unless otherwise stated, the parameters of the proposed metric were set to: $\Delta = 100$ m, $\alpha = 25$ m, $c = 25$ m and $p = p' = 1$. It was found that the metric is insensitive to the global assignment parameter Δ , as long as $\Delta > 10$ m. The choice of α will be discussed in the fourth case below. The localization base distance $d_b(\mathbf{x}, \mathbf{y})$ was adopted as the Euclidian distance in the $x - y$ plane. In all figures shown below, the curves of the proposed metric were obtained by averaging over 100 Monte Carlo runs.

Variation of the Crossing Angle

Figure 6.3 shows the proposed metric for two different scenarios: the solid line corresponds to the crossing angle between target trajectories of $\beta = 30^\circ$; the dashed line to the crossing angle of $\beta = 15^\circ$. In both cases the probability of detection was $p_D = 0.9$ and the average number of clutter points (false detection) per scan was $\lambda = 10$. From Fig. 6.3 one can observe the general trend of the OSPA-T distance in this scenario: high values both initially and at the end of the scenario. This is due to track initiation delay and the track termination delay. Furthermore, observe that the OSPA-T metric for the two different scenarios does not change until the target crossing event (after $k = 71$). After this point of time, the distance between the ground truth and the tracker output becomes much higher for the smaller crossing angle β scenario. The result is in accordance with our intuition. There is no reason for the difference in the metric before the crossing, as the tracks are well separated. After the crossing, when some estimated tracks are swapped or broken due to the

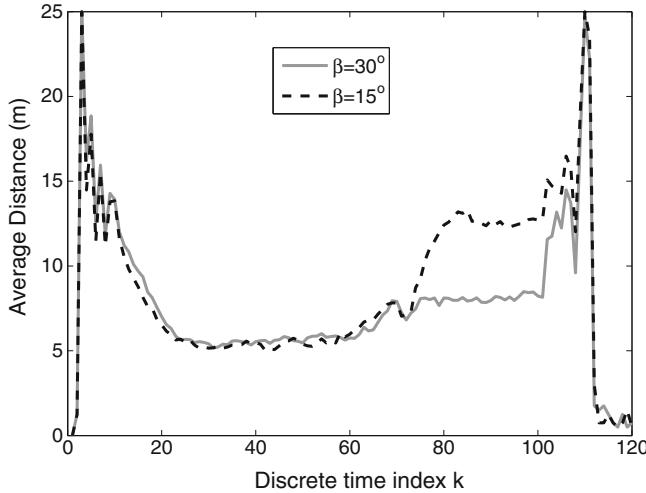


Fig. 6.3 Average OSPA-T distance: different crossing angles β case

difficulties in measurement-to-track association, the small β angle scenario results in the higher OSPA-T distance: this is to be expected as the small β case is much more difficult for tracking.

Variation of the Probability of Detection

Figure 6.4 shows the proposed metric for $p_D = 0.9$ (thin solid line with square markers), $p_D = 0.75$ (solid thick line) and $p_D = 0.6$ (dashed line). In all cases $\beta = 30^\circ$ and $\lambda = 10$. We observe that the higher value of p_D results in the smaller value of OSPA-T distance between the ground truth tracks and estimated tracks, during the entire observation interval. Again this is in accordance with our intuition. The track initiation and track termination are delayed at lower p_D (as reflected in the results). In addition, the track estimation accuracy is worse at lower p_D throughout the observation interval.

Variation of the Density of Clutter

Figure 6.5 compares the proposed metric at $\lambda = 1$ (solid line) and $\lambda = 50$ (dashed line). In both cases $p_D = 0.9$ and $\beta = 30^\circ$. In accordance with intuition, the higher the value of λ , the greater the OSPA distance between the ground truth and estimated tracks. This is due to the increased probability of false tracks and more difficult data association at higher values of λ .

Variation of the Labeling Error Penalty α

The parameters of the simulation setup were $p_D = 0.9$, $\beta = 15^\circ$ and $\lambda = 10$. Figure 6.6 shows the proposed metric using three different values of the metric parameter α : $\alpha = 0$, $\alpha = c/2 = 12.5$ m and $\alpha = c = 25$ m. Note that for $\alpha = 0$, the proposed metric for tracks is equivalent to the standard OSPA metric [16].

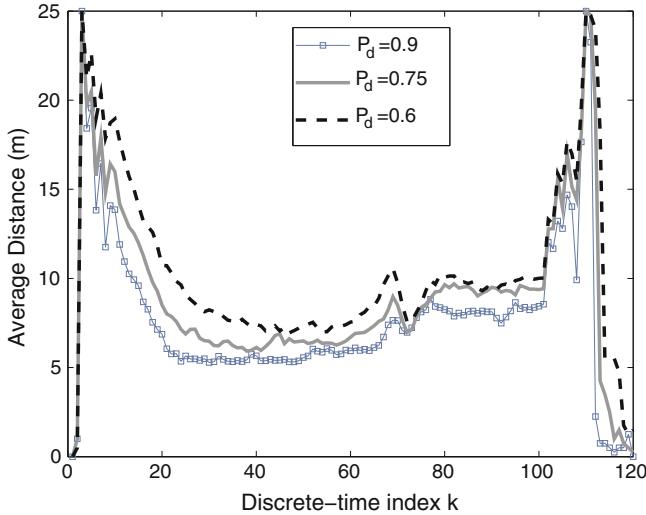


Fig. 6.4 Average OSPA-T: different values of probability of detection p_D

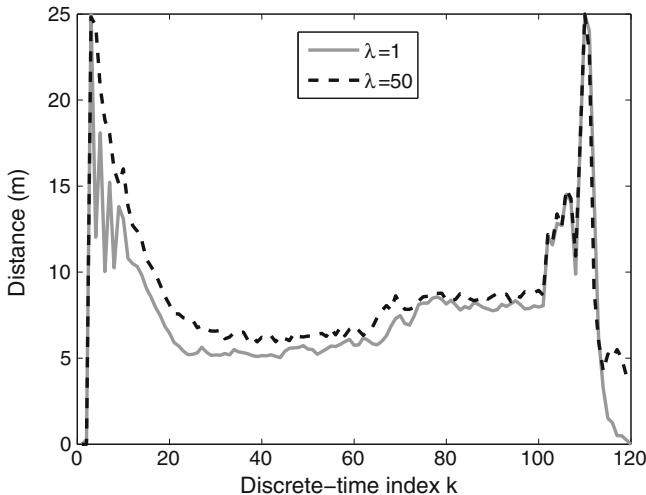


Fig. 6.5 Average OSPA-T: different values of the average number of false detections per scan λ

In the considered tracking scenario, the labeling error is expected to appear after the crossing of targets, when track labels are likely to swap. Figure 6.6 demonstrates this effect and therefore the significance of including the labeling error in the metric.

In summary, this section introduced a mathematically rigorous metric for measuring the distance between two sets of tracks. Typically one set of tracks is the ground truth, while the other is produced by a tracker. The proposed metric is developed for the purpose of performance evaluation of tracking algorithms, with important

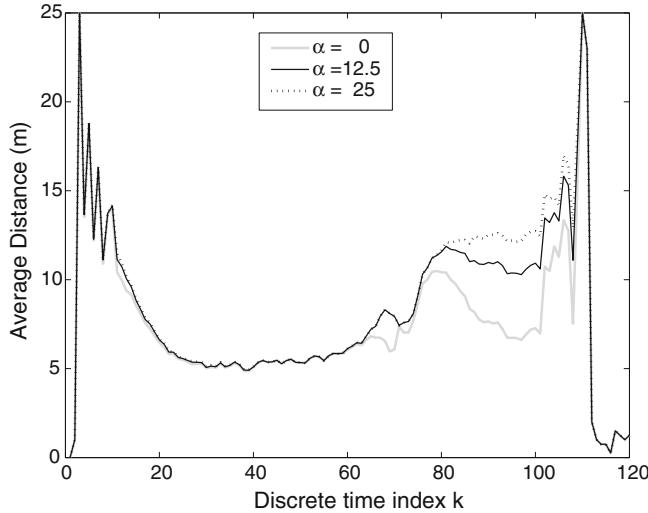


Fig. 6.6 Average OSPA-T: the effect of α parameter which penalizes the labeling error

implications to tracker design, parameter optimization and comparison. The proposed metric captures all aspects of the difference between the two sets of tracks: the cardinality error, the localization error and the labeling error. The inclusion of the labeling error is the novelty in relation to the standard OSPA metric (Sect. 2.4.3). The relative weighting of the three contributing errors to the proposed metric can be controlled by the appropriate selection of its parameters. Numerical analysis demonstrates that the OSPA-T metric behaves in agreement with our intuition and in the manner consistent with our expectations.

6.2 Trackers Based on Random Set Filters

Building trackers from random set filters proves to be a very popular research topic, with almost endless possibilities, see [1–6, 23] and references therein. This section presents four trackers based on random set particle filters, and compares their error performance using the OSPA-T metric.

6.2.1 Multi-Target Trackers Based on the Bernoulli PF

The Bernoulli filter, as the exact Bayes single-target detector/tracker, can be extended to track multiple targets by incorporating a suitable data association technique. The resulting tracker will be suboptimal because computational complexity of optimal

data association grows exponentially with time [12]. We consider two methods of association probability computation: the 2DA, also known as the global nearest neighbour data association [12] and the LM scheme [22]. The resulting trackers, being based on the Bernoulli particle filter, will be referred to as the Bern-PF-2DA and the Bern-PF-LM, respectively.

The pseudo-code of one cycle of the Bern-PF-2DA tracker is presented in Algorithm 11. The input are τ_{k-1} tracks resulting from the previous cycle, denoted \mathfrak{X}_{k-1} , as well as the measurement set at k , $\mathbf{Z}_k = \{\mathbf{z}_{k,1}, \dots, \mathbf{z}_{k,m_k}\}$. A track with index $n = 1, \dots, \tau_{k-1}$ consists of: (1) its label ℓ^n ; (2) the probability of existence $q_{k-1|k-1}^n$, and (3) the spatial PDF $s_{k-1|k-1}^n(\mathbf{x})$ represented by a particle system $\{w_k^{n,(i)}, \mathbf{x}_k^{n,(i)}\}_{i=1}^N$. The tracks are predicted to time t_k in line 2 followed by the computation of the assignment cost matrix \mathbf{C} . The prediction of each track is done independently of the other tracks as described in Sect. 4.1.1. The cost of assigning measurement $\mathbf{z}_{k,j}$, $j = 0, 1, \dots, m_k$ to track $n = 1, \dots, \tau_{k-1}$ is computed as follows:

$$\mathbf{C}[n, j] = -\frac{1 - \delta_k^{n,j}}{1 - \delta_k^{n,j} q_{k|k-1}^n} q_{k|k-1}^n \quad (6.20)$$

where

$$\delta_k^{n,j} = \begin{cases} p_{\text{D}}, & j = 0 \\ \frac{p_{\text{D}} \int g_k(\mathbf{z}_{k,j} | \mathbf{x}) s_{k|k-1}^n(\mathbf{x}) d\mathbf{x}}{\lambda c(\mathbf{z}_{k,j})}, & j = 1, \dots, m_k. \end{cases} \quad (6.21)$$

Matrix \mathbf{C} is therefore of dimension $\tau_{k-1} \times (m_k + 1)$. The case $j = 0$ in (6.21) is for the hypothesis that no measurement is assigned to track n . If the spatial PDF of track n , that is $s_{k|k-1}^n(\mathbf{x})$, is represented by a particle system $\{w_k^{n,(i)}, \mathbf{x}_k^{n,(i)}\}_{i=1}^N$, the integral in (6.21) is approximated as:

$$\int g_k(\mathbf{z}_{k,j} | \mathbf{x}) s_{k|k-1}^n(\mathbf{x}) d\mathbf{x} \approx \sum_{i=1}^N w_k^{n,(i)} \cdot g_k(\mathbf{z}_{k,j} | \mathbf{x}_k^{n,(i)}). \quad (6.22)$$

The next step is to determine the minimum cost assignment (this can be done using a 2DA algorithms [12], e.g. auction algorithm), followed by the update of tracks. Suppose measurement $j_* \in \{0, 1, \dots, m_k\}$ is assigned to track n . Then the track existence of track n is updated as $q_{k|k}^n = -\mathbf{C}[n, j_*]$. The updated spatial PDF is proportional to:

$$s_{k|k}^n(\mathbf{x}) \propto \begin{cases} \frac{p_{\text{D}} g_k(\mathbf{z}_{k,j_*} | \mathbf{x})}{\lambda c(\mathbf{z}_{k,j_*})} s_{k|k-1}^n(\mathbf{x}), & j_* \in \{1, \dots, m_k\} \\ (1 - p_{\text{D}}) s_{k|k-1}^n(\mathbf{x}), & j_* = 0. \end{cases} \quad (6.23)$$

Hence the updated particle weights representing $s_{k|k}^n(\mathbf{x})$ are:

$$w_{k|k}^{n,(i)} \propto \begin{cases} \frac{p_D g_k(\mathbf{z}_{k,j_*} | \mathbf{x}_{k|k-1}^{n,(i)})}{\lambda c(\mathbf{z}_{k,j_*})} w_{k|k-1}^{n,(i)}, & j_* \in \{1, \dots, m_k\} \\ (1 - p_D) w_{k|k-1}^{n,(i)}, & j_* = 0. \end{cases} \quad (6.24)$$

Algorithm 11 Pseudo-code of the Bern-PF-2DA tracker

- 1: **Input:** Tracks \mathfrak{X}_{k-1} ; measurements \mathbf{Z}_k
 - 2: Predict tracks to time t_k to obtain $\mathfrak{X}_{k|k-1}$
 - 3: Compute the assignment matrix \mathbf{C} between $\mathfrak{X}_{k|k-1}$ and \mathbf{Z}_k
 - 4: Solve the 2DA (e.g. auction algorithm) to obtain the best assignment
 - 5: Update $\mathfrak{X}_{k|k-1}$ using \mathbf{Z}_k and the best assignment to obtain $\mathfrak{X}_{k|k}$
 - 6: Manage track life stages to obtain surviving tracks $\mathfrak{X}_{k|k}^s$ and reporting tracks $\mathfrak{X}_{k|k}^r \subseteq \mathfrak{X}_{k|k}^s$
 - 7: Report $\mathfrak{X}_{k|k}^r$
 - 8: Initialize new tracks based on unused measurements in \mathbf{Z}_k : $\mathfrak{X}_{k|k}^n$
 - 9: **Output:** $\mathfrak{X}_k = \mathfrak{X}_{k|k}^s \cup \mathfrak{X}_{k|k}^n$
-

After the update step, in line 6 of Algorithm 11, the management of tracks is carried out. This means that tracks with the probability of existence smaller than the deletion threshold are deleted. At the same time, tracks whose $q_{k|k}^n$ is above the reporting threshold are reported in line 7. The measurements which have not been used in the update step, that is measurements not assigned to any predicted track, are used in line 8 to initialize new tracks with small probability of existence p_b . Each new track receives a new unique track label and a small probability of existence. Finally, in line 9, the new and updated surviving tracks are grouped together and passed on to the next cycle of the tracker for processing.

The second tracker based on the Bernoulli particle filter is using the LM data association [22], and is referred to as Bern-PF-LM. The pseudo-code of one cycle of the Bern-PF-LM tracker is presented in Algorithm 12. The main feature of the LM scheme is that it does not explicitly assign measurements to tracks. The update step, line 4 in Algorithm 12, is carried out for each track independently using standard Bernoulli update equations, see (2.47–2.49) and its particle filter implementation in Sect. 4.1.1. Suppose we want to update track $n \in \{1, \dots, \tau_{k-1}\}$. There is only one difference in relation to the standard Bernoulli filter: the clutter intensity $\lambda \cdot c(\mathbf{z}_{k,j})$, $j = 1, \dots, m_k$, which features in (2.48) and (2.49), is replaced with the following expression:

$$\kappa_k^n(\mathbf{z}_{k,j}) = \lambda c(\mathbf{z}_{k,j}) + \sum_{l=1, l \neq n}^{\tau_{k-1}} p_k^l(\mathbf{z}_{k,j}) \frac{P_k^l(\mathbf{z}_{k,j})}{1 - P_k^l(\mathbf{z}_{k,j})}, \quad (6.25)$$

where

$$p_k^n(\mathbf{z}_{k,j}) = \int g_k(\mathbf{z}_{k,j} | \mathbf{x}) s_{k|k-1}^n(\mathbf{x}) d\mathbf{x} \quad (6.26)$$

and

$$P_k^n(\mathbf{z}_{k,j}) \approx p_{\text{D}} q_{k|k-1}^n \frac{\frac{p_k^n(\mathbf{z}_{k,j})}{\lambda c(\mathbf{z}_{k,j})}}{\sum_{h=1}^{m_k} \frac{p_k^n(\mathbf{z}_{k,h})}{\lambda c(\mathbf{z}_{k,h})}}. \quad (6.27)$$

Effectively the clutter intensity of measurement j at predicted track n is increased by including the contribution from all other targets. As explained earlier, using particle filters for tracking, the expression for $p_k^n(\mathbf{z}_{k,j})$ in (6.26) is approximated according to (6.22). The remaining steps in Algorithm 12 should be self-explanatory.

