



RANDOM FINITE SETS IN MULTI-TARGET TRACKING

EFFICIENT SEQUENTIAL MCMC IMPLEMENTATION

MÉLANIE BOCQUEL

RANDOM FINITE SETS IN MULTI-TARGET TRACKING

EFFICIENT SEQUENTIAL MCMC IMPLEMENTATION

by

Mélanie Bocquel

Members of the doctoral committee:

Prof. dr. A. Bagchi	University of Twente, EWI	Promotor
Dr. P.K. Mandal	University of Twente, EWI	Assistant Promotor
Dr. ir. J.N. Driessen	Thales Nederland	Referent
Prof. dr. A.A. Stoorvogel	University of Twente, EWI	
Prof. dr. ir. R.N.J. Veldhuis	University of Twente, EWI	
Prof. dr. F. Le Gland	IRISA/INRIA Rennes, France	
Dr. D.E. Clark	Heriot-Watt university Edinburgh, Scotland	
Prof. dr. ir. A.J. Mouthaan	University of Twente, EWI	Chairman and Secretary

THALES

UNIVERSITY OF TWENTE.

The research described in this dissertation was undertaken at THALES NEDERLAND in partnership with the Department of Applied Mathematics, Faculty of Electrical Engineering, Mathematics and Computer Science, UNIVERSITY OF TWENTE.



This research work has been carried out in the MC IMPULSE project: <https://mcimpulse.isy.liu.se>. Funding for this research project was provided by the EU's Seventh Framework Programme under grant agreement n°238710.

Typeset with \LaTeX

Printed by Gildeprint Drukkerijen, The Netherlands.

Cover design by Patrick Schulenberg & Mélanie Bocquel.

CTIT

CTIT Ph.D. Thesis Series No. 13-270
Centre for Telematics and Information Technology
P.O. Box 217,
7500 AE Enschede, The Netherlands.

RANDOM FINITE SETS IN MULTI-TARGET TRACKING

EFFICIENT SEQUENTIAL MCMC IMPLEMENTATION

DISSERTATION

to obtain
the degree of doctor at the University of Twente,
on the authority of the rector magnificus,
prof. dr. H. Brinksma,
on account of the decision of the graduation committee,
to be publicly defended
on Friday, 25th October 2013 at 14.45
by

Mélanie Anne Édith Bocquel

born on 6 December 1987
in Rennes, France.

This dissertation has been approved by:

Promotor: Prof. dr. A. Bagchi,
Assistant-promotor: Dr. P.K. Mandal.

Copyright © 2013 by Mélanie Bocquel, Enschede, The Netherlands.

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

ISBN 978-90-365-0578-9

<http://dx.doi.org/10.3990/1.9789036505789>

ISSN 1381-3617 (CTIT Ph.D. thesis Series No. 13-270)

To Patrick and my family

Acknowledgements

The first acknowledgement goes to my supervisor dr. Hans Driessen, for his never-ending enthusiasm and encouragement along the way. His open door policy and constant availability were essential. Further I would like to thank prof. dr. Arun Bagchi and dr. Pranab Mandal for their academic guidance, patience and scientific support throughout the project.

I would like to thank the members of my defense committee, prof. dr. Anton Stoorvogel, prof. dr. ir. Raymond Veldhuis, prof. dr. François Le Gland and dr. Daniel Clark. Their reviews, comments and fruitful suggestions helped me improve the manuscript and reach its final shape.

During my period as a PhD student, I have had the chance to visit some prominent research groups, which have provided new perspectives. I would like to sincerely thank prof. dr. François Le Gland and dr. Daniel Clark. It has been a genuine pleasure to have had the opportunity to work with such passionate, stimulating and friendly people.

Furthermore, I would like to thank the Marie Curie fellowship researchers and colleagues, for providing me with a pleasant work environment. Especially dr. Yvo Boers, Fotios Katsilieris, dr. Francesco Papi and dr. Martin Podt. Your collaborations, discussions and feedback are immensely appreciated.

During my PhD studies I have met a lot of amazing people who definitely deserve a special mention. I would like to thank dr. Mayazzurra Ruggiano, dr. Rienk Bakker, Jitse Zwaga, Jan Karelse and Bob Fels. I would like to thank my student colleagues for their support and for the enjoyable time I had during my stay at Thales Nederland.

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

On a personal note, I am grateful to my family and friends for their loving support through all the ups and downs - especially Patrick for all the love, happiness and support he brought into my life. I finally have more time to spend with you.

Contents

1	Introduction	1
1.1	Motivation and Scope	1
1.2	Key Contributions and Outline	5
1.3	Publications	7
2	Introduction to Multi-target Tracking	9
2.1	Bayesian Filtering	10
2.1.1	The Bayes Filter	10
2.1.2	The Kalman Filter	12
2.1.3	The Particle Filter	14
2.2	Multi-target Tracking	18
2.2.1	Common Multi-target Tracking Techniques	18
2.2.2	Random Finite Set Multi-target Tracking	21
2.2.3	Multi-target State Estimation	25
2.2.4	Multi-target Performance Evaluation	28
3	RFS Approach to Multi-target TBD	33
3.1	Track Before Detect versus Classical Radar Tracking	33
3.1.1	Basic Radar Principles	33
3.1.2	Radar Signal Processing	34
3.2	Track-Before-Detect Tracking	37
3.2.1	Multi-target Bayes Filter (scheme)	37
3.2.2	Multi-target Dynamic Model	38
3.2.3	Multi-target Observation Model	41
3.2.4	Multi-target Bayes Recursion	44
4	SMC Implementations of the Multi-target Bayes Filter	47
4.1	Standard SMC Implementation	48
4.2	Sequential Markov Chain Monte Carlo	50
4.3	Interacting Population-based MCMC-PF	51
4.3.1	Justifications behind the IP-MCMC-PF Algorithm	51
4.3.2	IP-MCMC-PF Algorithm	52
4.4	IP based Reversible Jump MCMC-PF	57
4.4.1	Bayesian Estimation and Reversible-Jump MCMC	57
4.4.2	Reversible-Jump M-H Sampling Procedure	59
4.4.3	Reversible-Jump M-H Moves and Proposals	61
4.5	Multiple Cardinality Hypotheses Particle Filter	65
4.5.1	Justifications behind the MCHT	65
4.5.2	Labeled Random Finite Sets	67
4.5.3	Labeled Multi-target Bayes Recursion	70

4.5.4	Particle MCHT Implementation	74
4.6	System Setup	78
4.6.1	Dynamical Models	78
4.6.2	Ambiguities and Observation Model	80
4.6.3	Measurement Gating	85
4.6.4	Genetic Algorithms	86
4.7	Experimental Results	87
4.7.1	Scenario 1: Fixed and known number of targets ($n_k = 3$)	87
4.7.2	Scenario 2: Fixed and known number of targets ($n_k = 10$)	89
4.7.3	Scenario 3: Range/Doppler ambiguity and eclipsing issues	94
4.7.4	Scenario 4: Time varying and unknown number of targets ($n_{max} = 3$)	97
4.7.5	Scenario 5: Time varying and unknown number of targets ($n_{max} = 10$)	101
5	Approximations of the Multi-target Bayes Filter	107
5.1	Survey of the PHD and CPHD Filters	108
5.1.1	Probability Hypothesis Density (PHD) Filter	108
5.1.2	Particle PHD Filter Implementation	111
5.1.3	The Cardinalized Probability Hypothesis Density (CPHD) Filter	113
5.1.4	Particle CPHD Filter Implementations	116
5.2	Cardinality Balanced Multi-target Multi-Bernoulli Filter	121
5.2.1	Cardinality Balanced Multi-target Multi-Bernoulli (CBMeMBer) Filter	121
5.2.2	Particle CBMeMBer Filter Implementation	125
5.2.3	Multi-Bernoulli TBD Filter	127
5.2.4	Particle Multi-Bernoulli TBD Filter Implementation	128
5.3	Numerical Study	130
5.3.1	Scenario 1: Plot tracking using SMC implementation	130
5.3.2	Scenario 2: TBD tracking using SMC implementation	135
6	Exploiting External Knowledge	139
6.1	Constrained Bayesian Filtering	140
6.1.1	Introduction to Constrained Filtering	140
6.1.2	Exploiting External Knowledge through Constraints	141
6.1.3	Tracking with Hard Constraints - Review of Sequential Monte Carlo Methods	143
6.1.4	Bayesian Smoothing - Sequential Monte Carlo Approach	146
6.2	Knowledge-Based Fixed-Lag Smoother	150
6.2.1	Knowledge-Based Fixed-Lag Smoother	150
6.2.2	Multiscan Knowledge Exploitation Gain based on Entropy Reduction	152
6.3	Multiscan Knowledge Exploitation using IP-MCMC-PF	154
6.3.1	IP-MCMC-PF for Target Tracking while Exploiting External Knowledge	154
6.3.2	Experimental Results	158
7	Conclusions	163
7.1	Conclusions	163
7.2	Directions for Future Works	164







A	Finite Set Statistics (FISST)	167
A.1	Random Finite Set (RFS)	167
A.2	Multi-target Probability Density Functions	169
A.3	Set Integral	169
A.4	Probability Generating Functionals (p.g.fl.)	169
A.5	Functional Derivatives and Set Derivatives	171
A.6	Probability Hypothesis Density (PHD)	172
B	Common Point Processes	173
B.1	Bernoulli RFS	173
B.2	I.I.D. Cluster process	174
B.3	Poisson RFS with Poisson rate $\lambda > 0$	175
B.4	Poisson cluster process	176
B.5	Multi-Bernoulli RFS	176
C	Markov Chain Monte Carlo (MCMC)	179
C.1	Monte Carlo Integration	179
C.1.1	The Monte Carlo Principle	179
C.1.2	Rejection Sampling	180
C.1.3	Importance Sampling	181
C.2	Markov Chain Monte Carlo	183
C.2.1	Basic of Markov Chain Monte Carlo	184
C.2.2	Metropolis-Hastings and Gibbs Samplers	184
C.2.3	MCMC Convergence Diagnostics	187
C.2.4	Reversible Jump MCMC	188

List of Algorithms

1	SIR Particle Filter outline	16
2	Multi-target SIR Particle Filter outline	49
3	M-H algorithm	53
4	M-H sampling procedure	55
5	IP-MCMC-PF algorithm	56
6	IP-RJMCMC-PF algorithm	58
7	Reversible-Jump M-H sampling procedure	60
8	Sampling a labeled multi-Bernoulli RFS	68
9	MCHPF algorithm	75
10	IP-MCMC-PF algorithm	76
11	Random Walk M-H sampling procedure	77
12	Pseudo-code of the SMC-PHD filter	112
13	Pseudo-code of the SMC-CPHD filter	120
14	Pseudo-code of the SMC-MeMBer TBD filter	129
15	Rejection-Sampling Particle Filter	144
16	Pseudo-Measurement Particle Filter	145
17	Smoother realization	148
18	Knowledge-Based Fixed-Lag Smoother (Forward Filtering Backward Smoothing) . . .	151
19	Multi-target Tracker	156
20	Random Walk M-H sampling procedure	157
21	Rejection-Sampling algorithm	180
22	M-H algorithm	185
23	Gibbs sampler	186

List of Figures

2.1	Bayesian network	11
3.1	Radar principle	34
3.2	Data and signal processing	36
3.3	Bernoulli survival RFS $S(\mathbf{x}')$	38
4.1	Three-level tree-based structure.	66
4.2	Typical measurement data.	80
4.3	Range ambiguity illustration	82
4.4	Range eclipsing illustration	83
4.5	Doppler eclipsing illustration	84
4.6	True and relative bearings	84
4.7	True tracks on a 2-D plane for Scenario 1.	87
4.8	Empirical posterior distribution (.) and MAP estimate (+) over a single trial. Each filter uses $N = 500$ particles.	88
4.9	Estimated trajectories (i.e., particles-based conditional mean) over 100 Monte Carlo runs. Each filter uses $N = 500$ particles.	88
4.10	RMSE over 100 Monte Carlo trials: SIR-PF vs IP-MCMC-PF.	89
4.11	True tracks on a 2-D plane for Scenario 2.	90
4.12	Empirical posterior distribution (.) and MAP estimate (+) over a single trial. Each filter uses $N = 1000$ particles.	91
4.13	Estimated trajectories (i.e., particles-based conditional mean) over 100 Monte Carlo runs. Each filter uses $N = 1000$ particles.	91
4.14	RMSE over time for the SIR-PF (■ <i>blue</i>), PMMH (■ <i>green</i>), and IP-MCMC-PF(■ <i>red</i>).	92
4.15	True tracks on a 2-D plane for Scenario 3.	94
4.16	Empirical posterior distribution along MAP estimate over a single trial.	95
4.15	Empirical posterior distribution along MAP estimate over a single trial.	96
4.16	RMSE over time for the SIR-PF and the IP-MCMC-PF.	96
4.17	True tracks for Scenario 4.	97
4.18	Estimated trajectories (i.e., particles-based conditional mean) over 100 Monte Carlo runs. Each filter uses $N = 500$ particles.	98
4.19	Zoom on tracks produced by the IP-RJMCMC-PF filter.	98
4.20	Cardinality statistics (mean) versus time. The IP-RJMCMC-PF filter initiates and terminates tracks with a very short delay.	99
4.21	Cardinality statistics (mean and standard deviation) versus time.	99
4.22	OSPA distance ($p = 2$, $c = 100$ m)	100
4.23	OSPA localization and cardinality components	100
4.24	True tracks on a 2-D plane for Scenario 5.	101

4.25	Estimated trajectories (i.e., particles-based conditional mean) over 100 Monte Carlo runs. Each filter uses $N = 1000$ particles. The proposed MCHPF provides better tracking performance than the IP-RJMCMC-PF filter.	102
4.26	Zoom on tracks produced by the IP-RJMCMC-PF filter.	102
4.27	Zoom on tracks produced by the MCHPF filter.	103
4.28	OSPA localization and cardinality components	103
4.29	OSPA penalty ($p = 2$, $c = 100$ m)	104
4.30	Cardinality statistics (mean and standard deviation) versus time. The MCHPF filter initiates and terminates tracks with a very short delay.	104
4.31	Existence probability associated to each target versus time.	105
5.1	True tracks in polar coordinates for Scenario 1.	131
5.2	SMC-CBMeMber filter estimates, measurements and true target tracks in x and y coordinates versus time.	132
5.3	Cardinality statistics (mean and standard deviation) versus time.	133
5.4	OSPA distance ($p = 2$, $c = 100$ m)	134
5.5	OSPA localization and cardinality components	134
5.6	True tracks in polar coordinates for Scenario 2.	136
5.7	SMC-CBMeMber and SIR-PF filters estimates and true target tracks in x and y coordinates versus time.	136
5.8	OSPA distance ($p = 2$, $c = 5$ m)	137
5.9	OSPA localization and cardinality components	137
6.1	Scenario 1: a ship travels within a canal (shipping lane) at a nearly-constant speed of 30 m.s^{-1} . No measurement available between $k \in [10, 21] \cup [41, 54]$ s.	158
6.2	Empirical posterior distribution ( <i>green</i>) and conditional mean ( <i>red</i>) over a single trial. When no measurement is available, due to the loss of particles diversity the SIR-PF arbitrarily retains one way (i.e., self resolving), while the IP-MCMC-PF keeps all possible directions.	159
6.3	Empirical posterior distribution ( <i>green</i>) and conditional mean ( <i>red</i>) over a single trial, zoom for $k \in [5, 25]$. Each filter uses a lag $L = 5$ and $N = 250$ particles.	159
6.4	Estimated trajectories (i.e., particles-based conditional mean) over 100 Monte Carlo runs. Each filter uses a lag $L = 5$ and $N = 1000$ particles.	160
6.5	Scenario 2: 4 ships travel within a canal at nearly-constant speed.	160
6.6	Empirical posterior distribution (.) and conditional mean (+) over a single trial. Each filter uses a lag $L = 4$ and $N = 1000$ particles.	161
6.7	Estimated trajectories (i.e., particles-based conditional mean) over 100 Monte Carlo runs. Each filter uses a lag $L = 4$ and $N = 1000$ particles.	161
6.8	Time-Averaged Position RMSE over 100 Monte Carlo trials: SIR-PF and KB-Smoother ( <i>blue</i>), IP-MCMC-PF and KB-Smoother ( <i>red</i>).	162
C.1	Rejection sampling: Sample a candidate $x^{(i)}$ and a uniform variable u . Accept the candidate sample if $u \leq q(x^{(i)})/p(x^{(i)})$, otherwise reject it.	180
C.2	SIR vs M-H	183

List of Acronyms

ANOVA	ANalysis Of VAriance
AR	AutoRegressive
avg.CPU	average process time
CBMeMber	Cardinality-Balanced Multi-target Multi-Bernoulli
CPHD	Cardinalized Probability Hypothesis Density
DLC	Delay-Line Canceler
EAP	Expected A Posteriori
EKF	Extended Kalman Filter
FFBS	Forward Filtering Backward Smoothing
FISST	Finite-Set Statistics
FoV	Field-of-View
GA	Genetic Algorithm
GM	Gaussian Mixture
GNN	Global Nearest Neighbour
HMC	Hamiltonian Monte Carlo
HMM	Hidden Markov Model
i.i.d.	independent and identically distributed
IMM	Interacting Multiple dynamical Model
IP-MCMC-PF	Interacting Population-based MCMC-PF
IP-RJMCMC-PF	IP based Reversible-Jump MCMC-PF
IR	InfraRed
JoM	Joint Multi-target
JPDA	Joint Probabilistic Data Association
JTC	Joint Tracking and Classification
KB	Knowledge-Based
KB-Smoother	Knowledge-Based fixed-lag Smoother
KF	Kalman Filter
M ³ H	Multiple Model Multiple Hypothesis
M-H	Metropolis-Hastings
MALA	Metropolis-Adjusted Langevin Algorithm
MaM	Marginal Multi-target
MAP	Maximum A Posteriori
MC	Monte Carlo
MCHPF	Multiple Cardinality Hypotheses Particle Filter
MCHT	Multiple Cardinality Hypotheses Tracker
MCMC	Markov Chain Monte Carlo
MeMber	Multi-target Multi-Bernoulli
MHT	Multiple Hypothesis Tracking
MMH	Marginal Metropolis-Hastings

MMSE	Minimum Mean-Square Error
MRF	Markov Random Field
MTI	Moving Target Indicator
MTT	Multi-Target Tracking
MV	Minimum Variance estimate
NCT	Nearly Coordinated Turn
NCV	Nearly Constant Velocity
OMAT	Optimal MASS Transfer
OSPA	Optimal Sub-Pattern Assignment
p.g.fl.	probability generating functional
PCR	Pulse Compression Ratio [%]
PDF	Probability Density Function
PF	Particle Filter
PHD	Probability Hypothesis Density
PMMH	Particle Marginal Metropolis-Hastings
PRF	Pulse Repetition Frequency [Hz]
PRI	Pulse Repetition Interval [s]
PSRF	Potential Scale Reduction Factor
radar	radio detection and ranging
RF	Radio-Frequency [Hz]
RFS	Random Finite Set
RJMCMC	Reversible-Jump Markov Chain Monte Carlo
RMSE	Root-Mean-Square Error
SIR	Sampling Importance Resampling
SIS	Sampling Importance Sampling
SLLN	Strong Law of Large Numbers
SMC	Sequential Monte Carlo
SMCMC	Sequential Markov Chain Monte Carlo
SME	Symmetric Measurement Equation
SNR	Signal-to-Noise Ratio
TBD	Track-Before-Detect
TLR	Tracking Loss Rate
UKF	Unscented Kalman Filter

List of Symbols

$\mathbb{N}, \mathbb{Z}, \mathbb{R}, \mathbb{C}$	The sets natural number (≥ 0), integers, real numbers, and complex numbers, respectively
\mathbb{R}^n	The n dimensional Euclidean space
\mathbb{Z}^+	The strictly positive integers
Ω	The sample space
\emptyset	The empty set
\mathbb{P}	A probability measure
\mathcal{P}	A probability distribution
p	A probability density
$\delta_x(\cdot)$	The Dirac delta function centered on x
$1_{\mathbf{A}}(\cdot)$	The indicator function on the set \mathbf{A}
$ \mathbf{A} $	The cardinality of (number of elements in) the set \mathbf{A}
$\langle \cdot, \cdot \rangle$	The inner product between two continuous functions or two discrete sequences
$\mathbb{E}[\cdot]$	The expectation operator
$\mathbb{E}[\cdot \cdot]$	The conditional expectation operator
$\text{Var}[\cdot]$	The variance operator
$O(\cdot)$	The upper bound for asymptotic complexity
$\mathbf{A} \setminus \mathbf{B} = \mathbf{A} \cap \mathbf{B}^C$	The set difference between two set \mathbf{A} and \mathbf{B}
\mathcal{X}	The state space
\mathcal{Z}	The observation space
\mathbf{x}_k	The vector-valued state at time k
$\hat{\mathbf{x}}_k$	An estimate of the vector-valued state at time k
\mathbf{z}_k	The vector-valued observation at time k
$\mathbf{z}_{1:k}$	The sequence of vector-valued observations up to and including time k
$\pi_{k k-1}(\mathbf{x}_k \mathbf{x}_{k-1})$	The single-target Markov transition density from time $k-1$ to k
$g_k(\mathbf{z}_k \mathbf{x}_k)$	The single-target measurement likelihood at time k
$\mathcal{F}(\mathcal{X})$	The set of all finite subsets of the state space
$\mathcal{C}(\mathcal{X})$	The set of all closed subsets of the state space
$\mathcal{F}(\mathcal{Z})$	The set of all finite subsets of the observation space
\mathbf{X}_k	The finite-set-valued state at time k
$\hat{\mathbf{X}}_k$	An estimate of the finite-set-valued state at time k
\mathbf{Z}_k	The finite-set-valued observation at time k
$\mathbf{Z}_{1:k}$	The sequence of finite-set-valued observations up to and including time k
$\Pi_{k k-1}(\mathbf{X}_k \mathbf{X}_{k-1})$	The multi-target Markov transition density from time $k-1$ to k
$\vartheta_k(\mathbf{Z}_k \mathbf{X}_k)$	The multi-target measurement likelihood at time k
$\beta_{\mathbf{X}}(S) = \mathbb{P}(\mathbf{X} \subseteq S)$	The belief mass function of the RFS \mathbf{X}
$(dF)_{\mathbf{x}}$	The set derivative of a function F at the point \mathbf{x}

$(dF)_{\{\mathbf{x}_1, \dots, \mathbf{x}_n\}}$	The set derivative at a finite set $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$
$\int_S f(\mathbf{X}) \delta \mathbf{X}$	The set integral of a function f over a set S
$\delta_{\mathbf{X}}(\cdot) = \sum_{\mathbf{x} \in \mathbf{X}} \delta_{\mathbf{x}}(\cdot)$	The counting measure, sum of Dirac delta functions
$f_{k k-1}$	The Bayes predicted density from time $k-1$ to k
f_k	The Bayes posterior density at time k
$D_{k k-1}$	The predicted intensity function from time $k-1$ to k
D_k	The posterior intensity function at time k
$\rho_{k k-1}$	The predicted cardinality distribution from time $k-1$ to k
ρ_k	The posterior cardinality distribution at time k
$r_{k k-1}^{(\cdot)}$	The predicted multi-Bernoulli existence probability from time $k-1$ to k
$r_k^{(\cdot)}$	The posterior multi-Bernoulli existence probability at time k
$s_{k k-1}^{(\cdot)}$	The predicted multi-Bernoulli density from time $k-1$ to k
$s_k^{(\cdot)}$	The posterior multi-Bernoulli density at time k
$\mathbf{X}_k^{(i)} = [\mathbf{x}_{k,1}^{(i)}, \dots, \mathbf{x}_{k,n_k}^{(i)}]$	The multi-target state vector (joint state of n_k partitions) of the i^{th} sample (particle) at time k
$w_k^{(i)}$	The weight of the i^{th} sample (particle) at time k
$G_{k k-1}$	The predicted p.g.fl. from time $k-1$ to k
G_k	The posterior p.g.fl. at time k
$G_k^{(n)}$	The n -th order functional derivative of the p.g.fl. G_k
$\mathcal{N}(\cdot; m, P)$	The Gaussian density function with mean m and covariance P
N	The number of particles
N_{MCMC}	The number of MCMC chains operating in parallel
N_{M-H}	The number of samples per MCMC chain
B	The number of burn-in samples

Introduction

1.1 Motivation and Scope

Target tracking is an ever-growing research challenge arising in various disciplines ranging from econometrics, image/signal processing to biomedical engineering [51, 74, 100]. The heart of the matter lies in combining sensed data (uncertain and noisy observations) and a priori knowledge to provide a reliable, accurate and timely estimate of an unknown quantity or outcome (often called state). For example, one might wish to know the values of shares on the stock market, the location of surrounding aircraft or the vasculature of a tumor. Early research in this area focused on the single-target tracking problem. This problem presupposes that a single target is present, and this, throughout the whole process. As a result, the tracking problem reduces to the on-line estimation of the state of this target, based on the observations. The single-target tracking problem can be formulated and solved in a Bayesian setting by representing the target state probabilistically and incorporating statistical models for the sensing action and the target state transition [9, 12]. Research in Multi-Target Tracking (MTT) has shown significant progress in recent years. Compared to single-target tracking, the appealing MTT problem poses both additional challenges and opportunities.

To introduce the MTT problem, let us consider a simplified example: assume an air surveillance system comprising a radio detection and ranging (radar) which collects the echo signal received after reflection from aircrafts in the Field-of-View (FoV), called targets. Imagine that multiple aircrafts are crossing the FoV of the radar and that weather conditions have deteriorated such that some of the targets are not properly detected during the scan and that false observations are reported. From a Bayesian perspective, the multi-target tracking problem consists in inferring the number of targets, their identities and their individual kinematic properties, from a sequence of noisy and cluttered measurements provided by one or more sensors.

More specifically, a multi-target tracking algorithm should be able to:

- track the behavior of all targets under surveillance,
- recognize new targets as they enter as well as identify when a target disappears,
- distinguish targets as separate entities.

This definition hides much of the complexity of the multi-target tracking and poses significant technical challenges. For example, the algorithm does not know beforehand how many targets need to be tracked, and thus should be ready, at any time instant, to deal with any number of new targets. In addition, targets of interest may be temporally obscured, and the algorithm must be able to keep these targets in track even when they cannot be viewed directly. Targets may also interact, altering each others' behavior. Finally, the observations given to an algorithm are not assigned to unique, identifiable targets on beforehand, and thus the way observations correspond to targets must be inferred as well.

The primary focus of this dissertation is to develop an effective and efficient multi-target tracking algorithm dealing with an unknown and time-varying number of targets.

Traditional tracking algorithms are designed assuming that the sensor provides a set of point measurements¹. For example, in image processing, images are converted to spatial point patterns and in radar tracking, the radar video is converted to detection plots. Compressing the information into a finite set of points is efficient in terms of memory as well as computational requirements.

The last two decades have witnessed an intensive research interest in MTT with increasingly innovative systems. Although driven primarily by aerospace applications such as radar/sonar surveillance, navigation, and air traffic control, nowadays MTT applications span a wide variety of disciplines, including robotics, image processing, remote sensing and biomedical research to name a few.

A large proportion of the research undertaken relates to how to handle an unknown and time-varying number of targets. This represents a major challenge in multi-target tracking as opposed to the single-target case. At the current state of the art, there exist essentially two mainstream approaches to tackle this problem:

The traditional approach This approach adopts a divide-and-conquer strategy: first by performing efficient measurement-to-target association and then applying standard Bayesian filtering techniques to solve each of the multiple single-target tracking problems. This so-called Bayesian data association approach suffers from a number of technical difficulties in its mathematical foundation as well as the combinatorial growth in the number of hypotheses with the number of targets and measurements. In addition, most of these techniques generally use linearized models and Gaussian noise approximation so that the Kalman Filters (KFs) [82] can be applied and hence perform poorly when the non-linearity present in dynamical models is severe or the noise is no longer Gaussian. In this case the inaccuracies introduced by linearization are further added up to the errors caused by incorrect associations. Sequential Monte Carlo (SMC) techniques capable of handling nonlinear and non-Gaussian dynamical models are also problematic due to the large number of required particles.

Alternative formulations that avoid explicit associations between measurements and targets have been proposed, including inter alia Symmetric Measurement Equations (SMEs) [83] and Random Finite Sets (RFSs) [68, 95].

The SME filter This filter removes the data association uncertainty from the original measurement equation by applying a symmetric transformation to the measurement vector. However, existing SME filters suffer from strong nonlinearities and lack an intuitive semantic. Furthermore, the generalization of existing symmetric transformations to multidimensional states is nontrivial due to the so-called ghost target problem [89] resulting from non-injective transformations. Additionally, these symmetric transformations are unsuitable for larger target numbers as the order of the involved polynomial increases with the number of targets.

¹pre-processed measurements constructed from the original measurement data at every time step

The RFS approach This approach is an emerging and promising alternative to the traditional association-based methods. In the RFS formulation, the collection of individual targets is treated as a set-valued state, and the collection of individual observations is treated as a set-valued observation. Once the mathematical tools of Finite-Set Statistics (FISST) [95] are provided, modeling set-valued states and set-valued observations as RFSs allows the multi-target posterior distribution to be propagated using the Bayes recursion, as done in the single-target Bayes filter. FISST extends seamlessly the formal Bayes modeling to multi-target problems by generalizing probability densities and calculus methods so that ideas from statistics and information theory can be extended to RFSs.

An intent of this dissertation is to assert that the RFS framework not only is a natural, elegant and rigorous foundation, but also leads to practical, efficient and reliable algorithms for Bayesian multi-target tracking.

Another research focus concerns the tracking of targets in low Signal-to-Noise Ratio (SNR) environments. In the multi-target tracking case, point measurements based approaches can be very effective for a wide range of applications. However, these approaches may be undesirable for applications with low SNR, since the information loss incurred in the compression can be significant, and in such cases it can clearly be advantageous to make use of all information contained in the original or "raw" measurement, as proposed in the so-called Track-Before-Detect (TBD) approach [28, 31, 124]. The basic motivation for this approach is that all available information should be used to update the multi-target state estimate in the hope of squeezing more accurate and robust performance.

An objective of this dissertation is to provide several novel, efficient and reliable RFS based tracking algorithms suitable for, among others, the TBD surveillance application.

Central to the developed algorithms and methods is the notion of the recursive Bayesian filter, a commonly accepted approach for recursive state estimation. The most popular and commonly adopted filter is the Kalman Filter (KF) [82], a simple and elegant algorithm formulated more than 50 years ago as an optimal recursive Bayesian state estimator for the (restricted) class of linear Gaussian systems. Since its 1960 introduction, it has been the subject of extensive research projects and applications, particularly in the area of autonomous or assisted navigation systems and in the aerospace industry. In practice, the use of the KF is limited by the ubiquitous nonlinearity and non-Gaussianity of physical world. In general, the nonlinear filtering problem consists in finding the conditional probability distribution (or density) of the state given the observations up to current time. This leads us to the key research issue: a Probability Density Function (PDF) is fully characterized only when all its moments, which are oftentimes infinite in number, are taken into consideration. Therefore, apart from a few exceptions (finite state space filters, the KF discussed above and the Benès filter), the filtering problem is infinite dimensional [41] and thus it cannot be solved without numerical approximations. Three important classes of numerical methods are nowadays available:

The moment methods The state uncertainty is characterized by moments of the PDF truncated up to a pre-determined order [96]. The Extended Kalman Filter (EKF) is a special case which considers only the first two moments leading to Gaussian closure. As with EKF, the higher order filters are subject to similar issues of poor performance/divergence depending on the degree of nonlinearity and/or intensity of noise in the underlying system.

The projection techniques This class of filters assumes the state conditional PDF to belong to a family of functions and determines the parameters associated with the said functions [34]. In other words, the projection filter is a finite dimensional nonlinear filter based on the differential geometric approach to statistics. By using geometry, the infinite dimensional equation for the optimal filter is projected onto a finite dimensional space.

The SMC methods such as Particle Filter (PF) The state uncertainty is characterized in terms of the statistics of a sample of points called particles. Each particle carries a weight and is individually propagated forward through the stochastic dynamics of the system. The particle weights are updated as new measurements come in and the desired moments of the PDF are computed by studying the statistics of the weighted particles. Since the true PDF of the state is not known a-priori, the particles are sampled from a "proposal density function" or "importance sampling function", that is known and relatively easy to sample from [53, 70]. Non-Gaussian noise assumptions and incorporation of constraints on some of the system parameters can also be performed in a natural way by using these simulation based methods.

This dissertation will focus on this latter approach. In TBD context, the filtering distribution of interest is complex and highly nonlinear. PFs can be used to carry out the inference. However, given the high dimensionality of the state space, designing an efficient PF implementation is not straightforward. In recent years, there has been an increasing interest in Markov Chain Monte Carlo (MCMC) methods to simulate complex, nonstandard multivariate distributions. The research done on Markov chains and Hidden Markov Models (HMMs) has led the definition of the Metropolis-Hastings (M-H) algorithm through the notion of reversibility [43]. Such theoretical results have gathered interest in the mathematical community and allowed for the derivation of convergence results for Sequential Markov Chain Monte Carlo (SMCMC) methods [84, 126]. In SMC methods the correction between the proposal distribution and the "target" distribution is performed using an importance sampling step. SMCMC methods propose to replace the traditional importance sampling step with an MCMC step. These methods allow to design effective MCMC algorithms in complex, high-dimensional scenarios where standard SMC strategies failed. The efficiency of the SMCMC algorithms proposed in this dissertation are enhanced by incorporating various sampling improvement strategies.

Further research seeks to exploit external information to increase surveillance system performance. In multi-target scenarios, kinematic constraints from the interaction of targets with their environment or other targets can restrict target motion. Such motion constraint information could improve tracking performance if effectively used by the tracker.

1.2 Key Contributions and Outline

The scope of this dissertation is to derive from the RFS formalism the full multi-target Bayes filter and provide several novel, efficient and reliable tracking algorithms suitable for the specific TBD surveillance application. The organization of the dissertation is described below and the contributions are highlighted.

Chapter 1 provides an overview of the thesis, outlines the motivation and summarizes the key contributions.

Chapter 2 views the tracking problem as a Bayesian inference problem and highlights the benefits of this approach. This chapter sets out to substantiate the RFS approach [95], introduce the RFS formalism for Bayesian multi-target filtering and lay the foundation for the focus of this dissertation. The coverage is intended to be intuitive and self-explanatory.

Chapter 3 first introduces the concept of Track-Before-Detect (TBD) [29, 124] and emphasizes the potential payoffs of this approach. Then, an optimal multi-target Bayes filter is derived, from the RFS formalism, for Track-Before-Detect (TBD) applications.

Chapter 4 contains practical SMC implementations of the RFS multi-target Bayes recursion. Modern tracking systems need to efficiently process large amounts of measurement data and accurately estimate target states while operating in complex tracking settings. This chapter details the following main contributions dealing with both the complexity of tracking scenarios and the large amount of data to be processed.

- Section 4.3 introduces a novel algorithm well suited to deal with multi-target tracking problems for a given cardinality. The proposed algorithm, denoted by Interacting Population-based MCMC-PF (IP-MCMC-PF) [24], makes use of several M-H samplers running in parallel, which interact through genetic variation.

Extensions of the IP-MCMC-PF algorithm are then proposed to handle a large, unknown and time-varying number of targets,

- Section 4.4 describes a novel algorithm, denoted by IP based Reversible-Jump MCMC-PF (IP-RJMCMC-PF) [25], which exploits reversible jumps. Incorporation of Reversible-Jump Markov Chain Monte Carlo (RJMCMC) methods [72] into a tracking framework gives the possibility to naturally and efficiently deal with multiple appearing and disappearing targets, and makes the statistical inference more tractable. However, such an approach strongly relies on the correct knowledge of the Markov matrix that is used to model the evolution of the cardinality over time.
- Section 4.5 presents an alternative algorithm, denoted by Multiple Cardinality Hypotheses Particle Filter (MCHPF), which uses the IP-MCMC-PF as the core engine of a Multiple Cardinality Hypotheses Tracker (MCHT). To address the uniqueness of labels the proposed algorithm is built upon the concept of labeled RFSs [139], and formally incorporates the propagation and estimation of track labels within the RFS filtering framework.

This chapter also investigates some of the practical aspects of PF design for our specific application: Radar surveillance. Finally demonstrations and numerical studies of the proposed implementations are reported.

Chapter 5 considers sub-optimal moment, cardinality and RFS density approximations to the full multi-target Bayes filter. This chapter briefly summarizes the Probability Hypothesis Density (PHD) [92, 129], the Cardinalized Probability Hypothesis Density (CPHD) [93, 137] and the Cardinality-Balanced Multi-target Multi-Bernoulli (CBMeMBer) recursions [140]. The main contribution of this chapter is set out below.

- Section 5.2.3 proposes a novel CBMeMBer filter suitable for TBD applications.
- Section 5.2.4 describes a generic SMC implementation of the CBMeMBer-TBD recursion.
- Section 5.3 demonstrates the performance of the SMC-CBMeMBer filter on simulated data in a TBD context.

Chapter 6 seeks to exploit external information to increase surveillance system performance. This chapter introduces a multi-scan procedure for Bayes optimal knowledge exploitation, denoted by Knowledge-Based fixed-lag Smoother (KB-Smoother) [111]. The main contribution of this chapter is as follows.

- Section 6.3 proposes a robust and accurate tracker which combines the KB-Smoother and the IP-MCMC-PF [26].

Chapter 7 provides a synopsis and critical appraisal of the key results, and finally sketches future research directions.

1.3 Publications

Published papers of the author that are of relevance to this dissertation are listed below.

Reviewed Journal Articles

- M. Bocquel, F. Papi, M. Podt and H. Driessen. Multi-target Tracking with Multiscan Knowledge Exploitation using Sequential MCMC sampling. In *IEEE Journal of Selected Topics In Signal Processing*, volume 7(3), pages 532-542, June 2013.

Reviewed Conference Proceedings

- M. Bocquel, H. Driessen and A. Bagchi. Multi-target Tracking with IP Reversible Jump MCMC-PF. In *Proc. of the 16th International Conference on Information Fusion*, Istanbul 2013.
- M. Bocquel, H. Driessen and A. Bagchi. Multi-target Tracking with Interacting Population-based MCMC-PF. In *Proc. of the 15th International Conference on Information Fusion*, pages 74-81, Singapore 2012. (Student Paper Award/Travel Grant)
- F. Papi, M. Bocquel, M. Podt and Y. Boers. Fixed-Lag Smoothing for Bayes Optimal Exploitation of External Knowledge. In *Proc. of the 15th Int. Conf. on Information Fusion*, pages 463-470, Singapore 2012.
- M. Bocquel, H. Driessen and A. Bagchi. Multi-target Particle Filter addressing Ambiguous Radar Data in TBD. In *Proc. of the IEEE Radar Conference*, pages 575-580, Atlanta 2012.
- M. Bocquel, A. Lepoutre, O. Rabaste and F. Le Gland. Optimisation d'un filtre particulaire en contexte Track-Before-Detect (TBD) [in French]. In *Actes du 23ème Colloque GRETSI sur le Traitement du Signal et des Images*, Bordeaux 2011.

Memorandum

- M. Bocquel. *Labeled Random Finite Sets in Multi-target Track-Before-Detect*. Memorandum 2012, Department of Applied Mathematics, University of Twente, Enschede, September, 2013.

Symposium

- M. Bocquel, H. Driessen and A. Bagchi. Nonlinear Bayesian Filtering with interacting population-based MCMC-PF. In *CTIT symposium*, University of Twente, June 18, 2012.

Journal Articles Under Review

- F. Papi, M. Bocquel, M. Podt and Y. Boers. Fixed-Lag Smoothing for Bayes Optimal Knowledge Exploitation in Target Tracking. Submitted to *IEEE Transactions on Signal Processing*, May 2013.

Introduction to Multi-target Tracking

Bayesian filtering is aimed to apply the Bayesian statistics and Bayes rule to the stochastic filtering problem, its objective being to recursively estimate the state evolution of a stochastic dynamical system from partial observations. This approach yields analytical solutions, in closed form, only in the case of linear systems with Gaussian statistics. In the case of non-linearity and/or non-Gaussian statistics, numerical solutions can be obtained by applying SMC methods. In this approach, tracking is performed first by applying a predefined model of the expected dynamics to predict the target states, and then by using the noisy measurement to obtain the posterior probability of these states.

The Multi-Target Tracking (MTT) problem involves the joint detection and estimation of an unknown and possibly time varying number of targets and of their individual states from a sequence of observations provided by one or more sensors. To date, there are two mainstream approaches to address the multi-target tracking problem:

- The traditional approach adopts a divide-and-conquer strategy, by first performing efficient measurement-to-target association, on the basis of detection plots and then applying standard Bayesian filtering techniques to solve each of the multiple single-target tracking problems.
- The RFS approach regards the multi-target state and the measurement as RFSs to jointly estimate the number of targets and their states.

The chapter is organized as follows. The section 2.1 views the tracking problem as a Bayesian inference problem and highlights the benefits of this approach. The section 2.2 first presents a survey of traditional Bayesian multi-target tracking techniques and questions their soundness. This section then reviews the basics of RFS theory, and presents, in the Bayesian RFS framework, a mathematically consistent and association free formulation of the multi-target filtering and estimation problem.

2.1 Bayesian Filtering

In the following, a summary of the Bayes filter is given, as well as two common implementations, a special closed form solution in the linear Gaussian cases known as the Kalman filter, and a generic point mass approximation known as the Particle or Sequential Monte Carlo filter.

2.1.1 The Bayes Filter

Bayesian filtering is based on the mathematical theory of stochastic filtering [2, 10, 11] that allows us to model the uncertainty about the world and the outcomes of interest by incorporating prior knowledge and observable evidence. A key element of the Bayesian paradigm is the use of probability distributions to describe all relevant unknown quantities, interpreting the probability of an event as a conditional measure of uncertainty about the occurrence of the event under some specific conditions.

Let us consider the discrete-time state-space approach to the modeling of dynamical systems. At each time k , information about the system is described by a state vector \mathbf{x}_k taking values in a state space $\mathcal{X} \subseteq \mathbb{R}^{n_x}$. The system state is indirectly observed via a noisy measurement vector \mathbf{z}_k taking values in an observation space $\mathcal{Z} \subseteq \mathbb{R}^{n_z}$. The filtering or dynamic state estimation problem is concerned with sequentially estimating the state \mathbf{x}_k of the dynamic system given the measurement history $\mathbf{z}_{1:k} \equiv (\mathbf{z}_1, \dots, \mathbf{z}_k)$.

The time evolution of the state vector is described by a stochastic model in the form of a Markov transition:

$$\mathbf{x}_k = f_k(\mathbf{x}_{k-1}, \mathbf{v}_{k-1}), \quad (2.1)$$

where,

- $f_k : \mathbb{R}^{n_x} \times \mathbb{R}^{n_v} \rightarrow \mathbb{R}^{n_x}$ is a possibly nonlinear function of the state \mathbf{x}_{k-1} ,
- $\{\mathbf{v}_{k-1}, k \in \mathbb{N}\}$ is an independent and identically distributed (i.i.d.) process noise sequence,
- n_x, n_v are dimensions of the state and process noise vectors;

which specifies the transformation of any given state vector \mathbf{x}_{k-1} at time $k-1$ and system noise \mathbf{v}_{k-1} at time $k-1$ into a new state vector \mathbf{x}_k at time k . Alternatively, the time evolution of the state vector is described by a Markov transition density $\pi_{k|k-1}(\cdot|\cdot)$.

The modeling of measurement vectors is described by an observation equation:

$$\mathbf{z}_k = h_k(\mathbf{x}_k, \mathbf{w}_k), \quad (2.2)$$

where,

- $h_k : \mathbb{R}^{n_x} \times \mathbb{R}^{n_w} \rightarrow \mathbb{R}^{n_z}$ is a possibly nonlinear function of the state \mathbf{x}_k ,
- $\{\mathbf{w}_k, k \in \mathbb{N}\}$ is an i.i.d. measurement noise sequence,
- n_z, n_w are dimensions of the measurement and measurement noise vectors;

which specifies at time k the transformation of any given state vector \mathbf{x}_k and measurement noise \mathbf{w}_k into a measurement vector \mathbf{z}_k . Alternatively, the modeling of the measurement vector is described by a likelihood function $g_k(\cdot|\cdot)$, where $g_k(\mathbf{z}_k|\mathbf{x}_k)$ measures the adequacy of the state \mathbf{x}_k , first guessed, w.r.t. the measurement \mathbf{z}_k . The probability density of the measurement history $\mathbf{z}_{1:k}$ conditioned on the system trajectory $\mathbf{x}_{1:k}$ is given by:

$$\vartheta_{1:k}(\mathbf{z}_{1:k}|\mathbf{x}_{1:k}) = g_k(\mathbf{z}_k|\mathbf{x}_k) g_{k-1}(\mathbf{z}_{k-1}|\mathbf{x}_{k-1}) \dots g_1(\mathbf{z}_1|\mathbf{x}_1). \quad (2.3)$$

The above HMM is described pictorially in Figure 2.1 by a Bayesian network which shows the conditional independence relations.

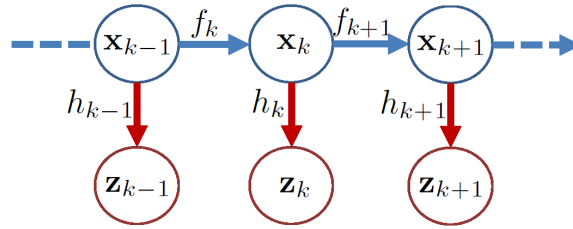


Figure 2.1: Bayesian network

From a Bayesian perspective, the tracking problem consists in inferring knowledge about the unobserved state of a dynamic system, which changes over time, using a sequence of noisy measurements. In a state-space approach to dynamic state estimation, the state vector \mathbf{x}_k of a system contains all relevant information necessary to describe the system. Bayesian estimation in this case is used to recursively estimate a time evolving posterior distribution or filtering distribution $f_k(\mathbf{x}_k|\mathbf{z}_{1:k})$, which describes the target state \mathbf{x}_k given all observations $\mathbf{z}_{1:k}$ up to time k . The rationale is that the posterior density f_k contains all information known about the state vector up to time k , in the sense that all modeling knowledge and measurement information available up to time k is captured in the posterior density. The exact solution to this problem can be constructed by specifying the Markovian probabilistic model of the state evolution, $\pi_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1})$, and the likelihood $g_k(\mathbf{z}_k|\mathbf{x}_k)$ which relates the noisy measurements to any state. The initial PDF, $f_0(\mathbf{x}_0|\mathbf{z}_{1:0}) \equiv f(\mathbf{x}_0)$, of the state vector, also known as the prior, is assumed available. The required probability density function $f_k(\mathbf{x}_k|\mathbf{z}_{1:k})$ can be recursively propagated by the Bayes recursion in two stages:

$$f_{k|k-1}(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int \pi_{k|k-1}(\mathbf{x}_k|\mathbf{x}) f_{k-1}(\mathbf{x}|\mathbf{z}_{1:k-1}) d\mathbf{x}, \quad (2.4)$$

$$f_k(\mathbf{x}_k|\mathbf{z}_{1:k}) = \frac{g_k(\mathbf{z}_k|\mathbf{x}_k) f_{k|k-1}(\mathbf{x}_k|\mathbf{z}_{1:k-1})}{\int g_k(\mathbf{z}_k|\mathbf{x}) f_{k|k-1}(\mathbf{x}|\mathbf{z}_{1:k-1}) d\mathbf{x}}, \quad (2.5)$$

where,

- \mathbf{x}_k is the state-vector of the target at time-step k and \mathbf{z}_k is the sensor observation collected at time-step k ,
- $f_k(\mathbf{x}_k|\mathbf{z}_{1:k})$ is the Bayes posterior distribution conditioned on the time-sequence $\mathbf{z}_{1:k}$ of measurements accumulated at time-step k ,

- $\pi_{k|k-1}(\mathbf{x}_k|\mathbf{x})$ is the Markov transition density that models the between-measurements motion of the target,
- $f_{k|k-1}(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ is the time-prediction of the prior $f_{k-1}(\mathbf{x}|\mathbf{z}_{1:k-1})$ to the time-step of the next measurement \mathbf{z}_k ,
- $g_k(\mathbf{z}_k|\mathbf{x}_k)$ is the sensor likelihood function.

The Bayesian discrete-time recursive nonlinear filtering equations (2.4)-(2.5) constitute the theoretical foundation for optimal single-sensor, single-target detection, tracking and identification.

Given the posterior density f_k at time k , an optimal estimate of the state vector can be obtained with respect to a prescribed criterion. A Bayes optimal estimate minimizes a certain objective called the Bayes risk. The most common estimates are the Expected A Posteriori (EAP) and the Maximum A Posteriori (MAP) estimates [23] given respectively by:

$$\begin{aligned}\hat{\mathbf{x}}_k^{EAP} &= \mathbb{E}[\mathbf{x}_k|\mathbf{z}_{1:k}] = \int \mathbf{x}_k f_k(\mathbf{x}_k|\mathbf{z}_{1:k}) d\mathbf{x}_k, \\ \hat{\mathbf{x}}_k^{MAP} &= \arg \sup_{\mathbf{x}_k} f_k(\mathbf{x}_k|\mathbf{z}_{1:k}).\end{aligned}$$

Due to the multiple integrations in the Bayes recursion (2.4)-(2.5), the full implementation of the Bayes filter is generally intractable in practice. While simple approaches such as state space discretization or numerical integration are possible, and are indeed the basis of the traditional approximate grid based and quadrature filters, these approaches are only useful in low dimensional problems since the complexity of such schemes grows exponentially with the dimension of the state space. In the following, an outline of an important special closed form solution as well as tractable approximations to the Bayes recursion is given.

2.1.2 The Kalman Filter

The Kalman Filter (KF) [82] is a closed form solution to the Bayes recursion in all linear Gaussian cases. Specifically, the Kalman filter assumes that the state dynamics and measurement models are linear transformations with additive Gaussian noise:

$$\mathbf{x}_k = F_{k-1}\mathbf{x}_{k-1} + \mathbf{v}_{k-1}, \quad (2.6)$$

$$\mathbf{z}_k = H_k\mathbf{x}_k + \mathbf{w}_k, \quad (2.7)$$

where F_{k-1} is an $n_x \times n_x$ transition matrix, H_k is an $n_z \times n_x$ observation matrix, and \mathbf{v}_{k-1} and \mathbf{w}_k are independent zero-mean Gaussian noise variables with covariance matrices Q_{k-1} and R_k of appropriate dimensions respectively. Thus, the transition density and measurement likelihood are:

$$\pi_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1}) = \mathcal{N}(\mathbf{x}_k; F_{k-1}\mathbf{x}_{k-1}, Q_{k-1}), \quad (2.8)$$

$$g_k(\mathbf{z}_k|\mathbf{x}_k) = \mathcal{N}(\mathbf{z}_k; H_k\mathbf{x}_k, R_k), \quad (2.9)$$

where $\mathcal{N}(\cdot; m, P)$ denotes a Gaussian density with mean and covariance m and P respectively.

Under these assumptions, the Kalman recursion proceeds as follows. If at time $k-1$, the posterior density is Gaussian of the form:

$$f_{k-1}(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) = \mathcal{N}(\mathbf{x}_{k-1}; m_{k-1}, P_{k-1}), \quad (2.10)$$

then the predicted density to time k is also Gaussian:

$$f_{k|k-1}(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \mathcal{N}(\mathbf{x}_k; m_{k|k-1}, P_{k|k-1}), \quad (2.11)$$

where,

$$m_{k|k-1} = F_{k-1}\mathbf{x}_{k-1}, \text{ the predicted state estimate,} \quad (2.12)$$

$$P_{k|k-1} = Q_{k-1} + F_{k-1} P_{k-1} F_{k-1}^T, \text{ the predicted estimate covariance matrix,} \quad (2.13)$$

and the updated density at time k (the posterior density at time k) is also Gaussian:

$$f_k(\mathbf{x}_k|\mathbf{z}_{1:k}) = \mathcal{N}(\mathbf{x}_k; m_k, P_k), \quad (2.14)$$

where,

$$m_k = m_{k|k-1} + K_k(\mathbf{z}_k - H_k\mathbf{x}_{k|k-1}), \text{ the updated state estimate,} \quad (2.15)$$

$$P_k = [I - K_k H_k] P_{k|k-1}, \text{ the updated estimate covariance matrix,} \quad (2.16)$$

with,

$$K_k = P_{k|k-1} H_k^T S_k^{-1}, \quad (2.17)$$

$$S_k = R_k + H_k P_{k|k-1} H_k^T. \quad (2.18)$$

The matrix K_k is referred to as the optimal Kalman gain, the residual $\mathbf{z}_k - H_k\mathbf{x}_{k|k-1}$ is referred to as the innovation and the matrix S_k is the innovation covariance.

The Kalman recursion establishes that for linear-Gaussian state dynamics and measurement models, if the posterior density at any time is Gaussian, then so are all subsequent predicted and updated densities. In the case where the posterior density is represented as a Gaussian Mixture (GM), the Gaussian sum filter analytically propagates the posterior density as a GM recursively in time, based on the basic result of the Kalman filter. However, the number of Gaussians required to represent the posterior density exactly increases exponentially with time.

When either the system state dynamics or the observation dynamics is nonlinear, the conditional probability density functions that provide the minimum mean-square estimate are no longer Gaussian. To overcome this problem, extensions of the linear Kalman filter have been proposed, such as the Extended and Unscented Kalman filters [13, 73]. The EKF [132] is a first order approximation to the Kalman filter based on local linearization. The key idea is to approximate the nonlinear functions f_k in (2.1) and g_k in (2.2) by using a first order Taylor expansion around the current state estimate. The Unscented Kalman Filter (UKF) [81] uses the sampling principles of the unscented transform to propagate the first and second moments of the predicted and updated densities. However, in the cases of strong nonlinear systems and non Gaussian noises, these two methods are deemed unreliable. Sequential Monte Carlo (SMC) methods should therefore be used as an efficient numerical approximation.

2.1.3 The Particle Filter

The Particle Filter (PF) [70] or Sequential Monte Carlo (SMC) recursion is based on random sample or point mass approximations to the densities in the Bayes filter, see standard references such as [9, 53, 70] for complete details or tutorial articles such as [8, 37] for introductory treatments.

Consider first the principle of perfect Monte Carlo sampling for an arbitrary probability density $\pi(\cdot)$. Let us assume available a set of N i.i.d. samples, called particles, drawn from π and denoted by $\{x^{(i)}\}_{i=1}^N$. Then, the Monte Carlo method approximates the density $\pi(\cdot)$ by the point mass representation:

$$\pi(x) \approx \frac{1}{N} \sum_{i=1}^N \delta_{x^{(i)}}(x), \quad (2.19)$$

in the sense that for any integrable function f , the Strong Law of Large Numbers (SLLN) for i.i.d. random variables ensures almost surely the asymptotic convergence:

$$I_{MC}(f) \triangleq \frac{1}{N} \sum_{i=1}^N f(x^{(i)}) \xrightarrow[N \rightarrow \infty]{\text{a.s.}} I(f) \triangleq \int f(x) \pi(x) dx. \quad (2.20)$$

Moreover, the variance of the above approximation is given by:

$$\text{Var}[I_{MC}(f)] = \frac{1}{N} \left(\int f^2(x) \pi(x) dx - I^2(f) \right). \quad (2.21)$$

The variance of the approximation error decreases at a rate of $O(1/N)$ regardless of the dimension of the integral and $f(x)$. Unfortunately, it is usually impossible to sample efficiently from the posterior distribution $\pi(\cdot)$, e.g. $\pi(\cdot)$ being multivariate, non standard and only known up to a proportionality constant, i.e $\pi(x) \propto p(x)$. In the latter case, an alternative solution consists of using the Bayesian Importance Sampling method. The basic idea is to generate samples from a known density $q(\cdot)$, referred to as the proposal density or importance function which is close to $\pi(\cdot)$, and then to weight these samples accordingly so as to obtain a weighted point mass approximation to $\pi(\cdot)$. Choose any density q which satisfies $\text{support}(\pi) \subseteq \text{support}(q)$ and generate N i.i.d. samples from q denoted by $\{x^{(i)}\}_{i=1}^N$. Then, the density $\pi(\cdot)$ can be approximated by the weighted point mass representation:

$$\pi(x) \approx \sum_{i=1}^N w(x^{(i)}) \delta_{x^{(i)}}(x), \quad (2.22)$$

where $\delta(\cdot)$ is the Dirac delta function and the weights:

$$\tilde{w}(x^{(i)}) = \frac{p(x^{(i)})}{q(x^{(i)})}, \quad w(x^{(i)}) = \frac{\tilde{w}(x^{(i)})}{\sum_{j=1}^N \tilde{w}(x^{(j)})}, \quad (2.23)$$

are known as the importance weights and the normalized importance weights respectively. With the importance sample approximation, the asymptotic convergence:

$$I_{NIS}(f) \triangleq \frac{1}{N} \sum_{i=1}^N w(x^{(i)}) f(x^{(i)}) \xrightarrow[N \rightarrow \infty]{\text{a.s.}} \int f(x) \pi(x) dx, \quad (2.24)$$

still holds almost surely.

Remark. Note that the normalized importance sampling estimate in (2.22), which divides the standard importance sampling estimate by the Monte Carlo average of the likelihood ratios, is biased for finite N . In the case $N = 1$, the estimate reduces to:

$$\frac{f(x^{(1)}) \tilde{w}(x^{(1)})}{\tilde{w}(x^{(1)})} = f(x^{(1)}),$$

with, $x^{(1)}$ sampled from $q(\cdot)$, the mean of the estimate is then $\mathbb{E}_q[f(x)]$ rather than $\mathbb{E}_\pi[f(x)]$.

The PF approach consists of using a sequential version of importance sampling (cf. C.1.3) to recursively construct point mass approximations to the posterior density. If at time $k - 1$, the posterior density $f_{k-1}(\cdot)$ is represented as a set of weighted particles $\{\mathbf{x}_{k-1}^{(i)}, w_{k-1}^{(i)}\}_{i=1}^N$, i.e.

$$f_{k-1}(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1}) \approx \sum_{i=1}^N w_{k-1}^{(i)} \delta_{\mathbf{x}_{k-1}^{(i)}}(\mathbf{x}_{k-1}), \quad (2.25)$$

then for a given proposal density $q_k(\cdot | \mathbf{x}_{k-1}^{(i)}, \mathbf{z}_k)$ satisfying $\text{support}(f_k) \subseteq \text{support}(q_k)$ the particle filter approximates the posterior density $f_k(\cdot)$ at time k as a new set of weighted particles $\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}_{i=1}^N$, i.e.

$$f_k(\mathbf{x}_k | \mathbf{z}_{1:k}) \approx \sum_{i=1}^N w_k^{(i)} \delta_{\mathbf{x}_k^{(i)}}(\mathbf{x}_k), \quad (2.26)$$

where,

$$\mathbf{x}_k^{(i)} \sim q_k(\cdot | \mathbf{x}_{k-1}^{(i)}, \mathbf{z}_k), \quad (2.27)$$

$$\tilde{w}_k^{(i)} = w_{k-1}^{(i)} \frac{g_k(\mathbf{z}_k | \mathbf{x}_k^{(i)}) \pi_{k|k-1}(\tilde{\mathbf{x}}_k^{(i)} | \mathbf{x}_{k-1}^{(i)})}{q_k(\mathbf{x}_k^{(i)} | \mathbf{x}_{k-1}^{(i)}, \mathbf{z}_k)}, \quad w_k^{(i)} = \frac{\tilde{w}_k^{(i)}}{\sum_{j=1}^N \tilde{w}_k^{(j)}}. \quad (2.28)$$

The basic Sampling Importance Sampling (SIS) algorithm is however subjected to the well known phenomenon of particle depletion or degeneracy where the variance of the importance weights increases over time. This phenomenon has the effect of degrading the performance of the filter and is usually mitigated with a resampling step. See [131] for a proof and [79] for more detail and an overview of convergence results. Note that there are many resampling schemes available, and that the choice of resampling scheme affects the computational load as well as the Monte Carlo approximation error. This resampling or selection step is what allows us to track moving target distributions efficiently by choosing the fittest particles.

Algorithm 1: SIR Particle Filter outline

input : $\{\mathbf{x}_{k-1}^{(i)}, w_{k-1}^{(i)}\}_{i=1}^N$ and a new measurement, \mathbf{z}_k

output: $\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}_{i=1}^N$

Sample initial particles $\{\mathbf{x}_0^{(i)}\}_{i=1}^N$ from $p(\mathbf{x}_0)$.

Set the weights $w_0^{(i)}$ to $\frac{1}{N}$.

1 - *Prediction:*

while $i \leftarrow 1$ **to** N **do**

 | Draw a sample: $\tilde{\mathbf{x}}_k^{(i)} \sim q_k(\cdot | \mathbf{x}_{k-1}^{(i)}, \mathbf{z}_k)$.

end

2 - *Update:*

while $i \leftarrow 1$ **to** N_p **do**

 | Compute weights : $\tilde{w}_k^{(i)} = w_{k-1}^{(i)} \frac{g(\mathbf{z}_k | \tilde{\mathbf{x}}_k^{(i)}) \pi(\tilde{\mathbf{x}}_k^{(i)} | \mathbf{x}_{k-1}^{(i)})}{q(\tilde{\mathbf{x}}_k^{(i)} | \mathbf{x}_{k-1}^{(i)}, \mathbf{z}_k)}$.

end

Normalize the weights : $\check{w}_k^{(i)} = \frac{\tilde{w}_k^{(i)}}{\sum_{j=1}^N \tilde{w}_k^{(j)}}$.

3 - *Compute the weight degeneracy:*

Effective Sample Size: $\hat{N}_{\text{eff}} = \left[\sum_{i=1}^N \left(\check{w}_k^{(i)} \right)^2 \right]^{-1}$.

4 - *Resample:*

if $\hat{N}_{\text{eff}} \leq N_{\text{thr}}$ **then**

 | Generate a new set of particles $\{\mathbf{x}_k^{(i)}\}_{i=1}^N$, so that for any i , $P(\mathbf{x}_k^{(i)} = \tilde{\mathbf{x}}_k^{(i)}) = \check{w}_k^{(i)}$.

 | Set the weights $w_k^{(i)}$ to $\frac{1}{N}$.

else

 | Copy $(\tilde{\mathbf{x}}_k^{(i)}, \check{w}_k^{(i)})$ to $(\mathbf{x}_k^{(i)}, w_k^{(i)})$, **for** $i = 1, \dots, N$.

end

An additional MCMC step of invariant distribution $f_k(\mathbf{x}_k|\mathbf{z}_{1:k})$ on each particle can then be used to rejuvenate the particles [64]. The basic idea is that if the particles are distributed according to the posterior distribution $f_k(\mathbf{x}_k|\mathbf{z}_{1:k})$, then applying a Markov chain transition kernel $K(\cdot|\mathbf{x}_k)$ with invariant distribution $f_k(\cdot|\mathbf{z}_{1:k})$ such that $\int K(\mathbf{x}_k^*|\mathbf{x}_k)f_k(\mathbf{x}_k|\mathbf{z}_{1:k}) = f_k(\mathbf{x}_k^*|\mathbf{z}_{1:k})$, still results in a set of particles distributed according to the posterior of interest. By applying a Markov transition kernel, the total variation of the current distribution with respect to the invariant distribution can only decrease.

In summary, in the SMC approach, the target distributions are approximated by a large number of random samples, termed particles, which are carried forward over time by using a combination of Sampling Importance Sampling (SIS) and resampling steps. These methods have become the tools of choice for sequential Bayesian inference. The SMC methodology is now well established and many theoretical convergence results are available [50]. Nevertheless, in practice, it is typically impossible to, a priori, determine the number of particles necessary to achieve a fixed precision for a given application. In such scenarios, users typically perform multiple runs for an increasing number of particles until stabilization of the Monte Carlo estimates is observed.

The performance/efficiency of the SMC methods largely depends on the design of the proposal distribution. Better techniques for the single target case can be implemented based on using the optimal importance function by minimizing the variance of the importance weights instead of sampling from the prior and reweighing by the likelihood [90].

Rao-Blackwellization [54, 55] techniques can be incorporated with the PF to improve performance for particular classes of state space models. The underlying idea is to partition the state vector into a linear Gaussian component and a non-linear non-Gaussian component. Then, the former is solved analytically using a Kalman filter and the latter with a PF so that the computational effort is appropriately focused.

Quasi Monte Carlo [104, 110] techniques have also been applied to the PF approach in which sampling is performed deterministically with the promise of a faster rate of convergence as well as better understood error propagation characteristics. Continuous approximations to the posterior density can furthermore be obtained with kernel smoothing techniques.

Sequential Monte Carlo (SMC) implementations with provable convergence and closed form solutions have opened the door to numerous novel extensions and applications.

2.2 Multi-target Tracking

The multi-target tracking problem is a well-known problem which involves the joint detection and estimation of an unknown and possibly time varying number of targets and of their individual states from a sequence of observations provided by one or more sensors. This problem poses significant technical challenges and has found application in many diverse disciplines including for example image processing and radar/sonar based tracking.

2.2.1 Common Multi-target Tracking Techniques

Traditional solutions have subsequently addressed the multi-target tracking problem by first estimating associations between detection plots and targets and then applying standard Bayesian filtering techniques. The following review aims to highlight the key ideas behind common algorithms: Global Nearest Neighbour (GNN), Joint Probabilistic Data Association (JPDA) and Multiple Hypothesis Tracking (MHT).

Global Nearest Neighbour

The Global Nearest Neighbour (GNN) filter [14, 20] is an extension of the Nearest Neighbour (NN) filter to the multiple target case, but the number of targets is assumed fixed and known. Given the previous estimate of the mean and covariance for each target state, the GNN filter first performs the standard Kalman prediction to obtain the predicted measurement and its covariance for each target state.

For the data association, the GNN filter searches for the unique joint association of measurements to targets that minimizes/maximizes a total cost, such as a total summed distance or likelihood, under the constraint that a measurement can be associated with at most one target. The assignment problem is well studied and many solutions have been proposed, for instance the Hungarian algorithm [88] and the Munkres assignment algorithm [33]. Faster methods include the Jonker-Volgenant relaxation [80] as well as the auction algorithm [17, 18].

For the update, the GNN filter simply assumes that the chosen joint association is correct, and performs the standard Kalman update for each target using these associated measurements directly.

Joint Probabilistic Data Association

The Joint Probabilistic Data Association (JPDA) filter [12] is an extension of the Probabilistic Data Association (PDA) filter to the case where there are multiple targets present but the number of targets is fixed and known. Given the previous estimate of the mean and covariance for each target state, the JPDA filter first performs the standard Kalman prediction to obtain the predicted measurement and its covariance for each target state.

For the data association, the JPDA filter uses joint association events and joint association probabilities in order to avoid conflicting measurement-to-track assignments in the presence of multiple targets. For the update, the JPDA filter hedges against incorrect measurement-to-target association by performing the Kalman update using an average of all measurements weighted according to their association probabilities. However, the complexity of the calculation for joint association probabilities grows exponentially with the number of targets and the number of measurements. The basic JPDA algorithm

is thus generally infeasible and so several approximation approaches have been proposed such as the deterministic suboptimal strategies in [123] and the MCMC based strategies in [109]. The JPDA approach has further been extended to non-linear non-Gaussian models with SMC methods in [136].

Multiple Hypothesis Tracking

The Multiple Hypothesis Tracking (MHT) filter [20, 47, 117] is a deferred decision approach to data association based multiple target tracking. The strategy of MHT is to mitigate association uncertainty at the current time step by searching over all previous time steps for all possible combinations of measurement-to-target associations that are likely to constitute target tracks or trajectories. An exhaustive association of all received measurements (past and present) to either a single track or as clutter is known as an hypothesis. At each time step, the MHT filter attempts to maintain a small set of hypotheses with high posterior probability. When a new set of measurements arrives, a new set of hypotheses is created from the existing hypotheses and their posterior probabilities are updated using Bayes rule.

Note that in the generation of new hypotheses, a measurement can be assigned either to clutter, an existing track or a completely new track. In this way, the MHT approach inherently handles initiation and termination of tracks, and hence accommodates tracking of an unknown and time-varying number of targets.

Example 2.2.1. To better understand the MHT algorithm, let us consider the following example:

1. First, suppose there is only one measurement.
This measurement can be:

- H_0 : false alarm (assigned to a "dummy" track 0),
- H_1 : true (assigned to a new track 1),

with respective probabilities P_0 and P_1 , proportional to the false alarm probability P_{FA} and to the new track probability P_{NT} .

2. Then, a new measurement arrives.

Under H_0 the new measurement can be:

- H_0 : false alarm with probability P_{00} ,
- H_2 : true (assigned to a new track 2) with probability P_{02} .

Under H_1 the new measurement can be:

- H_0 : false alarm with probability P_{10} ,
- H_1 : true (assigned to the same track 1) with probability P_{11} ,
- H_2 : true (assigned to a new track 2) with probability P_{12} .

Going ahead in this way, for further steps, the tree of hypotheses grows exponentially.

Thus, the basic idea in MHT is that hypotheses with high posterior probability are propagated, and at each time step the hypothesis with the highest posterior probability is sought. Having obtained the

best hypothesis, a standard Kalman filter can be used on the measurements in each track to estimate the states and trajectories of individual targets.

The combinatorial nature of MHT is its biggest limitation since the total number of possible hypotheses increases exponentially with time. In practice, traditional implementations of MHT usually require validation/gating of measurements as well as heuristic pruning/merging of hypotheses to reduce computational requirements [14].