Algorithm 12 Pseudo-code of the Bern-PF-LM tracker

- 1: **Input:** Tracks \mathfrak{X}_{k-1} ; measurement set \mathbf{Z}_k
 - 2: Predict tracks to time t_k to obtain $\mathfrak{X}_{k|k-1}$
 - 3: Compute LM quantities: $p_k^n(\mathbf{z}_{k,j})$, $P_j^n(\mathbf{z}_{k,j})$ and $\kappa_k^n(\mathbf{z}_{k,j})$, for $j = 1, \dots, m_k$, $n = 1, \dots, \tau_{k-1}$
 - 4: Update $\mathfrak{X}_{k|k-1}$ using the Bernoulli update eqs. with \mathbf{Z}_k and $\kappa_k^n(\mathbf{z}_{k,j})$, for $j = 1, \dots, m_k$, $n = 1, \dots, \tau_{k-1}$, to obtain $\mathfrak{X}_{k|k}$
 - 5: Manage track life stages to obtain surviving tracks $\mathfrak{X}_{k|k}^s$ and reporting tracks $\mathfrak{X}_{k|k}^r \subseteq \mathfrak{X}_{k|k}^s$
 - 6: Report $\mathfrak{X}_{k|k}^r$
 - 7: Initialize new tracks using all $\mathbf{z} \in \mathbf{Z}_k$ such that $\sum_{n=1}^{\tau_{k-1}} P_k^n(\mathbf{z}) < thr$; this results in $\mathfrak{X}_{k|k}^n$
 - 8: **Output:** $\mathfrak{X}_k = \mathfrak{X}_{k|k}^s \cup \mathfrak{X}_{k|k}^n$
-

6.2.2 Multi-Target Trackers Based on the PHD Particle Filter

Recall from Sect. 4.2.5 that a particle approximation of a posterior PDF of a particular target can be obtained by resampling all particles approximating the PHD, using the weights $\{w_{k|k,p}^{(i,j)}, i = 1, \dots, N_{k-1}\}$ computed using a particular measurement $\mathbf{z}_{k,j}$. Recall that $W_{k,p}^{(j)} = \sum_{i=1}^{N_{k-1}} w_{k|k,p}^{(i,j)} \leq 1$ is effectively the probability of existence of the target that is updated by measurement $\mathbf{z}_{k,j}$. This is the rationale for building a multi-target tracker using the PHD particle filter as a basis.

Each target track of the multi-target tracker is effectively represented by its own PHD. The total PHD at time k can then be expressed as:

$$D_{k|k}(\mathbf{x}) \approx \sum_{n=1}^{\tau_k} D_{k|k}^n(\mathbf{x})$$

where $D_{k|k}^n(\mathbf{x})$ is the PHD of track $n = 1, \dots, \tau_k$, such that $W_k^n = \int D_{k|k}^n(\mathbf{x}) d\mathbf{x} \leq 1$. Each track consists of a unique label and a particle system $\{w_{k|k}^n, \mathbf{x}_{k|k}^{n,(i)}\}_{i=1}^N$ which approximates its local PHD $D_{k|k}^n(\mathbf{x})$. The probability of existence can be computed from the weights as $W_{k|k}^n = \sum_{i=1}^N w_{k|k}^{n,(i)} \leq 1$.

The PHD filter update prediction and update equations are applied for each target track separately and independently of other tracks. Again we develop two versions of the tracker, one using the 2DA and the other based on the LM scheme in the update. The update equation for each track $n \in \{1, \dots, \tau_{k-1}\}$ is given by (4.28), implemented as in Sect. 4.2.3.

When using the 2DA algorithm, only $\mathbf{z} \in \mathbf{Z}_k$ which has been associated with the track is used in the update Eq. (4.28). If no measurement is associated with the track, then the track is updated as if the measurement set \mathbf{Z}_k in (4.28) is empty. The cost of association of track $n \in \{1, \dots, \tau_{k-1}\}$ with measurement $\mathbf{z}_{j,k}$, where $j \in \{0, 1, \dots, m_k\}$, is computed based on (4.28) as:

$$\mathbf{C}[n, j] = \begin{cases} -\frac{\int p_D(\mathbf{x}) g_k(\mathbf{z}_{j,k} | \mathbf{x}) D_{k|k-1}^n(\mathbf{x}, 0) d\mathbf{x}}{\kappa_k(\mathbf{z}_{j,k}) + \langle g_k(\mathbf{z}_{j,k} | \cdot), \gamma_{k|k-1} \rangle + \langle p_D g_k(\mathbf{z}_{j,k} | \cdot), D_{k|k-1}^n(\cdot, 0) \rangle} & j = 1, \dots, m_k \\ -\int [1 - p_D(\mathbf{x})] D_{k|k-1}^n(\mathbf{x}, 0) d\mathbf{x} & j = 0. \end{cases} \quad (6.28)$$

Since the local PHD of track n , $D_{k|k-1}^n(\mathbf{x}, 0)$, is represented by a particle system, as usual, the integrals in (6.28) are approximated by sums.

In the case of the LM scheme, the entire \mathbf{Z}_k measurement set is used in the update Eq. (4.28), however the clutter intensity $\kappa_k(\mathbf{z})$ is replaced by (6.25).

The trackers built in the described manner are referred to as PHD-PF-2DA and PHD-PF-LM. Track initiation and track life management is implemented as described in Sect. 6.2.1.

6.2.3 Error Performance Comparison Using the OSPA-T Error

The error performance of the four described multi-target trackers is evaluated using the simulation setup similar to the one described in Sect. 6.1.3. The total observation interval is $K = 120$ time steps long, with five targets moving from left to right and intersecting at $k = 71$ at angle $\beta = 15^\circ$. The targets appear and disappear at different times, however from $k = 10$ to $k = 101$ all five targets are present. The probability of detection is $p_D = 0.9$, clutter density uniform over the observation space with $\lambda = 10$. The standard deviation of measurements is $\sigma_r = 0.2$ m for range and $\sigma_\theta = 3^\circ$ for azimuth.

The averaged OSPA-T metrics of the four trackers are shown in Fig. 6.7. Averaging was done over 100 Monte Carlo runs. The logarithmic scale is used for y-axis in order to emphasize the differences. The time and ensemble averaged OSPA-T error for all

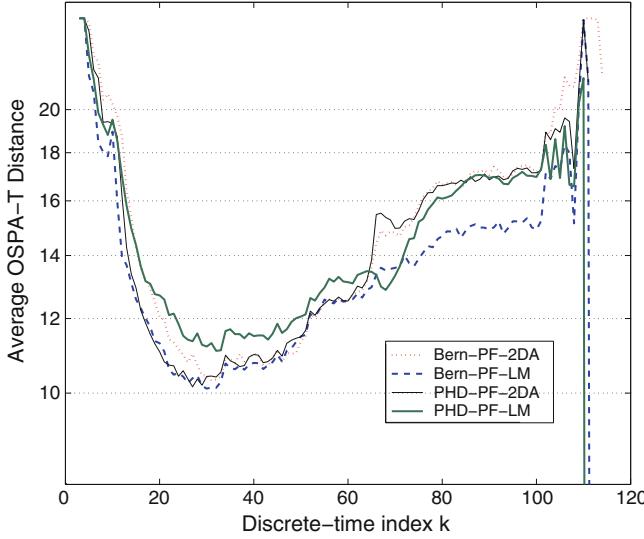


Fig. 6.7 Error performance of four random-set based trackers: average OSPA-T

Table 6.1 Time and ensemble average OSPA-T for the four contesting trackers

Bern-PF-2DA	Bern-PF-LM	PHD-PF-2DA	PHD-PF-LM
14.4	12.6	13.4	13.3

four trackers is given in Table 6.1 (entries in the table obtained by summing all values in Fig. 6.7 and dividing by $K = 120$).

According to Fig. 6.7 and Table 6.1 the clear winner, as the most accurate tracker, is the Bernoulli PF with LM data association (Bern-PF), described in Algorithm 12. It outperforms the other three trackers after the crossing of the targets (when data association is particularly difficult). In the first half of the scenario it performs equally good as the PHD-PF-2DA. This is quite a remarkable tracker, because its computational complexity grows linearly with the number of targets. While standard trackers, such as the multi-dimensional assignment and multi-hypothesis tracker [12], are expected to outperform the Bern-PF-LM, they are also computationally more expensive (their computational complexity grows exponentially with the number of targets).

6.3 Application: Pedestrian Tracking

Next we apply random-set based multi-target trackers to track pedestrians observed by a surveillance camera. Typically pedestrian detections are obtained using the histogram of oriented gradients (HOG) detector [24], trained using either head

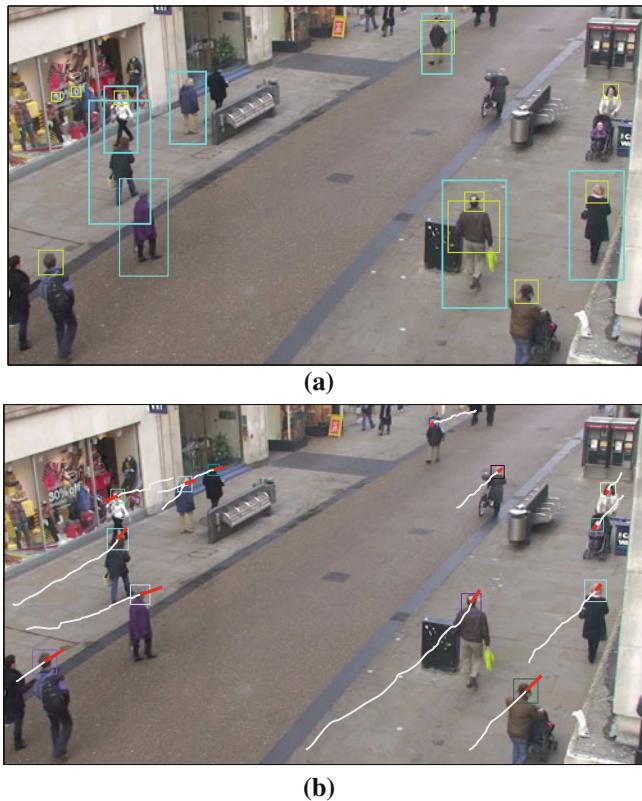


Fig. 6.8 Frame 320 from the dataset used for performance evaluation. **a** Head and body detections (rectangles). **b** Tracker output (the result of recursive processing of 79 previous frames and the current frame); track trajectories indicated by white lines; black lines are velocity vectors; squares indicate estimated positions of heads

images (for head detections) or body images (for body detections). An example of head and body detections is shown in Fig. 6.8a. Head and body detections are unreliable in the sense that not all pedestrians are detected in every frame and the chance of false detections is quite real, with the spatial density of false detections typically non-uniform. Three tracking algorithm, all based on ransom sets, are tested for this application. A typical tracker output is shown in Fig. 6.8b.

6.3.1 Video Dataset and Detections

The video dataset and the (hand labelled) ground truth are downloaded from the website [25]. The dataset is a video of an occasionally busy town center street (the

average number of pedestrians visible at any time is 16). The video was recorded in high definition (1920×1080 pixels at 25 frames per second). Only the first 4,500 frames of the video are used for performance evaluation. Figure 6.8 displays the frame number 320 of the dataset.

Head detection and pedestrian body detection algorithms were applied to each frame in the video sequence. The fastHOG GPU implementation [26] of the HOG detection algorithm [23] was used for both detectors. The HOG detector applies a sliding window to the image at a regular grid of locations and scales, and classifies each sub-window as containing or not containing an object (head or body of a pedestrian). Classification is performed using a pre-trained linear Support Vector Machine, the input to which is a set of block-wise histograms of image gradient orientation. A classification threshold of 0.75 was used for both detectors. Sliding window detectors tend to give multiple detections for one object due to their tolerance to shift and scale, so a post-processing step groups overlapping detections.

The head and body of pedestrian detections have some complementary characteristics. Since the HOG detector is only partially tolerant to occlusions, the head detector tends to have a higher probability of detection than the body detector, as the heads are generally more visible than whole bodies in a surveillance video. On the other hand, pedestrian textures are more distinctive than head textures, so the head detector tends to have a higher false alarm rate, picking up on round-like background objects such as clocks and signs. In addition, the pedestrian body detector is more able to detect people at a distance where the head becomes too small in the image.

Head and body detections are treated as if they are independent. A pedestrian tracking algorithm can then be regarded as a centralized multi-source fusion node, where one source of detections is the head detector while the other is the body detector. In order to make both types of detections compatible, the rectangles corresponding to body detections are *converted* to head-like detections. In this way, both “independent sources” of detections provide head detections. The conversion is done as follows. Suppose a body detection rectangle is specified by its upper-left corner (χ_b, η_b) , width w_b and height h_b . Then its corresponding head-like detection has the upper-left corner coordinates computed as: $\chi'_h = \chi_b + 0.325 w_b$ and $\eta'_h = \eta_b + 0.09 h_b$. The width and height of a head-like detection are $w'_h = 0.35 w_b$ and $h'_h = 0.19 h_b$, respectively.

6.3.2 Description of Algorithms

The state vector of a single pedestrian is adopted for all algorithms as $\mathbf{x} = [x \ \dot{x} \ y \ \dot{y}]^\top$, where (x, y) is the position (in pixels) of the pedestrian head centroid and (\dot{x}, \dot{y}) is its velocity vector (in pixels/s). The number of objects from frame to frame varies. The random set of *head* detections at frame k is denoted $\mathbf{Z}_k^{(1)} = \{\boldsymbol{\xi}_{k,1}^{(1)}, \dots, \boldsymbol{\xi}_{k,m_k^{(1)}}^{(1)}\}$. Accordingly, the random set of body detections converted to heads is

$\mathbf{Z}_k^{(2)} = \{\boldsymbol{\xi}_{k,1}^{(2)}, \dots, \boldsymbol{\xi}_{k,m_k^{(2)}}^{(2)}\}$. Each detection $\boldsymbol{\xi} \in \mathbf{Z}_k^{(i)}$, $i = 1, 2$, thus represents a rectangle specified by the upper-left corner, width and height, i.e. (χ, η, w, h) . The adopted pedestrian motion model is the nearly constant velocity model, see Sect. 5.1.2.1.

Algorithms 1 and 2 are both multi-target tracking algorithms based on the Bernoulli particle filters with LM data association, described in Sect. 6.2.1. The only difference is in their treatment of measurements.

Since the state vector of a pedestrian head does not contain the size of the object, then head detections (from both sources) can be interpreted as instances of *imprecise* measurements: they represent rectangular regions (two-dimensional intervals) within which the true head centroid is located. As such, they can be modelled as random closed sets (rather than random variables), as we discussed in Sect. 2.3.1. Algorithm 1 treats head detections (from both head and body detectors) as imprecise measurements and therefore involves the generalized likelihood function. Let $\boldsymbol{\xi} \in \mathbf{Z}_k^{(i)}$, be a head detection from source $i = 1, 2$ (head or body detector, respectively) originating from a pedestrian head in the state \mathbf{x} . The likelihood function used in Algorithm 1 is then given by (2.16). The measurement function $h(\mathbf{x})$ in this case is linear because detections represent the position of the head. Thus $h(\mathbf{x}) = \mathbf{Hx}$ where $\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$. The covariance matrix Σ which features in (2.16) is adopted as $\Sigma^{(i)} = \text{diag}[\sigma_x^{(i)2}, \sigma_y^{(i)2}]$ for the two types of head detections, with $\sigma_x^{(1)} = \sigma_y^{(1)} = 1$ and $\sigma_x^{(2)} = \sigma_y^{(2)} = 25$.

Algorithm 2 converts each head detection into a precise measurement, that is a point representing the centroid of the measurement rectangle. Thus $\boldsymbol{\xi} \in \mathbf{Z}_k^{(i)}$ is converted into the standard measurement $\mathbf{z} = [\chi + w/2, \eta + h/2]^\top$, with the associated covariance matrix $\mathbf{R} = \text{diag}[(w/6)^2, (h/6)^2]$. The likelihood function of \mathbf{z} which originates from a head in state \mathbf{x} is adopted as:

$$g_k^{(i)}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \mathbf{Hx}, \mathbf{R}).$$

Algorithm 3 is an adaptation of the Cardinalized PHD filter (Sect. 2.5.2) for multi-target tracking [7]. The CPHD filter has been implemented using the Gaussian sum approximation [29] with adaptive birth intensity. Head detections are converted to standard point measurements, as explained above. Additional data association logic is implemented to deal with track labeling. The key idea is to form the clusters of targets, and to apply the CPHD filter update to each cluster separately. The update uses every available detection (measurement) to calculate the *weight* of the track-to-measurement association. The weight of no-measurement association is also computed. Finally the weights are used to form an association matrix which is solved using the 2DA algorithm. At last each predicted track is updated with the measurement assigned by the assignment algorithm. Since we have at our disposal two types of detections ($Z_k^{(i)}$, $i = 1, 2$), the update step in Algorithm 3 is applied twice, first using $Z_k^{(1)}$ and then using $Z_k^{(2)}$. Although this is not an optimal approach [27], it has been suggested as a reasonable approximation.

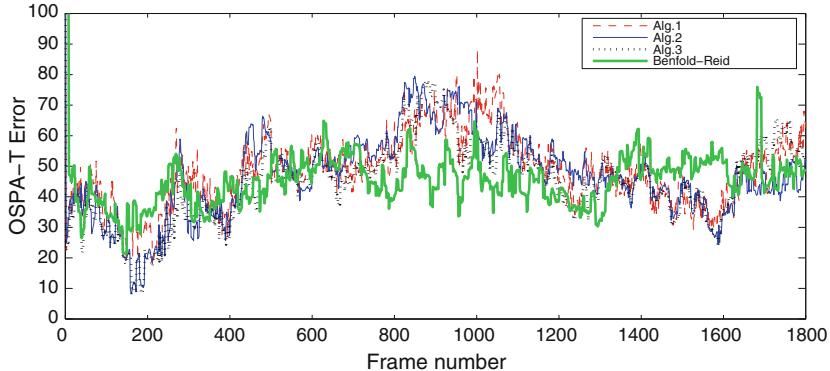


Fig. 6.9 The performance of pedestrian tracking algorithms: OSPA-T error for the first 1,800 frames of the total of 4,500. The parameters of OSPA-T metric: $p = p' = 1$, $c = 100$ and $\alpha = 75$

The density of clutter (false detections) is estimated from a segment of the dataset. Two clutter maps are formed: one for head detections and the other for body-converted-to-head detections. All three algorithms used identical clutter maps. The probability of detection was set to $P_D^{(1)} = 0.58$ and $P_D^{(2)} = 0.52$.

6.3.3 Numerical Results

Evaluation of pedestrian tracking error performance has been carried out using the OSPA-T metric. The localization base distance of OSPA-T error, $d_b(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|$, only takes into account the positional error (i.e. neglecting the velocity error). Figure 6.9 shows the resulting OSPA-T error for the three random-set based tracking algorithms. The parameters of the OSPA-T metric used in evaluation were: $p = p' = 1$, $c = 100$ and $\alpha = 75$. Fig. 6.9 also shows, as a guideline, the OSPA-T error of the Benfold-Reid (BR) algorithm [28], whose tracking results are available online [25]. We point out that the comparison between the BR algorithm and the three random-set based trackers is not fair because the BR algorithm is a smoother (operates as a batch algorithm over a sliding window of image frames) and does not use body/head detections in every frame.

Figure 6.9 shows the OSPA-T error during the first 1,800 frames (of the total of 4,500). From this figure one can observe that the ranking of algorithms according to OSPA-T metric varies with time. For example, from the frame number 800 to 1,100, the BR algorithm is far superior than the random-set based trackers, but the opposite is true from the frame number 1,400 to 1,600. In order to obtain an overall ranking, the time averaged OSPA-T error has be computed: its values for Algorithms 1, 2, 3 and the BR algorithm are 45.2, 42.8, 40.7 and 40.4, respectively. The conclusion is that the most accurate of the three random-set tracking algorithms is Algorithm 3.

Furthermore, it appears that the imprecise measurement model is not justified in the adopted context: the transformation of head and body-to-head rectangles (imprecise detections) into random precise measurement points provides better tracking results. This can be explained by the nature of head and body-to-head rectangular detections; it has been observed that if a detection is not false, then its rectangular center is a very accurate estimate of the center of a pedestrian head. Thus the likelihood (2.16), which is based on the interpretation that the true head centroid is somewhere inside the rectangle, appears to be too cautious and consequently does not use the full information content of each measurement.

The OSPA-T error computations are repeated for the case when α parameter of the OSPA-T metric was set to zero (i.e. no penalty for labeling errors). This case corresponds to the original OSPA error proposed in Schuhmacher et al. [16]. The obtained time averaged OSPA-T error for Algorithms 1, 2, 3 and the BR algorithm in this case were 34.1, 29.5, 27.4, and 30.2, respectively. Again Algorithm 3 performs the best among the random-set based trackers, and even outperforms the BR algorithm. This result reveals that the problem with Algorithm 3 is the lack of track consistency—it creates too many broken tracks with different labels, which by adopting $\alpha = 0$ in OSPA-T is not penalized. Track consistency in general can be improved by smoothing over multiple image frames.

References

1. K. Panta, “Improved probability hypothesis density (PHD) filter for multitarget tracking”, in *Proc. Int. Conf. Intelligent Sensing and Inform. Processing*, (Bangalore, India), pp. 213–218, 2005.
2. D. E. Clark, J. Bell, Y. de Saint-Pern, and Y. Petillot, “PHD filter for multi-target tracking in 3D sonar”, in *Proc. IEEE OCEANS-05-Europe*, (Brest, France), pp. 265–270, 2005.
3. L. Lin, Y. bar Shalom, and T. Kirubarajan, “Track labelling and PHD filter for multitarget tracking.” *IEEE Trans Aerospace and Electronic Systems*, vol. 42, no. 3, pp. 778–795, 2006.
4. E. Maggio, M. Taj, and A. Cavallaro, “Efficient multitarget visual tracking using random finite sets”, *IEEE Trans. Circuits & Systems for Video Technology*, vol. 18, no. 8, pp. 1016–1027, 2008.
5. D. G. Danu, T. Lang, T. Kirubarajan, “Assignment-based particle labeling for PHD particle filter”, in, *Proc. SPIE*, vol. 7445, 2009.
6. T. Wood, D. Clark, and B. Ristic, “Efficient resampling and basic track continuity for the SMC-PHD filter”, in *Proc. Cognitive systems with Interactive Sensors (CogIS)*, (Thales UK, Crawley), Nov. 2010.
7. Y. Petetin, D. Clark, B. Ristic, and D. Maltese, “A tracker based on a CPHD filter approach for infrared applications”, in *Proc. SPIE*, vol. 8050, 2011.
8. B. Ristic, B.-N. Vo, and D. Clark, “Performance evaluation of multi-target tracking using the OSPA metric”, in *Proc. Int. Conf. Information Fusion*, (Edinburgh, UK), July 2010.
9. J. L. Williams, “Experiments with graphical model implementations of multiple target multiple bernoulli filters”, in *Proc. 7th Int. Conf. Intell. Sensors, Sensor Networks and Inform. Proc. (ISSNIP)*, (Adelaide, Australia), pp. 532–537, Dec. 2011.
10. B. E. Fridling and O. E. Drummond, “Performance evlauation methods for multiple target tracking algorithms”, in *Proc. SPIE, Signal and Data Processing of Small Targets*, vol. 1481, pp. 371–383, 1991.

11. B. Ristic, “A Tool for Track-While-Scan Algorithm Evaluation”, in *Proc. Information, Decision, Control (IDC99)*, (Adelaide, Australia), pp. 105–110, Feb. 1999.
12. S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Artech House, 1999.
13. S. B. Colegrove, L. M. Davis, and S. J. Davey, “Performance assessment of tracking systems”, in *Proc. Intern. Symp. Signal Proc. and Applic. (ISSPA)*, (Gold Coast, Australia), pp. 188–191, Aug. 1996.
14. R. L. Rothrock and O. E. Drummond, “Performance metrics for multi-sensor, multi-target tracking”, in *Proc. SPIE, Signal and Data Processing of Small Targets*, vol. 4048, pp. 521–531, 2000.
15. B. Ristic, B. N. Vo, D. Clark and B. T. Vo, “A Metric for Performance Evaluation of Multi-Target Tracking Algorithms”, *IEEE Trans. Signal Processing*, vol. 59, pp. 3452–3457, July 2011.
16. D. Schuhmacher, B.-T. Vo, and B.-N. Vo, “A consistent metric for performance evaluation of multi-object filters”, *IEEE Trans. Signal Processing*, vol. 56, pp. 3447–3457, Aug. 2008.
17. E. Kreyszig, *Introductory functional analysis with applications*. Wiley, 1989.
18. L. Svensson, D. Svensson, and P. Willett, “Set JPDA algorithm for tracking unordered sets of targets”, in *Proc. 12th Int. Conf. Information Fusion*, (Seattle, USA), ISIF, July 2009.
19. 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 Int. Conf. Information Fusion*, (Edinburgh, UK), ISIF, July 2010.
20. Á. F. García-Fernández and B. Ristic, “Erratum to ‘A metric for performance evaluation of multi-target tracking algorithms’”, arXiv, Oct. 2012.
21. E. W. Weisstein, *CRC Concise encyclopedia of mathematics*. Chapman & Hall/CRC, 2nd ed., 2003.
22. D. Musicki and B. L. Scala, “Multi-target tracking in clutter without measurement assignment”, *IEEE Trans. Aerospace and Electronic Systems*, vol. 44, pp. 877–896, July 2008.
23. J. Yang and H. Ji, “A novel track maintenance algorithm for PHD/CPHD filter”, *Signal Processing*, vol. 92, pp. 2371–2380, 2012.
24. N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection”, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2005.
25. U. o. O. Active Vision Group, Dept. Engineering Science, “Town centre dataset”. <http://www.robots.ox.ac.uk/ActiveVision/Research/Projects/2009bbenfoldheadpose/project.html#datasets>, Accessed Dec. 2011
26. V. Prisacariu and I. Reid, “fastHOG - a real-time GPU implementation of HOG”, tech. rep., University of Oxford, 2009.
27. R. Mahler, “The multisensor PHD filter I: General solution via multitarget calculus,” in *Proc. SPIE*, vol. 7336, 2009.
28. B. Benfold and I. D. Reid, “Stable multi-target tracking in real-time surveillance video”, in *Proc Computer Vision and Pattern Recognition (CVPR)*, 2011.
29. B.-T. Vo, B.-N. Vo, and A. Cantoni, “Analytic implementations of the cardinalized probability hypothesis density filter”, *IEEE Trans. Signal Processing*, vol. 55, no. 7, pp. 3553–3567, 2007.

Chapter 7

Advanced Topics

This chapter is devoted to two tracking problems where random set models are particularly useful, because they offer elegant solutions unmatched by the standard approaches. At first, a formulation of an exact solution and its particle filter approximation are presented for the problem of joint detection and tracking of an extended object in clutter. Then, the calibration of multi-target tracking systems is solved using only noncooperative target measurements. The calibration algorithm is demonstrated in the context of sensor bias estimation.

7.1 Bernoulli Filter for Extended Target Tracking

Standard tracking algorithms [1] assume that sensors can report at most one measurement per target (moving object). A target which results in zero or one detection is often referred to as the *point* target. In many practical situations the point target assumption is not appropriate. For example, high-resolution radars are able to resolve individual features of the target, known as scattering or measurement generating points, and typically report multiple detections per object. In computer vision this is particularly prevalent, because an object consists of a group of pixels which can generate a number of feature points, such as corners and invariant points, [2, Chap. 10]. A target which can cause multiple detections is referred to as an *extended* or nonpoint object. The problem of tracking an extended object is more difficult than that of tracking a point target. However, being relevant in practical applications, it has recently attracted significant interest, see [3–6].

Among various approaches to extended target tracking, of particular interest are those based on the random set theory [7]. Using the Poisson model of extended object detections [8], Mahler formulated the PHD filter for extended target tracking [9]. The cardinalized PHD (CPHD) filter for extended targets has been subsequently proposed in Orguner et al. [10]. Both aforementioned formulations of the PHD and CPHD filters are applicable to *multiple* extended object filtering. A different approach

was taken in Swain and Clark [11]: the problem was formulated in the framework of doubly stochastic point process theory, with analytic solution provided as the first-order moment approximation. The filter in Swain and Clark [11] provides sequential estimate of individual feature points but is limited to a single extended object which exists all the time.

In this section, we propose another random set based approach to extended target tracking. We formulate the Bernoulli filter, as the exact Bayesian filter for joint detection and state estimation of an extended object in the presence of clutter. While the theoretical formulation is exact, the implementation of this filter is based on approximations and is carried out in the sequential Monte Carlo framework. The main characteristic of the proposed Bernoulli filter is that it estimates jointly the posterior probability of object existence and the posterior probability density function (PDF) of the extended object over the state space. As usual, the probability of existence plays a role in determining the presence or absence of the object, and hence solves the detection aspect of the problem. From the posterior PDF over the state space one can estimate the state vector, its covariance and higher order moments. The proposed Bernoulli filter does not estimate individual feature points and consequently can be computationally tractable, as it will be demonstrated by numerical examples.

7.1.1 Mathematical Models

Consider an extended object whose characteristics of interest at time t_k , such as position, heading, speed, shape, and orientation parameters, are contained in the state vector $\mathbf{x}_k \in \mathcal{X}$. The evolution of target state in time as usual is assumed to be a Markov process with transition density $\pi_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1})$. In order to model the dynamics of object presence or absence from the surveillance volume of interest, a binary random variable $\varepsilon_k \in \{0, 1\}$, referred to as the *target existence* is introduced. As before, the convention is that $\varepsilon_k = 1$ means that object exists at time k . The dynamics of ε_k is modeled by a two-state Markov chain with a TPM Π as in (5.12). The elements of the TPM are defined as $[\Pi]_{ij} = P\{\varepsilon_{k+1} = j - 1 | \varepsilon_k = i - 1\}$ for $i, j \in \{1, 2\}$. We adopt a TPM as follows:

$$\Pi = \begin{bmatrix} (1 - p_b) & p_b \\ (1 - p_s) & p_s \end{bmatrix} \quad (7.1)$$

where $p_b := P\{\varepsilon_{k+1} = 1 | \varepsilon_k = 0\}$ is the probability of target “birth” and $p_s := P\{\varepsilon_{k+1} = 1 | \varepsilon_k = 1\}$ the probability of target “survival”. These two probabilities together with the initial target existence probability $q_0 = P\{\varepsilon_0 = 1\}$ are modeling parameters, assumed to be known. Typically $q_0 = 0$, p_b is small and p_s is close to 1. If the object appears during the time interval $T_k = t_k - t_{k-1}$, the PDF $b_{k|k-1}(\mathbf{x})$ denotes its birth density. Thus, the moving object is effectively modeled by a Bernoulli RFS \mathbf{X}_k which is either empty (if $\varepsilon_k = 0$) or a singleton $\mathbf{X}_k = \{\mathbf{x}\}$ (if $\varepsilon_k = 1$).

The measurements produced by a detector at time k can be represented by a random finite set $\mathbf{Z}_k = \{\mathbf{z}_{k,1}, \mathbf{z}_{k,2}, \dots, \mathbf{z}_{k,m_k}\}$, where $\mathbf{z}_{k,j} \in \mathcal{Z}$, for $j = 1, \dots, m_k$. The key feature of this representation is that the order of measurements is irrelevant and that both cardinality $m_k = |\mathbf{Z}_k|$ and the position of individual elements of \mathbf{Z}_k in the measurement space $\mathcal{Z} \subseteq \mathbb{R}^{n_z}$, are random.

The measurement RFS \mathbf{Z}_k can be seen as a union of two independent RFSs:

$$\mathbf{Z}_k = \mathbf{K}_k \cup \mathbf{W}_k \quad (7.2)$$

where \mathbf{K}_k is the RFS of false detections (clutter) and \mathbf{W}_k is the RFS of measurements due to the object of interest for tracking. As usual, the RFS \mathbf{K}_k is modeled as a Poisson RFS, whose FISST PDF is given by (see Sect. 2.4.1):

$$\kappa(\mathbf{K}_k) = e^{-\lambda} \prod_{\mathbf{z} \in \mathbf{K}_k} \lambda c(\mathbf{z}). \quad (7.3)$$

The parameter λ is the mean number of false detections at time k , while $c(\mathbf{z})$ is the spatial distribution of false detections over the measurement space \mathcal{Z} .

Modeling \mathbf{W}_k is discussed next. Suppose the moving object contains L_k scattering points (or scatterers) at time k . Each scatterer can be detected with probability p_d . Then, one can be tempted to adopt the model $\mathbf{W}_k = \cup_{j=1}^{L_k} \mathbf{W}_{k,j}$, where $\mathbf{W}_{k,j}$, $j = 1, \dots, L_k$ are mutually independent Bernoulli RFSs, each corresponding to one scattering point. This type of RFS is known as the multi-Bernoulli RFS (see Sect. 2.4.1). If we were to continue the derivation in this framework, we would practically develop a method that tracks L_k individual scattering points as in Swain and Clark [11]. This approach, however, is exceptionally complex and the resulting algorithm computationally difficult to implement in a general nonlinear/nonGaussian case.

We take a different approach and model the RFS $\mathbf{W}_k = \{\mathbf{w}_{k,1}, \dots, \mathbf{w}_{k,i}\}$ as a binomial RFS (see Sect. 2.4.1). The key feature of this model is to treat all object originated detections as having a *single* source. In this way, instead of tracking individual scattering points, we will track the object centroid and its shape/size. The cardinality distribution of the binomial RFS \mathbf{W}_k is the binomial distribution, while the elements of \mathbf{W}_k are IID with the spatial distribution $g_k(\mathbf{w}|\mathbf{x})$. This distribution represents the likelihood of scattering point measurement \mathbf{w} given the state of the object \mathbf{x} . The FISST PDF of RFS \mathbf{W}_k is then given by:

$$\eta(\mathbf{W}_k|\{\mathbf{x}\}) = \frac{L_k!}{(L_k - |\mathbf{W}_k|)!} p_d^{|\mathbf{W}_k|} (1 - p_d)^{L_k - |\mathbf{W}_k|} \prod_{\mathbf{w} \in \mathbf{W}_k} g_k(\mathbf{w}|\mathbf{x}) \quad (7.4)$$

for cardinalities $|\mathbf{W}_k| = 0, 1, 2, \dots, L_k$, and zero otherwise. The probability of detection of a scattering point, p_d , can be made dependent on the object state \mathbf{x} , but this is omitted for simplicity.

The likelihood function of a measurement set \mathbf{Z} , expressed as the union (7.2), can be derived using the fundamental convolution formula [7, Eq.(11.253)]. If the object is not present, the likelihood is simply:

$$\varphi(\mathbf{Z}|\emptyset) = \kappa(\mathbf{Z}). \quad (7.5)$$

where $\kappa(\mathbf{Z})$ was defined in (7.3). If the object is present, the likelihood is given by:

$$\varphi(\mathbf{Z}|\{\mathbf{x}\}) = \kappa(\mathbf{Z}) \left\{ (1-p_d)^L + \sum_{\Omega \in \mathcal{P}_{1:L}(\mathbf{Z})} \frac{L!}{(L-|\Omega|)!} p_d^{|\Omega|} (1-p_d)^{L-|\Omega|} \prod_{\mathbf{z} \in \Omega} \frac{g(\mathbf{z}|\mathbf{x})}{\lambda c(\mathbf{z})} \right\} \quad (7.6)$$

where L is the number of scatters of an object in state \mathbf{x} , $\mathcal{P}_{1:L}(\mathbf{Z})$ is the set of all subsets of \mathbf{Z} with cardinalities equal to $1, 2, \dots, L$.