Critical Analysis of Traditional Association-based Approaches

The majority of existing multi-target filtering techniques is based on variants of the JPDA or MHT filters outlined above. These techniques rely on the idea of hypothesizing associations between measurements and targets. Let ω denote an association hypothesis, that is the event that particular measurements originate from particular targets and/or clutter. Let $\Omega(\mathbf{Z})$ denote the space of all hypotheses defined from the measurement set \mathbf{Z} . Association-based approaches compute the posterior probability of a hypothesis ω given the measurement set \mathbf{Z} using Bayes rule as follows:

$$p(\omega|\mathbf{Z}) = \frac{p(\mathbf{Z}|\omega) p(\omega)}{p(\mathbf{Z})}. \quad (2.29)$$

Since each hypothesis ω is an element of $\Omega(\mathbf{Z})$, ω itself depends on \mathbf{Z} , and equation (2.29) should be written explicitly as:

$$p(\omega(\mathbf{Z})|\mathbf{Z}) = \frac{p(\mathbf{Z}|\omega(\mathbf{Z})) p(\omega(\mathbf{Z}))}{p(\mathbf{Z})}. \quad (2.30)$$

Closer examination reveals a number of conceptual issues with the application of Bayes rule in equation (2.30), which arise from the conditioning of the hypothesis on the measurement. The following issues have been raised in [137, p. 8]. Firstly, it is not obvious whether $p(\omega(\mathbf{Z}))$ is a valid prior, since it depends on future information. Secondly, $p(\mathbf{Z}|\omega(\mathbf{Z}))$, the probability density of the measurement set \mathbf{Z} conditioned on $\omega(\mathbf{Z})$, is not likely a valid likelihood, since the conditioning variable $\omega(\mathbf{Z})$ is conditioned on \mathbf{Z} . Thirdly, in Bayes rule it is necessary that the product of the likelihood and the prior, in this case $p(\mathbf{Z}|\omega(\mathbf{Z})) p(\omega(\mathbf{Z}))$, gives a valid joint probability density of the associated random variables $p(\omega(\mathbf{Z}), \mathbf{Z})$. Nonetheless, under this assumption, the marginal $p(\omega(\mathbf{Z}))$ should result from integrating out the observation variable \mathbf{Z} in the product density $p(\omega(\mathbf{Z}), \mathbf{Z})$, i.e.

$$\begin{aligned} p(\omega(\mathbf{Z})) &= \int p(\omega(\mathbf{Z}), \mathbf{Z}) d\mathbf{Z} \\ &= \int p(\mathbf{Z}|\omega(\mathbf{Z})) p(\omega(\mathbf{Z})) d\mathbf{Z}. \end{aligned}$$

However, the right side of the equation is independent of the measurement \mathbf{Z} since it is integrated out, which contradicts the inherent dependence of ω on \mathbf{Z} indicated on the left side of the equation. In light of the issues discussed above, it is not obvious whether traditional techniques are consistent with the Bayesian paradigm. Alternatively, the Random Finite Set (RFS) [95] approach to multi-target filtering is a mathematically consistent and association free formulation.

2.2.2 Random Finite Set Multi-target Tracking

Traditional Multi-Target Tracking (MTT) formulations, described in subsection 2.2.1, involve explicit associations between measurements and targets. Alternative formulations that avoid explicit associations between measurements and targets include Symmetric Measurement Equations (SMEs) [83] and Random Finite Sets (RFSs) [68]. In the RFS formulation, the collection of individual targets is treated as a set-valued state, and the collection of individual observations is treated as a set-valued observation. Modeling set-valued states and set-valued observations as RFSs allows the problem of dynamically estimating multiple targets in the presence of clutter and association uncertainty to be cast in a Bayesian filtering framework. The key to a rigorous formulation of multi-target estimation as Bayesian filtering problem is to conceptually view the target set as a single-meta target and the set of observations collected by the sensor as a single meta-observation. FISST provides a set of mathematical tools that allows direct application of Bayesian inferencing to multi-target problems.

Finite Set Statistics

Random set theory was first introduced by Mathéron [97], in the context of integral geometry, to describe polygons with an unknown number of vertices as well as unknown coordinates for the vertices and later exploited by Mahler [95] for multi-target and/or multi-sensor information fusion.

Let \mathcal{X} be a topological space e.g. $\mathcal{X} = \mathbb{R}^d$ for some integer $d \geq 1$. Further let $\mathcal{F}(\mathcal{X})$ and $\mathcal{C}(\mathcal{X})$ be the classes of finite and closed subsets of \mathcal{X} , respectively. In essence, a Random Finite Set (RFS) X on \mathcal{X} is defined as a measurable mapping from the sample space Ω to the "hyperspace" $\mathcal{F}(\mathcal{X})$ of finite subsets of \mathcal{X} :

$$X : \Omega \rightarrow \mathcal{F}(\mathcal{X}). \quad (2.31)$$

The realization of the RFS X can either have the form $X = \emptyset$ (empty set) or $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ for some finite integer $n \geq 1$.

Like any random variable, an RFS is completely described by its probability distribution. The probability distribution of the RFS X on \mathcal{X} is the probability measure \mathbb{P} on $\mathcal{F}(\mathcal{X})$ defined by:

$$P(\mathcal{T}) = \mathbb{P}(\{X \in \mathcal{T}\}), \quad (2.32)$$

for any Borel subset \mathcal{T} of $\mathcal{F}(\mathcal{X})$, where $\{X \in \mathcal{T}\}$ denotes the measurable subset $\{\omega \in \Omega : X(\omega) \in \mathcal{T}\}$ of Ω . Apart from the probability distribution, the belief-mass function $\beta_X(\cdot)$ of an RFS X is defined by:

$$\begin{aligned} \beta_X(S) &= \mathbb{P}(X \subseteq S) = \sum_{n=0}^{\infty} \mathbb{P}(X \subseteq S, |X| = n) \\ &= \sum_{n=0}^{\infty} \int f_X(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n; n) d\mathbf{x}_1 d\mathbf{x}_2 \dots d\mathbf{x}_n, \end{aligned} \quad (2.33)$$

for all closed subset $S \subseteq \mathcal{X}$, where $\beta_X(\mathcal{X}) = 1$, and $f_X(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n; n)$ is the joint PDF over the hybrid state space $\mathcal{X} \times \mathbb{Z}^+$ of the elements $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n \in \mathcal{X}$ and of the cardinality $n \in \mathbb{Z}^+$. The belief-mass function can be interpreted as the counterpart of the cumulative distribution function $\mathbb{P}(\mathbf{x} \leq x)$ of a random variable \mathbf{x} in \mathbb{R} . For the modeling of multi-target systems, the belief-mass function can

be more convenient than the probability distribution, since the former deals with closed subsets of \mathcal{X} whereas the latter deals with subsets of $\mathcal{F}(\mathcal{X})$. However, the belief-mass function is not a measure and hence the standard measure theoretic notion of a density is not directly applicable to the belief functional, that is their Radon-Nikodým derivatives (or densities) are not defined. To circumvent this difficulty, the theory of Finite-Set Statistics (FISST) provides an alternative notion of "density" for belief-mass functions through set integrals and set derivatives [68].

Set Derivative Let $\mathcal{C}(\mathcal{X})$ denote the collection of closed subsets of \mathcal{X} . The set derivative of a function $F : \mathcal{C}(\mathcal{X}) \rightarrow [0, \infty)$ at a point $\mathbf{x} \in \mathcal{X}$ is a mapping $\frac{\delta F}{\delta \mathbf{x}} : \mathcal{C}(\mathcal{X}) \rightarrow [0, \infty)$ defined as:

$$(d_K F)_{\mathbf{x}}(S) \triangleq \frac{\delta F}{\delta \mathbf{x}}(S) \triangleq \lim_{\lambda_K(\Delta_{\mathbf{x}}) \rightarrow 0} \frac{F(S \cup \Delta_{\mathbf{x}}) - F(S)}{\lambda_K(\Delta_{\mathbf{x}})}, \quad (2.34)$$

where $\lambda_K(\Delta_{\mathbf{x}})$ denotes the volume of a neighborhood $\Delta_{\mathbf{x}}$ of \mathbf{x} in units of K . Note that $\lambda_K = K \lambda$, with λ the unitless Lebesgue measure. This is a simplified version of the complete definition given in [68]. Furthermore the set derivative at a generic finite set $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ is also defined by the recursion:

$$(d_K F)_{\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}}(T) \triangleq (d_K (d_K F)_{\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{n-1}\}})_{\mathbf{x}_n}(T), \quad (2.35)$$

where $(d_K F)_{\emptyset} = F$ by convention. A list of basic differentiation rules exists cf. Appendix A.5 [95, pp. 386-394], thus avoiding to resort to the formal limit-based definitions.

Set Integral Let f be a function defined by $f(X) = (d_K F)_X(\emptyset)$. Then the set integral of f over a (closed) subset $S \subseteq \mathcal{X}$ is defined as follows:

$$\begin{aligned} \int_S f(X) \delta_K X &\triangleq \sum_{i=0}^{\infty} \frac{1}{i!} \int_{S^i} f(\{\mathbf{x}_1, \dots, \mathbf{x}_i\}) \lambda_K^i(d\mathbf{x}_1 \dots d\mathbf{x}_i) \\ &= f(\emptyset) + \int_S f(\{\mathbf{x}_1\}) \lambda_K(d\mathbf{x}_1) + \frac{1}{2} \int_{S \times S} f(\{\mathbf{x}_1, \mathbf{x}_2\}) \lambda_K^2(d\mathbf{x}_1 d\mathbf{x}_2) + \dots \end{aligned} \quad (2.36)$$

The set integral and set derivative are related by the following generalized fundamental theorem of multi-target calculus,

$$f(X) = (d_K F)_X(\emptyset) \text{ if and only if } F(S) = \int_S f(X) \delta_K X. \quad (2.37)$$

The set integral and set derivative are inverse operations.

RFS Multi-target Bayes Filtering

Suppose that at time k there are n_k target states $\mathbf{x}_{k,1}, \dots, \mathbf{x}_{k,n_k}$, each taking values in a state space $\mathcal{X} \subseteq \mathbb{R}^{n_x}$, and m_k measurements (detections) $\mathbf{z}_{k,1}, \dots, \mathbf{z}_{k,m_k}$, each taking values in the observation space $\mathcal{Z} \subseteq \mathbb{R}^{n_z}$. At each time step some targets may disappear, others may survive and transit into a new state, and new targets may appear. Due to the imperfections in the detector, some of the surviving and newborn objects may not be detected, whereas the observation set \mathbf{Z}_k may include false detections

(or clutter). The evolution of the targets and the origin of measurements are unknown. Analogous to single target system, where uncertainty is characterized by modeling the state and measurement by random vectors, uncertainty in a multi-target system, i.e. in both multi-target state and multi-target measurement, is modeled by RFSs.

The multi-target estimation can be formulated in a Bayesian framework by modeling as RFSs:

- the hidden set of states $\mathbf{X} \subset \mathcal{X}$ distributed according to a multi-target prior $f(\mathbf{X}_0)$:

$$\mathbf{X}_k = \{\mathbf{x}_{k,1}, \dots, \mathbf{x}_{k,n_k}\} \in \mathcal{F}(\mathcal{X}), \quad (2.38)$$

- the set of observations $\mathbf{Z} \subset \mathcal{Z}$ described by the multi-target likelihood function $\vartheta(\mathbf{Z}|\mathbf{X})$ which encapsulates observation noise, detection uncertainty, clutter and data association uncertainty:

$$\mathbf{Z}_k = \{\mathbf{z}_{k,1}, \dots, \mathbf{z}_{k,m_k}\} \in \mathcal{F}(\mathcal{Z}), \quad (2.39)$$

where $\mathcal{F}(\mathcal{E})$ denotes the collection of all finite subsets of the space \mathcal{E} .

The objective of the recursive multi-target Bayesian estimator is to determine at each time step k the posterior probability density of the multi-target state $f_k(\mathbf{X}_k|\mathbf{Z}_{1:k})$, where $\mathbf{Z}_{1:k} = (\mathbf{Z}_1, \dots, \mathbf{Z}_k)$ denotes the accumulated observation sets up to time k . The uncertainty in a multi-target system is characterized by modeling the multi-target states and the multi-target measurements as random finite sets, Ξ_k and Σ_k , respectively. Individual target motion in a multi-target problem is often modeled by a transition density on the single-target state space \mathcal{X} while the measurement process is modeled as a likelihood on the single-target observation space \mathcal{Z} . FISST converts the construction of multi-target densities from multi-target models into computing set-derivatives of belief-mass functions.

For any closed subsets $S \subseteq \mathcal{X}$ and $T \subseteq \mathcal{Z}$, let

$$\beta_k(S|\mathbf{Z}_{1:k}) \triangleq \mathbb{P}(\Xi_k \subseteq S|\mathbf{Z}_{1:k})$$

denote the (posterior) belief-mass function of the RFS Ξ_k given all the observation sets up to time k ,

$$\beta_{k|k-1}(S|\mathbf{X}_{k-1}) \triangleq \mathbb{P}(\Xi_k \subseteq S|\mathbf{X}_{k-1})$$

denote the belief-mass function of the RFS Ξ_k , and

$$\beta_{k|k-1}(T|\mathbf{X}_k) \triangleq \mathbb{P}(\Sigma_k \subseteq T|\mathbf{X}_k)$$

denote the belief-mass function of the RFS Σ_k . Then, the FISST multi-target posterior density $f_k(\cdot|\mathbf{Z}_{1:k})$, the FISST multi-target transition density $\Pi_{k|k-1}(\cdot|\mathbf{X}_{k-1})$ and the FISST multi-target likelihood $\vartheta_k(\mathbf{Z}_k|\mathbf{X}_k)$ are the set derivatives of $\beta_k(\cdot|\mathbf{Z}_{1:k})$, $\beta_{k|k-1}(\cdot|\mathbf{X}_{k-1})$ and $\beta_{k|k-1}(\cdot|\mathbf{X}_k)$, respectively.

The multi-target posterior can be computed sequentially via the prediction and 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 k has been received. Then the FISST multi-target Bayes filter, proposed in [68], is given by the recursion:

$$f_{k|k-1}(\mathbf{X}_k|\mathbf{Z}_{1:k-1}) = \int \Pi_{k|k-1}(\mathbf{X}_k|\mathbf{X}_{k-1}) f_{k-1}(\mathbf{X}_{k-1}|\mathbf{Z}_{1:k-1}) \delta \mathbf{X}_{k-1}, \quad (2.40)$$

$$f_k(\mathbf{X}_k|\mathbf{Z}_{1:k}) = \frac{\vartheta_k(\mathbf{Z}_k|\mathbf{X}_k) f_{k|k-1}(\mathbf{X}_k|\mathbf{Z}_{1:k-1})}{\int \vartheta_k(\mathbf{Z}_k|\mathbf{X}) f_{k|k-1}(\mathbf{X}|\mathbf{Z}_{1:k-1}) \delta \mathbf{X}}, \quad (2.41)$$

where $\Pi_{k|k-1}(\mathbf{X}_k|\mathbf{X}_{k-1})$ is the multi-target transition density and $\vartheta_k(\mathbf{Z}_k|\mathbf{X}_k)$ is the multi-target likelihood.

Note that this recursion, theoretically optimal approach to multi-sensor multi-target detection, tracking, and identification, is a non-trivial generalization of the recursive Bayes filter, since the transition density needs to consider the uncertainty in target number, which can change over time due to targets entering and leaving the state space, and the multi-target likelihood needs to consider detection uncertainty and false alarms. It should be stressed that the integrals in the recursion (2.40-2.41) are nonstandard set integrals and the functions involved have units.

The multi-target Bayes recursion (2.40-2.41) can alternatively be stated in terms of probability generating functionals (p.g.fl.s) as follows. Let $G_{k|k-1}[\cdot|\mathbf{Z}_{1:k-1}]$ denote the p.g.fl. of the multi-target predicted density $f_{k|k-1}(\cdot|\mathbf{Z}_{1:k-1})$ and let $G_{\pi,k|k-1}[\cdot|\mathbf{X}_{k-1}]$ denote the p.g.fl. of the multi-target transition density $\Pi_{k|k-1}(\cdot|\mathbf{X}_{k-1})$ at time k . Then, the multi-target Bayes prediction (2.40) can be stated in terms of p.g.fl.s as:

$$G_{k|k-1}[h|\mathbf{Z}_{1:k-1}] = \int G_{\pi,k|k-1}[h|\mathbf{X}_{k-1}] f_{k-1}(\mathbf{X}_{k-1}|\mathbf{Z}_{1:k-1}) \delta\mathbf{X}_{k-1}. \quad (2.42)$$

Let $G_k[\cdot|\mathbf{Z}_{1:k}]$ denote the p.g.fl. of the multi-target posterior density $f_k(\cdot|\mathbf{Z}_{1:k})$ and let $G_{\vartheta,k}[\cdot|\mathbf{X}_k]$ denote the p.g.fl. of the multi-target likelihood $\vartheta_k(\cdot|\mathbf{X}_k)$ at time k . Then, the multi-target Bayes update (2.41) can be stated in terms of p.g.fl.s as:

$$G_k[h|\mathbf{Z}_{1:k}] = \frac{U_k^{(|\mathbf{Z}_k|)}[0, h; \delta_{\mathbf{Z}_k}|\mathbf{Z}_{1:k}]}{U_k^{(|\mathbf{Z}_k|)}[0, 1; \delta_{\mathbf{Z}_k}|\mathbf{Z}_{1:k}]}, \quad (2.43)$$

$$U_k[l, h|\mathbf{Z}_{1:k}] = \int h^{\mathbf{X}} G_{\vartheta,k}[l|\mathbf{X}] f_{k|k-1}(\mathbf{X}|\mathbf{Z}_{1:k-1}) \delta\mathbf{X}, \quad (2.44)$$

where $U_k[\cdot, \cdot|\mathbf{Z}_{1:k}]$ denotes the joint p.g.fl. of the unnormalized multi-target Bayes update step and $U_k^{(n)}[0, h; \delta_{\mathbf{Z}_k}|\mathbf{Z}_{1:k}]$ is the n th functional derivative of $U_k[\cdot, h|\mathbf{Z}_{1:k}]$ at $l = 0$ in the directions δ_z for $z \in \mathbf{Z}_k$.

Remark. The role of a probability generating functional (p.g.fl.) in an RFS is analogous to the role of a probability generating function in a discrete random variable. The p.g.fl. G of an RFS \mathbf{X} on \mathcal{X} is defined by:

$$G[h] \equiv \mathbb{E}[h^{\mathbf{X}}], \quad (2.45)$$

for any real-valued function h on \mathcal{X} such that $0 \leq h(x) \leq 1$, where

$$h^{\mathbf{X}} \equiv \prod_{x \in \mathbf{X}} h(x), \text{ with, } h^{\emptyset} = 1 \text{ by convention.} \quad (2.46)$$

The joint p.g.fl. G of two RFSs \mathbf{X} on \mathcal{X} and \mathbf{Z} on \mathcal{Z} is defined as:

$$G[l, h] \equiv \mathbb{E}[l^{\mathbf{X}} h^{\mathbf{Z}}], \quad (2.47)$$

for any real valued functions l and h such that $0 \leq l(x) \leq 1$ and $0 \leq h(x) \leq 1$. The joint p.g.fl. of a finite number of RFSs can be defined in a similar way.

2.2.3 Multi-target State Estimation

This subsection discusses multi-target state estimation, i.e. the process of extracting target state estimates from the multi-target posterior density. First, the common pitfall of applying naive extensions to traditional Bayes-optimal state estimators, the Expected A Posteriori (EAP) and Maximum A Posteriori (MAP) estimators, is highlighted. Then, an overview of principled approaches for performing multi-target state estimation is provided. In particular two Bayes-optimal multi-target state estimators, the Marginal Multi-target (MaM) and Joint Multi-target (JoM) estimators, are described and compared.

Failures of Traditional EAP and MAP Estimators

The purpose of this paragraph is to demonstrate, using an example, the following:

- The multi-target MAP is mathematically well defined only when single-target state space has no units of measurement.
- A multi-target posterior expectation is meaningless even when it happens to be mathematically well defined.

Example 2.2.2. (Naïve multi-target EAP/MAP: 0 or 1 Target) Consider a situation in which there is at most one target in the one dimensional-interval $[0 \ 2]$ with unit of distances given in meters. Assume that, according to current evidence, there is a 50/50 chance that no target exists. However, if a target is present, it is a single target that is equally likely to be anywhere in the interval $[0 \ 2]$. In this case the multi-target posterior distribution is

$$f(X) = \begin{cases} 0.5, & \text{if } X = \emptyset; \\ 0.25 \text{ m}^{-1}, & \text{if } X = \{x\}, 0 \leq x \leq 2; \\ 0, & \text{otherwise.} \end{cases}$$

First consider the naïve EAP estimate:

$$\hat{X}^{EAP} = \int X f(X) \delta X = \emptyset f(\emptyset) + \int_0^2 x f(x) dx = \frac{1}{2}(\emptyset + 2m).$$

We see first-hand the problem, the naïve EAP estimate is not mathematically well defined, since the addition and subtraction of finite sets is not meaningfully defined. Now consider the naïve MAP estimate:

$$\hat{X}^{MAP} = \arg \sup_X f(X) = \emptyset.$$

From this we conclude that the scene contains no targets. Now change units of measurement from meters to kilometers. Then the multi-target posterior is

$$f(X) = \begin{cases} 0.5, & \text{if } X = \emptyset; \\ 250 \text{ km}^{-1}, & \text{if } X = \{x\}, 0 \leq x \leq 0.002; \\ 0, & \text{otherwise.} \end{cases}$$

In this case, the naïve MAP estimate becomes:

$$\hat{X}^{MAP} = \arg \sup_X f(X) = \{x\} \quad \forall 0 \leq x \leq 0.002.$$

We now conclude that a target is present, even though nothing has changed except the units of measurement. The paradox arises because the naïve MAP estimator is defined only if units-mismatch difficulties do not arise.¹

The apparent inconsistencies with applying the conventional EAP and MAP estimators as shown above can be ascribed to mismatches in the units of each cardinality of the multi-target density. That is, comparing sets of differing cardinalities is meaningless, at least without a proper theoretical basis for attempting such comparisons. Resolution of such question is one of the advances made possible by FISST.

FISST Optimal Multi-target Estimators

The generalization of the concept of Bayes estimator to randomly varying finite sets [68, pp. 189-190] gives us new solutions to perform multi-target state estimation. In particular this allows the definition of global "analogs" of the MAP estimator, as follows:

Marginal Multi-target (MaM) estimator The global MAP estimator of first kind (GMAP-I), better known as MaM estimator, is defined by the following procedure:

1. First, compute the cardinality distribution of the posterior distribution $f_k(\cdot|\mathbf{Z}_{1:k})$:

$$\rho_k(n|\mathbf{Z}_{1:k}) = \frac{1}{n!} \int f_k(\{\mathbf{x}_{k,1}, \dots, \mathbf{x}_{k,n}\}) d\mathbf{x}_{k,1} \dots d\mathbf{x}_{k,n}; \quad n = 0, 1, 2, \dots \quad (2.48)$$

2. Then, determine that value $n = \hat{n}$ which maximizes the cardinality distribution:

$$\hat{n} \triangleq \arg \sup_n \rho_k(n|\mathbf{Z}_{1:k}). \quad (2.49)$$

3. Finally, maximize the posterior subject to the constraint that $|X| = \hat{n}$. The individual target states are estimated using a MAP estimator or an EAP estimator on the posterior density under the constraint $n = \hat{n}$, i.e. $X_{\hat{n}} = \{X : |X| = \hat{n}\}$

$$\hat{X}_k^{MaM} = \begin{cases} \arg \sup_{X_{\hat{n}}} f_k(X_{\hat{n}}|\mathbf{Z}_{1:k}), & \text{MAP estimate;} \\ \int X_{\hat{n}} f_k(X_{\hat{n}}|\mathbf{Z}_{1:k}) dX_{\hat{n}}, & \text{EAP estimate.} \end{cases} \quad (2.50)$$

It has been shown that the MaM estimator is Bayes optimal, cf. proof [68, pp. 192-194], however convergence results are not currently known.

¹This example originally appeared in [95, pp. 494-496].

Joint Multi-target (JoM) estimator The global MAP estimator of second kind (GMAP-II), better known as JoM estimator is evaluated using a two step procedure:

1. First, for each $n \geq 0$, determine the MAP estimate:

$$\hat{X}^{(n)} = \arg \sup_{X: |X|=n} f_k(X|\mathbf{Z}_{1:k}). \quad (2.51)$$

2. Second, set

$$\hat{X}_k^{JoM,c} = \hat{X}^{(\hat{n})}, \text{ with } \hat{n} = \arg \sup_n f_k(\hat{X}^{(n)}|\mathbf{Z}_{1:k}) \frac{c^n}{n!}, \quad (2.52)$$

where the constant c determines the desired accuracy for the state estimate as well as the rate of convergence of the estimator, in the sense that smaller c provides better accuracy with slower convergence. The value of c should be approximately equal to the accuracy to which the state is to be estimated, as long as $f_k(\hat{X}^{(n)}|\mathbf{Z}_{1:k}) c^n \leq 1$.

It has been shown that the JoM estimator is Bayes optimal, cf. proof [68, pp. 192-194] and is statistically consistent [95].

Example 2.2.3. (MaM/JoM: 0 or 1 Target) Consider a situation in which there is at most one target, suppose that a multi-target posterior has the form

$$f(X) = \begin{cases} 1 - q, & \text{if } X = \emptyset; \\ q \mathcal{N}(x, \sigma^2), & \text{if } X = \{x\}, 0 \leq x \leq 2; \\ 0, & \text{otherwise;} \end{cases}$$

where σ is very small, this results in a narrow-range and high-peak distribution. In this case, the MaM estimator will decide in favor of the presence of a target if:

$$1 - q = f(\emptyset) < f(1) = q \text{ that is, if } q > \frac{1}{2}.$$

That is, the MaM estimator ignores critical information contained in the variance σ^2 . The JoM estimator, on the other hand, will decide in favor of the presence of a target if, given the small-magnitude estimation constant c ,

$$1 - q = f(\emptyset) < c \sup_x f(\{x\}) = \frac{c q}{\sqrt{2\pi}\sigma} \text{ that is, if } \sigma \left(\frac{1}{q} - 1 \right) < \frac{c}{\sqrt{2\pi}}.$$

That is, the JoM estimator makes a more nuanced decision by balancing the information contained in q against the relevant information contained in σ . For example, let $c = 0.1$ and $q = 0.2$. Then the target should be exceedingly well-localized ($\sigma < 0.004$) if a target-in-presence decision is to be supported. However, if $q = 0.999$, an accurate localization of the target is not required ($\sigma < 39.9$).

2.2.4 Multi-target Performance Evaluation

This subsection reviews existing metrics for performance evaluation of multi-target estimation and discusses their strengths and limitations. A satisfactory multi-target miss-distance needs to capture the "difference" between two sets of vectors, namely the reference multi-target state $\mathbf{X}_k = \{\mathbf{x}_{k,1}, \dots, \mathbf{x}_{k,n}\}$ and the estimated multi-target state $\hat{\mathbf{X}}_k = \{\hat{\mathbf{x}}_{k,1}, \dots, \hat{\mathbf{x}}_{k,\hat{n}}\}$, in a mathematically rigorous manner. Thus, a miss-distance, to be consistent for multi-target performance evaluation, should comply the fundamental requirement:

- be a metric on $\mathcal{F}(\mathcal{X})$, the space of finite subsets of the single-target state space \mathcal{X} .

For completeness, we recall the definition of a metric. Let \mathcal{S} be an arbitrary non-empty set. A function $d : \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R}_+ = [0, \infty)$ is called a metric if it satisfies the following three axioms:

- *identity*: $d(x, y) = 0$, if and only if $x = y$,
- *symmetry*: $d(x, y) = d(y, x)$, for all $x, y \in \mathcal{S}$,
- *triangular inequality*: $d(x, y) \leq d(x, z) + d(z, y)$, for all $x, y, z \in \mathcal{S}$.

In the context of multi-target miss distances, a closed and bounded observation window $\mathcal{W} \subset \mathcal{X}$ is usually fixed, and \mathcal{S} is set to be $\mathcal{F}(\mathcal{S})$, the space of finite subsets of \mathcal{W} . In the following, d denotes the underlying metric used on \mathcal{W} , while appropriate indices are attached for the various metrics considered on $\mathcal{F}(\mathcal{W})$ (e.g. d_H , d_p or $d_{p,c}$).

Remark. For target tracking, the underlying metric d between tracks $x^\ell \equiv (\ell, x)$ and $y^{\ell'} \equiv (\ell', y)$ includes the labeling error [120] and is defined as:

$$d(x^\ell, y^{\ell'}) = (d(x, y)^p + d(\ell, \ell')^p)^{\frac{1}{p}}, \quad (2.53)$$

where,

- $1 < p < \infty$ is the distance order parameter,
- $d(x, y)$ is the spatial distance typically adopted as the p -norm:

$$d(x, y) = \|x - y\|_p,$$

- $d(\ell, \ell')$ is the labeling error, a metric on \mathbb{N} , adopted as:

$$d(\ell, \ell') = \alpha \bar{\delta}_{\ell, \ell'},$$

with $\bar{\delta}_{\ell, \ell'}$ is the complement of the Kronecker delta, that is $\bar{\delta}_{\ell, \ell'} = 0$ if $\ell = \ell'$ and 1 otherwise. The parameter $\alpha \in [0, c]$ determines the penalty assigned to the labeling error $d(\ell, \ell')$ interpreted relative to the spatial distance $d(x, y)$. The case $\alpha = 0$ assigns no penalty, and $\alpha = c$ assigns the maximum penalty.

Hausdorff Metric

For finite non-empty subsets X and Y of \mathcal{W} , define:

$$d_H(X, Y) \triangleq \max \left(\max_{x \in X} \min_{y \in Y} d(x, y), \max_{y \in Y} \min_{x \in X} d(x, y) \right). \quad (2.54)$$

The function d_H is called the Hausdorff metric. The Hausdorff metric is traditionally used as a measure of dissimilarity between binary images for which it is well suited from a theoretical and practical points of view. However, the Hausdorff metric is not particularly suited to multi-target performance evaluation for the following reasons:

- it is very insensitive to differing cardinalities of the reference and estimated sets;
- it penalizes outliers heavily, to the extent that the outlier can sometimes completely determine the value of the miss-distance;
- it can not be reasonably defined if one of the sets is empty.

From a mathematical point of view, the Hausdorff metric generates the standard topology considered on the set of closed subsets of \mathcal{W} , which is used to define random sets. In the context of finite set statistics (FISST) this topology is usually called the Mathéron topology.

Optimal MAss Transfer (OMAT) Metric

In [77], The Optimal MAss Transfer (OMAT) metric was introduced to overcome some of the problems associated with the Hausdorff metric.

The OMAT metric is defined, for finite non-empty subsets $X = \{x_1, \dots, x_m\}$ and $Y = \{y_1, \dots, y_n\}$ of \mathcal{W} , as follows:

$$d_p(X, Y) \triangleq \min_C \left(\sum_{i=1}^m \sum_{j=1}^n C_{i,j} d(x_i, y_j)^p \right)^{1/p}, \quad \text{if } 1 \leq p < \infty \text{ and} \quad (2.55)$$

$$d_\infty(X, Y) = \min_C \max_{1 \leq i \leq m, 1 \leq j \leq n} \tilde{C}_{i,j} d(x_i, y_j), \quad (2.56)$$

where the minima are taken over all $m \times n$ transportation matrices $C = (C_{i,j})$ and where $\tilde{C}_{i,j} = 1$ if $C_{i,j} \neq 0$ and $\tilde{C}_{i,j} = 0$ otherwise. An $m \times n$ matrix C is a transportation matrix if all of its entries are non-negative and if:

$$\sum_{j=1}^n C_{i,j} = \frac{1}{m} \text{ for } 1 \leq i \leq m, \quad \sum_{i=1}^m C_{i,j} = \frac{1}{n} \text{ for } 1 \leq j \leq n. \quad (2.57)$$

The OMAT metric of order p , d_p , is also known as Wasserstein metric since its definition yields the p -th order Wasserstein metric between the empirical distributions of the point patterns X and Y .

The OMAT metric partially alleviates the undesirable cardinality behavior of the Hausdorff metric and the penalty applied to outliers is better proportioned by the introduction of the parameter p . However, the OMAT metric still entails issues outlined as follows:

- it does not consistently detect cardinality errors, the OMAT distance actually depends on how well balanced the estimated points are distributed amongst the true points;
- it is not intuitively appealing, often results in distances that are difficult to interpret, and geometry dependent;
- it is not defined if one of the sets is empty.

Optimal Sub-pattern Assignment (OSPA) Metric

In [125], the Optimal Sub-Pattern Assignment (OSPA) metric was proposed. This metric is still based on a Wasserstein construction, but eliminates most of the problems faced by the OMAT metric.

Let $d^{(c)}(x, y) \triangleq \min(c, d(x, y))$ denote the distance between $x, y \in \mathcal{W}$ cut of at $c > 0$, and let Π_k denote the set of permutations on $\{1, 2, \dots, k\}$ for any positive integer k . Let $X = \{x_1, \dots, x_m\}$ and $Y = \{y_1, \dots, y_n\}$ be two finite sets, with elements taking values on \mathcal{W} . Then, for $1 \leq p < \infty$ and $c > 0$, the OSPA metric of order p with cut-off c is defined as:

$$d_{p,c}(X, Y) \triangleq \begin{cases} 0, & m = n = 0; \\ \left(\frac{1}{n} \left(\min_{\pi \in \Pi_n} \sum_{i=1}^m d^{(c)}(x_i, y_{\pi(i)})^p + c^p(n-m) \right) \right)^{\frac{1}{p}}, & \text{if } m \leq n; \\ d_{p,c}(Y, X), & \text{if } m > n; \end{cases} \quad (2.58)$$

moreover,

$$d_{\infty,c}(X, Y) \triangleq \begin{cases} 0, & \text{if } m = n = 0; \\ \min_{\pi \in \Pi_n} \max_{1 \leq i \leq n} d_c(x_i, y_{\pi(i)}), & \text{if } m = n; \\ c, & \text{if } m \neq n. \end{cases} \quad (2.59)$$

Note that, if one of the set is empty, the OSPA metric becomes insensitive to the cardinality of the non-empty set, e.g. if $X = \emptyset$, $m = 0$ then, $d_{p,c}(\emptyset, Y) = c$.

In the context of multi-object performance evaluation, the OSPA distance is interpreted as a p -th order per-target error. This error is comprised of two components each separately accounting for localization and cardinality errors, i.e.

$$d_{p,c}(X, Y) = (c_{p,c}^{\text{loc}}(X, Y) + c_{p,c}^{\text{card}}(X, Y))^{\frac{1}{p}},$$

with, if $m \leq n$

$$\begin{aligned} c_{p,c}^{\text{loc}}(X, Y) &= \frac{1}{n} \min_{\pi \in \Pi_n} \sum_{i=1}^m d_c(x_i, y_{\pi(i)})^p, \\ c_{p,c}^{\text{card}}(X, Y) &= \frac{c^p(n-m)}{n}. \end{aligned}$$

Remark. If $m \leq n$, the definition is the same with $c_{p,c}^{\text{loc}}(X, Y) \triangleq c_{p,c}^{\text{loc}}(Y, X)$ and $c_{p,c}^{\text{card}}(X, Y) \triangleq c_{p,c}^{\text{card}}(Y, X)$.

These components can thus be interpreted as contributions due to localization only (within the optimal subpattern assignment) and cardinality only (penalized at maximal distance). The order parameter p determines the sensitivity of the metric $d_{p,c}$ in penalizing outlier estimates. Larger values of p tends to

blunt the sensitivity. The cut-off parameter c determines the relative weighting of the penalties assigned to cardinality and localization errors. Smaller values of c tend to emphasize localization errors and make the metric mostly insensitive to cardinality errors, whereas larger values of c predominantly indicate cardinality errors and ignore localization errors.

From a practical point of view, the OSPA distance between two finite sets (point patterns) X and Y , for $p < \infty$ and assuming $m = |X| \leq n = |Y|$, is obtained by going through the following three steps:

1. Select, Y^* , the m point subpattern (subset consisting of m elements) of Y that is closest to X in terms of the p -th order *OMAT* metric (paragraph 2.2.4). Let $y_j \in Y^* \rightarrow x_{\pi_j} \in X$ denote an optimal point assignment between X and Y , obtained using the Hungarian algorithm [88].
2. For each point y_j of Y , let α_j be the cut-off value c if there is no point assigned to it or else let α_j be the minimum between c and the distance to its assigned point in X , i.e.

$$\alpha_j = \begin{cases} c, & \text{if } y_j \notin Y^*; \\ \min(c, d(x_{\pi_j}, y_j)), & \text{if } y_j \in Y^*. \end{cases}$$

3. Compute the OSPA metric as the p -th order average of $\alpha_1, \alpha_2, \dots, \alpha_n$ i.e.

$$d_{p,c}(X, Y) = \left(\frac{1}{n} \sum_{j=1}^n \alpha_j^p \right)^{\frac{1}{p}}.$$

Remark. The Hungarian method is known to have an asymptotic complexity that is cubic in the dimension of the distance matrix, so that we obtain $O(\max(m, n)^3)$ for computing the OSPA metric. For further details see [125].

In summary, the OSPA metric has two parameters, p and c which have intuitive interpretations as "outlier sensitivity" and "cardinality penalty" respectively. These parameters provide the user with a certain degree of flexibility to adjust the metric to suit the objectives of particular situations. For example, larger values of c cater for applications in which the number of targets in the scene is crucial, whereas smaller values of c address the requirement for well-localized targets. In chapter 4, the OSPA metric for tracks [120] is retained to assess and benchmark the performance of the designed tracking algorithms.

RFS Approach to Multi-target Track-Before-Detect

In many applications involving video data, the estimation is often performed on data that have been preprocessed into point measurements. For example, in radar tracking, radar videos are converted to detection plots. Compressing the information on the video into a finite set of points is efficient in terms of memory as well as computational requirements. When combined with point measurement based approaches, multi-target tracking can be very effective for a wide range of applications. However, this approach may be undesirable for applications with low SNR, since the information loss incurred during the compression can be significant, and in such cases it can clearly be advantageous to make use of all information contained in the video. For the specific Track-Before-Detect (TBD) surveillance application, the multi-target estimation problem can be formulated in a Bayesian framework by modeling the hidden set of states \mathbf{X}_k as a finite set. Then computing the posterior distribution of the RFS of states given the raw measurement \mathbf{z}_k requires solving a set integral only for the Chapman Kolmogorov equation (2.40). The multi-target likelihood is used directly to update each target.

The chapter is organized as follows. Section 3.1 introduces the concept of TBD and emphasizes the potential payoffs of this approach. Section 3.2 adopts the Bayesian RFS approach to describe and address the multi-target filtering and estimation problem in the TBD context. In particular, a specification of the underlying RFS models for the multi-target dynamics and multi-target observation is given.

3.1 Track Before Detect versus Classical Radar Tracking

3.1.1 Basic Radar Principles

Radar systems are widely used in military contexts for surveillance and weapon control and in civil applications to ensure safe and secure travel of naval vessels and aircrafts. A radar operates by radiating electromagnetic energy and detecting the echo returned from reflecting objects (targets). The nature of the echo signal provides information about the targets. The range, or distance to the target, is determined from the time it takes for the radiated energy to travel to the target and back. The shift in frequency of the received echo signal due to the Doppler effect caused by a moving target allows to distinguish moving targets (such as aircrafts or ships) from stationary targets (such as land and sea clutter) and to extract the range rate, or radial speed of the targets. The bearing and elevation angles of the target are found with a fan-beam¹ or multi-beam antenna, to sense the angle of arrival of the

¹A fan-beam antenna is a directional antenna producing a main beam having a narrow beamwidth in one dimension and a wider beamwidth in the other dimension.

echo signal.

The basic principle behind radar is described in Figure 3.1 [107,130].

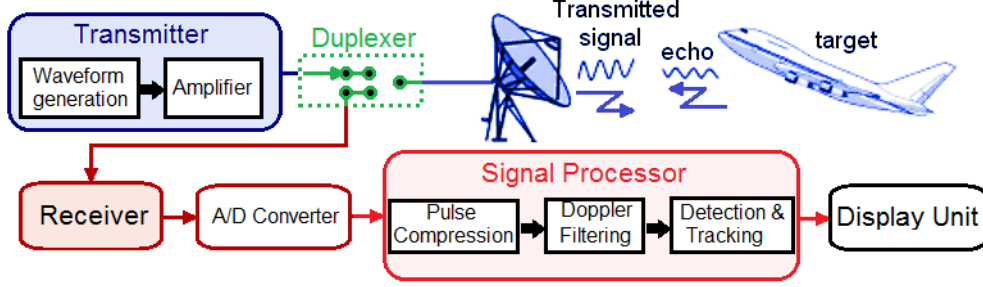


Figure 3.1: Radar principle

1. The Radio-Frequency [Hz] (RF) signal, usually a repetitive train of short pulses, is generated by the transmitter and radiated into space by the antenna.
2. The signal is reflected off targets and returned as echo.
3. The returned echo signal is collected by the radar antenna and, the receiver filters out the interference, increases the SNR and amplifies the signal.

Remark. A frequency and timing control synchronizes the transmitter and the receiver; the duplexer switches between both paths, i.e. transmission and reception.

4. The signal & data processor further processes the received video signal to provide detection information and details about each detected target, that is, its location in range and angle, its size and shape. The rate of change of target location can also be measured from the change in range and angle with time, from which the track can be established.
5. The display shows only the information that is relevant to the user.

3.1.2 Radar Signal Processing

Let us now focus on the signal processor unit. The analog output of the receiver is down-converted to baseband via quadrature mixing. The in-phase and quadrature signals are passed through a matched filter and converted to digital words by an analog-to-digital (A/D) converter. Following the A/D is typically a delay-line clutter canceler and a Doppler filter bank for main-beam clutter rejection, and coherent integration.

In the classical setup there is a pre-processing chain before the tracking chain that consists of a detection stage, a cluster stage and an extraction stage. This pre-processing allows to solve the detection problem, which consists of deciding whether the radar reflections are target originated or not, by comparing the output of the receiver, i.e. the raw measurements, to some threshold level. Traditionally, target detection involves thresholding at three stages: signal strength thresholding on hit level, thresholding after binary integration of plot level, and thresholding after binary integration on track level.

(1) Detection - thresholding on hit level

In the detection stage the total scanned area is divided into cells. The measured reflected energy in each cell is compared to a reference level or threshold. A hit detection is declared in a cell if the received signal is above the threshold. In particular, target detection in a single scan of a surveillance radar system is performed by binary integration of target detections (so-called hits) in consecutive coherent processing intervals, also called bursts. Each scan of a surveillance radar consists of a large number of bursts, each covering a bearing interval. Since the radar beamwidth in bearing typically is several times the size of the bearing interval covered by a burst, target signal will be present in a number of consecutive bursts of a scan. In case of a pulse-Doppler surveillance radar, the received signal in a range-Doppler frame is obtained from each burst after appropriate range sampling and application of a doppler filter bank. At a certain range and Doppler speed in this frame a hit detection is declared if the received signal is above the threshold providing a predetermined constant false alarm probability.

(2) Cluster/Extraction - thresholding on plot level

The cluster stage consists to group the neighboring cells, in which a hit detection has been declared, probably associated to the same target. In particular, a target detection in a scan, a plot, is declared if for the same range and Doppler speed there are n_h hits in n_b consecutive bursts. The parameters n_h , n_b and the threshold on hit level are chosen such that a certain false alarm probability on plot level is guaranteed.

Plots, which typically consist of range, range rate (Doppler), bearing and elevation measurements, are then extracted by determining the center of the energy in each cluster.

(3) Tracking - thresholding on track level

Next, each target plot is used to initialize a track filter. The track filter prediction for subsequent scans is used to identify possible target plots that can be associated to the track. After n_s scans, confirmed target track detection is declared if there are n_p plots out of the possible n_s associated to the track. The parameters n_s , n_p and the threshold on plot level are chosen such that a certain false alarm probability on track level is respected.

This approach allows a considerable reduction in the amount of data to be processed during the tracking step, but it obviously results in a loss of information. That is of no consequence at high SNR. By cons, at low SNR, a compromise must be struck. A high detection threshold avoids false alarms while decreasing the detection probability. A lower detection threshold leads to a significant number of false alarms and increases the complexity of the plot-to-track association problem.

In the classical method a weak target does not always lead to a detection at each scan. When there is no detection no plots of the weak target can be constructed and therefore this target will be declared lost. To overcome this problem, the Track-before-Detect, so-called TBD [28, 31, 124] approach proposes to base the tracking on the raw measurements instead of plots and thus avoid an explicit data association step. Thus, the conventional three-stage track detection scheme is replaced by a single stage involving only signal strength thresholding on track level. By delaying the thresholding and thereby allowing the

target signal to build up, a large improvement in detection probability can be achieved over traditional multi-stage track detection schemes at equal false alarm probability.

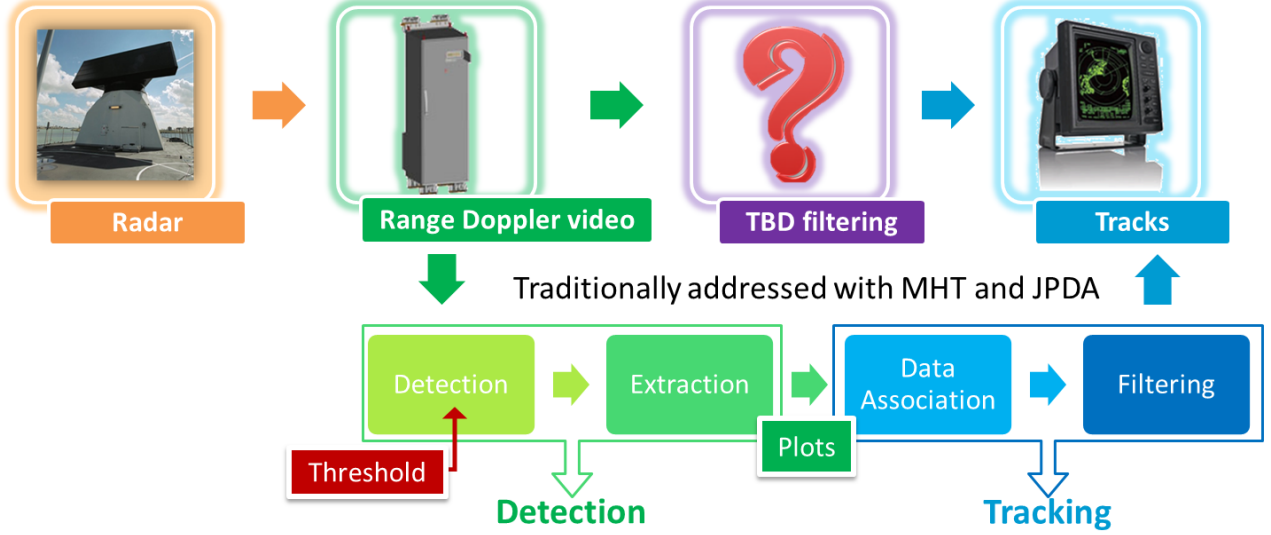


Figure 3.2: Data and signal processing

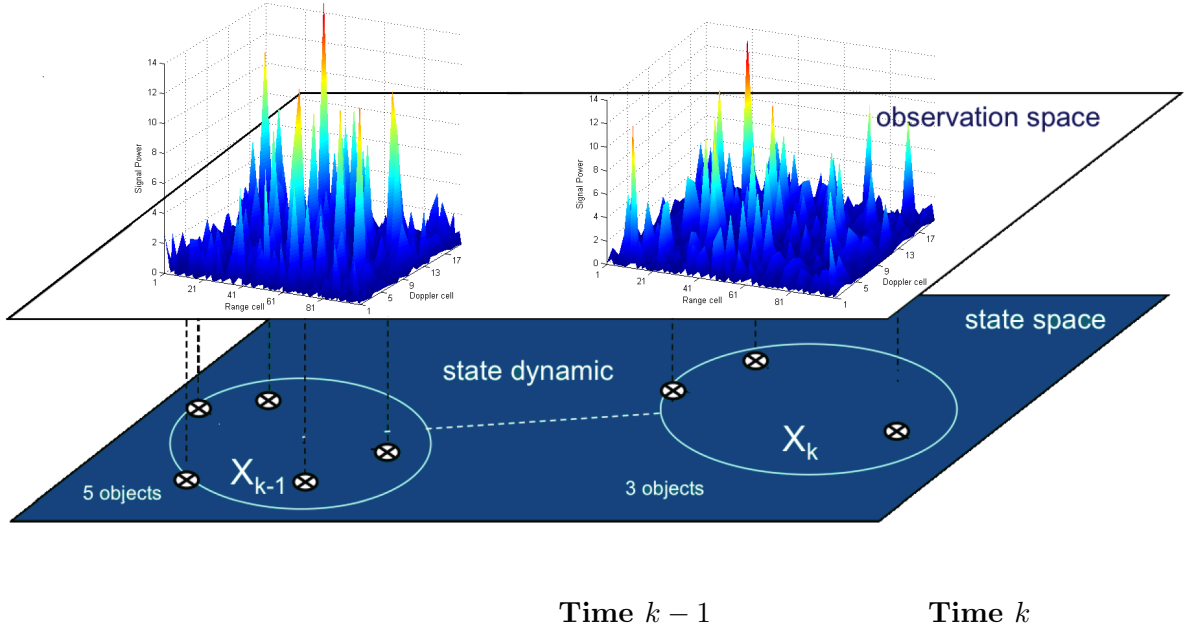
Examples of previously developed techniques for TBD include direct Maximum Likelihood solved by dynamic programming [15] or by Hough transform [39]. While effective, these batch methods are unsatisfactory for the reasons set out below. These methods require discretization of the state space and incur intensive computational burden. For most problems, much of this computation is wasted effort propagating the tails of the distribution where there may be of negligible probability. On the other hand, a recursive Bayesian filter incorporates the complete raw measurements as a highly non-linear measurement, which can be readily implemented by using Particle Filter (PF) techniques [9]. In this study this latter approach is adopted. Particle based TBD was first presented in [124] and [29] in independent parallel research. These methods have since been extended from single target filter to multi-target although significant research work still needs to be completed [27, 28, 48, 60].

In the next section 3.2, the multi-target tracking and estimation problem in TBD context is formally formulated in the Bayesian RFS framework. This naturally leads to the chapter 4 where novel Particle based TBD are introduced.

3.2 Track-Before-Detect Tracking

Let us consider the multi-target filtering in TBD context, and derive from the RFS formalism an optimal Bayes filter that subsumes a number of popular multi-target Bayesian filtering approaches.

3.2.1 Multi-target Bayes Filter (scheme)



$$\begin{aligned} \text{Multi-target state:} \quad & \mathbf{X}_{k-1} = \{\mathbf{x}_{k-1,1}, \dots, \mathbf{x}_{k-1,5}\} \quad \mathbf{X}_k = \{\mathbf{x}_{k,1}, \dots, \mathbf{x}_{k,3}\} \\ \text{Multi-target measurement:} \quad & \mathbf{z}_{k-1} = [\mathbf{z}_{k-1}^1, \dots, \mathbf{z}_{k-1}^{2000}] \quad \mathbf{z}_k = [\mathbf{z}_k^1, \dots, \mathbf{z}_k^{2000}] \end{aligned}$$

Suppose that at time k there are n_k target states $\{\mathbf{x}_{k,1}, \dots, \mathbf{x}_{k,n_k}\}$, each taking values in a state space $\mathcal{X} \subseteq \mathbb{R}^{n_x}$, and a single measurement $\mathbf{z}_k = [\mathbf{z}_k^1, \dots, \mathbf{z}_k^m]$ comprising an array of m cell values, where each $\mathbf{z}_k^c \in \mathbb{C}$ is the complex (I/Q) signal in the radar cell indexed by c . In the RFS approach, the finite set of target states and the vector of intensities measured per cell at time k , are referred to as the multi-target state and the multi-target observation, respectively:

$$\mathbf{X}_k = \{\mathbf{x}_{k,1}, \dots, \mathbf{x}_{k,n_k}\} \in \mathcal{F}(\mathcal{X}), \quad (3.1)$$

$$\mathbf{z}_k = [\mathbf{z}_k^1, \dots, \mathbf{z}_k^m] \in \mathcal{Z}, \quad (3.2)$$

where $\mathcal{F}(\mathcal{X})$ is the multi-target state space, i.e. the finite subsets of \mathcal{X} .

At each time step, some targets may disappear, some targets may survive and evolve to a new state, and new targets may appear. Thus, the behavior of the targets present is not known. Such uncertainty or randomness is naturally characterized by using RFS theory to construct stochastic model for the dynamics of the multi-target state as follows.

3.2.2 Multi-target Dynamic Model

The following is a specification of the standard RFS multi-target dynamical model considered in this dissertation. More general dynamical models are available within the RFS framework, however, these are not falling into the scope of this thesis.

Given a multi-target state \mathbf{X}_{k-1} at time $k-1$, each $\mathbf{x}_{k-1} \in \mathbf{X}_{k-1}$ either continues to exist at time k with probability $p_{S,k}(\mathbf{x}_{k-1})$ and moves to a new state \mathbf{x}_k with transition probability density $\pi_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1})$, or dies with probability $1 - p_{S,k}(\mathbf{x}_{k-1})$. Thus, given a target with state $\mathbf{x}_{k-1} \in \mathbf{X}_{k-1}$ at time $k-1$, its behavior at time k is modeled by:

- a BERNOULLI RFS $S_{k|k-1}(\mathbf{x}_{k-1})$ with $r_S = p_{S,k}(\mathbf{x}_{k-1})$ and $s_S(\cdot) = \pi_{k|k-1}(\cdot|\mathbf{x}_{k-1})$.

This transition is commonly known in point process theory as a Markov shift.

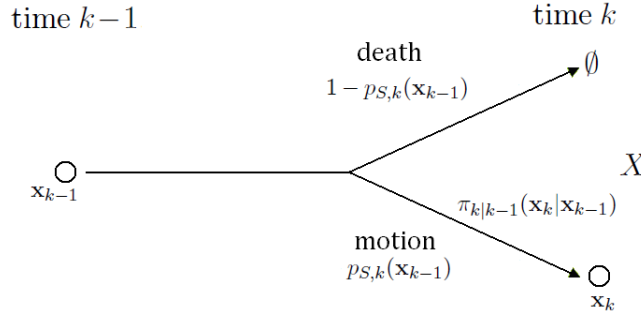


Figure 3.3: Bernoulli survival RFS $S(\mathbf{x}')$

The survival or death of all existing targets from time $k-1$ to k is hence modeled by the RFS:

$$T_{k|k-1}(\mathbf{X}_{k-1}) = \bigcup_{\mathbf{x}_{k-1} \in \mathbf{X}_{k-1}} S_{k|k-1}(\mathbf{x}_{k-1}). \quad (3.3)$$

Remark. A multi-Bernoulli RFS \mathbf{X} on \mathcal{X} is a union of a fixed number of independent Bernoulli RFSs $\mathbf{X}^{(i)}$ with existence probability $r^{(i)} \in [0, 1]$ and probability density $s^{(i)}$ defined on \mathcal{X} for $i = 1, \dots, M_S$, i.e.

$$\mathbf{X} = \bigcup_{i=1}^{M_S} \mathbf{X}^{(i)}. \quad (3.4)$$

A multi-Bernoulli RFS is thus completely described by the corresponding multi-Bernoulli parameter set $\{(r^{(i)}, s^{(i)})\}_{i=1}^{M_S}$ equivalently $\{q^{(i)} := \frac{r^{(i)} s^{(i)}}{1-r^{(i)}}\}_{i=1}^{M_S}$. Its probability density is:

$$\Pi(\{x_1, \dots, x_n\}) = \begin{cases} \prod_{i=1}^{M_S} (1 - r^{(i)}), & \text{if } n = 0; \\ K^n \prod_{i=1}^{M_S} (1 - r^{(i)}) \sum_{1 \leq i_1 \neq \dots \neq i_n \leq M_S} \prod_{j=1}^n \frac{r^{(i_j)} s^{(i_j)}(x_j)}{1 - r^{(i_j)}}, & \text{if } n \leq M_S; \\ 0, & \text{if } n > M_S. \end{cases} \quad (3.5)$$

Alternatively, (3.5) can be written more compactly in terms of the parameters $\{q^{(i)}\}_{i=1}^{M_S}$ as follows:

$$\Pi(\{X\}) := K^{|X|} \left(\frac{1}{1 + \langle q^{(\cdot)}, 1 \rangle} \right)^{\{1, \dots, M_S\}} \sum_{\tau \in \mathcal{T}(X, \{1, \dots, M_S\})} q_{\tau}^X, \quad (3.6)$$

where $q_{\tau}(x) = q^{(\tau(x))}(x)$ and $\mathcal{T}(X, Y)$ denotes the set of all one-to-one functions taking a finite set X to a finite set Y , with the convention that the summation over $\mathcal{T}(X, Y)$ is zero when $|X| > |Y|$ and unity when $X = \emptyset$. Note that $1 - r^{(i)} = \frac{1}{1 + \langle q^{(\cdot)}, 1 \rangle}$. Moreover, $q^X \equiv \prod_{x \in X} q(x)$, as defined in Appendix B.

Assuming that, conditional on \mathbf{X}_{k-1} , the Bernoulli RFSs in (3.3) are mutually independent, then the RFS of surviving targets $T_{k|k-1}(\mathbf{X}_{k-1})$ is a multi-Bernoulli RFS with parameter set:

$$\{(p_{S,k}(\mathbf{x}_{k-1}), \pi_{k|k-1}(\cdot | \mathbf{x}_{k-1})) : \mathbf{x}_{k-1} \in \mathbf{X}_{k-1}\}.$$

It follows from (3.6) that the probability transition density $\Pi_{T,k|k-1}(\mathbf{X}_k | \mathbf{X}_{k-1})$ of the RFS of surviving targets is:

$$\Pi_{T,k|k-1}(\mathbf{X}_k | \mathbf{X}_{k-1}) := K_S^{|\mathbf{X}_k|} (1 - p_{S,k}(\mathbf{x}_{k-1}))^{|\mathbf{X}_{k-1}|} \sum_{\tau \in \mathcal{T}(\mathbf{X}_k, \mathbf{X}_{k-1})} q_{S,k,\tau}^{\mathbf{X}_k}, \quad (3.7)$$

where K_S is the unit of volume on \mathcal{X} and

$$q_{S,k,\tau}(x) = \frac{p_{S,k}(\tau(x)) \pi_{k|k-1}(x | \tau(x))}{1 - p_{S,k}(\tau(x))}. \quad (3.8)$$

In addition, births at time k can be described by:

1. a POISSON RFS Ψ_k with intensity function $\psi_k(\cdot) = \mu_b b_{k|k-1}(\cdot)$. In this case the probability transition density $\Pi_{\Gamma,k}(\mathbf{X}_k)$ of the RFS of newborn targets is:

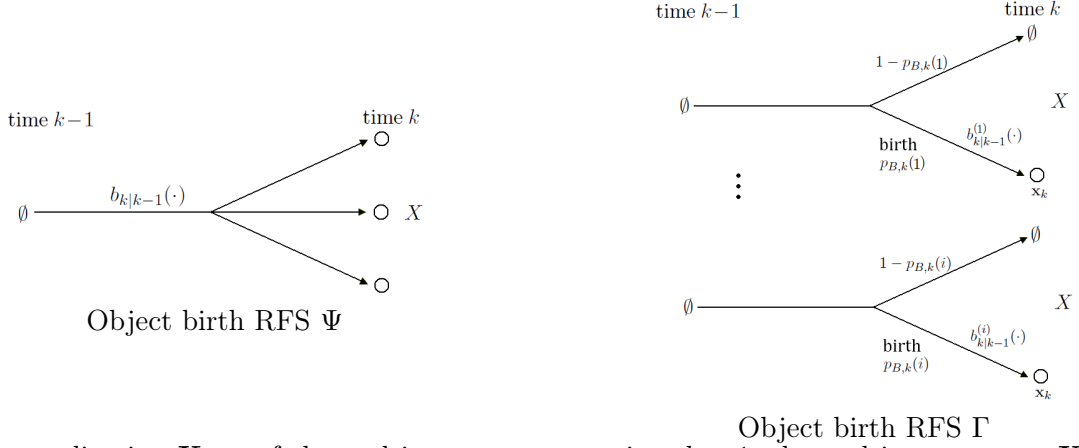
$$\Pi_{\Gamma,k}(\mathbf{X}_k) := e^{-\langle \psi_k, 1 \rangle} K_S^{|\mathbf{X}_k|} \psi_k^{\mathbf{X}_k}. \quad (3.9)$$

2. a MULTI-BERNOULLI RFS Γ_k with the Multi-Bernoulli parameters $\{r_{\Gamma}^{(i)}, s_{\Gamma}^{(i)}(\cdot)\}_{i=1}^{M_{\Gamma}}$, with $r_{\Gamma}^{(i)} = p_{B,k}(i)$ and $s_{\Gamma}^{(i)} = b_{k|k-1}^{(i)}(\cdot)$ or equivalently $\{q_{\Gamma,k}^{(i)}\}_{i=1}^{M_{\Gamma}}$. In this case the probability transition density $\Pi_{\Gamma,k}(\mathbf{X}_k)$ of the RFS of newborn targets is:

$$\Pi_{\Gamma,k}(\mathbf{X}_k) := K_S^{|\mathbf{X}_k|} \left(\frac{1}{1 - \langle q_{\Gamma,k}^{(\cdot)}, 1 \rangle} \right)^{\{1, \dots, M_{\Gamma}\}} \sum_{v \in \mathcal{T}(\mathbf{X}_k, \{1, \dots, M_{\Gamma}\})} q_{\Gamma,k,v}^{\mathbf{X}_k}, \quad (3.10)$$

where,

$$q_{\Gamma,k,v}(x) = q_{\Gamma,k}^{(v(x))} = \frac{p_{B,k}(v(x)) b_{k|k-1}^{(v(x))}(x)}{1 - p_{B,k}(v(x))}. \quad (3.11)$$



Thus, given a realization \mathbf{X}_{k-1} of the multi-target state at time $k-1$, the multi-target state \mathbf{X}_k at time k can be modeled by the union:

$$\mathbf{X}_k \equiv \Xi_{k|k-1} = \underbrace{T_{k|k-1}(\mathbf{X}_{k-1})}_{\text{Multi-Bernoulli Survival}} \cup \underbrace{\Gamma_k}_{\text{Multi-Bernoulli Birth}}. \quad (3.12)$$

Markov Transition Density

Assuming that the RFSs $T_{k|k-1}(\mathbf{X}_{k-1})$ and Γ_k are mutually independent, the RFS multi-target transition equation (3.12) can alternatively be stated in the form of the multi-target transition density $\Pi_{k|k-1}(\cdot|\cdot)$. Given the probability density $\Pi_{T,k|k-1}(\cdot|\cdot)$ of the RFS $T_{k|k-1}(\mathbf{X}_{k-1})$ and the probability density $\Pi_{\Gamma,k}(\cdot)$ of the spontaneous birth RFS Γ_k , the multi-target transition density $\Pi_{k|k-1}(\mathbf{X}_k|\mathbf{X}_{k-1})$ is given by:

$$\Pi_{k|k-1}(\mathbf{X}_k|\mathbf{X}_{k-1}) = \sum_{W \subseteq \mathbf{X}_k} \Pi_{T,k|k-1}(W|\mathbf{X}_{k-1}) \Pi_{\Gamma,k}(\mathbf{X}_k - W), \quad (3.13)$$

where the difference operation denotes the set difference. Both (3.12) and (3.13) describe the time evolution of the multi-target state and incorporate the underlying models of target motion, birth and death. The statistical behavior of the RFS Ξ_k is characterized by the conditional probability density $\Pi_{k|k-1}(\mathbf{X}_k|\mathbf{X}_{k-1})$ in an analogous fashion to the Markov transition density for random vector.

Explicit formulae for the multi-target transition density are given as follows. In the special case where Γ_k is a Poisson RFS with intensity function ψ_k , i.e. $\Pi_{\Gamma,k}(X) := e^{-\langle \psi_k, 1 \rangle} K_S^{[X]} \psi_k^X$, the transition density simplifies to:

$$\Pi_{k|k-1}(\mathbf{X}_k|\mathbf{X}_{k-1}) = (1 - p_{S,k}(\mathbf{x}_{k-1}))^{\mathbf{x}_{k-1}} \frac{(K_S \psi_k)^{|\mathbf{X}_k|}}{e^{-\langle \psi_k, 1 \rangle}} \sum_{W \subseteq \mathbf{X}_k} \sum_{\tau \in \mathcal{T}(W, \mathbf{X}_{k-1})} \left(\frac{q_{S,k,\tau}}{\psi_k} \right)^W. \quad (3.14)$$

If Γ_k is a multi-Bernoulli RFS with parameter set $\{r_{\Gamma}^{(i)}, s_{\Gamma}^{(i)}(\cdot)\}_{i=1}^{M_{\Gamma}}$ or equivalently $\{q_{\Gamma,k}^{(i)}\}_{i=1}^{M_{\Gamma}}$, then the transition density is given by:

$$\Pi_{k|k-1}(\mathbf{X}_k|\mathbf{X}_{k-1}) = \frac{(1 - p_{S,k}(\mathbf{x}_{k-1}))^{\mathbf{x}_{k-1}} K_S^{|\mathbf{X}_k|}}{\left(1 - \langle q_{\Gamma,k}^{(\cdot)}, 1 \rangle\right)^{\{1, \dots, M_{\Gamma}\}}} \sum_{W \subseteq \mathbf{X}_k} \sum_{\tau \in \mathcal{T}(W, \mathbf{X}_{k-1})} \sum_{v \in \mathcal{T}(W, \{1, \dots, M_{\Gamma}\})} q_{\Gamma,k,v}^{\mathbf{x}_k} \left(\frac{q_{S,k,\tau}}{q_{\Gamma,k,v}} \right)^W. \quad (3.15)$$

3.2.3 Multi-target Observation Model

The following is a specification of the RFS multi-target observation model in the TBD context. The TBD approach defines a model for the raw measurement in terms of a multi-target state hypothesis. In this study, the target return signals measured by the radar are assumed to fluctuate according to the Swerling return amplitude fluctuation models [134]. These models are widely used as a means of estimating radar detection performance [99, 130]. Here the Swerling fluctuation models are incorporated into the likelihood function of the filter, to account for the target return fluctuations.

A given target $\mathbf{x}_{k,j} \in \mathbf{X}_k$, at time k , illuminates a set of cells denoted by $C(\mathbf{x}_{k,j})$. For example, $C(\mathbf{x}_{k,j})$ could be the set of cells whose centers fall within a certain distance from the position of the target. The complex (I/Q) signal in each cell of the observation image at time k , \mathbf{z}_k^c , is assumed to be:

$$\mathbf{z}_k^c = \begin{cases} \sum_{j:c \in C(\mathbf{x}_{k,j})} h^c(\mathbf{x}_{k,j}) + \mathbf{w}_k^c, & \mathcal{H}_1: \text{if at least one target illuminates the cell } c; \\ \mathbf{w}_k^c, & \mathcal{H}_0: \text{if there is no target;} \end{cases} \quad (3.16)$$

where $h^c(\mathbf{x}_{k,j})$, the contribution of a target with state $\mathbf{x}_{k,j}$ to the signal in cell c , depends on the so-called sensor point spread function (in this case pulse form), the target location and the complex target echo. \mathbf{w}_k^c is the measurement noise in cell c with known statistics. The noise is assumed to be independent from cell to cell and identically distributed according to the PDF φ^c .

Likelihood

Let the measurement likelihood in cell c in the presence of at least one target be denoted by $\phi^c(\mathbf{z}_k^c | \mathbf{X}_k)$, and the likelihood under the hypothesis of no targets be $\varphi^c(\mathbf{z}_k^c)$. Since conditioned on the multi-target state \mathbf{X}_k , the values of the cells are independently distributed, the multi-target likelihood of the whole observation \mathbf{z}_k is simply the product over the cells given by:

$$\vartheta_k(\mathbf{z}_k | \mathbf{X}_k) = \left(\prod_{c \in C(\mathbf{x}): \mathbf{x} \in \mathbf{X}_k} \phi^c(\mathbf{z}_k^c | \mathbf{X}_k) \right) \left(\prod_{\substack{c \notin \bigcup \\ \mathbf{x} \in \mathbf{X}_k} C(x)} \varphi^c(\mathbf{z}_k^c) \right). \quad (3.17)$$

Hence,

$$\vartheta_k(\mathbf{z}_k | \mathbf{X}_k) = f(\mathbf{z}_k) \prod_{\mathbf{x} \in \mathbf{X}_k} g_{\mathbf{z}_k}(\mathbf{x}), \quad (3.18)$$

where,

$$f(\mathbf{z}_k) = \prod_{c=1}^m \varphi^c(\mathbf{z}_k^c), \quad (3.19)$$

$$g_{\mathbf{z}_k}(\mathbf{x}) = \prod_{c \in C(\mathbf{x}): \mathbf{x} \in \mathbf{X}_k} \mathfrak{L}(\mathbf{z}_k^c | \mathbf{X}_k), \quad (3.20)$$

with $\mathfrak{L}(\mathbf{z}_k^c | \mathbf{X}_k)$, the likelihood ratio for cell c given by:

$$\mathfrak{L}(\mathbf{z}_k^c | \mathbf{X}_k) = \frac{\phi^c(\mathbf{z}_k^c | \mathbf{X}_k)}{\varphi^c(\mathbf{z}_k^c)}. \quad (3.21)$$

In many cases it is more convenient to deal with the likelihood ratio of the data, rather than the measurement PDF.

Example 3.2.1. Consider a radar system, located at the Cartesian origin, that collects, at discrete instants k , a measurement \mathbf{z}_k . Assuming that the measurement $\mathbf{z}_k = [\mathbf{z}_k^1 \dots \mathbf{z}_k^m]^T$ consists of the power signal for all m range-Doppler-bearing cells. The power measurements per (range-Doppler-bearing) cell is given by:

$$\mathbf{z}_k^c = |\mathbf{z}_{A,k}^c|^2 \quad k \in \mathbb{N},$$

where $\mathbf{z}_{A,k}^c$ represents the complex (I/Q) signal in cell c , i.e.

$$\mathbf{z}_{A,k}^c = \sum_{\mathbf{x} \in \mathbf{X}_k: c \in C(\mathbf{x})} A(\mathbf{x}) h_A(\mathbf{x}) + \mathbf{w}_k,$$

where $A(\mathbf{x})$ is the complex echo of the target of state \mathbf{x} , $h_A(\mathbf{x})$ represents the point spread function for a target of state \mathbf{x} and \mathbf{w}_k is the measurement noise.