Suppose for example that $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3\}$ and $L = 5$. Then $\mathcal{P}_{1:L}(\mathbf{Z}) = \{\{\mathbf{z}_1\}, \{\mathbf{z}_2\}, \{\mathbf{z}_3\}, \{\mathbf{z}_1, \mathbf{z}_2\}, \{\mathbf{z}_1, \mathbf{z}_3\}, \{\mathbf{z}_2, \mathbf{z}_3\}, \{\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3\}\}$ and according to (7.6) we have:

$$\begin{aligned} \varphi_k(\mathbf{Z}|\{\mathbf{x}\}) = & \kappa(\mathbf{Z})(1-p_d)^L \left\{ 1 + \frac{p_d g_k(\mathbf{z}_1|\mathbf{x})}{(1-p_d)\lambda c(\mathbf{z}_1)} + \frac{p_d g_k(\mathbf{z}_2|\mathbf{x})}{(1-p_d)\lambda c(\mathbf{z}_2)} + \right. \\ & \frac{p_d g_k(\mathbf{z}_3|\mathbf{x})}{(1-p_d)\lambda c(\mathbf{z}_3)} + \frac{p_d^2 g_k(\mathbf{z}_1|\mathbf{x})g_k(\mathbf{z}_2|\mathbf{x})}{(1-p_d)^2 \lambda^2 c(\mathbf{z}_1)c(\mathbf{z}_2)} + \frac{p_d^2 g_k(\mathbf{z}_1|\mathbf{x})g_k(\mathbf{z}_3|\mathbf{x})}{(1-p_d)^2 \lambda^2 c(\mathbf{z}_1)c(\mathbf{z}_3)} + \\ & \left. \frac{p_d^2 g_k(\mathbf{z}_2|\mathbf{x})g_k(\mathbf{z}_3|\mathbf{x})}{(1-p_d)^2 \lambda^2 c(\mathbf{z}_2)c(\mathbf{z}_3)} + \frac{p_d^3 g_k(\mathbf{z}_1|\mathbf{x})g_k(\mathbf{z}_2|\mathbf{x})g_k(\mathbf{z}_3|\mathbf{x})}{(1-p_d)^3 \lambda^3 c(\mathbf{z}_1)c(\mathbf{z}_2)c(\mathbf{z}_3)} \right\}. \end{aligned} \quad (7.7)$$

Similarly, it can be verified that for $L = 1$, model (7.6) simplifies to the likelihood function of the standard (point-target) Bernoulli filter, given by (2.46), i.e.,

$$\varphi(\mathbf{Z}|\{\mathbf{x}\}) = \kappa(\mathbf{Z}) \left[1 - p_d(\mathbf{x}) + p_d(\mathbf{x}) \sum_{\mathbf{z} \in \mathbf{Z}} \frac{g(\mathbf{z}|\mathbf{x})}{\lambda c(\mathbf{z})} \right]. \quad (7.8)$$

Note that the number of scattering points L_k in (7.6) is unknown. The implication of unknown L_k will be discussed in Sect. 7.1.3.

7.1.2 Equations of the Bernoulli Filter for an Extended Target

Recall from Sect. 2.5.1 that the Bernoulli filter is completely specified by the pair $(q_{k|k}, s_{k|k}(\mathbf{x}))$, where $q_{k|k}$ is the posterior probability of object existence and $s_{k|k}(\mathbf{x})$ is the spatial posterior PDF.

We adopt the state vector of extended object to consist of: (1) object centroid position \mathbf{p} ; (2) the centroid velocity vector \mathbf{v} ; and (3) the shape/size parametric representation, in the form of a vector Ω . For example, let us consider extended

object in the two-dimensional (2D) Cartesian coordinate system. Then $\mathbf{p} = [x \ y]^\top$ and $\mathbf{v} = [\dot{x} \ \dot{y}]^\top$. Furthermore, if we represent the shape of the extended object by an ellipse, Ω will consist of three parameters which uniquely define its major axis, its minor axis and its orientation in 2D.

Since the state vector does not include individual scattering points, the prediction equations are identical to those of the standard Bernoulli filter, see Sect. 2.5.1. The predicted FISST PDF of the Bernoulli filter then can be expressed as:

$$f_{k|k-1}(\mathbf{X}|\mathbf{Z}_{1:k-1}) = \begin{cases} 1 - q_{k|k-1} & \text{if } \mathbf{X} = \emptyset \\ q_{k|k-1} \cdot s_{k|k-1}(\mathbf{x}) & \text{if } \mathbf{X} = \{\mathbf{x}\} \\ 0 & \text{if } |\mathbf{X}| > 1 \end{cases}. \quad (7.9)$$

where the predicted existence $q_{k|k-1}$ and spatial PDF $s_{k|k-1}(\mathbf{x})$ are obtained from (2.44) and (2.45), respectively.

The update step of the Bernoulli filter will be derived next following [7, Sect. G.24]. Recall from (2.36) that the update equation of the multi-object Bayes filter is given by:

$$f_{k|k}(\mathbf{X}|\mathbf{Z}_{1:k}) = \frac{\varphi_k(\mathbf{Z}_k|\mathbf{X}) \cdot f_{k|k-1}(\mathbf{X}|\mathbf{Z}_{1:k-1})}{f(\mathbf{Z}_k|\mathbf{Z}_{1:k-1})} \quad (7.10)$$

where

$$f(\mathbf{Z}_k|\mathbf{Z}_{1:k-1}) = \int \varphi_k(\mathbf{Z}_k|\mathbf{X}) \cdot f_{k|k-1}(\mathbf{X}|\mathbf{Z}_{1:k-1}) \delta \mathbf{X}. \quad (7.11)$$

First we evaluate (7.11) in order to find the expression for $f(\mathbf{Z}_k|\mathbf{Z}_{1:k-1})$. This involves the set integral and leads to the expression:

$$f(\mathbf{Z}_k|\mathbf{Z}_{1:k-1}) = \kappa(\mathbf{Z}_k) \left\{ 1 - q_{k|k-1} + q_{k|k-1} (1 - p_d)^{L_k} \right. \\ \left. + \sum_{\Omega \in \mathcal{P}_{1:L_k}(\mathbf{Z}_k)} \frac{L_k! p_d^{|\Omega|}}{(L_k - |\Omega|)! (1 - p_d)^{|\Omega|-L_k}} \frac{\int \prod_{\mathbf{z} \in \Omega} g_k(\mathbf{z}|\mathbf{x}) s_{k|k-1}(\mathbf{x}) d\mathbf{x}}{\prod_{\mathbf{z} \in \Omega} \lambda c(\mathbf{z})} \right\} \quad (7.12)$$

Next we solve (7.10) for the case $\mathbf{X} = \emptyset$. The posterior FISST PDF $f_{k|k}(\mathbf{X}|\mathbf{Z}_{1:k})$ is also Bernoulli specified by the pair $(q_{k|k}, s_{k|k}(\mathbf{x}))$. For $\mathbf{X} = \emptyset$ we have then: $f_{k|k}(\emptyset|\mathbf{Z}_{1:k}) = 1 - q_{k|k}$, and $f_{k|k-1}(\emptyset|\mathbf{Z}_{1:k-1}) = 1 - q_{k|k-1}$. From (7.5), we have $\varphi_k(\mathbf{Z}_k|\emptyset) = \kappa(\mathbf{Z}_k)$. Now (7.10) can be written as:

$$1 - q_{k|k} = \frac{\kappa(\mathbf{Z}_k)(1 - q_{k|k-1})}{f(\mathbf{Z}_k|\mathbf{Z}_{1:k-1})}. \quad (7.13)$$

Upon the substitution of (7.12) into (7.13), one can express the update equation for the probability of existence as follows:

$$q_{k|k} = \frac{1 - \Delta_k}{1 - \Delta_k q_{k|k-1}} \quad (7.14)$$

where

$$\Delta_k = 1 - (1 - p_d)^{L_k} - \sum_{\Omega \in \mathcal{P}_{1:L_k}(\mathbf{Z}_k)} \frac{L_k! p_d^{|\Omega|}}{(L_k - |\Omega|)! (1 - p_d)^{|\Omega|-L_k}} \frac{\int \prod_{\mathbf{z} \in \Omega} g(\mathbf{z}|\mathbf{x}) s_{k|k-1}(\mathbf{x}) d\mathbf{x}}{\prod_{\mathbf{z} \in \Omega} \lambda c(\mathbf{z})}. \quad (7.15)$$

Similar procedure for solving (7.10) in the case $\mathbf{X} = \{\mathbf{x}\}$ leads to the update equation for the spatial posterior PDF:

$$s_{k|k}(\mathbf{x}) = \left[(1 - p_d)^{L_k} + \sum_{\Omega \in \mathcal{P}_{1:L_k}(\mathbf{Z}_k)} \frac{L_k! p_d^{|\Omega|}}{(L_k - |\Omega|)! (1 - p_d)^{|\Omega|-L_k}} \prod_{\mathbf{z} \in \Omega} \frac{g_k(\mathbf{z}|\mathbf{x})}{\lambda c(\mathbf{z})} \right] \frac{s_{k|k-1}(\mathbf{x})}{1 - \Delta_k}. \quad (7.16)$$

It can be easily verified that by setting $L_k = 1$, Eqs. (7.14) and (7.16) simplify to the update equations of the standard Bernoulli filter, that is (2.47) and (2.48), respectively.

7.1.3 Numerical Implementation

The numerical implementation of the proposed Bernoulli filter for an extended object (BPF-X), as it is usually the case with Bayes filters, is based on approximations. First, the spatial PDF $s_{k|k}(\mathbf{x})$ is approximated by a particle system with normalized weights, as usual denoted by $\{w_{k|k}^{(i)}, \mathbf{x}_{k|k}^{(i)}\}_{i=1}^N$. The prediction and update of both $q_{k|k}$ and $s_{k|k}(\mathbf{x})$ are then implemented as in the standard (point target) Bernoulli particle filter, described in detail in Sect. 4.1.1. There are, however, some important differences in relation to Algorithm 3.

1. Line 9 in Algorithm 3 has to be modified to agree with (7.15);
2. The unnormalized weights of particles are updated according to (7.16) as:

$$\begin{aligned} w_{k|k}^{(i)} &\propto w_{k|k-1}^{(i)} \cdot \left[(1 - p_d)^{L_k} + \sum_{\Omega \in \mathcal{P}_{1:L_k}(\mathbf{Z}_k)} \frac{L_k! p_d^{|\Omega|}}{(L_k - |\Omega|)! (1 - p_d)^{|\Omega|-L_k}} \prod_{\mathbf{z} \in \Omega} \frac{g_k(\mathbf{z}|\mathbf{x})}{\lambda c(\mathbf{z})} \right] \\ &\propto w_{k|k-1}^{(i)} \cdot \varphi(\mathbf{Z}|\{\mathbf{x}_{k|k-1}^{(i)}\}) \end{aligned} \quad (7.17)$$

which replaces Eq. (4.8). Line 11 in Algorithm 3 is accordingly changed.

Recall that L_k , which is required in (7.15) and (7.16), is unknown and not included in the state vector \mathbf{x}_k . Instead, L_k can be estimated as follows:

$$\widehat{L}_k = \left\lfloor \frac{|\mathbf{Z}_k| - \lambda}{p_d} \right\rfloor \quad (7.18)$$

where $\lfloor \cdot \rfloor$ denotes the *nearest integer* operation.

On the Influence of Error in L_k

In general, the estimate \widehat{L}_k will differ to some degree from the true value of L_k . Let us investigate the effect this error in L_k on the term $\varphi_k(\mathbf{Z}_k | \{\mathbf{x}\})$ which features in (7.17).

Consider the following simple example. Suppose the state of an extended object includes only its centroid position in one dimension $x \in \mathbb{R}$, and its extent parameter $\sigma \in \mathbb{R}^+$, such that the object occupies an interval on \mathbb{R} specified by $[x - 3\sigma, x + 3\sigma]$. Therefore, the state vector contains only two components, i.e., $\mathbf{x} = [x, \sigma]^T$. The scattering points (features) and measurements are also the points on the real axis, i.e., the measurement space is $\mathcal{Z} \subset \mathbb{R}$. The scattering point likelihood of an object originated measurement $z \in \mathcal{Z}_k$ is specified by the Gaussian distribution, that is $g_k(z | \mathbf{x}) = \mathcal{N}(z; x, \sigma^2)$. Suppose the actual measurement set is $\mathbf{Z}_k = \{1, 4.5, 4.7, 15\}$, while the measurement model parameters are $c(z) = 1/20$ for $z \in [0, 20]$ and zero otherwise, $\lambda = 1$ and $p_d = 0.6$. Let us assume that there are three predicted particles $\mathbf{x}_{k|k-1}^{(i)}$, and we want to compute the likelihood of each of them using (7.17). The three particles are: $\mathbf{x}_{k|k-1}^{(1)} = [5, 0.3]^T$, $\mathbf{x}_{k|k-1}^{(2)} = [5, 2]^T$ and $\mathbf{x}_{k|k-1}^{(3)} = [10, 3]^T$. Figure 7.1a shows the scattering point likelihoods $g_k(z | \mathbf{x}_{k|k-1}^{(i)})$ for the three particles $i = 1, 2, 3$. The actual measurements are indicated by crosses on the x -axis.

Next we compute $\varphi_k(\mathbf{Z}_k | \{\mathbf{x}_{k|k-1}^{(i)}\})$ for $i = 1, 2, 3$ using $L_k = 1, 2, \dots, 9$. Figure 7.1b shows the result. Observe from Fig. 7.1b that the choice of L_k , although important, is not critical, because the relative ranking of the three likelihoods,

$$\varphi_k(\mathbf{Z}_k | \{\mathbf{x}_{k|k-1}^{(1)}\}) > \varphi_k(\mathbf{Z}_k | \{\mathbf{x}_{k|k-1}^{(2)}\}) > \varphi_k(\mathbf{Z}_k | \{\mathbf{x}_{k|k-1}^{(3)}\}),$$

appears to be invariant to the changes in L_k . Moreover, for any value of $L_k = 1, 2, 3, 4, 5$ or 6 , the unnormalized weights in (7.17) seem to end up having a similar ratio. Since the weights are normalized, only their ratio is important for the survival of particles in the resampling step. In summary, we conclude that small errors in the estimation of L_k should not cause a problem for the BPF-X.

Since the cardinality of $\mathcal{P}_{1:L}(\mathbf{Z})$ grows exponentially with L , so does the computational complexity of the BPF-X. Having established empirically that small errors in the estimation of L_k do not appear to be critical, we introduce another approximation by selecting L_k as:

$$\widehat{\widehat{L}}_k = \min\{\widehat{L}, L^*\}.$$

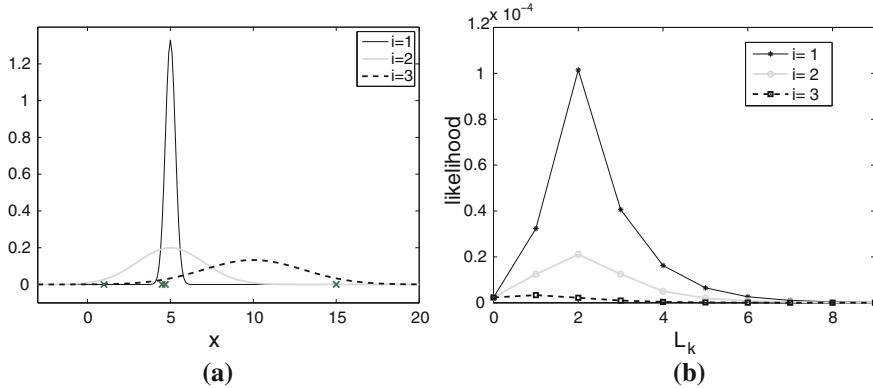


Fig. 7.1 **a** Scattering point likelihoods $g_k(z|x_{k|k-1}^{(i)})$ of particles $\mathbf{x}_{k|k-1}^{(1)} = [5, 0.3]^\top$, $\mathbf{x}_{k|k-1}^{(2)} = [5, 2]^\top$ and $\mathbf{x}_{k|k-1}^{(3)} = [10, 3]^\top$; measurements in $\mathbf{Z}_k = \{1, 4.5, 4.7, 15\}$ are indicated by crosses on the x axis. **b** Measurement set likelihoods $\varphi_k(\mathbf{Z}_k|\{\mathbf{x}_{k|k-1}^{(i)}\})$ for $i = 1, 2, 3$ as a function of L_k

Here $L^* > 0$ is the upper limit of L , adopted as a trade-off between the computational speed and accuracy.

Finally, we introduce an approximation which involves *gating* of measurements, that is the selection of a subset $\mathbf{Z}_k^* \subseteq \mathbf{Z}_k$ which has a reasonable chance of being due to the target. A measurement $\mathbf{z} \in \mathbf{Z}_k$ will be included in \mathbf{Z}_k^* if $\sum_{i=1}^{N+B_{k-1}} g_k(\mathbf{z}|\mathbf{x}_{k|k-1}^{(i)}) > \eta$, where $\eta \geq 0$ is a user specified threshold. The update step of the Bernoulli filter is in practice carried out using \mathbf{Z}_k^* , rather than \mathbf{Z}_k , in order to speed up the computation.

7.1.4 Simulation Results

Consider a two-dimensional surveillance area in Cartesian coordinates, specified by the lower left corner $(0, 0)$ and upper right corner $(500, 300)$. The state vector is specified by the object centroid position $\mathbf{p} = [x, y]^\top$, its velocity vector $\mathbf{v} = [\dot{x}, \dot{y}]^\top$ and the ellipsoidal shape defined by vector $\boldsymbol{\Omega} = [a, b, c]^\top$. Similarly to [5], we adopt that $\boldsymbol{\Omega}$ defines the elements of the covariance matrix which determines the spread of a Gaussian PDF with mean \mathbf{p} . Thus a and b are the diagonal elements of this covariance matrix, while c determines the cross-covariance term (effectively the orientation of the ellipse). The state vector thus consists of seven components: $\mathbf{x} = [\mathbf{p}^\top \ \mathbf{v}^\top \ \boldsymbol{\Omega}^\top]^\top$. Each particle effectively represents an ellipse with velocity vector.

The target in state $\mathbf{p} = [180, 0]^\top$, $\mathbf{v} = [4, 4]^\top$ and $\boldsymbol{\Omega} = [60, 30, -5]^\top$ enters the surveillance area at $k = 5$ and leaves the area at $k = 72$. Its dynamics is described by the transitional density $\pi_{k|k-1}(\mathbf{x}|\mathbf{x}') = \mathcal{N}(\mathbf{x}; \mathbf{F}\mathbf{x}', \mathbf{Q})$, where \mathbf{F} and \mathbf{Q} are selected, so that the centroid of the target travels with the nearly constant velocity [12]. The time evolution of the shape parameter vector is modeled by a random walk [12].

Figure 7.2c shows a typical trajectory of the target centroid. The “true” number of scattering points L_k is selected to be proportional to the size of the object and is shown in Fig. 7.2b by a solid line. The probability of scattering point detection is set to $p_d = 0.6$; the false detections are uniformly distributed over the surveillance area with average count $\lambda = 5$. The likelihood function of a scattering point is adopted to be Gaussian $g_k(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \mathbf{p}, \mathbf{S})$, with covariance $\mathbf{S} = \begin{bmatrix} a & c \\ c & b \end{bmatrix}$.

The location of target-birth particles at k is randomly generated from a Gaussian distribution with mean equal to the mean of the measurements in subset $\Omega \in \mathcal{P}(\mathbf{Z}_{k-1})$. The velocities of target-birth particles are randomly sampled from a uniform distribution over the interval $[-v_{\max}, +v_{\max}]$, with $v_{\max} = 1$ pixel/s. The shape vector of target-birth particles is also Gaussian distributed with the mean corresponding to the elements of the covariance matrix of measurements in $\Omega \in \mathcal{P}(\mathbf{Z}_{k-1})$. If Ω is a singleton, then the shape vector represents a circle with a small random diameter.

The results of a single run of the proposed Bernoulli particle filter for extended target are shown in Fig. 7.2. The BPF-X was implemented using $N_b = 100$ and $N = 5,000$ particles, with the gating threshold $\eta = 100$. The probability $q_{k|k}$ in Fig. 7.2a determines with a small delay the presence of the object at about $k = 10$. It reliably stays at a very high value, almost equal to 1, until, again with a small delay, determines the that the objects have disappeared at about $k = 75$. The true and estimated trajectories of the target centroid are shown in Fig. 7.2c. The estimates of the extended target are shown in Fig. 7.2d at $k = 20, 31, 42$ and 53 . We notice a remarkable accuracy in estimating both the centroid and the size/orientation of the object. Finally, Fig. 7.2b displays the estimate \hat{L}_k using $L^* = 7$ during the observation interval. As we have anticipated in Sect. 7.1.3, the error in the estimation of L_k does not appear to have a significant effect on the performance of the BPF-X.

Next the error performance of the BPF-X is established using the OSPA metric with parameter $p = 2$, introduced in Sect. 2.4.3. Suppose at time k the true object state is \mathbf{X}_k , while the estimated state (the output from the BPF-X) is $\hat{\mathbf{X}}_k$. Since the sets \mathbf{X}_k and $\hat{\mathbf{X}}_k$ can be either empty or singletons $\mathbf{X}_k = \{\mathbf{x}_k\}$ and $\hat{\mathbf{X}}_k = \{\hat{\mathbf{x}}_k\}$, the definition of OSPA error in Sect. 2.4.3 simplifies to:

$$d_{p=2,c}(\mathbf{X}_k, \hat{\mathbf{X}}_k) = \begin{cases} \min\{c, d_b(\mathbf{x}_k, \hat{\mathbf{x}}_k)\}, & \text{if } \mathbf{X}_k = \{\mathbf{x}_k\} \text{ and } \hat{\mathbf{X}}_k = \{\hat{\mathbf{x}}_k\} \\ c, & \text{if } |\mathbf{X}| \neq |\hat{\mathbf{X}}_k| \\ 0, & \text{if } \mathbf{X}_k = \hat{\mathbf{X}}_k = \emptyset, \end{cases}$$

where c is the cut-off parameter and $d_b(\mathbf{x}_k, \hat{\mathbf{x}}_k)$ is the base distance defined over \mathcal{X} . We will consider two cases of OSPA error. In the first case, the base distance is defined as the Euclidian distance between the location of the true object centroid \mathbf{p}_k and its estimate $\hat{\mathbf{p}}_k$:

$$d'_b(\mathbf{x}_k, \hat{\mathbf{x}}_k) = \| \mathbf{p}_k - \hat{\mathbf{p}}_k \| \quad (7.19)$$

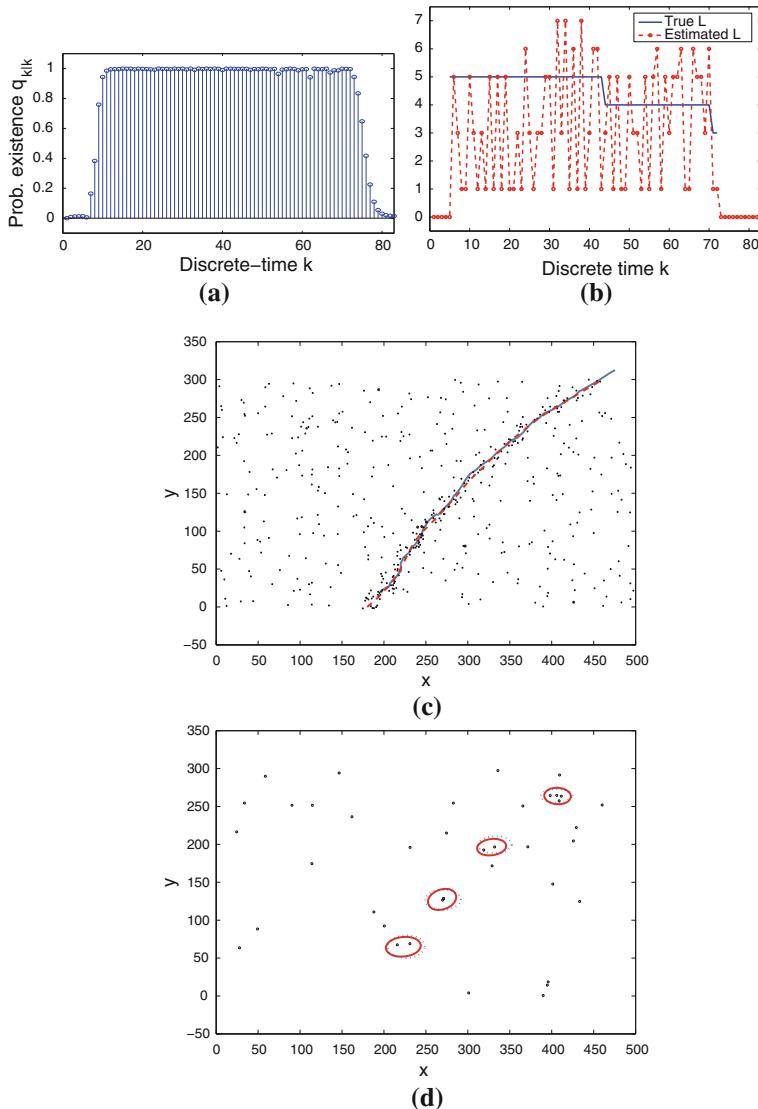


Fig. 7.2 A single run of the BPF-X. **a** Posterior probability of existence $q_{k|k}$. **b** True L_k and its estimate \hat{L}_k using $L^* = 7$. **c** All detections over the observation period, together with the true (dashed line) and estimated (solid line) centroid trajectories. **d** Overlayed snapshots at $k = 20, 31, 42, 53$: detections (dots), true extended object (dotted line ellipse) and estimated object (solid line ellipse)

This is referred to as the centroid localization OSPA error, and its cut-off parameter is adopted as $c' = 10$. In the second case, the base distance is defined as the Euclidian distance between the shape/size parameter vectors Ω_k (true) and $\hat{\Omega}_k$ (estimate):

$$d_b''(\mathbf{x}_k, \hat{\mathbf{x}}_k) = \| \boldsymbol{\Omega}_k - \hat{\boldsymbol{\Omega}}_k \| \quad (7.20)$$

This is referred to as the shape/size OSPA error and its cut-off parameter is adopted as $c'' = 60$.

The two OSPA errors were averaged over 100 independent Monte Carlo runs and shown in Fig. 7.3. The true object trajectory and shape/size was created with zero process noise (i.e., identical in each run), with the target present from $k = 5$ to 78. The results are computed for different values of L^* , that is $L^* = 1, 3, 5, 7$.

We can observe some general trends in the behavior of both OSPA errors in Fig. 7.3. Initially, during the absence of the target, the OSPA errors are zero. Then at $k = 5$, when the target appears, both the localization and size/shape mean OSPA errors jump to their respective cut-off values, due to the delay in track formation. Subsequently, both OSPA errors reduce as the base distance component becomes the dominant source of error. The errors during this interval are reduced due to the BPF-X convergence. Then at time $k = 78$, the errors jump back to their respective cut-off values, as a result of a delay in track termination. Eventually, from $k = 88$ onwards, both OSPA errors are zero.

Regarding the centroid localization mean OSPA error in Fig. 7.3a, one can observe that the choice of $L^* > 0$ does not seem to make a difference. This is a remarkable result, because the BPF-X at $L^* = 1$ reduces to the standard point-target BPF (Sect. 4.1.1), characterized by a linear computational complexity with respect to the number of measurements. Based on Fig. 7.3a we can conclude that if one is interested only in the extended object centroid estimate, then it is adequate to use the standard point-target BPF (i.e., $L^* = 1$).

The shape/size mean OSPA error in Fig. 7.3b, however, reveals that the choice of $L^* = 1$ results in a poor estimation of the object shape/size parameter vector $\boldsymbol{\Omega}$. This agrees with our expectation: recall that the true value of L_k in this simulation setup was initially 5 and then dropped to 4 (see Fig. 7.2b). According to Fig. 7.3b, the BPF-X using $L^* = 3, 5, 7$ produces similarly accurate estimates of the size/shape parameter vector in this example.

7.1.5 Application to a Surveillance Video

The proposed BPF-X has been applied to a video sequence taken by a surveillance camera. The objective is to detect and track a moving car in this video. The car enters the scene from the right and during the observation period changes its size and aspect ratio. The size of each image is 768×576 pixels. Eight frames from the sequence are shown in Fig. 7.5. The object of interest, the (red) car, begins entering the scene at frame $k = 12$ and leaves the scene at frame $k = 225$. The dataset (PETS 2000) was downloaded from the website [13]. Detections are created using an implementation of the corner detector algorithm of Shi and Tomasi [14] available in OpenCV [2, Chap. 10]. Detections, created for each frame of the sequence, are filtered using

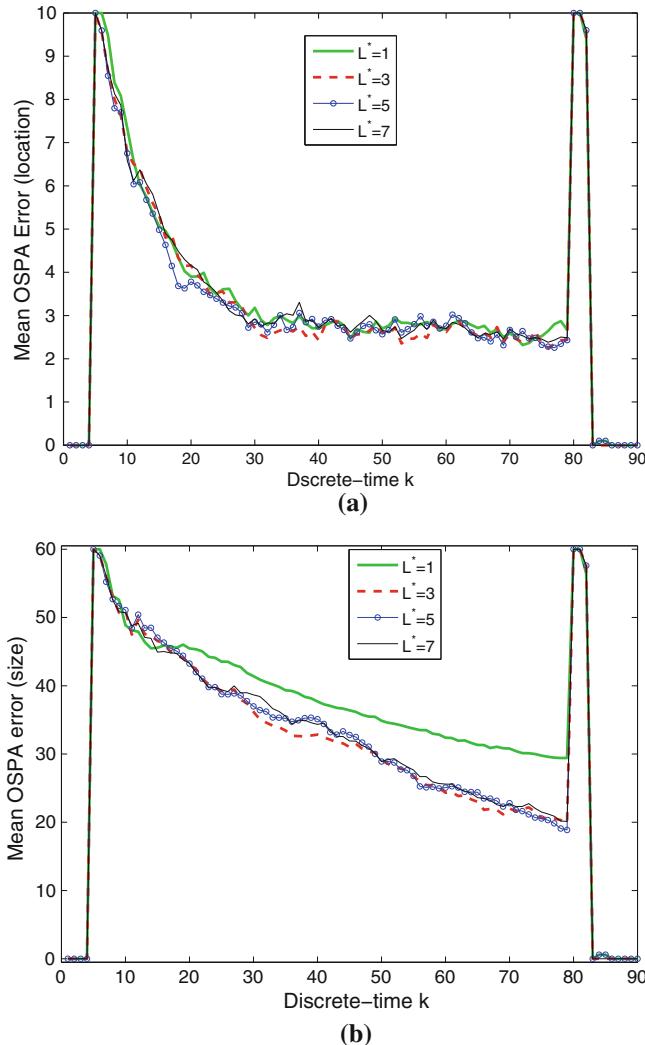


Fig. 7.3 Mean OSPA error using $L^* = 1, 3, 5, 7$. **a** Object centroid localization. **b** Object size/shape

a clutter map created from 30 frames of background images. In this way, detections which arise in areas deemed to be part of the background, are likely to be removed.

The parameters used in the proposed BPF-X for detection/tracking an extended object were: $\lambda = 0.5$, $p_D = 0.65$, $L^* = 5$, $N_b = 250$, $N = 5,000$, $\eta = 50$. The number of gated detections varied from frame to frame, with the maximum value of $|\mathbf{Z}_k^*| = 11$ in frame 38. The spatial density of clutter $c(\mathbf{z})$ is assumed uniform over

the extent of each image. While the density of false detections was very low and therefore relatively easy to handle, the main challenges with this dataset were:

- The variation of the size of the object over time (especially as the object grows as it enters the scene).
- The variation of the object speed (starts at approximately 150 pixels/s, to drop toward the end to only 5 pixels/s). In order to deal with this variability, the algorithm was applied with a relatively high level of process noise.

The results obtained using the proposed filter are shown in Figs. 7.4 and 7.5. The estimated probability of object existence $q_{k|k}$, shown in Fig. 7.4a, demonstrates reliable detection of both appearance and disappearance of the object of interest.

Figure 7.4b displays the estimated trajectory of the centroid of the object of interest (solid line), as well as all detections accumulated over the observation period of 227 frames. Due to the high level of process noise, the trajectory is not smooth, but it follows fairly accurately the cluster of detections. Estimation results can be better seen in Fig. 7.5. This figure displays eight image frames of the video sequence, with overlayed ellipses indicating the estimates. As the object enters the scene (frames 14 and 24), both the object and its estimate grow rapidly. In frame 24 the estimate appears to be somewhat biased, although this bias is quickly eliminated as shown in frame 28. When the object moves far away, both its speed and size slowly reduce, which is correctly reflected by the estimates.

In summary, the numerical analysis demonstrates that the BPF-X reliably estimates the existence probability as well as the object centroid, velocity, and its size/shape parameter. It is left for future work to formulate a multi-target tracker for extended objects based on the BPF-X. This can be carried out following the ideas presented in Sect. 6.2.1.

7.2 Calibration of Tracking Systems

Multi-target tracking systems [1] are based on mathematical models which typically include many parameters. The principal models are the target dynamic model (for target birth and motion) and the sensor measurement model. The parameters typically used in these two models are: the process noise level, probability of birth, the false alarm rate, the probability of detection, sensor biases, and various factors such as the propagation losses, receiver gains, etc. Calibration of tracking systems, through estimation of its model parameters, is an important prerequisite for their operational deployment. Yet, apart from sensor registration, calibration has received little attention by the research community and is mainly carried out in an ad hoc manner.