In particular, assume there is n_k closely spaced targets, then the complex (I/Q) signal is given by:

$$\mathbf{z}_{A,k} = \sum_{j=1}^{n_k} A(\mathbf{x}_{k,j}) h_A(\mathbf{x}_{k,j}) + \mathbf{w}_k.$$

Here,

- $A(\mathbf{x}_{k,j})$, the complex echo of the target j , is modeled by:

$$A(\mathbf{x}_{k,j}) = \bar{A}(\mathbf{x}_{k,j}) e^{i\varphi_k} + \mathbf{a}_k^{(j)},$$

with $\bar{A}(\mathbf{x}_{k,j})$ a known amplitude, φ_k an unknown phase, uniformly distributed on $[0, 2\pi)$, and $\mathbf{a}_k^{(j)}$ a zero-mean complex Gaussian variable with variance $\sigma_{\mathbf{a}_k^{(j)}}^2$. Then the PDF of the complex target echo conditioned on the amplitude and the variance, is given by

$$p\left(A(\mathbf{x}_{k,j}) | \bar{A}(\mathbf{x}_{k,j}), \sigma_{\mathbf{a}_k^{(j)}}^2\right) = \frac{1}{\pi \sigma_{\mathbf{a}_k^{(j)}}^2} \exp\left(-\frac{|A(\mathbf{x}_{k,j})|^2 + (\bar{A}(\mathbf{x}_{k,j}))^2}{\sigma_{\mathbf{a}_k^{(j)}}^2}\right) I_0\left(\frac{2\bar{A}(\mathbf{x}_{k,j})|A(\mathbf{x}_{k,j})|}{\sigma_{\mathbf{a}_k^{(j)}}^2}\right),$$

where $I_0(\cdot)$ is the modified Bessel function of zero order defined by:

$$I_0(y) := \sum_{j=0}^{\infty} \frac{\left(\frac{y^2}{4}\right)^j}{j! \Gamma(j+1)}.$$

- $h_A(\mathbf{x}_{k,j}) = [h_A(\mathbf{x}_{k,j})^1 \dots h_A(\mathbf{x}_{k,j})^m]^T$ represents the point spread function for the target of state vector $\mathbf{x}_{k,j}$,
- \mathbf{w}_k is a m -dimensional vector representing the measurement noise assumed to be a zero mean, white complex Gaussian noise with variance $\sigma_{\mathbf{w}_k}^2$.

Proposition 3.2.1. *Under the assumption of a non-fluctuating target model (Swierling 0), the probability density of the measurements \mathbf{z}_k^c conditioned on the multi-target state \mathbf{X}_k , i.e. the multi-target observation likelihood $\phi^c(\mathbf{z}_k^c|\mathbf{X}_k)$, is a noncentral chi-squared distribution and reduces to an exponential distribution, if \mathbf{z}_k^c covers only clutter noise and thermal noise.*

Proof. Under the assumption of a non-fluctuating target model, the complex echo of the target of state \mathbf{x} is modeled by:

$$A(\mathbf{x}) = \bar{A}(\mathbf{x}) e^{j\varphi_k}, \quad \varphi_k \in [0, 2\pi).$$

Let $\Sigma_{h_k^c} := \sum_{\mathbf{x} \in \mathbf{X}_k: c \in C(\mathbf{x})} \bar{A}(\mathbf{x}) h_A(\mathbf{x})$ for notational convenience, then the complex (I/Q) signal in cell c is defined by:

$$\begin{aligned} \mathbf{z}_{A,k}^c &= \sum_{\mathbf{x} \in \mathbf{X}_k: c \in C(\mathbf{x})} A(\mathbf{x}) h_A(\mathbf{x}) + \mathbf{w}_k^c, \\ &= \Sigma_{h_k^c} e^{j\varphi_k} + \mathbf{w}_k^c. \end{aligned}$$

Thus the power signal measured in cell c is given by:

$$\begin{aligned} \mathbf{z}_k^c &= |\mathbf{z}_{A,k}^c|^2 \\ &= |\Sigma_{h_k^c} e^{j\varphi_k} + \mathbf{w}_k^c|^2 \\ &= \left(\Sigma_{h_k^c} \cos(\varphi_k) + \Re(\mathbf{w}_k^c) \right)^2 + \left(\Sigma_{h_k^c} \sin(\varphi_k) + \Im(\mathbf{w}_k^c) \right)^2 \\ &= U_R^2 + U_I^2. \end{aligned}$$

$U_R \sim \mathcal{N}\left(\Sigma_{h_k^c} \cos(\varphi_k), \frac{\sigma_{\mathbf{w}_k}^2}{2}\right)$ and $U_I \sim \mathcal{N}\left(\Sigma_{h_k^c} \sin(\varphi_k), \frac{\sigma_{\mathbf{w}_k}^2}{2}\right)$ are statistically independent normal random variables. It follows that $\sqrt{U_R^2 + U_I^2}$ has a Rice distribution, and reduces to a Rayleigh distribution when $\Sigma_{h_k^c} = 0$. Hence, the measurement likelihood is a noncentral chi-squared distribution which can be written as:

$$\phi(\mathbf{z}_k^c|\mathbf{X}_k) = \frac{1}{\sigma_{\mathbf{w}_k}^2} \exp\left(-\frac{\mathbf{z}_k^c + \Sigma_{h_k^c}}{\sigma_{\mathbf{w}_k}^2}\right) I_0\left(\sqrt{\frac{4 \mathbf{z}_k^c \Sigma_{h_k^c}}{\sigma_{\mathbf{w}_k}^2}}\right),$$

and reduces to an exponential distribution, if there is only clutter noise and thermal noise in the cell c , as:

$$\varphi(\mathbf{z}_k^c) = \frac{1}{\sigma_{\mathbf{w}_k}^2} \exp\left(-\frac{\mathbf{z}_k^c}{\sigma_{\mathbf{w}_k}^2}\right),$$

where $I_0(\cdot)$ is the modified Bessel function. □

Remark. Other target fluctuation models are modeled and treated in a similar way, for instance under the assumption of a Swierling fluctuation model I/II [130], the conditional likelihood of the complex signal is given by:

$$\phi(\mathbf{z}_{A,k}|\mathbf{X}_k) = \frac{1}{|\pi Q_z|} \exp(-\|\mathbf{z}_{A,k}\|_{Q_z}^2 - \|\Sigma_{h_k}\|_{Q_z}^2) I_0(2|\mathbf{z}_{A,k}^H Q_z^{-1} \Sigma_{h_k}|), \quad (3.22)$$

with $x^H := \overline{x^T}$ the conjugate transpose (Hermitian conjugate) of x , $\|x\|_Q^2$ a mathematical shorthand for $x^H Q^{-1} x$, $\Sigma_{h_k} := \sum_{j=1}^{n_k} \bar{A}(\mathbf{x}_{k,j}) h_A(\mathbf{x}_{k,j})$ and the covariance matrix:

$$Q_z = \sum_{j=1}^{n_k} h_A(\mathbf{x}_{k,j}) h_A(\mathbf{x}_{k,j})^H (\bar{A}(\mathbf{x}_{k,j}) \sigma_{\mathbf{a}_k^{(j)}})^2 + I_{m \times m} \sigma_{\mathbf{w}_k}^2.$$

The resulting likelihood seems to be involved at first sight involving the calculation of the determinant and the inverse of a covariance matrix with dimension equal to the number of cells. However, by keeping in mind some elementary linear algebra and realizing that the point spread function as a limited support, these calculations reduce to a tractable complexity.

For $\sigma_{\mathbf{a}_k^{(j)}}^2 = 0$, equation (3.22) reduces to the complex zero mean Gaussian distribution,

$$\phi(\mathbf{z}_{A,k} | \mathbf{X}_k, \sigma_{\mathbf{a}_k^{(j)}}^2 = 0) = \frac{1}{|\pi \sigma_{\mathbf{w}_k}^2|} \exp \left(-\frac{|\mathbf{z}_{A,k}|^2 - |\Sigma_{h_k}|^2}{\sigma_{\mathbf{w}_k}^2} \right) I_0 \left(\frac{2 |\mathbf{z}_{A,k} \Sigma_{h_k}|}{\sigma_{\mathbf{w}_k}^2} \right).$$

Hence, $\phi(|\mathbf{z}_{A,k}| / \mathbf{X}_k, \sigma_{\mathbf{a}_k^{(j)}}^2 = 0)$ is the regular Rice distribution.

3.2.4 Multi-target Bayes Recursion

The multi-target filtering problem can be cast as a Bayes filter on the state space $\mathcal{F}(\mathcal{X})$ and observation space \mathcal{Z} . Let $\mathbf{z}_{1:k} = \{\mathbf{z}_1, \dots, \mathbf{z}_k\}$ denote the set of measurements collected up to and including time k . Then, the multi-target Bayes recursion propagates the multi-target posterior density $f_k(\cdot | \mathbf{z}_{1:k})$ in time according to:

PREDICTION:

$$f_{k|k-1}(\mathbf{X}_k | \mathbf{z}_{1:k-1}) = \int \Pi_{k|k-1}(\mathbf{X}_k | \mathbf{X}_{k-1}) f_{k-1}(\mathbf{X}_{k-1} | \mathbf{z}_{1:k-1}) \delta \mathbf{X}_{k-1}, \quad (3.23)$$

UPDATE:

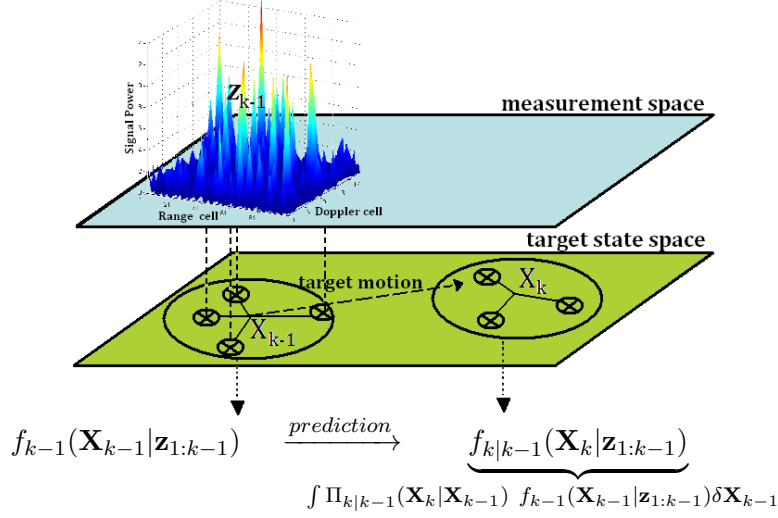
$$f_k(\mathbf{X}_k | \mathbf{z}_{1:k}) = \frac{\vartheta_k(\mathbf{z}_k | \mathbf{X}_k) f_{k|k-1}(\mathbf{X}_k | \mathbf{z}_{1:k-1})}{\int \vartheta_k(\mathbf{z}_k | \mathbf{X}) f_{k|k-1}(\mathbf{X} | \mathbf{z}_{1:k-1}) \delta \mathbf{X}}, \quad (3.24)$$

where $\Pi_{k|k-1}(\cdot | \cdot)$ is the multi-target (Markov) transition density (3.13), from time $k-1$ to k , $\vartheta_k(\cdot | \cdot)$ is the multi-target likelihood function (3.22) at time k and the integrals above are set integrals defined for any function $f : \mathcal{F}(\mathbb{X} \times \mathbb{L}) \rightarrow \mathbb{R}$ by:

$$\int f(\mathbf{X}) \delta \mathbf{X} = \sum_{i=0}^{\infty} \frac{1}{i!} \int f(\{\mathbf{x}_1, \dots, \mathbf{x}_i\}) d(\{\mathbf{x}_1, \dots, \mathbf{x}_i\}).$$

The multi-target prediction (3.23) is simply the total probability theorem for the previous posterior and current transition densities. The multi-target update (3.24) is simply the Bayes rule for probability densities.

Multi-target Bayes Prediction (p.g.fl.)



Let $G_{k|k-1}[\cdot | \mathbf{z}_{1:k-1}]$ denote the p.g.fl. of the multi-target predicted density $f_{k|k-1}(\cdot | \mathbf{z}_{1:k-1})$ and let $G_{\pi,k|k-1}[\cdot | \mathbf{X}_{k-1}]$ denote the p.g.fl. of the multi-target Markov transition density $\Pi_{k|k-1}(\cdot | \mathbf{X}_{k-1})$ at time k . Then, the multi-target Bayes prediction (3.23) can alternatively be stated in terms of p.g.fl.s as follows:

$$G_{k|k-1}[h | \mathbf{z}_{1:k-1}] = \int G_{\pi,k|k-1}[h | \mathbf{X}_{k-1}] f_{k-1}(\mathbf{X}_{k-1} | \mathbf{z}_{1:k-1}) \delta \mathbf{X}_{k-1}. \quad (3.25)$$

Let $\mathbf{X}_{k-1} = \{\mathbf{x}_1, \dots, \mathbf{x}_{n'}\}$, given a realization \mathbf{X}_{k-1} of the multi-target state at time $k-1$, the multi-target state \mathbf{X}_k at time k is given by the union (3.12):

$$\mathbf{X}_k \equiv \Xi_{k|k-1} = T_{k|k-1}(\mathbf{X}_{k-1}) \cup \Gamma_k.$$

This can be written in p.g.fl. form as the superposition of independent point processes, i.e.

MULTI-TARGET MARKOV P.G.FL.

$$G_{k|k-1}[h | \mathbf{X}_{k-1}] := G_1[h | \mathbf{x}_1] \dots G_{n'}[h | \mathbf{x}_{n'}] \cdot G_\Gamma[h], \quad (3.26)$$

with,

$G_i[h | x_i]$ the INDIVIDUAL TARGET P.G.FL. (Bernoulli RFS)

$$G_i[h | x_i] = (1 - p_S(\mathbf{x}_i)) + p_S(\mathbf{x}_i) \int h(x) \pi_{k|k-1}(x | \mathbf{x}_i) dx, \quad (3.27)$$

and

$G_\Gamma[h]$ is the BIRTH PROCESS P.G.FL.

1. If Γ_k is a Poisson RFS with intensity function ψ_k , then

$$G_\Gamma[h] = e^{\langle \psi_k, h-1 \rangle}. \quad (3.28)$$

2. If Γ_k is a multi-Bernoulli RFS with parameter set $\{r_\Gamma^{(i)}, s_\Gamma^{(i)}(\cdot)\}_{i=1}^{M_\Gamma}$, then

$$G_\Gamma[h] = \prod_{i=1}^{M_\Gamma} \left((1 - r_\Gamma^{(i)}) + r_\Gamma^{(i)} \int h(x) s_\Gamma^{(i)}(x) dx \right). \quad (3.29)$$

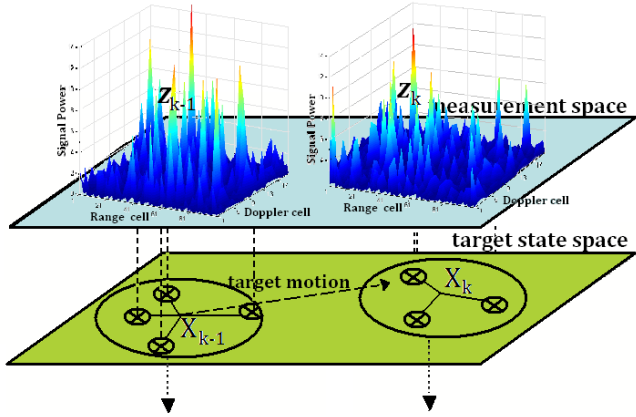
Then, we find back the formulae for the multi-target transition density (3.14)-(3.15). That is, in the special case where Γ_k is a Poisson RFS with intensity function ψ_k , the transition density simplifies to:

$$\Pi_{k|k-1}(X|X') = (1 - p_{S,k}(x'))^{X'} \frac{(K_S \psi_k)^{|X|}}{e^{-\langle \psi_k, 1 \rangle}} \sum_{W \subseteq X} \sum_{\tau \in \mathcal{T}(W, X')} \left(\frac{q_{S,k,\tau}}{\psi_k} \right)^W.$$

If Γ_k is a multi-Bernoulli RFS with parameter set $\{r_\Gamma^{(i)}, s_\Gamma^{(i)}(\cdot)\}_{i=1}^{M_\Gamma}$ or equivalently $\{q_{\Gamma,k}^{(i)}\}_{i=1}^{M_\Gamma}$, then the transition density is given by:

$$\Pi_{k|k-1}(X|X') = \frac{(1 - p_{S,k}(x'))^{X'} K_S^{|X|}}{(1 - \langle q_{\Gamma,k}, 1 \rangle)^{\{1, \dots, M_\Gamma\}}} \sum_{W \subseteq \mathbf{X}_k} \sum_{\tau \in \mathcal{T}(W, X')} \sum_{v \in \mathcal{T}(W, \{1, \dots, M_\Gamma\})} q_{\Gamma,k,v}^X \left(\frac{q_{S,k,\tau}}{q_{\Gamma,k,v}} \right)^W.$$

Multi-target Bayes Update



$$f_{k|k-1}(\mathbf{X}_k | \mathbf{z}_{1:k-1}) \xrightarrow{\text{update}} \underbrace{f_k(\mathbf{X}_k | \mathbf{z}_{1:k})}_{\frac{\vartheta_k(\mathbf{z}_k | \mathbf{X}_k) f_{k|k-1}(\mathbf{X}_k | \mathbf{z}_{1:k-1})}{\int \vartheta_k(\mathbf{z}_k | \mathbf{X}_k) f_{k|k-1}(\mathbf{X}_k | \mathbf{z}_{1:k-1}) \delta \mathbf{X}_k}} \quad (3.30)$$

By substituting the multi-target likelihood $\vartheta_k(\mathbf{z}_k | \mathbf{X}_k)$ given by equation (3.18) into the expression of the posterior density $f_k(\mathbf{X}_k | \mathbf{z}_{1:k})$ given by equation (3.24), $f(\mathbf{z}_k)$ cancels out. Hence, the posterior density simplifies as follows:

$$f_k(\mathbf{X}_k | \mathbf{z}_{1:k}) = \frac{\prod_{x \in \mathbf{X}_k} g_{\mathbf{z}_k}(x) f_{k|k-1}(\mathbf{X}_k | \mathbf{z}_{1:k-1})}{\int \prod_{x \in X} g_{\mathbf{z}_k}(x) f_{k|k-1}(X | \mathbf{z}_{1:k-1}) \delta X}. \quad (3.31)$$

Sequential Monte-Carlo Implementations of the Multi-target Bayes Filter

The main goal of this chapter is to apply and extend model-based integrated detection and tracking for a Pulse Doppler radar by means of Sequential Monte-Carlo methods, allowing also for ambiguity problems to be solved over time. This chapter describes practical SMC implementations of the RFS multi-target Bayes recursion (3.23)-(3.24).

The basic strategy is to propagate a set of weights and samples, representing the posterior density, recursively in time. Section 4.1 discusses a direct application of standard particle approximation techniques (cf. [9, 54] for complete details on these techniques). When the state space is very high, the performance of particle filters depends to a large extent on the choice of proposal distribution. In order to tackle more general and more complex probability distributions, Markov Chain Monte Carlo methods are needed. In the particle filtering framework, MCMC can be used for drawing the samples from an invariance distribution, either in sampling step or resampling step. Readers without a background in MCMC simulation are encouraged to consult Appendix C for an introduction to the field. Section 4.2 reviews sequential Markov chain Monte Carlo methods, in particular the marginalized MCMC-Based particle filter [84].

In section 4.3 we introduce our new algorithm, which we called IP-MCMC-PF, discuss the improvements attainable by using a fully parallelizable PF, and give theoretical justifications for our approach. We then extend our algorithm to handle a varying number of targets. We consider two approaches. The first approach reviewed exploits Reversible Jumps [72](Section 4.4). However, such an approach strongly relies on the correct knowledge of the Markov matrix that is used to model the cardinality time evolution. An alternative approach uses the IP-MCMC-PF as the core engine of a multiple cardinality hypotheses tracker (Section 4.5).

Then section 4.6 investigates some of the practical aspects of particle filter design for our specific application: Radar surveillance. Finally demonstrations and numerical studies of the proposed implementations are presented in section 4.7.

4.1 Standard SMC Implementation

This section presents a generic SMC implementation of the Bayes multi-target filter. The single-target particle filter can be directly generalized to the multi-target case. The statistical behavior of the RFS \mathbf{X}_k , modeled by (3.12), is characterized by the multi-target transition density $\Pi_{k|k-1}(\cdot|\cdot)$ (3.13) in an analogous fashion to the single-target transition density $\pi_{k|k-1}(\cdot|\cdot)$. The multi-target transition density $\Pi_{k|k-1}(\cdot|\cdot)$ incorporates all aspects of motion of multiple targets such as the time-varying number of targets, individual target motion, target birth/death and target interactions. Likewise, the statistical behavior of the vector \mathbf{z}_k , modeled by (3.16), can be described by the multi-target likelihood $\vartheta_k(\mathbf{z}_k|\mathbf{X}_k)$ (3.31) in an analogous fashion to the single-target likelihood function $g_k(\mathbf{z}_k|\mathbf{x}_k)$.

SMC filtering techniques permit recursive propagation of the set of weighted particles that approximate the posterior density [54, 56]. The propagation of the multi-target posterior density recursively in time involves the evaluation of multiple set integrals and hence the computational requirement is much more intensive than single-target filtering. Central in Monte Carlo methods is the notion of approximating the integrals of interest using random samples. In the multi-target context, each particle is a finite set and the particles themselves can thus be of varying dimensions.

Suppose at time $k - 1$, a set of weighted particles $\{w_{k-1}^{(i)}, \mathbf{X}_{k-1}^{(i)}\}_{i=1}^N$ representing the multi-target posterior $f_{k-1}(\cdot)$ is available, i.e.

$$f_{k-1}(\mathbf{X}_{k-1}|\mathbf{z}_{1:k-1}) \approx \sum_{i=1}^N w_{k-1}^{(i)} \delta_{\mathbf{X}_{k-1}^{(i)}}(\mathbf{X}_{k-1}). \quad (4.1)$$

Then, for a given proposal density $q_k(\cdot|\mathbf{X}_{k-1}^{(i)}, \mathbf{z}_k)$ satisfying $\text{support}(f_k) \subseteq \text{support}(q_k)$, the multi-target posterior $f_k(\cdot)$ at time k is approximated by a new set of weighted particles $\{w_k^{(i)}, \mathbf{X}_k^{(i)}\}_{i=1}^N$, i.e.

$$f_k(\mathbf{X}_k|\mathbf{z}_{1:k}) \approx \sum_{i=1}^N w_k^{(i)} \delta_{\mathbf{X}_k^{(i)}}(\mathbf{X}_k), \quad (4.2)$$

where,

$$\tilde{\mathbf{X}}_k^{(i)} \sim q_k(\cdot|\mathbf{X}_{k-1}^{(i)}, \mathbf{z}_k), \quad (4.3)$$

$$\tilde{w}_k^{(i)} = w_{k-1}^{(i)} \frac{\vartheta_k(\mathbf{z}_k|\tilde{\mathbf{X}}_k^{(i)}) \Pi_{k|k-1}(\tilde{\mathbf{X}}_k^{(i)}|\mathbf{X}_{k-1}^{(i)})}{q(\tilde{\mathbf{X}}_k^{(i)}|\mathbf{X}_{k-1}^{(i)}, \mathbf{z}_k)}, \quad (4.4)$$

$$w_k^{(i)} = \frac{\tilde{w}_k^{(i)}}{\sum_{i=1}^N \tilde{w}_k^{(i)}}. \quad (4.5)$$

The recursion is initialized by generating a set of weighted particles $\{\mathbf{X}_0^{(i)}, w_0^{(i)}\}_{i=1}^N$ representing f_0 . Equations (4.3)-(4.4)-(4.5) then provide a recursion for computing the set of weighted particles representing $f_k(\cdot)$ from those representing $f_{k-1}(\cdot)$ when a new measurement arrives. In this context, the importance sampling density $q_k(\cdot|\mathbf{X}_{k-1}^{(i)}, \mathbf{z}_k)$ is a multi-target density and $\tilde{\mathbf{X}}_k^{(i)} \sim q_k(\cdot|\mathbf{X}_{k-1}^{(i)}, \mathbf{z}_k)$ is a sample from a point process. Details on sampling from a point process can be found in the spatial statistics literature [46]. Selection of the importance distribution is a critical part of the design phase in particle filtering. The importance (sampling) proposals are in general selected either by using the minimum variance or the transition prior approach [36].

Statistics are calculated across the set of particles to provide estimates of the states, for example the Minimum Mean-Square Error (MMSE) estimator or the Maximum A Posteriori (MAP) estimator.

Finally, a resampling step is used to prevent the variance of the particle weights to increase over time. There are many possible ways to perform the resampling step. Most methods consist of copying each particle $\tilde{\mathbf{X}}_k^{(i)}$ $N_k^{(i)}$ times under the constraint $\sum_{i=1}^N N_k^{(i)} = N$ to obtain $\{\mathbf{X}_k^{(i)}\}_{i=1}^N$. The (random) resampling mechanism is chosen such that $\mathbb{E}[N_k^{(i)}] = N\alpha_k^{(i)}$ where $\alpha_k^{(i)} > 0$, $\sum_{i=1}^N \alpha_k^{(i)} = 1$ is a sequence of weights set by the user. This resampling step could be achieved using multinomial resampling but the efficient stratified resampling algorithm described in [86] has better statistical properties. The new weights are set to $w_k^{(i)} \propto \tilde{w}_k^{(i)} / \alpha_k^{(i)}$, $\sum_{i=1}^N w_k^{(i)} = 1$. Typically, $\alpha_k^{(i)} = \tilde{w}_k^{(i)}$ but alternatively we can select $\alpha_k^{(i)} \propto (\tilde{w}_k^{(i)})^\beta$ where $\beta \in [0, 1]$. Under standard assumptions, the mean squared error of the SMC approximation is inversely proportional to the number of particles. Thus, the particle filter proceeds as described in Algorithm 2.

Algorithm 2: Multi-target SIR Particle Filter outline

input : $\{\mathbf{X}_{k-1}^{(i)}, w_{k-1}^{(i)}\}_{i=1}^N$ and a new measurement, \mathbf{z}_k

output: $\{\mathbf{X}_k^{(i)}, w_k^{(i)}\}_{i=1}^N$

At time $k \geq 1$,

1 - *Sampling Step:*

for $i \leftarrow 1$ **to** N **do**

Sample $\tilde{\mathbf{X}}_k^{(i)} \sim q_k(\cdot|\mathbf{X}_{k-1}^{(i)}, \mathbf{z}_k)$.

Set weights : $\tilde{w}_k^{(i)} = w_{k-1}^{(i)} \frac{\vartheta_k(\mathbf{z}_k|\tilde{\mathbf{X}}_k^{(i)}) \Pi_{k|k-1}(\tilde{\mathbf{X}}_k^{(i)}|\mathbf{X}_{k-1}^{(i)})}{q(\tilde{\mathbf{X}}_k^{(i)}|\mathbf{X}_{k-1}^{(i)}, \mathbf{z}_k)}$.

end

Normalize the weights : $\sum_{i=1}^N \tilde{w}_k^{(i)} = 1$.

2 - *Resampling Step:*

Resample $\{\tilde{\mathbf{X}}_k^{(i)}, \tilde{w}_k^{(i)}\}_{i=1}^N$ to get $\{\mathbf{X}_k^{(i)}, w_k^{(i)}\}_{i=1}^N$.

In this algorithm, each particle corresponds to a sample from a point process and the particles can thus be of varying dimensions. The weight update can be computationally demanding due to the combinatorial nature of the multi-target Markov transition and likelihood. However, the main practical problem with this implementation is the need to perform importance sampling in very high dimensional spaces if many targets are present. First, if too few particles are used, all but a few importance weights will be near zero. Resampling will then lead to a loss of diversity among the particles, also known as the sample impoverishment problem. Second, it can be difficult to come up with an efficient importance density. A poor selection of $q_k(\cdot|\mathbf{X}_{k-1}^{(i)})$ will lead to a high-variance estimator. In particular, a naive choice of importance density such as $q_k(\cdot|\mathbf{X}_{k-1}^{(i)}, \mathbf{z}_k) = f_{k|k-1}(\cdot|\mathbf{X}_{k-1}^{(i)})$ will typically lead to an algorithm whose efficiency decreases exponentially with the number of targets for a fixed number of particles. To deal with multi-modal posterior probability density functions in large dimensional state spaces, such as in our application, where hundreds of (possibly disappearing and reappearing) targets need to be tracked, Markov Chain Monte Carlo (MCMC) methods can be efficiently used [54, 84, 133].

Remark. Due to particle degeneracy, the standard multi-target Sampling Importance Resampling (SIR)-PF is not able to consistently maintain the multi-modality of the target distributions that may arise due to ambiguity or the presence of multiple targets. See [6, 7, 22, 32] for complete details on this, so called self-resolving, phenomenon and its consequences on tracking.

4.2 Sequential Markov Chain Monte Carlo

Several approaches combining SMC with MCMC methods have already been proposed in the literature. For instance, MCMC kernels have been used to design efficient regularization steps in order to rejuvenate degenerate particles in the SIR-PF [64]. Recently, Sequential Markov Chain Monte Carlo (SMCMC) approaches were proposed in [5, 84, 126]. These methods are distinct from the resample-move scheme [64] since an SMC algorithm is used to design efficient high-dimensional proposal distributions for an MCMC sampler. In other words the inefficient importance sampling step of the standard SIR-PF implementation is replaced by a more efficient MCMC sampling step. These methods allow to design effective MCMC algorithms in complex scenarios where standard strategies failed.

Within the Bayesian estimation framework, our aim is to compute the filtering PDF $f_k(\mathbf{X}_k|\mathbf{z}_{1:k})$ recursively by:

$$f_k(\mathbf{X}_k|\mathbf{z}_{1:k}) \propto \int \vartheta_k(\mathbf{z}_k|\mathbf{X}_k) \Pi_{k|k-1}(\mathbf{X}_k|\mathbf{X}_{k-1}) f_{k-1}(\mathbf{X}_{k-1}|\mathbf{z}_{1:k-1}) \delta \mathbf{X}_{k-1}. \quad (4.6)$$

Let us represent the density $f_{k-1}(\mathbf{X}_{k-1}|\mathbf{z}_{1:k-1})$ by a set of unweighted particles,

$$f_{k-1}(\mathbf{X}_{k-1}|\mathbf{z}_{1:k-1}) \approx \frac{1}{N} \sum_{i=1}^N \delta_{\mathbf{X}_{k-1}^{(i)}}(\mathbf{X}_{k-1}), \quad (4.7)$$

where N is the number of particles and (i) the particle index. Then, by plugging this particle approximation into (4.6):

$$f_k(\mathbf{X}_k|\mathbf{z}_{1:k}) \approx \frac{1}{N} \vartheta_k(\mathbf{z}_k|\mathbf{X}_k) \sum_{i=1}^N \Pi(\mathbf{X}_k|\mathbf{X}_{k-1}^{(i)}). \quad (4.8)$$

In [84], an M-H algorithm (cf. Appendix C.2.2) is designed in a sequential setting in order to approximate the filtering distribution (4.6). This is achieved by using the approximate posterior (4.8) as the target distribution with a proposal distribution $q(\cdot|\mathbf{X}_k^{n_{M-H}})$. The acceptance ratio is then given by:

$$\alpha = \min \left(1, \frac{f_k(\mathbf{X}_k^*|\mathbf{z}_{1:k})}{f_k(\mathbf{X}_k^{n_{M-H}}|\mathbf{z}_{1:k})} \frac{q(\mathbf{X}_k^{n_{M-H}}|\mathbf{X}_k^*)}{q(\mathbf{X}_k^*|\mathbf{X}_k^{n_{M-H}})} \right) \quad (4.9)$$

$$= \min \left(1, \frac{p(\mathbf{X}_k^*)}{p(\mathbf{X}_k^{n_{M-H}})} \frac{\vartheta_k(\mathbf{z}_k|\mathbf{X}_k^*)}{\vartheta_k(\mathbf{z}_k|\mathbf{X}_k^{n_{M-H}})} \frac{q(\mathbf{X}_k^{n_{M-H}}|\mathbf{X}_k^*)}{q(\mathbf{X}_k^*|\mathbf{X}_k^{n_{M-H}})} \right). \quad (4.10)$$

The desired approximation $\hat{f}_k(\mathbf{X}_k|\mathbf{z}_{1:k})$ is obtained by storing every N_{thin} th accepted sample after the initial burn-in iterations.

To avoid numerical integration of the predictive density at every MCMC iteration, an alternative algorithm has been developed in [126]. This algorithm, which considers the joint posterior distribution $f_{k,k-1}(\mathbf{X}_k, \mathbf{X}_{k-1}|\mathbf{z}_{1:k})$, involves a joint M-H proposal step where both \mathbf{X}_k and \mathbf{X}_{k-1} are updated jointly, as well as individual refinement Metropolis-within-Gibbs steps where \mathbf{X}_k and \mathbf{X}_{k-1} are updated individually. Furthermore, most of the methodologies developed in the SMC setting can be directly reapplied here. This includes Rao-Blackwellisation techniques to reduce the dimensionality of the target distributions [55, 90] or auxiliary particle-type methods to build distributions biased toward "promising" regions of the space [76, 115].

4.3 Interacting Population-based MCMC-PF

This section focuses on computational issues arising in the real-time implementation of multi-target Bayesian filtering. As a first contribution, a novel algorithm well suited to deal with multi-target tracking problems for a given cardinality is proposed. The Interacting Population-based MCMC-PF (IP-MCMC-PF) algorithm, first introduced in [24], makes use of several M-H samplers running in parallel, which interact through genetic variation.

This section is organized as follows. Subsection 4.3.1 gives insights for our approach. Subsection 4.3.2 introduces the proposed IP-MCMC-PF algorithm and addresses the proposal density design.

4.3.1 Justifications behind the IP-MCMC-PF Algorithm

There are two basic ideas behind the proposed algorithm:

(1) **RELIABLE STATISTICAL INFERENCES** for the target distribution is required. For this purpose we run multiple M-H sampling chains, each starting from different seeds, in parallel. A single, possibly time varying, transition kernel is used for all parallel chains. The only difference is the region of the space explored by each chain. After a sufficiently long burn-in period, the set of samples formed by the union of the converged M-H outputs approximates the stationary distribution. This means that, once enough stationary simulations have been drawn, mixing all sets of draws provides a good estimate of the target distribution.

Using multiple chains enjoys several benefits compared to a single chain scheme. This strategy explores larger regions of the sample space in a reasonable time. The simulations from the chains are spread across various high probability regions of the target distribution. Furthermore, having a population of chains facilitates the implementation of interaction operators that manipulate information from different realizations for improving the next generation of samples.

Note that, with parallel processing engines, using multiple chains is usually computationally more efficient than using a single long chain. However, even when running multiple chains, a burn-in period is required for each chain to move from its random starting state to a high probability region of the target distribution.

(2) RAPID MIXING within each MCMC chain is required. The rate of convergence to stationarity of a finite Markov chain is typically measured by the so-called mixing time, defined as the number of steps required for the Markov chain to approach the stationary distribution, also referred as the steady-state distribution, "close enough" [103]. "Rapid mixing" means the mixing time is polynomial in the logarithm of the state space size. Our aim is to speed up the MCMC convergence for minimizing the burn-in period. For that the history from all the chains is used to adapt the kernel and therefore to guide any particular population member in the exploration of the state space toward regions of higher probability. Thus, a time homogeneous Markov transition kernel can be constructed from both local¹ proposal moves and global² moves between the MCMC chains and this in turn leads to faster mixing rate.

Furthermore, the proposed implementation resorts to within-chain analysis to monitor stationarity and between/within chains comparisons to monitor mixing. Combined, stationarity and mixing lead to the convergence of the set of MCMC chains.

4.3.2 IP-MCMC-PF Algorithm

Let us consider a multi-target setting where $\mathbf{X}_k = [\mathbf{x}_{k,1}, \dots, \mathbf{x}_{k,n_k}]$ is the multi-target state vector, with $\mathbf{x}_{k,j}$ being the state vector of the j^{th} target at time k . Assume the prior distribution over target states at time $k-1$ is represented by a set of N particles $\{\mathbf{X}_{k-1}^{(i)}\}_{i=1}^N$. Each particle contains the joint state of n_k partitions corresponding to the different targets. We refer to $\mathbf{x}_{k-1,j}^{(i)}$ as the states of the j^{th} partition of the particle i at time step $k-1$. The particle state vector is given by:

$$\mathbf{X}_{k-1}^{(i)} = [\mathbf{x}_{k-1,1}^{(i)}, \dots, \mathbf{x}_{k-1,n_k}^{(i)}]. \quad (4.11)$$

Let N_{MCMC} denote the number of MCMC chains that can operate in parallel. The parameter N_{MCMC} depends on firstly, the experiment setup (targets in track and modeling complexity) and secondly, the system specification (parallel processing potential, memory resources). At each time step k , we use N_{MCMC} M-H samplers, running in parallel, to generate a set of samples from the approximate posterior (4.8) over the joint target configurations \mathbf{X}_k . Each sampler s is initiated with a configuration

¹Local moves are made by selecting a chain and applying a dynamic model.

²Global moves are made by randomly selecting two chains, swapping over the values of each chain, and applying a dynamic model.

$\{\mathbf{X}_s^{(0)}; \ell_s^{(0)}\}$, where $\mathbf{X}_s^{(0)}$ denotes a particle and $\ell_s^{(0)}$ denotes its associated joint log-likelihood. In particular, each seed $\mathbf{X}_s^{(0)}$ is obtained as follows: a particle is randomly selected from the prior $\{\mathbf{X}_{k-1}^{(i)}\}_{i=1}^N$, this particle is then propagated based on the dynamic model.

The pseudo-code for each M-H algorithm in this context is as follows:

Algorithm 3: M-H algorithm

Start with an initial configuration $\{\mathbf{X}_s^{(0)}; \ell_s^{(0)}\}$.

for $l \leftarrow 1$ **to** $B + N_{M-H}$ **do**

 Propose a new configuration \mathbf{X}_s^* by sampling from the proposal density function $q(\mathbf{X}_s^* | \mathbf{X}_s^{(l-1)})$.

 Sample $u \sim \mathcal{U}_{[0,1]}$, where $\mathcal{U}_{[0,1]}$ a uniform distribution in $[0, 1]$.

 Calculate the acceptance ratio:

$$\alpha = \frac{f_k(\mathbf{X}_s^* | \mathbf{z}_{1:k})}{f_k(\mathbf{X}_s^{(l-1)} | \mathbf{z}_{1:k})} \frac{q(\mathbf{X}_s^{(l-1)} | \mathbf{X}_s^*)}{q(\mathbf{X}_s^* | \mathbf{X}_s^{(l-1)})}. \quad (4.12)$$

if $u \leq \min(1, \alpha)$ **then**

 Accept move: $\mathbf{X}_s^{(l)} = \mathbf{X}_s^*$, $\ell_s^{(l)} = \ell_s^*$,

else

 Reject move: $\mathbf{X}_s^{(l)} = \mathbf{X}_s^{(l-1)}$, $\ell_s^{(l)} = \ell_s^{(l-1)}$.

end

end

Discard B initial burn-in samples to allow the M-H sampler to converge to the stationary distribution. Store the other samples.

At the end of the procedure, the initial burn-in simulations, which are strongly influenced by starting values, are discarded, while the remaining samples $\mathbf{X}_s \triangleq \{\mathbf{X}_s^{(i)}\}_{i=B+1}^{B+N_{M-H}}$ are stored, with $N_{M-H} := N/N_{MCMC}$. The set of samples formed by the union of the converged M-H outputs $\{\mathbf{X}_s\}_{s=1}^{N_{MCMC}}$ summarizes the target distribution at time step k .

Well thought-out design of the proposal density plays an important role in the success of M-H sampling. We consider a proposal density q that selects a single partition j , chosen from all partitions with uniform probability, and updates its state $\mathbf{x}_{s,j}^{(l-1)}$ by sampling from the single target proposal density $\pi(\mathbf{x}_{s,j}^* | \mathbf{x}_{k-1,j}^{(i)})$, where $\mathbf{x}_{k-1,j}^{(i)}$ correspond to the partition j of a particle i , which is randomly picked out of the prior $\{\mathbf{X}_{k-1}^{(i)}\}_{i=1}^N$. Given the previous configuration $\{\mathbf{X}_s^{(l-1)}; \ell_s^{(l-1)}\}$, a new configuration $\{\mathbf{X}_s^*; \ell_s^*\}$ is obtained by replacing partition j by $\mathbf{x}_{s,j}^*$ and updating the joint log-likelihood. Therefore the proposal $q(\mathbf{X}_s^* | \mathbf{X}_s^{(l-1)})$ is given by:

$$q(\mathbf{X}_s^* | \mathbf{X}_s^{(l-1)}) = \sum_{j \in \mathcal{P}_k} q(j) q(\mathbf{X}_s^* | \mathbf{X}_s^{(l-1)}, j), \quad (4.13)$$

where $q(j)$ selects the partition index to be updated from the set of partitions \mathcal{P}_k , and $q(\mathbf{X}_s^* | \mathbf{X}_s^{(l-1)}, j)$ is the partition-specific proposal distribution:

$$q(j) = \frac{1}{n_k}, \text{ with } n_k \text{ the number of targets,} \quad (4.14)$$

and

$$q(\mathbf{X}_s^* | \mathbf{X}_s^{(l-1)}, j) = \pi(\mathbf{x}_{s,j}^* | \mathbf{x}_{k-1,j}^{(i)}) \underbrace{\prod_{i \neq j} \pi(\mathbf{x}_{s,i}^* | \mathbf{x}_{s,i}^{(l-1)})}_{\text{keep all other partitions fixed}} \delta(\mathbf{x}_{s,i}^* - \mathbf{x}_{s,i}^{(l-1)}). \quad (4.15)$$

In particular, the move $\pi(\mathbf{x}_{s,j}^* | \mathbf{x}_{k-1,j}^{(i)})$ seeks to update the single partition $\mathbf{x}_{k-1,j}^{(i)}$ via a Markov kernel that describes the state dynamics.

Remark. Note that, this move, also known as Mutation (subsection 4.6.4), allows for local exploration of the state space, maintains genetic diversity and ensures the required irreducibility of the Markov chain.

This proposal density design is computationally efficient: by only perturbing the state of one target per iteration, most factors in the acceptance ratio (4.12) cancel. By replacing the target and proposal distributions by their definitions, the resulting acceptance ratio only depends on the likelihoods and can be written as:

$$\alpha = \frac{\vartheta_k(\mathbf{z}_k | \mathbf{X}_s^*)}{\vartheta_k(\mathbf{z}_k | \mathbf{X}_s^{(l-1)})} = \frac{\ell_s^*}{\ell_s^{(l-1)}}. \quad (4.16)$$

Remark. Let us denote by $\mathbf{p}\ell_{s,j}^{(i)}$ the partial joint log-likelihood of all partitions interacting in the ambiguous measurement space with the partition j at the i^{th} M-H iteration. Then, by caching the previous joint log-likelihood $\ell_s^{(l-1)}$ as well as $\mathbf{p}\ell_{s,j}^{(l-1)}$, only the evaluation of partial joint log-likelihood $\mathbf{p}\ell_{s,j}^*$ is necessary for each iteration of the M-H algorithm. In particular the joint log-likelihood ℓ_s^* can be compute as follows:

$$\ell_s^* = \ell_s^{(l-1)} - \mathbf{p}\ell_{s,j}^{(l-1)} + \mathbf{p}\ell_{s,j}^*. \quad (4.17)$$

The proposal density has the appealing property that the joint likelihood evaluation does not depend on the number of tracked targets, but only on the number of targets in interaction. In comparison, the joint SIR-PF requires a likelihood evaluation for each target per sample and suffers from exponential complexity in the number of tracked targets.

In the proposed algorithm, a validation test is performed to check the convergence on the basis of Gelman and Rubin [62] diagnostic. This approach consists, for each state parameter, of three steps: first computing the variance of the samples from each chain (after discarding burn-in); second averaging these within-chain variances; and third comparing this to the variances of all the chains mixed together via the potential scale reduction factor [35], $\hat{\mathbf{R}}$. For further details cf. Appendix C.2.3. The parallel chains are considered well-mixed when $\hat{\mathbf{R}}$ is less or equal than 1.1 for all parameters. Once the set of chains have reached approximate convergence, the M-H sampling chains outputs, i.e. the N_{MCMC} sets of simulations, all mixed together give the new set of particles $\{\mathbf{X}_k^{(i)}\}_{i=1}^N$ which approximates the target distribution $f_k(\mathbf{X}_k | \mathbf{z}_{1:k})$.

The proposed IP-MCMC-PF algorithm is summarized in Algorithm 5. A description of the M-H sampling procedure is given by Algorithm 4.

Algorithm 4: M-H sampling procedure

input : $\{\mathbf{X}_s^{(0)}; \ell_s^{(0)}\}$ initial configuration and $\{\mathbf{X}_k^{(i)}\}_{i=1}^N$.

output: $\{\mathbf{X}_s^{(l)}\}_{l=1}^{N_{\text{M-H}}}$ with $N_{\text{M-H}} := N/N_{\text{MCMC}}$.

for $l \leftarrow 1$ **to** $B + N_{\text{M-H}}$ **do**

 1 - *Propose a new configuration* \mathbf{X}_s^* :

 Randomly select a partition $\mathbf{x}_{k-1,j}^{(i)}$ out of $\{\mathbf{X}_{k-1}^{(i)}\}_{i=1}^N$.

 Given $\mathbf{x}_{k-1,j}^{(i)}$, draw $\mathbf{x}_{s,j}^* = t_k(\mathbf{x}_{k-1,j}^{(i)}, \mathbf{v}_{k-1,j})$.

 Propose new configuration \mathbf{X}_s^* , such that

$\mathbf{X}_s^* = \{\mathbf{x}_{s,1}^{(l-1)}, \dots, \mathbf{x}_{s,j-1}^{(l-1)}, \mathbf{x}_{s,j}^*, \mathbf{x}_{s,j+1}^{(l-1)}, \dots, \mathbf{x}_{s,n_k}^{(l-1)}\}$.

 Compute the joint log-likelihood $\ell_s^* = \ell_s^{(l-1)} - \mathbf{p}\ell_{s,j}^{(l-1)} + \mathbf{p}\ell_{s,j}^*$.

 2 - *Accept/Reject*:

 Sample $u \sim \mathcal{U}_{[0,1]}$, $\mathcal{U}_{[0,1]}$ a uniform distribution in $[0, 1]$.

 Calculate the M-H acceptance ratio: $\alpha = \frac{\ell_s^*}{\ell_s^{(l-1)}}$.

if $u \leq \min(1, \alpha)$ **then**

 | Accept move: $\mathbf{X}_s^{(l)} = \mathbf{X}_s^*$, $\ell_s^{(l)} = \ell_s^*$,

else

 | Reject move: $\mathbf{X}_s^{(l)} = \mathbf{X}_s^{(l-1)}$, $\ell_s^{(l)} = \ell_s^{(l-1)}$.

end

end

Discard B initial burn-in samples to allow the M-H sampler to converge to the stationary distribution.

Store the remaining samples $\{\mathbf{X}_s^{(l)}\}_{l=B+1}^{B+N_{\text{M-H}}} \rightarrow \{\mathbf{X}_s^{(l)}\}_{l=1}^{N_{\text{M-H}}}$.

Algorithm 5: IP-MCMC-PF algorithm

input : $\{\mathbf{X}_{k-1}^{(i)}\}_{i=1}^N$ and a new measurement, \mathbf{z}_k .

output: $\{\mathbf{X}_k^{(i)}\}_{i=1}^N$.

1 - *Initialize the N_{MCMC} M-H samplers:*

Select a random subset $\{\tilde{\mathbf{X}}_s^{(0)}\}_{s=1}^{N_{\text{MCMC}}} \subset \{\mathbf{X}_{k-1}^{(i)}\}_{i=1}^N$.

while $s \leftarrow 1$ **to** N_{MCMC} **do**

 Propagate $\tilde{\mathbf{X}}_s^{(0)}$ according to the dynamic model:

while $j \leftarrow 1$ **to** n_k **do**

$\mathbf{x}_{s,j}^{(0)} = t_k(\tilde{\mathbf{x}}_{s,j}^{(0)}, \mathbf{v}_{k-1,j})$.

end

 Compute the associated joint log-likelihood:

$\ell_s^{(0)} = \log \vartheta_k(\mathbf{z}_k | \mathbf{X}_s^{(0)})$.

end

Store the seeds in a cache $\{\mathbf{X}_s^{(0)}; \ell_s^{(0)}\}_{s=1}^{N_{\text{MCMC}}}$.

2 - *Apply M-H sampling procedure:*

Run $s = 1, \dots, N_{\text{MCMC}}$ M-H samplers in parallel (Algorithm 4):

$\{\mathbf{X}_s^{(l)}\}_{l=1}^{N_{\text{M-H}}} = \text{M-H}\left(\{\mathbf{X}_s^{(0)}; \ell_s^{(0)}\}, \{\mathbf{X}_{k-1}^{(i)}\}_{i=1}^N\right)$.

3 - *Check convergence:*

Compute $\hat{\mathbf{R}}$: cf. Appendix C.2.3 eq. C.32.

if $\hat{\mathbf{R}} \geq 1.1$ **then**

 Run the chains out longer to improve convergence to the stationary distribution and increase the precision of inferences.

else

 Mix all the N_{MCMC} sets of simulations together to obtain $\{\mathbf{X}_k^{(i)}\}_{i=1}^N$.

end

4.4 IP based Reversible Jump MCMC-PF

As a second contribution, in order to solve the tracking problem for a large, unknown and time-varying number of targets, an extension of the IP-MCMC-PF [25], which falls within the Reversible-Jump Markov Chain Monte Carlo (RJMCMC) framework [72], is proposed. Incorporation of RJMCMC methods into a tracking framework gives the possibility to naturally and efficiently deal with multiple appearing and disappearing targets.

This section is organized as follows. Subsection 4.4.1 summarizes the main aspects of the IP-RJMCMC-PF algorithm. Subsection 4.4.2 details the modified M-H sampling procedure. Subsection 4.4.3 addresses the design of the set of reversible moves and of the corresponding proposal distributions.

4.4.1 Bayesian Estimation and Reversible-Jump MCMC

In the reversible jump case, the filtering PDF over the varying dimension state space \mathcal{X} is denoted by $f_k(\mathbf{X}_k, \mathcal{I}_{n_k} | \mathbf{z}_{1:k})$, with \mathcal{I}_{n_k} the set of identifiers (labels) of targets currently in view. To simplify the notation let $\mathbf{X}_{\mathcal{I},k}$ denote $\{\mathbf{X}_k, \mathcal{I}_{n_k}\}$. The Monte Carlo approximation of the posterior $f_k(\mathbf{X}_{\mathcal{I},k} | \mathbf{z}_{1:k})$ is then given by a set of unweighted particles $\{\mathbf{X}_{\mathcal{I},k}^{(i)}\}_{i=1}^N$. In particular, the filtering PDF $f_k(\mathbf{X}_{\mathcal{I},k} | \mathbf{z}_{1:k})$ can be expressed recursively using the Bayes filter equation by:

$$f(\mathbf{X}_{\mathcal{I},k} | \mathbf{z}_{1:k}) \propto \vartheta(\mathbf{z}_k | \mathbf{X}_{\mathcal{I},k}) \times \sum_{\mathcal{I}_{n_{k-1}}} \int \Pi(\mathbf{X}_{\mathcal{I},k} | \mathbf{X}_{\mathcal{I},k-1}) f(\mathbf{X}_{\mathcal{I},k-1} | \mathbf{z}_{1:k-1}) d\mathbf{X}_{k-1}, \quad (4.18)$$

where the multi-target transition density $\Pi_{k|k-1}(\cdot | \cdot)$, defined according to (3.13), incorporates all aspects of motion of multiple targets such as the time-varying number of targets, individual target motion, target birth/death and target interactions. Thus, the following Monte Carlo approximation is obtained:

$$f_k(\mathbf{X}_{\mathcal{I},k} | \mathbf{z}_{1:k}) \approx \vartheta(\mathbf{z}_k | \mathbf{X}_{\mathcal{I},k}) \frac{1}{N} \sum_{i=1}^N \Pi(\mathbf{X}_{\mathcal{I},k} | \mathbf{X}_{\mathcal{I},k-1}^{(i)}). \quad (4.19)$$

Each sample now includes a set of identifiers $\mathcal{I}_{n_{k-1}}^{(i)}$ of partitions currently active.

As in the IP-MCMC-based particle filter, at each time step k , we use N_{MCMC} M-H samplers, running in parallel, to generate a set of samples from the approximate posterior (4.19). Each modified M-H algorithm inherently sets up a Markov chain whose invariant distribution corresponds to the posterior of interest $f_k(\mathbf{X}_{\mathcal{I},k} | \mathbf{z}_{1:k})$. After an appropriate burn-in period, which is required for the Markov chain to reach equilibrium, the states at successive iterations represent samples from the distribution of interest $f_k(\mathbf{X}_{\mathcal{I},k} | \mathbf{z}_{1:k})$. In the reversible jump case, candidate samples are chosen from a set of proposal distributions, which are randomly accepted according to an acceptance ratio that ensures reversibility and, therefore, the invariance of the Markov chain with respect to the desired posterior distribution $f_k(\mathbf{X}_{\mathcal{I},k} | \mathbf{z}_{1:k})$.

The proposed IP based Reversible-Jump MCMC-PF (IP-RJMCMC-PF) algorithm is summarized in Algorithm 6.

Algorithm 6: IP-RJMCMC-PF algorithm

input : $\{\mathbf{X}_{k-1}^{(i)}, \mathcal{I}_{n_{k-1}}^{(i)}\}_{i=1}^N$ and a new measurement, \mathbf{z}_k .

output: $\{\mathbf{X}_k^{(i)}, \mathcal{I}_{n_k}^{(i)}\}_{i=1}^N$.

1 - *Initialize the N_{MCMC} M-H samplers:*

Select a random subset $\{\tilde{\mathbf{X}}_s^{(0)}\}_{s=1}^{N_{\text{MCMC}}} \subset \{\mathbf{X}_{k-1}^{(i)}\}_{i=1}^N$.

while $s \leftarrow 1$ **to** N_{MCMC} **do**

 Propagate $\tilde{\mathbf{X}}_s^{(0)}$ according to the dynamic model:

while $j \leftarrow 1$ **to** n_k **do**

$\mathbf{x}_{s,j}^{(0)} = t_k(\tilde{\mathbf{x}}_{s,j}^{(0)}, \mathbf{v}_{k-1,j})$.

end

 Compute the associated joint log-likelihood:

$\ell_s^{(0)} = \log \vartheta_k(\mathbf{z}_k | \mathbf{X}_s^{(0)})$.

end

Store the seeds in a cache $\{\mathbf{X}_s^{(0)}, \mathcal{I}_{s,n_k^{(0)}}; \ell_s^{(0)}\}_{s=1}^{N_{\text{MCMC}}}$.

2 - *Apply modified M-H sampling procedure:*

Run N_{MCMC} reversible jump M-H samplers in parallel (Algorithm 7):

$\{\mathbf{X}_s^{(i)}, \mathcal{I}_{s,n_k^{(i)}}\}_{i=1}^{N_{\text{M-H}}} = \text{RJ-MH}\left(\{\mathbf{X}_s^{(0)}, \mathcal{I}_{s,n_k^{(0)}}; \ell_s^{(0)}\}, \{\mathbf{X}_{k-1}^{(i)}\}_{i=1}^N\right)$.

3 - *Check convergence:*

Compute $\hat{\mathbf{R}}$: cf. Appendix C.2.3 eq. C.32.

if $\hat{\mathbf{R}} \geq 1.1$ **then**

 Run the chains out longer to improve convergence to the stationary distribution.

else

 Mix all the N_{MCMC} sets of simulations together to obtain $\{\mathbf{X}_k^{(i)}, \mathcal{I}_{n_k^{(i)}}\}_{i=1}^N$.

end

4.4.2 Reversible-Jump M-H Sampling Procedure

Within the Reversible-Jump Markov Chain Monte Carlo (RJMCMC) framework [72], the M-H algorithm is modified to define a Markov chain over a variable dimension state space. Each sampler s is initiated with a configuration $\{\mathbf{X}_s^{(0)}, \mathcal{I}_{s, n_k^{(0)}}; \ell_s^{(0)}\}$ and iterates until $N_{\text{M-H}} := N/N_{\text{MCMC}}$ samples are acquired. $\mathbf{X}_s^{(0)}$ is the particle seed, $\mathcal{I}_{s, n_k^{(0)}}$ and $\ell_s^{(0)}$ denote the set of identifiers of partitions currently active within $\mathbf{X}_s^{(0)}$ and the joint log-likelihood associated to $\mathbf{X}_s^{(0)}$ respectively. To simplify the notation let $\mathbf{X}_{\mathcal{I}, s}^{(l)}$ denote $\{\mathbf{X}_s^{(l)}, \mathcal{I}_{s, n_k^{(l)}}\}$ after the l^{th} M-H iteration.

At each M-H iteration, a move type ν is first selected with probability p_ν from a finite set of moves \mathcal{V} . Each move either increases the dimensionality of the state (target birth), decreases it (target death), or leaves it unchanged (state update). A move that changes the dimensionality of the state is referred to as a jump.

Then, a new state $\mathbf{X}_{\mathcal{I}, s}^* := \{\mathbf{X}_s^*, \mathcal{I}_{s, n_k^*}\}$ is proposed using the proposal density $q_\nu(\mathbf{X}_{\mathcal{I}, s}^* | \mathbf{X}_{\mathcal{I}, s}^{(l-1)})$. Let $q_{\nu^R}(\mathbf{X}_{\mathcal{I}, s}^{(l-1)} | \mathbf{X}_{\mathcal{I}, s}^*)$ denote the reverse proposal density, where ν^R is the reverse jump corresponding to ν . Crucially, every jump should have a corresponding reverse jump defined, e.g. a birth move requires a corresponding death move, hence the name Reversible-Jump MCMC.

Finally, the new state is accepted with probability α_ν , $\mathbf{X}_{\mathcal{I}, s}^{(l)} = \mathbf{X}_{\mathcal{I}, s}^*$, and is used for the next iteration. In the case of rejection, the current state is kept $\mathbf{X}_{\mathcal{I}, s}^{(l)} = \mathbf{X}_{\mathcal{I}, s}^{(l-1)}$. The acceptance ratio α_ν gauges the improvement in the quality of the proposed sample $\mathbf{X}_{\mathcal{I}, s}^*$ over the previous sample $\mathbf{X}_{\mathcal{I}, s}^{(l-1)}$. In the reversible-jump M-H sampling procedure, dimension jumps occur. Comparing likelihoods of different dimension would be meaningless. To overcome this problem, when a move from the $n_k^{(l-1)}$ -dimensional space to the n_k^* -dimensional space is proposed, we define the dimension matching functions $f_{n_k^{(l-1)} \rightarrow n_k^*}$ and $f_{n_k^* \rightarrow n_k^{(l-1)}}$, which allow us to extend one space into the other, and vice versa. In this case, the acceptance ratio is then computed as follows:

$$\alpha_\nu = \frac{f_k(\mathbf{X}_{\mathcal{I}, s}^* | \mathbf{z}_{1:k})}{f_k(\mathbf{X}_{\mathcal{I}, s}^{(l-1)} | \mathbf{z}_{1:k})} \frac{p_{\nu^R}}{p_\nu} \frac{q_{\nu^R}(\mathbf{X}_{\mathcal{I}, s}^{(l-1)} | \mathbf{X}_{\mathcal{I}, s}^*)}{q_\nu(\mathbf{X}_{\mathcal{I}, s}^* | \mathbf{X}_{\mathcal{I}, s}^{(l-1)})} \left| J_{f_{n_k^{(l-1)} \rightarrow n_k^*}} \right|, \quad (4.20)$$

which will achieve detailed balance at the desired target density $f_k(\mathbf{X}_{\mathcal{I}, k} | \mathbf{z}_{1:k})$. p_ν and p_{ν^R} are the probabilities of choosing move ν and the reverse move ν^R , respectively. $J_{f_{n_k^{(l-1)} \rightarrow n_k^*}}$ denotes the Jacobian determinant of the diffeomorphism from $\mathbf{X}_{\mathcal{I}, s}^{(l-1)}$ to $\mathbf{X}_{\mathcal{I}, s}^*$. The choice of the extended spaces, deterministic transformation $f_{n_k^{(l-1)} \rightarrow n_k^*}$ and proposal distributions for $q_\nu(\mathbf{X}_{\mathcal{I}, s}^* | \mathbf{X}_{\mathcal{I}, s}^{(l-1)})$ and $q_{\nu^R}(\mathbf{X}_{\mathcal{I}, s}^{(l-1)} | \mathbf{X}_{\mathcal{I}, s}^*)$ is problem dependent and needs to be addressed on a case by case basis. The procedure continues in the same fashion long enough to allow for convergence to the equilibrium distribution.

Thus, the reversible-jump M-H sampling procedure can be summarized as Algorithm 7.

Algorithm 7: Reversible-Jump M-H sampling procedure

input : $\{\mathbf{X}_s^{(0)}, \mathcal{I}_{s, n_k^{(0)}}; \ell_s^{(0)}\}$ initial configuration and $\{\mathbf{X}_k^{(i)}\}_{i=1}^N$.

output: $\{\mathbf{X}_s^{(i)}, \mathcal{I}_{s, n_k^{(i)}}\}_{i=1}^{N_{\text{M-H}}}$ with $N_{\text{M-H}} := N/N_{\text{MCMC}}$.

for $l \leftarrow 1$ **to** $B + N_{\text{M-H}}$ **do**

1 - *Propose a move type ν :*

Sample the move type $\nu \sim p_\nu$.

2 - *Propose a new configuration $\{\mathbf{X}_s^*, \mathcal{I}_{s, n_k^*}\}$:*

Sample from the proposal distribution:

$$q_\nu(\mathbf{X}_{\mathcal{I}, s}^* | \mathbf{X}_{\mathcal{I}, s}^{(l-1)}) = \sum_j q_\nu(j) q_\nu(\mathbf{X}_{\mathcal{I}, s}^* | \mathbf{X}_{\mathcal{I}, s}^{(l-1)}, j).$$

Compute the joint log-likelihood $\ell_s^* = \log \vartheta_k(\mathbf{z}_k | \mathbf{X}_s^*)$.

3 - *Accept/Reject:*

Sample $u \sim \mathcal{U}_{[0,1]}$, $\mathcal{U}_{[0,1]}$ a uniform distribution in $[0, 1]$.

Calculate the M-H acceptance ratio: α_ν .

if $u \leq \min(1, \alpha_\nu)$ **then**

Accept move:

$$\{\mathbf{X}_s^{(l)}, \mathcal{I}_{s, n_k^{(l)}}\} = \{\mathbf{X}_s^*, \mathcal{I}_{s, n_k^*}\}, \quad \ell_s^{(l)} = \ell_s^*,$$

else

Reject move:

$$\{\mathbf{X}_s^{(l)}, \mathcal{I}_{s, n_k^{(l)}}\} = \{\mathbf{X}_s^{(l-1)}, \mathcal{I}_{s, n_k^{(l-1)}}\}, \quad \ell_s^{(l)} = \ell_s^{(l-1)}.$$

end

end

Discard B initial burn-in samples to allow the M-H sampler to converge to the stationary distribution.

Store the remaining samples $\{\mathbf{X}_s^{(i)}, \mathcal{I}_{s, n_k^{(i)}}\}_{i=B+1}^{B+N_{\text{M-H}}} \rightarrow \{\mathbf{X}_s^{(i)}, \mathcal{I}_{s, n_k^{(i)}}\}_{i=1}^{N_{\text{M-H}}}$.

4.4.3 Reversible-Jump M-H Moves and Proposals

Practical application of the reversible-jump M-H sampling procedure requires the definition of a set of reversible moves and of their corresponding proposal distributions $q_\nu(\cdot|\cdot)$ and $q_{\nu^R}(\cdot|\cdot)$. In this work, three general types of moves, $\mathcal{V} = \{\text{birth } (b), \text{death } (d), \text{update } (u)\}$, are designed to explore the variable dimension space of possible target configurations. Choosing the move type is done by sampling from a move specific prior distribution, $\nu \sim p_\nu$, such that $p_b + p_d + p_u = 1$. The birth and death moves are a reversible pair, and the update move is self-reversible. The Reversible-Jump M-H sampler generates new samples by sampling from the proposal distribution:

$$q_\nu(\mathbf{X}_{\mathcal{I},s}^*|\mathbf{X}_{\mathcal{I},s}^{(l-1)}) = \sum_j q_\nu(j)q_\nu(\mathbf{X}_{\mathcal{I},s}^*|\mathbf{X}_{\mathcal{I},s}^{(l-1)}, j), \quad (4.21)$$

where the move specific proposal $q_\nu(j)$ selects a single target (partition) j that should be updated, added, or removed from $\mathbf{X}_s^{(l-1)}$.

Remark. Note that, by only perturbing the state of a single partition per iteration, these moves ensure that the absolute value of the Jacobian terms, which account for state dimensionality changes, reduce to unity.

Well thought-out design of the proposal densities plays an important role in the success of reversible-jump M-H sampling.

(1) Birth / Death Moves - Detection-driven Proposals

Let us assume the presence of a noisy target detector that returns at each time step k a set of identifiers \mathcal{I}_{d_k} , related to the detection plots, along with their estimated state $\mathbf{X}_{k,d}$.

Birth move

Let $\mathcal{I}_{b_k} \triangleq \mathcal{I}_{d_k} \setminus \mathcal{I}_{s,n_k}^{(l-1)}$ denote the subset of detection identifiers having a corresponding partition not yet active within the configuration $\mathbf{X}_{\mathcal{I},s}^{(l-1)}$.

We consider a birth proposal density q_b that:

1. selects an identifier j , chosen from \mathcal{I}_{b_k} with uniform probability,
2. appends a new partition state $\mathbf{x}_{s,j}^* \sim \mathcal{N}(\mathbf{x}_{k,d_j}, \sigma^2)$ to $\mathbf{X}_s^{(l-1)}$, while keeping all other partitions fixed,
3. adds the label j to $\mathcal{I}_{s,n_k}^{(l-1)}$.

Therefore, the birth proposal $q_b(\mathbf{X}_{\mathcal{I},s}^*|\mathbf{X}_{\mathcal{I},s}^{(l-1)})$ is given by:

$$q_b(\mathbf{X}_{\mathcal{I},s}^*|\mathbf{X}_{\mathcal{I},s}^{(l-1)}) = \sum_{j \in \mathcal{I}_{b_k}} q_b(j) q_b(\mathbf{X}_{\mathcal{I},s}^*|\mathbf{X}_{\mathcal{I},s}^{(l-1)}, j), \quad (4.22)$$

where $q_b(j)$ selects the partition index to be set, and $q_b(\mathbf{X}_{\mathcal{I},s}^*|\mathbf{X}_{\mathcal{I},s}^{(l-1)}, j)$ is the partition-specific birth proposal distribution:

$$q_b(j) = \frac{1}{|\mathcal{I}_{b_k}|}, \text{ with } |\cdot| \text{ denoting the cardinality of the set,} \quad (4.23)$$

and

$$q_b(\mathbf{X}_{\mathcal{I},s}^*|\mathbf{X}_{\mathcal{I},s}^{(l-1)}, j) = q_b(\mathbf{x}_{s,j}^*|\mathbf{x}_{k,d_j}) \underbrace{\prod_{i \neq j} \pi(\mathbf{x}_{s,i}^*|\mathbf{x}_{s,i}^{(l-1)}) \delta(\mathbf{x}_{s,i}^* - \mathbf{x}_{s,i}^{(l-1)})}_{\text{keep all other partitions fixed}}. \quad (4.24)$$

Remark. If all the partitions associated to \mathcal{I}_{d_k} are already active, we instead use a birth proposal density \tilde{q}_b that:

1. selects a partition j inactive within $\mathbf{X}_{\mathcal{I},s}^{(l-1)}$,
2. adds a new partition by sampling from the single target proposal density $q_b(\mathbf{x}_{s,j}^*|\mathbf{x}_{k-1,j}^{(i)})$, such that $\mathbf{x}_{k-1}^{(i)} \in \{\mathbf{X}_{k-1}^{(i)}\}_{i=1}^N$ and $\mathbf{x}_{k-1,j}^{(i)}$ an active partition,
3. adds the label j to $\mathcal{I}_{s,n_k}^{(l-1)}$.

The resulting birth proposal $q_b(\mathbf{X}_{\mathcal{I},s}^*|\mathbf{X}_{\mathcal{I},s}^{(l-1)})$ is then given by:

$$q_b(\mathbf{X}_{\mathcal{I},s}^*|\mathbf{X}_{\mathcal{I},s}^{(l-1)}) = \sum_{j \in \mathcal{I}_{s,n_k}^{(l-1)c}} q_b(j) q_b(\mathbf{X}_{\mathcal{I},s}^*|\mathbf{X}_{\mathcal{I},s}^{(l-1)}, j), \quad (4.25)$$

where $\mathcal{I}_{s,n_k}^{(l-1)c} \triangleq \mathcal{P}_k \setminus \mathcal{I}_{s,n_k}^{(l-1)}$, with \mathcal{P}_k denoting the set of partitions,

$$q_b(j) = \frac{1}{|\mathcal{I}_{s,n_k}^{(l-1)c}|}, \quad (4.26)$$

and

$$q_b(\mathbf{X}_{\mathcal{I},s}^*|\mathbf{X}_{\mathcal{I},s}^{(l-1)}, j) = q_b(\mathbf{x}_{s,j}^*|\mathbf{x}_{k-1,j}^{(i)}) \underbrace{\prod_{i \neq j} \pi(\mathbf{x}_{s,i}^*|\mathbf{x}_{s,i}^{(l-1)}) \delta(\mathbf{x}_{s,i}^* - \mathbf{x}_{s,i}^{(l-1)})}_{\text{keep all other partitions fixed}}. \quad (4.27)$$

If all the partitions within $\mathbf{X}_{\mathcal{I},s}^{(l-1)}$ are active, the probability p_b of selecting the birth proposal is set to zero.

Death move

Let us consider $\mathcal{I}_{s,n_k^{(l-1)}}$ as the indices of the partitions that are active within the configuration $\mathbf{X}_{\mathcal{I},s}^{(l-1)}$. The death move proposes a new multi-target configuration $\mathbf{X}_{\mathcal{I},s}^*$ by deleting one of the active partition from $\mathbf{X}_{\mathcal{I},s}^{(l-1)}$ and updating the joint log-likelihood. In particular, the death proposal density q_d selects a partition j , chosen from $\mathcal{I}_{s,n_k^{(l-1)}}$ with uniform probability, and then removes the partition and the label $\{\mathbf{x}_{s,j}^{(l-1)}, j\}$ from the configuration $\mathbf{X}_{\mathcal{I},s}^{(l-1)}$.

Therefore, the death proposal $q_d(\mathbf{X}_{\mathcal{I},s}^*|\mathbf{X}_{\mathcal{I},s}^{(l-1)})$ is given by:

$$q_d(\mathbf{X}_{\mathcal{I},s}^*|\mathbf{X}_{\mathcal{I},s}^{(l-1)}) = \sum_{j \in \mathcal{I}_{s,n_k^{(l-1)}}} q_d(j) q_d(\mathbf{X}_{\mathcal{I},s}^*|\mathbf{X}_{\mathcal{I},s}^{(l-1)}, j), \quad (4.28)$$

where $q_d(j)$ selects the partition index to be deleted, and $q_d(\mathbf{X}_{\mathcal{I},s}^*|\mathbf{X}_{\mathcal{I},s}^{(l-1)}, j)$ is the partition-specific death proposal distribution:

$$q_d(j) = \frac{1}{|\mathcal{I}_{s,n_k^{(l-1)}}|}, \text{ with } |\cdot| \text{ denoting the cardinality of the set,} \quad (4.29)$$

and

$$q_d(\mathbf{X}_{\mathcal{I},s}^*|\mathbf{X}_{\mathcal{I},s}^{(l-1)}, j) = q_d(\mathbf{x}_{s,j}^*) \underbrace{\prod_{i \neq j} \pi(\mathbf{x}_{s,i}^*|\mathbf{x}_{s,i}^{(l-1)}) \delta(\mathbf{x}_{s,i}^* - \mathbf{x}_{s,i}^{(l-1)})}_{\text{keep all other partitions fixed}}. \quad (4.30)$$

Remark. If all the partitions are inactive, the probability p_d of selecting the death proposal is set to zero.