This section is devoted to the estimation of static parameters which may feature in the probabilistic models that describe target dynamics (the transitional density) and sensor measurements (the likelihood function) using detections from noncooperative targets. Let a random vector $\theta \in \Theta$ represents the (static) vector of parameters of

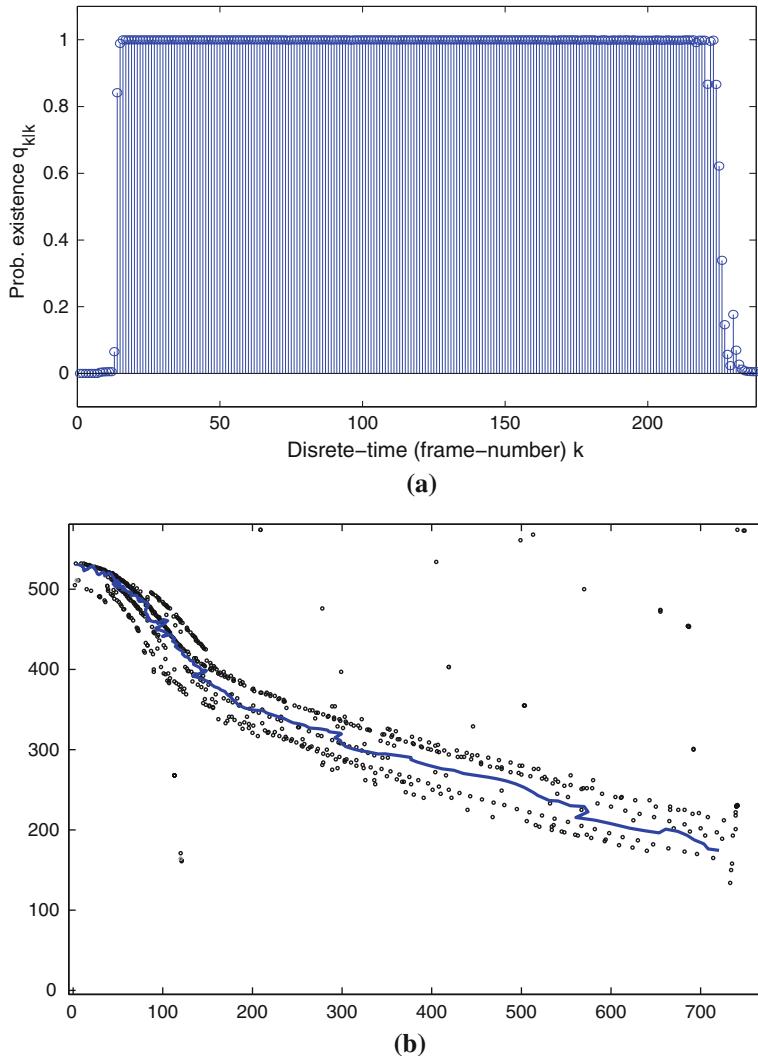


Fig. 7.4 Video surveillance sequence. **a** Estimated probability of existence over time. **b** Estimated trajectory (blue solid line) and all detections (circles)

interest for estimation/calibration. This section proposes a batch method for Bayesian estimation of the parameter vector θ . Since θ is unknown we ascribe to it a prior density and carry out Bayesian inference on the joint space of the multi-target dynamic state and the parameter vector θ . The multi-target state, modeled by a random finite set [7], determines the number of targets and their location in the target state space at a particular time. The multi-target state, conditioned on θ , is estimated using the PHD filter (Sect. 2.5.2). The data used for estimation of θ are the sets of measurements

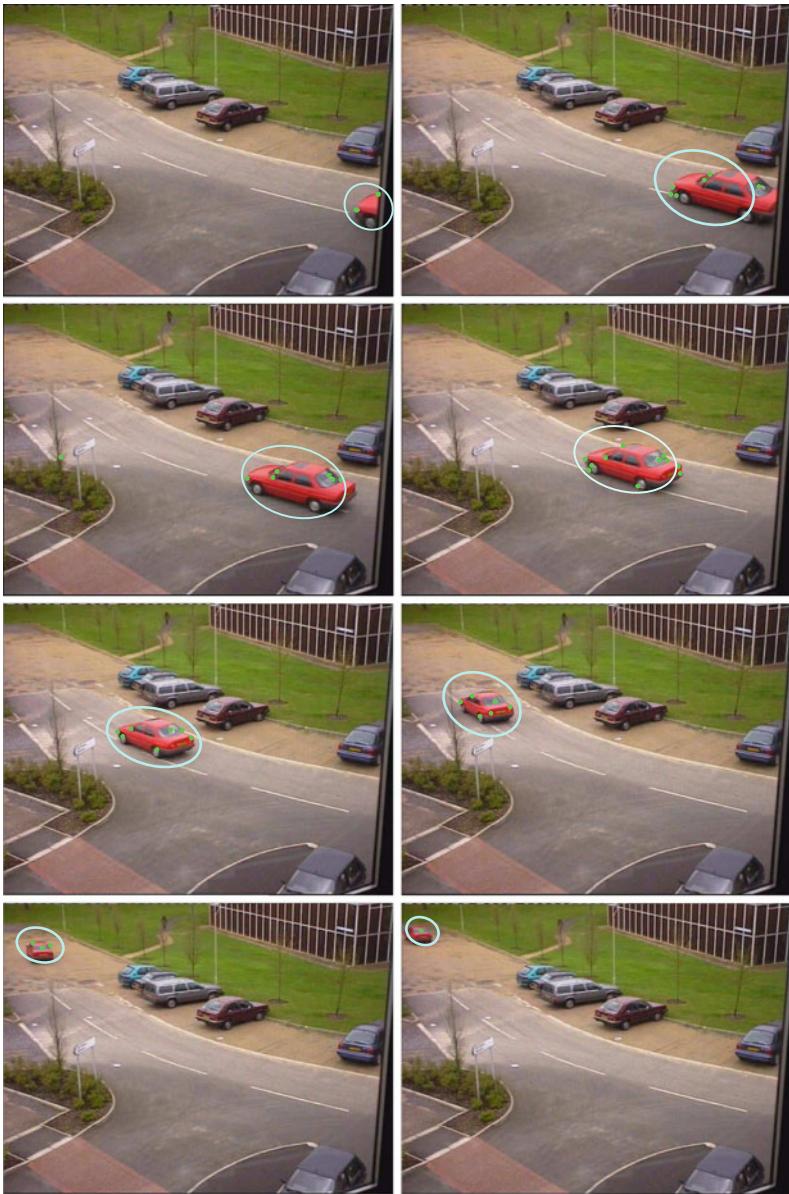


Fig. 7.5 The results of detection and tracking of a red car using BPF-X (PETS 2000 dataset). The frames shown: row 1—frames 14 and 24; row 2—frames 28 and 38; row 3—frames 55 and 80; row 4—frames 155 and 200. Each ellipse (*solid line*) represents the estimate of the extended object

collected by the sensor(s) in the presence of noncooperative targets, that is, targets that happen to be in the surveillance volume during the data collection interval. The number of such targets and their states (positions/velocities), at any point of time, are unknown.

The approach is motivated by, and strongly related to, a new class of PHD filters based on doubly stochastic point process models. Using the single-cluster model [11], which is a special case of a multi-group multi-object filter, Bayesian filters for tracking groups of targets [15], and extended objects [16] have been developed following the earlier proposed methods by Mahler [17, 18]. The preliminary results on tracking system calibration were reported in Ristic and Clark [19] in the context of doubly stochastic point process models and a dynamic calibration vector θ . The same models have also been developed for the problem of simultaneous localization and mapping (SLAM) for robotics applications [20, 21].

The proposed Bayesian estimator of the calibration parameter vector θ does not have a closed-form analytic solution and is implemented using importance sampling [22] in combination with the PHD particle filter. The PHD particle filter serves to evaluate the likelihood of the measurement dataset. Monte Carlo methods for parameter estimation in the context of single-target filtering have been considered in the past [23]; see also for an overview [24, Sect. IV] and references therein. One such a method, known as the particle Markov chain Monte Carlo (MCMC) [25, 26] has particularly become popular among the practitioners. The method described in this section can be seen as an extension of the parameter estimation problem from *single-object* nonlinear nonGaussian systems to *multiple-object* nonlinear nonGaussian systems in the presence of imperfect detection (false and missed detections).

After describing the calibration algorithm in the general framework of parameter estimation for a tracking system, we will focus on a particular problem of multi-sensor bias estimation. Whereas most standard algorithms for multi-sensor bias estimation require access to track-associated observations from the sensors [27, 28], (some even require the multi-sensor observations to be synchronous [29]), the method we present here is free of such restrictions.

7.2.1 Background and Problem Formulation

Suppose at time $t_k, k = 0, 1, 2 \dots$ there are n_k targets in the surveillance volume with states $\mathbf{x}_{k,1}, \dots, \mathbf{x}_{k,n_k}$, taking values in the state space $\mathcal{X} \subseteq \mathbb{R}^{n_x}$. Both the number of targets n_k and their individual states in \mathcal{X} are random and time varying. As usual the multi-target state at k , represented by a finite set $\mathbf{X}_k = \{\mathbf{x}_{k,1}, \dots, \mathbf{x}_{k,n_k}\} \in \mathcal{F}(\mathcal{X})$, can conveniently be modeled as a random finite set on \mathcal{X} . The targets are assumed to be noncooperative, meaning that their number and their individual states at any point of time are unknown.

The sensor detection process is imperfect in the sense that some of the targets in \mathbf{X}_k are detected, while the others are missed. In addition, the detector typically creates false detections. Suppose measurement sets are available at time

$k = 1, 2, \dots$. A measurement set at time k contains m_k elements, each taking a value in the observation space $\mathcal{Z} \subseteq \mathbb{R}^{n_z}$. Then the multi-target observation set, $\mathbf{Z}_k = \{\mathbf{z}_{k,1}, \dots, \mathbf{z}_{k,m_k}\} \in \mathcal{F}(\mathcal{Z})$, whose cardinality and individual points in the measurement space are random, can also be modeled by a random finite set.

The Markov multi-target state process is characterized by its initial density $\mathbf{X}_0 \sim \mu(\mathbf{X}_0|\boldsymbol{\theta})$ and the multi-target transitional density $\Pi_{k|k-1}(\mathbf{X}_k|\mathbf{X}_{k-1}, \boldsymbol{\theta})$, conditioned on a static parameter vector $\boldsymbol{\theta} \in \Theta$. The multi-object process is not observed directly, but through the observation process. The observation process is assumed to be conditionally independent given the multi-target state process, and fully specified by the multi-target likelihood $\varphi_k(\mathbf{Z}_k|\mathbf{X}_k, \boldsymbol{\theta})$, again conditioned on $\boldsymbol{\theta} \in \Theta$. For the standard measurement model, the expressions for $\Pi_{k|k-1}(\mathbf{X}_k|\mathbf{X}_{k-1}, \boldsymbol{\theta})$ and $\varphi_k(\mathbf{Z}_k|\mathbf{X}_k, \boldsymbol{\theta})$ are given in (2.33) and (2.34), respectively. The unknown parameters of $\Pi_{k|k-1}(\mathbf{X}_k|\mathbf{X}_{k-1}, \boldsymbol{\theta})$ may include the process noise variance, birth and survival probabilities, birth process intensity, etc. The unknown parameters of $\varphi_k(\mathbf{Z}_k|\mathbf{X}_k, \boldsymbol{\theta})$ may include sensor or environmental characteristics, such as the probability of detection, propagation factors, clutter parameters, sensor biases, measurement noise variances, and so on.

The problem is to estimate the posterior density $p(\boldsymbol{\theta}|\mathbf{Z}_{1:K})$, where $\mathbf{Z}_{1:K} \equiv \mathbf{Z}_1, \dots, \mathbf{Z}_K$ is the observation set sequence accumulated up to discrete-time index K . Having the prior $p_0(\boldsymbol{\theta})$, the solution in the Bayesian framework is given by:

$$p(\boldsymbol{\theta}|\mathbf{Z}_{1:K}) \propto f(\mathbf{Z}_{1:K}|\boldsymbol{\theta}) p_0(\boldsymbol{\theta}). \quad (7.21)$$

The computation of $f(\mathbf{Z}_{1:K}|\boldsymbol{\theta})$, however, is not trivial since it requires to perform inference on the joint space of the parameter vector $\boldsymbol{\theta}$ and the multi-target trajectory (history) $\mathbf{X}_{1:K}$. Thus, our problem is effectively a component of a broader problem where the goal is to find the posterior density:

$$f(\boldsymbol{\theta}, \mathbf{X}_{1:K}|\mathbf{Z}_{1:K}) \propto f(\mathbf{X}_{1:K}|\mathbf{Z}_{1:K}, \boldsymbol{\theta}) f(\mathbf{Z}_{1:K}|\boldsymbol{\theta}) p_0(\boldsymbol{\theta}) \quad (7.22)$$

where $f(\mathbf{X}_{1:K}|\mathbf{Z}_{1:K}, \boldsymbol{\theta})$ is the posterior of the multi-target trajectory given all observation sets and conditioned on $\boldsymbol{\theta}$.

7.2.2 The Proposed Calibration Algorithm

The proposed solution follows the same line of thought as the Monte Carlo batch techniques for parameter estimation of stochastic dynamic systems, reviewed in Cappé et al. [24, Sect. IV]. The key idea is to run a sequential Monte Carlo method to obtain an estimate of $f(\mathbf{X}_{1:K}|\mathbf{Z}_{1:K}, \boldsymbol{\theta})$ for a given value of $\boldsymbol{\theta}$. This provides a simple way of evaluating the observation likelihood $f(\mathbf{Z}_{1:K}|\boldsymbol{\theta})$ and thus to a practical solution for $p(\boldsymbol{\theta}|\mathbf{Z}_{1:K})$.

In practice, however, the difficulty is that the sequential Monte Carlo implementation of the optimal multi-target Bayes filter, which estimates $f(\mathbf{X}_{1:K}|\mathbf{Z}_{1:K}, \boldsymbol{\theta})$, can

be implemented only for a small number of targets (see Sect. 4.3). For this reason we propose to approximate (7.22) with its first-order moment, that is with the probability hypothesis density. Thus (7.22) is replaced with

$$D(\boldsymbol{\theta}, \mathbf{x}_{1:K} | \mathbf{Z}_{1:K}) \propto D(\mathbf{x}_{1:K} | \mathbf{Z}_{1:K}, \boldsymbol{\theta}) f(\mathbf{Z}_{1:K} | \boldsymbol{\theta}) p_0(\boldsymbol{\theta}) \quad (7.23)$$

The step from (7.22) to (7.23) follows from the relationship between the PHD and the multi-target density [7, p. 577]. The advantage of this is that the posterior PHD $D(\mathbf{x}_{1:K} | \mathbf{Z}_{1:K}, \boldsymbol{\theta})$, as opposed to the posterior density $f(\mathbf{X}_{1:K} | \mathbf{Z}_{1:K}, \boldsymbol{\theta})$, is defined on a single-target state space \mathcal{X} .

The equations of the PHD filter for recursive estimation of $D_{k|k}(\mathbf{x}_k | \mathbf{Z}_{1:k}, \boldsymbol{\theta})$ have been presented in Sect. 2.5.2. The sequential Monte Carlo implementation of the PHD filter, which recursively estimates the posterior PHD, has been described in Sect. 4.2.3.

The observation likelihood $f(\mathbf{Z}_k | \mathbf{Z}_{1:k-1}, \boldsymbol{\theta})$, defined as [30]:

$$f(\mathbf{Z}_k | \mathbf{Z}_{1:k-1}, \boldsymbol{\theta}) = \int f(\mathbf{Z}_k | \mathbf{X}_k, \boldsymbol{\theta}) f(\mathbf{X}_k | \mathbf{Z}_{1:k-1}, \boldsymbol{\theta}) \delta \mathbf{X}_k, \quad (7.24)$$

is a by-product of the PHD filter. It can be expressed as [30, Eq. (116)]:

$$\begin{aligned} f(\mathbf{Z}_k | \mathbf{Z}_{1:k-1}, \boldsymbol{\theta}) &= \alpha e^{-\int p_D(\mathbf{x}_k | \boldsymbol{\theta}) D_{k|k-1}(\mathbf{x}_k | \mathbf{Z}_{1:k-1}, \boldsymbol{\theta}) d\mathbf{x}_k} \times \\ &\prod_{\mathbf{z} \in \mathbf{Z}_k} \left(\kappa(\mathbf{z} | \boldsymbol{\theta}) + \int p_D(\mathbf{x}_k | \boldsymbol{\theta}) g(\mathbf{z} | \mathbf{x}_k, \boldsymbol{\theta}) D_{k|k-1}(\mathbf{x}_k | \mathbf{Z}_{1:k-1}, \boldsymbol{\theta}) d\mathbf{x}_k \right) \end{aligned} \quad (7.25)$$

where α is a constant of proportionality, which will be shown later to be unimportant (it cancels out). Therefore, without loss of generality, we adopt $\alpha = 1$. One can compute $f(\mathbf{Z}_{1:K} | \boldsymbol{\theta})$, the likelihood of the measurement set sequence, which features in (7.21) and (7.23), using the decomposition:

$$f(\mathbf{Z}_{1:K} | \boldsymbol{\theta}) = \prod_{k=1}^K f(\mathbf{Z}_k | \mathbf{Z}_{1:k-1}, \boldsymbol{\theta}), \quad (7.26)$$

where $f(\mathbf{Z}_k | \mathbf{Z}_{1:k-1}, \boldsymbol{\theta})$ was given by (7.25). Alternatively, $f(\mathbf{Z}_{1:K} | \boldsymbol{\theta})$ can be computed with higher accuracy by a forward–backward pass of the PHD filter/smother.

The PHD particle filter approximates the PHD $D_{k|k}(\mathbf{x}_k | \mathbf{Z}_{1:k}, \boldsymbol{\theta})$ by a weighted set of particles $\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}_{i=1}^N$, where the weights satisfy $w_k^{(i)} \geq 0$, $i = 1, \dots, N$. The sum of the weights results in an estimate of the expected number of targets. See for details Sect. 4.2.3. Recall that for estimation of the posterior $p(\boldsymbol{\theta} | \mathbf{Z}_{1:K})$, we are only interested in the PHD filter provided approximation of $f(\mathbf{Z}_k | \mathbf{Z}_{1:k-1}, \boldsymbol{\theta})$, given by (7.25).

Let the particle approximation of the predicted PHD $D_{k|k-1}(\mathbf{x}_k|\mathbf{Z}_{1:k-1}, \boldsymbol{\theta})$ be, see (4.39):

$$\widehat{D}_{k|k-1}(\mathbf{x}_k|\mathbf{Z}_{1:k-1}, \boldsymbol{\theta}) = \sum_{i=1}^{N_k} w_{k|k-1}^{(i)} \delta_{\mathbf{x}_{k|k-1}^{(i)}}(\mathbf{x}_k) \quad (7.27)$$

Then the expression in (7.25) can be approximated as:

$$\begin{aligned} \widehat{f}(\mathbf{Z}_k|\mathbf{Z}_{1:k-1}, \boldsymbol{\theta}) &= \exp \left\{ - \sum_{i=1}^{N_k} p_D(\mathbf{x}_{k|k-1}^{(i)}; \boldsymbol{\theta}) w_{k|k-1}^{(i)} \right\} \times \\ &\quad \prod_{\mathbf{z} \in \mathbf{Z}_k} \left(\kappa(\mathbf{z}|\boldsymbol{\theta}) + \sum_{i=1}^{N_k} p_D(\mathbf{x}_{k|k-1}^{(i)}|\boldsymbol{\theta}) g(\mathbf{z}|\mathbf{x}_{k|k-1}^{(i)}, \boldsymbol{\theta}) w_{k|k-1}^{(i)} \right) \end{aligned} \quad (7.28)$$

Although we are primarily interested in the estimation of $\boldsymbol{\theta}$, an estimate of the multi-target history $\widehat{\mathbf{X}}_{1:K}$ may be desirable as a by-product. The particle PHD filter can provide this estimate. An estimation algorithm of multi-target state from a particle representation of the PHD has been described in Sect. 4.2.5.

7.2.3 Importance Sampling with Progressive Correction

It remains to explain how $p(\boldsymbol{\theta}|\mathbf{Z}_{1:K})$ is computed from (7.21) using the estimate of $f(\mathbf{Z}_{1:K}|\boldsymbol{\theta})$ provided by the PHD particle filter/smooth. There is no analytic solution for $p(\boldsymbol{\theta}|\mathbf{Z}_{1:K})$ and hence (7.21) is solved using another Monte Carlo technique, this time the importance sampling method with progressive correction [31].

Quantities of interest related to $\boldsymbol{\theta}$ can be computed from the posterior, for example, the posterior mean is

$$E(\boldsymbol{\theta}|\mathbf{Z}_{1:K}) = \int \boldsymbol{\theta} p(\boldsymbol{\theta}|\mathbf{Z}_{1:K}) d\boldsymbol{\theta}. \quad (7.29)$$

Approximation of (7.29) via importance sampling involves drawing a sample $\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_M$, where M is the sample size, from an importance density q and approximating the integral in (7.29) by a sum,

$$\widehat{\boldsymbol{\theta}} = \sum_{j=1}^M \ell_j \boldsymbol{\theta}_j. \quad (7.30)$$

The weights $\ell_j \geq 0$, $j = 1, \dots, M$ are specified as:

$$\ell_j = \frac{\tilde{\ell}_j}{\sum_{m=1}^M \tilde{\ell}_m}, \text{ with } \tilde{\ell}_j = \frac{f(\mathbf{Z}_{1:K} | \boldsymbol{\theta}_j) p_0(\boldsymbol{\theta}_j)}{q(\boldsymbol{\theta}_j)} \quad (7.31)$$

Note that the calculation of weights requires only a function which is proportional to the posterior density. As discussed earlier, we can only compute an estimate of $f(\mathbf{Z}_{1:K} | \boldsymbol{\theta}_j)/\alpha^K$ using the particle PHD filter. Now we see why the factor of proportionality α is not important: it cancels out by the normalization of weights.

For a large class of importance densities q , approximation (7.30) becomes increasingly accurate as the sample size $M \rightarrow \infty$. However, for a finite M , the accuracy of the approximation depends greatly on the particular importance density. A desirable property of an importance density is that it produces weights with a small variance. Equation (7.31) implies that a good importance density should resemble the posterior. Since the posterior is unknown and importance sampling using the prior is inaccurate for any reasonable sample size, we propose to apply importance sampling with tempering [22, p. 540] or progressive correction [31].

The idea behind progressive correction is to construct a sequence of target distributions from which it is desired to draw samples sequentially. The first target distribution is typically similar to the prior, while the final target distribution is the posterior. The consecutive target distributions in the sequence should not differ too greatly.

Let S denotes the number of stages and ϑ_s , $s = 1, \dots, S$, denote the target distribution for the s th stage. Note that at the final stage S we have $\vartheta_S = p(\boldsymbol{\theta} | \mathbf{Z}_{1:K})$. A series of target distributions which starts from the prior and gradually becomes similar to the posterior can be constructed by setting, for $s = 1, \dots, S$

$$\vartheta_s(\boldsymbol{\theta}) \propto [f(\mathbf{Z}_{1:K} | \boldsymbol{\theta})]^{\Lambda_s} p_0(\boldsymbol{\theta}) \quad (7.32)$$

where $\Lambda_s = \sum_{j=1}^s \beta_j$ with $\beta_j \in (0, 1]$ and $\Lambda_S = 1$. In this way Λ_s is an increasing function of s , upper bounded by one. Note that the intermediate likelihood used for $s < S$ will be broader than the true likelihood, particularly in the early stages. In the later stages, the intermediate likelihood sharpens, so that the sample gradually concentrates in the area of the parameter space suggested by the true likelihood.

The procedure for sequentially drawing samples from target distributions $\vartheta_1, \vartheta_2, \dots, \vartheta_S$ is described as follows. Suppose a sample $\{\boldsymbol{\theta}_j^{s-1}\}_{j=1}^M$ from ϑ_{s-1} is available and we wish to produce a sample from ϑ_s . Note that for $s = 1$, the sample $\{\boldsymbol{\theta}_j^{s-1}\}_{j=1}^M$ is drawn from the prior $p_0 = \vartheta_0$. The first step is to compute weights for samples in $\{\boldsymbol{\theta}_j^{s-1}\}_{j=1}^M$ as: $\tilde{\ell}_j^s = f(\mathbf{Z}_{1:K} | \boldsymbol{\theta}_j^{s-1})^{\beta_s}$ for $j = 1, \dots, M$. The weights are then normalized. In order to derive any benefits from tempering, it is necessary to remove the lower weighted members of the sample $\{\boldsymbol{\theta}_j^{s-1}\}_{j=1}^M$ and diversify the remaining ones. Hence the next step is the resampling step, followed by the MCMC move step [32].

Resampling involves selecting M indices i_1^s, \dots, i_M^s such that $Pr\{i_j^s = j\} = \ell_j^s$. The resampled sample $\{\tilde{\boldsymbol{\theta}}_j^s\}_{1 \leq j \leq M}$ will almost certainly contain duplicate members since the weights $\ell_1^s, \dots, \ell_M^s$ will most likely be uneven. In order to remove dupli-

cation and thus increase the sample diversity, the MCMC move step is applied to the resampled sample $\{\tilde{\theta}_j^s\}_{1 \leq j \leq M}$. For each member of $\{\tilde{\theta}_j^s\}_{1 \leq j \leq M}$, a new sample member is proposed as a draw

$$\theta_j^{s,*} \sim g_s(\cdot | \tilde{\theta}_j^s)$$

where g_s denotes the proposal density for the s th stage. Then the Metropolis-Hastings scheme is applied, whereby $\theta_j^{s,*}$ is accepted with certain probability or rejected. In this way we form a new sample $\{\theta_j^s\}_{j=1}^M$, where $\theta_j^s = \theta_j^{s,*}$, if the move is accepted, and $\theta_j^s = \tilde{\theta}_j^s$, if the move is rejected. The acceptance probability is adopted as

$$\eta = \min \left\{ 1, \frac{\vartheta_s(\theta_j^{s,*}) g_s(\tilde{\theta}_j^s | \theta_j^{s,*})}{\vartheta_s(\tilde{\theta}_j^s) g_s(\theta_j^{s,*} | \tilde{\theta}_j^s)} \right\}.$$

The proposal density g_s should produce candidates over a reasonably large area of the parameter space (in order to increase diversity), but within the area of high likelihood. A suitable proposal can be selected as follows [33]:

$$g_s(\theta^* | \theta) = g_s(\theta^*) = \mathcal{N}(\theta^*; \hat{\mu}_s, \varepsilon \hat{\mathbf{C}}_s) \quad (7.33)$$

where as usual $\mathcal{N}(\theta; \mu, \mathbf{C})$ is the Gaussian distribution; $\hat{\mu}_s$ and $\hat{\mathbf{C}}_s$ are the sample mean and covariance matrix, respectively, of a weighted sample $\{\ell_j^s, \theta_j^{s-1}\}_{j=1}^M$, and ε is a user defined parameter.

The computational expense of tempering depends on the number of stages S and the correction factors β_1, \dots, β_S . While a small value of S is favored for computational reasons, the successive intermediate distributions ϑ_s are made more similar by choosing large S . An adaptive scheme proposed in Musso et al. [31] is used here to balance the conflicting requirements for S .

The pseudo-code of the proposed algorithm for joint estimation of θ and $\mathbf{X}_{1:K}$ is given in Algorithm 13. It incorporates the adaptive selection of correction factors β_s , in lines 11 and 12. Here $1 \leq H \leq M$ and $\phi > 0$ are user defined parameters which control the the number and values of correction factors. The proposed estimation algorithm for the calibration vector θ is primarily designed to compute the sample $\{\theta_j^S\}_{j=1}^M$ which approximates the posterior $p(\theta | \mathbf{Z}_{1:K})$. As a by-product, however, it can also compute an estimate of the multi-target history $\widehat{\mathbf{X}}_{1:K}$ (see line 31 in Algorithm 13).

7.2.4 Application to Sensor Bias Estimation

Multi-sensor bias estimation has attracted considerable interest in the tracking community, see for example Okello and Ristic [29], Lin et al. [27], Qi et al. [28] and

Algorithm 13 Parameter estimation using particle PHD importance sampling

1: **Input:** Accumulated measurement sets $\mathbf{Z}_{1:K}$; prior p_0
2: **Initialization:**
3: Set $s \leftarrow 0$, $\Lambda_0 \leftarrow 0$
4: **for** $j = 1, \dots, M$ **do**
5: Draw $\theta_j^0 \sim p_0$
6: Run the PHD particle filter (see Algorithm 6) using θ_j^0 to estimate $\psi_j^0 = \log f(\mathbf{Z}_{1:K} | \theta_j^0)$
7: **end for**
8: **Progressive correction:**
9: **while** $\Lambda_s < 1$ and $s < S$ **do**
10: $s \leftarrow s + 1$
11: Sort negative log-likelihoods: $-\psi_{(1)}^{s-1} < -\psi_{(2)}^{s-1} < \dots < -\psi_{(M)}^{s-1}$
12: $\beta_s \leftarrow \min\{1 - \Lambda_{s-1}, -\phi/\psi_{(H)}^{s-1}\}$
13: $\Lambda_s \leftarrow \Lambda_{s-1} + \beta_s$
14: For $j = 1, \dots, M$, compute $\tilde{\ell}_j^s = \exp\{\beta_s \cdot \psi_j^{s-1}\}$
15: Normalization: $\ell_j^s = \tilde{\ell}_j^s / \sum_{m=1}^M \tilde{\ell}_m^s$
16: Compute the sample mean $\hat{\mu}_s$ and covariance matrix $\hat{\mathbf{C}}_s$
17: **for** $j = 1, \dots, M$ **do**
18: Select $i_j = j$ with probability ℓ_j^s and set $\tilde{\theta}_j^s = \theta_{i_j}^{s-1}$ and $\tilde{\psi}_j^s = \psi_{i_j}^{s-1}$
19: Draw $\theta_j^{s,*} \sim g_s(\cdot; \hat{\mu}_s, \hat{\mathbf{C}}_s)$
20: Run the PHD particle filter (Algorithm 6) using $\theta_j^{s,*}$ to estimate $\psi_j^{s,*} = \log f(\mathbf{Z}_{1:K} | \theta_j^{s,*})$
21: Compute acceptance probability $\eta = \min \left\{ 1, \exp \left[\beta_s (\psi_j^{s,*} - \tilde{\psi}_j^s) \right] \frac{g_s(\tilde{\theta}_j^s)}{g_s(\theta_j^{s,*})} \right\}$
22: Draw $u \sim \mathcal{U}_{[0,1]}$
23: **if** $u < \eta$ **then**
24: Set $\theta_j^s = \theta_j^{s,*}$ and $\psi_j^s = \psi_j^{s,*}$
25: **else**
26: Set $\theta_j^s = \tilde{\theta}_j^s$ and $\psi_j^s = \tilde{\psi}_j^s$
27: **end if**
28: **end for**
29: **end while**
30: Compute the mean or the maximum a posteriori estimate $\widehat{\theta}$ from the sample $\{\theta_j^s\}_{j=1}^M$
31: Run PHD particle filter using $\widehat{\theta}$ to obtain an estimate of the multi-target history $\widehat{\mathbf{X}}_{1:K}$
32: **Output:** Sample $\{\theta_j^s\}_{j=1}^M$ and $\widehat{\mathbf{X}}_{1:K}$

references therein. This section illustrates the proposed calibration algorithm as a solution to multi-sensor bias estimation using the targets of opportunity. The sensors can operate asynchronously with imperfect detection ($p_d < 1$ and false alarms), the targets can dynamically appear and disappear from the surveillance volume, and the usual requirement in sensor bias estimation, to know the association of measurements to targets, is not required.

7.2.4.1 Specification of the Bias Estimation Problem

Let us adopt a 2D scenario, where measurements are collected by two static sensors with overlapping coverage. The state vector of each individual target is $\mathbf{x} = [x \ \dot{x} \ y \ \dot{y}]^\top$, where (x, y) denotes the position of a target and (\dot{x}, \dot{y}) its velocity. Sensor measurements are affected by two sources of error: the systematic error (or the bias) and the stochastic error (measurement noise). Let the measurement set reported at time k by sensor $r_k \in \{1, 2\}$ be denoted $\mathbf{Z}_k^{(r_k)}$. Let us assume that sensors measure the range and azimuth to targets. For a measurement $\mathbf{z} \in \mathbf{Z}_k^{(r_k)}$ which originates from a target $\mathbf{x} \in \mathbf{X}_k$, the single-object likelihood is given by:

$$g_k^{(r_k)}(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}) = \mathcal{N}\left(\mathbf{z}; h_k^{(r_k)}(\mathbf{x}) - \boldsymbol{\theta}^{(r_k)}, \boldsymbol{\Sigma}_k^{(r_k)}\right). \quad (7.34)$$

where the measurement function is given by

$$h_k^{(r)}(\mathbf{x}) = \begin{bmatrix} \sqrt{(x - x_r)^2 + (y - y_r)^2} \\ \arctan\left(\frac{x - x_r}{y - y_r}\right) \end{bmatrix}, \quad (7.35)$$

and (x_r, y_r) are the coordinates of sensor r . Vector $\boldsymbol{\theta}$ consist of four components, i.e.,

$$\boldsymbol{\theta} = [\Delta\rho_1 \ \Delta\eta_1 \ \Delta\rho_2 \ \Delta\eta_2]^\top$$

where $\boldsymbol{\theta}^{(r)} = [\Delta\rho_r \ \Delta\eta_r]^\top$, which features in (7.34), is the bias vector for sensor $r = 1, 2$ ($\Delta\rho$ is the range bias, whereas $\Delta\eta$ is the azimuth bias). The covariance matrix in (7.34) is $\boldsymbol{\Sigma}_k^{(r)} = \text{diag}\left[\left(\sigma_\rho^{(r)}\right)^2, \left(\sigma_\eta^{(r)}\right)^2\right]$. The probability of detection of sensor r is $p_d^{(r)}$. The false detections are modeled by a Poisson point process, that is the intensity function $\kappa^{(r)}(\mathbf{z}|\boldsymbol{\theta})$ for sensor r , which features in (2.52), is $\kappa^{(r)}(\mathbf{z}|\boldsymbol{\theta}) = \lambda_r s_r(\mathbf{z}|\boldsymbol{\theta})$, where λ_r is the mean number of false detections (at one time instant) and $s_r(\mathbf{z}|\boldsymbol{\theta})$ is their distribution over the measurement space \mathcal{L} .

Each object state evolves in time according to the Markov transitional density, which is independent of the parameter vector $\boldsymbol{\theta}$. In particular, we adopt the nearly constant velocity model, that is:

$$\pi_{k|k-1}(\mathbf{x}|\mathbf{x}') = \mathcal{N}(\mathbf{x}; \mathbf{F}_k \mathbf{x}', \mathbf{Q}_k) \quad (7.36)$$

where as usual

$$\mathbf{F}_k = \mathbf{I}_2 \otimes \begin{bmatrix} 1 & T_k \\ 0 & 1 \end{bmatrix}, \quad \mathbf{Q}_k = \mathbf{I}_2 \otimes \varpi \begin{bmatrix} \frac{T_k^3}{2} & \frac{T_k^2}{2} \\ \frac{T_k^2}{2} & T_k \end{bmatrix}, \quad (7.37)$$

Table 7.1 Correction factors and MCMC acceptance rates during the progressive correction

s	1	2	3	4	5	6	7	8	9
β_s	0.0229	0.0473	0.0768	0.1032	0.1308	0.1577	0.1845	0.2114	0.0653
Acceptance rate (%)	9.3	20.7	20.2	22.8	11.9	12.2	5.3	6.9	33.6

with \otimes being the Kroneker product, $T_k = t_{k+1} - t_k$ the sampling interval and ϖ the intensity of process noise.

7.2.4.2 Numerical Results

The considered scenario with a typical measurement set sequence $\mathbf{Z}_{1:K}$ is illustrated in Fig. 7.6a. The total number of measurement sets available for estimation is $K = 15$. A varying number of targets is present in the scenario, with $n_k = 3$ for $k = 1, 2, 3$, $n_k = 5$ for $k = 4, \dots, 13$ and $n_k = 4$ for $k = 14, 15$. The two sensors operate asynchronously; sensor 1 collects measurement sets at $k = 1, 3, 5, 7, 9, 11, 13, 15$, while sensor 2 collects measurement sets at $k = 2, 4, 6, 8, 10, 12, 14$. The time interval between k and $k - 1$ is constant and equals $T_k = 3$ s. The two sensors are placed at $(x_1, y_1) = (0, 0)$ and $(x_2, y_2) = (120 \text{ km}, 35 \text{ km})$. The probability of detection is $p_d^{(1)} = p_d^{(2)} = 0.95$, the clutter parameters are: $\lambda_1 = \lambda_2 = 10$, while $s_1(\mathbf{z}|\boldsymbol{\theta}) = s_2(\mathbf{z}|\boldsymbol{\theta})$ are uniform distributions over $\mathcal{L} = [0 \text{ km}, 300 \text{ km}] \times [-\pi, \pi]$. The standard deviations of measurement noise are $\sigma_\rho^{(1)} = \sigma_\rho^{(2)} = 50 \text{ m}$ and $\sigma_\eta^{(1)} = \sigma_\eta^{(2)} = 0.5^\circ$. The true values of sensor biases are $\Delta\rho_1 = 6.8 \text{ km}$, $\Delta\eta_1 = -3.5^\circ$, $\Delta\rho_2 = -5 \text{ km}$, $\Delta\eta_2 = 2^\circ$.