Acceptance ratio

Substituting the filtering distributions (4.19) and the birth and death proposals from (4.22) and (4.28), the acceptance ratio for the birth move α_b becomes:

$$\alpha_b = \frac{\ell_s^*}{\ell_s^{(l-1)}} \frac{p_d}{p_b} \frac{q_d(j)}{q_b(j)}. \quad (4.31)$$

Similarly the acceptance ratio for the death move α_d becomes:

$$\alpha_d = \frac{\ell_s^*}{\ell_s^{(l-1)}} \frac{p_b}{p_d} \frac{q_b(j)}{q_d(j)}. \quad (4.32)$$

(2) Update move

This move is similar to the classic M-H update used in the IP-MCMC-PF. That is, the state of a single partition j , chosen from all active partitions $\mathcal{I}_{s,n_k^{(l-1)}}$ with uniform probability, is updated by sampling from the single target proposal density $\pi(\mathbf{x}_{s,j}^* | \mathbf{x}_{k-1,j}^{(i)})$, where $\mathbf{x}_{k-1,j}^{(i)}$ correspond to the partition j of a particle i , which is randomly picked out of the prior $\{\mathbf{X}_{k-1}^{(i)}\}_{i=1}^N$.

Therefore, the update proposal $q_u(\mathbf{X}_{\mathcal{I},s}^* | \mathbf{X}_{\mathcal{I},s}^{(l-1)})$ is given by:

$$q_u(\mathbf{X}_{\mathcal{I},s}^* | \mathbf{X}_{\mathcal{I},s}^{(l-1)}) = \sum_{j \in \mathcal{I}_{b_k}} q_u(j) q_u(\mathbf{X}_{\mathcal{I},s}^* | \mathbf{X}_{\mathcal{I},s}^{(l-1)}, j), \quad (4.33)$$

where $q_u(j)$ selects the partition index to be updated from the set of partitions actives $\mathcal{I}_{s,n_k^{(l-1)}}$, and $q_u(\mathbf{X}_{\mathcal{I},s}^* | \mathbf{X}_{\mathcal{I},s}^{(l-1)}, j)$ is the partition-specific update proposal distribution:

$$q_u(j) = \frac{1}{|\mathcal{I}_{s,n_k^{(l-1)}}|}, \text{ with } |\cdot| \text{ denoting the cardinality of the set,} \quad (4.34)$$

and

$$q_u(\mathbf{X}_{\mathcal{I},s}^* | \mathbf{X}_{\mathcal{I},s}^{(l-1)}, j) = \pi(\mathbf{x}_{s,j}^* | \mathbf{x}_{k-1,j}^{(i)}) \underbrace{\prod_{i \neq j} \pi(\mathbf{x}_{s,i}^* | \mathbf{x}_{s,i}^{(l-1)})}_{\text{keep all other partitions fixed}} \delta(\mathbf{x}_{s,i}^* - \mathbf{x}_{s,i}^{(l-1)}). \quad (4.35)$$

In particular, the move $\pi(\mathbf{x}_{s,j}^* | \mathbf{x}_{k-1,j}^{(i)})$ seeks to update the single partition $\mathbf{x}_{k-1,j}^{(i)}$ via a Markov kernel that describes the state dynamic. This proposal density design is computationally efficient: by only perturbing the state of one target per iteration, most factors in the acceptance ratio in (4.20) cancel. The resulting acceptance ratio only depends on the likelihoods and can be written as:

$$\alpha = \frac{\ell_s^*}{\ell_s^{(l-1)}}. \quad (4.36)$$

Having defined the necessary models and acceptance ratios, we run the reversible-jump M-H sampling procedure to obtain an approximation of the distribution $f_k(\mathbf{X}_k, \mathcal{I}_{n_k} | \mathbf{z}_{1:k})$, where the constructed Markov chain represents the believe distribution of the current target configuration given the observations.

4.5 Multiple Cardinality Hypotheses Particle Filter

As a third contribution, in order to handle a large, unknown and time-varying number of targets, an alternative approach is proposed. This approach consists to use the IP-MCMC-PF as the core engine of a Multiple Cardinality Hypotheses Tracker (MCHT), where each cardinality is treated independently (up to a limiting factor n_{max}). The proposed multi-target filter bears some similarity to the classical track-oriented multiple hypothesis tracking algorithm [20].

This section is organized as follows. In subsection 4.5.1, we give insights for our approach. Subsection 4.5.2 recalls the concept of labeled RFSs [138, 139] and some properties. Subsection 4.5.3 details the labeled multi-target Bayes recursion. Subsection 4.5.4 introduces the proposed MCHT and describes its SMC implementation.

4.5.1 Justifications behind the MCHT

The basic strategy is to propagate a time-varying number of hypothesized tracks, which are characterized by an identity or a label. Note that care must be taken in the application of the multi-target Bayes recursion to ensure distinctness of labels. To address the uniqueness of labels the proposed algorithm is built upon the concept of labeled RFSs [139](cf. subsection 4.5.2 below), and formally incorporates the propagation and estimation of track labels within the RFS filtering framework.

The basic idea is as follows. From a given configuration of targets at time $k - 1$, possible configurations at time k are proposed based on the dynamical and birth models. Then, the likelihood associated to each of these scenarios given the measurement is computed. Finally, only the hypotheses with the most significant probability are retained for the next step.

A three-level tree-based structure, that allows efficient and systematic management of all the possible configurations of targets, is used. The first level, referred as "Cardinality", lists hypotheses on the number of targets. Each of these cardinality hypotheses in turn gives rise to a series of combination of targets. The second level, denoted as "Label component", expands explicitly the discrete label component of the targets. Finally, the third level, called "Particle clouds", represents as particles sets the kinematic components of the targets state. The proposed structure also leads to simple and efficient schemes for pruning of unlikely components, and helps reduce the computational cost of the recursion.

The multi-target prediction and update steps are recursively applied to propagate the multi-target posterior forward in time on component by component basis. The multi-target dynamical model involves thinning, Markov shifts and superposition of new targets. The prediction for each existence probability is essentially the previous posterior density integrated with the survival probability, and subsequently scaled down by the previous existence probability. The prediction for each hypothesized track density is very similar to the single-target Bayes prediction, except for the incorporation of the survival probability and subsequent normalization.

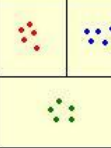
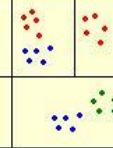
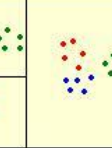
Cardinality	0	1	2	3
Label component	[]	[1] [2] [3]	[1,2] [1,3] [2,3]	[1,2,3]
Particle clouds				

Figure 4.1: Three-level tree-based structure.

Example 4.5.1. To better understand the MCHT algorithm, let us consider the following example:

1. First, suppose, at time $k - 1$, there are three target configurations:

- \mathbf{H}_1 A single target is present, with label $\ell_{k-1,1} = (k - 2, 1)$.
- \mathbf{H}_2 A single target is present, with label $\ell_{k-1,1} = (k - 1, 1)$.
- \mathbf{H}_3 Two targets are present, with labels $\ell_{k-1,1} = (k - 2, 1)$ and $\ell_{k-1,2} = (k - 1, 1)$ respectively.

A particle set represents the kinematic components of each target configuration.

2. Let us assume the birth process is a multi-Bernoulli density $\pi_\Gamma = \{(p_{B,k}(j), b_k^{(j)})\}_{j=1}^{|\mathcal{I}_{d_k}|}$ where $b_k^{(j)}(x) = \mathcal{N}(x; m_\gamma^{(j)}, P_\gamma)$ with $m_\gamma^{(j)}$ the estimated state of the j^{th} detection plot \mathbf{x}_{k,d_j} . In addition, consider a single detection plot. Then, at time k , the possible target configurations can be inferred from the dynamical and birth models.

Under \mathbf{H}_1 the new target configuration can be:

- \mathbf{H}_0 No target present.
- \mathbf{H}_1 A single target is present, with label $\ell_{k,1} = (k - 2, 1)$.
- \mathbf{H}_2 A single target is present, with label $\ell_{k,1} = (k, 1)$.
- \mathbf{H}_3 Two targets are present, with labels $\ell_{k,1} = (k - 2, 1)$ and $\ell_{k,2} = (k, 1)$ respectively.

Under \mathbf{H}_2 the new target configuration can be:

- \mathbf{H}_0 No target present.
- \mathbf{H}_1 A single target is present, with label $\ell_{k,1} = (k - 1, 1)$.
- \mathbf{H}_2 A single target is present, with label $\ell_{k,1} = (k, 1)$.
- \mathbf{H}_3 Two targets are present, with labels $\ell_{k,1} = (k - 1, 1)$ and $\ell_{k,2} = (k, 1)$ respectively.

Under \mathbf{H}_3 the new target configuration can be:

- \mathbf{H}_0 No target present.
- \mathbf{H}_1 A single target is present, with label $\ell_{k,1} = (k - 2, 1)$.
- \mathbf{H}_2 A single target is present, with label $\ell_{k,1} = (k - 1, 1)$.
- \mathbf{H}_3 A single target is present, with label $\ell_{k,1} = (k, 1)$.
- \mathbf{H}_4 Two targets are present, with labels $\ell_{k,1} = (k - 2, 1)$ and $\ell_{k,2} = (k - 1, 1)$ respectively.

- **H₅** Two targets are present, with labels $\ell_{k,1} = (k-2, 1)$ and $\ell_{k,2} = (k, 1)$ respectively.
- **H₆** Two targets are present, with labels $\ell_{k,1} = (k-1, 1)$ and $\ell_{k,2} = (k, 1)$ respectively.
- **H₇** Three targets are present, with labels $\ell_{k,1} = (k-2, 1)$, $\ell_{k,2} = (k-1, 1)$ and $\ell_{k,3} = (k, 1)$ respectively.

Going ahead in this way, for further steps, the tree of hypotheses grows exponentially. To ensure tractability, the predicted and updated multi-target densities should be truncated by keeping only components with the most significant probability.

4.5.2 Labeled Random Finite Sets

In applications such as multi-target tracking, apart from the target state, the target identity is also required, so that the target paths are consistently estimated. The targets should be distinctly identified by a label. This can be achieved by appending a label $\ell \in \mathbb{L}$ to each state $x \in \mathbb{X}$ where \mathbb{L} denotes the discrete space of distinct labels, and \mathbb{X} is the state space. To address the uniqueness of labels the concept of labeled RFSs, first introduced in [138, 139], is required.

Labeled Random Finite Sets

Definition 4.5.1. For any $\mathbf{X}_\ell \subset \mathbb{X} \times \mathbb{L}$, let $\mathcal{L} : \mathbb{X} \times \mathbb{L} \rightarrow \mathbb{L}$ be the projection $\mathcal{L}(\mathbf{X}_\ell) \triangleq \{\ell : (x, \ell) \in \mathbf{X}_\ell \text{ for some } x \in \mathbb{X}\}$. A labeled RFS \mathbf{X}_ℓ with state space \mathbb{X} and label space \mathbb{L} is an RFS on $\mathbb{X} \times \mathbb{L}$ such that each realization (x, ℓ) has distinct labels, i.e.

$$|\mathcal{L}(\mathbf{X}_\ell)| = |\mathbf{X}_\ell|.$$

Note that the set integral for a function $p : \mathcal{F}(\mathbb{X} \times \mathbb{L}) \rightarrow \mathbb{R}$ is defined by:

$$\int p(\mathbf{X}_\ell) \delta \mathbf{X}_\ell = \sum_{n=0}^{\infty} \frac{1}{n!} \int \sum_{(\ell_1, \dots, \ell_n) \in \mathbb{L}^n} p(\{(x_1, \ell_1), \dots, (x_n, \ell_n)\}) dx_1 \dots dx_n.$$

To evaluate set integrals involving labeled RFSs, the following Lemma 4.5.1 [138, 139] is required:

Lemma 4.5.1. Let $\Delta(\mathbf{X}_\ell)$ denote the distinct label indicator $\delta_{|\mathbf{X}_\ell|}(|\mathcal{L}(\mathbf{X}_\ell)|)$. Then for $h : \mathcal{F}(\mathbb{L}) \rightarrow \mathbb{R}$ and $g : \mathbb{X} \times \mathbb{L} \rightarrow \mathbb{R}$, integrable on \mathbb{X} ,

$$\int \Delta(\mathbf{X}_\ell) h(\mathcal{L}(\mathbf{X}_\ell)) g^{\mathbf{X}_\ell} \delta \mathbf{X}_\ell = \sum_{\mathbf{L} \subseteq \mathbb{L}} h(\mathbf{L}) \left[\int g(x, \cdot) dx \right]^{\mathbf{L}}. \quad (4.37)$$

Proof.

$$\begin{aligned} & \int \Delta(\mathbf{X}_\ell) h(\mathcal{L}(\mathbf{X}_\ell)) g^{\mathbf{X}_\ell} \delta \mathbf{X}_\ell \\ &= \sum_{n=0}^{\infty} \frac{1}{n!} \sum_{(\ell_1, \dots, \ell_n) \in \mathbb{L}^n} \delta_n(|\{\ell_1, \dots, \ell_n\}|) h(\{\ell_1, \dots, \ell_n\}) \int \left(\prod_{i=1}^n g(x_i, \ell_i) \right) dx_1 \dots dx_n, \\ &= \sum_{n=0}^{\infty} \frac{1}{n!} \sum_{(\ell_1, \dots, \ell_n) \in \mathbb{L}^n} \delta_n(|\{\ell_1, \dots, \ell_n\}|) h(\{\ell_1, \dots, \ell_n\}) \prod_{i=1}^n \left(\int g(x_i, \ell_i) dx_i \right). \end{aligned}$$

Due to the term $\delta_n(|\{\ell_1, \dots, \ell_n\}|)$, the sum becomes a sum over indices in \mathbb{L}^n with distinct components.

All $n!$ permutations of (ℓ_1, \dots, ℓ_n) share the same functional:

$$f(\{\ell_1, \dots, \ell_n\}) \triangleq h(\{\ell_1, \dots, \ell_n\}) \prod_{i=1}^n \left(\int g(x_i, \ell_i) dx_i \right),$$

and define an equivalence class $\{\ell_1, \dots, \ell_n\} \in F_n(\mathbb{L})$.

Thus,

$$\sum_{(\ell_1, \dots, \ell_n) \in \mathbb{L}^n} \delta_n(|\{\ell_1, \dots, \ell_n\}|) f(\{\ell_1, \dots, \ell_n\}) = n! \sum_{\{\ell_1, \dots, \ell_n\} \in F_n(\mathbb{L})} f(\{\ell_1, \dots, \ell_n\}).$$

Hence,

$$\int \Delta(\mathbf{X}_\ell) h(\mathcal{L}(\mathbf{X}_\ell)) g^{\mathbf{X}_\ell} \delta \mathbf{X}_\ell = \sum_{n=0}^{\infty} \sum_{\{\ell_1, \dots, \ell_n\} \in F_n(\mathbb{L})} h(\{\ell_1, \dots, \ell_n\}) \prod_{i=1}^n \left(\int g(x_i, \ell_i) dx_i \right).$$

The double sum can be combined as a sum over the subsets of \mathbb{L} and the result follows. \square

Example 4.5.2. A labeled multi-Bernoulli RFS \mathbf{X}_ℓ with state space \mathbb{X} , label space \mathbb{L} and finite parameter set $\{(r^{(\theta)}, s^{(\theta)}) : \theta \in \Theta\}$, is a multi-Bernoulli RFS on \mathbb{X} augmented with distinct labels corresponding to the successful (non-empty) Bernoulli components, i.e., if the Bernoulli component $(r^{(\theta)}, s^{(\theta)})$ yields a non-empty set, then the label assigned to the corresponding state is given by $\alpha(\theta) \in \mathbb{L}$ such that $\alpha : \Theta \rightarrow \mathbb{L}$ define a one-to-one mapping [139].

The following procedure illustrates how a sample from such a labeled multi-Bernoulli RFS is generated:

Algorithm 8: Sampling a labeled multi-Bernoulli RFS

```

Initialize  $\mathbf{X} = \emptyset$ .
for  $\theta \in \Theta$  do
    Sample  $u \sim \mathcal{U}[0, 1]$ .
    if  $u \leq r^{(\theta)}$  then
        Sample  $x \sim s^{(\theta)}(\cdot)$ .
        Set  $\mathbf{X} = \mathbf{X} \cup \{(x, \alpha(\theta))\}$ .
    end
end

```

The set of labeled states generated by the algorithm 8 has probability density given by:

$$\pi(\mathbf{X}_\ell) = \Delta(\mathbf{X}_\ell) 1_{\alpha(\Theta)}(\mathcal{L}(\mathbf{X}_\ell)) [\Phi(\mathbf{X}_\ell; \cdot)]^\Theta, \quad (4.38)$$

where $\Delta(\cdot)$ is the distinct label indicator, $\mathcal{L}(\mathbf{X}_\ell)$ the set of labels of \mathbf{X}_ℓ , and

$$\begin{aligned} \Phi(\mathbf{X}_\ell; \theta) &= \sum_{(x, \ell) \in \mathbf{X}_\ell} \delta_{\alpha(\theta)}(\ell) r^{(\theta)} s^{(\theta)}(x) + (1 - 1_{\mathcal{L}(\mathbf{X}_\ell)}(\alpha(\theta))) (1 - r^{(\theta)}), \\ &= \begin{cases} r^{(\theta)} s^{(\theta)}(x), & \text{if } (x, \alpha(\theta)) \in \mathbf{X}_\ell; \\ 1 - r^{(\theta)}, & \text{if } \alpha(\theta) \notin \mathcal{L}(\mathbf{X}_\ell). \end{cases} \end{aligned}$$

For instance, let us consider the probability density that the above procedure (algorithm 8) generates the points $(x_1, \ell_1), \dots, (x_n, \ell_n)$ in that order:

$$\pi(\{(x_1, \ell_1), \dots, (x_n, \ell_n)\}) = \delta_n(|\{\ell_1, \dots, \ell_n\}|) \prod_{\theta \in \Theta} (1 - r^{(\theta)}) \prod_{j=1}^n \frac{1_{\alpha(\Theta)}(\ell_j) r^{(\alpha^{-1}(\ell_j))} s^{(\alpha^{-1}(\ell_j))}(x_j)}{1 - r^{(\alpha^{-1}(\ell_j))}}, \quad (4.39)$$

where $1_Y(X) = 1$ if $X \subseteq Y$ and 0 otherwise.

The products $\prod_{\theta \in \Theta} (1 - r^{(\theta)}) \prod_{j=1}^n \frac{1_{\alpha(\Theta)}(\ell_j) r^{(\alpha^{-1}(\ell_j))} s^{(\alpha^{-1}(\ell_j))}(x_j)}{1 - r^{(\alpha^{-1}(\ell_j))}}$ can be written as:

$$\prod_{\theta \in \Theta} \left(\sum_{(x_j, \ell_j) \in \mathbf{X}_\ell} \delta_{\alpha(\theta)}(\ell_j) r^{(\theta)} s^{(\theta)}(x_j) + (1 - 1_{\mathcal{L}(\mathbf{X}_\ell)}(\alpha(\theta))) (1 - r^{(\theta)}) \right).$$

Generalized labeled multi-Bernoulli RFS

Definition 4.5.2. A generalized labeled multi-Bernoulli RFS is a labeled RFS with state space \mathbb{X} and label space \mathbb{L} distributed according to:

$$p(\mathbf{X}_\ell) = \Delta(\mathbf{X}_\ell) \sum_{c \in \mathbb{C}} w^{(c)}(\mathcal{L}(\mathbf{X}_\ell)) [p^{(c)}]^{\mathbf{X}_\ell}, \quad (4.40)$$

where,

- $\Delta(\mathbf{X}_\ell)$ is the distinct label indicator $\delta_{|\mathbf{X}_\ell|}(|\mathcal{L}(\mathbf{X}_\ell)|)$,
- \mathbb{C} denotes a discrete index set,
- the weight $w^{(c)}(\mathcal{L}(\mathbf{X}_\ell))$ only depends on the labels and satisfies:

$$\sum_{\mathbf{L} \subseteq \mathbb{L}} \sum_{c \in \mathbb{C}} w^{(c)}(\mathbf{L}) = 1,$$

- the multi-target exponential $[p^{(c)}]^{\mathbf{X}_\ell}$ depends on the entire multi-target state and satisfies

$$\int p^{(c)}(x, \ell) dx = 1.$$

Remark. A generalized labeled multi-Bernoulli can be interpreted as a mixture of multi-target exponentials [139]. Each term in the mixture (4.40) consists of a weight $w^{(c)}(\mathcal{L}(\mathbf{X}_\ell))$, and a multi-target exponential $[p^{(c)}]^{\mathbf{X}_\ell}$ that depends on the entire multi-target state.

The cardinality distribution of a generalized labeled multi-Bernoulli RFS is given by:

$$\rho(n) = \sum_{c \in \mathbb{C}} \sum_{\mathbf{L} \in \mathcal{F}(\mathbb{L})} w^{(c)}(\mathbf{L}).$$

Remark. It can be shown that generalized labeled multi-Bernoulli RFSs cover labeled multi-Bernoulli RFSs discussed above [139]. In particular the labeled multi-Bernoulli RFS described in (4.39) is a special case of the generalized labeled multi-Bernoulli RFS with:

$$\begin{aligned} p^{(c)}(x, \ell) &= s^{(\ell)}(x), \\ w^{(c)}(\mathbf{L}) &= \prod_{i=1}^{n_{MBer}} (1 - r^{(i)}) \prod_{j=1}^n \frac{1_{\mathbb{L}}(\ell) r^{(\ell_j)}}{1 - r^{(\ell_j)}}. \end{aligned}$$

4.5.3 Labeled Multi-target Bayes Recursion

Each target is identified by a unique label $\ell_k = (k, j)$, with k the time of birth and $j \in \mathbb{N}$ an index to distinguish targets born at the same time. Let us consider the following notations for the label spaces:

- \mathbb{B}_k denotes the label space for targets born at time k , i.e.

$$\mathbb{B}_k \triangleq \{k\} \times \mathbb{N}.$$

Thus a target born at time k has state $\mathbf{x}_k^\ell \in \mathbb{X} \times \mathbb{B}_k$.

- $\mathbb{L}_{0:k}$ denotes the label space for targets born up to and including time k and is constructed recursively by:

$$\mathbb{L}_{0:k} \triangleq \mathbb{L}_{0:k-1} \cup \{k\} \times \mathbb{N}.$$

Thus the multi-target state \mathbf{X}_k^ℓ at time k is a finite subset of $\mathbb{X} \times \mathbb{L}_{0:k}$.

Note that $\mathbb{L}_{0:k-1}$ and \mathbb{B}_k are disjoint.

Suppose that at time k there are n_k target states $\{\mathbf{x}_{k,1}^\ell, \dots, \mathbf{x}_{k,n_k}^\ell\}$, each taking values in the (labeled) state space $\mathbb{X} \times \mathbb{L}$. In the RFS formulation [95] the finite set of targets at time k is referred as the multi-target state:

$$\mathbf{X}_k^\ell = \{\mathbf{x}_{k,1}^\ell, \dots, \mathbf{x}_{k,n_k}^\ell\} \in \mathcal{F}(\mathbb{X} \times \mathbb{L}),$$

where $\mathcal{F}(\mathbb{X} \times \mathbb{L})$ is the multi-target state space, i.e. the finite subsets of $\mathbb{X} \times \mathbb{L}$. In TBD the multi-target observation at time k is an array $\mathbf{z}_k = [\mathbf{z}_k^1, \dots, \mathbf{z}_k^m]$, where each $\mathbf{z}_k^c \in \mathbb{C}$ is the complex (I/Q) signal in the radar cell indexed by c .

Let $\mathbf{z}_{1:k} = \{\mathbf{z}_1, \dots, \mathbf{z}_k\}$ denote the set of measurements collected up to and including time k . Then, the labeled multi-target Bayes recursion propagates the multi-target posterior density $f_k(\cdot | \mathbf{z}_{1:k})$ in time according to the following update and prediction:

PREDICTION:

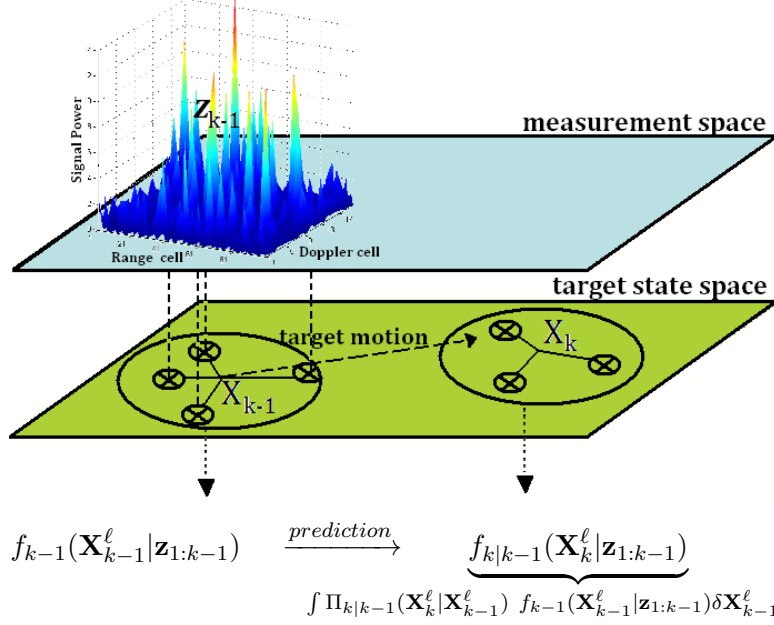
$$f_{k|k-1}(\mathbf{X}_k^\ell | \mathbf{z}_{1:k-1}) = \int \Pi_{k|k-1}(\mathbf{X}_k^\ell | \mathbf{X}_{k-1}^\ell) f_{k-1}(\mathbf{X}_{k-1}^\ell | \mathbf{z}_{1:k-1}) \delta \mathbf{X}_{k-1}^\ell, \quad (4.41)$$

UPDATE:

$$f_k(\mathbf{X}_k^\ell | \mathbf{z}_{1:k}) = \frac{\vartheta_k(\mathbf{z}_k | \mathbf{X}_k^\ell) f_{k|k-1}(\mathbf{X}_k^\ell | \mathbf{z}_{1:k-1})}{\int \vartheta_k(\mathbf{z}_k | \mathbf{X}^\ell) f_{k|k-1}(\mathbf{X}^\ell | \mathbf{z}_{1:k-1}) \delta \mathbf{X}^\ell}, \quad (4.42)$$

where the integrals above are set integrals, $\Pi_{k|k-1}(\cdot | \cdot)$ is the labeled RFS multi-target transition kernel, from time $k-1$ to k , and $\vartheta_k(\cdot | \cdot)$ denotes the labeled RFS multi-target likelihood function at time k .

Prediction



Let \mathbb{L}_{k-1} denote the label space at time $k-1$, \mathbb{B}_k the label space for targets born at time k and \mathbb{L}_k the label space at time k such that $\mathbb{L}_{k-1} \cap \mathbb{B}_k = \emptyset$ and $\mathbb{L}_k = \mathbb{L}_{k-1} \cup \mathbb{B}_k$. The multi-target dynamical model is described by the Markov multi-target transition kernel $\Pi_{k|k-1}(\mathbf{X}_k^\ell | \mathbf{X}_{k-1}^\ell)$ involving thinning of disappearing targets, Markov shifts of surviving targets and superposition of new targets.

Given the multi-target state \mathbf{X}_{k-1}^ℓ at time $k-1$, each $(\mathbf{x}_{k-1}, \ell_{k-1}) \in \mathbf{X}_{k-1}^\ell$ either continues to exist with probability $p_{S,k}(\mathbf{x}_{k-1}, \ell_{k-1})$ and moves to a new state (\mathbf{x}_k, ℓ_k) with probability density $\pi_{k|k-1}(\mathbf{x}_k | \mathbf{x}_{k-1}, \ell_{k-1}) \delta_{\ell_{k-1}}(\ell_k)$, or disappears with probability $q_{S,k}(\mathbf{x}_{k-1}, \ell_{k-1}) = 1 - p_{S,k}(\mathbf{x}_{k-1}, \ell_{k-1})$. Thus, assuming that, conditional on \mathbf{X}_{k-1}^ℓ , the transition of the target kinematic states are mutually independent, then the set $\mathbf{X}_S^\ell \subset \mathbf{X}_k^\ell$ of surviving targets at time k is modeled by:

- a LABELED MULTI-BERNOULLI RFS $S_{k|k-1}(\mathbf{x}_{k-1}, \ell_{k-1})$ with parameter set:

$$\{r^{(\mathbf{x}_{k-1}, \ell_{k-1})}, s^{(\mathbf{x}_{k-1}, \ell_{k-1})} : (\mathbf{x}_{k-1}, \ell_{k-1}) \in \mathbf{X}_{k-1}^\ell\}$$

with $r^{(\mathbf{x}_{k-1}, \ell_{k-1})} = p_{S,k}(\mathbf{x}_{k-1}, \ell_{k-1})$ and $s^{(\mathbf{x}_{k-1}, \ell_{k-1})}(\cdot) = \pi_{k|k-1}(\cdot | \mathbf{x}_{k-1}, \ell_{k-1})$.

The labeling function $\alpha : \mathbf{X}_{k-1}^\ell \rightarrow \mathbb{L}_{0:k-1}$ is defined by $\alpha(\mathbf{x}_{k-1}, \ell_{k-1}) = \ell_{k-1}$.

Remark. Note that the label of the target is preserved in the transition, only the kinematic part of state changes.

Hence \mathbf{X}_S^ℓ , the set of surviving targets at time k , is distributed according to:

$$\Pi_S(\mathbf{X}_S^\ell | \mathbf{X}_{k-1}^\ell) = \Delta(\mathbf{X}_S^\ell) 1_{\mathcal{L}(\mathbf{X}_{k-1}^\ell)}(\mathcal{L}(\mathbf{X}_S^\ell)) \left[\Phi_{k|k-1}(\mathbf{X}_S^\ell; \cdot) \right]^{\mathbf{X}_{k-1}^\ell}, \quad (4.43)$$

where,

$$\begin{aligned}
\Phi_{k|k-1}(\mathbf{X}_S^\ell; \mathbf{x}_{k-1}, \ell_{k-1}) &= \sum_{(\mathbf{x}_k, \ell_k) \in \mathbf{X}_S^\ell} \delta_{\ell_{k-1}}(\ell_k) p_{S,k}(\mathbf{x}_{k-1}, \ell_{k-1}) \pi_{k|k-1}(\mathbf{x}_k | \mathbf{x}_{k-1}, \ell_{k-1}) \\
&\quad + \left(1 - 1_{\mathcal{L}(\mathbf{X}_S^\ell)}(\ell_{k-1})\right) q_{S,k}(\mathbf{x}_{k-1}, \ell_{k-1}) \\
&= \begin{cases} p_{S,k}(\mathbf{x}_{k-1}, \ell_{k-1}) \pi_{k|k-1}(\mathbf{x}_k | \mathbf{x}_{k-1}, \ell_{k-1}), & \text{if } (\mathbf{x}_{k-1}, \ell_{k-1}) \in \mathbf{X}_S^\ell; \\ q_{S,k}(\mathbf{x}_{k-1}, \ell_{k-1}), & \text{if } \ell_{k-1} \notin \mathcal{L}(\mathbf{X}_S^\ell). \end{cases}
\end{aligned}$$

In addition, the set $\mathbf{X}_B^\ell \subset \mathbf{X}_k^\ell$ of new born targets, assumed independent of the surviving targets, can be modeled by a labeled RFS with label space \mathbb{B}_k . We consider the birth density to be of the following form:

$$\boxed{\Pi_B(\mathbf{X}_B^\ell) = \Delta(\mathbf{X}_B^\ell) w_B(\mathcal{L}(\mathbf{X}_B^\ell)) [\Phi_{B,k}(\cdot)]^{\mathbf{X}_B^\ell}.} \quad (4.44)$$

This birth model (4.44) covers both:

1. LABELED POISSON RFS with intensity function $\psi_k(\cdot) = \mu_b b_{k|k-1}(\cdot)$:

$$\begin{aligned}
w_B(\mathcal{L}(\mathbf{X}_B^\ell)) &= \frac{e^{-\lambda} \lambda^n}{n!} \text{ with } \lambda \triangleq \int \psi_k(x) dx \text{ and } n \triangleq |\mathbf{X}_B^\ell|, \\
\Phi_{B,k}(x, \ell) &= \frac{\psi_k(x)}{\int \psi_k(x) dx}.
\end{aligned}$$

Hence, the birth density is given by:

$$\boxed{\Pi_B(\mathbf{X}_B^\ell) = \delta_{\mathbb{B}_k(|\mathbf{X}_B^\ell|)}(L(\mathbf{X}_B^\ell)) \frac{e^{-\int \psi_k(x) dx}}{|\mathbf{X}_B^\ell|!} \prod_{(x, \ell) \in \mathbf{X}_B^\ell} \psi_k(x).} \quad (4.45)$$

2. LABELED MULTI-BERNOULLI RFS with the multi-Bernoulli parameters $\{p_{B,k}(l), b_{k|k-1}^{(l)}(\cdot)\}_{l \in \mathbb{B}_k}$ (section 3.2)

$$\begin{aligned}
w_B(\mathcal{L}(\mathbf{X}_B^\ell)) &= \prod_{l \in \mathbb{B}_k} (1 - p_{B,k}(l)) \prod_{\ell \in \mathcal{L}(\mathbf{X}_B^\ell)} \frac{1_{\mathbb{B}_k}(\ell) p_{B,k}(\ell)}{1 - p_{B,k}(\ell)}, \\
\Phi_{B,k}(x, \ell) &= b_{k|k-1}^{(\ell)}(x).
\end{aligned}$$

Hence, the birth density is given by:

$$\boxed{\Pi_B(\mathbf{X}_B^\ell) = \delta_{\mathbb{B}_k(|\mathbf{X}_B^\ell|)}(L(\mathbf{X}_B^\ell)) \prod_{l \in \mathbb{B}_k} (1 - p_{B,k}(l)) \prod_{\ell \in \mathcal{L}(\mathbf{X}_B^\ell)} \frac{p_{B,k}(\ell) b_{k|k-1}^{(\ell)}(x)}{1 - p_{B,k}(\ell)}.} \quad (4.46)$$

Thus, given a realization \mathbf{X}_{k-1}^ℓ of the multi-target state at time $k-1$, the multi-target state \mathbf{X}_k^ℓ at time k is given by the superposition of the sets of surviving and new born targets $\mathbf{X}_k^\ell = \mathbf{X}_S^\ell \cup \mathbf{X}_B^\ell$. The LABELED RFS MULTI-TARGET TRANSITION KERNEL is the convolution:

$$\Pi_{k|k-1}(\mathbf{X}_k^\ell | \mathbf{X}_{k-1}^\ell) = \sum_{\mathbf{X}_S^\ell \subseteq \mathbf{X}_k^\ell} \Pi_S(\mathbf{X}_S^\ell | \mathbf{X}_{k-1}^\ell) \Pi_B(\mathbf{X}_k^\ell \setminus \mathbf{X}_S^\ell). \quad (4.47)$$

The labeled RFS multi-target transition equation (4.47) incorporates target motion, birth and death. *Remark.* Note that since the label space \mathbb{B}_k of the birth targets and the label space $\mathbb{L}_{0:k-1}$ of the surviving targets are mutually exclusive, the superposition of both sets of targets is indeed a labeled RFS.

Proposition 4.5.2. *If the prior distribution is a generalized labeled multi-Bernoulli RFS then under the labeled RFS multi-target transition kernel (4.47), the predicted distribution is also a generalized labeled multi-Bernoulli RFS:*

$$\Pi_{k|k-1}(\mathbf{X}_k^\ell | \mathbf{X}_{k-1}^\ell) = \Delta(\mathbf{X}_k^\ell) \sum_{c \in \mathbb{C}} w_k^{(c)}(\mathcal{L}(\mathbf{X}_k^\ell)) \left[p_k^{(c)}(\cdot) \right]^{\mathbf{X}_k^\ell}, \quad (4.48)$$

where,

- the weight $w_k^{(c)}(\mathcal{L}(\mathbf{X}_k^\ell))$ is the product of the weight $w_B(\mathcal{L}(\mathbf{X}_B^\ell))$ of the birth labels and the weight $w_S^{(c)}(\mathcal{L}(\mathbf{X}_S^\ell))$ of the surviving labels, i.e.

$$w_k^{(c)}(\mathcal{L}(\mathbf{X}_k^\ell)) = w_B(\mathcal{L}(\mathbf{X}_B^\ell)) \cdot w_S^{(c)}(\mathcal{L}(\mathbf{X}_S^\ell)),$$

where

$$w_S^{(c)}(L_S) = \left[\eta_S^{(c)}(\cdot) \right]^{L_S} \sum_{I \subseteq \mathcal{L}(\mathbf{X}_{k-1}^\ell)} 1_I(L_S) w_{k-1}^{(c)}(I) \left[q_S^{(c)}(\cdot) \right]^{I \setminus L_S},$$

with

$$q_S^{(c)}(\ell) = \int q_{S,k}(\mathbf{x}_{k-1}, \ell) p_{k-1}^{(c)}(x, \ell) dx,$$

and

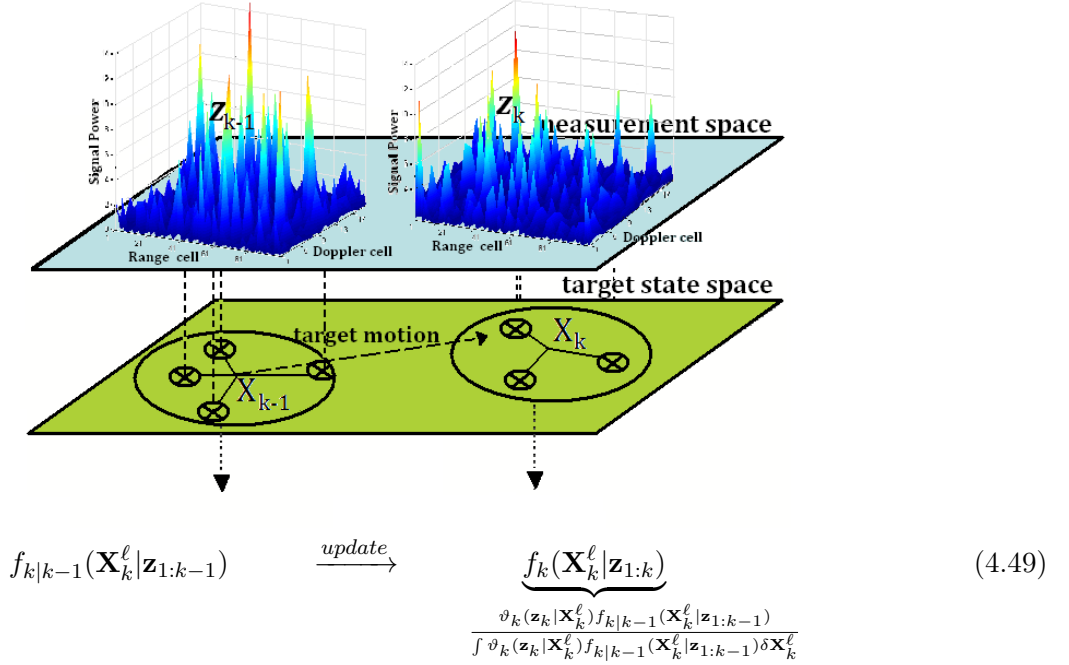
$$\eta_S^{(c)}(\ell) = \int \left(\int p_{S,k}(\mathbf{x}_{k-1}, \ell) \pi_{k|k-1}(x | \mathbf{x}_{k-1}, \ell) p_{k-1}^{(c)}(\mathbf{x}_{k-1}, \ell) d\mathbf{x}_{k-1} \right) dx.$$

The survival set weight $w_S^{(c)}(L_S)$ involves a weighted sum of the prior weights $w_{k-1}^{(c)}(\cdot)$ over all label sets that contains the surviving set L_S .

- the predicted single-target density for a given label $p_k^{(c)}(\cdot, \ell)$ is either the density $\Phi_{B,k}(\cdot, \ell)$ of a newly born target or the density of a surviving target computed from the prior density $p_{k-1}^{(c)}(\cdot, \ell)$ via the single-target prediction with transition $\pi_{k|k-1}(\cdot | \cdot, \ell)$ weighted by the probability of survival $p_{S,k}(\cdot, \ell)$, i.e.

$$\begin{aligned} p_k^{(c)}(x, \ell) &= 1_{\mathcal{L}(\mathbf{X}_S^\ell)}(\ell) \Phi_{S,k}^{(c)}(x, \ell) + 1_{\mathcal{L}(\mathbf{X}_B^\ell)}(\ell) \Phi_{B,k}(\cdot, \ell), \\ \Phi_{S,k}^{(c)}(x, \ell) &= \frac{\int p_{S,k}(y, \ell) \pi_{k|k-1}(x | y, \ell) p_{k-1}^{(c)}(y, \ell) dy}{\eta_S^{(c)}(\ell)}. \end{aligned}$$

Update



For the specific TBD surveillance application, the labeled multi-target likelihood, i.e. the probability density of the observation \mathbf{z}_k conditioned on the labeled multi-target state \mathbf{X}_k^ℓ , is used directly to update each target. Note that:

$$\boxed{\vartheta_k(\mathbf{z}_k | \mathbf{X}_k^\ell) = \vartheta_k(\mathbf{z}_k | \mathbf{X}_k).} \quad (4.50)$$

4.5.4 Particle MCHT Implementation

The Multiple Cardinality Hypotheses Tracker (MCHT) is implemented via the previously derived closed form solution. The (continuous) kinematic component of the target state, and hence the individual target tracks, are represented as sets of particles which are predicted and updated using the IP-MCMC-PF filter. The labels of the target state uniquely identify target tracks. The multi-target prediction and update steps are then calculated on component by component basis, using Munkres algorithm [33] to calculate newly predicted and updated components in descending order of probability of occurrence. For each existing component, the number of new components calculated and stored in each forward propagation is set to be proportional to the weight of the original component, subject to each cardinality retaining a minimum of C_{min} terms. The resultant posterior at each time step is then further truncated to a maximum of $C_{tot,max}$ terms. Components with weights below a predefined threshold are discarded. Let \mathfrak{T}_{k-1} denote the three-level tree-based structure containing M_{k-1} hypothesized tracks with their probability of occurrence. Thus, the MCHPF can be summarized as Algorithm 9.

Algorithm 9: MCHPF algorithm

input : \mathfrak{T}_{k-1} and a new measurement, \mathbf{z}_k **output:** \mathfrak{T}_k 1 - *Generate M -best ranked hypotheses using Munkres algorithm:*

Set up a cost matrix defining the probability of occurrence at time k of new target configurations (cardinality, and label component) given the previous hypothesized configurations, based on the previous posterior density, the survival probability, the previous existence probability. Apply Munkres algorithm to find out the M -best ranked hypotheses H_m .

2 - *Prediction Step:***for** $m \leftarrow 1$ **to** M **do** **for** $i \leftarrow 1$ **to** N **do**

 Sample $\mathbf{X}_{k,m}^\ell$ from $\mathbf{X}_{k-1,m}^\ell$ according to the model evolution described by the hypothesis H_m see eq.(4.47).

end**end**3 - *Update Step:*

Run M IP-MCMC-PF algorithms (as given in Algorithm 10 below) in parallel; one for each of the M -best ranked hypothesized tracks

$$\{\mathbf{X}_{k,m}^\ell\}_{i=1}^N = \text{IP-MCMC-PF}\left(\{\mathbf{X}_{k,m}^\ell\}_{i=1}^N, \mathbf{z}_k\right).$$

Update the existence probabilities associated to each hypothesized tracks.

4 - *Pruning Step:*

Discard the hypothesized tracks with existence probabilities below a threshold τ_P , that is, M_k hypothesized tracks are kept at the next time step.

Algorithm 10: IP-MCMC-PF algorithm

input : $\{\mathbf{X}_k^{(i)}\}_{i=1}^N$ and a new measurement, \mathbf{z}_k .

output: $\{\mathbf{X}_k^{(i)}\}_{i=1}^N$.

1 - *Initialize the N_{MCMC} M-H samplers:*

Select a random subset $\{\mathbf{X}_s^{(0)}\}_{s=1}^{N_{\text{MCMC}}} \subset \{\mathbf{X}_k^{(i)}\}_{i=1}^N$.

Compute the associated joint log-likelihood $\ell_s^{(0)} = \log \vartheta_k(\mathbf{z}_k | \mathbf{X}_s^{(0)})$.

Store the seeds $\{\mathbf{X}_s^{(0)}; \ell_s^{(0)}\}_{s=1}^{N_{\text{MCMC}}}$ in a cache.

2 - *Apply random-walk M-H sampling procedure:*

Run $s = 1, \dots, N_{\text{MCMC}}$ random-walk M-H samplers (Algorithm 11) in parallel

$$\{\mathbf{X}_s^{(l)}; \ell_s^{(l)}\}_{l=1}^{N_{\text{M-H}}} = \text{RW-MCMC}\left(\{\mathbf{X}_s^{(0)}; \ell_s^{(0)}\}, \{\mathbf{X}_k^{(i)}\}_{i=1}^N\right).$$

3 - *Check convergence:*

Compute $\hat{\mathbf{R}}$: cf. Appendix C.2.3 eq. C.32.

if $\hat{\mathbf{R}} \geq 1.1$ **then**

 Run the chains out longer to improve convergence to the stationary distribution and increase the precision of inferences.

else

 Mix all the N_{MCMC} sets of simulations together to obtain $\{\mathbf{X}_k^{(i)}; \ell_k^{(k)}\}_{i=1}^N$.

end

Algorithm 11: Random Walk M-H sampling procedure

input : $\{\mathbf{X}_s^{(0)}; \ell_s^{(0)}\}$ initial configuration and $\{\mathbf{X}_k^{(i)}\}_{i=1}^N$.**output**: $\{\mathbf{X}_s^{(l)}; \ell_s^{(l)}\}_{l=1}^{N_{\text{M-H}}}$ with $N_{\text{M-H}} := N/N_{\text{MCMC}}$.**for** $l \leftarrow 1$ **to** $B + N_{\text{M-H}}$ **do**1 - *Propose a new configuration* \mathbf{X}_s^* :Randomly select a partition $\mathbf{x}_{k-1,j}^{(i)}$ out of $\{\mathbf{X}_{k-1}^{(i)}\}_{i=1}^N$.Given $\mathbf{x}_{k-1,j}^{(i)}$, draw $\mathbf{x}_{s,j}^* = \mathbf{x}_{k-1,j}^{(i)} + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma_{RW}^2)$.Propose new configuration \mathbf{X}_s^* , such that $\mathbf{X}_s^* = \{\mathbf{x}_{s,1}^{(l-1)}, \dots, \mathbf{x}_{s,j-1}^{(l-1)}, \mathbf{x}_{s,j}^*, \mathbf{x}_{s,j+1}^{(l-1)}, \dots, \mathbf{x}_{s,n_k}^{(l-1)}\}$.Compute the joint log-likelihood $\ell_s^* = \ell_s^{(l-1)} - \mathbf{p}\ell_{s,j}^{(l-1)} + \mathbf{p}\ell_{s,j}^*$.2 - *Accept/Reject*:Sample $u \sim \mathcal{U}_{[0,1]}$, $\mathcal{U}_{[0,1]}$ a uniform distribution in $[0, 1]$.Calculate the M-H acceptance ratio: $\alpha = \frac{\ell_s^*}{\ell_s^{(l-1)}}$.**if** $u \leq \min(1, \alpha)$ **then**| Accept move: $\mathbf{X}_s^{(l)} = \mathbf{X}_s^*$, $\ell_s^{(l)} = \ell_s^*$,**else**| Reject move: $\mathbf{X}_s^{(l)} = \mathbf{X}_s^{(l-1)}$, $\ell_s^{(l)} = \ell_s^{(l-1)}$.**end****end**Discard B initial burn-in samples to allow the M-H sampler to converge to the stationary distribution.Store the remaining samples $\{\mathbf{X}_s^{(l)}; \ell_s^{(l)}\}_{l=B+1}^{B+N_{\text{M-H}}} \rightarrow \{\mathbf{X}_s^{(l)}; \ell_s^{(l)}\}_{l=1}^{N_{\text{M-H}}}$.

4.6 System Setup

Having described practical SMC implementations for multi-target tracking, we now tailor these solutions to our specific application: Radar surveillance. This requires making choices regarding the models involved as well as a number of computational and practical issues.

Specifically, subsection 4.6.1 describes two commonly used dynamical models, i.e. the Nearly Constant Velocity (NCV) model that describes the straight-line motion and the Nearly Coordinated Turn (NCT) model that deals with the maneuvers of the target. In addition, subsection 4.6.2 proposes an extension of the standard observation model for TBD applications and the corresponding likelihood function to accommodate Range/Doppler ambiguities and eclipsing effects. The robustness and computational efficiency of the algorithm are improved by using measurement gating, in subsection 4.6.3, and applying Genetic Algorithms (GAs), in subsection 4.6.4.

4.6.1 Dynamical Models

The performance of the tracking system depends for a great deal on target motion modeling [21]. In order to model the dynamic behavior of each target state $\mathbf{x}_{k,j}$, we adopt two commonly used models. The first one describes the motion of a target moving with a constant velocity without any maneuvers. The second model describes a turn performed by the target with a constant velocity.

(1) Nearly constant velocity model

The single target state is represented by the state vector $\mathbf{x}_{k,j}$ comprising the position and velocity of the j^{th} target in the Cartesian plane, $\mathbf{s}_{k,j} = [x_{k,j}, \dot{x}_{k,j}, y_{k,j}, \dot{y}_{k,j}]$, and a measure of the average target return amplitude, $I_{k,j}$:

$$\mathbf{x}_{k,j} = [\mathbf{s}_{k,j}, I_{k,j}]^T. \quad (4.51)$$

The NCV model is adopted to describe target position and velocity $\mathbf{s}_{k,j}$. This model presumes that the target moves along a straight line with (nearly) constant speed. In addition the fluctuation of the target return amplitude $I_{k,j}$ is modeled as a random walk.

Remark. In general, the return amplitude will be a function of the target's distance from the sensor ($\propto \frac{1}{r_{k,j}^4}$, where $r_{k,j}$ is the range to the target). For the purpose of this discussion, the return amplitude is assumed approximately independent of the position of the target, for trajectories which are short in comparison to the distance from the sensor to the target.

The model uncertainty is handled by the process noise $\mathbf{v}_{k,j}$, which is assumed to be standard white Gaussian noise with covariance G . Under these assumptions, the dynamical model is linear Gaussian given by:

$$\mathbf{x}_{k,j} = F \mathbf{x}_{k-1,j} + \mathbf{v}_{k-1,j}, \quad (4.52)$$

with the transition matrix F :

$$F = \begin{bmatrix} F_s & 0 & 0 \\ 0 & F_s & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad F_s = \begin{bmatrix} 1 & \Delta_T \\ 0 & 1 \end{bmatrix}, \quad (4.53)$$

and the process noise covariance G :

$$G = \begin{bmatrix} \sigma_{v_x}^2 G_s & 0 & 0 \\ 0 & \sigma_{v_y}^2 G_s & 0 \\ 0 & 0 & \sigma_i^2 \Delta_T \end{bmatrix}, \quad G_s = \begin{bmatrix} \frac{\Delta_T^3}{3} & \frac{\Delta_T^2}{2} \\ \frac{\Delta_T^2}{2} & \Delta_T \end{bmatrix}, \quad (4.54)$$

with a fixed sampling period Δ_T and where the parameters σ_{v_x} , σ_{v_y} [m.s⁻²], σ_i [dB.m⁻²] denote the standard deviations of the process noise in object motion and amplitude, respectively. Furthermore, target velocities are assumed bounded. This prior information is target dependent and will be used for state initialization.

This model correctly approximates small accelerations in the object motion and small fluctuations in object intensity.

(2) Nearly coordinated turn model

When a target performs a turn, the motion is highly correlated across the tracking directions. The developed accelerations caused by the turn are incorporated in the state of the target by a turning rate $\omega_{k,j}$. Hence, the single target state is now represented by the state vector $\mathbf{x}_{k,j}$ comprising the position and velocity of the j^{th} target in the Cartesian plane, $\mathbf{s}_{k,j} = [x_{k,j}, \dot{x}_{k,j}, y_{k,j}, \dot{y}_{k,j}]$, the turn rate, $\omega_{k,j}$, and a measure of the average target return amplitude, $I_{k,j}$:

$$\mathbf{x}_{k,j} = [\mathbf{s}_{k,j}, \omega_{k,j}, I_{k,j}]^T. \quad (4.55)$$

The NCT model is adopted to describe object position and velocity $[\mathbf{s}_{k,j}, \omega_{k,j}]$ [21]. This model presumes that the target moves with (nearly) constant speed and (nearly) constant angular (turn) rate ω . In addition the fluctuation of the target return amplitude $I_{k,j}$ is modeled as a random walk. As above, the return amplitude is assumed approximately independent of the position of the target. Under these assumptions, the evolution of the state $\mathbf{x}_{k,j}$ is modeled by the stochastic process:

$$\mathbf{s}_{k,j} = F(\omega_{k-1,j}) \mathbf{s}_{k-1,j} + G \mathbf{v}_{k-1,j}, \quad (4.56)$$

$$\omega_{k,j} = \omega_{k-1,j} + \Delta_T \mathbf{u}_{k-1,j}, \quad (4.57)$$

$$I_{k,j} = I_{k-1,j} + \Delta_T \mathbf{i}_{k-1,j}, \quad (4.58)$$

where the transition matrix F is given by:

$$F(\omega) = \begin{bmatrix} 1 & \frac{\sin \omega \Delta_T}{\omega} & 0 & -\frac{1 - \cos \omega \Delta_T}{\omega} \\ 0 & \cos \omega \Delta_T & 0 & -\sin \omega \Delta_T \\ 0 & \frac{1 - \cos \omega \Delta_T}{\omega} & 1 & \frac{\sin \omega \Delta_T}{\omega} \\ 0 & \sin \omega \Delta_T & 0 & \cos \omega \Delta_T \end{bmatrix}, \quad (4.59)$$

and G , the noise gain, is given by:

$$G = \begin{bmatrix} \frac{\Delta_T^2}{2} & 0 \\ \Delta_T & 0 \\ 0 & \frac{\Delta_T^2}{2} \\ 0 & \Delta_T \end{bmatrix}, \quad (4.60)$$

with a fixed sampling period Δ_T and $\mathbf{v}_{k,j} \sim \mathcal{N}(\cdot; 0, \sigma_v^2)$, $\mathbf{u}_{k,j} \sim \mathcal{N}(\cdot; 0, \sigma_u^2)$ and $\mathbf{i}_{k,j} \sim \mathcal{N}(\cdot; 0, \sigma_i^2)$ where the parameters σ_v [m.s⁻²], σ_u [rad.s⁻¹], σ_i [dB.m⁻²] denote the standard deviations of the process noise in object motion, turn-rate and amplitude, respectively.

This model enables more accurate tracking on curved roads and better accommodates sharp turns performed by the target.

4.6.2 Ambiguities and Observation Model

At discrete instants k , the radar system positioned at the Cartesian origin records a noisy signal \mathbf{z}_k , with a sampling interval Δ_T . Each measurement \mathbf{z}_k is a three-dimensional image consisting of $m = m_R \times m_D \times m_B$ reflected intensity measurements \mathbf{z}_k^c , where m_R , m_D and m_B are the number of range, Doppler and bearing cells. Each matrix index is an ordered trio of integers $c = (c_r, c_d, c_b)$, where $1 \leq c_r \leq m_R$, $1 \leq c_d \leq m_D$ and $1 \leq c_b \leq m_B$.

A typical observation for a fixed bearing angle is shown in Figure 4.2.

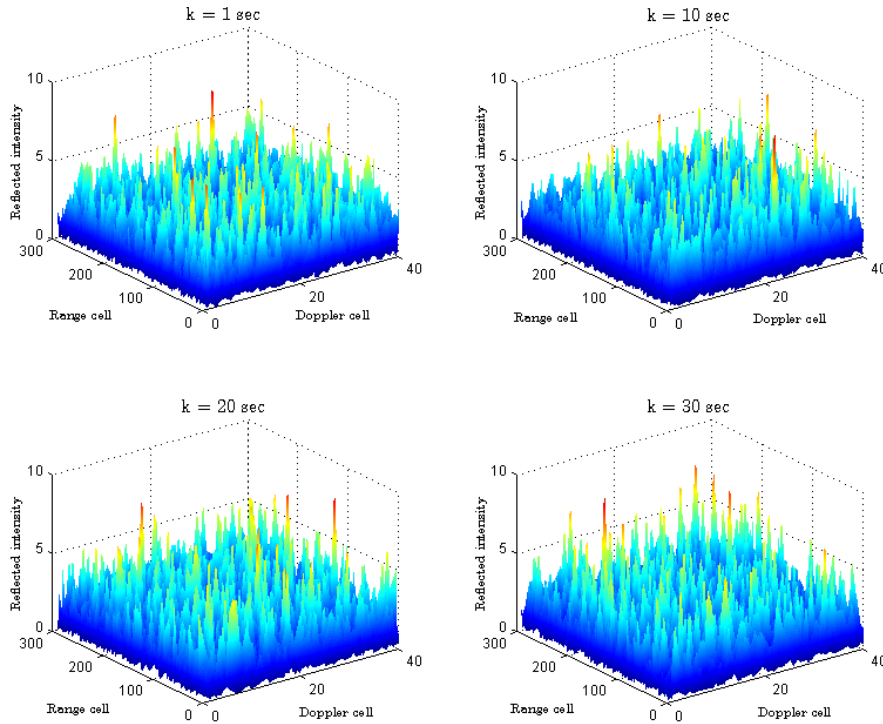


Figure 4.2: Typical measurement data.

(1) Observation Model

The likelihood expression implemented is based on the Swerling II fluctuation model [130]. The power of a target echo in one range-Doppler cell is assumed to follow the exponential distribution. Additionally, the target echo is assumed to be independent from cell to cell. These assumptions lead to a simple likelihood expression for the measured power conditioned on the state, i.e. it results in simple expressions for calculating the likelihood of each particle. The power of a target echo in one range-Doppler-bearing cell, \mathbf{z}_k^c , conditioned on the state, \mathbf{X}_k , is assumed to follow the exponential distribution:

$$\phi^c(\mathbf{z}_k^c|\mathbf{X}_k) = \frac{1}{\mu_0^c} \exp\left(-\frac{1}{\mu_0^c} \mathbf{z}_k^c\right), \quad (4.61)$$

with,

$$\mu_0^c = \mathbb{E}[\mathbf{z}_k^c] \quad (4.62)$$

$$= \sum_{j:c \in C(\mathbf{x}_{k,j})} I_{k,j} h^c(\mathbf{x}_{k,j}) + 2\sigma_w^2, \quad (4.63)$$

where σ_w^2 is the noise variance, $I_{k,j}$ is the source intensity and h^c is the point spread function. Assume h^c , the reflection form of the c^{th} target, given by:

$$h^c(\mathbf{x}_{k,j}) := \exp\left(-\frac{(c_r \Delta_R - r_{k,j})^2}{R} - \frac{(c_d \Delta_D - d_{k,j})^2}{D} - \frac{(c_b \Delta_B - b_{k,j})^2}{B}\right), \quad (4.64)$$

with,

$$r_{k,j} = \sqrt{x_{k,j}^2 + y_{k,j}^2}, \quad d_{k,j} = \frac{(x_{k,j} v_{x_{k,j}} + y_{k,j} v_{y_{k,j}})}{\sqrt{x_{k,j}^2 + y_{k,j}^2}} \text{ and } b_{k,j} = \arctan\left(\frac{y_{k,j}}{x_{k,j}}\right). \quad (4.65)$$

R , D and B are constants related to the size of a range, a Doppler and a bearing cell.

Remark. In many practical situations, this model does not capture accurately enough the true reflection characteristics. The exponential distribution is too pessimistic, i.e. it has an entropy that is too large in many applications. The assumption of independency is often incorrect, since in many applications the amplitude or even the complex echo is constant from cell to cell. In this cases, the generalized reflection model is required. The generalized reflection model is also of interest in tuning the TBD algorithm to a special type of targets, or when integrating target tracking and classification, exploiting the specific differences in target reflection characteristics as recognition features.

(2) Range/Doppler Ambiguities and Eclipsing Effects

A different modeling setup is proposed to accommodate Range/Doppler ambiguities and eclipsing effects described below.

Range Ambiguity

Target range is determined by measuring elapsed time while the pulse travels to and returns from the target as a reflected echo. In particular the range is related to the round-trip traveling time t by the speed of light c , i.e.

$$r = \frac{c t}{2}. \quad (4.66)$$

However, if the period between successive pulses is too short, an echo from a distant target may return after transmitting another pulse. This makes it impossible to state whether the observed pulse is the echo of the pulse just transmitted or the echo of the preceding pulse. This produces a situation referred to as RANGE AMBIGUITY, see Figure 4.3.

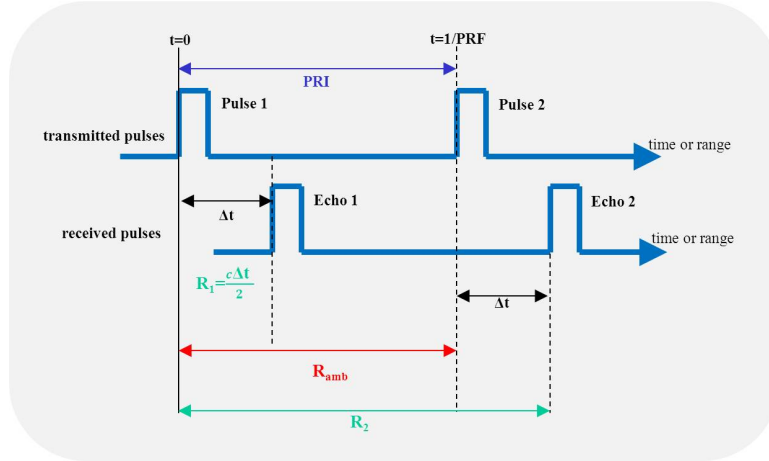


Figure 4.3: Range ambiguity illustration

Let r_{fold} denote the maximum distance RF energy can travel round trip between two consecutive pulses,

$$r_{fold} = \frac{c}{2 \text{ PRF}}. \quad (4.67)$$

It follows that the apparent range (r_{app}) is a modulo function of the true range (r_{true}),

$$r_{app} \equiv r_{true} \pmod{r_{fold}}. \quad (4.68)$$

Remark. A high Pulse Repetition Frequency [Hz] (PRF) improves resolution and range accuracy by sampling the position of the target more often. However, the greater the PRF, the shorter the maximum unambiguous range (r_{fold}).

Eclipsing in Range

The duplexer alternately switches the antenna between the transmitter and receiver. This switching is necessary because the high-power pulses of the transmitter would destroy the receiver if energy were allowed to enter the receiver. Note that any reflected pulses from close targets returning before the receiver is connected to the antenna will be undetected. This results in blind range (eclipsing) defined as:

$$r_{blind} = \frac{n_r c PRI}{2} \text{ [m] with, } n_r \in \mathbb{N}_+. \quad (4.69)$$

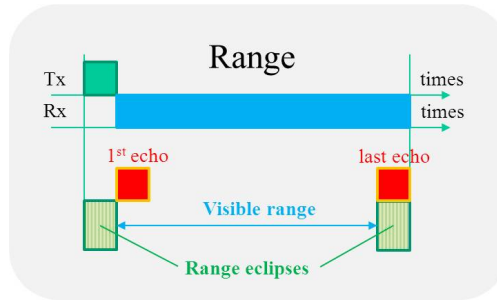


Figure 4.4: Range eclipsing illustration

Doppler Ambiguity

Doppler radars have the ability to extract target radial speed to distinguish moving from stationary targets, such as clutter, by measuring the doppler shift that a target imposes on the reflected pulse. In each pulse the frequency shift cannot be used directly, but one can measure the phase variation from pulse to pulse given by:

$$\Delta_\varphi \equiv \frac{-2\pi \dot{r}_{true}}{PRF \lambda} \pmod{2\pi}, \quad (4.70)$$

where λ is the wavelength of the transmitted energy. This introduces a modulo operation onto the apparent frequency of the reflected signal and causes reflected signals to be folded for high speed targets where radial velocity produces a frequency shift above the PRF, so that the apparent radial velocity (\dot{r}_{app}) is a modulo function of true radial velocity (\dot{r}_{true}), i.e.

$$\dot{r}_{app} \equiv \dot{r}_{true} \left(\pmod{\frac{PRF \lambda}{2}} \right). \quad (4.71)$$

Eclipsing in Doppler

In a Moving Target Indicator (MTI) Radar, a Delay-Line Canceler (DLC) filter is used to reject stationary clutter at zero frequency. This filter presents two characteristics that limit performance in tracking. First its frequency response function has zero response when a moving target has a doppler frequency at the PRF or one of its harmonics. Secondly the clutter spectrum is not a delta function of zero width, so that clutter will appear in the pass band of the DLC filter. The former property results in blind speed as the radial velocity at which the target appears stationary, that is, when the Doppler frequency lies close to the PRF. The blind speed is given by:

$$\dot{r}_{blind} = \text{PRF} \left(\frac{n \lambda}{2} \right) \text{ [m/s] with, } n \in \mathbb{N}. \quad (4.72)$$

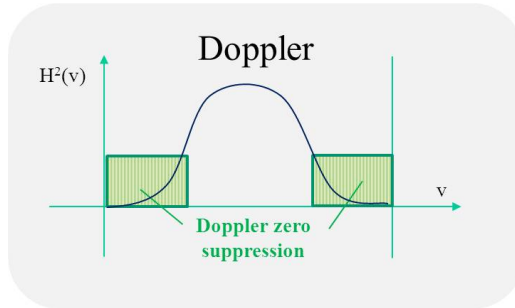


Figure 4.5: Doppler eclipsing illustration

Bearing Angle

The true bearing (referenced to true north) of a radar target is the angle between true north and a line pointed directly at the target. This angle is measured in the horizontal plane and in a clockwise direction from true north. The bearing angle to the radar target may also be measured in a clockwise direction from the centerline of your own ship or aircraft and is referred to as the relative bearing. Both true and relative bearing angles are illustrated in Figure 4.6

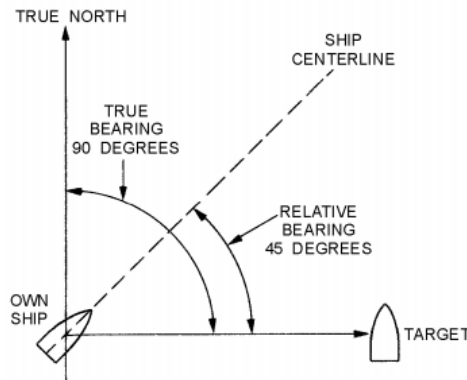


Figure 4.6: True and relative bearings

Modified Modeling Setup

A different modeling setup is proposed to deal with Range/Doppler ambiguities and eclipsing effects. First to handle ambiguities we propose the following corrected relationship between the measurement space and the target space:

$$r_{app} = \sqrt{x^2 + y^2} \left(\bmod \frac{c}{2 \text{ PRF}} \right), \quad d_{app} = \frac{(x v_x + y v_y)}{\sqrt{x^2 + y^2}} \left(\bmod \frac{\text{PRF} \lambda}{2} \right). \quad (4.73)$$

Secondly to take into account the eclipsing effects, the reflected signal of the targets lying within blind range areas is assumed null.

In addition ambiguity resolution is performed using multiple PRFs, e.g. a staggered (or jittered) PRF system [130], i.e the repetitive (or random) use of at least three different PRFs. This has the effect of changing the apparent target range estimated by each pulse, or pulse burst, and allows for some ambiguity resolution in either Doppler or range, depending on the particular application and prevents blind spots in the range and in the radial velocity coverage. In particular, the selection of the PRF directly affects the maximum fold range and Doppler respectively r_{fold} and d_{fold} , the range and Doppler resolutions associated to the Radar respectively Δ_r and Δ_d , the blind range areas width r_{ecl} and the blind velocities \dot{r}_{ecl} .

4.6.3 Measurement Gating

Multi-target tracking requires gating, or measurement selection. The purpose of gating is to reduce computational expense by eliminating from consideration measurements which are far from the predicted measurement location. Gating is performed for each track at each time step k by defining an area in the surveillance space which is called the gate. All measurements positioned within the gate are selected and used for the track update step (3.24), while measurements outside the gate are ignored in these computations. The main objective is to reduce the size of the measurement set that the tracking filter needs to process and thus to limit computational requirements. When constructing the validation gate we, on the one hand, try to accomplish that the true target position almost certainly lies within it. On the other hand, we try to keep the validation gate as small as possible, so as to restrict the amount of video data that has to be considered in the update step. To this end, a pre-selection stage is performed to identify at each scan the set of radar cells which are illuminated by at least one target. For example, we can define the gate for each of the partition as follows:

$$C(\mathbf{x}_{k,j}) = \{c \in \{1, \dots, m\} : (p_c - \bar{p}_{k,j})^T \Sigma_{j,k}^{-1} (p_c - \bar{p}_{k,j}) \leq C_0\}, \quad (4.74)$$

where p_c denotes the position of the cell c and the parameter C_0 specifies the size of the gate, which is proportional to the probability that the target falls within the gate. In our experiments, C_0 is defined as a 3-standard-deviation level gate. Measurement gating is here accomplished by using a Gaussian approximation of the predicted partition cloud, centered at the position predicted from the particle representation of $f_{k|k-1}(\mathbf{x}_{k,j} | \mathbf{z}_{1:k-1})$:

$$\bar{p}_{k,j} \approx \sum_{i=1}^N p_{k,j}^{(i)} w_{k-1}^{(i)}, \quad (4.75)$$

where the $p_{k,j}^{(i)}$ denote the position of the partition state vector

$$\mathbf{x}_{k,j}^{(i)} \sim \pi_{k|k-1}(\mathbf{x}_{k,j} | \mathbf{x}_{k-1,j}^{(i)}), \quad i = \{1, \dots, N\}. \quad (4.76)$$

Similarly, the covariance matrix is calculated as:

$$\Sigma_{j,k} \approx \frac{\sum_{i=1}^N w_{k-1}^{(i)}}{\left(\sum_{i=1}^N w_{k-1}^{(i)}\right)^2 - \sum_{i=1}^N (w_{k-1}^{(i)})^2} \sum_{i=1}^N w_{k-1}^{(i)} (p_c - p_{k,j}^{(i)})(p_c - p_{k,j}^{(i)})^T. \quad (4.77)$$

4.6.4 Genetic Algorithms

Genetic Algorithms (GAs) [78], as a stochastic searching algorithm, are widely used as a process optimizer and are developed inspired by Darwin's Theory of Evolution. A GA simulates the evolution of individuals in competing for survival. Fitter individuals cross-breed and produce better off-springs hence promoting the fitness of the whole population. Moreover, mutation also occurs during the production of off-springs. In our framework, genetic operators can be performed to generate improved particles [19, 114].

(1) Mutation operator The mutation operator generates a new particle individual \mathbf{X}^* by altering the j th partition of a given particle $\mathbf{X}^{(i)}$ with probability p_m . The new particle individual can be accepted if consistent with mutation criteria.

(2) Exchange operator The exchange operator works by switching genetic material between two parent samples (possibly from two different chain realizations) for producing offspring. The two parents $\mathbf{X}_k^{(i_1)}$ and $\mathbf{X}_k^{(i_2)}$ are manipulated as follows: for any j , the partitions $\mathbf{x}_{k,j}^{(i_1)}$ and $\mathbf{x}_{k,j}^{(i_2)}$ are swapped with probability p_x . At this point an M-H step is performed for deciding whether the resulting offspring particles will be a part of the improved population. In order to maintain an adequate approximation of the target distribution, on acceptance both new candidates should replace their parents, otherwise both parents should be retained.

(3) Crossover operator The crossover operator works by selecting and cross-breeding two parent samples (possibly from two different chain realizations) for producing offspring. The two parents $\mathbf{X}_k^{(i_1)}$ and $\mathbf{X}_k^{(i_2)}$ are mingled as follows: for any j , the partitions $\mathbf{x}_{k,j}^{(i_1)}$ and $\mathbf{x}_{k,j}^{(i_2)}$ are combined with probability p_c according to:

$$\mathbf{x}_{k,j}^* = \alpha \mathbf{x}_{k,j}^{(i_1)} + (1 - \alpha) \mathbf{x}_{k,j}^{(i_2)}, \quad (4.78)$$

$$\mathbf{x}_{k,j}^{**} = \beta \mathbf{x}_{k,j}^{(i_2)} + (1 - \beta) \mathbf{x}_{k,j}^{(i_1)}, \quad (4.79)$$

where α and β can be seen as weight coefficients, for example:

$$\alpha = \frac{\varphi(\mathbf{x}_{k,j}^{(i_1)})}{\varphi(\mathbf{x}_{k,j}^{(i_1)}) + \varphi(\mathbf{x}_{k,j}^{(i_2)})}, \quad \beta = \frac{\varphi(\mathbf{x}_{k,j}^{(i_2)})}{\varphi(\mathbf{x}_{k,j}^{(i_1)}) + \varphi(\mathbf{x}_{k,j}^{(i_2)})},$$

with $\varphi(\cdot)$ the fitness function. Each new particle individual can be accepted if consistent with crossover criteria.

4.7 Experimental Results

Demonstrations and numerical studies of the proposed implementations are now presented. In this section, we investigate through simulations the improvements provided by interacting population-based simulation over standard SMC.

4.7.1 Scenario 1: Fixed and known number of targets ($n_k = 3$)

This scenario considers a nonlinear TBD problem in which a fixed and known number of targets follow curved paths. Three targets are present throughout the entire scenario lasting $T_{max} = 40$ s. The single target state $\mathbf{x}_{k,j}$ comprises the position and velocity of the j^{th} target, $s_{k,j}$, the turn rate, $\omega_{k,j}$, and a measure of the average target return amplitude, $I_{k,j}$. To describe the dynamic of each target state $\mathbf{x}_{k,j}$, we adopt an NCT model for the target position and velocity, and a random walk model for the object amplitude $I_{k,j}$ (cf. paragraph 4.6.1), with $\Delta_T = 1$ s, $\sigma_v = 2$ m.s⁻², $\sigma_u = \pi/180$ rad.s⁻¹ and $\sigma_i = (2/3)$ dB.m⁻².

In this demonstration, the observation region covers the intervals $[1000 \ 4000]$ m in range, $[-40 \ 40]$ m.s⁻¹ in Doppler and the angles $[-\frac{\pi}{20} \ \frac{\pi}{20}]$ rad in bearing. The Radar parameters and the targets settings used in simulation are reported in Table 4.1.

Table 4.1: Parameters used in simulation

Radar parameters	Beamwidth in bearing Range-quant size Doppler-bin size	$\Delta_b = 1$ [degree] $\Delta_r = 10$ [m] $\Delta_d = 2$ [m.s ⁻¹]
Targets settings	Bearing area Range area Radial velocity area Acceleration in turn Angle of turn SNR	$b_0 \sim U[-6, 6]$ [degrees] $r_0 \sim U[2500, 2600]$ [m] $ v_0 \sim U[0, 30]$ [m.s ⁻¹] $a_0 \sim U[0.1, 0.3]$ [m.s ⁻²] $\varphi_0 = 2$ [degrees] $SNR \sim U[8, 14]$ [dB]

The true tracks on a 2-D plane are shown in Figure 4.7.1.

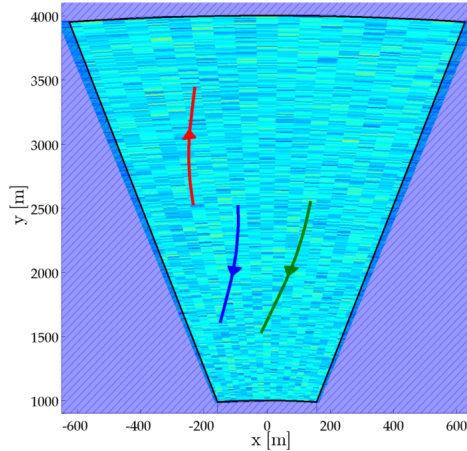


Table 4.2: Targets initial parameters

Target	SNR [dB]	v [m.s ⁻¹]
1	8	23
2	12	-26
3	11	-23

Figure 4.7: True tracks on a 2-D plane for Scenario 1.

The conventional (a) SIR-PF (Algorithm 2) and the (b) IP-MCMC-PF (Algorithm 5) are considered. Figure 4.8 reports the empirical posterior PDF and the MAP estimate over a single trial.

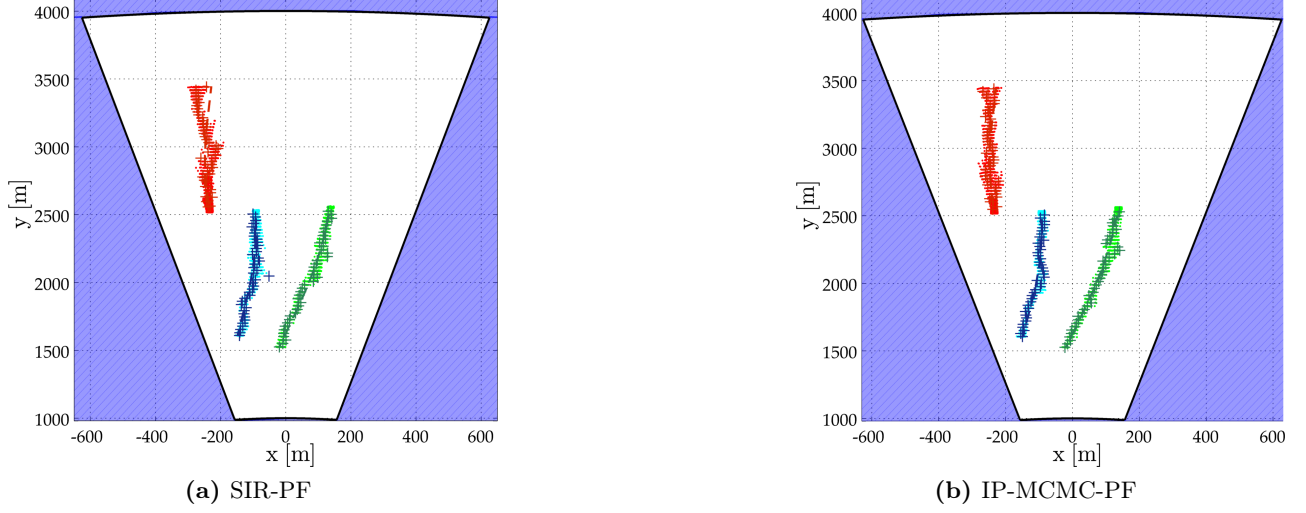


Figure 4.8: Empirical posterior distribution (.) and MAP estimate (+) over a single trial. Each filter uses $N = 500$ particles.

The tracks (the conditional means of each set of target samples) produced by the two algorithms over 100 Monte Carlo trials are shown in Figure 4.9.

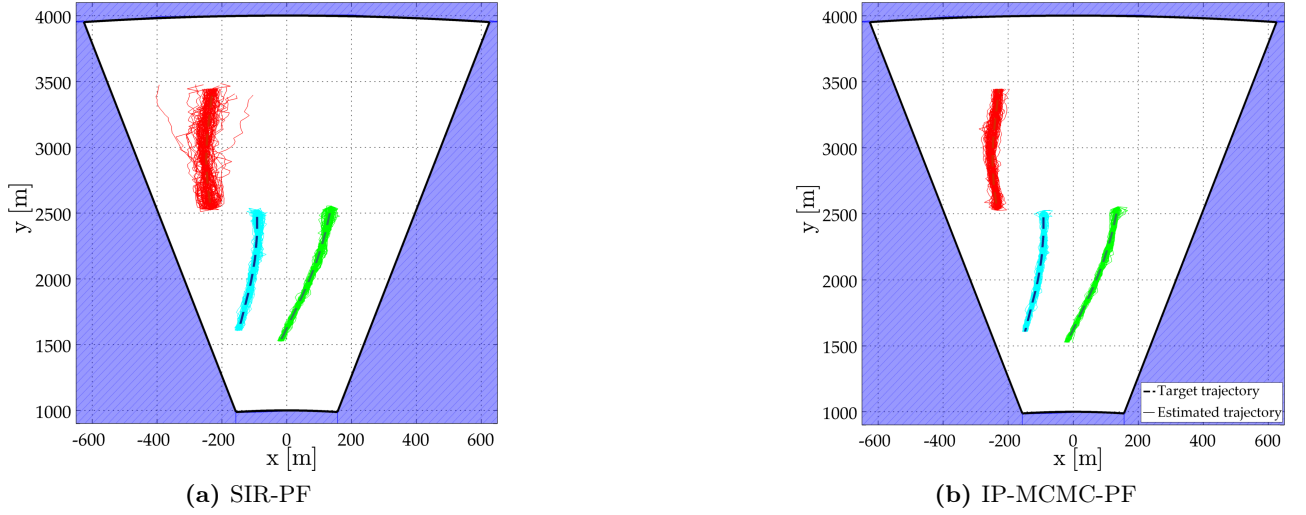


Figure 4.9: Estimated trajectories (i.e., particles-based conditional mean) over 100 Monte Carlo runs. Each filter uses $N = 500$ particles.

These figures further confirm that the proposed IP-MCMC-PF implementation provides better tracking performance than the SIR-PF in particular for weak or receding targets.

Finally the tracking accuracy of the methods is measured in terms of the Root-Mean-Square Error (RMSE). The results are reported in Figure 4.10.

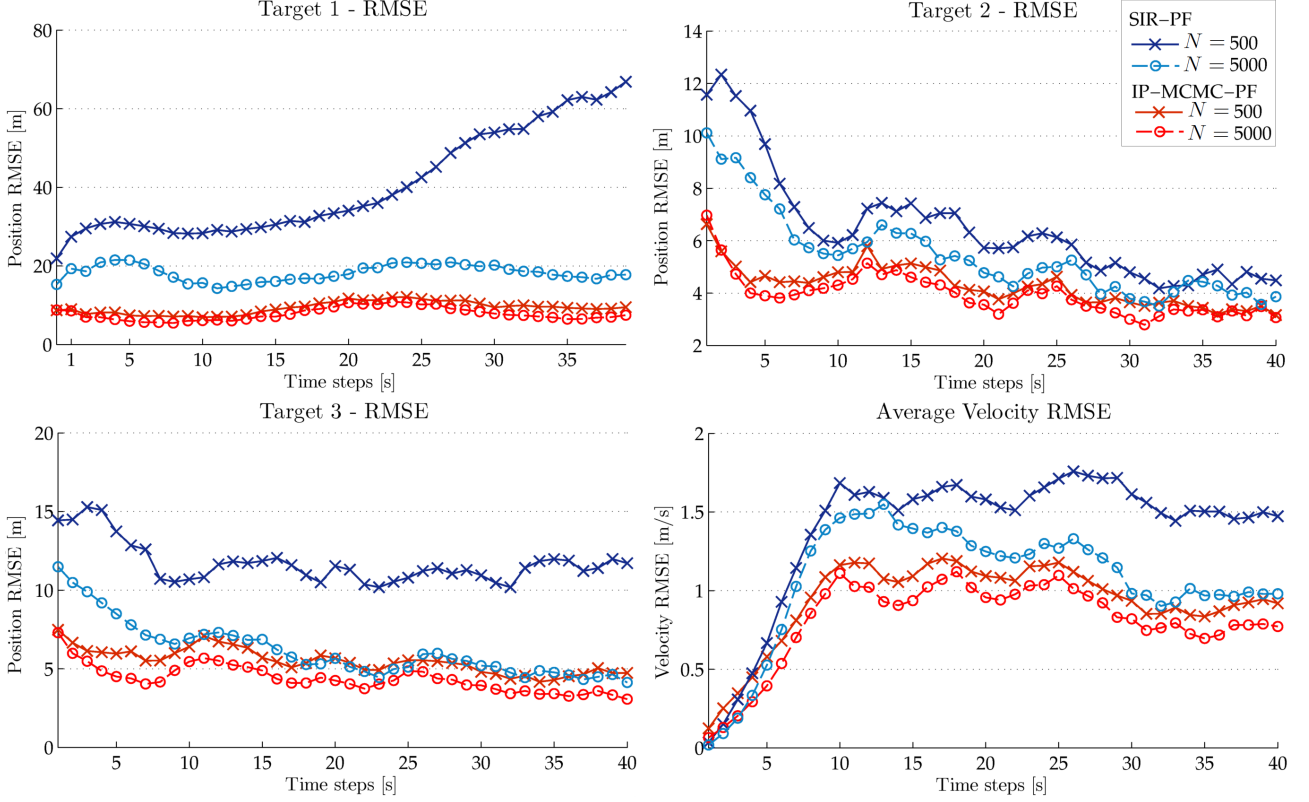


Figure 4.10: RMSE over 100 Monte Carlo trials: SIR-PF vs IP-MCMC-PF.

The proposed IP-MCMC-PF significantly outperforms the SIR-PF filter in terms of track accuracy.

4.7.2 Scenario 2: Fixed and known number of targets ($n_k = 10$)

This scenario considers a nonlinear TBD problem in which a fixed and known number of targets move along straight lines. Ten targets are present throughout the entire scenario lasting $T_{max} = 40$ s. The single target state $\mathbf{x}_{k,j}$ comprises only the position and velocity of the j^{th} target, $s_{k,j}$, and a measure of the average target return amplitude, $I_{k,j}$. To describe the dynamic of each target state $\mathbf{x}_{k,j}$, we adopt an NCV model for the object position and velocity and a random walk model for the object amplitude $I_{k,j}$, with $\Delta_T = 1$ s, $\sigma_v = 5$ m.s⁻² and $\sigma_i = (2/3)$ dB.m⁻² (cf. paragraph 4.6.1).

The radar parameters and the targets settings used in simulation are reported in Table 4.3.

Table 4.3: Parameters used in simulation

Radar parameters	Radar sampling time Beamwidth in bearing Range-quant size Doppler-bin size	$T = 1$ [sec] $B = 1$ [degree] $R = 10$ [m] $D = 2$ [m.s ⁻¹]
Targets settings	Bearing area Range area Radial velocity area Acceleration in turn Angle of turn SNR Maximum Target Speed	$b_0 \sim U[-6, 6]$ [degrees] $r_0 \sim U[2700, 2900]$ [m] $ v_0 \sim U[0, 15]$ [m.s ⁻¹] $a_0 \sim U[0, 1]$ [m.s ⁻²] $\varphi_0 \sim U[-3, 3]$ [degrees] $SNR \sim U[6, 16]$ [dB] $v_{max} = 40$ [m.s ⁻¹]

The true tracks on a 2-D plane are shown in Figure 4.7.2.

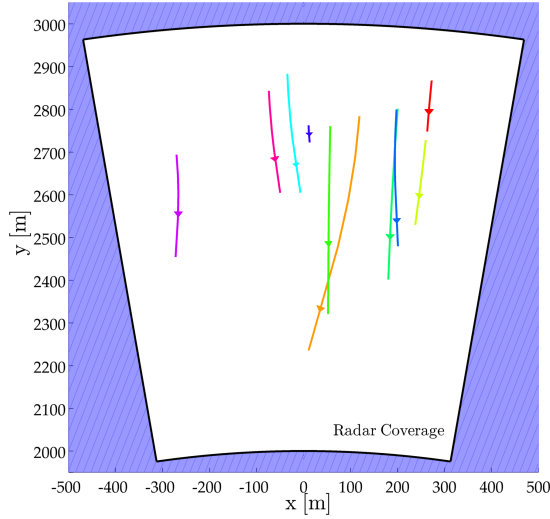


Table 4.4: Targets initial parameters

Target	SNR [dB]	v [m.s ⁻¹]
1	15	3
2	8	14
3	9	5
4	8	11
5	12	10
6	11	7
7	13	8
8	13	1
9	7	6
10	14	6

Figure 4.11: True tracks on a 2-D plane for Scenario 2.

Figure 4.12 reports the empirical posterior PDF and the MAP estimate over a single trial.

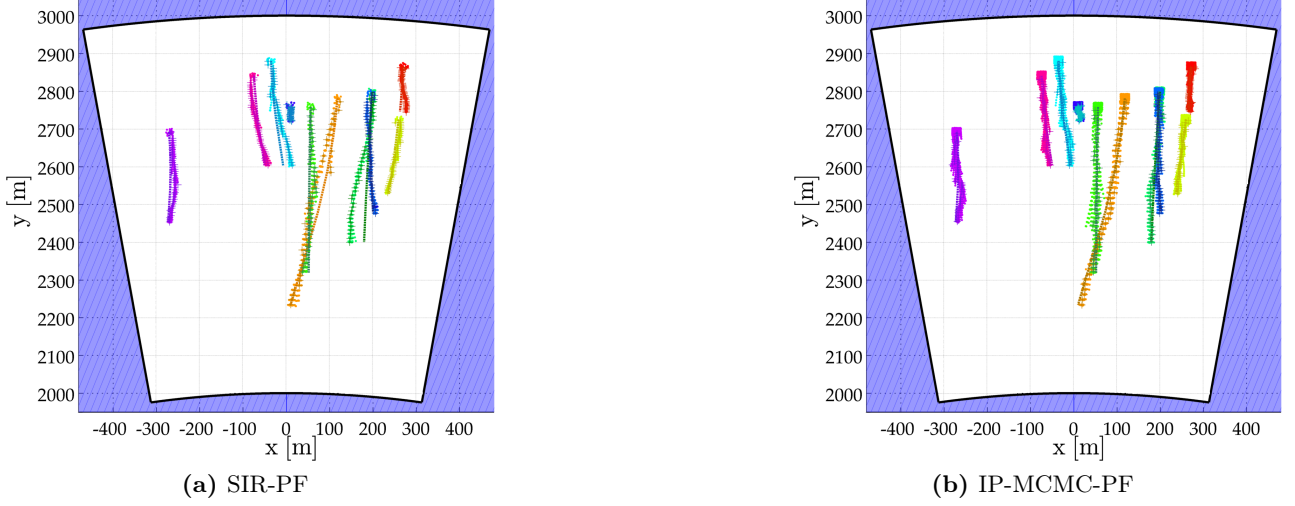


Figure 4.12: Empirical posterior distribution (.) and MAP estimate (+) over a single trial. Each filter uses $N = 1000$ particles.

The tracks produced by each algorithm over 100 Monte Carlo trials are shown in Figure 4.13.

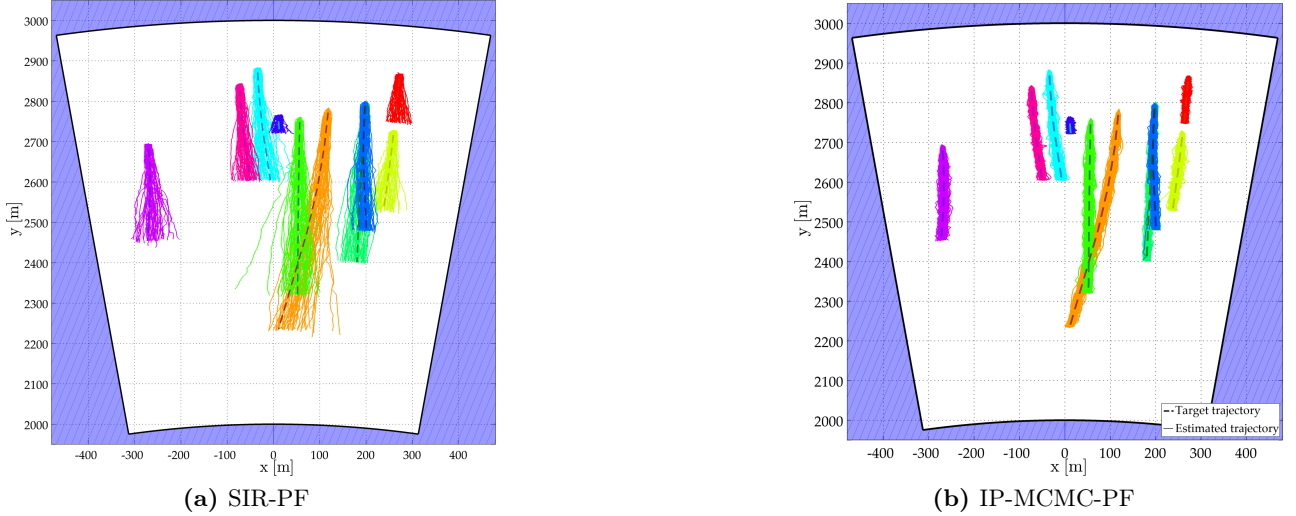


Figure 4.13: Estimated trajectories (i.e., particles-based conditional mean) over 100 Monte Carlo runs. Each filter uses $N = 1000$ particles.

We then compared the tracking accuracy, the computational cost and the robustness of the algorithms over 100 Monte-Carlo simulations. The recently developed (c) PMMH ([5] batch processing) is used as a benchmark for illustrative purposes.

Remark. The Particle Marginal Metropolis-Hastings (PMMH) algorithm aims an exact approximation of a Marginal Metropolis-Hastings (MMH) update. The implementation of the PMMH scheme requires an SMC approximation targeting $p(\mathbf{x}_{1:k}|\mathbf{z}_{1:k}, \theta)$ and the filter estimate of the marginal likelihood $p(\mathbf{z}_{1:k}|\theta)$ with $\theta \in \Theta$ some static parameter. Full details of the PMMH scheme including a proof establishing that the method leaves the full joint posterior density $p(\theta, \mathbf{x}_{1:k}|\mathbf{z}_{1:k})$ invariant can be found in [3]. Note that to obtain reasonable mixing of the resulting MCMC kernels, it was reported in [3] that a fairly high number of particles was required in the SMC scheme. Since every iteration of the PMMH scheme requires a run of a PF, a lot of computational resources are needed.

Figure 4.14 shows the target-averaged position and velocity RMSE over 100 Monte Carlo trials.

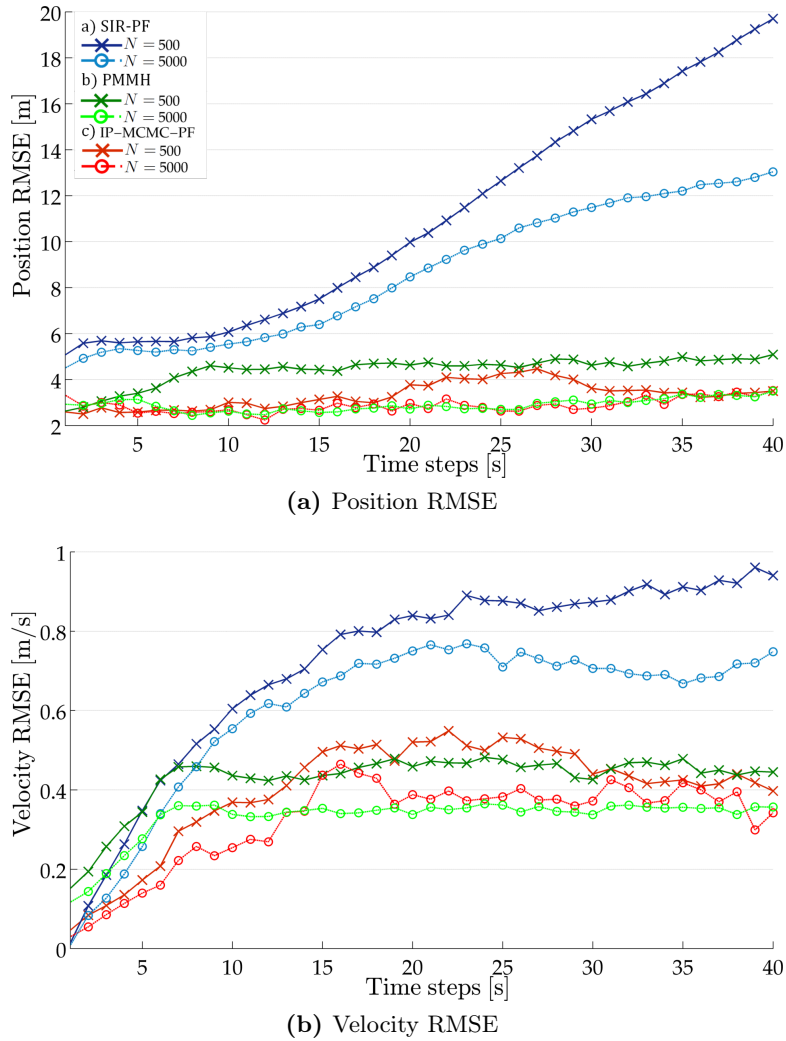


Figure 4.14: RMSE over time for the SIR-PF (■ blue), PMMH (■ green), and IP-MCMC-PF (■ red).

The performance of the methods is also compared via the average RMSE, the Tracking Loss Rate (TLR) and the average process time (avg.CPU), which are listed in Table 4.5.

Table 4.5: Performance comparison averaged over 100 MC runs

Algorithm	N_p	RMSE	TLR	avg.CPU
(a) SIR-PF	500	10.76 [m]	6%	0.4
	5000	8.44 [m]	3%	3.8
(b) PMMH	500	4.32 [m]	1%	0.5
	5000	2.82 [m]	0%	2.4
(c) IP-MCMC-PF	500	4.01 [m]	0%	0.45
	5000	2.81 [m]	0%	0.65

The Tracking Loss Rate (TLR) is defined as the ratio of the number of simulations, in which the target is lost in track (Maximum Position error $> 20[m]$), to the total number of simulations carried out. The average process time (avg.CPU) is the CPU time needed to execute one time step in MATLAB R2011b (win64) on an Intel Core i7 (Nehalem microarchitecture) operating under Windows 7 Ultimate (Version 6.1). All the important specifications and features are reported in Table 4.6.

Table 4.6: Specifications & Features

Processor	Type Model Frequency Cores	Intel(R) Core(TM) i7 920 4.2 GHz 4 (8 threads)
Memory	DRAM Type	6144 MB (3 x 2048 DDR3-SDRAM) PC3-8500 (533 MHz)

The proposed IP-MCMC-PF provides better tracking performance. This holds in terms of track accuracy and robustness to a low number of particles. In fact, sampling particles from the target posterior distribution via interacting population-based simulation avoids sample impoverishment and accelerates the MCMC convergence rate. Furthermore, the proposed algorithm matches the tracking performance of the PMMH, while allowing for a dramatic reduction of the computational time.

4.7.3 Scenario 3: Range/Doppler ambiguity and eclipsing issues

This scenario considers a nonlinear TBD problem in which a fixed and known number of targets move along straight lines. This scenario illustrates the Range/Doppler ambiguity and eclipsing issues associated with radar systems analysis and design principles. Three targets are present throughout the entire scenario lasting $T_{max} = 40$ s. The single target state $\mathbf{x}_{k,j}$ comprises only the position and velocity of the j^{th} target, $s_{k,j}$, and a measure of the average target return amplitude, $I_{k,j}$. To describe the dynamic of each target state $\mathbf{x}_{k,j}$, we adopt an NCV model for the object position and velocity and a random walk model for the object amplitude $I_{k,j}$ (cf. paragraph 4.6.1), with $\Delta_T = 1$ s, $\sigma_v = 2$ m.s⁻² and $\sigma_i = (2/3)$ dB.m⁻².

The proposed modeling setup (4.73) is used to accommodate Range/Doppler ambiguities and eclipsing effects. In addition ambiguity resolution is performed by using staggered PRF [130]. The selection of the PRF directly affects the maximum fold range and Doppler respectively r_{fold} and d_{fold} , the range and Doppler resolutions associated to the Radar respectively Δ_r and Δ_d , the blind range areas width r_{ecl} and the blind velocities \dot{r}_{ecl} . In Surveillance Mode the Radar has an instrumented maximum range of 250 km and an instrumented radial velocity domain of $[-200 + 200]$ m.s⁻¹.

The true tracks on a 2-D plane are shown in Figure 4.15.

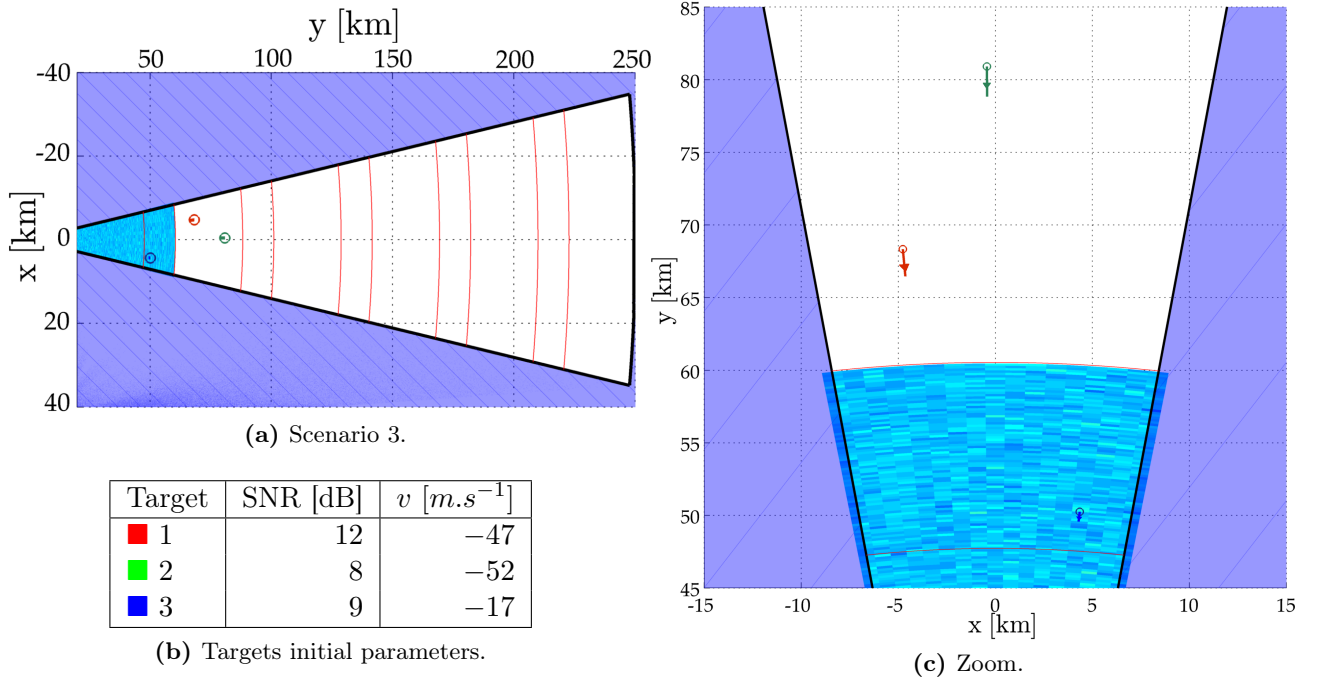


Figure 4.15: True tracks on a 2-D plane for Scenario 3.

The targets settings and the burst characteristics used in simulation are listed in Table 4.1:

Table 4.7: Parameters used in simulation

Burst characteristics	Pulse repetition frequency Carrier frequency Uncompressed pulse length Pulse compression ratio	PRF : mean 4330, min 3060, max 6250 [Hz] RF : 2.9 – 3.3 G[Hz] $\tau_{pulse} : \{0.16, 0.20, 0.24\}$ [s] PCR : 4.5 [%]
Targets settings	Bearing area Range area Radial velocity area SNR	$b_0 \sim U[-6, 6]$ [degrees] $r_0 \sim U[50, 150]$ [km] $ v_0 \sim U[10, 100]$ [m.s ⁻¹] $SNR \sim U[8, 14]$ [dB]

Given the burst characteristics, the range and doppler resolutions are given by:

$$\Delta_r = PCR \frac{\tau_{pulse} \cdot c}{2}, \quad \Delta_d = \frac{\lambda \cdot PRF}{2 \cdot n_{pulses}},$$

where $\lambda = \frac{c}{RF}$ and $n_{pulses} = \frac{1}{PRF \times \tau_{pulse}}$ correspond respectively to the wavelength of the waveform carrier (of the transmitted energy) and the number of transmitted pulses.

Figure 4.15 reports the empirical posterior PDF along with the MAP estimate produced by the IP-MCMC-PF from $k = 1$ to 6s.

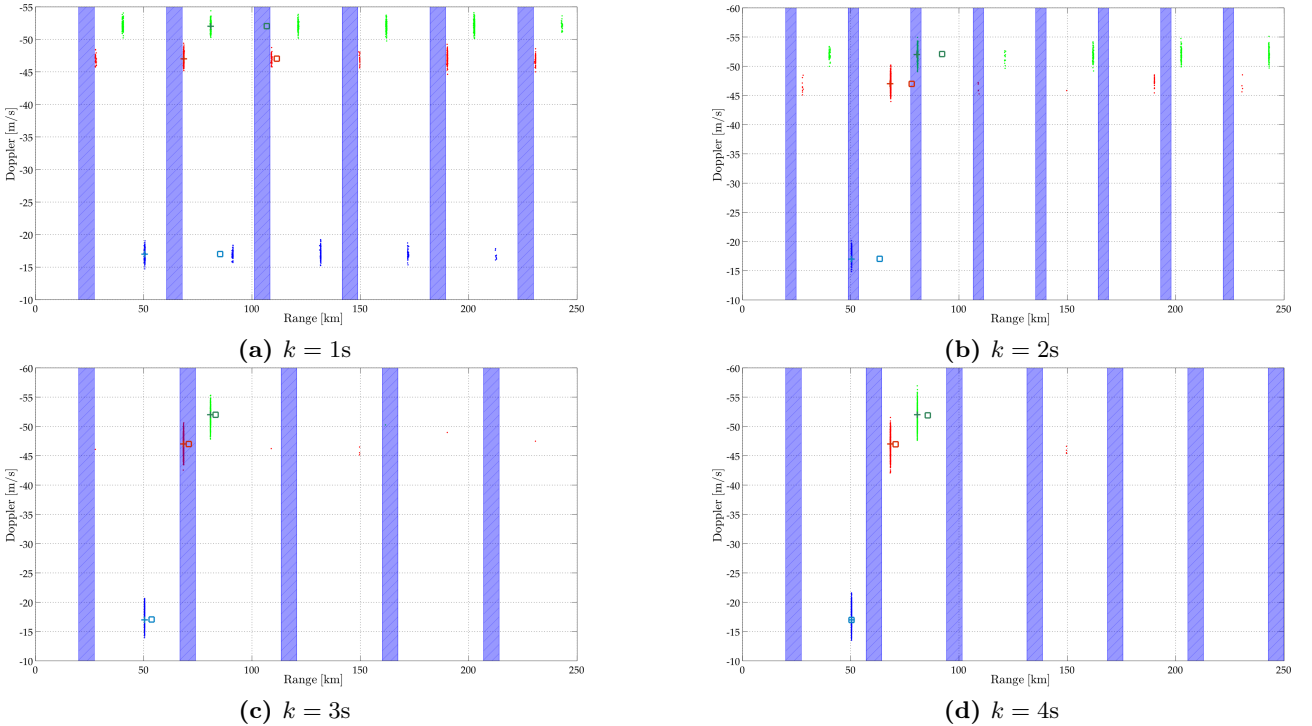


Figure 4.16: Empirical posterior distribution along MAP estimate over a single trial.

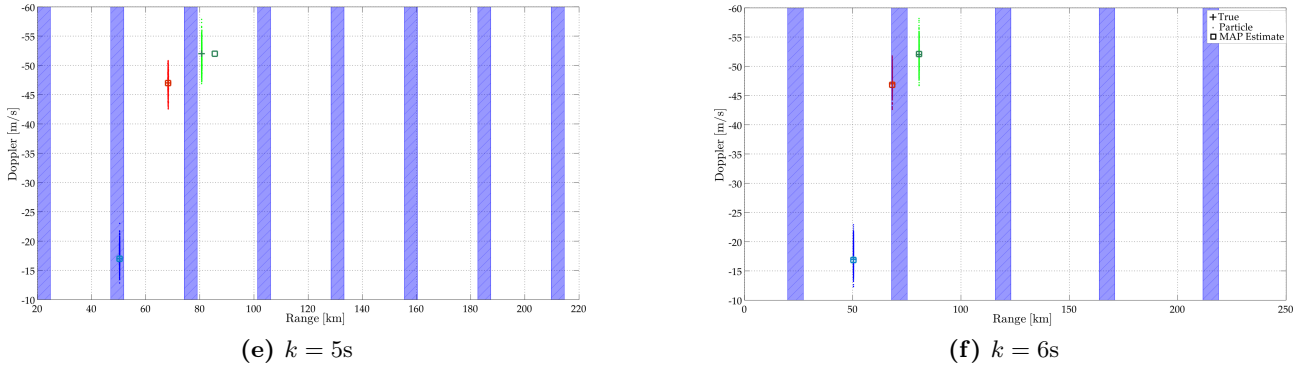


Figure 4.15: Empirical posterior distribution along MAP estimate over a single trial.

Figure 4.16 shows the target-averaged range and Doppler RMSE over 100 Monte Carlo trials.

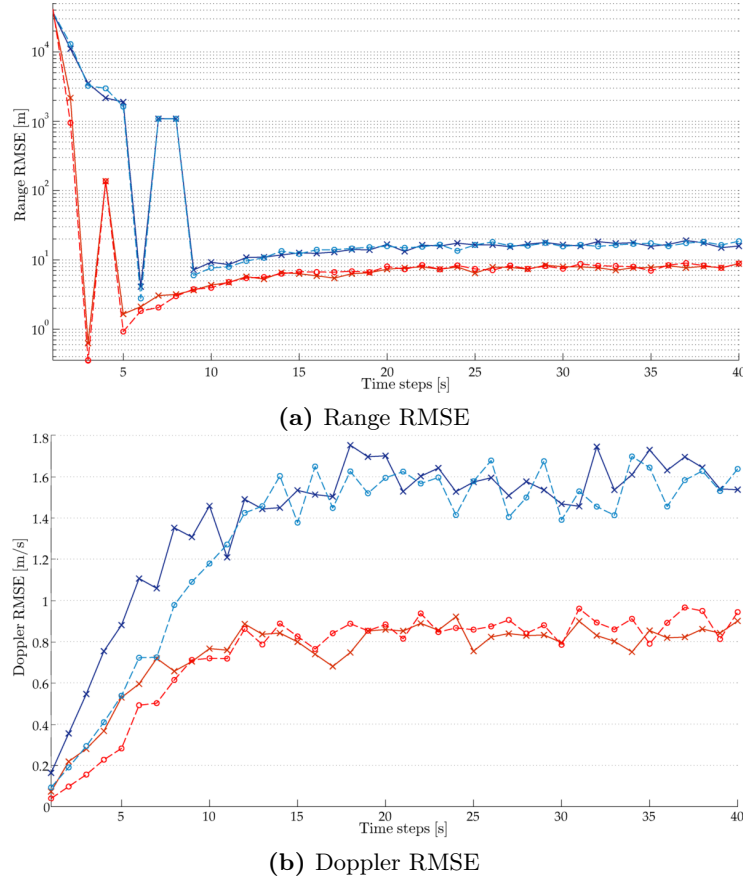


Figure 4.16: RMSE over time for the SIR-PF and the IP-MCMC-PF.

After initially having a large estimation error, both filters gradually converge towards the correct state. The proposed scheme can adapt to any type of waveform and allows us to fully benefit from the frequency agility of the radar.

4.7.4 Scenario 4: Time varying and unknown number of targets ($n_{max} = 3$)

To further demonstrate the potential of the proposed SMC implementations, this experiment includes a time varying and unknown number of targets. A maximum of 3 targets are present at any time, and there are various target births and deaths throughout the scenario duration of $T_{max} = 50$ s. Targets are initialized at separate locations. The single target state $\mathbf{x}_{k,j}$ comprises only the position and velocity of the j^{th} target, $s_{k,j}$, and a measure of the average target return amplitude, $I_{k,j}$. To describe the dynamic of each target state $\mathbf{x}_{k,j}$, we adopt an NCV model for the object position and velocity and a random walk model for the object amplitude $I_{k,j}$ (cf. paragraph 4.6.1), with $\Delta_T = 1$ s, $\sigma_v = 2$ m.s $^{-2}$ and $\sigma_i = (2/3)$ dB.m $^{-2}$.

The true tracks on a 2-D plane are depicted in Figure 4.7.4.

Table 4.8: Targets initial parameters

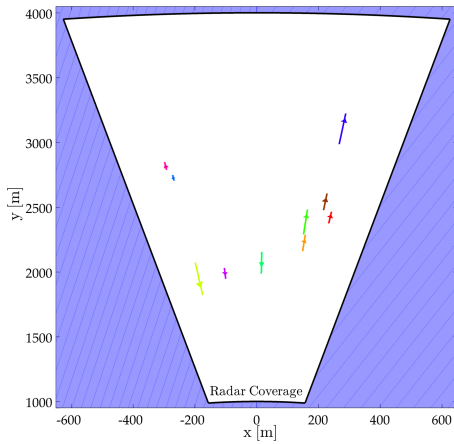


Figure 4.17: True tracks for Scenario 4.

Target	k_{birth}	k_{death}	SNR [dB]	v [m.s $^{-1}$]
1	1	8	14	13
2	2	10	10	16
3	6	17	14	-23
4	10	18	11	24
5	12	20	11	-21
6	23	28	11	26
7	23	32	11	5
8	30	44	12	17
9	30	37	9	-12
10	34	39	9	12

We compared the tracking accuracy and the track initiation/termination performances of the conventional (a) multi-target SIR-PF (Algorithm 2) and the (b) IP-RJMCMC-PF (Algorithm 6) over 100 Monte-Carlo simulations. For these comparisons, both algorithms are given the same initialization, so as to ensure a fair overall comparison.

In the case of the multi-target SIR-PF, the same dynamical model (cf. paragraph 4.6.1) is used except that in addition, births and deaths are included to accommodate a time varying number of targets. The birth process is a Multi-Bernoulli density $\pi_{\Gamma} = \{(p_{B,k}(j), b_k^{(j)})\}_{j=1}^{|I_{d_k}|}$ where $p_{B,k}(j) = 0.3$, $b_k^{(j)}(x) = \mathcal{N}(x; m_{\gamma}^{(j)}, P_{\gamma})$ with $m_{\gamma}^{(j)}$ the estimated state of the j^{th} detection plot \mathbf{x}_{k,d_j} and $P_{\gamma} = \text{diag}([5, 0.1, 5, 0.1, 1.2]^T)^2$. The death process is specified by a uniform single target probability of target survival $p_{S,k}(\mathbf{x}_{k,j}) = 0.9$. For the IP-RJMCMC-PF implementation, the move specific probabilities p_b and p_d are set to 0.3 and 0.4 respectively.

State estimation is performed with the following procedure. A target is declared present if the estimated probability of existence is greater than 0.5, otherwise no target is declared. If a target is declared, the state estimate is given by the mean of the posterior state distribution, otherwise if no target is declared, there is no state estimate.

The tracks produced by each algorithm over 100 Monte Carlo trials are shown in Figure 4.18.

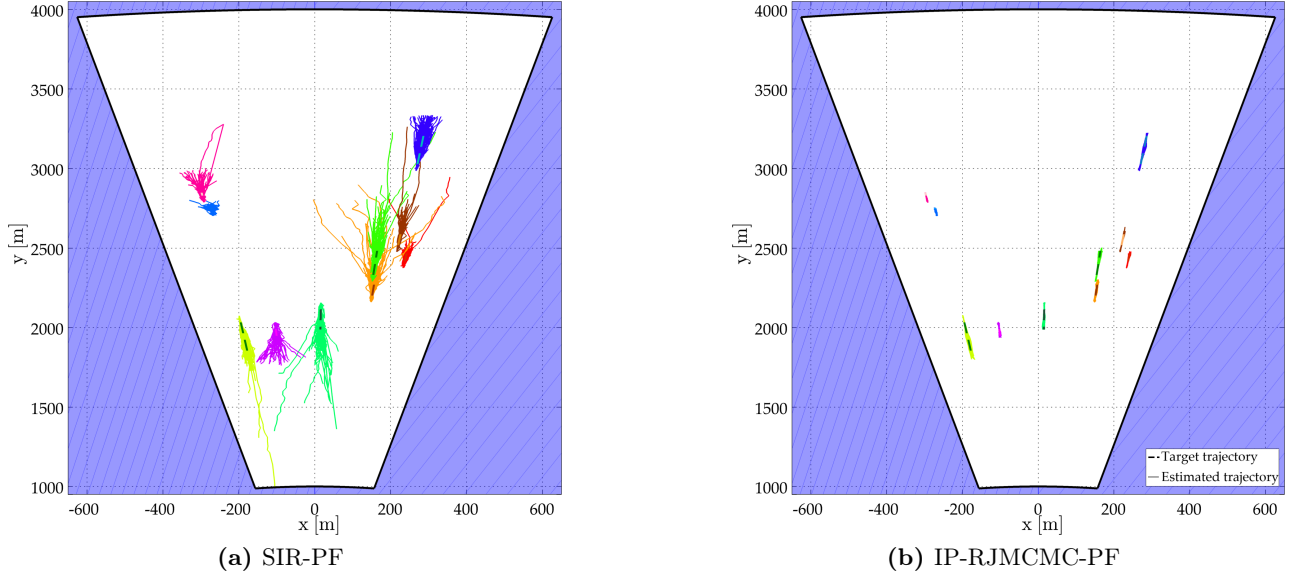


Figure 4.18: Estimated trajectories (i.e., particles-based conditional mean) over 100 Monte Carlo runs. Each filter uses $N = 500$ particles.

The proposed IP-RJMCMC-PF achieves better accuracy as well as greater consistency in tracking than the conventional SIR-PF filter.

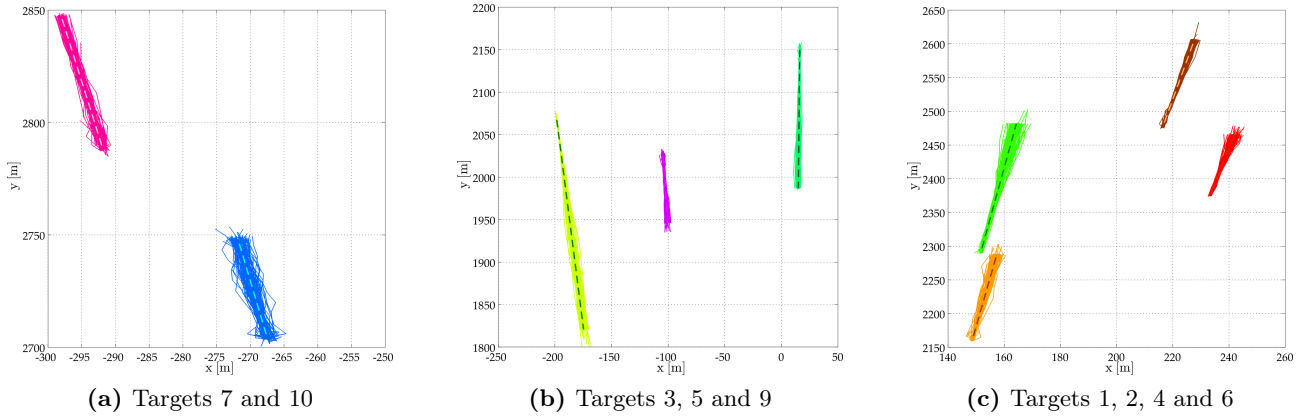


Figure 4.19: Zoom on tracks produced by the IP-RJMCMC-PF filter.

The IP-RJMCMC-PF filter produces accurate estimates of the individual target states and the total target numbers, with very few dropped or false tracks.

Figure 4.20 compares the true and estimated cardinality for the two filters versus time. Figure 4.21 reports the mean and standard deviation of the estimated cardinality distribution versus time for both filters.

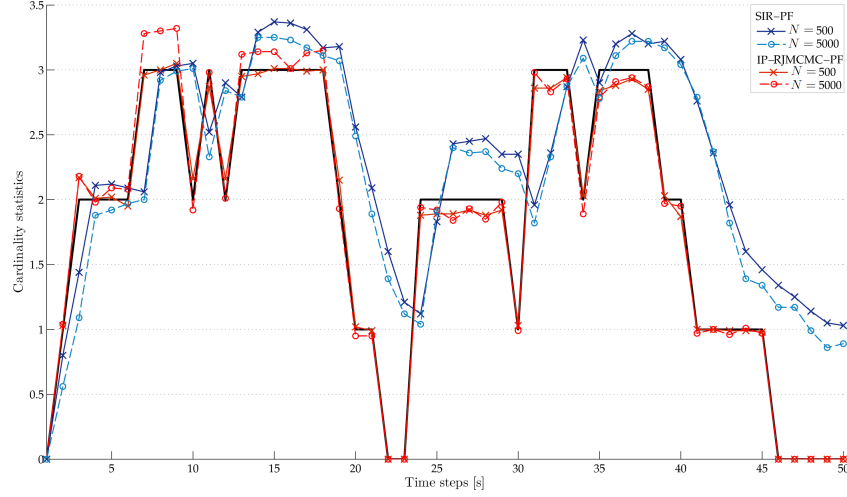


Figure 4.20: Cardinality statistics (mean) versus time. The IP-RJMCMC-PF filter initiates and terminates tracks with a very short delay.

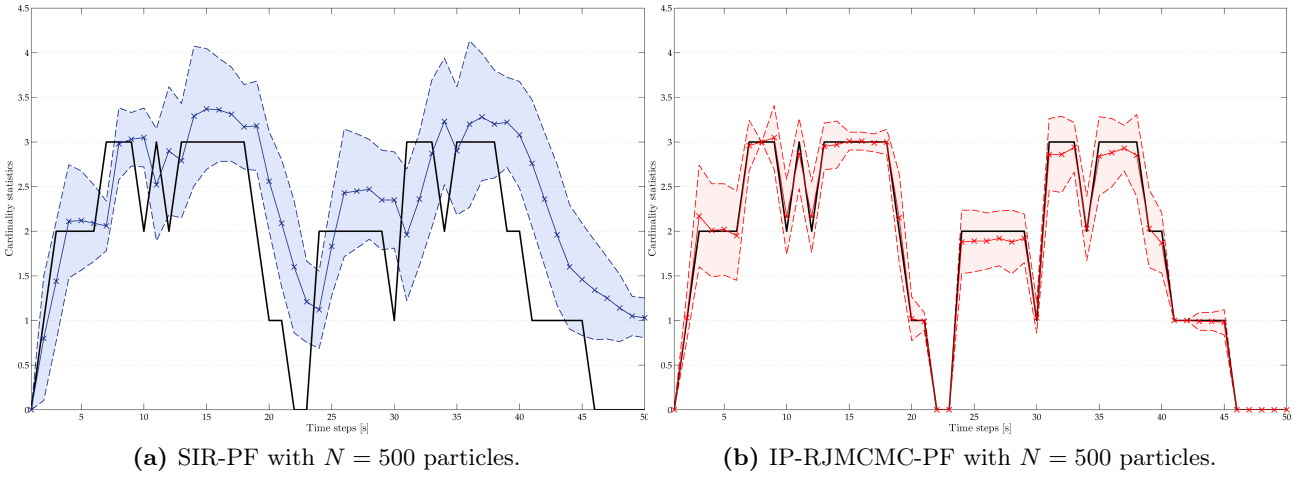


Figure 4.21: Cardinality statistics (mean and standard deviation) versus time.

It can be seen that the multi-target SIR filter initiates and terminates the tracks with some delay, while the IP-RJMCMC-PF filter does this almost instantaneously, and that the state estimates from the IP-RJMCMC-PF filter are significantly improved over that of the multi-target SIR-PF filter.

Finally, the filters performance are evaluated using the Optimal Sub-Pattern Assignment (OSPA) multi-target miss-distance, which jointly captures cardinality errors and state errors meaningfully. The OSPA metric is computed using the Hungarian algorithm for optimal point assignment. For further details see [125]. Figure 4.22 compares the OSPA penalty ($p = 2$, $c = 100$ m) for the two filters, further confirming that the IP-RJMCMC-PF filter generally outperforms the multi-target SIR-PF filter. Figure 4.23 shows the OSPA localization and cardinality components.

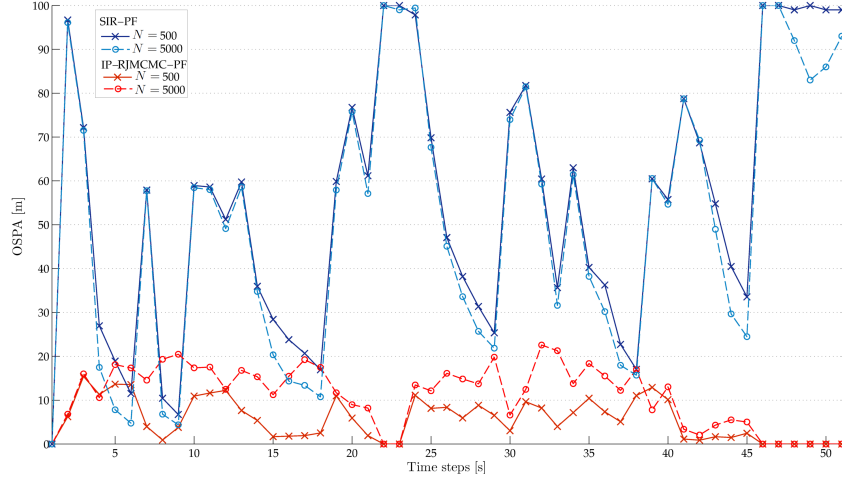


Figure 4.22: OSPA distance ($p = 2$, $c = 100$ m)

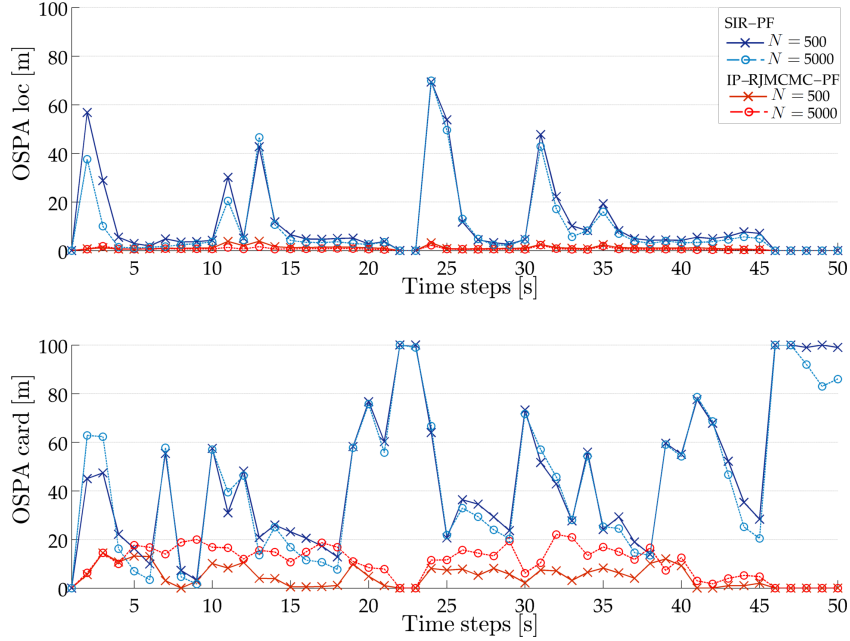


Figure 4.23: OSPA localization and cardinality components

4.7.5 Scenario 5: Time varying and unknown number of targets ($n_{max} = 10$)

This scenario considers a nonlinear TBD problem in which a maximum of 10 targets are present at any time. There are various target births and deaths throughout the scenario duration of $T_{max} = 50$ s. Targets are initialized at separate locations. The single target state $\mathbf{x}_{k,j}$ comprises only the position and velocity of the j^{th} target, $s_{k,j}$, and a measure of the average target return amplitude, $I_{k,j}$. To describe the dynamic of each target state $\mathbf{x}_{k,j}$, we adopt a NCV model for the object position and velocity and a random walk model for the object amplitude $I_{k,j}$ (cf. paragraph 4.6.1), with $\Delta_T = 1$ s, $\sigma_v = 2 \text{ m.s}^{-2}$ and $\sigma_i = (2/3) \text{ dB.m}^{-2}$.

The true tracks on a 2-D plane are depicted in Figure 4.24.

Table 4.9: Targets initial parameters

Target	k_b	k_d	SNR	$ v $	Target	k_b	k_d	SNR	$ v $
1	1	11	8	10	11	14	28	13	8
2	1	13	12	10	12	18	30	9	30
3	1	14	10	19	13	18	33	8	15
4	1	13	13	17	14	18	32	9	10
5	1	13	11	20	15	19	31	8	17
6	1	20	11	24	16	20	35	11	7
7	2	16	8	15	17	21	36	13	20
8	4	17	12	9	18	21	31	11	11
9	4	19	9	20	19	22	39	14	9
10	11	29	11	27	20	33	51	13	26

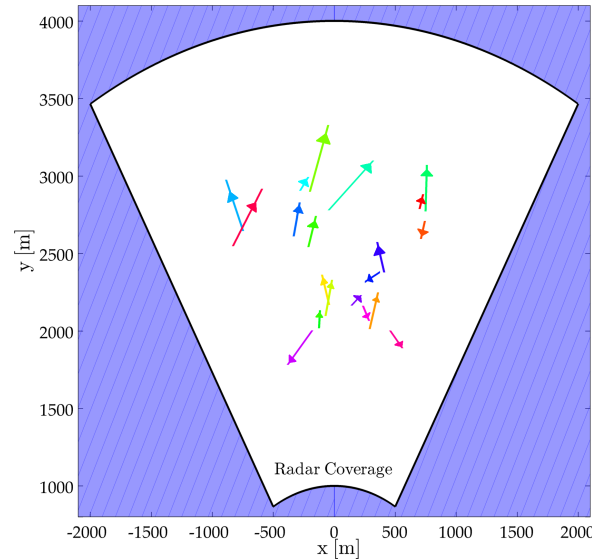


Figure 4.24: True tracks on a 2-D plane for Scenario 5.

We compared the tracking accuracy and the track initiation/termination performances of the conventional (a) IP-RJMCMC-PF (Algorithm 6) and the (b) MCHPF (Algorithm 9) over 100 Monte-Carlo simulations. For these comparisons, both algorithms are given the same initialization, so as to ensure a fair overall comparison.

State estimation is performed with the following procedure. A target is declared present if the estimated probability of existence is greater than 0.5, otherwise no target is declared. If a target is declared, the state estimate is given by the mean of the posterior state distribution, otherwise if no target is declared, there is no state estimate.

The tracks produced by each algorithm over 100 Monte Carlo trials are shown in Figure 4.25.

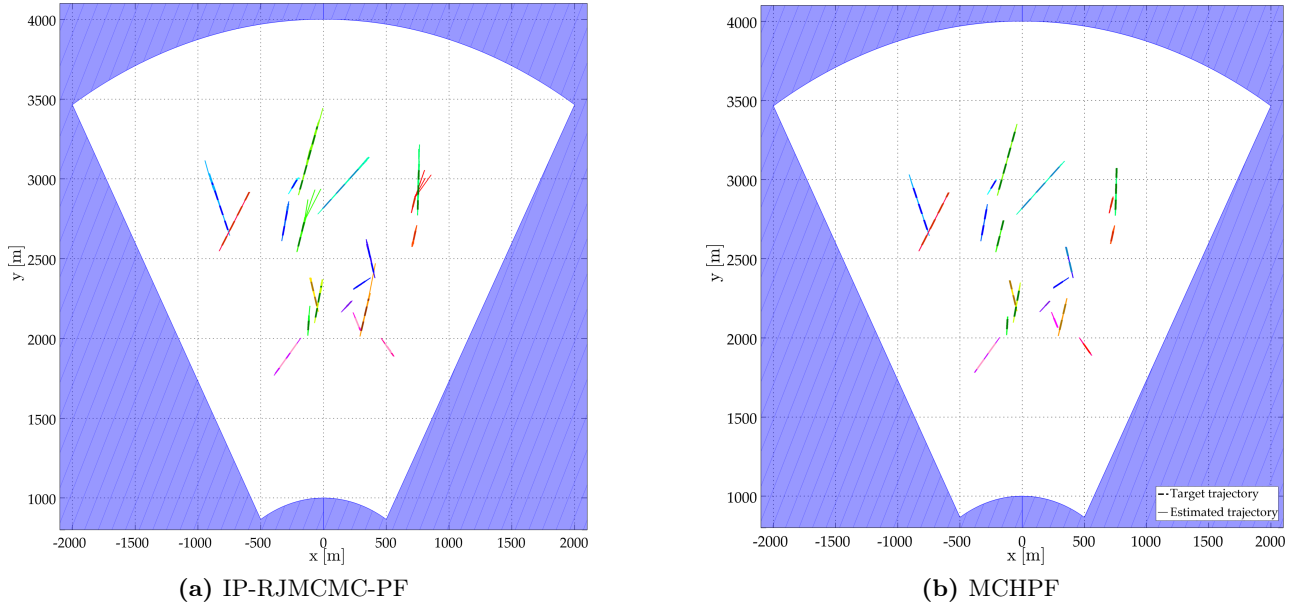


Figure 4.25: Estimated trajectories (i.e., particles-based conditional mean) over 100 Monte Carlo runs. Each filter uses $N = 1000$ particles. The proposed MCHPF provides better tracking performance than the IP-RJMCMC-PF filter.

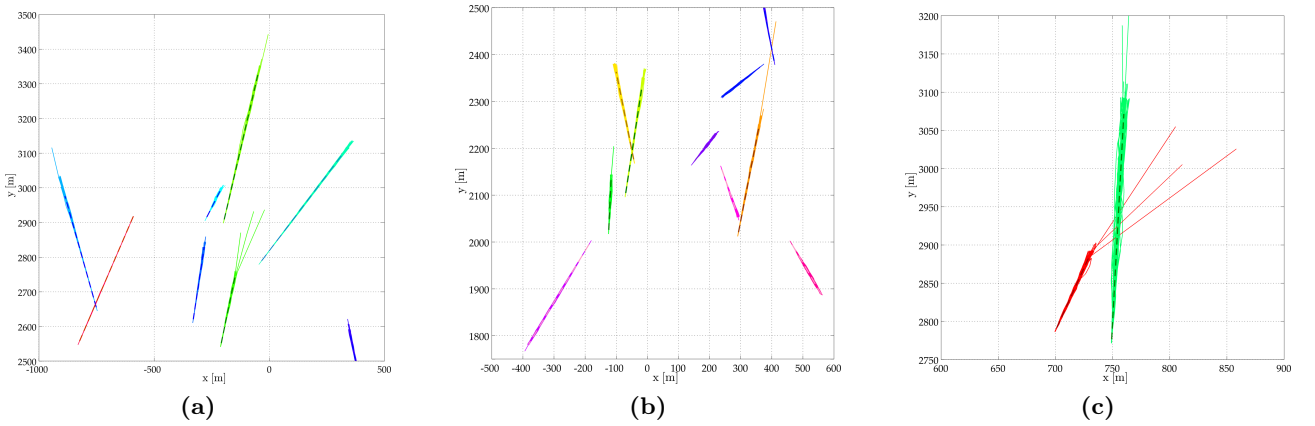


Figure 4.26: Zoom on tracks produced by the IP-RJMCMC-PF filter.

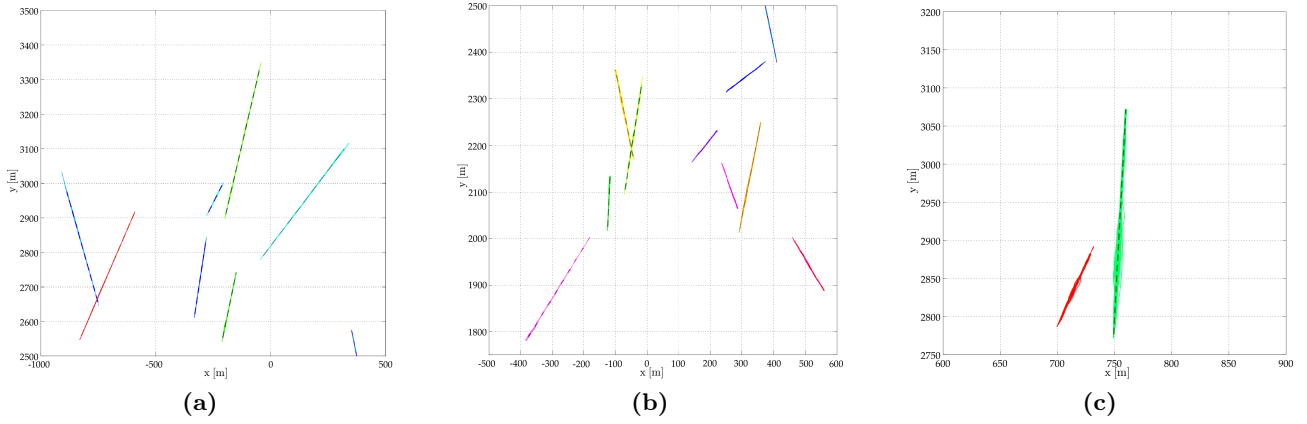


Figure 4.27: Zoom on tracks produced by the MCHPF filter.

Figure 4.29 compares the OSPA penalty ($p = 2$, $c = 100$ m) for the two filters, further confirming that the MCHPF filter generally outperforms the IP-RJMCMC-PF filter. Figure 4.28 shows the OSPA localization and cardinality components.

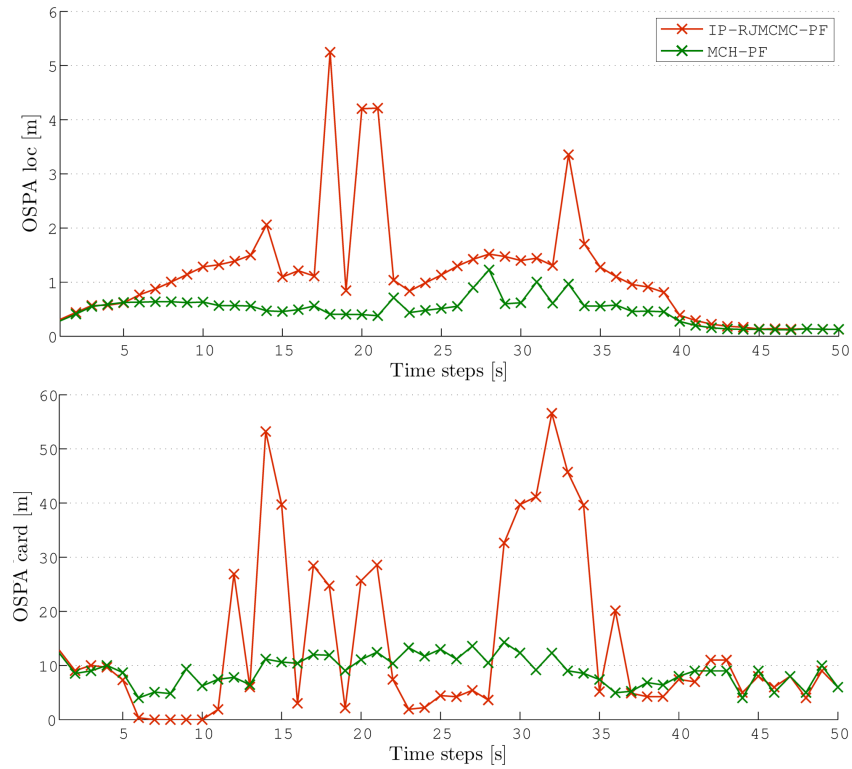


Figure 4.28: OSPA localization and cardinality components

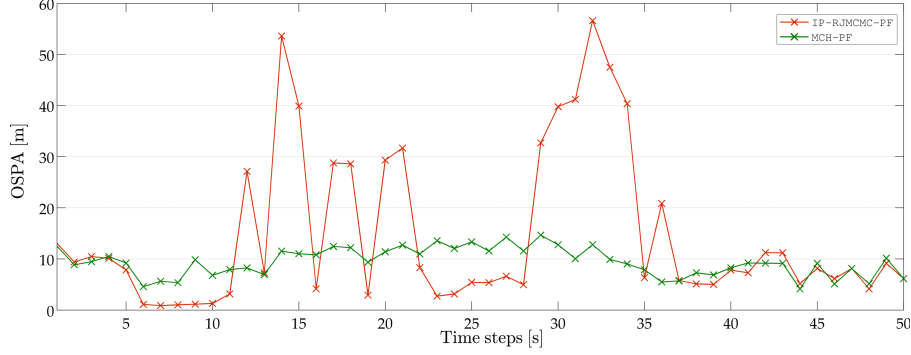


Figure 4.29: OSPA penalty ($p = 2$, $c = 100$ m)

Preliminary results indicate that the proposed MCHPF filter performs remarkably well, and significantly outperforms the IP-RJMCMC-PF filter in both localization and cardinality estimation.