The PHD particle filter uses $M_b = 500$ particle per newborn target and $M_p = 1,000$ particles per persistent target, with $p_s = 0.95$ and the expected number of target birth is $v_{k|k-1}^b = 0.05$. The prior $p_0(\boldsymbol{\theta})$ is a uniform distribution over $[-10 \text{ km}, +10 \text{ km}] \times [-5^\circ, +5^\circ] \times [-10 \text{ km}, +10 \text{ km}] \times [-5^\circ, +5^\circ]$. The number of samples used in the bias space is $M = 1,000$. The parameters used in adaptive selection of correction factors are: $\phi = 50$ and $H = 0.6M$. This resulted in a sequence of correction factors listed in Table 7.1. The table also shows the acceptance rate of the MCMC move at each stage $s = 1, \dots, S = 9$.

The results of sensor bias estimation are shown in Fig. 7.7. Each figure in Fig. 7.7 displays the (normalized) histogram of the output sample $\{\boldsymbol{\theta}_j^s\}_{j=1}^M$, which approximates the posterior $p(\boldsymbol{\theta}|\mathbf{Z}_{1:K})$, marginalized to one dimension: (a) $\Delta\rho_1$, (b) $\Delta\eta_1$, (c) $\Delta\rho_2$, (d) $\Delta\eta_2$. The kernel density estimate (KDE) of the corresponding marginalized density is also shown for each sensor bias (solid lines), as well as the true value of the bias (vertical dashed line). The maximum a posterior (MAP) estimates of sensor biases (computed from the KDE) are shown in Table 7.2.

Figure 7.6b shows an estimate of the multi-target state estimate $\hat{\mathbf{X}}_{1:K}$, obtained using the MAP estimate of the sensor bias vector.

Fig. 7.6 The simulation setup. **a** The location of sensors (marked by square and denoted S_1 and S_2) and the accumulated dataset $\mathbf{Z}_{1:K}$. **b** The multi-target state estimate $\hat{\mathbf{X}}_{1:K}$ (i.e., five short trajectories) resulting from the proposed algorithm

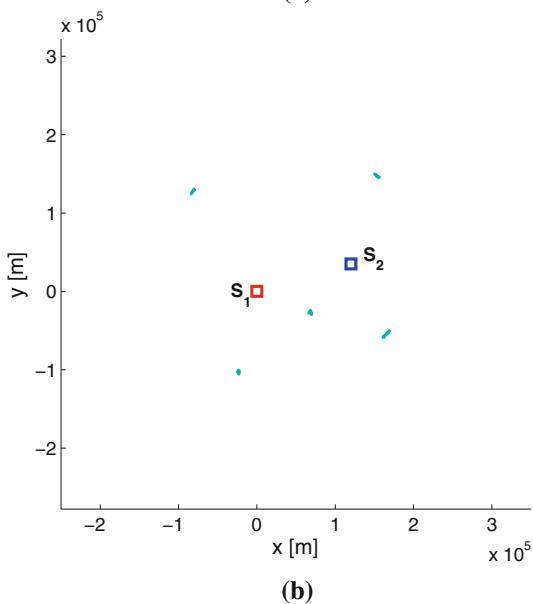
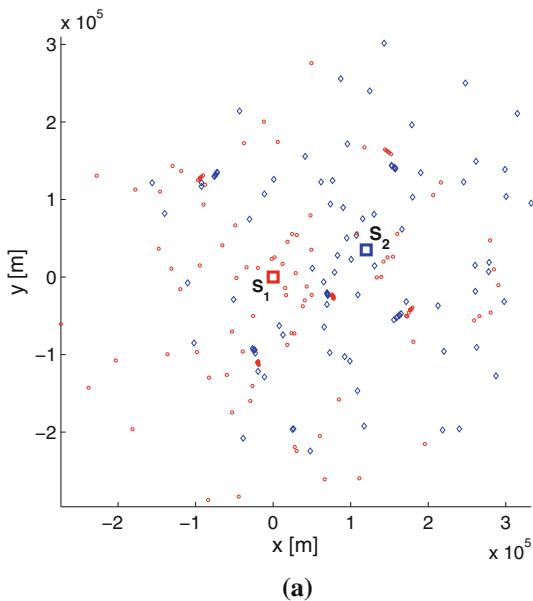
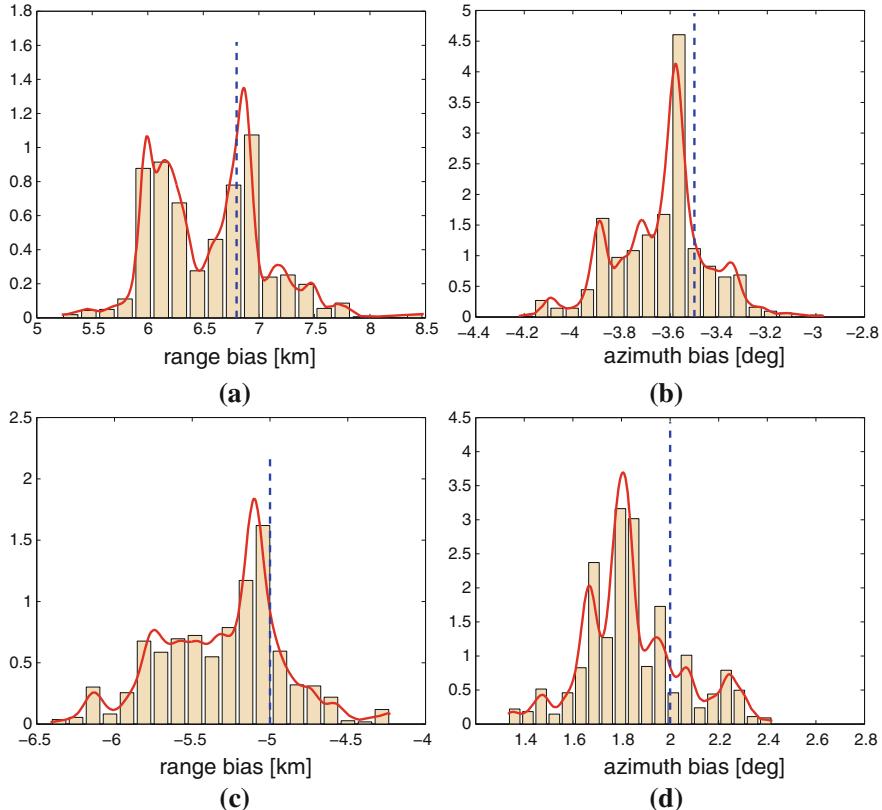


Table 7.2 True and estimated sensor biases

	$\Delta\rho_1$ [km]	$\Delta\eta_1$ [deg]	$\Delta\rho_2$ [km]	$\Delta\eta_2$ [deg]
True values	6.8	-3.5	-5	2
Estimated values	6.87	-3.58	-5.10	1.80

**Fig. 7.7** Normalized histograms of the bias sample $\{\boldsymbol{\theta}_j^S\}_{j=1}^M$ approximating the posterior $p(\boldsymbol{\theta}|\mathbf{Z}_{1:K})$, marginalized to $\Delta\rho_1$ (a), $\Delta\eta_1$ (b), $\Delta\rho_2$ (c), and $\Delta\eta_2$ (d). The kernel density estimates shown by red solid lines; true values indicated by vertical blue dashed lines

7.2.4.3 Discussion

The obtained numerical results indicate a remarkably accurate performance of the proposed algorithm, even using a relatively small dataset of only 15 scans. Some important questions, however, remain for future work. For example, while it is expected that observability of parameter $\boldsymbol{\theta}$ improves with the increase in the number of targets n_k (for example, if $n_k = 0$ then $\boldsymbol{\theta}$ cannot be estimated), a more rigorous study of observability remains elusive.

We have seen that the PHD particle filter provides the estimate of the likelihood $f(\mathbf{Z}_{1:K} | \boldsymbol{\theta})$. Hence, the accuracy of the proposed calibration method depends significantly on the number of particles used in the PHD particle filter. In order to reduce the Monte Carlo variance of the likelihood estimate, there is a need to use a reasonably large number of particles. This variability, however, can be also reduced by application of the backward pass through the data [26], as in the PHD smoother [34, 35].

The accuracy of parameter estimation also depends on the number of samples M in the parameter (bias vector) space, and the selection of correction factors β in the progressive correction scheme. The results in Table 7.1 indicate a fairly low acceptance rate in our simulations. The MCMC acceptance rate can be increased using a lower value of parameter ϕ , which would result in smaller increments of correction factors and consequently to an increase in the number of stages S .

A recursive version of the proposed technique, applicable to the case where $\boldsymbol{\theta}$ is slowly varying, has been presented in Ristic and Clark [19].

References

1. S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Artech House, 1999.
2. G. Bradski and A. Kaehler, *Learning OpenCV*. O'Reilly, Media, 2008.
3. J. Vermaak, N. Ikoma, and S. Godsill, "A sequential Monte Carlo framework for extended object tracking," *IEE Proc. Radar, Sonar and Navig.*, vol. 152, no. 5, pp. 353–363, 2005.
4. K. Gilholm and D. Salmond, "Spatial distribution model for tracking extended objects," *IEE Proc. Radar, Sonar Navig.*, vol. 152, no. 5, pp. 364–371, 2005.
5. J. W. Koch, "Bayesian approach to extended object and cluster tracking using random matrices," *IEEE Trans Aerospace and Electronic Systems*, vol. 44, no. 3, pp. 1042–1059, 2008.
6. in *Proc. Int. Conf. Information Fusion*, (Chicago, USA), July 2011. Special Session: Extended Object and group target tracking.
7. R. Mahler, *Statistical Multisource Multitarget Information Fusion*. Artech House, 2007.
8. K. Gilholm, S. Godsill, S. Maskell, D. Salmond, "Poisson models for extended target and group tracking," in, *Proc. SPIE*, vol. 5913, 2005.
9. R. Mahler, "PHD filters for nonstandard targets I: Extended targets," in *Proc. 12th Int. Conf. Information Fusion*, (Seattle, USA), July 2009.
10. U. Orguner, C. Lundquist, and K. Granström, "Extended target tracking with a cardinalized probability hypothesis density filter," in *Proc. 14th Int. Conf. Information Fusion*, (Chicago, USA), July 2011.
11. A. Swain and D. Clark, "The single-group PHD filter: an analytic solution," in *Proc. 14th Int. Conf. Information Fusion*, (Chicago, USA), July 2011.
12. Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, 2001.
13. "PETS: performance evaluation of tracking and surveillance." <http://www.cvg.rdg.ac.uk/slides/pets.html>
14. J. Shi and C. Tomasi, "Good features to track," in *Proc. 9th IEEE Conference on Computer Vision and, Pattern Recognition*, pp. 593–600, June 1994.
15. A. Swain and D. E. Clark, "First-moment filters for spatial independent cluster processes," *Proceedings of SPIE* 7697, 2010.

16. A. Swain and D. Clark, "Extended object filtering using spatial independent cluster processes," *Proc. conf. Information Fusion (FUSION)*, 2010.
17. R. Mahler, "Detecting, tracking, and classifying group targets: a unified approach," in *Proceedings of SPIE*, vol. 4380, pp. 217–228, 2001.
18. R. Mahler, "An Extended First-Order Bayes Filter for Force Aggregation," in *Proceedings of SPIE*, vol. 4728, pp. 196–207, 2002.
19. B. Ristic and D. Clark, "Particle filter for joint estimation of multi-object dynamic state and multi-sensor bias," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, (Kyoto, Japan), pp. 3877–3880, March 2012.
20. C.-S. Lee, D. E. Clark, and J. Salvi, "SLAM with Single Cluster PHD Filters," *Proc. International Conference on Robotics and Automation*, 2012.
21. D. E. Clark, C.-S. Lee, and S. Nagappa, "Single-cluster PHD filtering and smoothing for SLAM applications," *Proc. International Conference on Robotics and Automation*, 2012.
22. C. P. Robert and G. Casella, *Monte Carlo statistical methods*. Springer, 2nd ed., 2004.
23. B. Jungbacker and S. J. Koopman, "Monte Carlo estimation for nonlinear non-gaussian state space models," *Biometrika*, vol. 94, no. 4, pp. 827–839, 2007.
24. O. Cappé, S. J. Godsill, and E. Moulines, "An overview of existing methods and recent advances in sequential Monte Carlo," *Proc. IEEE*, vol. 95, no. 5, pp. 899–924, 2007.
25. C. Andrieu, A. Doucet, and R. Holenstein, "Particle Markov chain Monte Carlo methods," *Journ. Royal Statistical Soc. B*, vol. 72, no. Part 3, pp. 269–342, 2010.
26. F. Lindsten and T. B. Schön, "On the use of backward simulation in the particle Gibbs sampler," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, (Kyoto, Japan), pp. 3845–3848, March 2012.
27. X. Lin, Y. Bar-Shalom, and T. Kirubarajan, "Multisensor-multitarget bias estimation for general asynchronous sensors," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 3, pp. 899–921, 2005.
28. Y. Qi, Z. Jing, and S. Hu, "General solution for asynchronous sensor bias estimation," in *Proc. 11th Int. Conf. Information Fusion*, (Cologne, Germany), 2008.
29. N. Okello and B. Ristic, "Maximum likelihood registration for multiple dissimilar sensors," *IEEE Trans. Aerospace and Electronic Systems*, vol. 39, no. 3, 2003.
30. R. P. S. Mahler, "Multi-target Bayes filtering via first-order multi-target moments," *IEEE Trans. Aerospace & Electronic Systems*, vol. 39, no. 4, pp. 1152–1178, 2003.
31. C. Musso, N. Oudjane, and F. LeGland, "Improving regularised particle filters," in *Sequential Monte Carlo methods in Practice* (A. Doucet, N. deFreitas, and N. J. Gordon, eds.), ch. 12, New York: Springer-Verlag, 2001.
32. B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman filter: Particle filters for tracking applications*. Artech House, 2004.
33. M. Morelande and B. Ristic, "Radiological source detection and localisation using bayesian techniques," *IEEE Trans Signal Processing*, vol. 57, no. 11, pp. 4220–4231, 2009.
34. R. P. S. Mahler, B.-T. Vo, and B. N. Vo, "Forward-backward probability hypothesis density smoothing," *IEEE Trans. Aerospace and Electronic Systems*, 2012.
35. D. E. Clark, "First-Moment Multi-Object Forward-Backward Smoothing," *Int. Conf. on Information Fusion*, 2010.

Index

A

Autonomous bearings-only tracking, 87

B

Bernoulli

- box-particle filter, 56, 59, 61
- filter, 22, 23, 53, 121, 132
 - extended target, 145, 146, 148, 150
- particle filter, 55, 62, 128, 134, 140
 - sensor control, 87, 91
- random finite set, 17, 22

Binomial random finite set, 17, 18, 147

Bootstrap filter, 6, 7, 9, 10

Box-particle filter, 14

C

Cardinality distribution, 16–18, 24–26, 66, 70, 73, 82

CPHD

- filter, 22, 25, 62, 104, 140
- particle filter, 67, 70, 71, 73, 76

D

Dempster–Shafer theory, 45–47, 50

E

Exact Bayes multi-object filter, 20, 79, 83

G

Generalized likelihood function, 10, 11, 14, 29, 31, 32, 37, 41, 42, 47

H

Histogram of oriented gradients, 137, 139

I

- IID cluster random finite set, 17, 105
- Importance density, 5, 6, 54, 80
- Imprecise
 - likelihood, 11, 14, 45–47
 - measurement, 10, 12, 56, 57
 - measurement function, 11, 12, 15, 29, 30
- Information gain, 7, 8, 89, 91, 99, 102, 104, 115

K

Kernel density estimation, 76, 168

M

- MCMC move, 6, 165
- Mixed labeling, 81, 114
- Model-based classification, 44
- Multi-Bernoulli

- filter, 22
- random finite set, 17, 19, 147
- Multi-sensor bias estimation, 160, 165
- Multi-target state particle filter, 80, 82
 - sensor control, 112
- Multi-target tracking, 121, 122, 129, 140, 157

N

Natural language, 14, 39

O

OSPA metric, 20, 21, 79, 123–125, 153
 for tracks, 127, 128
 OSPA-T metric, 102, 128, 141

P

PHD
 filter, 22, 24, 104, 121, 136, 160
 particle filter, 67, 70, 73, 76, 79, 107, 109,
 110, 112, 135, 160
 sensor control, 108
 Poisson random finite set, 17, 19, 24, 105
 Progressive correction, 6, 164

R

Renyi divergence, 8, 95, 104, 106, 107, 110,
 112
 Random finite set, 15, 16, 81, 85, 147
 Received signal strength, 29
 Reward function, 7–9, 85

S

Sensor control, 6, 7, 96, 104
 Set
 Dirac delta function, 17
 integral, 16, 20, 85, 86, 149
 SIR compartmental model, 34, 35
 Syndromic data, 36

T

Tracker performance evaluation, 122

U

Uncertain implication rule, 12, 42