Figure 4.30 reports the mean and standard deviation of the estimated cardinality distribution versus time for both filters, further confirming that the MCHPF filter outperforms the IP-RJMCMC-PF filter.

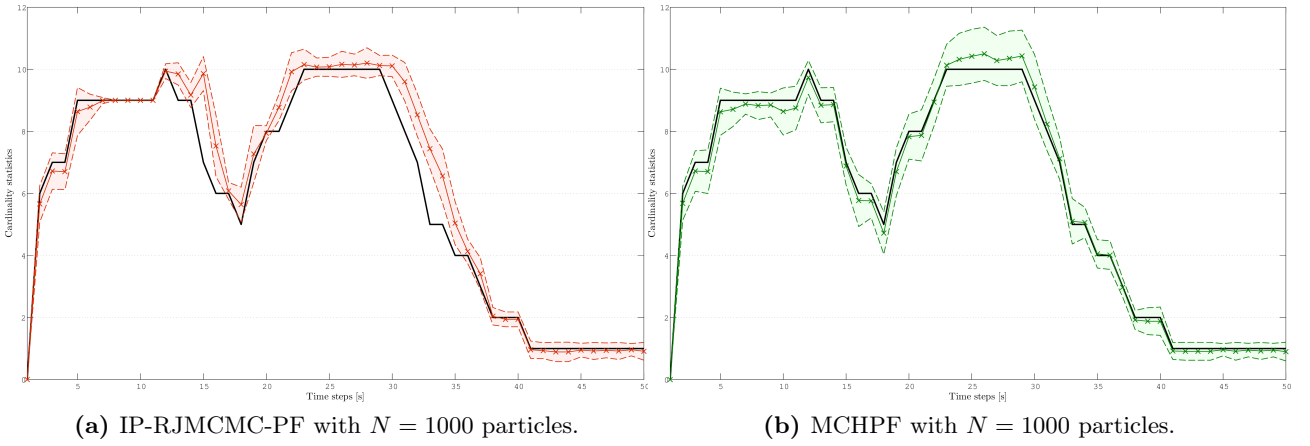


Figure 4.30: Cardinality statistics (mean and standard deviation) versus time. The MCHPF filter initiates and terminates tracks with a very short delay.

Finally Figure 4.31 shows the existence probability associated to each target versus time for the two filters.

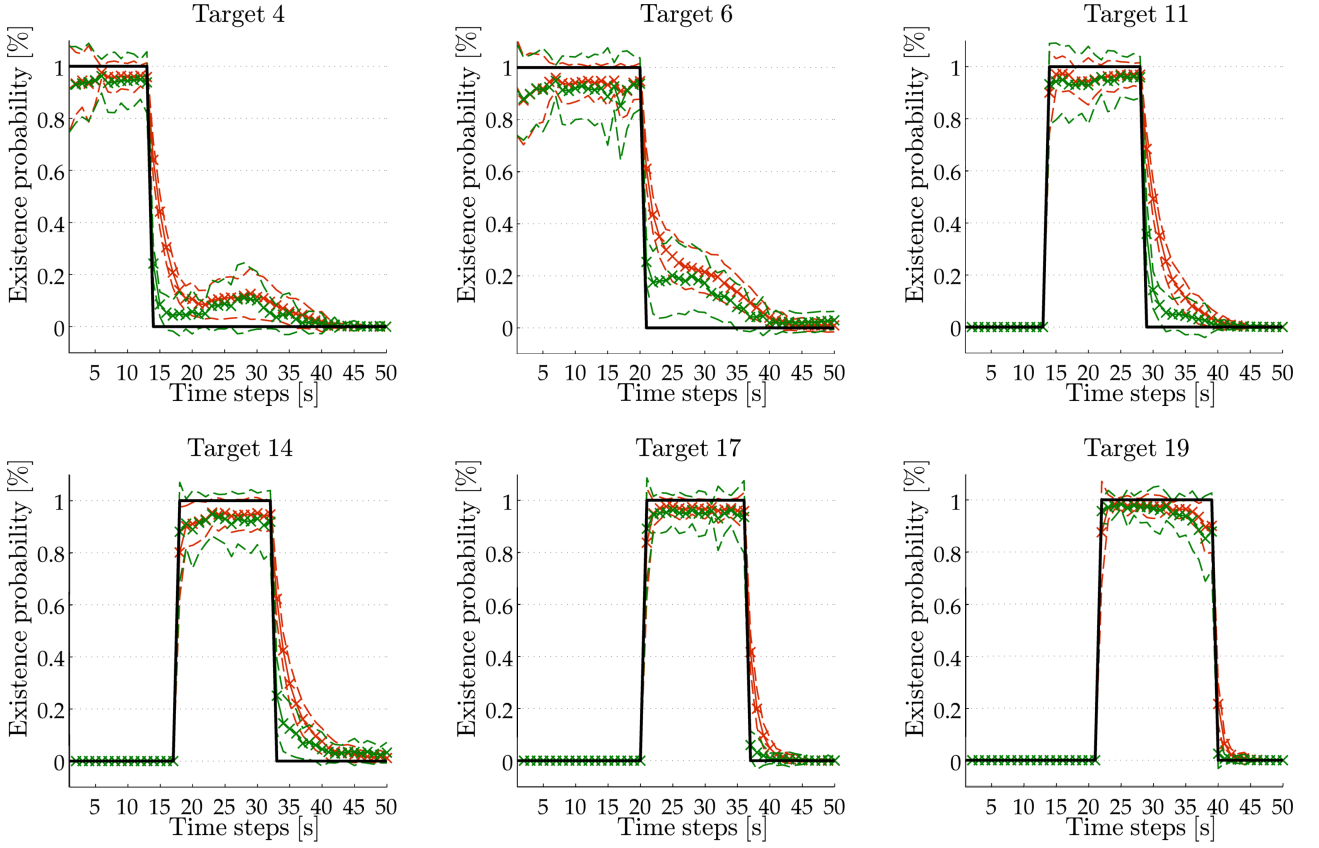


Figure 4.31: Existence probability associated to each target versus time.

Approximations of the Multi-target Bayes Filter

Although SMC implementations of the multi-target Bayes filter are computationally tractable, they are still expensive, especially when the number of targets is large. Thus, it is important to search for computationally cheaper alternatives. The PHD filter [44, 92, 129] and the CPHD filter [93, 137] have generated a great deal of interest in recent years. The PHD and CPHD filters are moment and cardinality approximations of the Bayes multi-target filter, which operate on the single-target state space. These filters do not keep the identities of the targets, and thus avoid the combinatorial problem that arises from data association. However, this can prove to be a drawback as often consistency of labels is required.

In addition to the PHD and CPHD filters, the CBMeMBer recursion has recently been proposed as a tractable approximation to the Bayes multi-target recursion under low clutter density scenarios [95, 140]. Unlike the PHD and CPHD recursions, which propagate moments and cardinality distributions, the Multi-target Multi-Bernoulli (MeMBer) recursion propagates a set of multi-Bernoulli parameters characterizing the posterior multi-target RFS, i.e. propagates (approximately) the multi-target posterior density. These principled approximations of the general multi-target Bayes recursive filter have been developed in the context of point measurement based approaches. An extension for our specific TBD application is still needed.

This chapter is organized as follows. Section 5.1 reviews the PHD and CPHD filters for point measurements. Section 5.2 briefly summarizes the CBMeMBer recursion and proposes a new update step suitable for our specific TBD application. Finally section 5.3 demonstrates the performance of the SMC-CBMeMBer filter on simulated data in a TBD context.

Remark. In this chapter, generic SMC implementations reminiscent of the SIR-PF of the proposed approximated filter are detailed. Nonetheless more efficient particle implementations can be derived from chapter 4.

5.1 Survey of the PHD and CPHD Filters

Since $f_k(\mathbf{X}_k|\mathbf{Z}_{1:k})$ is defined over $\mathcal{F}(\mathcal{X})$, practical implementation of the multi-target Bayes nonlinear filter is challenging and usually limited to a small number of targets. In order to overcome this limitation, Mahler [95] proposed to propagate moments of $f_k(\mathbf{X}_k|\mathbf{Z}_{1:k})$ and cardinality distributions.

5.1.1 Probability Hypothesis Density (PHD) Filter

The Probability Hypothesis Density (PHD) or intensity function is the first moment of the multiple target posterior distribution. The PHD is defined as the density $D_k(\mathbf{x}|\mathbf{Z}_{1:k}) = \int \delta_{\mathbf{x}}(\mathbf{x}) f_k(\mathbf{X}|\mathbf{Z}_{1:k}) \delta \mathbf{X}$ defined over \mathcal{X} . The integral of the PHD over any region S of the state space, that is

$$\int_S D_k(\mathbf{x}|\mathbf{Z}_{1:k}) d\mathbf{x} = \int |\mathbf{X}_k \cap S| f_k(\mathbf{X}_k|\mathbf{Z}_{1:k}) \delta \mathbf{X}_k, \quad (5.1)$$

provides an estimate of the number of targets contained in S and the target states can be estimated by determining the peaks of this distribution.

The PHD filter is a first-moment filter which propagates the PHD of a dynamic point process instead of the posterior. In order to derive a closed-form recursion, a Poisson point process assumption is made after the prediction and the update steps.

The PHD filter recursion, interpreted relative to the general multi-target Bayes recursion, is pictorially described below.

MULTI-TARGET BAYES RECURSION VS. PHD RECURSION:

$$\begin{array}{ccccccc} \dots \rightarrow & f_{k-1}(\mathbf{X}_{k-1}|\mathbf{Z}_{1:k-1}) & \xrightarrow{\text{prediction}} & f_{k|k-1}(\mathbf{X}_k|\mathbf{Z}_{1:k-1}) & \xrightarrow{\text{Bayes' rule}} & f_k(\mathbf{X}_k|\mathbf{Z}_{1:k}) & \rightarrow \dots \\ & \downarrow & & \downarrow & & \downarrow & \\ \dots \rightarrow & D_{k-1}(\mathbf{x}_{k-1}|\mathbf{Z}_{1:k-1}) & \xrightarrow[\text{prediction}]{\text{PHD}} & D_{k|k-1}(\mathbf{x}_k|\mathbf{Z}_{1:k-1}) & \xrightarrow[\text{update}]{\text{PHD}} & D_k(\mathbf{x}_k|\mathbf{Z}_{1:k}) & \rightarrow \dots \end{array}$$

The derivation for the PHD prediction and update equations are obtained by considering the multi-target prior distribution (2.40) and the multi-target posterior distribution (2.41) respectively. In order to derive the PHD filter, it is more convenient to represent the Bayes filter in the p.g.fl. form, i.e.

- P.G.FL. BAYES PREDICTION

$$G_{k|k-1}[h] = \int h^W p_{k|k-1}(W) \delta W, \quad (5.2)$$

- P.G.FL. BAYES UPDATE

$$G_{k|k}[h] = \int h^W p_{k|k}(W) \delta W. \quad (5.3)$$

The PHD filter is then found by taking the first-order moment of the p.g.fl. Bayes prediction and the p.g.fl. Bayes update, i.e.

- P.G.FL. PHD PREDICTION

$$D_{k|k-1}[h] = \frac{\delta}{\delta x} \left[\int h^W p_{k|k-1}(W) \delta W, \right]_{h=1} \quad (5.4)$$

- P.G.FL. PHD UPDATE

$$D_{k|k}[h] = \frac{\delta}{\delta x} \left[\int h^W p_{k|k}(W) \delta W, \right]_{h=1} \quad (5.5)$$

A full derivation of this process is provided in [92]. Suppose that the RFS of targets at time k is given by (3.12):

$$\Xi_{k|k-1} := \left[\bigcup_{\mathbf{x}_{k-1} \in \mathbf{X}_{k-1}} S_{k|k-1}(\mathbf{x}_{k-1}) \right] \cup \Gamma_k.$$

Proposition 5.1.1. *If the intensity measures of the multi-target probability distributions admit densities, then the prediction equation for the PHD filter is given by:*

THE PHD FILTER PREDICTION EQUATION

$$\boxed{D_{k|k-1}(\mathbf{x}_k | \mathbf{Z}_{1:k-1}) = \gamma_k(\mathbf{x}_k) + \langle p_S D_{k-1}(\cdot | \mathbf{Z}_{1:k-1}), \pi_{k|k-1}(\mathbf{x}_k | \cdot) \rangle}, \quad (5.6)$$

where,

- $\pi_{k|k-1}(\mathbf{x} | \mathbf{x}')$ is the single-target Markov transition density from time $k-1$ to k ;
- $p_S(\mathbf{x}') \stackrel{abbr}{=} p_{S,k|k-1}(\mathbf{x}')$ is the probability that a target with state \mathbf{w} at time-step $k-1$ will survive into time-step k ;
- $\gamma_k(\mathbf{x})$ is the intensity function of "newborn" targets between time $k-1$ and k ;
- $D_{k-1}(\mathbf{x} | \mathbf{Z}_{1:k-1})$ the intensity function of the multi-target posterior at time $k-1$, $f_{k-1}(\mathbf{X} | \mathbf{Z}_{1:k-1})$;
- $\langle g, f(\mathbf{x} | \cdot) \rangle = \int f(\mathbf{x} | \mathbf{x}') g(\mathbf{x}') d\mathbf{x}'$.

Suppose now that a new observation-scan has been collected at time-step k by a single sensor. Let us consider the classical setup, that is, a pre-processing chain isolates detection plots before the tracking chain. Thus, the set of noisy observations \mathbf{Z}_k , which may include false alarms and may have missed detections, is modeled as follows. A given target $\mathbf{x}_k \in \mathbf{X}_k$, at time k , is either detected with probability $p_{D,k}(\mathbf{x}_k)$ and generates an observation \mathbf{z}_k with likelihood $g_k(\mathbf{z}_k|\mathbf{x}_k)$, or missed with probability $1 - p_{D,k}(\mathbf{x}_k)$, i.e. each state $\mathbf{x}_k \in \mathbf{X}_k$ generates:

- a BERNOULLI RFS $\Delta_k(\mathbf{x}_k)$ with $r = p_{D,k}(\mathbf{x}_k)$ and $s(\cdot) = g_k(\cdot|\mathbf{x}_k)$.

Moreover, the observation set \mathbf{Z}_k typically includes, in addition to target originated detections, a set of false alarms or clutter which can be modeled as:

- a POISSON RFS Θ_k with intensity function $\kappa_k(\cdot) = \lambda c(\cdot)$.

Thus, given a realization \mathbf{X}_k of the multi-target state at time k , the multi-target measurement \mathbf{Z}_k can be modeled by the union:

$$\Sigma_k = \left[\bigcup_{\mathbf{x}_k \in \mathbf{X}_k} \underbrace{\Delta_k(\mathbf{x}_k)}_{\text{Bernoulli Detection}} \right] \cup \underbrace{\Theta_k}_{\text{False alarms \& Clutter}}. \quad (5.7)$$

The RFS multi-target measurement equation (5.7) accounts for detection uncertainty and clutter. The statistical behavior of the RFS Σ_k is described by the conditional probability density $\vartheta_k(\mathbf{Z}_k|\mathbf{X}_k)$ in an analogous fashion to the likelihood function for random vector observations.

Proposition 5.1.2. *Assume that the prediction distribution is Poisson, and that the intensity measures admit densities. Then $D_{k|k-1}(\mathbf{x}_k|\mathbf{Z}_{1:k-1})$ is updated to a new PHD $D_k(\mathbf{x}_k|\mathbf{Z}_{1:k})$ using the following approximate Bayes corrector equation:*

THE PHD FILTER UPDATE EQUATION

$$D_k(\mathbf{x}_k|\mathbf{Z}_{1:k}) = \left[(1 - p_D(\mathbf{x}_k)) + \sum_{\mathbf{z} \in \mathbf{Z}_k} \frac{p_D(\mathbf{x}_k) g_k(\mathbf{z}|\mathbf{x}_k)}{\kappa_k(\mathbf{z}) + \langle p_D g_k(\mathbf{z}|\cdot), D_{k|k-1}(\cdot|\mathbf{Z}_{1:k-1}) \rangle} \right] D_{k|k-1}(\mathbf{x}_k|\mathbf{Z}_{1:k-1}), \quad (5.8)$$

where,

- $p_D(\mathbf{x}) \stackrel{\text{abbr}}{=} p_{D,k}(\mathbf{x})$ is the probability that an observation will be detected at time k from a target with state \mathbf{x} ;
- $g_k(\mathbf{z}|\mathbf{x})$ is the single-target measurement likelihood at time k ;
- $\kappa_k(\mathbf{z}) = \lambda_k c_k(\mathbf{z})$ is the clutter model with λ_k being the Poisson parameter specifying the expected number of Poisson-distributed false alarms and c_k is the probability distribution over the observation space of clutter points.

5.1.2 Particle PHD Filter Implementation

The SMC implementation of the PHD filter is based on the random sample approximation of the PHD:

$$D_k(\mathbf{x}_k | \mathbf{Z}_{1:k}) \approx \sum_{i=1}^N w_k^{(i)} \delta_{\mathbf{x}_k^{(i)}}(\mathbf{x}_k), \quad (5.9)$$

where $\delta_{\mathbf{y}}(\mathbf{x})$ is the Dirac delta function and $\{(w_k^{(i)}, \mathbf{x}_k^{(i)})\}_{i=1}^N$ is the weighted particle set, consisting of random samples $\mathbf{x}_k^{(i)}$ and their weights $w_k^{(i)}$. Let us recall that:

$$\int_{\mathcal{X}} D_k(\mathbf{x} | \mathbf{Z}_{1:k}) d\mathbf{x} = \nu_k \in \mathbb{R}, \quad (5.10)$$

is the expected number of targets in the state space \mathcal{X} . Then according to (5.9) and (5.10), for large N , the sum of importance weights approximates the expected number of targets, i.e.

$$\sum_{n=1}^N w_k^{(i)} \approx \nu_k. \quad (5.11)$$

The SMC implementation of the PHD filter is described pictorially below.

$$\begin{array}{ccccccc} \dots \rightarrow & D_{k-1}(\mathbf{x}_{k-1} | \mathbf{Z}_{1:k-1}) & \xrightarrow{\text{PHD prediction}} & D_k(\mathbf{x}_k | \mathbf{Z}_{1:k-1}) & \xrightarrow{\text{PHD update}} & D_k(\mathbf{x}_k | \mathbf{Z}_{1:k}) & \rightarrow \dots \\ & \downarrow & & \downarrow & & \downarrow & \\ \dots \rightarrow & \{\mathbf{x}_{k-1}^{(i)}, w_{k-1}^{(i)}\} & \xrightarrow{\text{SMC prediction}} & \{\mathbf{x}_{k|k-1}^{(i)}, w_{k|k-1}^{(i)}\} & \xrightarrow{\text{SMC update}} & \{\mathbf{x}_k^{(i)}, w_k^{(i)}\} & \rightarrow \dots \end{array}$$

The prediction step forms two types of particles, corresponding to persisting and newborn targets. The particles for newborn targets are drawn from density $\gamma_k(\mathbf{x}) / \int \gamma_k(\mathbf{x}) d\mathbf{x}$. The number of newborn particles corresponds to the nearest integer to $\int \gamma_k(\mathbf{x}) d\mathbf{x}$. Let N_{k-1} be the number of particles predicted forward and $N_{k,B}$ be the number of new particles arisen from the birth process. The generic SMC implementation of the PHD recursion is described in Algorithm 12.

Note that in the above formulation of the PHD filter, the intensity function of newborn targets $\gamma_k(\mathbf{x})$ 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 \mathcal{X} . Another approach consists to initialize new tracks based on measurements. Typically, the newborn particles are placed 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. However, if the birth density is adapted in accordance with the measurements, the PHD equations must be applied in a different form, otherwise the PHD filter cardinality estimates will be biased [118].

Algorithm 12: Pseudo-code of the SMC-PHD filter

input : $\{\mathbf{x}_{k-1}^{(i)}, w_{k-1}^{(i)}\}_{i=1}^{N_{k-1}}$ and a new measurement, \mathbf{Z}_k

output: $\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}_{i=1}^{N_k}$

Initialization at $k = 0$

Draw from a prior distribution N_0 particles.

Set the initial weights $w_0^{(i)}$ to $\frac{\hat{n}_0}{N_0}$, with \hat{n}_0 the expected initial number of targets.

At time $k > 0$,

1 - *Prediction* (5.6)

for $i \leftarrow 1$ **to** N_{k-1} **do**

Sample $\tilde{\mathbf{x}}_k^{(i)}$ from a proposal density $q_k(\cdot|\mathbf{x}_{k-1}^{(i)}, \mathbf{Z}_k)$.

Compute the predicted weights:

$$\tilde{w}_{k|k-1}^{(i)} = \frac{p_S(\mathbf{x}_{k-1}^{(i)}) \pi_{k|k-1}(\tilde{\mathbf{x}}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})}{q_k(\tilde{\mathbf{x}}_k^{(i)}|\mathbf{x}_{k-1}^{(i)}, \mathbf{Z}_k)} w_{k-1}^{(i)}. \quad (5.12)$$

end

for $i \leftarrow N_{k-1} + 1$ **to** $N_{k-1} + N_{k,B}$ **do**

Sample $\tilde{\mathbf{x}}_k^{(i)}$ from another proposal density $b_k(\cdot|\mathbf{Z}_k)$.

Compute the weights of new born particles:

$$\tilde{w}_{k|k-1}^{(i)} = \frac{\gamma_k(\tilde{\mathbf{x}}_k^{(i)})}{b_k(\tilde{\mathbf{x}}_k^{(i)}|\mathbf{Z}_k)} \frac{1}{N_{k,B}}. \quad (5.13)$$

end

2 - *Update* (5.8)

Let $\mathbf{Z}_k = \{\mathbf{z}_{k,1}, \dots, \mathbf{z}_{k,m_k}\}$ be the observations received at time step k .

foreach observation $\mathbf{z}_{k,j}$ **in** \mathbf{Z}_k **do**

Compute the likelihoods of each particle $g_k(\mathbf{z}_{k,j}|\tilde{\mathbf{x}}_k^{(i)})$.

Compute $C_k(\mathbf{z}_{k,j}) = \sum_{i=1}^{N_{k-1}+N_{k,B}} p_D(\tilde{\mathbf{x}}_k^{(i)}) g_k(\mathbf{z}_{k,j}|\tilde{\mathbf{x}}_k^{(i)}) \tilde{w}_{k|k-1}^{(i)}$.

end

for $i \leftarrow 1$ **to** $N_{k-1} + N_{k,B}$ **do**

$$\tilde{w}_k^{(i)} = \left[(1 - p_D(\tilde{\mathbf{x}}_k^{(i)})) + \sum_{\mathbf{z}_{k,j} \in \mathbf{Z}_k} \frac{p_D(\tilde{\mathbf{x}}_k^{(i)}) g_k(\mathbf{z}_{k,j}|\tilde{\mathbf{x}}_k^{(i)})}{\kappa_k(\mathbf{z}_{k,j}) + C_k(\mathbf{z}_{k,j})} \right] w_{k|k-1}^{(i)}. \quad (5.14)$$

end

3 - *Estimation of target location*

Compute the total mass of the particles $\nu_k = \sum_{i=1}^{N_{k-1}+N_{k,B}} \tilde{w}_k^{(i)}$.

Target estimates are found by fitting a Gaussian mixture model to the particles where the number of components in the mixture is the expected number of targets at time k , $\lfloor \nu_k \rfloor$ (the nearest integer to ν_k).

If there are no more incoming measurements then exit, otherwise proceed.

4 - *Resampling*

Resample $\{\tilde{w}_k^{(i)}/\nu_k, \tilde{\mathbf{x}}_k^{(i)}\}_{i=1}^{N_{k-1}+N_{k,B}}$ to get $\{w_k^{(i)}/\nu_k, \mathbf{x}_k^{(i)}\}_{i=1}^{N_k}$.

5.1.3 The Cardinalized Probability Hypothesis Density (CPHD) Filter

Let us relax the Poisson assumption on the number of objects and consider a generalization of the PHD recursion, known as the Cardinalized Probability Hypothesis Density (CPHD) recursion. Let $\rho_k(n|\mathbf{Z}_{1:k})$ denote the cardinality distribution at time k . The CPHD filter jointly propagates the intensity function (PHD), $D_k(\mathbf{x}_k|\mathbf{Z}_{1:k})$ and the cardinality distribution (the distribution of the number of targets), $\rho_k(n|\mathbf{Z}_{1:k})$. The CPHD recursion thus carries the same advantages as the PHD recursion, and supposedly alleviates the major limitation regarding the accuracy of cardinality estimates.

The CPHD filter recursion, interpreted relative to the general multi-target Bayes recursion, is pictorially described below.

MULTI-TARGET BAYES RECURSION VS. CPHD RECURSION:

$$\begin{array}{ccccccc}
 \dots \rightarrow & p_{k-1|k-1}(\mathbf{X}_{k-1}|\mathbf{Z}_{1:k-1}) & \xrightarrow{\text{prediction}} & p_{k|k-1}(\mathbf{X}_k|\mathbf{Z}_{1:k-1}) & \xrightarrow{\text{Bayes' rule}} & p_{k|k}(\mathbf{X}_k|\mathbf{Z}_{1:k}) & \rightarrow \dots \\
 & \downarrow & & \downarrow & & \downarrow & \\
 \dots \rightarrow & D_{k-1|k-1}(\mathbf{X}_{k-1}|\mathbf{Z}_{1:k-1}) & \xrightarrow[\text{prediction}]{\text{PHD}} & D_{k|k-1}(\mathbf{X}_k|\mathbf{Z}_{1:k-1}) & \xrightarrow[\text{update}]{\text{PHD}} & D_{k|k}(\mathbf{X}_k|\mathbf{Z}_{1:k}) & \rightarrow \dots \\
 \dots \rightarrow & \rho_{k-1|k-1}(n|\mathbf{Z}_{1:k-1}) & \xrightarrow[\text{prediction}]{\text{cardinality}} & \rho_{k|k-1}(n|\mathbf{Z}_{1:k-1}) & \xrightarrow[\text{update}]{\text{cardinality}} & \rho_{k|k}(n|\mathbf{Z}_{1:k-1}) & \rightarrow \dots
 \end{array}$$

Analogous to how the PHD recursion is based on Poisson RFS approximations, the CPHD recursion rests on i.i.d. cluster RFS approximations. The CPHD recursion assumes:

- Each target evolves and generates a measurement independently of one another.
- The birth RFS is an i.i.d. cluster RFS. New targets appear in the scene independently of existing targets.
- The clutter RFS is an i.i.d. cluster RFS and is independent of the target generated measurement RFSs.
- The predicted and posterior multi-target RFSs are approximated as i.i.d. cluster RFSs.

Based on the above assumptions, the p.g.fl. form of the multi-target Bayes recursion is used to derive the predicted and posterior p.g.fl.s. Then, the predicted and posterior p.g.fl.s are used to calculate the predicted and posterior intensities and cardinality distributions, respectively, using standard properties of the p.g.fl.. A full derivation of this process is provided in [93].

Suppose that the RFS of targets at time k is given by: (3.12)

$$\Xi_{k|k-1} := \left[\bigcup_{\mathbf{x}_{k-1} \in \mathbf{X}_{k-1}} S_{k|k-1}(\mathbf{x}_{k-1}) \right] \cup \Gamma_k.$$

Proposition 5.1.3. *Then the predicted cardinality distribution can be written as a convolution:*

THE CPHD CARDINALITY PREDICTION

$$\rho_{k|k-1}(n|\mathbf{Z}_{1:k-1}) = \sum_{j=0}^n \rho_{\Gamma,k}(n-j) \rho_{S,k|k-1}[D_{k-1}, \rho_{k-1}](j), \quad (5.15)$$

where,

- $\rho_{\Gamma,k}(l)$ is the probability of having l spontaneous births at time k ,
- $\rho_{S,k|k-1}[\cdot](j)$ is the probability of having j surviving targets, i.e.,

$$\rho_{S,k|k-1}[D, \rho](j) = \sum_{l=j}^{\infty} C_j^l \frac{\langle p_S, D \rangle^j \langle 1 - p_S, D \rangle^{l-j}}{\langle 1, D \rangle^l} \rho(l),$$

with C_j^l the binomial coefficient $\frac{l!}{j!(l-j)!}$, and $\langle \cdot, \cdot \rangle$ the inner product defined between two real valued functions α and β by:

$$\langle \alpha, \beta \rangle = \int \alpha(x) \beta(x) dx.$$

The predicted intensity is given by (5.6):

THE CPHD INTENSITY PREDICTION

$$D_{k|k-1}(\mathbf{x}_k|\mathbf{Z}_{1:k-1}) = \gamma_k(\mathbf{x}_k) + \langle p_S, D_{k-1}(\cdot|\mathbf{Z}_{1:k-1}), \pi_{k|k-1}(\mathbf{x}_k|\cdot) \rangle, \quad (5.16)$$

where,

- $\pi_{k|k-1}(\mathbf{x}|\mathbf{x}')$ is the single-target Markov transition density from time $k-1$ to k ;
- $p_S(\mathbf{x}') \stackrel{\text{abbr}}{=} p_{S,k|k-1}(\mathbf{x}')$ is the probability that a target with state \mathbf{x}' at time-step $k-1$ will survive at time step k ;
- $\gamma_k(\mathbf{x})$ is the intensity function of "newborn" targets between time $k-1$ and k ;
- $D_{k-1}(\mathbf{x}|\mathbf{Z}_{1:k-1})$ the intensity function of the multi-target posterior at time $k-1$, $f_{k-1}(\mathbf{X}|\mathbf{Z}_{1:k-1})$;
- $\langle g, f(\mathbf{x}|\cdot) \rangle = \int f(\mathbf{x}|\mathbf{x}') g(\mathbf{x}') d\mathbf{x}'$ is the inner product.

Remark. The CPHD cardinality prediction (5.15) is simply a CONVOLUTION of the cardinality distributions of the independent birth and surviving targets. The CPHD intensity prediction (5.16) is the same as the PHD prediction.

Suppose now that a new observation-scan has been collected at time-step k by a single sensor. Let us consider the classical setup, that is, a pre-processing chain isolates detection plots before the tracking chain. Thus, the set of noisy observations \mathbf{Z}_k , which may include false alarms and have missed detections, is modeled by the RFS (5.7):

$$\Sigma_k = \left(\bigcup_{\mathbf{x}_k \in \mathbf{X}_k} \Delta_k(\mathbf{x}_k) \right) \cup \Theta_k.$$

Proposition 5.1.4. *Let $D_{k|l} = D_{k|l}(x_k|\mathbf{Z}_{1:l})$, $\rho_{k|l}(n) = \rho_{k|l}(n|\mathbf{Z}_{1:l})$, and $p_D = p_{D,k}(x_k)$. Then the updated cardinality distribution ρ_k can be written as:*

THE CPHD CARDINALITY UPDATE

$$\boxed{\rho_k(n|\mathbf{Z}_{1:k}) = \frac{\Upsilon_k^0[D_{k|k-1}, Z_k](n)\rho_{k|k-1}(n)}{\langle \Upsilon_k^0[D_{k|k-1}, Z_k], \rho_{k|k-1} \rangle}}, \quad (5.17)$$

where,

- the inner product $\langle \Upsilon_k^0[D, Z_k], \rho \rangle = \sum_{i=0}^{\infty} \Upsilon_k^0[D, Z_k](i)\rho(i)$;
- the sequence $\Upsilon_k^u[D, Z](n)$ is defined for $u \in \{0, 1\}$ as follows:

$$\Upsilon_k^u[D, Z](n) = \sum_{j=0}^{\min(|Z|, n)} (|Z| - j)! \rho_{\Theta, k}(|Z| - j) P_{j+u}^n \text{esf}_j(\Xi_k(D, Z)) \frac{\langle 1 - p_D, D \rangle^{n-(j+u)}}{\langle 1, D \rangle^n},$$

with P_j^n the permutation coefficient $\frac{n!}{(n-j)!}$, $\rho_{\Theta, k}(z)$ the cardinality distribution of clutter at time k , esf_j the elementary symmetric function of order j defined for a finite set \mathbf{Z} of real numbers by:

$$\text{esf}_j(Z) = \sum_{S \subseteq Z, |S|=j} \left(\prod_{\zeta \in S} \zeta \right), \quad \text{with } \text{esf}_0(Z) = 1 \text{ by convention,}$$

and $\Xi_k(D, Z) = \frac{\langle 1, \kappa_k \rangle}{\kappa_k(z)} \{ \langle p_D, D, g_k(z|\cdot) \rangle : z \in Z \}$.

The updated intensity D_k is given by:

THE CPHD INTENSITY UPDATE

$$D_k(\mathbf{x}_k | \mathbf{Z}_{1:k}) = D_{k|k-1} \left((1 - p_D) \frac{\langle \Upsilon_k^1[D_{k|k-1}, Z_k], \rho_{k|k-1} \rangle}{\langle \Upsilon_k^0[D_{k|k-1}, Z_k], \rho_{k|k-1} \rangle} + \sum_{z \in Z_k} \frac{\langle \Upsilon_k^1[D_{k|k-1}, Z_k - \{z\}], \rho_{k|k-1} \rangle}{\langle \Upsilon_k^0[D_{k|k-1}, Z_k], \rho_{k|k-1} \rangle} \frac{\langle 1, \kappa_k \rangle}{\kappa_k(z)} p_D(\mathbf{x}_k) g_k(z | \mathbf{x}_k) \right), \quad (5.18)$$

where,

- $p_D(\mathbf{x}) \stackrel{abbr}{=} p_{D,k}(\mathbf{x})$ is the probability that a target with state \mathbf{x} will be detected at time k ;
- $g_k(z | \mathbf{x})$ is the single-target measurement likelihood at time k ;
- $\kappa_k(z)$ is the intensity of clutter measurements at time k .

Remark. Note that the CPHD cardinality and intensity prediction (5.15)-(5.16) are uncoupled, while the CPHD cardinality and intensity update (5.17)-(5.18) are coupled. Nonetheless, the CPHD intensity update (5.18) is similar to the PHD update in the sense that both have one missed detection term and $|Z_k|$ detection terms. The cardinality update (5.17) incorporates the clutter cardinality, the measurement set, the predicted intensity and predicted cardinality distribution.

5.1.4 Particle CPHD Filter Implementations

The SMC implementation of the CPHD filter is based on the random sample approximation of the intensity:

$$D_k(\mathbf{x}_k | \mathbf{Z}_{1:k}) \approx \sum_{i=1}^N w_k^{(i)} \delta_{\mathbf{x}_k^{(i)}}(\mathbf{x}_k), \quad (5.19)$$

where,

- $\delta_{\mathbf{y}}(\mathbf{x})$ is the Dirac delta function;
- $\{(w_k^{(i)}, \mathbf{x}_k^{(i)})\}_{i=1}^N$ is the weighted particle set, consisting of random samples $\mathbf{x}_k^{(i)}$ and their weights $w_k^{(i)}$.

If at time $k-1$, the intensity D_{k-1} and the cardinality distribution ρ_{k-1} are given, and

$$D_{k-1}(\mathbf{x}_{k-1} | \mathbf{Z}_{1:k-1}) \approx \sum_{i=1}^{N_{k-1}} w_{k-1}^{(i)} \delta_{\mathbf{x}_{k-1}^{(i)}}(\mathbf{x}_{k-1}), \quad (5.20)$$

then given importance (or proposal) densities $q_k(\cdot | \mathbf{x}_{k-1}, \mathbf{Z}_k)$ and $b_k(\cdot | \mathbf{Z}_k)$, the predicted cardinality distribution $\rho_{k|k-1}(n | \mathbf{Z}_{1:k-1})$ can be computed as follows:

THE CPHD CARDINALITY PREDICTION

$$\rho_{k|k-1}(n|\mathbf{Z}_{1:k-1}) \approx \sum_{j=0}^n \rho_{\Gamma,k}(n-j) \sum_{l=j}^{\infty} C_j^l \frac{\langle p_S^{(1:N_{k-1})}, w_{k-1}^{(1:N_{k-1})} \rangle^j \langle 1 - p_S^{(1:N_{k-1})}, w_{k-1}^{(1:N_{k-1})} \rangle^{l-j}}{\langle 1, w_{k-1}^{(1:N_{k-1})} \rangle^l} \rho_{k-1}(l), \quad (5.21)$$

where,

$$p_S^{(1:N_{k-1})} = [p_S(\mathbf{x}_{k-1}^{(1)}), \dots, p_S(\mathbf{x}_{k-1}^{(N_{k-1})})]^T, \quad w_{k-1}^{(1:N_{k-1})} = [w_{k-1}^{(1)}, \dots, w_{k-1}^{(N_{k-1})}]^T.$$

The predicted intensity $D_{k|k-1}(\mathbf{x}_k|\mathbf{Z}_{1:k-1})$ is given by:

THE CPHD INTENSITY PREDICTION

$$D_{k|k-1}(\mathbf{x}_k|\mathbf{Z}_{1:k-1}) \approx \sum_{i=1}^{N_{k-1}} \tilde{w}_{P,k|k-1}^{(i)} \delta_{\tilde{\mathbf{x}}_{P,k|k-1}^{(i)}}(\mathbf{x}_k) + \sum_{i=1}^{N_{k,B}} \tilde{w}_{\Gamma,k}^{(i)} \delta_{\tilde{\mathbf{x}}_{\Gamma,k}^{(i)}}(\mathbf{x}_k), \quad (5.22)$$

where,

$$\begin{aligned} \tilde{\mathbf{x}}_{P,k|k-1}^{(i)} &\sim q_k(\cdot|\mathbf{x}_{k-1}, \mathbf{Z}_k), \forall i = 1, \dots, N_{k-1}, & \tilde{w}_{P,k|k-1}^{(i)} &= w_{k-1}^{(i)} \frac{p_S(\mathbf{x}_{k-1}^{(i)}) \pi_{k|k-1}(\tilde{\mathbf{x}}_{P,k|k-1}^{(i)}|\mathbf{x}_{k-1}^{(i)})}{q_k(\tilde{\mathbf{x}}_{P,k|k-1}^{(i)}|\mathbf{x}_{k-1}^{(i)}, \mathbf{Z}_k)}, \\ \tilde{\mathbf{x}}_{\Gamma,k}^{(i)} &\sim b_k(\cdot|\mathbf{Z}_k), \forall i = 1, \dots, N_{k,B}, & \tilde{w}_{\Gamma,k}^{(i)} &= \frac{\Gamma_k(\tilde{\mathbf{x}}_{\Gamma,k}^{(i)})}{b_k(\tilde{\mathbf{x}}_{\Gamma,k}^{(i)}|\mathbf{Z}_k)} \frac{1}{N_{k,B}}. \end{aligned}$$

If at time k , the predicted intensity $D_{k|k-1}$ and the predicted cardinality distribution $\rho_{k|k-1}$ are given, and

$$D_{k|k-1}(\mathbf{x}_k|\mathbf{Z}_{1:k-1}) \approx \sum_{i=1}^{N_{k|k-1}} \tilde{w}_{k|k-1}^{(i)} \delta_{\tilde{\mathbf{x}}_k^{(i)}}(\mathbf{x}_k), \quad (5.23)$$

then the updated cardinality distribution $\rho_k(n|\mathbf{Z}_{1:k})$ can be computed as follows:

THE CPHD CARDINALITY UPDATE

$$\rho_k(n|\mathbf{Z}_{1:k}) \approx \frac{\Upsilon_k^0[\tilde{w}_{k|k-1}^{(1:N_{k|k-1})}, Z_k](n) \rho_{k|k-1}(n|\mathbf{Z}_{1:k-1})}{\langle \Upsilon_k^0[\tilde{w}_{k|k-1}^{(1:N_{k|k-1})}, Z_k], \rho_{k|k-1} \rangle}, \quad (5.24)$$

where,

$$\begin{aligned} p_D^{(1:N_{k|k-1})} &= [p_D(\tilde{\mathbf{x}}_k^{(1)}), \dots, p_D(\tilde{\mathbf{x}}_k^{(N_{k|k-1})})]^T, & \tilde{w}_{k-1}^{(1:N_{k|k-1})} &= [\tilde{w}_{k-1}^{(1)}, \dots, \tilde{w}_{k-1}^{(N_{k|k-1})}]^T, \\ \Upsilon_k^u[w, Z](n) &= \sum_{j=0}^{\min(|Z|, n)} (|Z| - j)! \rho_{\Theta,k}(|Z| - j) P_{j+u}^n \text{esf}_j(\Xi_k(w, Z)) \frac{\langle 1 - p_D^{(1:N_{k|k-1})}, w \rangle^{n-(j+u)}}{\langle 1, w \rangle^n}, \end{aligned}$$

with P_j^n the permutation coefficient $\frac{n!}{(n-j)!}$, esf_j the elementary symmetric function of order j computed by applying the Newton-Girard formulae [127], and $\Xi_k(w, Z) = \{\langle w, \xi_{k,z}^{(1:N_{k|k-1})} \rangle : z \in Z\}$ with

$$\xi_{k,z}^{(1:N_{k|k-1})} = \frac{\langle 1, \kappa_k \rangle}{\kappa_k(z)} [g_k(z|\tilde{\mathbf{x}}_k^{(1)}) p_D(\tilde{\mathbf{x}}_k^{(1)}), \dots, g_k(z|\tilde{\mathbf{x}}_k^{(N_{k|k-1})}) p_D(\tilde{\mathbf{x}}_k^{(N_{k|k-1})})]^T.$$

The updated intensity $D_k(\mathbf{x}_k|\mathbf{Z}_{1:k})$ is given by:

THE CPHD INTENSITY UPDATE

$$D_k(\mathbf{x}_k|\mathbf{Z}_{1:k}) \approx \sum_{i=1}^{N_{k|k-1}} \tilde{w}_k^{(i)} \delta_{\tilde{\mathbf{x}}_k^{(i)}}(\mathbf{x}_k), \quad (5.25)$$

where,

$$\begin{aligned} \tilde{w}_k^{(i)} &= \tilde{w}_{k|k-1}^{(i)} \left[(1 - p_D^{(1:N_{k|k-1})}) \frac{\langle \Upsilon_k^1[\tilde{w}_{k|k-1}^{(1:N_{k|k-1})}, Z_k], \rho_{k|k-1} \rangle}{\langle \Upsilon_k^0[\tilde{w}_{k|k-1}^{(1:N_{k|k-1})}, Z_k], \rho_{k|k-1} \rangle} \right. \\ &\quad \left. + \sum_{z \in Z_k} \frac{\langle \Upsilon_k^1[\tilde{w}_{k|k-1}^{(1:N_{k|k-1})}, Z_k - \{z\}], \rho_{k|k-1} \rangle}{\langle \Upsilon_k^0[\tilde{w}_{k|k-1}^{(1:N_{k|k-1})}, Z_k], \rho_{k|k-1} \rangle} \frac{\langle 1, \kappa_k \rangle}{\kappa_k(z)} p_D(\tilde{\mathbf{x}}_k^{(i)}) g_k(z|\tilde{\mathbf{x}}_k^{(i)}) \right]. \end{aligned}$$

Note that the standard formulations of both the PHD and CPHD filters assume that the intensity function of newborn targets $\gamma_k(\mathbf{x})$ is known a priori, meaning that, it is independent of the measurements. Moreover, if a target appears outside the region covered by the predefined birth intensity, the PHD and CPHD filters will not be aware of its existence. One way to circumvent this drawback is to create, at each scan, a myriad number of hypothesized newborn targets covering the entire state-space \mathcal{X} , which is clearly inefficient. Another approach consists to initialize new tracks based on measurements. Typically, the newborn particles are placed in the region of the state-space $\mathbf{x} \in \mathcal{X}$ for which the likelihood $g_k(z|\mathbf{x})$ will have high values. The PHD and CPHD equations must be applied in a different form [119].

Let us describe the SMC implementation of the CPHD filter, which enables an adaptive design of the target birth intensity based on the received measurements. The current measurements Z_k are used to place the newborn particles in the region of the state space where the inner product $\langle g_k(\mathbf{z}|\cdot), \gamma_k(\cdot) \rangle$ will have non-zero values. Thus for each $z \in Z_k$, a set of $M_{k,B}$ newborn particles is generated in such a manner that z can be considered as a random sample from the pdf $g_k(\mathbf{z}|\cdot)$. The total number of newborn target particles generated in this way is $N_{k,B} = M_{k,B} \cdot |Z_k|$. The weights of the new target particles are uniform. Hence, the prediction step forms two types of particles, corresponding to alive and newborn targets, $\{\tilde{\mathbf{x}}_{P,k|k-1}^{(i)}, \tilde{w}_{P,k|k-1}^{(i)}\}_{i=1}^{N_{k-1}}$ and $\{\tilde{\mathbf{x}}_{\Gamma,k}^{(i)}, \tilde{w}_{\Gamma,k|k-1}^{(i)}\}_{i=1}^{N_{k,B}}$.

The update step is performed differently for these two types of particles. If at time k , the predicted intensities $D_{P,k|k-1}$, $D_{\Gamma,k|k-1}$ and the predicted cardinality distribution $\rho_{k|k-1}$ are given, and

$$D_{P,k|k-1}(\tilde{\mathbf{x}}_{P,k|k-1}^{(i)}|\mathbf{Z}_{1:k-1}) \approx \sum_{i=1}^{N_{k-1}} \tilde{w}_{P,k|k-1}^{(i)} \delta_{\tilde{\mathbf{x}}_{P,k|k-1}^{(i)}}(\mathbf{x}_k), \quad (5.26)$$

$$D_{\Gamma,k|k-1}(\tilde{\mathbf{x}}_{\Gamma,k|k-1}^{(i)}|\mathbf{Z}_{1:k-1}) \approx \sum_{i=1}^{N_{k,B}} \tilde{w}_{\Gamma,k|k-1}^{(i)} \delta_{\tilde{\mathbf{x}}_{\Gamma,k}^{(i)}}(\mathbf{x}_k), \quad (5.27)$$

then the updated cardinality distribution $\rho_k(n|\mathbf{Z}_{1:k})$ keeps the same form (5.24), while the updated intensity $D_k(\mathbf{x}_k|\mathbf{Z}_{1:k})$ can now be written as:

THE MODIFIED CPHD INTENSITY UPDATE

$$D_{P,k}(\tilde{\mathbf{x}}_{P,k}^{(i)}|\mathbf{Z}_{1:k}) \approx \sum_{i=1}^{N_{k-1}} \tilde{w}_{P,k}^{(i)} \delta_{\tilde{\mathbf{x}}_{P,k|k-1}^{(i)}}(\mathbf{x}_k), \quad (5.28)$$

where,

$$\begin{aligned} \tilde{w}_{P,k}^{(i)} &= \tilde{w}_{P,k|k-1}^{(i)} \left[(1 - p_D(\tilde{\mathbf{x}}_{P,k}^{(i)})) \frac{\langle \Upsilon_k^1[w_{k|k-1}, Z_k], \rho_{k|k-1} \rangle}{\langle \Upsilon_k^0[w_{k|k-1}, Z_k], \rho_{k|k-1} \rangle} \right. \\ &\quad \left. + \sum_{z \in Z_k} \frac{\langle \Upsilon_k^1[w_{k|k-1}, Z_k - \{z\}], \rho_{k|k-1} \rangle}{\langle \Upsilon_k^0[w_{k|k-1}, Z_k], \rho_{k|k-1} \rangle} \frac{\langle 1, \kappa_k \rangle}{\kappa_k(z)} p_D(\tilde{\mathbf{x}}_{P,k}^{(i)}) g_k(z|\tilde{\mathbf{x}}_{P,k}^{(i)}) \right], \end{aligned}$$

and

$$D_{\Gamma,k}(\tilde{\mathbf{x}}_{\Gamma,k}^{(i)}|\mathbf{Z}_{1:k}) \approx \sum_{i=1}^{N_{k,B}} \tilde{w}_{\Gamma,k}^{(i)} \delta_{\tilde{\mathbf{x}}_{\Gamma,k}^{(i)}}(\mathbf{x}_k), \quad (5.29)$$

where,

$$\tilde{w}_{\Gamma,k}^{(i)} = \sum_{z \in Z_k} \frac{\langle \Upsilon_k^1[w_{k|k-1}, Z_k - \{z\}], \rho_{k|k-1} \rangle}{\langle \Upsilon_k^0[w_{k|k-1}, Z_k], \rho_{k|k-1} \rangle} \frac{\langle 1, \kappa_k \rangle}{\kappa_k(z)} p_D(\tilde{\mathbf{x}}_{\Gamma,k}^{(i)}) g_k(z|\tilde{\mathbf{x}}_{\Gamma,k}^{(i)}) \gamma_k(\tilde{\mathbf{x}}_{\Gamma,k}^{(i)}),$$

with,

$$\Upsilon_k^u[w, Z](n) = \sum_{j=0}^{\min(|Z|, n)} (|Z| - j)! \rho_{\Theta,k}(|Z| - j) P_{j+u}^n \text{esf}_j(\Xi_k(w, Z)) \frac{\left(\sum_{i=1}^{N_{k-1}} (1 - p_D(\tilde{\mathbf{x}}_{P,k|k-1}^{(i)})) \tilde{w}_{P,k|k-1}^{(i)} \right)^{n-(j+u)}}{\left(\sum_{i=1}^{N_{k-1}} \tilde{w}_{P,k|k-1}^{(i)} + \sum_{i=1}^{N_{k,B}} \tilde{w}_{\Gamma,k}^{(i)} \right)^n} \quad (5.30)$$

and

$$\Xi_k(w, Z) = \frac{\langle 1, \kappa_k \rangle}{\kappa_k(z)} \sum_{i=1}^{N_{k-1}} p_D(\tilde{\mathbf{x}}_{P,k|k-1}^{(i)}) g_k(z|\tilde{\mathbf{x}}_{P,k|k-1}^{(i)}) \tilde{w}_{P,k|k-1}^{(i)} + \sum_{i=1}^{N_{k,B}} \tilde{w}_{\Gamma,k}^{(i)} : z \in Z. \quad (5.31)$$

Finally, the new born particles become alive at the next time step, this is carried out by taking the union of the two particle sets, that is

$$\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}_{i=1}^{N_k} = \{\mathbf{x}_{P,k}^{(i)}, w_{P,k}^{(i)}\}_{i=1}^{N_{k,P}} \cup \{\mathbf{x}_{\Gamma,k}^{(i)}, w_{\Gamma,k}^{(i)}\}_{i=1}^{N_{k,B}}. \quad (5.32)$$

The pseudo-code of the corresponding SMC-CPHD filter is given in Algorithm 13.

Algorithm 13: Pseudo-code of the SMC-CPHD filter

input : $\{\mathbf{x}_{k-1}^{(i)}, w_{k-1}^{(i)}\}_{i=1}^{N_{k-1}}$, $\rho_{k-1}(n|\mathbf{Z}_{1:k-1})$ and a new measurement, \mathbf{Z}_k
output: $\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}_{i=1}^{N_k}$

Initialization at $k = 0$
 Draw from a prior distribution N_0 particles.
 Set the initial weights $w_0^{(i)}$ to $\frac{\hat{n}_0}{N_0}$, with \hat{n}_0 the expected initial number of targets.

At time $k > 0$,
 1 - *Prediction*
 Predict the cardinality distribution $\rho_{k-1}(n|\mathbf{Z}_{1:k-1})$ to $\rho_{k|k-1}(n|\mathbf{Z}_{1:k-1})$, using eq.(5.21).
for $i \leftarrow 1$ **to** N_{k-1} **do**
 Sample $\tilde{\mathbf{x}}_{P,k|k-1}^{(i)}$ from a proposal density $q_k(\cdot|\mathbf{x}_{k-1}^{(i)}, \mathbf{Z}_k)$.
 Compute the predicted weights $\tilde{w}_{P,k|k-1}^{(i)} = w_{k-1}^{(i)} \frac{ps(\mathbf{x}_{k-1}^{(i)}) \pi_{k|k-1}(\tilde{\mathbf{x}}_{P,k|k-1}^{(i)}|\mathbf{x}_{k-1}^{(i)})}{q_k(\tilde{\mathbf{x}}_{P,k|k-1}^{(i)}|\mathbf{x}_{k-1}^{(i)}, \mathbf{Z}_k)}$.
end
 Let $\mathbf{Z}_k = \{\mathbf{z}_{k,1}, \dots, \mathbf{z}_{k,m_k}\}$ be the observations received at time step k .
 $N_{k,B} = M_{k,B} \cdot m_k$
foreach observation $\mathbf{z}_{k,j}$ in \mathbf{Z}_k **do** **for** $l \leftarrow 1$ **to** $M_{k,B}$ **do**
 Sample $\tilde{\mathbf{x}}_{\Gamma,k}^{(i)}$ from another proposal density $b_k(\cdot|\mathbf{Z}_k)$.
 Compute the weights of new born particles, $\tilde{w}_{\Gamma,k}^{(i)} = \frac{\hat{n}_{k,B}}{N_{k,B}}$.
end

2 - *Update*
 Update the cardinality distribution $\rho_{k|k-1}(n|\mathbf{Z}_{1:k-1})$ to $\rho_k(n|\mathbf{Z}_{1:k})$, using eq.(5.24).
for $i \leftarrow 1$ **to** N_{k-1} **do**
 Update the weights $\tilde{w}_{P,k|k-1}^{(i)}$ to $\tilde{w}_{P,k}^{(i)}$, using eq.(5.28).
end
for $i \leftarrow 1$ **to** $N_{k,B}$ **do**
 Update the weights $\tilde{w}_{\Gamma,k|k-1}^{(i)}$ to $\tilde{w}_{\Gamma,k}^{(i)}$, using eq.(5.29).
end

3 - *Estimation of target location*
 Compute the total mass of each type of particles $\nu_{P,k} = \sum_{i=1}^{N_{k-1}} \tilde{w}_{P,k}^{(i)}$ and $\nu_{\Gamma,k} = \sum_{i=1}^{N_{k,B}} \tilde{w}_{\Gamma,k}^{(i)}$.
 Target estimates are found by fitting a Gaussian mixture model to the particles where the number of components in the mixture is the expected number of targets at time k , $\lfloor \nu_{\cdot,k} \rfloor$
 // $\lfloor \cdot \rfloor$ denotes the nearest integer

If there are no more incoming measurements then exit, otherwise proceed.

4 - *Resampling*
 Resample $\{\tilde{w}_{P,k}^{(i)}/\nu_{P,k}, \tilde{\mathbf{x}}_{P,k|k-1}^{(i)}\}_{i=1}^{N_{k-1}}$ to get $\{w_{P,k}^{(i)}, \mathbf{x}_{P,k}^{(i)}\}_{i=1}^{N_{k,P}}$, with $N_{k,P} = \lfloor M_{k,P} \cdot \nu_{P,k} \rfloor$ and $w_{P,k}^{(i)} = \frac{\nu_{P,k}}{N_{k,P}}$.
 Resample $\{\tilde{w}_{\Gamma,k}^{(i)}/\nu_{\Gamma,k}, \tilde{\mathbf{x}}_{\Gamma,k}^{(i)}\}_{i=1}^{N_{k,B}}$ to get $\{w_{\Gamma,k}^{(i)}, \mathbf{x}_{\Gamma,k}^{(i)}\}_{i=1}^{N_{k,B}}$, with $w_{\Gamma,k}^{(i)} = \frac{\nu_{\Gamma,k}}{N_{k,B}}$.

5 - *Union of particle sets*

$$\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}_{i=1}^{N_k} = \{\mathbf{x}_{P,k}^{(i)}, w_{P,k}^{(i)}\}_{i=1}^{N_{k,P}} \cup \{\mathbf{x}_{\Gamma,k}^{(i)}, w_{\Gamma,k}^{(i)}\}_{i=1}^{N_{k,B}}.$$

5.2 Cardinality Balanced Multi-target Multi-Bernoulli Filter

In addition to the PHD and CPHD filters, the Multi-target Multi-Bernoulli (MeMBer) filter has recently been proposed in [95, 137, 140] as a tractable approximation to the Bayes multi-target recursion, based on multi-Bernoulli RFS. The multi-Bernoulli representation allows reliable and inexpensive extraction of states estimates.

In this section, a brief summary of the original CARDINALITY-BALANCED MULTI-TARGET MULTI-BERNOULLI (CBMEMBER) [137, 140] recursion is given along with a description of a generic Sequential Monte Carlo (SMC) implementation of the multi-Bernoulli filter for point measurements. Moreover, a novel multi-Bernoulli update step, suitable for our specific TBD application, is proposed.

5.2.1 Cardinality Balanced Multi-target Multi-Bernoulli (CBMeMBer) Filter

Intuitively, the MeMBer recursion propagates the multi-target posterior probability density in time by propagating a finite but time-varying number of hypothesized tracks, each characterized by the probability of existence and the probability density of the current hypothesized state.

The multi-target RFS at each time step is approximated by a multi-Bernoulli RFS, based on the following modeling assumptions:

- Each target evolves and generates measurements independently of one another,
- Target births follow a multi-Bernoulli RFS independent of target survivals,
- Clutter follows a Poisson RFS(not too dense, i.e with parameter λ relatively small), and is independent of target-generated measurements RFSs (with reasonably high detection probability).

The MeMBer recursion, interpreted relative to the general multi-target Bayes recursion, is pictorially described below.

MULTI-TARGET BAYES RECURSION VS. MEMBER RECURSION:

$$\begin{array}{ccccccc}
 \dots \rightarrow & f_{k-1}(\mathbf{X}_{k-1} | \mathbf{Z}_{1:k-1}) & \xrightarrow{\text{prediction}} & f_{k|k-1}(\mathbf{X}_k | \mathbf{Z}_{1:k-1}) & \xrightarrow{\text{Bayes' rule}} & f_k(\mathbf{X}_k | \mathbf{Z}_{1:k}) & \rightarrow \dots \\
 & \downarrow & & \downarrow & & \downarrow & \\
 \dots \rightarrow & \{(r_{k-1}^{(i)}, s_{k-1}^{(i)})\}_{i=1}^{M_{k-1}} & \xrightarrow[\text{prediction}]{\text{MeMBer}} & \{(r_{k|k-1}^{(i)}, s_{k|k-1}^{(i)})\}_{i=1}^{M_{k|k-1}} & \xrightarrow[\text{update}]{\text{MeMBer}} & \{(r_k^{(i)}, s_k^{(i)})\}_{i=1}^{M_k} & \rightarrow \dots
 \end{array}$$

Suppose that the RFS of targets at time k is given by (3.12):

$$\Xi_{k|k-1} := \left[\bigcup_{\mathbf{x}_{k-1} \in \mathbf{X}_{k-1}} S_{k|k-1}(\mathbf{x}_{k-1}) \right] \cup \Gamma_k.$$

Proposition 5.2.1. *If at time $k-1$, the posterior multi-target density is a multi-Bernoulli of the form:*

$$\pi_{k-1} = \{(r_{k-1}^{(i)}, s_{k-1}^{(i)})\}_{i=1}^{M_{k-1}},$$

then the predicted multi-target density is also a multi-Bernoulli and is given by:

THE MEMBER PREDICTION

$$\pi_{k|k-1} = \underbrace{\{(r_{P,k|k-1}^{(i)}, s_{P,k|k-1}^{(i)})\}_{i=1}^{M_{k-1}}}_{\text{parameters for the surviving targets}} \cup \underbrace{\{(r_{\Gamma,k}^{(i)}, s_{\Gamma,k}^{(i)})\}_{i=1}^{M_{\Gamma,k}}}_{\text{parameters for the target births}}, \quad (5.33)$$

where,

$$r_{P,k|k-1}^{(i)} = r_{k-1}^{(i)} \langle s_{k-1}^{(i)}, p_S \rangle, \quad s_{P,k|k-1}^{(i)}(\mathbf{x}) = \frac{\langle \pi_{k|k-1}(\mathbf{x}|\cdot), s_{k-1}^{(i)} p_S \rangle}{\langle s_{k-1}^{(i)} p_{S,k|k-1} \rangle}, \quad (5.34)$$

with $\pi_{k|k-1}(\mathbf{x}|\mathbf{x}')$ the single-target Markov transition density from time $k-1$ to k and $p_S(\mathbf{x}') \stackrel{\text{abbr}}{=} p_{S,k|k-1}(\mathbf{x}')$ the probability that a target with state \mathbf{x}' at time-step $k-1$ will survive at time step k .

The total number of predicted hypothesized tracks is $M_{k|k-1} = M_{k-1} + M_{\Gamma,k}$.

Suppose now that a new observation-scan has been collected at time-step k by a single sensor. Let us consider the classical setup, that is, a pre-processing chain isolates detection plots before the tracking chain. Thus, the set of noisy observations \mathbf{Z}_k , which may include false alarms and have missed detections, is modeled by the RFS (5.7):

$$\Sigma_k = \left[\bigcup_{\mathbf{x}_k \in \mathbf{X}_k} \Delta_k(\mathbf{x}_k) \right] \cup \Theta_k.$$

Proposition 5.2.2. *If at time k , the predicted multi-target density is a multi-Bernoulli of the form:*

$$\pi_{k|k-1} = \{(r_{k|k-1}^{(i)}, s_{k|k-1}^{(i)})\}_{i=1}^{M_{k|k-1}},$$

then the posterior multi-target density can be approximated by a multi-Bernoulli as follows:

THE MEMBER UPDATE

$$\pi_k \approx \underbrace{\{(r_{L,k}^{(i)}, s_{L,k}^{(i)})\}_{i=1}^{M_{k|k-1}}}_{\text{parameters for the legacy tracks}} \cup \underbrace{\{(r_{U,k}(z), s_{U,k}(\cdot; z))\}_{z \in \mathbf{Z}_k}}_{\text{parameters for the measurement-corrected tracks}}, \quad (5.35)$$

where,

UPDATE FOR THE LEGACY (MISSED-DETECTIONS) TRACKS COMPONENTS

$$\begin{aligned} r_{L,k}^{(i)} &= r_{k|k-1}^{(i)} \frac{1 - \langle s_{k|k-1}^{(i)}, p_D \rangle}{1 - r_{k|k-1}^{(i)} \langle s_{k|k-1}^{(i)}, p_D \rangle}; \\ s_{L,k}^{(i)}(\mathbf{x}) &= s_{k|k-1}^{(i)} \frac{1 - p_D(\mathbf{x})}{1 - \langle s_{k|k-1}^{(i)}, p_D \rangle}; \end{aligned}$$

UPDATE FOR THE MEASUREMENT-CORRECTED TRACKS COMPONENTS

$$\begin{aligned}
r_{U,k}(z) &= \frac{\sum_{i=1}^{M_{k|k-1}} \frac{r_{k|k-1}^{(i)} (1-r_{k|k-1}^{(i)}) \langle s_{k|k-1}^{(i)}, g_k(z|\cdot) p_D \rangle}{(1-r_{k|k-1}^{(i)}) \langle s_{k|k-1}^{(i)}, p_D \rangle^2}}{\kappa_k(z) + \sum_{i=1}^{M_{k|k-1}} \frac{r_{k|k-1}^{(i)} \langle s_{k|k-1}^{(i)}, g_k(z|\cdot) p_D \rangle}{1-r_{k|k-1}^{(i)} \langle s_{k|k-1}^{(i)}, p_D \rangle}}; \\
s_{U,k}(\mathbf{x}; z) &= \frac{\sum_{i=1}^{M_{k|k-1}} \frac{r_{k|k-1}^{(i)}}{1-r_{k|k-1}^{(i)}} s_{k|k-1}^{(i)} g_k(z|\cdot) p_D}{\sum_{i=1}^{M_{k|k-1}} \frac{r_{k|k-1}^{(i)}}{1-r_{k|k-1}^{(i)}} \langle s_{k|k-1}^{(i)}, g_k(z|\cdot) p_D \rangle};
\end{aligned}$$

with $p_D(\mathbf{x}) \stackrel{abbr}{=} p_{D,k}(\mathbf{x})$ the probability that a target with state \mathbf{x} will be detected at time k , $g_k(z|\mathbf{x})$ the single-target measurement likelihood at time k and $\kappa_k(z)$ the intensity of clutter measurements at time k .

The total number of posterior hypothesized tracks is $M_k = M_{k|k-1} + M_{|\mathbf{Z}_k|}$.

While the time prediction step of the CBMeMBer recursion is exact, the data update step is based on the following approximation to the p.g.fl. of the posterior multi-target state at time k :

$$G_k[h] \approx \prod_{i=1}^{M_{k|k-1}} G_{L,k}^{(i)}[h] \prod_{z \in \mathbf{Z}_k} G_{U,k}[h; z],$$

where,

- The product $\prod_{i=1}^{M_{k|k-1}} G_{L,k}^{(i)}[h]$ corresponds to the set of legacy tracks and

$$G_{L,k}^{(i)}[h] = \frac{1 - r_{k|k-1}^{(i)} + r_{k|k-1}^{(i)} \langle s_{k|k-1}^{(i)}, h (1 - p_D) \rangle}{1 - r_{k|k-1}^{(i)} + s_{k|k-1}^{(i)} \langle s_{k|k-1}^{(i)}, 1 - p_D \rangle} \quad (5.36)$$

can be written in Bernoulli form

$$G_{L,k}^{(i)}[h] = 1 - r_{L,k}^{(i)} + r_{L,k}^{(i)} \langle s_{L,k}^{(i)}, h \rangle.$$

Hence the mean cardinality of the legacy tracks is:

$$\sum_{i=1}^{M_{k|k-1}} r_{L,k}^{(i)}.$$

- The product $\prod_{z \in \mathbf{Z}_k} G_{U,k}[h; z]$ corresponds to the set of measurement-updated tracks which is not a multi-Bernoulli RFS,

$$G_{U,k}[h; z] = \frac{\kappa_k(z) + \sum_{i=1}^{M_{k|k-1}} G_{U,k}^{(i)}[h; z]}{\kappa_k(z) + \sum_{i=1}^{M_{k|k-1}} G_{U,k}^{(i)}[1; z]}, \quad (5.37)$$

where,

$$G_{U,k}^{(i)}[h; z] = \frac{r_{k|k-1}^{(i)} \langle s_{k|k-1}^{(i)}, h g_k(z|\cdot) p_D \rangle}{1 - r_{k|k-1}^{(i)} + r_{k|k-1}^{(i)} \langle s_{k|k-1}^{(i)}, h (1 - p_D) \rangle}. \quad (5.38)$$

Nonetheless, the corresponding mean cardinality can be computed exactly. Substituting $h(x) = y$ into the p.g.fl. (5.37)-(5.38) then differentiating at $y = 1$, yields $G'_{U,k}[1; z] = r_{U,k}(z)$. It follows that the mean cardinality of the measurement-updated tracks is:

$$\sum_{z \in \mathbf{Z}_k} r_{U,k}(z).$$

Remark. This is a reasonable approximation when clutter is not too dense and the probability of detection is high. The complexity of the CBMeMBer recursion is linear in the number of targets and linear in the number of measurements. For further details cf. [95, 140].

The multi-Bernoulli representation $\pi_k = \{(r_k^{(i)}, s_k^{(i)})\}_{i=1}^{M_k}$ has an intuitive interpretation that facilitates multi-target state estimation from the posterior multi-target density. The existence probability $r_k^{(i)}$ indicates how likely the i^{th} hypothesized track is a true track, and the posterior density $s_k^{(i)}$ describes the estimated current state of the track. Hence, a multi-target state estimate can be obtained by choosing the means or modes from the posterior densities of the hypothesized tracks with existence probabilities exceeding a given threshold. Alternatively, the following basic two-stage procedure can be used. First, we estimate the number of targets from the posterior cardinality distribution by taking its mean or mode (more stable). Then, we take the corresponding number of hypothesized tracks having the highest probabilities of existence and compute the individual means or modes from the individual posterior densities.

The CBMeMBer recursions can be extended to propagate track continuity, by appropriately labeling and updating the individual Bernoulli components of the posterior multi-Bernoulli density.

Remark. Thus, given a multi-Bernoulli density $\pi_k = \{(r_k^{(i)}, s_k^{(i)})\}_{i=1}^{M_k}$, to each (Bernoulli) component $(r_k^{(i)}, s_k^{(i)})$ a unique track label $\ell_k^{(i)}$ is assigned to identify hypothesized tracks. While there are many possible schemes for propagating track labels, a simple scheme is presented as follows.

1. In prediction, surviving components retain their original labels, and new born components are assigned new labels.
2. In update, legacy components maintain their original labels, and measurement-updated components are assigned the label of the predicted track which has the largest contribution to the current measurement-updated probability of existence.

Although simple to implement, the proposed scheme is expected to perform poorly when targets are close together. To solve this, a more advanced label management scheme can be adopted [128].

5.2.2 Particle CBMeMber Filter Implementation

In the following, a generic SMC implementation of the CBMeMber recursion is described. The basic strategy is to propagate multiple sets of weights and samples, where each set represents each hypothesized track, and to propagate these recursively in time.

Suppose that at time $k - 1$ the multi-Bernoulli multi-target prior density,

$$\pi_{k-1} = \{(r_{k-1}^{(i)}, s_{k-1}^{(i)})\}_{i=1}^{M_{k-1}},$$

is given and each $s_{k-1}^{(i)}$ for $i = 1, \dots, M_{k-1}$, is represented by a set of weighted samples $\{\mathbf{x}_{k-1}^{(i,j)}, w_{k-1}^{(i,j)}\}_{j=1}^{N_{k-1}^{(i)}}$, i.e.

$$s_{k-1}^{(i)}(\mathbf{x}_{k-1}) \approx \sum_{j=1}^{N_{k-1}^{(i)}} w_{k-1}^{(i,j)} \delta_{\mathbf{x}_{k-1}^{(i,j)}}(\mathbf{x}_{k-1}). \quad (5.39)$$

Then, given importance (or proposal) densities $q_k^{(i)}(\cdot|\mathbf{x}_{k-1}, \mathbf{Z}_k)$ and $b_k^{(i)}(\cdot|\mathbf{Z}_k)$, the predicted multi-Bernoulli multi-target density,

$$\pi_{k|k-1} = \{(r_{P,k|k-1}^{(i)}, s_{P,k|k-1}^{(i)})\}_{i=1}^{M_{k-1}} \cup \{(r_{\Gamma,k}^{(i)}, s_{\Gamma,k}^{(i)})\}_{i=1}^{M_{\Gamma,k}},$$

can be computed as follows:

$$r_{P,k|k-1}^{(i)} = r_{k-1}^{(i)} \sum_{j=1}^{N_{k-1}^{(i)}} w_{k-1}^{(i,j)} p_S(\mathbf{x}_{k-1}^{(i,j)}), \quad s_{P,k|k-1}^{(i)}(\mathbf{x}) = \sum_{j=1}^{N_{k-1}^{(i)}} w_{P,k|k-1}^{(i,j)} \delta_{\mathbf{x}_{P,k|k-1}^{(i,j)}}(\mathbf{x}), \quad (5.40)$$

where,

$$\begin{aligned} \mathbf{x}_{P,k|k-1}^{(i,j)} &\sim q_k^{(i)}(\cdot|\mathbf{x}_{k-1}^{(i,j)}, \mathbf{Z}_k), \forall j = 1, \dots, N_{k-1}^{(i)}, \\ \tilde{w}_{P,k|k-1}^{(i,j)} &= w_{k-1}^{(i,j)} \frac{p_S(\mathbf{x}_{k-1}^{(i,j)}) \pi_{k|k-1}(\mathbf{x}_{P,k|k-1}^{(i,j)}|\mathbf{x}_{k-1}^{(i,j)})}{q_k^{(i)}(\mathbf{x}_{P,k|k-1}^{(i,j)}|\mathbf{x}_{k-1}^{(i,j)}, \mathbf{Z}_k)}, \quad w_{P,k|k-1}^{(i,j)} = \frac{\tilde{w}_{P,k|k-1}^{(i,j)}}{\sum_{j=1}^{N_{k-1}^{(i)}} \tilde{w}_{P,k|k-1}^{(i,j)}}, \end{aligned}$$

and

$$r_{\Gamma,k}^{(i)} \text{ parameter given by the birth model,} \quad s_{\Gamma,k}^{(i)}(\mathbf{x}) = \sum_{j=1}^{N_{k,B}^{(i)}} w_{\Gamma,k}^{(i,j)} \delta_{\mathbf{x}_{\Gamma,k}^{(i,j)}}(\mathbf{x}), \quad (5.41)$$

where,

$$\begin{aligned} \mathbf{x}_{\Gamma,k}^{(i,j)} &\sim b_k^{(i)}(\cdot|\mathbf{Z}_k), \forall j = 1, \dots, N_{k,B}^{(i)}, \\ \tilde{w}_{\Gamma,k}^{(i,j)} &= \frac{s_{\Gamma,k}^{(i)}(\mathbf{x}_{\Gamma,k}^{(i,j)})}{b_k^{(i)}(\mathbf{x}_{\Gamma,k}^{(i,j)}|\mathbf{Z}_k)}, \quad w_{\Gamma,k}^{(i,j)} = \frac{\tilde{w}_{\Gamma,k}^{(i,j)}}{\sum_{j=1}^{N_{k,B}^{(i)}} \tilde{w}_{\Gamma,k}^{(i,j)}}. \end{aligned}$$

Suppose that at time k the predicted multi-Bernoulli multi-target density,

$$\pi_{k|k-1} = \{(r_{k|k-1}^{(i)}, s_{k|k-1}^{(i)})\}_{i=1}^{M_{k|k-1}},$$

is given and each $s_{k|k-1}^{(i)}$ for $i = 1, \dots, M_{k|k-1}$, is represented by a set of weighted samples: $\{\mathbf{x}_{k|k-1}^{(i,j)}, w_{k|k-1}^{(i,j)}\}_{j=1}^{N_{k|k-1}^{(i)}}$, i.e.

$$s_{k|k-1}^{(i)}(\mathbf{x}_{k|k-1}) \approx \sum_{j=1}^{N_{k|k-1}^{(i)}} w_{k|k-1}^{(i,j)} \delta_{\mathbf{x}_{k|k-1}^{(i,j)}}(\mathbf{x}_{k|k-1}). \quad (5.42)$$

Then, the updated multi-Bernoulli multi-target density,

$$\pi_k = \{(r_{L,k}^{(i)}, s_{L,k}^{(i)})\}_{i=1}^{M_{k|k-1}} \cup \{(r_{U,k}(z), s_{U,k}(\cdot; z))\}_{z \in \mathbf{Z}_k},$$

can be computed as follows:

$$r_{L,k}^{(i)} = r_{k|k-1}^{(i)} \frac{1 - \sum_{j=1}^{N_{k|k-1}^{(i)}} w_{k|k-1}^{(i,j)} p_D(\mathbf{x}_{k|k-1}^{(i,j)})}{1 - r_{k|k-1}^{(i)} \sum_{j=1}^{N_{k|k-1}^{(i)}} w_{k|k-1}^{(i,j)} p_D(\mathbf{x}_{k|k-1}^{(i,j)})}, \quad s_{L,k}^{(i)}(\mathbf{x}) = \sum_{j=1}^{N_{k|k-1}^{(i)}} w_{L,k}^{(i,j)} \delta_{\mathbf{x}_{k|k-1}^{(i,j)}}(\mathbf{x}), \quad (5.43)$$

where,

$$\tilde{w}_{L,k}^{(i,j)} = w_{k|k-1}^{(i,j)} (1 - p_D(\mathbf{x}_{k|k-1}^{(i,j)})), \quad w_{L,k}^{(i,j)} = \frac{\tilde{w}_{L,k}^{(i,j)}}{\sum_{j=1}^{N_{k|k-1}^{(i)}} \tilde{w}_{L,k}^{(i,j)}},$$

and

$$r_{U,k}(z) = \frac{\sum_{i=1}^{M_{k|k-1}} \frac{r_{k|k-1}^{(i)} (1 - r_{k|k-1}^{(i)}) \sum_{j=1}^{N_{k|k-1}^{(i)}} w_{k|k-1}^{(i,j)} g_k(z | \mathbf{x}_{k|k-1}^{(i,j)}) p_D(\mathbf{x}_{k|k-1}^{(i,j)})}{\left(1 - r_{k|k-1}^{(i)} \sum_{j=1}^{N_{k|k-1}^{(i)}} w_{k|k-1}^{(i,j)} p_D(\mathbf{x}_{k|k-1}^{(i,j)})\right)^2}}{\kappa_k(z) + \sum_{i=1}^{M_{k|k-1}} \frac{r_{k|k-1}^{(i)} \sum_{j=1}^{N_{k|k-1}^{(i)}} w_{k|k-1}^{(i,j)} g_k(z | \mathbf{x}_{k|k-1}^{(i,j)}) p_D(\mathbf{x}_{k|k-1}^{(i,j)})}{1 - r_{k|k-1}^{(i)} \sum_{j=1}^{N_{k|k-1}^{(i)}} w_{k|k-1}^{(i,j)} p_D(\mathbf{x}_{k|k-1}^{(i,j)})}}, \quad (5.44)$$

$$s_{U,k}(\mathbf{x}; z) = \sum_{i=1}^{M_{k|k-1}} \sum_{j=1}^{N_{k|k-1}^{(i)}} w_{U,k}^{(i,j)}(z) \delta_{\mathbf{x}_{k|k-1}^{(i,j)}}(\mathbf{x}), \quad (5.45)$$

where,

$$\tilde{w}_{U,k}^{(i,j)}(z) = w_{k|k-1}^{(i,j)} \frac{r_{k|k-1}^{(i)}}{1 - r_{k|k-1}^{(i)}} g_k(z | \mathbf{x}_{k|k-1}^{(i,j)}) p_D(\mathbf{x}_{k|k-1}^{(i,j)}), \quad w_{U,k}^{(i,j)}(z) = \frac{\tilde{w}_{U,k}^{(i,j)}(z)}{\sum_{i=1}^{M_{k|k-1}} \sum_{j=1}^{N_{k|k-1}^{(i)}} \tilde{w}_{U,k}^{(i,j)}(z)}.$$

To reduce the effect of degeneracy, for each hypothesized track, the particles are then resampled. The number of particles given to each hypothesized track is reallocated in proportion to its probability of existence $r_k^{(i)}$. To reduce the growing number of tracks (and particles), targets with existence probabilities below a predefined threshold are discarded. For further details cf. [140].

5.2.3 Multi-Bernoulli TBD Filter

Recently, a multi-Bernoulli filter suitable for TBD application was proposed in [141]. The multi-Bernoulli TBD filter requires the assumption of a separable multi-target likelihood, that is of the form:

$$\vartheta(\mathbf{z}|\mathbf{X}) = f(\mathbf{z}) \prod_{x \in \mathbf{X}} g_{\mathbf{z}}(x), \quad (5.46)$$

where $f(\cdot)$ is probability density independent of \mathbf{X} , and $g_{\mathbf{z}}(\cdot)$ is a probability density parameterized by \mathbf{z} .

Note that the multi-target likelihood, considered in this work, given by (3.18):

$$\vartheta_k(\mathbf{z}_k|\mathbf{X}_k) = f(\mathbf{z}_k) \prod_{x \in \mathbf{X}_k} g_{\mathbf{z}_k}(x),$$

where,

$$\begin{aligned} f(\mathbf{z}_k) &= \prod_{c=1}^m \varphi(\mathbf{z}_k^c), \\ g_{\mathbf{z}_k}(x) &= \prod_{c \in C(x)} \mathfrak{L}(\mathbf{z}_k^c|\mathbf{X}_k), \end{aligned}$$

fulfills the criterion of separability.

Assuming likelihood separability, the posterior density of the multi-Bernoulli multi-target RFS admits a closed form recursion by using the Chapman-Kolmogorov equation and the Bayes update equation, i.e. the multi-Bernoulli RFS density admits an exact prediction and update no approximations such cardinality balancing.

The SMC implementation of the MeMber-TBD filter is described pictorially below.

$$\begin{array}{ccccccc} \dots \rightarrow & f_{k-1}(\mathbf{X}_{k-1}|\mathbf{Z}_{1:k-1}) & \xrightarrow{\text{prediction}} & f_{k|k-1}(\mathbf{X}_k|\mathbf{Z}_{1:k-1}) & \xrightarrow{\text{Bayes' rule}} & f_k(\mathbf{X}_k|\mathbf{Z}_{1:k}) & \rightarrow \dots \\ & \downarrow & & \downarrow & & \downarrow & \\ \dots \rightarrow & \{(r_{k-1}^{(i)}, s_{k-1}^{(i)})\}_{i=1}^{M_{k-1}} & \xrightarrow[\text{prediction}]{\text{MeMber TBD}} & \{(r_{k|k-1}^{(i)}, s_{k|k-1}^{(i)})\}_{i=1}^{M_{k|k-1}} & \xrightarrow[\text{update}]{\text{MeMber TBD}} & \{(r_k^{(i)}, s_k^{(i)})\}_{i=1}^{M_k} & \rightarrow \dots \end{array}$$

Suppose that the RFS of targets at time k is given by (3.12):

$$\Xi_{k|k-1} := \left[\bigcup_{\mathbf{x}_{k-1} \in \mathbf{X}_{k-1}} S_{k|k-1}(\mathbf{x}_{k-1}) \right] \cup \Gamma_k.$$

Proposition 5.2.3. *If at time $k-1$, the posterior multi-target density is a multi-Bernoulli of the form:*

$$\pi_{k-1} = \{(r_{k-1}^{(i)}, s_{k-1}^{(i)})\}_{i=1}^{M_{k-1}},$$

with $r_{k-1}^{(i)}$ the existence probability and $s_{k-1}^{(i)}$ the spatial PDF of target i at time $k-1$. Then the predicted multi-target density is also a multi-Bernoulli and is given by:

THE MEMBER TBD PREDICTION

$$\pi_{k|k-1} = \underbrace{\{(r_{P,k|k-1}^{(i)}, s_{P,k|k-1}^{(i)})\}_{i=1}^{M_{k-1}}}_{\text{parameters for the surviving targets}} \cup \underbrace{\{(r_{\Gamma,k}^{(i)}, s_{\Gamma,k}^{(i)})\}_{i=1}^{M_{\Gamma,k}}}_{\text{parameters for the target births}}, \quad (5.47)$$

where,

$$r_{P,k|k-1}^{(i)} = r_{k-1}^{(i)} \langle s_{k-1}^{(i)}, p_S \rangle, \quad (5.48)$$

$$s_{P,k|k-1}^{(i)}(\mathbf{x}) = \frac{\langle \pi_{k|k-1}(\mathbf{x}|\cdot), s_{k-1}^{(i)} p_S \rangle}{\langle s_{k-1}^{(i)} p_{S,k|k-1} \rangle}, \quad (5.49)$$

with $\pi_{k|k-1}(\mathbf{x}|\mathbf{x}')$ the single-target Markov transition density from time $k-1$ to k and $p_S(\mathbf{x}') \stackrel{\text{abbr}}{=} p_{S,k|k-1}(\mathbf{x}')$ the probability that a target with state \mathbf{x}' at time-step $k-1$ will survive at time step k .

Suppose now that a new observation-scan has been collected at time-step k by a single sensor. Let us consider the TBD approach, that is, the tracking process is performed on the basis of raw measurements, e.g. the reflected power of the targets plus noise.

Proposition 5.2.4. *If at time k , the predicted multi-target density is a multi-Bernoulli of the form*

$$\pi_{k|k-1} = \{(r_{k|k-1}^{(i)}, s_{k|k-1}^{(i)})\}_{i=1}^{M_{k|k-1}},$$

then the posterior multi-target density is also a multi-Bernoulli and is given by:

THE MEMBER TBD UPDATE

$$\pi_k = \{(r_k^{(i)}, s_k^{(i)})\}_{i=1}^{M_{k|k-1}}, \quad (5.50)$$

where, each component is re-weighted using the likelihood ratio $g_{\mathbf{z}_k}(\cdot)$

$$r_k^{(i)} = \frac{r_{k|k-1}^{(i)} \langle s_{k|k-1}^{(i)}, g_{\mathbf{z}_k}(\cdot) \rangle}{1 - r_{k|k-1}^{(i)} + r_{k|k-1}^{(i)} \langle s_{k|k-1}^{(i)}, g_{\mathbf{z}_k}(\cdot) \rangle}, \quad (5.51)$$

$$s_k^{(i)}(\mathbf{x}) = \frac{s_{k|k-1}^{(i)}(\mathbf{x}) g_{\mathbf{z}_k}(\mathbf{x})}{\langle s_{k|k-1}^{(i)}, g_{\mathbf{z}_k}(\cdot) \rangle}. \quad (5.52)$$

Remark. Future work will concentrate on extending the proposed approach to accommodate a more general form of the likelihood.

5.2.4 Particle Multi-Bernoulli TBD Filter Implementation

A generic SMC implementation of the MeMber TBD recursion is described in Algorithm 14.

Algorithm 14: Pseudo-code of the SMC-MemBer TBD filter

input : $\{(r_{k-1}^{(i)}, s_{k-1}^{(i)})\}_{i=1}^{M_{k-1}}$ and a new measurement, \mathbf{z}_k
output: $\{(r_k^{(i)}, s_k^{(i)})\}_{i=1}^{M_k}$

At time $k \geq 1$,
for $i \leftarrow 1$ **to** M_{k-1} **do**
 1 - *Prediction Step*:
 for $j \leftarrow 1$ **to** $N_{k-1}^{(i)}$ **do**
 Sample $\mathbf{x}_{P,k|k-1}^{(i,j)} \sim q_k^{(i)}(\cdot | \mathbf{x}_{k-1}^{(i,j)}, \mathbf{Z}_k)$.
 Set weights : $\tilde{w}_{P,k|k-1}^{(i,j)} = w_{k-1}^{(i,j)} \frac{p_S(\mathbf{x}_{k-1}^{(i,j)}) \pi_{k|k-1}(\mathbf{x}_{P,k|k-1}^{(i,j)} | \mathbf{x}_{k-1}^{(i,j)})}{q_k^{(i)}(\mathbf{x}_{P,k|k-1}^{(i,j)} | \mathbf{x}_{k-1}^{(i,j)}, \mathbf{Z}_k)}$.
 end
 Normalize the weights : $\sum_{j=1}^{N_{k-1}^{(i)}} \tilde{w}_{P,k|k-1}^{(i,j)} = 1$.
 Set $r_{P,k|k-1}^{(i)} = r_{k-1}^{(i)} \sum_{j=1}^{N_{k-1}^{(i)}} w_{k-1}^{(i,j)} p_S(\mathbf{x}_{k-1}^{(i,j)})$.
 Set $r_{\Gamma,k}^{(i)}$ according to the birth model parameters.
 Set the number of particles per birth term: $N_{k,B}^{(i)} = r_{\Gamma,k}^{(i)} L_{max}$. **for** $j \leftarrow 1$ **to** $N_{k,B}^{(i)}$ **do**
 Sample $\mathbf{x}_{\Gamma,k}^{(i,j)} \sim b_k^{(i)}(\cdot | \mathbf{Z}_k)$.
 Set weights : $\tilde{w}_{\Gamma,k}^{(i,j)} = \frac{s_{\Gamma,k}^{(i)}(\mathbf{x}_{\Gamma,k}^{(i,j)})}{b_k^{(i)}(\mathbf{x}_{\Gamma,k}^{(i,j)} | \mathbf{Z}_k)}$.
 end
 Normalize the weights : $\sum_{j=1}^{N_{k,B}^{(i)}} \tilde{w}_{\Gamma,k}^{(i,j)} = 1$.
 2 - *Union of particle sets*

$$\{\mathbf{x}_{k|k-1}^{(i,j)}, w_{k|k-1}^{(i,j)}\}_{j=1}^{N_{k|k-1}^{(i)}} = \{\mathbf{x}_{P,k|k-1}^{(i,j)}, w_{P,k|k-1}^{(i,j)}\}_{j=1}^{N_{k-1}^{(i)}} \cup \{\mathbf{x}_{\Gamma,k}^{(i,j)}, w_{\Gamma,k}^{(i,j)}\}_{j=1}^{N_{k,B}^{(i)}}$$

 3 - *Update Step*:
 for $j \leftarrow 1$ **to** $N_{k|k-1}^{(i)}$ **do**
 Re-weight each component: $\tilde{w}_k^{(i,j)} = w_{k|k-1}^{(i,j)} g_{\mathbf{z}_k}(\mathbf{x}_{k|k-1}^{(i,j)})$.
 end
 Set $r_k^{(i)} = \frac{r_{k|k-1}^{(i)} \sum_{j=1}^{N_{k|k-1}^{(i)}} \tilde{w}_k^{(i,j)}}{1 - r_{k|k-1}^{(i)} + r_{k|k-1}^{(i)} \sum_{j=1}^{N_{k|k-1}^{(i)}} \tilde{w}_k^{(i,j)}}$.
 Normalize the weights : $\sum_{j=1}^{N_{k|k-1}^{(i)}} \tilde{w}_k^{(i,j)} = 1$.
 4 - *Resampling Step*:
 Set the number of particles per updated track $N_k^{(i)} = \max(r_k^{(i)} L_{max}, L_{max})$.
 for $j \leftarrow 1$ **to** $N_{k|k-1}^{(i)}$ **do**
 Resample $\{\mathbf{x}_{k|k-1}^{(i,j)}, \tilde{w}_k^{(i,j)}\}_{j=1}^{N_{k|k-1}^{(i)}}$ to get $\{\mathbf{x}_{k-1}^{(i,j)}, w_k^{(i,j)}\}_{j=1}^{N_k^{(i)}}$.
 end
end
5 - *Pruning Step*:
Discard the hypothesized tracks i with existence probabilities $r_k^{(i)}$ below a threshold τ_P , that is,
 M_k hypothesized tracks are kept at the next time step.

5.3 Numerical Study

In this section, we illustrate the performance of the SMC-CBMeMber filter proposed in section 5.2 with two separate examples. A performance comparison with the PHD (Algorithm 12), the CPHD (Algorithm 13) and the generic SIR-PF (Algorithm 2) is also shown. We evaluate filter performance using the OSPA multi-target miss-distance, which jointly captures cardinality errors and state errors meaningfully.

5.3.1 Scenario 1: Plot tracking using SMC implementation

We demonstrate the performance of the SMC-CBMeMber filter and compare its performance with SMC implementations of the PHD (Algorithm 12) and CPHD (Algorithm 13) filters. Consider a non-linear bearing and range tracking example with a time varying number of targets observed in clutter. The observation region covers the half disc of radius 2000 m. A maximum of 10 targets appears on the scene for any given instant with various births and deaths throughout the scenario duration of $T_{max} = 100$ s. An NCT model (cf. paragraph 4.6.1) having varying turn rate together with noisy bearing and range measurements is considered. The single target state $\mathbf{x}_{k,j}$ comprises the position and velocity of the j^{th} target, $s_{k,j} = [x_{k,j}, \dot{x}_{k,j}, y_{k,j}, \dot{y}_{k,j}]$ as well as the turn rate, $\omega_{k,j}$. The state transition model is:

$$\mathbf{s}_{k,j} = F(\omega_{k-1,j}) \mathbf{s}_{k-1,j} + G \mathbf{v}_{k-1,j}, \quad (5.53)$$

$$\omega_{k,j} = \omega_{k-1,j} + \Delta T \mathbf{u}_{k-1,j}, \quad (5.54)$$

where the transition matrix F is given by:

$$F(\omega) = \begin{bmatrix} 1 & \frac{\sin \omega \Delta T}{\omega} & 0 & -\frac{1 - \cos \omega \Delta T}{\omega} \\ 0 & \cos \omega \Delta T & 0 & -\sin \omega \Delta T \\ 0 & \frac{1 - \cos \omega \Delta T}{\omega} & 1 & \frac{\sin \omega \Delta T}{\omega} \\ 0 & \sin \omega \Delta T & 0 & \cos \omega \Delta T \end{bmatrix}, \quad (5.55)$$

and G , the noise gain, is given by:

$$G = \begin{bmatrix} \frac{\Delta_T^2}{2} & 0 \\ \Delta_T & 0 \\ 0 & \frac{\Delta_T^2}{2} \\ 0 & \Delta_T \end{bmatrix}, \quad (5.56)$$

with $\Delta_T = 1$ s, $\mathbf{v}_{k,j} \sim \mathcal{N}(\cdot; 0, \sigma_v^2)$, $\sigma_v = 15$ m.s⁻², $\mathbf{u}_{k,j} \sim \mathcal{N}(\cdot; 0, \sigma_u^2)$ and $\sigma_u = \pi/180$ rad.s⁻¹. The death process is specified by a uniform single target probability of target survival $p_{S,k}(\mathbf{x}_{k,j}) = 0.99$. If the object is detected, the observation is a noisy bearing and range vector given by:

$$\mathbf{z}_{k,j} = \begin{bmatrix} \arctan(y_{k,j}/x_{k,j}) \\ \sqrt{x_{k,j}^2 + y_{k,j}^2} \end{bmatrix} + \epsilon_{k,j}, \quad (5.57)$$

where $\epsilon_{k,j} \sim \mathcal{N}(\cdot; 0, R_k)$, with $R_k = \text{diag}([\sigma_b^2 \sigma_r^2]^T)$, $\sigma_b = \pi/180$ rad and $\sigma_r = 5$ m.

The birth process is a Multi-Bernoulli density $\pi_\Gamma = \{(p_{B,k}(j), b_k^{(j)})\}_{j=1}^4$ where $p_{B,k}(j)$ is the probability of existence for term j and $b_k^{(j)}(x) = \mathcal{N}(x; m_\gamma^{(j)}, P_\gamma)$ with $m_\gamma^{(j)}$ is the estimated state of the j^{th} detection plot and $P_\gamma = \text{diag}([50, 50, 50, 50, 6\pi/180]^T)^2$.

Table 5.1: Bernoulli parameter set

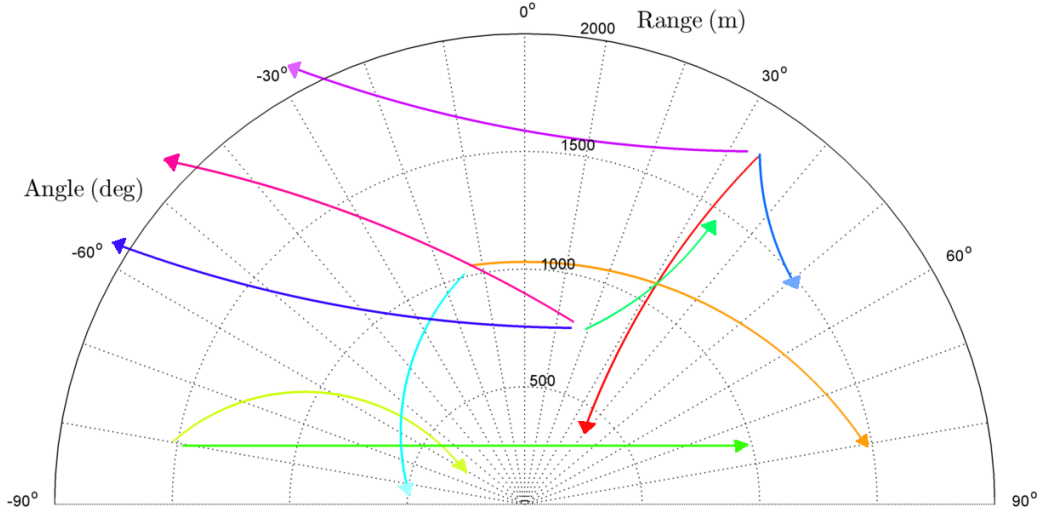
Component j	$p_{B,k}(j)$	$m_\gamma^{(j)}$
1	0.02	$[-1500, 0, 250, 0, 0]$
2	0.02	$[-250, 0, 1000, 0, 0]$
3	0.03	$[250, 0, 750, 0, 0]$
4	0.03	$[1000, 0, 1500, 0, 0]$

The probability of detection is state dependent and defined as:

$$p_{D,k}(\mathbf{x}_{k,j}) = \frac{0.98 \mathcal{N}([x_{k,j}, y_{k,j}]; 0, 6000^2 I_2)}{\mathcal{N}(0; 0, 6000^2 I_2)}.$$

Clutter follows a Poisson RFS with intensity $\lambda_c = 1.6 \times 10^{-3} \text{ (rad.m)}^{-1}$ over the half disc $[-\pi/2, \pi/2]$ rad $\times [0, 2000]$ m (giving an average of 10 clutter points per scan).

The true trajectories¹ are shown in Figure 5.1.

**Figure 5.1:** True tracks in polar coordinates for Scenario 1.

At each time step, in the SMC implementation of the CBMeMber filter, the state transition model is used as the proposal for each hypothesized track. A maximum of $L_{\max} = 1000$ and a minimum of $L_{\min} = 300$ particles per hypothesized track is imposed, with resampling performed so that the number of particles representing each track is proportional to its probability of existence. Additionally, pruning of hypothesized tracks is performed with a weight threshold of $\tau_P = 10^{-3}$ and a maximum of $H_{\max} = 100$ tracks.

¹This scenario originally appeared in [140, p. 9].

In Figure 5.2, the x and y components of the measurements, true trajectories, and SMC-CBMeMber filter estimates are shown versus time. It can be seen that the SMC-CBMeMber filter is able to detect all target births and deaths, as well as successfully accommodating nonlinearities.

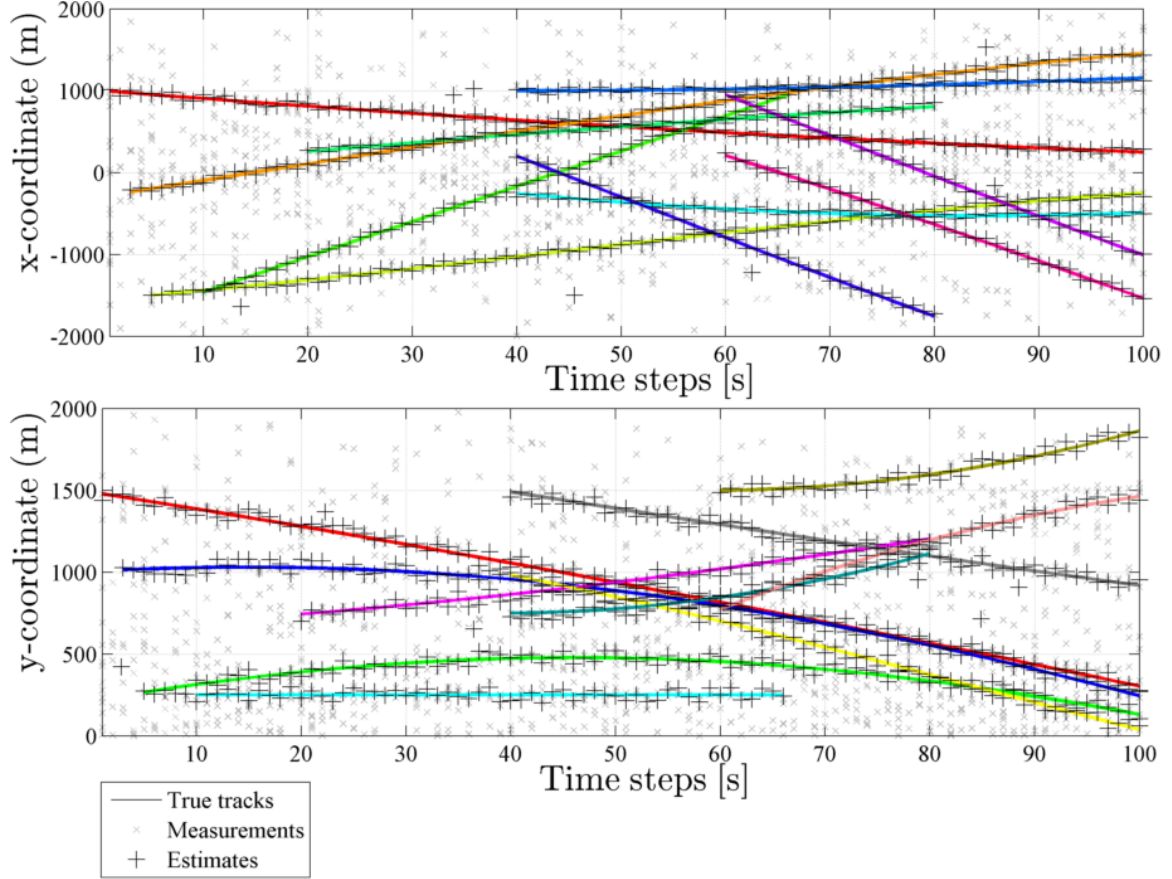


Figure 5.2: SMC-CBMeMber filter estimates, measurements and true target tracks in x and y coordinates versus time.

We then compared the tracking accuracy and the track initiation/termination performances of the (a) SMC-PHD (Algorithm 12), the (b) SMC-CPHD (Algorithm 13) and the (c) SMC-CBMeMber (Algorithm 14) filters over 100 Monte Carlo (MC) runs. For these comparisons, all the algorithms are given the same target trajectories with independently generated clutter and (target generated) measurements, so as to ensure a fair overall comparison.

At each time step, in the SMC implementations of the PHD and CPHD filters, the state transition model is used as the proposal to update the intensity D_k , and resampling is performed so that an average of 1000 particles is kept for each expected target.

Figure 5.3 shows the mean and standard deviation of the estimated cardinality distribution versus time.

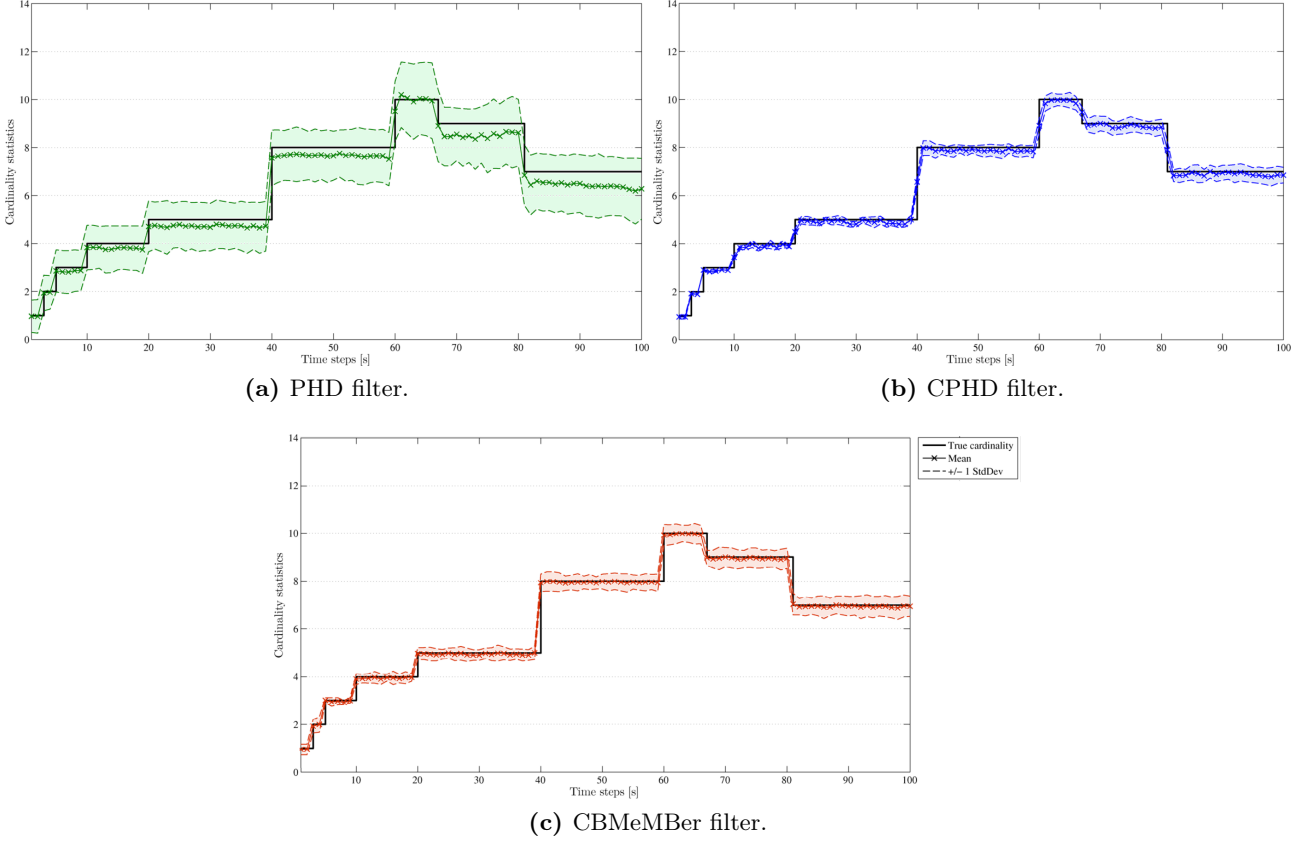


Figure 5.3: Cardinality statistics (mean and standard deviation) versus time.

This confirms that the CBMeMber, PHD and CPHD filters are unbiased in estimating the cardinality. Notice that the CBMeMber filter has a smaller variance on its estimated cardinality than the PHD filter, but has a larger variance than the CPHD filter. This is an expected behavior since the CBMeMber filter propagates a parameterized approximation to the posterior cardinality distribution, whereas the PHD filter propagates only the mean and the CPHD filter propagates the entire distribution.

Figure 5.4 compares the OSPA penalty ($p = 2$, $c = 100$ m) for the three filters. Note that the SMC-CBMeMber, SMC-PHD and SMC-CPHD filters produce average errors of approximately 46 m, 43 m and 21 m per target. These results suggest that the CBMeMber filter outperforms the CPHD filter which in turn outperforms the PHD filter. In part this can be ascribed to the error-prone clustering step, required in the case of the moment approximations (i.e. PHD and CPHD, to extract the state estimates, whereas the multi-Bernoulli representation of the posterior density allows direct and accurate state estimates extraction. The curves exhibit peaks at time instants involving cardinality changes, which are smaller in magnitude in the PHD and CBMeMber filters than in the CPHD filter. This means that the CPHD filter is slower than the PHD and CBMeMber filters to respond to cardinality

changes. Conversely, during time intervals when there are no cardinality changes, the PHD filter incurs a higher penalty than the CPHD filter.

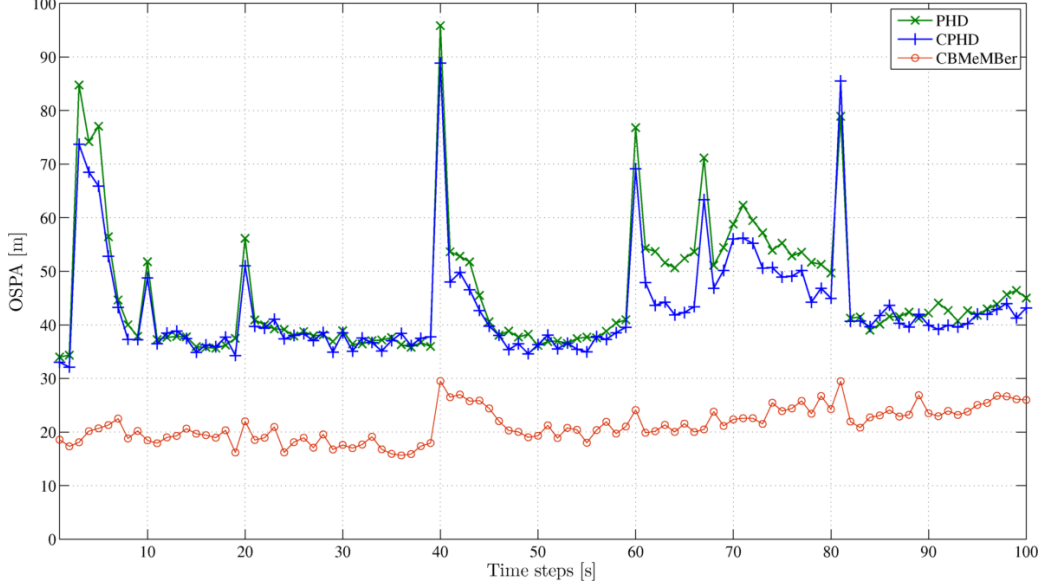


Figure 5.4: OSPA distance ($p = 2$, $c = 100$ m)

Figure 5.5 shows the OSPA localization and cardinality components, further confirming the results so far. In terms of localization error, the SMC-CBMeMBer filter outperforms both the SMC-CPHD and SMC-PHD filters. In terms of cardinality error, the SMC-CBMeMBer filter performs similarly to SMC-CPHD filter, and better than the SMC-PHD filter.

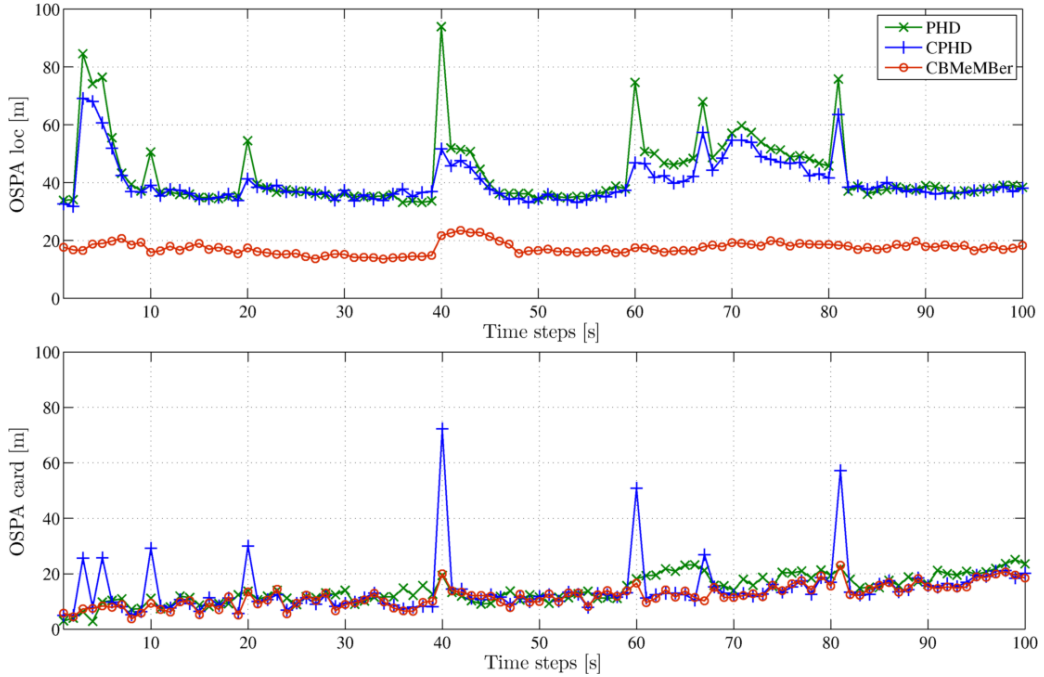


Figure 5.5: OSPA localization and cardinality components

Remark. Since the CBMeMber filter is derived under the assumption of reasonably low clutter, we also dig deeper to identify the breakdown of the proposed filter by increasing the clutter rate. Our simulations suggest that the proposed SMC-CBMeMber filter performs well up to a clutter intensity of approximately $\lambda_c = 3.4 \times 10^{-3} \text{ (rad.m)}^{-1}$, and down to a probability of detection of approximately $p_{D,k} = 0.9$.

5.3.2 Scenario 2: TBD tracking using SMC implementation

We demonstrate the SMC-CBMeMber filter and compare its performance with the generic SIR-PF (Algorithm 2), full multi-target Bayes filter implementation. This scenario considers a nonlinear TBD problem in which a time varying and unknown number of targets follow curved paths. A maximum of 4 targets are present at any time, and there are various target births and deaths throughout the scenario duration of $T_{max} = 20 \text{ s}$.

Targets are initialized at separate locations. The single target state $\mathbf{x}_{k,j}$ comprises the position and velocity of the j^{th} target, $s_{k,j} = [x_{k,j}, \dot{x}_{k,j}, y_{k,j}, \dot{y}_{k,j}]$ as well as the turn rate, $\omega_{k,j}$. The same dynamical model is used with the following parameters $\Delta_T = 1 \text{ s}$, $\mathbf{v}_{k,j} \sim \mathcal{N}(\cdot; 0, \sigma_v^2)$, $\sigma_v = 0.1 \text{ m.s}^{-2}$, $\mathbf{u}_{k,j} \sim \mathcal{N}(\cdot; 0, \sigma_u^2)$ and $\sigma_u = \pi/90 \text{ rad.s}^{-1}$. The death process is specified by a uniform single target probability of target survival $p_{S,k}(\mathbf{x}_{k,j}) = 0.99$.

The birth process is a Multi-Bernoulli density $\pi_\Gamma = \{(p_{B,k}(j), b_k^{(j)})\}_{j=1}^4$ where $p_{B,k}(j)$ is the probability of existence for term j and $b_k^{(j)}(x) = \mathcal{N}(x; m_\gamma^{(j)}, P_\gamma)$ with $m_\gamma^{(j)}$ is the estimated state of the j^{th} detection plot and $P_\gamma = \text{diag}([0.5, 0.5, 0.5, 0.5, 6\pi/180]^T)^2$.

Table 5.2: Bernoulli parameter set

Component j	$p_{B,k}(j)$	$m_\gamma^{(j)}$
1	0.01	$[7, 1, 25, -1, \pi/360]$
2	0.01	$[10, 1, 10, 1, \pi/360]$
3	0.01	$[1, 1, 1, 1, \pi/360]$
4	0.01	$[25, -1, 30, -1, \pi/360]$

In this demonstration, the observation region covers the intervals $[0 \ 40] \text{ m}$ in range, $[-3 \ 3] \text{ m.s}^{-1}$ in Doppler and the angles $[10 \ 80] \text{ degree}$ in bearing. The Radar parameters used in simulation are reported in Table 5.3.

Table 5.3: Radar parameters used in simulation

Beamwidth in bearing	$\Delta_b = 0.2 \text{ [degree]}$
Range-quant size	$\Delta_r = 1.5 \text{ [m]}$
Doppler-bin size	$\Delta_d = 0.5 \text{ [m.s}^{-1}\text{]}$

The true trajectories are shown Figure 5.6.

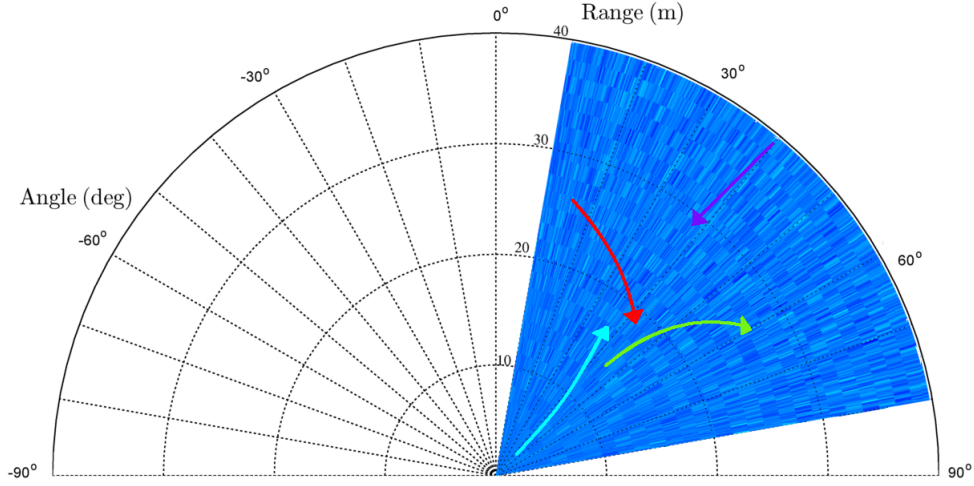


Figure 5.6: True tracks in polar coordinates for Scenario 2.

In Figure 5.7, The tracking results for a single scan of the SMC-CBMeMber and SIR-PF filters are shown. It suggests that both filters are able to initiate new tracks, correctly track the targets in view and terminate tracks. Note, however, that occasional delays in initiating and terminating tracks occurred with the SMC-CBMeMber filter.

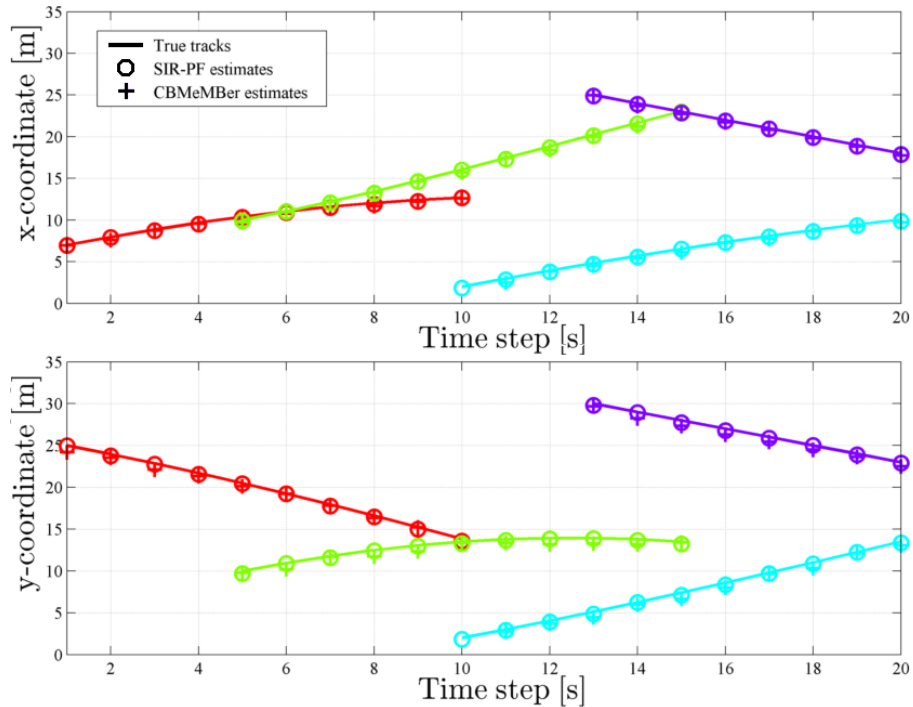


Figure 5.7: SMC-CBMeMber and SIR-PF filters estimates and true target tracks in x and y coordinates versus time.

At each time step, in the SMC implementation of the CBMeMBer filter, the state transition model is used as the proposal for each hypothesized track. A maximum of $L_{\max} = 5000$ and a minimum of $L_{\min} = 1000$ particles per hypothesized track is imposed, with resampling performed so that the number of particles representing each track is proportional to its probability of existence. Additionally, hypothesized tracks with existence probabilities less than $\tau_P = 10^{-2}$ are dropped and a maximum of $H_{\max} = 100$ tracks are kept.

Figure 5.8 compares the OSPA penalty ($p = 2$, $c = 5$ m) for two filters. Note that the SMC-CBMeMBer and the SIR-PF filters produce average errors of approximately 2 m and 0.7 m per target respectively. These results suggest that the SMC-CBMeMBer filter offers an attractive computationally cheaper alternative.

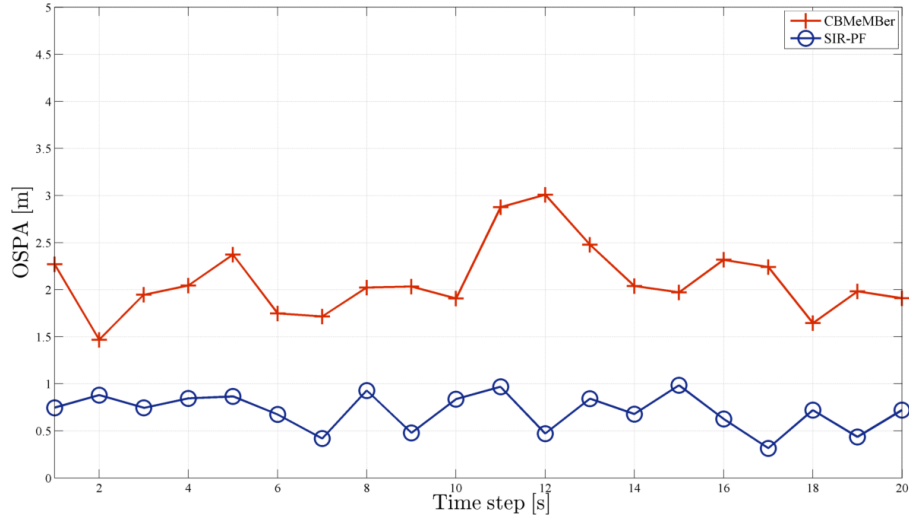


Figure 5.8: OSPA distance ($p = 2$, $c = 5$ m)

Figure 5.9 shows the OSPA localization and cardinality components, further confirming the results so far. The localization error does not grow with time, indicating that the position estimates are accurate and stable. The cardinality errors are also relatively stable, exhibiting jumps at instants involving target births or deaths.

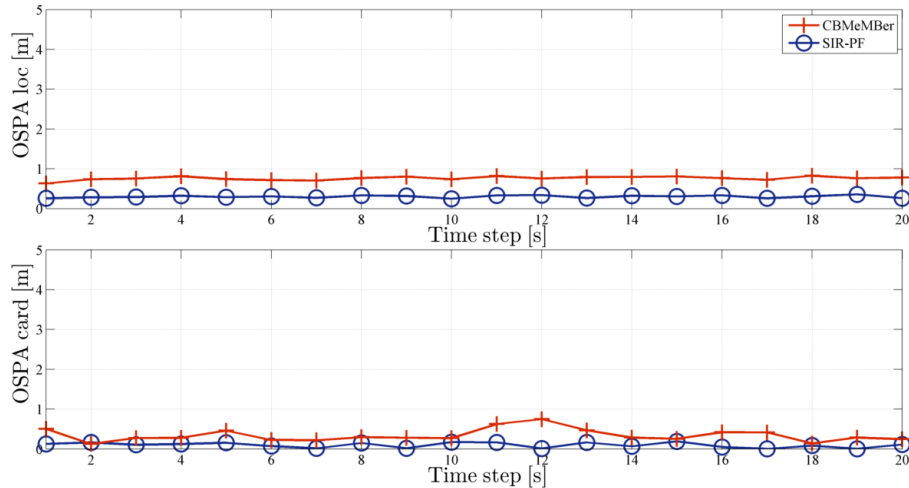


Figure 5.9: OSPA localization and cardinality components

Exploiting External Knowledge and Learning from the Tracking Process

Target tracking based on measurements collected by a radar or a similar sensor is an important example where statistical signal processing techniques can be applied to achieve optimal performance. Additional knowledge on the environment and/or the targets is oftentimes ignored or used only on an ad hoc basis. This chapter seeks to exploit external information to increase surveillance system performance. The main contents of this chapter have also been published in [26].

This chapter is organized as follows. Section 6.1 describes the single target constrained filtering problem and discusses different approaches to embed constraints in the particle filter framework. Section 6.2 introduces the proposed multi-scan method for processing of external knowledge, i.e., the KB-Smoother, along with a proof of improvements. Finally section 6.3 proposes a robust and accurate tracker which combines the KB-Smoother and the IP-MCMC-PF.

6.1 Constrained Bayesian Filtering

6.1.1 Introduction to Constrained Filtering

Many sources of external knowledge are available, e.g. 2D/3D digital maps of the environment and/or output of targets classifier (friendly/hostile, radar cross section, min/max velocities, etc.). This information can be formalized in terms of constraints on the system dynamics. Due to nonlinearities, the superposition principle cannot be applied, so that optimal performance cannot be achieved by first solving the nonlinear filtering problem and then projecting the solution onto the constrained state space. Therefore, solving a constrained version of the filtering problem is required.

Targets constrained by terrain or a road network is an example of state-constrained estimation problem. In [1], a PF algorithm for tracking and identifying ground targets in a road-constrained environment is developed. Inequality constraints on target speed and off-road distance are treated using a min-max saturation approach, which requires a low computational load while leading to suboptimal constraints satisfaction. Similarly, the knowledge on shipping lanes and sea/land distinction can strongly improve the littoral tracking and detection performance. This is achieved, in [69], by formulating the problem as Joint Tracking and Classification (JTC), where a target class is assigned for each isolated land or water region.

Targets motion constrained by maximum speeds and accelerations is another example of state-constrained estimation problem. In [42], a PF algorithm for tracking airplanes given their flight envelopes (e.g. max speed constraint) is proposed. A rejection-sampling approach is adopted such that samples fall inside the region of interest and approximately represent the truncated distribution of the conventional tracking problem at the bounds of the flight envelope.

In section 6.2, a novel approach for multiscan Bayes optimal knowledge exploitation is proposed and applied to single target tracking problems. The core idea is to use a Knowledge-Based (KB) Fixed-Lag Smoothing [111] procedure to obtain the predicted particles cloud at the next time step. Such multiscan algorithm, i.e. named KB-Smoother, is particularly interesting since a proof of improvements in terms of entropy reduction can be inferred 6.2.2, and sensibly enhanced track accuracy is achieved for a finite number of particles. Furthermore, the method can be easily implemented by means of a standard SIR-PF. However, in practice, the SIR-PF suffers from depletion problems, which are further amplified by the Smoothing technique, thus making the approach less suited to handle multi-modal PDFs (e.g. multi-target and/or ambiguous tracking problems). Sequential MCMC methods represent an efficient alternative to the standard SIR-PF approach, in particular in high dimensional spaces. Furthermore, computational concerns arising in real-time multi-target Bayesian filtering motivate the need of a fully parallelizable MCMC approach like in section 4.3 [24].

In section 6.3 we propose and analyze a combination of the KB-Smoother along with the IP-MCMC-PF. As mentioned, the combination of the two methods is suggested to achieve an improved track accuracy while mitigating the loss of particles diversity. In particular, simulation results for typical tracking scenarios confirm the benefits of the proposed approach for single and multi-target tracking problems. Notice that we focus on solving the tracking problem for a given cardinality n_k since we are interested in enlightening the benefits of external knowledge exploitation. Solutions to handle time-varying number of targets can be obtained by following the approaches described in section 4.4 and in section 4.5.

6.1.2 Exploiting External Knowledge through Constraints

Let us recall the single target Bayesian filtering problem,

$$\begin{aligned}\mathbf{x}_k &= f_k(\mathbf{x}_{k-1}, \mathbf{v}_{k-1}), \\ \mathbf{z}_k &= h_k(\mathbf{x}_k, \mathbf{w}_k),\end{aligned}\tag{6.1}$$

with,

- $\mathbf{x}_k \in \mathbb{R}^{n_x}$ the system state, $\mathbf{z}_k \in \mathbb{R}^{n_z}$ the measurement vector;
- $\mathbf{v}_k \sim p_{\mathbf{v}_k}(\mathbf{v})$ the process noise, $\mathbf{w}_k \sim p_{\mathbf{w}_k}(\mathbf{w})$ the measurement noise;
- $f_k : \mathbb{R}^{n_x} \times \mathbb{R}^{n_v} \rightarrow \mathbb{R}^{n_x}$ and $h_k : \mathbb{R}^{n_x} \times \mathbb{R}^{n_w} \rightarrow \mathbb{R}^{n_z}$ possibly nonlinear vector-valued functions.

In the state space description of (6.1), the state variables usually correspond to physical quantities of interest, and validity regions are helpful for the filter design. For instance, speed constraints can be defined based on the output of the classification problem and/or based on the type of ground/sea crossed.

We focus on processing of external knowledge that is available in terms of hard constraints on the system dynamics, which is defined as constraints perfectly known by the tracking system and which cannot be infringed by the targets. Practical examples are the tracking of ground vehicles moving on a road network, or the tracking of ships traveling on canals. In general, all the constraints arising from physical laws can be formulated in terms of hard constraints.

Let external information be available in terms of nonlinear inequality constraints, i.e.,

$$\mathbf{a}_k \leq \mathbf{C}_k(\mathbf{x}_k) \leq \mathbf{b}_k,\tag{6.2}$$

where $\mathbf{C}_k : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_c}$, and the inequality sign holds for all elements. For convenience, let \mathcal{C}_k be the set of all states satisfying the inequality constraint eq.(6.2), i.e.

$$\mathcal{C}_k \triangleq \{\mathbf{x}_k : \mathbf{x}_k \in \mathbb{R}^{n_x}, \mathbf{a}_k \leq \mathbf{C}_k(\mathbf{x}_k) \leq \mathbf{b}_k\},\tag{6.3}$$

and $\mathcal{C}_{1:k} \triangleq \{\mathcal{C}_0, \mathcal{C}_1, \dots, \mathcal{C}_k\}$ be the sequence of \mathcal{C}_k up to time k . From a Bayesian viewpoint, exploitation of external knowledge boils down to finding an approximation the a posteriori PDF:

$$f_k(\mathbf{x}_k | \mathbf{z}_{1:k}, \mathcal{C}_{1:k}) \propto \begin{cases} f_k(\mathbf{x}_k | \mathbf{z}_{1:k}), & \mathbf{x}_1 \in \mathcal{C}_1, \dots, \mathbf{x}_k \in \mathcal{C}_k \\ 0, & \text{otherwise,} \end{cases}\tag{6.4}$$

where conditioning is performed also with respect to the sequence $\mathcal{C}_{1:k}$ of constrained state variables. Assume that we are able to define a two-step recursion such that:

$$\begin{aligned}f_{k-1}(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1}, \mathcal{C}_{1:k-1}) &\xrightarrow{\text{prediction}} f_{k|k-1}(\mathbf{x}_k | \mathbf{z}_{1:k-1}, \mathcal{C}_{1:k-1}) \\ f_{k|k-1}(\mathbf{x}_k | \mathbf{z}_{1:k-1}, \mathcal{C}_{1:k-1}) &\xrightarrow{\text{update}} f_k(\mathbf{x}_k | \mathbf{z}_{1:k}, \mathcal{C}_{1:k})\end{aligned}$$

Then, for a sufficiently large number of particles, a PF approximation of such recursion will converge to the exact a posteriori PDF. Notice that in the above recursion we are implicitly assuming that processing of the constraints at time k is done in the update step. However, it is possible to define two Bayesian recursions for constrained filtering, in which knowledge about the constraints is used in the prediction step in one case, and in the update step in the other case [112].

(1) Using Knowledge in the Prediction Step

To exploit the constraint in the prediction step, we define the following predictive PDF:

$$\begin{aligned}
 f_{k|k-1}(\mathbf{x}_k | \mathbf{z}_{1:k-1}, \mathcal{C}_{1:k}) &= \int f(\mathbf{x}_k, \mathbf{x}_{k-1} | \mathbf{z}_{1:k-1}, \mathcal{C}_{1:k}) d\mathbf{x}_{k-1} \\
 &= \int \pi(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{z}_{1:k-1}, \mathcal{C}_{1:k}) f(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1}, \mathcal{C}_{1:k}) d\mathbf{x}_{k-1} \\
 &= \int \pi_{k|k-1}(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathcal{C}_k) f_{k-1|k-1}(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1}, \mathcal{C}_{1:k-1}) d\mathbf{x}_{k-1}. \tag{6.5}
 \end{aligned}$$

Proceeding, the update step of the Bayesian recursion is given by

$$\begin{aligned}
 f_k(\mathbf{x}_k | \mathbf{z}_{1:k}, \mathcal{C}_{1:k}) &= \frac{p(\mathbf{x}_k, \mathbf{z}_k, \mathbf{z}_{1:k-1}, \mathcal{C}_{1:k})}{p(\mathbf{z}_k, \mathbf{z}_{1:k-1}, \mathcal{C}_{1:k})} \\
 &= \frac{g(\mathbf{z}_k | \mathbf{x}_k, \mathbf{z}_{1:k-1}, \mathcal{C}_{1:k}) f(\mathbf{x}_k | \mathbf{z}_{1:k-1}, \mathcal{C}_{1:k}) p(\mathbf{z}_{1:k-1}, \mathcal{C}_{1:k})}{p(\mathbf{z}_k | \mathbf{z}_{1:k-1}, \mathcal{C}_{1:k}) p(\mathbf{z}_{1:k-1}, \mathcal{C}_{1:k})} \\
 &= \frac{g_k(\mathbf{z}_k | \mathbf{x}_k) f_{k|k-1}(\mathbf{x}_k | \mathbf{z}_{1:k-1}, \mathcal{C}_{1:k})}{p(\mathbf{z}_k | \mathbf{z}_{1:k-1}, \mathcal{C}_{1:k})}. \tag{6.6}
 \end{aligned}$$

This amounts to truncating the PDFs of the conventional tracking problem at the bounds of $\mathcal{C}_{1:k}$ and renormalizing it.

(2) Using Knowledge in the Update Step

Conversely, to exploit the constraint in the update step, the following PDF is targeted in prediction,

$$\begin{aligned}
 f_{k|k-1}(\mathbf{x}_k | \mathbf{z}_{1:k-1}, \mathcal{C}_{1:k-1}) &= \int f(\mathbf{x}_k, \mathbf{x}_{k-1} | \mathbf{z}_{1:k-1}, \mathcal{C}_{1:k-1}) d\mathbf{x}_{k-1} \\
 &= \int \pi(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{z}_{1:k-1}, \mathcal{C}_{1:k-1}) f(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1}, \mathcal{C}_{1:k-1}) d\mathbf{x}_{k-1} \\
 &= \int \pi_{k|k-1}(\mathbf{x}_k | \mathbf{x}_{k-1}) f_{k-1|k-1}(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1}, \mathcal{C}_{1:k-1}) d\mathbf{x}_{k-1}, \tag{6.7}
 \end{aligned}$$

and then updated by using both the new measurement \mathbf{z}_k and the constraints \mathcal{C}_k at time k , i.e.

$$\begin{aligned}
 f_k(\mathbf{x}_k | \mathbf{z}_{1:k}, \mathcal{C}_{1:k}) &= \frac{p(\mathbf{x}_k, \mathbf{z}_k, \mathbf{z}_{1:k-1}, \mathcal{C}_k, \mathcal{C}_{1:k-1})}{p(\mathbf{z}_k, \mathbf{z}_{1:k-1}, \mathcal{C}_k, \mathcal{C}_{1:k-1})} \\
 &= \frac{g_k(\mathbf{z}_k | \mathbf{x}_k) p(\mathcal{C}_k | \mathbf{x}_k) f_{k|k-1}(\mathbf{x}_k | \mathbf{z}_{1:k-1}, \mathcal{C}_{1:k-1})}{p(\mathbf{z}_k | \mathbf{z}_{1:k-1}, \mathcal{C}_{1:k}) p(\mathcal{C}_k | \mathcal{C}_{1:k-1})}. \tag{6.8}
 \end{aligned}$$

By using Bayes' theorem to compare the a posteriori PDFs defined by (6.6)-(6.8), it can be shown that the two Bayesian recursions for constrained filtering are equivalent.

6.1.3 Tracking with Hard Constraints - Review of Sequential Monte Carlo Methods

SMC methods can be used to obtain a PF approximation of the above introduced Bayesian recursions. The former recursion (1) is more intuitive since the constraints are characteristics of the target physics. While this approach generally leads to computationally demanding algorithms such as REJECTION-SAMPLING PF [42], the latter approach (2) can be efficiently implemented by the PSEUDO-MEASUREMENTS PF [135]. A brief discussion on these two techniques is presented below.

Rejection-Sampling (Prediction Step)

Consider an elucidation of the conditional probability theorem using the following example:

$$p(\mathbf{x}|\mathcal{C}) = \frac{p(\mathbf{x}, \mathcal{C})}{p(\mathcal{C})}, \quad (6.9)$$

where the conditioning event \mathcal{C} represents constraints on \mathbf{x} such that $\mathcal{C} = \{\mathbf{x} : \mathbf{x} \leq c\}$.

Then the following holds:

$$p(\mathbf{x}|\mathcal{C}) = p(\mathbf{x}|\mathbf{x} \leq c) = \frac{p(\mathbf{x}, \mathbf{x} \leq c)}{p(\mathbf{x} \leq c)}.$$

In other words,

$$p(\mathbf{x}|\mathcal{C}) = \begin{cases} \frac{p(\mathbf{x})}{p(\mathbf{x} \leq c)}, & \text{if } \mathbf{x} \leq c ; \\ 0, & \text{otherwise;} \end{cases} \quad (6.10)$$

which is the original $p(\mathbf{x})$ restricted to $\mathbf{x} \leq c$ and suitably normalized. Samples from the truncated version of $p(\mathbf{x})$ i.e. $p(\mathbf{x}|\mathcal{C})$ are generated by using the Algorithm 15 [42]. This will ensure that the samples fall inside the region of interest and approximately represent the renormalized truncated PDF. This simple modification can be incorporated into standard particle filtering techniques. The main drawback of this solution is the lack of knowledge on the time required to perform Rejection-Sampling.

Algorithm 15: Rejection-Sampling Particle Filter

input : $\{\mathbf{x}_{k-1}^{(i)}, w_{k-1}^{(i)}\}_{i=1}^N$ and the new measurement \mathbf{z}_k

output: $\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}_{i=1}^N$.

while $i \leftarrow 1$ **to** N (*Prediction step*) **do**

while $\mathbf{x}_k^{(i)} \notin \mathcal{C}_k$ (*Rejection-Sampling*) **do**

 Generate a new particle $\tilde{\mathbf{x}}_k^{(i)} \sim p_k(\mathbf{x}_k^{(i)} | \mathbf{x}_{k-1}^{(i)})$.

end

end

while $i \leftarrow 1$ **to** N (*Update step*) **do**

 Compute weights $\tilde{w}_k^{(i)} = w_{k-1}^{(i)} p(\mathbf{z}_k | \mathbf{x}_k^{(i)})$.

end

Normalization step $w_k^{(i)} = \frac{\tilde{w}_k^{(i)}}{\sum_{i=1}^N \tilde{w}_k^{(i)}}$, for $i = 1 : N$.

Effective sampling size $N_{eff} = \frac{1}{\sum_{i=1}^N (w_k^{(i)})^2}$.

if $N_{eff} \leq \beta N$ (*Resampling step*) **then**

 Draw a new set of particles $\{\mathbf{x}_k^{(i)}, 1/N\}_{i=1}^N$ s.t. $P(\tilde{\mathbf{x}}_k^{(i)} = \mathbf{x}_k^{(i)}) = w_k^{(i)}$.

else

 Copy $\{\tilde{\mathbf{x}}_k^{(i)}, w_k^{(i)}\}$ to $\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}$, for $i = 1 : N$.

end

Pseudo-Measurements (Update Step)

The Pseudo-Measurements [135] approach interprets the constraints on the state variables as additional measurement equations, called generalized measurements [94], i.e. rather than constraining the PDF, this approach deals with constraining the instantaneous measurement. The Pseudo-Measurements (generalized) likelihood $p(\mathcal{C}_k|\mathbf{x}_k)$ can be chosen as the indicator function of \mathcal{C}_k in the case of hard constraints, i.e.,

$$p(\mathcal{C}_k|\mathbf{x}_k) = \begin{cases} 1, & \text{if } \mathbf{a}_k \leq \mathbf{C}_k(\mathbf{x}_k) \leq \mathbf{b}_k; \\ 0, & \text{otherwise.} \end{cases} \quad (6.11)$$

Thus, the only change necessary in the PF algorithm is the use of an additional likelihood function in the evaluation of the unnormalized weights:

$$\tilde{w}_k^{(i)} = w_{k-1}^{(i)} \frac{p(\mathbf{z}_k|\mathbf{x}_k^{(i)}) p(\mathcal{C}_k|\mathbf{x}_k^{(i)}) p(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})}{q_k(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)}, \mathbf{z}_k)}. \quad (6.12)$$

Furthermore, if we use the transition kernel as proposal distribution, we have the common simplification:

$$\tilde{w}_k^{(i)} = w_{k-1}^{(i)} p(\mathbf{z}_k|\mathbf{x}_k^{(i)}) p(\mathcal{C}_k|\mathbf{x}_k^{(i)}), \quad (6.13)$$

which leads us to the definition of the PF described in Algorithm 16.

Algorithm 16: Pseudo-Measurement Particle Filter

input : $\{\mathbf{x}_{k-1}^{(i)}, w_{k-1}^{(i)}\}_{i=1}^N$ and the new measurement \mathbf{z}_k

output: $\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}_{i=1}^N$.

while $i \leftarrow 1$ **to** N (*Prediction step*) **do**

 | Generate a new particle $\tilde{\mathbf{x}}_k^{(i)} \sim p_k(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})$.

end

while $i \leftarrow 1$ **to** N (*Update step*) **do**

 | Compute weights $\tilde{w}_k^{(i)} = w_{k-1}^{(i)} p(\mathbf{z}_k|\mathbf{x}_k^{(i)}) p(\mathcal{C}_k|\mathbf{x}_k^{(i)})$.

end

Normalization step $w_k^{(i)} = \frac{\tilde{w}_k^{(i)}}{\sum_{i=1}^N \tilde{w}_k^{(i)}}$, for $i = 1 : N$.

Effective sampling size $N_{eff} = \frac{1}{\sum_{i=1}^N (w_k^{(i)})^2}$.

if $N_{eff} \leq \beta N$ (*Resampling step*) **then**

 | Draw a new set of particles $\{\mathbf{x}_k^{(i)}, 1/N\}_{i=1}^N$ s.t. $P(\tilde{\mathbf{x}}_k^{(i)} = \mathbf{x}_k^{(i)}) = w_k^{(i)}$.

else

 | Copy $\{\tilde{\mathbf{x}}_k^{(i)}, w_k^{(i)}\}$ to $\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}$, for $i = 1 : N$.

end

Remark. Note that the pseudo measurement approach, which balances between the current state estimate (that may infringe the constraints) and the pseudo measurement (from within the bound constraints), cannot rigidly enforce the constraint.

Remark. The approach requires a large number of particles since many of them are discarded at each time step. However, the computational load is drastically reduced with respect to Rejection-Sampling.

6.1.4 Bayesian Smoothing - Sequential Monte Carlo Approach

Let us briefly review methodology for Monte Carlo smoothing based upon particle filters. Note that smoothing is particularly relevant in complex dynamical systems, since simple filtering often yields fairly simplistic and uninformative state estimates, while the "LOOKAHEAD" allowed by smoothing enables much more accurate estimates to be achieved retrospectively.

Within the Bayesian framework, applying the probability chain rule yields the following expression for the joint smoothing distribution:

$$f_{k|k-1}(\mathbf{x}_{1:k}|\mathbf{z}_{1:k}) = f_0(\mathbf{x}_0) \left(\prod_{i=1}^k \pi_{i|i-1}(\mathbf{x}_i|\mathbf{x}_{i-1}) \right) \left(\prod_{i=1}^k g_i(\mathbf{z}_i|\mathbf{x}_i) \right), \quad (6.14)$$

where $\mathbf{x}_{1:k} \triangleq \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k\}$.

Let us recall the filtering recursion:

$$\begin{aligned} f_k(\mathbf{x}_k|\mathbf{z}_{1:k}) &= \frac{g_k(\mathbf{z}_k|\mathbf{x}_k, \mathbf{z}_{1:k-1}) f_{k|k-1}(\mathbf{x}_k|\mathbf{z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})} \\ &\propto g_k(\mathbf{z}_k|\mathbf{x}_k) f_{k|k-1}(\mathbf{x}_k|\mathbf{z}_{1:k-1}), \end{aligned}$$

where,

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

Note that the filtering version of the PF actually provides us with an empirical approximation of the joint smoothing distribution at no extra cost, since the equations are defined for the whole path of the process from time 1 up to time k [38]. In fact, the stored particle trajectories $\{\mathbf{x}_{1:k}^{(i)}\}_{i=1}^N$ and their associated weights $\{w_k^{(i)}\}_{i=1}^N$ can be viewed as weighted samples from the joint smoothing distribution $f_k(\mathbf{x}_{1:k}|\mathbf{z}_{1:k})$, i.e.

$$f_k(\mathbf{x}_{1:k}|\mathbf{z}_{1:k}) \approx \sum_{i=1}^N w_k^{(i)} \delta_{\mathbf{x}_{1:k}^{(i)}}(\mathbf{x}_{1:k}), \quad \sum_{i=1}^N w_k^{(i)} = 1, w_k^{(i)} \geq 0. \quad (6.15)$$

To update the smoothing density from time k to time $k+1$, factorizes it as:

$$f_{k+1}(\mathbf{x}_{1:k+1}|\mathbf{z}_{1:k+1}) = f_k(\mathbf{x}_{1:k}|\mathbf{z}_{1:k}) \times \frac{g_{k+1}(\mathbf{z}_{k+1}|\mathbf{x}_{k+1}) \pi_{k+1|k}(\mathbf{x}_{k+1}|\mathbf{x}_k)}{p(\mathbf{z}_{k+1}|\mathbf{z}_{1:k})}, \quad (6.16)$$

where the denominator is constant for a given data set. To proceed from time k to $k+1$, we select trajectories from the approximation (6.15) [52].

From these joint draws one may readily obtain fixed-lag or fixed-interval smoothed samples by simply extracting the required components from the sampled particles and retaining the same weights.

Given some smoothing lag L , from the joint draws $\left\{\mathbf{x}_{1:k}^{(i)}, w_k^{(i)}\right\}_{i=1}^N$, an approximation of the fixed-lag smoothing distribution $f_{k-L|k}(\mathbf{x}_{k-L}|\mathbf{z}_{1:k})$ can be extracted as follows:

$$f_{k-L|k}(\mathbf{x}_{k-L}|\mathbf{z}_{1:k}) \approx \sum_{i=1}^N w_k^{(i)} \delta_{\mathbf{x}_{k-L}^{(i)}}(\mathbf{x}_{k-L}). \quad (6.17)$$

This approach to smoothing is known as trajectory-based smoothing, or ancestry tree. However, due to the resampling step, for large value of L the approximation to the smoothing distribution turns out to be strongly depleted and inaccurate, which might lead to poor performance.

Similarly, one can study the dependencies (cross-correlation) over time of the state variables by extracting sub-sequences from the path particle representation. Given $L_0 > L$ and an interval $\Delta = L - L_0 > 0$ the fixed-interval smoothing distribution can be approximated as follows:

$$f_{\Delta|k}(\mathbf{x}_{k-(L_0+1):k-L}|\mathbf{z}_{1:k}) \approx \sum_{i=1}^N w_k^{(i)} \delta_{\mathbf{x}_{k-(L_0+1):k-L}^{(i)}}(\mathbf{x}_{k-(L_0+1):k-L}). \quad (6.18)$$

While these appealingly simple schemes can be successful for certain models and small lags L and L_0 , the selection (resampling) procedure will lead to a strongly depleted and potentially inaccurate representation of the required smoothing distributions for large values of lags. In fact, while the diversity of the particles is satisfactory for the current time index, successive resampling steps imply that, for time-lags that are back in the past, the diversity among the particles decreases eventually reaching a point where all current particles share a common ancestor. This is demonstrated in simulations in [67] and motivates the development of novel smoothing methods based on backward simulation.

There are various ways to improve upon the performance of the basic above scheme. First let us consider the backward smoothing recursions, which can be thought of as the natural extension of the Kalman backward smoothing recursions to nonlinear and non-Gaussian state-space models. Note that the joint smoothing distribution may be factorized as follows:

$$f_{T|T}(\mathbf{x}_{1:T}|\mathbf{z}_{1:T}) = f_{T|T}(\mathbf{x}_T|\mathbf{z}_{1:T}) \prod_{k=1}^{T-1} p(\mathbf{x}_k|\mathbf{x}_{k+1:T}, \mathbf{z}_{1:T}), \quad (6.19)$$

where, using the Markovian assumptions of the model, we can write

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{x}_{k+1:T}, \mathbf{z}_{1:T}) &= p(\mathbf{x}_k|\mathbf{x}_{k+1}, \mathbf{z}_{1:k}) \\ &= \frac{f_k(\mathbf{x}_k|\mathbf{z}_{1:k}) \pi_{k+1|k}(\mathbf{x}_{k+1}|\mathbf{x}_k)}{f_{k+1|k}(\mathbf{x}_{k+1}|\mathbf{z}_{1:k})} \\ &\propto f_k(\mathbf{x}_k|\mathbf{z}_{1:k}) \pi_{k+1|k}(\mathbf{x}_{k+1}|\mathbf{x}_k). \end{aligned} \quad (6.20)$$

These formulae then form the basis of a sequence-based smoother using the weighted sample generated in the forward pass of the SMC procedure [67].

The Forward Filtering Backward Smoothing (FFBS), also known as Baum-Welch algorithm [16], assumes that Bayesian filtering has already been performed on the entire data set, leading to an approximate representation of $f_k(\mathbf{x}_k|\mathbf{z}_{1:k})$ for each time step $k \in \{1, \dots, T\}$, consisting of weighted particles $\{\mathbf{x}_{1:k}^{(i)}, w_k^{(i)}\}_{i=1}^N$. Using this weighted sample representation, it is straightforward to construct a particle approximation to $p(\mathbf{x}_k|\mathbf{x}_{k+1}, \mathbf{z}_{1:T})$ from eq.(6.20) as follows:

$$p(\mathbf{x}_k|\mathbf{x}_{k+1}, \mathbf{z}_{1:T}) \approx \sum_{i=1}^N w_{k|k+1}^{(i)} \delta_{\mathbf{x}_k^{(i)}}(\mathbf{x}_k), \quad (6.21)$$

with the modified weights defined as:

$$w_{k|k+1}^{(i)} \triangleq \frac{w_k^{(i)} \pi_{k+1|k}(\mathbf{x}_{k+1}|\mathbf{x}_k^{(i)})}{\sum_{j=1}^N w_k^{(j)} \pi_{k+1|k}(\mathbf{x}_{k+1}|\mathbf{x}_k^{(j)})}. \quad (6.22)$$

This particle approximation can now be used to generate states successively in the reverse-time direction, conditioning on future states, using the sampling importance resampling idea. Specifically, given a random sample $\tilde{\mathbf{x}}_{k+1:T}$ drawn approximately from $f(\mathbf{x}_{k+1:T}|\mathbf{z}_{1:T})$, take one step back in time and sample $\tilde{\mathbf{x}}_k$ from the particle approximation (6.21) to $p(\mathbf{x}_k|\tilde{\mathbf{x}}_{k+1}, \mathbf{z}_{1:T})$. The pair $(\tilde{\mathbf{x}}_k, \tilde{\mathbf{x}}_{k+1:T})$ is then approximately a random realization from $f(\mathbf{x}_{k:T}|\mathbf{z}_{1:T})$. Repeating this process sequentially back over time produces the general particle smoother outlined in Algorithm 17.

Algorithm 17: Smoother realization

```

for  $k \leftarrow 1$  to  $T$                                      /* Forward Filtering */
do
    Run Particle filter, storing at each time step the particles and weights  $\{\mathbf{x}_{1:k}^{(i)}, w_k^{(i)}\}_{i=1}^N$ 
end
Choose  $\tilde{\mathbf{x}}_T = \mathbf{x}_T^{(i)}$  with probability  $w_T^{(i)}$ 
for  $k \leftarrow T - 1$  to  $1$                                  /* Backward Smoothing */
do
    foreach  $i \leftarrow 1$  to  $N$  do
        Calculate  $w_{k|k+1}^{(i)} \propto w_k^{(i)} f(\tilde{\mathbf{x}}_{k+1}|\mathbf{x}_k^{(i)})$ .
    end
    Normalize the modified weights.
    Choose  $\tilde{\mathbf{x}}_k = \mathbf{x}_k^{(i)}$  with probability  $w_{k|k+1}^{(i)}$ .
end
 $\tilde{\mathbf{x}}_{1:T} = (\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \dots, \tilde{\mathbf{x}}_T)$  is an approximate realization from  $p(\mathbf{x}_{1:T}|\mathbf{z}_{1:T})$ .

```

Further independent realizations are obtained by repeating this procedure as many times as needed. The computational complexity for each random realization is $O(NT)$, in contrast with the $O(N^2T)$ required for marginal smoothing procedure described below.

However, one is oftentimes interested in the marginal smoothing distributions denoted by $f_{k|T}(\mathbf{x}_k|\mathbf{z}_{1:T})$ for some time $k < T$. Then the following factorization can be used:

$$\begin{aligned} f_{k|T}(\mathbf{x}_k|\mathbf{z}_{1:T}) &= f_k(\mathbf{x}_k|\mathbf{z}_{1:k}) \int \left[\frac{f_{k+1|T}(\mathbf{x}_{k+1}|\mathbf{z}_{1:T}) \pi_{k+1|k}(\mathbf{x}_{k+1}|\mathbf{x}_k)}{\int f(\mathbf{x}|\mathbf{z}_{1:k}) \pi(\mathbf{x}_{k+1}|\mathbf{x}) d\mathbf{x}} \right] d\mathbf{x}_{k+1} \\ &= \int f_{k+1|T}(\mathbf{x}_{k+1}|\mathbf{z}_{1:T}) p(\mathbf{x}_k|\mathbf{x}_{k+1}, \mathbf{z}_{1:T}) d\mathbf{x}_{k+1}, \end{aligned} \quad (6.23)$$

where $p(\mathbf{x}_k|\mathbf{x}_{k+1}, \mathbf{z}_{1:T})$ simplifies as before in (6.20).

In a direct Monte Carlo implementation of (6.23) [90], one recursively obtains particle estimates of the marginal smoothing distribution at the next time instant, i.e. $f_{k+1|T}(\mathbf{x}_{k+1}|\mathbf{z}_{1:T})$ and combines these with the particle filtering estimate of $f_k(\mathbf{x}_k|\mathbf{z}_{1:k})$ in eq.(6.20). One complication, w.r.t the sequence-based smoothers of [67], is that a Monte Carlo estimate is required to approximate the denominator term in eq.(6.20), that is, $f_{k+1|k}(\mathbf{x}_{k+1}|\mathbf{z}_{1:k}) = \int f_k(\mathbf{x}_k|\mathbf{z}_{1:k}) \pi_{k+1|k}(\mathbf{x}_{k+1}|\mathbf{x}_k) d\mathbf{x}_k$, the normalizing constant. Proceeding, if we now approximate the smoothing distribution $f_{k+1|T}(\mathbf{x}_{k+1}|\mathbf{z}_{1:T})$ using the weighted samples $\left\{ \mathbf{x}_{k+1}^{(i)}, w_{k+1|1:T}^{(i)} \right\}_{i=1}^N$ i.e.,

$$f_{k+1|T}(\mathbf{x}_{k+1}|\mathbf{z}_{1:T}) \approx \sum_{i=1}^N w_{k+1|1:T}^{(i)} \delta_{\mathbf{x}_{k+1}^{(i)}}(\mathbf{x}_{k+1}), \quad (6.24)$$

an approximation of the marginal smoothing distribution is obtained as follows:

$$f_{k|T}(\mathbf{x}_k|\mathbf{z}_{1:T}) \approx \sum_{i=1}^N w_{k|1:T}^{(i)} \delta_{\mathbf{x}_k^{(i)}}(\mathbf{x}_k), \quad (6.25)$$

where the new weights are recursively updated according to:

$$w_{k|1:T}^{(i)} = w_k^{(i)} \left(\frac{\sum_{j=1}^N \frac{w_{k+1|1:T}^{(j)} (\mathbf{x}_{k+1}^{(j)}|\mathbf{x}_k^{(j)})}{\sum_{l=1}^N \pi_{k+1|k}(\mathbf{x}_{k+1}^{(j)}|\mathbf{x}_k^{(l)}) w_k^{(l)}} \right). \quad (6.26)$$

Remark. With respect to ancestry tree smoothing, the FFBS approach alleviates depletion problems and guarantees satisfactory performance. However, a straightforward implementation of such approach would lead to $O(N^2)$ complexity on particles number, and hence, would inherently be very expensive to compute as the number of particles becomes large. In particular, the bottleneck in evaluating the backward weights is due to the term:

$$f(\mathbf{x}_{k+1}^{(j)}) = \sum_{i=1}^N q(\mathbf{x}_k^{(i)}) \pi_{k+1|k}(\mathbf{x}_{k+1}^{(j)}|\mathbf{x}_k^{(i)}), \quad (6.27)$$

also known as the sum-kernel problem [71]. If we assume the function $\pi_{k+1|k}(\mathbf{x}_{k+1}|\mathbf{x}_k^{(i)})$ be a similarity kernel defined on a metric space, then the sum-kernel problem can be approximated in $O(N \log N)$ using algorithms from N-body simulation, which provide an approximate solution with controlled error [87].

Other forms of marginal smoothing can be obtained using the so-called two-filter formula [85], based on the following factorization:

$$\begin{aligned} f_{k|T}(\mathbf{x}_k|\mathbf{z}_{1:T}) &= \frac{f_k(\mathbf{x}_k|\mathbf{z}_{1:k}) p(\mathbf{z}_{k+1:T}|\mathbf{z}_{1:k}, \mathbf{x}_k)}{p(\mathbf{z}_{k+1:T}|\mathbf{z}_{1:k})} \\ &\approx f_k(\mathbf{x}_k|\mathbf{z}_{1:k}) p(\mathbf{z}_{k+1:T}|\mathbf{x}_k). \end{aligned} \quad (6.28)$$

Note that this requires the approximation from the usual forward filter plus a backward "anti-causal prediction" function $p(\mathbf{z}_{k+1:T}|\mathbf{x}_k)$. Furthermore it is not always straightforward to initialize or implement the required backward filtering pass.

6.2 Knowledge-Based Fixed-Lag Smoother

Exploitation of external knowledge through constrained filtering guarantees improved performance. In this section we introduce a new approach for multiscan processing of possibly available external knowledge based on a Fixed-Lag Smoothing procedure, and sketch a proof of improvements with respect to single-scan constrained filtering.

6.2.1 Knowledge-Based Fixed-Lag Smoother

In [111] we proposed a novel approach for multiscan Bayes optimal knowledge exploitation, which resorts to a Knowledge-Based Fixed-Lag Smoothing procedure at the previous step in order to determine the current predicted particles cloud. The KB-Smoother [111] is described in Algorithm 18. At time step k , before processing the new measurement \mathbf{z}_k , a prediction from time $k-1$ to time k has to be performed for each particle. While doing so one can exploit the constraints at time k and thus reduce the intrinsic uncertainty in the system modeling. One can extend this concept and determine the predicted cloud at time k while exploiting the constraints at times $k, k+1, \dots, k+L$ and thus further reduce the intrinsic uncertainty. Consider for instance the tracking of a ship that is moving in a canal. If the particles trajectory is constrained by the canal topology at each time instant, then the track accuracy can only be improved.

Remark. Notice that since the mathematical structure of the sets $\mathcal{C}_k, \mathcal{C}_{k+1}, \dots, \mathcal{C}_{k+L-1}$ defining the constraints at successive time instants is known at time k , the algorithm can be implemented on-line without any delay.

Algorithm 18: Knowledge-Based Fixed-Lag Smoother (Forward Filtering Backward Smoothing)

input : $\left\{ \mathbf{x}_{k-1}^{(i)}, w_{k-1}^{(i)} \right\}_{i=1}^N$ and a new measurement, \mathbf{z}_k

output: $\left\{ \mathbf{x}_k^{(i)}, w_k^{(i)} \right\}_{i=1}^N$

1 - *Forward Recursion*

while $t \leftarrow k$ **to** $k + L - 1$ **do**

while $i \leftarrow 1$ **to** N **do**

 Predict Particles: $\tilde{\mathbf{x}}_t^{(i)} \sim \pi_{t|t-1}(\mathbf{x}_t | \mathbf{x}_{t-1}^{(i)})$.

 Evaluate Weights: $\tilde{w}_t^{(i)} = w_{t-1}^{(i)} p(\mathcal{C}_t | \tilde{\mathbf{x}}_t^{(i)})$.

end

 Normalize Weights: $w_t^{(i)} = \tilde{w}_t^{(i)} / \sum_{i=1}^N \tilde{w}_t^{(i)}, \forall i = 1, \dots, N$.

 Resample if necessary:

if $\left(\sum_{i=1}^N (w_t^{(i)})^2 \right)^{-1} \leq \gamma N$; /* Resampling */

then

 Store resampling indices: $j_t^i \sim \sum_{i=1}^N w_t^{(i)} \delta_{\tilde{\mathbf{x}}_t^{(i)}}(\mathbf{x}) \quad \forall i = 1, \dots, N$.

 Extract New Particles: $\left\{ \mathbf{x}_t^{(i)}, w_t^{(i)} \right\} = \left\{ \tilde{\mathbf{x}}_t^{(j_t^i)}, 1/N \right\} \quad \forall i = 1, \dots, N$.

end

end

end

2 - *Backward Recursion*

while $t \leftarrow k + L - 2$ **to** k **do**

 Evaluate Backward Weights: $w_{t|k:T}^{(i)} = w_t^{(i)} \left(\frac{\sum_{j=1}^N \frac{w_{t+1|k:T}^{(j)} \pi_{t+1|t}(\mathbf{x}_{t+1}^{(j)} | \mathbf{x}_t^{(i)})}{\sum_{l=1}^N w_t^{(l)} \pi_{t+1|t}(\mathbf{x}_{t+1}^{(j)} | \mathbf{x}_t^{(l)})} \right)$.

end

$p(\mathbf{x}_k | \mathcal{C}_{k:T}) \approx \sum_{i=1}^N w_{k|k:T}^{(i)} \delta_{\mathbf{x}_k^{(i)}}(\mathbf{x}_k)$.

3 - *Standard Update Step*

while $i \leftarrow 1$ **to** N **do**

 Evaluate Weights: $\tilde{w}_k^{(i)} = w_{k|k+L}^{(i)} g_k(\mathbf{z}_k | \mathbf{x}_k^{(i)})$.

end

Normalization: $w_k^{(i)} = \tilde{w}_k^{(i)} / \sum_{i=1}^N \tilde{w}_k^{(i)} \quad \forall i = 1, \dots, N$.

Resample if necessary:

if $\left(\sum_{i=1}^N (w_k^{(i)})^2 \right)^{-1} \leq \beta N$; /* Resampling */

then

$\left\{ \tilde{\mathbf{x}}_k^{(i)}, 1/N \right\} \sim \left\{ \mathbf{x}_k^{(i)}, w_k^{(i)} \right\} \quad \forall i = 1, \dots, N$.

end

6.2.2 Multiscan Knowledge Exploitation Gain based on Entropy Reduction

Let \mathbf{x} and \mathcal{C} be two random variables, then the conditional entropy $h(\mathbf{x}|\mathcal{C})$ is given by the following, i.e.

$$h(\mathbf{x}|\mathcal{C}) = \int p(\bar{\mathbf{x}}, \bar{\mathcal{C}}) \log p(\bar{\mathbf{x}}|\bar{\mathcal{C}}) d\bar{\mathbf{x}} d\bar{\mathcal{C}} \quad (6.29)$$

$$= \int p(\bar{\mathcal{C}}) h(\mathbf{x}|\bar{\mathcal{C}}) d\bar{\mathcal{C}}, \quad (6.30)$$

where $\bar{\mathbf{x}}$ and $\bar{\mathcal{C}}$ are realizations of \mathbf{x} and \mathcal{C} , respectively, and $h(\mathbf{x}|\bar{\mathcal{C}})$ is given by:

$$h(\mathbf{x}|\bar{\mathcal{C}}) = \int p(\bar{\mathbf{x}}|\bar{\mathcal{C}}) \log p(\bar{\mathbf{x}}|\bar{\mathcal{C}}) d\bar{\mathbf{x}}. \quad (6.31)$$

Notice that $h(\mathbf{x}|\bar{\mathcal{C}})$ is not the conditional entropy but the entropy of the PDF $p(\mathbf{x}|\bar{\mathcal{C}})$, which is the PDF obtained when conditioning on the single realization $\bar{\mathcal{C}}$. This distinction is important since we are interested in studying the behavior of our method for each realization of the tracking problem. Whilst it is known that conditioning on a random variable \mathcal{C} reduces the entropy of the posterior PDF, this result does not hold in general when conditioning on a single realization $\bar{\mathcal{C}}$ [142, p. 43].

Theorem 6.2.1. *Let $\mathcal{C}_k, \mathcal{C}_{k+1}, \dots, \mathcal{C}_T$ be the sequence of Pseudo-Measurements describing the system constraints from time k to time $T = k + L$, $L > 0$ being the fixed-lag. Given a single realization $\bar{\mathcal{C}}_k, \bar{\mathcal{C}}_{k+1}, \dots, \bar{\mathcal{C}}_T$, the entropy of the fixed-lag smoothing PDF, $p(\mathbf{x}_k|\bar{\mathcal{C}}_{k:T})$, is less or equal than the entropy of the filtering PDF, $p(\mathbf{x}_k|\bar{\mathcal{C}}_k)$ [111], i.e.,*

$$h(\mathbf{x}_k|\bar{\mathcal{C}}_{k:T}) \leq h(\mathbf{x}_k|\bar{\mathcal{C}}_k), \quad \forall \bar{\mathcal{C}}_{k:T} \in S_{\mathbf{c}}. \quad (6.32)$$

Proof. The entropy of the FILTERING PDF is defined as:

$$h(\mathbf{x}_k|\bar{\mathcal{C}}_k) = - \int_{S_{\mathbf{x}_k}} p(\mathbf{x}_k|\bar{\mathcal{C}}_k) \log p(\mathbf{x}_k|\bar{\mathcal{C}}_k) d\mathbf{x}_k, \quad (6.33)$$

while the entropy of the SMOOTHING PDF is defined as:

$$h(\mathbf{x}_k|\bar{\mathcal{C}}_{k:T}) = - \int_{S_{\mathbf{x}_k}} p(\mathbf{x}_k|\bar{\mathcal{C}}_{k:T}) \log p(\mathbf{x}_k|\bar{\mathcal{C}}_{k:T}) d\mathbf{x}_k. \quad (6.34)$$

Let us define $S_{\mathbf{x}}^{\geq 1} = \{\mathbf{x} \in S_{\mathbf{x}} : p(\mathbf{x}) \geq 1\}$ and $S_{\mathbf{x}}^{< 1} = \{\mathbf{x} \in S_{\mathbf{x}} : p(\mathbf{x}) < 1\}$, where $p(\mathbf{x})$ is a generic PDF. Given $\alpha(\cdot)$ such that $\alpha(\mathbf{x}) \geq 1$, $\forall \mathbf{x} \in S_{\mathbf{x}}$, the following properties hold, i.e.

$$- \int_{S_{\mathbf{x}}^{\geq 1}} \alpha(\mathbf{x}) p(\mathbf{x}) \log \alpha(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} \leq - \int_{S_{\mathbf{x}}^{\geq 1}} p(\mathbf{x}) \log p(\mathbf{x}) d\mathbf{x} \leq 0 \quad (6.35)$$

$$- \int_{S_{\mathbf{x}}^{< 1}} \alpha(\mathbf{x}) p(\mathbf{x}) \log \alpha(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} \leq - \int_{S_{\mathbf{x}}^{< 1}} p(\mathbf{x}) \log p(\mathbf{x}) d\mathbf{x} \quad (6.36)$$

as well as the two filter formula, i.e.

$$p(\mathbf{x}_k|\bar{\mathcal{C}}_{k:T}) = \frac{p(\bar{\mathcal{C}}_{k+1:T}|\bar{\mathcal{C}}_k, \mathbf{x}_k)}{p(\bar{\mathcal{C}}_{k+1:T}|\bar{\mathcal{C}}_k)} p(\mathbf{x}_k|\bar{\mathcal{C}}_k) = \alpha(\mathbf{x}_k) p(\mathbf{x}_k|\bar{\mathcal{C}}_k). \quad (6.37)$$

Let us now show that $\alpha(\mathbf{x}_k) \geq 1$ holds independently from \mathbf{x}_k and $\bar{\mathcal{C}}_k, \dots, \bar{\mathcal{C}}_T$. Consider first the numerator, i.e.

$$p(\bar{\mathcal{C}}_{k+1:T}|\bar{\mathcal{C}}_k, \mathbf{x}_k) = \int_{\times_{l=k+1}^T \mathcal{C}_l} \prod_{j=0}^{L-1} p(\mathbf{x}_{k+j+1}|\mathbf{x}_{k+j}) d\mathbf{x}_{k+1:T}, \quad (6.38)$$

$$\begin{aligned} p(\bar{\mathcal{C}}_{k+1:T}|\bar{\mathcal{C}}_k, \mathbf{x}_k) &= \int_{\times_{l=k+1}^T S_{\mathbf{x}_l}} p(\bar{\mathcal{C}}_{k+1:T}, \mathbf{x}_{k+1:T}|\bar{\mathcal{C}}_k, \mathbf{x}_k) d\mathbf{x}_{k+1:T} \\ &= \int_{\times_{l=k+1}^T S_{\mathbf{x}_l}} p(\bar{\mathcal{C}}_{k+1:T}, \mathbf{x}_{k+1:T}) p(\mathbf{x}_{k+1:T}|\mathbf{x}_k) d\mathbf{x}_{k+1:T} \\ &= \int_{\times_{l=k+1}^T S_{\mathbf{x}_l}} \prod_{j=1}^L p(\bar{\mathcal{C}}_{k+j}|\mathbf{x}_{k+j}) \prod_{j=0}^{L-1} p(\mathbf{x}_{k+j+1}|\mathbf{x}_{k+j}) d\mathbf{x}_{k+1:T} \\ &= \int_{\times_{l=k+1}^T \mathcal{C}_l} \prod_{j=0}^{L-1} p(\mathbf{x}_{k+j+1}|\mathbf{x}_{k+j}) d\mathbf{x}_{k+1:T}, \end{aligned} \quad (6.39)$$

where $\times_{l=k+1}^T \mathcal{C}_l = \mathcal{C}_{k+1} \times \dots \times \mathcal{C}_T$ stands for Cartesian product.

Consider now the denominator, i.e.

$$\begin{aligned} p(\bar{\mathcal{C}}_{k+1:T}|\bar{\mathcal{C}}_k) &= \int_{\times_{l=k+1}^T S_{\mathbf{x}_l}} p(\bar{\mathcal{C}}_{k+1:T}, \mathbf{x}_{k+1:T}|\bar{\mathcal{C}}_k) d\mathbf{x}_{k+1:T} \\ &= \int_{\times_{l=k+1}^T S_{\mathbf{x}_l}} p(\bar{\mathcal{C}}_{k+1:T}, \mathbf{x}_{k+1:T}) p(\mathbf{x}_{k+1:T}|\bar{\mathcal{C}}_k) d\mathbf{x}_{k+1:T} \\ &= \int_{\times_{l=k+1}^T \mathcal{C}_l} \int_{S_{\mathbf{x}_k}} p(\mathbf{x}_{k+1:T}|\mathbf{x}_k) p(\mathbf{x}_k|\bar{\mathcal{C}}_k) d\mathbf{x}_k d\mathbf{x}_{k+1:T}. \end{aligned} \quad (6.40)$$

Consider the integral over $S_{\mathbf{x}_k}$ in the above formula, i.e.

$$\begin{aligned} \int_{S_{\mathbf{x}_k}} p(\mathbf{x}_{k+1:T}|\mathbf{x}_k) p(\mathbf{x}_k|\bar{\mathcal{C}}_k) d\mathbf{x}_k &= \int_{S_{\mathbf{x}_k}} \prod_{j=0}^{L-1} p(\mathbf{x}_{k+j+1}|\mathbf{x}_{k+j}) p(\mathbf{x}_k|\bar{\mathcal{C}}_k) d\mathbf{x}_k \\ &\leq \prod_{j=0}^{L-1} p(\mathbf{x}_{k+j+1}|\mathbf{x}_{k+j}). \end{aligned} \quad (6.41)$$

Thus, by using the inequality (6.41) to compare the eqs. (6.39) and (6.40), the following holds true, i.e.

$$\alpha(\mathbf{x}_k) \geq 1, \forall \mathbf{x}_k \in S_{\mathbf{x}_k}, \mathcal{C}_k \in S_{\mathbf{z}_k}, \dots, \mathcal{C}_T \in S_{\mathbf{z}_T}. \quad (6.42)$$

The entropy of the smoothed density is given by:

$$h(\mathbf{x}_k | \bar{\mathcal{C}}_{k:T}) = - \int_{S_{\mathbf{x}_k}} \alpha(\mathbf{x}_k) p(\mathbf{x}_k | \bar{\mathcal{C}}_k) \log \alpha(\mathbf{x}_k) p(\mathbf{x}_k | \bar{\mathcal{C}}_k) d\mathbf{x}_k. \quad (6.43)$$

The entropies in eqs.(6.33) and (6.43) can be rewritten as follows:

$$h(\mathbf{x}_k | \bar{\mathcal{C}}_k) = h(\mathbf{x}_k | \bar{\mathcal{C}}_k, S_{\mathbf{x}}^{\geq 1}) + h(\mathbf{x}_k | \bar{\mathcal{C}}_k, S_{\mathbf{x}}^{< 1}) \quad (6.44)$$

$$h(\mathbf{x}_k | \bar{\mathcal{C}}_{k:T}) = h(\mathbf{x}_k | \bar{\mathcal{C}}_{k:T}, S_{\mathbf{x}}^{\geq 1}) + h(\mathbf{x}_k | \bar{\mathcal{C}}_{k:T}, S_{\mathbf{x}}^{< 1}). \quad (6.45)$$

By using eqs.(6.35) and (6.36), we prove our claim, i.e.

$$h(\mathbf{x}_k | \bar{\mathcal{C}}_{k:T}) \leq h(\mathbf{x}_k | \bar{\mathcal{C}}_k), \forall \mathcal{C}_k \in S_{\mathbf{z}_k}, \dots, \mathcal{C}_T \in S_{\mathbf{z}_T}. \quad (6.46)$$

□

In summary, in the sense of the entropy of the posterior PDF, a fixed-lag smoothing approach for processing of external knowledge will always perform equally well as or better than classical single-step constrained filtering.

6.3 Multiscan Knowledge Exploitation using IP-MCMC-PF

In practice, the SIR-PF suffers from depletion problems, which are further amplified by the smoothing technique. Conceptually this problem could be effectively addressed by sequential MCMC methods.

6.3.1 IP-MCMC-PF for Target Tracking while Exploiting External Knowledge

Let us consider a multi-target setting where $\mathbf{X}_k = [\mathbf{x}_{k,1}, \dots, \mathbf{x}_{k,n_k}]$ is the multi-target state vector, with $\mathbf{x}_{k,j}$ being the state vector of the j^{th} target at time k . In the proposed IP-MCMC-PF algorithm, the posterior distribution over target states at time $k-1$ is represented by a set of N particles $\{\mathbf{X}_{k-1}^{(i)}\}_{i=1}^N$. Each particle contains the joint state of n_k partitions corresponding to the different targets. We refer to $\mathbf{x}_{k-1,j}^{(i)}$ as the states of the j^{th} partition of the particle i at time step $k-1$. The particle state vector is then given by:

$$\mathbf{X}_{k-1}^{(i)} = [\mathbf{x}_{k-1,1}^{(i)}, \dots, \mathbf{x}_{k-1,n_k}^{(i)}]. \quad (6.47)$$

At each time step k , the predicted particles cloud $\{\mathbf{X}_k^{(i)}\}_{i=1}^N$ is sampled while exploiting the system constraints at times $k, \dots, k+L$, by means of the KB-Smoother (Algorithm 18). Then, N_{MCMC} particles $\{\mathbf{X}_s^{(0)}\}_{s=1}^{N_{\text{MCMC}}}$ are randomly selected from $\{\mathbf{X}_k^{(i)}\}_{i=1}^N$ to be the starting points in the MCMC sampling procedure. The MCMC sampling procedure consists on running N_{MCMC} M-H samplers in parallel. Each sampler s is initiated with the configuration $\{\mathbf{X}_s^{(0)}; \ell_s^{(0)}\}$ and is used to generate a set of N samples from $f_k(\mathbf{X}_k | \mathbf{z}_{1:k})$. At each M-H iteration l , a single partition j of a particle i , which is randomly picked

out of the predictive prior $\{\mathbf{X}_k^{(i)}\}_{i=1}^N$, is randomly selected. Given this partition $\mathbf{x}_{k,j}^{(i)}$, a partition move $\mathbf{x}_{s,j}^*$ is proposed. This move seeks to update the single partition $\mathbf{x}_{k,j}^{(i)}$ via a Markov kernel describing a random walk. The proposed multi-target tracker is summarized in Algorithm 19, and the pseudo-code for each random-walk M-H algorithm is given by Algorithm 20.

Remark. The standard random-walk Metropolis algorithm [101] proposes local, symmetric moves around the current position. In many cases (especially in high dimensions) the variance of the proposal must be small for the corresponding acceptance probability to be satisfactory. However smaller proposal variance leads to higher autocorrelations, and large computing time to explore the state space. In contrast, hybrid Monte Carlo [59,108], also known as Hamiltonian Monte Carlo (HMC), exploits the information on the derivative of the target probability log-density to deliver guided, global moves, with higher acceptance probability. HMC is closely related to the so-called Metropolis-Adjusted Langevin Algorithm (MALA) [122] which uses the derivative of the log-density to propose steepest-ascent moves in the state space. MALA employs Langevin dynamics, i.e. "Brownian dynamics"; the proposal is derived from an Euler discretization of a Langevin stochastic differential equation that leaves the target density invariant. Despite the potential efficiency gains to be obtained in MCMC sampling from such proposal mechanisms that are inherent in the MALA and HMC methods, the tuning of these MCMC methods remains a major issue especially for challenging inference problems [66].

Algorithm 19: Multi-target Tracker

input : $\{\mathbf{X}_{k-1}^{(i)}\}_{i=1}^N$ and a new measurement, \mathbf{z}_k .

output: $\{\mathbf{X}_k^{(i)}\}_{i=1}^N$.

1 - *Predict Particles*:

Apply the KB-Smoother to obtain predicted particles(Algorithm 18)

$$\{\mathbf{X}_k^{(i)}\}_{i=1}^N = \text{KB-Smoother}\left(\{\mathbf{X}_{k-1}^{(i)}, 1/N\}_{i=1}^N\right).$$

2 - *Initialize the N_{MCMC} M-H samplers*:

Select a random subset $\{\mathbf{X}_s^{(0)}\}_{s=1}^{N_{\text{MCMC}}} \subset \{\mathbf{X}_k^{(i)}\}_{i=1}^N$.

Compute the associated joint log-likelihood $\ell_s^{(0)} = \log \vartheta_k(\mathbf{z}_k | \mathbf{X}_s^{(0)})$ Store the seeds in a cache $\{\mathbf{X}_s^{(0)}; \ell_s^{(0)}\}_{s=1}^{N_{\text{MCMC}}}$.

3 - *Apply random-walk M-H sampling procedure*:

Run $s = 1, \dots, N_{\text{MCMC}}$ random-walk M-H samplers in parallel (Algorithm 20)

$$\{\mathbf{X}_s^{(l)}\}_{l=1}^{N_{\text{M-H}}} = \text{RW-MCMC}\left(\{\mathbf{X}_s^{(0)}; \ell_s^{(0)}\}, \{\mathbf{X}_k^{(i)}\}_{i=1}^N\right).$$

3 - *Check convergence*:

Compute $\hat{\mathbf{R}}$: cf. Appendix C.2.3 eq. C.32.

if $\hat{\mathbf{R}} \geq 1.1$ **then**

 Run the chains out longer to improve convergence to the stationary distribution and increase the precision of inferences.

else

 Mix all the N_{MCMC} sets of simulations together to obtain $\{\mathbf{X}_k^{(i)}\}_{i=1}^N$.

end

Algorithm 20: Random Walk M-H sampling procedure

input : $\{\mathbf{X}_s^{(0)}; \ell_s^{(0)}\}$ initial configuration and $\{\mathbf{X}_k^{(i)}\}_{i=1}^N$.

output: $\{\mathbf{X}_s^{(l)}\}_{l=1}^{N_{\text{M-H}}}$ with $N_{\text{M-H}} := N/N_{\text{MCMC}}$.

for $l \leftarrow 1$ **to** $B + N_{\text{M-H}}$ **do**

 1 - *Propose a new configuration* \mathbf{X}_s^* :

 Randomly select a partition $\mathbf{x}_{k-1,j}^{(i)}$ out of $\{\mathbf{X}_{k-1}^{(i)}\}_{i=1}^N$.

 Given $\mathbf{x}_{k-1,j}^{(i)}$, draw $\mathbf{x}_{s,j}^* = \mathbf{x}_{k-1,j}^{(i)} + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma_{RW}^2)$.

 Propose new configuration \mathbf{X}_s^* , such that

$\mathbf{X}_s^* = \{\mathbf{x}_{s,1}^{(l-1)}, \dots, \mathbf{x}_{s,j-1}^{(l-1)}, \mathbf{x}_{s,j}^*, \mathbf{x}_{s,j+1}^{(l-1)}, \dots, \mathbf{x}_{s,n_k}^{(l-1)}\}$.

 Compute the joint log-likelihood $\ell_s^* = \ell_s^{(l-1)} - \mathbf{p}\ell_{s,j}^{(l-1)} + \mathbf{p}\ell_{s,j}^*$.

 2 - *Accept/Reject*:

 Sample $u \sim \mathcal{U}_{[0,1]}$, $\mathcal{U}_{[0,1]}$ a uniform distribution in $[0, 1]$.

 Calculate the M-H acceptance ratio: $\alpha = \frac{\ell_s^*}{\ell_s^{(l-1)}}$.

if $u \leq \min(1, \alpha)$ **then**

 | Accept move: $\mathbf{X}_s^{(l)} = \mathbf{X}_s^*$, $\ell_s^{(l)} = \ell_s^*$,

else

 | Reject move: $\mathbf{X}_s^{(l)} = \mathbf{X}_s^{(l-1)}$, $\ell_s^{(l)} = \ell_s^{(l-1)}$.

end

end

Discard B initial burn-in samples to allow the M-H sampler to converge to the stationary distribution.

Store the remaining samples $\{\mathbf{X}_s^{(l)}\}_{l=B+1}^{B+N_{\text{M-H}}} \rightarrow \{\mathbf{X}_s^{(l)}\}_{l=1}^{N_{\text{M-H}}}$.

6.3.2 Experimental Results

In this subsection, we assess and benchmark tracking performance achievable by combining the KB-Smoother and the IP-MCMC-PF. Simulations are performed in a typical ground radar scenario, where one or several targets move in the plane with unknown constant velocity. We adopt the dynamic model described in paragraph 4.6.1). Additional knowledge is available and is formalized in terms of constraints on the target dynamics, i.e.

$$\mathcal{C}_k \triangleq \{\mathbf{x}_{k,j} \in \mathbb{R}^5 : \mathbf{s}_{k,j} \in \mathfrak{P}_k\}, \quad (6.48)$$

where \mathfrak{P}_k is a polygon representing the constraints on the position and velocity vectors. Thus, the knowledge-based likelihood function $p(\bar{\mathcal{C}}_k|\mathbf{x}_{k,j})$ required for the *KB Fixed-Lag Smoother* is defined as follows:

$$p(\bar{\mathcal{C}}_k|\mathbf{x}_{k,j}) = \begin{cases} 1, & \text{if } \mathbf{s}_{k,j} \in \mathfrak{P}_k; \\ 0, & \text{Otherwise.} \end{cases} \quad (6.49)$$

Let us first consider a simplified version of the TBD problem, where a single-target moves within the surveillance area and never leaves the sensor FoV. In particular, we focus on the scenario depicted in fig. 6.1. The radar signal does not contain reflections from the target between $k \in [10, 21] \cup [41, 54]$ s. That is, during this time interval the tracking is carry out thanks to the available external knowledge. Notice that the lack of measurements starts just before road junctions. This means that in order to be able to track the target once a measurement is again available, the tracker has to keep track of both modalities (i.e., possible directions), which makes the loss of diversity among the particles a critical issue.

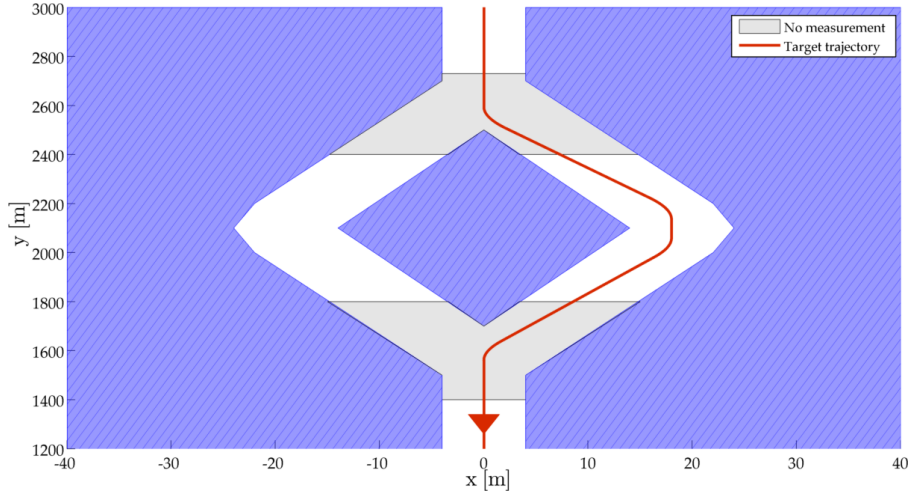


Figure 6.1: Scenario 1: a ship travels within a canal (shipping lane) at a nearly-constant speed of 30 m.s^{-1} . No measurement available between $k \in [10, 21] \cup [41, 54]$ s.

Figure 6.2 reports the empirical posterior PDF and the Minimum Variance estimate (MV) estimate over a single trial. Each filter uses 250 particles, and the KB-Smoother is implemented following the FFBS approach of Algorithm 18 for a lag $L = 5$.

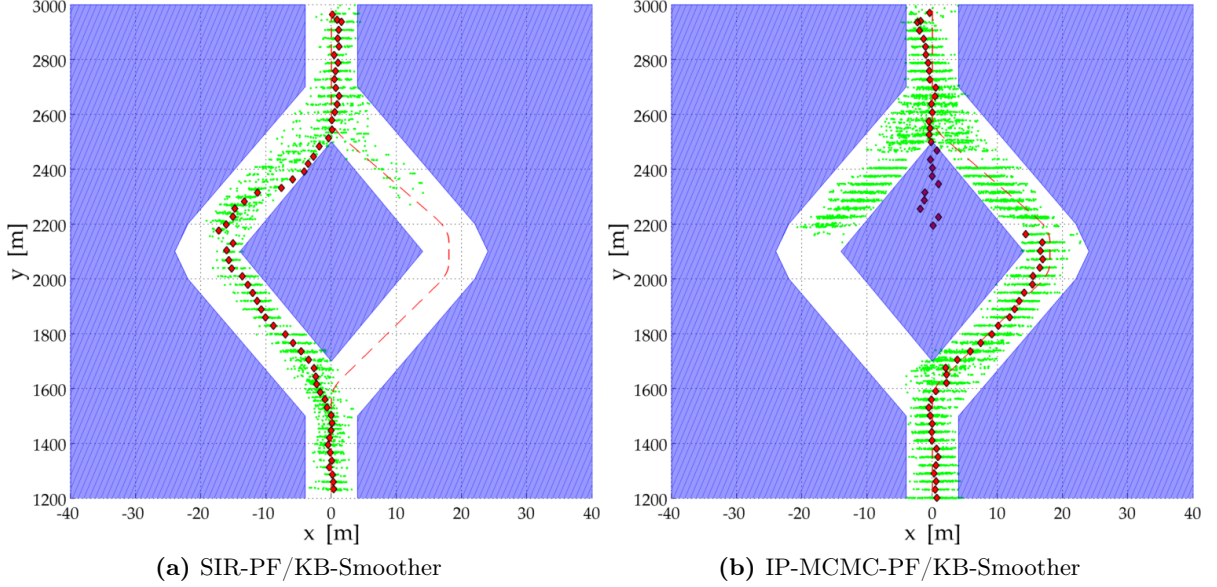


Figure 6.2: Empirical posterior distribution (■ *green*) and conditional mean (◆ *red*) over a single trial. When no measurement is available, due to the loss of particles diversity the SIR-PF arbitrarily retains one way (i.e., self resolving), while the IP-MCMC-PF keeps all possible directions.

That is, combining KB-Smoother and IP-MCMC-PF avoids loss of tracks due to wrong premature decisions. Furthermore, for large values of L the approximation to the smoothing distribution turns out to be strongly depleted, which might lead to poor performance. Given a number of particles N , the MCMC-PF alleviates the depletion problem, this way allowing the use of a longer lag L , which in turn leads to sensibly improved track accuracy.

The improvements in terms of uncertainty reduction and particles diversity are better shown in Figure 6.3.

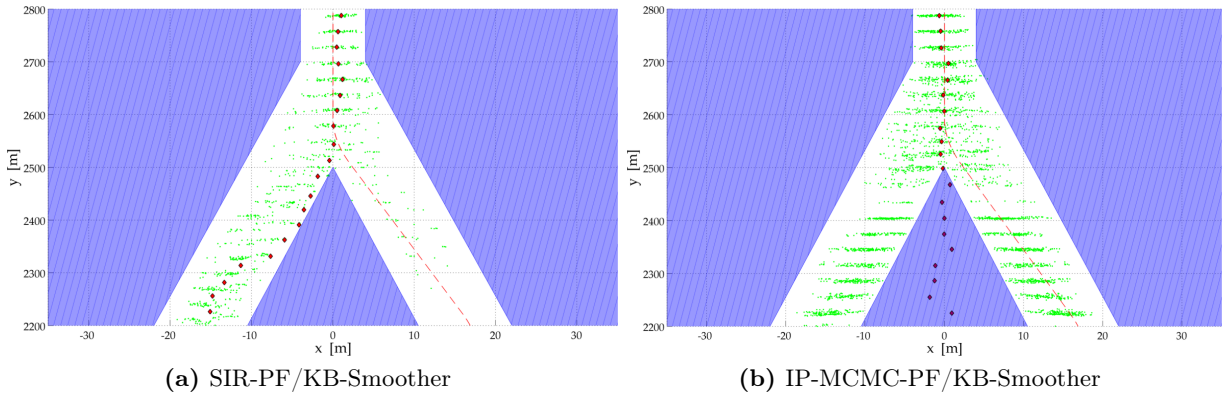


Figure 6.3: Empirical posterior distribution (■ *green*) and conditional mean (◆ *red*) over a single trial, zoom for $k \in [5, 25]$. Each filter uses a lag $L = 5$ and $N = 250$ particles.

Finally Figure 6.4 depicts the time-behavior of the conditional mean over 100 Monte Carlo trials for both combinations of algorithms.

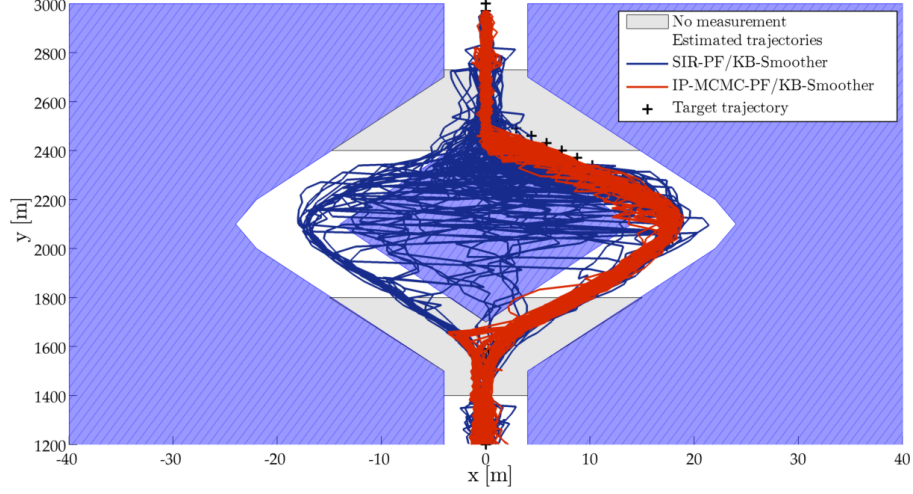


Figure 6.4: Estimated trajectories (i.e., particles-based conditional mean) over 100 Monte Carlo runs. Each filter uses a lag $L = 5$ and $N = 1000$ particles.

Let us now consider the multi-target scenario depicted in Figure 6.3.2, in which four targets move within a canal at nearly-constant speed.

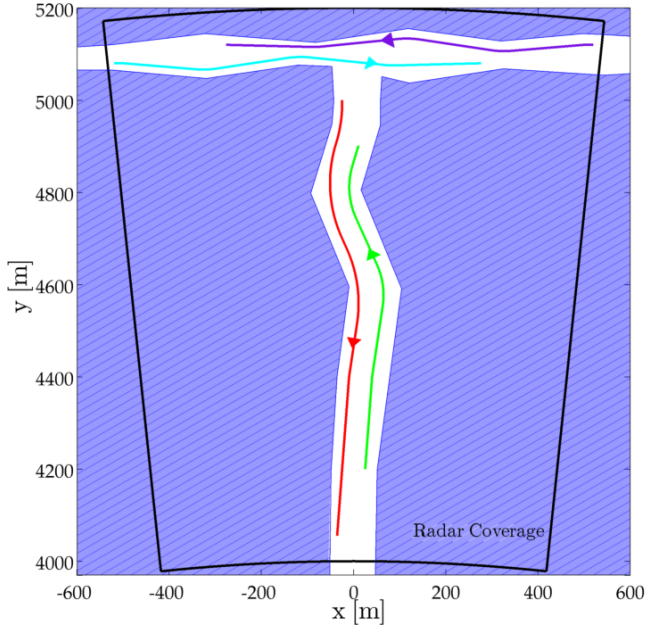


Table 6.1: Targets initial parameters

Target	SNR [dB]	$v [m.s^{-1}]$
1	9	12
2	11	9
3	10	10
4	10	10

Figure 6.5: Scenario 2: 4 ships travel within a canal at nearly-constant speed.

Figure 6.6 reports the empirical posterior PDF and MV estimate over a single trial. Each filter uses 1000 particles, and the KB-Smoother is implemented following the FFBS approach of Algorithm 18 for a lag $L = 4$.

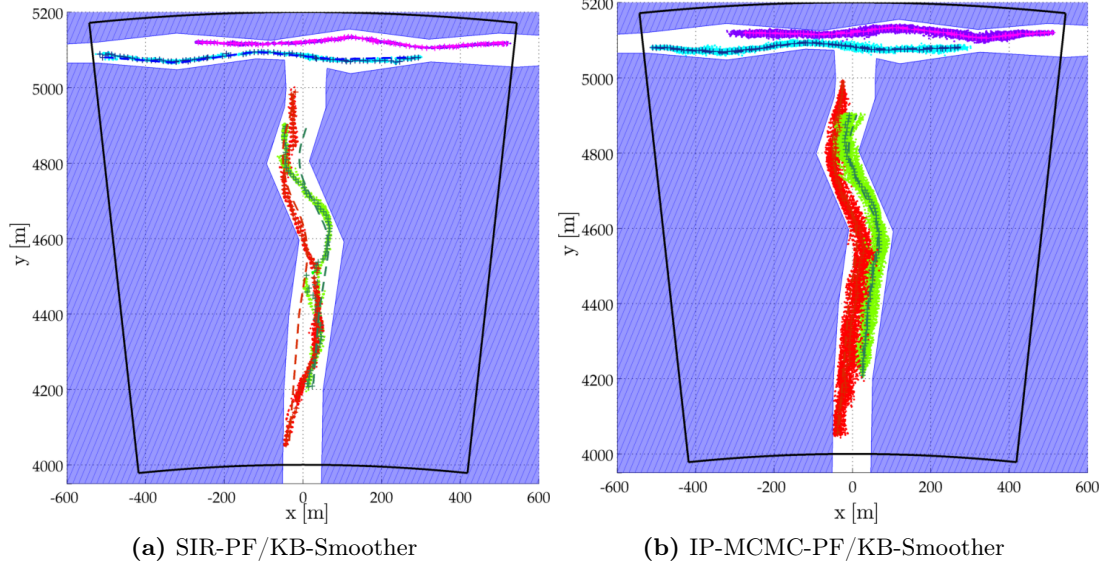


Figure 6.6: Empirical posterior distribution (.) and conditional mean (+) over a single trial. Each filter uses a lag $L = 4$ and $N = 1000$ particles.

Figure 6.7 depicts the time-behavior of the conditional mean over 100 Monte Carlo trials for both combinations of algorithms.

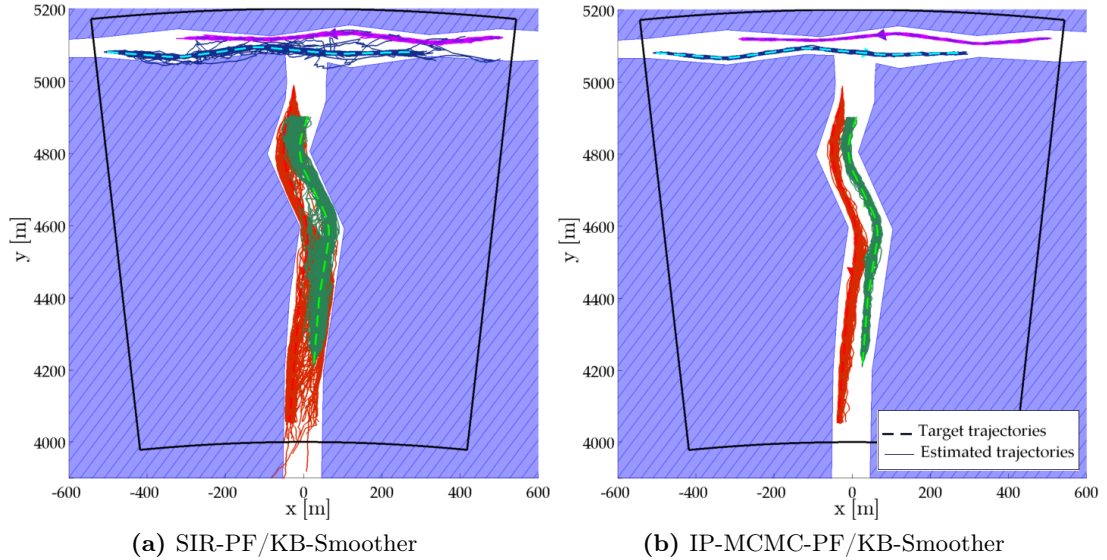


Figure 6.7: Estimated trajectories (i.e., particles-based conditional mean) over 100 Monte Carlo runs. Each filter uses a lag $L = 4$ and $N = 1000$ particles.

The SIR-PF/KB-Smoother is enable to correctly track the targets 1 and 2 after they have interacted due to an excessively depleted particles cloud. Conversely, the proposed combination IP-MCMC-PF/KB-Smoother yields to accurate and robust tracks. These figures further confirm that the proposed IP-MCMC-PF/KB-Smoother implementation provides better tracking performance than the plain SIR-PF/KB-Smoother combination. This holds in terms of track accuracy and robustness to a low number of particles.

Finally the performance of the methods is also compared via the the time-averaged RMSE, which are reported in Figure 6.8. The two approaches are analyzed according to the number of particles and the smoothing lags.

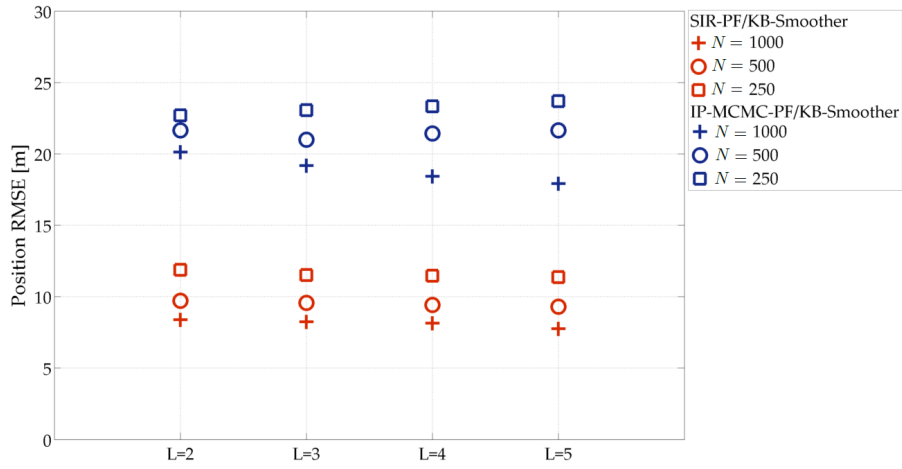


Figure 6.8: Time-Averaged Position RMSE over 100 Monte Carlo trials: SIR-PF and KB-Smoother (■ *blue*), IP-MCMC-PF and KB-Smoother (■ *red*).

As already mentioned, an increase of the smoothing lag leads in theory to a further reduced intrinsic uncertainty in the PDF. That is with a longer lag, the tracking performance should sensibly improve or at the very least be the same. However, Figure 6.8 shows that, given a finite number of particles, increasing the lag L with the SIR-PF may lead to worse tracking performance. This is due to an excessively depleted particles cloud. Conversely, sequential MCMC sampling allows us to fully exploit the potential benefits guaranteed by the KB-Smoother.

Conclusions

This chapter provides closing arguments for this dissertation. Specifically, a summary of contributions, critical assessments and conclusions, as well as directions for future works are presented.

7.1 Conclusions

This dissertation has addressed the Bayesian Multi-Target Tracking (MTT) problem dealing with an unknown and possibly time varying number of targets. MTT sets a major challenge for researchers in the broad fields of estimation and information fusion. Due to the complex nature of this class of problems, MTT methods usually involve wise implementation of tightly coupled data association and filtering schemes.

In this manuscript, we have formulated the multi-target tracking and estimation problem in the Bayesian Random Finite Set (RFS) framework and designed several novel, efficient and reliable tracking algorithms suitable for a Track-Before-Detect (TBD) radar surveillance application. The TBD approach proposes to base the tracking on the raw non-thresholded measurement data, thereby exploiting all available information to update the multi-target state estimate and allowing more accurate, sensitive and robust performance in comparison with traditional point measurement based approaches.

Specifically, a novel algorithm, well suited to deal with multi-target tracking problems for a given cardinality, has been derived from the Marginalized MCMC-Based Particle [84] method. The proposed algorithm, called IP-MCMC-PF and detailed in section 4.3, makes use of several Metropolis-Hastings samplers running in parallel and incorporates an interaction procedure for producing improved proposals. The performance of the proposed algorithm was then assessed and benchmarked in simulated scenarios, and shown to outperform conventional filtering approaches.

Moreover, to handle a large, unknown and time-varying number of targets, two approaches were introduced. The first approach (IP-RJMCMC-PF), detailed in section 4.4, exploits Reversible Jumps. However, this approach has been shown to strongly rely on the correct knowledge of the Markov matrix that is used to model the cardinality time evolution. The second approach (MCHPF), detailed in section 4.5, is built upon the concepts of labeled RFSs and multiple cardinality hypotheses. In simulations, the proposed MCHPF filter performs remarkably well, and outperforms the IP-RJMCMC-PF filter as well as the conventional multi-target SIR-PF in terms of track accuracy and consistency.

Furthermore, a computationally cheaper alternative, namely a multi-Bernoulli filter suitable for TBD applications, was proposed. The performance of the proposed algorithm was compared with the full multi-target PF implementation, and shown to offer an attractive alternative under low clutter and relatively high SNR.

Finally, a novel approach for multi-scan Bayes optimal knowledge exploitation was proposed. In multi-target scenarios, kinematic constraints from the interaction of targets with their environment or other targets can restrict target motion. A fixed-lag smoothing procedure, named Knowledge-Based fixed-lag Smoother (KB-Smoother) was utilized to propagate the particles given such motion constraint information. In simulations, the proposed combination IP-MCMC-PF/KB-Smoother yields significant gains in accuracy.

7.2 Directions for Future Works

Given the wide range of applicability of multi-target tracking and the great number of engineering problems related to the recursive estimation of the state of multiple targets, many questions remain open. There are some important possible extensions to the work discussed in this manuscript. They include, on one hand, generalizations of our algorithms to handle more complex scenarios and, on the other hand, the development of solutions for problems dealing with specific aspects of a tracking system. An outline of these possible directions for future research are sketched below.

- Various sampling improvement strategies can be incorporated into the Metropolis-Hastings (M-H) sampling procedure to enhance the efficiency of each proposed algorithm. In particular, notions from genetic algorithms and parallel tempering will be considered. The MCMC can become trapped in a local mode and fail to fully explore other modes which are significant. This problem is very similar to the one encountered in finding a global minimum in nonlinear optimization. One solution to that problem involves simulated annealing by introducing a temperature parameter. The analogous process applied to drawing samples from a target probability distribution is often referred to as tempering: instead of cooling down to make the distribution sharper and sharper, we rather heating up the distribution to make it flatter and flatter [102, 105].
- An investigation of the connection between the proposed MCHPF algorithm and Multiple Hypothesis Tracking (MHT) techniques [20, 117] can provide further insights.
- Smoothing or retrodiction, which uses measurements beyond the current estimation time and thus produces delayed estimates, results in better estimates of the targets state [57, 106]. An extension of these results to multi-target smoothing would be highly informative for reliable long-term trend analyses.
- A generalization of the RFS based algorithms introduced in this manuscript for maneuvering target tracking problems is essential in practice. In maneuvering target tracking problems, the detection and tracking of changes in the target motion model are crucial to get accurate state estimates. For such problems, Interacting Multiple dynamical Model (IMM) systems have been shown to be highly effective [98]. In particular, the combination of the IP-MCMC-PF with the IMM filter, detailed in [30], and the Multiple Model Multiple Hypothesis (M^3H) filter, first introduced in [58], need further investigation.

- Typically for point target models the underlying assumption is that the target occupies one resolution cell. For some sensor-targets configurations, this assumption is no longer suitable and the physical extent of the targets has to be accounted for in the processing. Thus, it would be interesting to investigate the problem of extended or unresolved targets. One possible approach assumes a diffuse spatial distribution over each target extent [63]. This means that we define a target model which describes how measurement sources are distributed over the target. In this way, the target extent and orientation are inferred from the measurements.
- In complex problems, fusion of multi-sensor data provides significant advantages over single-sensor data. In particular, the fusion of the data provided by two sensors with different capabilities will lead to enhanced tracking performance (better than either of the two independent sensors), but at a cost of additional computation and communication load. Thus, it would be valuable to develop a management procedure to select appropriate sensors in such a way that for a given computational load the effectiveness of the system is maximized.
- Finally, there is potential scope for applying target classification, to improve the reliability of the tracking process, and to derive behavioral features which may enhance knowledge about target identity and/or activities.

Remark. After the submission of the draft of this dissertation to the committee members, the author has been involved, with F. Papi, B.T. Vo and B.N. Vo, in the derivation of a labeled multi-Bernoulli TBD algorithm, which significantly differs from the plot based version [139]. This extension to TBD is partly based on the results described in subsection 3.2.3 and section 4.5. This work has resulted in a forthcoming conference publication [113].

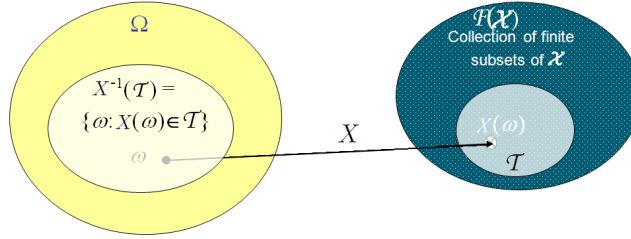
Finite Set Statistics (FISST)

This Appendix outlines the basics of Random Finite Set (RFS) and summarizes concepts of set-integrals and set-derivatives in the Finite-Set Statistics (FISST) framework. Detailed discussion about the mathematical foundations can be found in [46, 68, 91, 95].

A.1 Random Finite Set (RFS)

In essence, a Random Finite Set (RFS) X on \mathcal{X} (subset of \mathbb{R}^n) is defined as a measurable mapping from a sample space Ω with a probability measure \mathbb{P} defined on a σ -algebra of events $\sigma(\Omega)$ to $\mathcal{F}(\mathcal{X})$, the space of finite subsets of \mathcal{X} , which is equipped with the myopic topology.

$$X : \Omega \rightarrow \mathcal{F}(\mathcal{X}) \quad (\text{A.1})$$



In practice, an RFS (or spatial point process) is a probabilistic representation of spatial point patterns that accounts for uncertainty in both

- the number of objects,
- the spatial locations of the objects.

An RFS is simply a random variable that take values as unordered finite sets, i.e. a finite-set-valued random variable. The cardinality of an RFS variable X is random and modeled by a discrete distribution $\rho(n) = \Pr\{|X| = n\}$, where $n \in \mathbb{N}_0$. An RFS is completely specified by its cardinality distribution $\rho(n)$, $n \geq 0$ and a family of symmetric joint distributions¹ $p_n(x_1, \dots, x_n)$, $n \in \mathbb{N}$, that characterise the distribution of its elements over the state space, conditioned on cardinality n .

Let consider the probability distribution of the RFS X on \mathcal{X} , for any Borel subset \mathcal{T} of $\mathcal{F}(\mathcal{X})$, where $\{X \in \mathcal{T}\}$ denotes the measurable subset $\{\omega \in \Omega : X(\omega) \in \mathcal{T}\}$ of Ω , is the probability measure P on

¹A joint distribution function $p_n(x_1, \dots, x_n)$ is said to be symmetric if its value remains unchanged for all of the $n!$ possible permutations of its variables.

$\mathcal{F}(\mathcal{X})$ defined by:

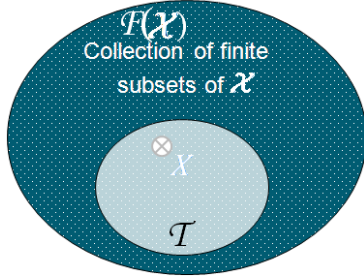
$$\begin{aligned} P_X(\mathcal{T}) &= \mathbb{P}(\{X \in \mathcal{T}\}) \\ &= P_X(\mathcal{T}_I) + P_X(\mathcal{T}_\infty) + \dots + P_X(\mathcal{T}_{\parallel}) + \dots \\ &= \int_{\mathcal{T}} p_X(X) \mu(dX) \end{aligned}$$

with, $p_X : \mathcal{F}(\mathcal{X}) \rightarrow [0, \infty)$. Note that a conventional integral is used.

One of the central entities in FISST is the belief-mass function (or belief functional) which is a useful descriptor of an RFS [95]. The belief mass function β_X of an RFS X , for all closed $\mathcal{K} \subseteq \mathcal{X}$, is defined by:

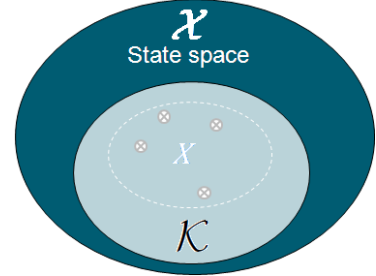
$$\begin{aligned} \beta_X(\mathcal{K}) &= \mathbb{P}(X \subseteq \mathcal{K}) \\ &= \mathbb{P}(X = \emptyset) + \mathbb{P}(X = x, x \in \mathcal{K}) + \dots + \mathbb{P}(X = x_1, \dots, x_k, x_i \in \mathcal{K}) + \dots \\ &= \int_{\mathcal{K}} f_X(X) \delta X \end{aligned}$$

with, $f_X : \mathcal{F}(\mathcal{X}) \rightarrow [0, \infty)$. Note that a set integral is used.



$$P_X(\mathcal{T}) = \mathbb{P}(\{X \in \mathcal{T}\})$$

(a) PROBABILITY DISTRIBUTION of X



$$\beta_X(\mathcal{K}) = \mathbb{P}(X \subseteq \mathcal{K})$$

(b) BELIEF-MASS FUNCTION of X

For the modeling of multi-target systems, the belief-mass function can be more convenient than the probability distribution, since the former deals with closed subsets of \mathcal{X} whereas the latter deals with subsets of $\mathcal{F}(\mathcal{X})$. The belief-mass functions defined directly on the closed subsets of the single-target state space and the single-target observation space allows descriptions of multi-target motion and measurements models to be systematically constructed from the single-target motion and observation models respectively. The belief-mass function, however, is not a measure and hence the standard measure theoretic notion of a density is not directly applicable. To circumvent this difficulty, the theory of FISST provides an alternative notion of density for belief functionals via the constructs of set-integrals and set-derivatives.

A.2 Multi-target Probability Density Functions

With point processes, we can construct a multi-target PDF of an RFS variable:

$$X = \underbrace{\{x_1, \dots, x_n\}}_{\text{set of individual target states}}. \quad (\text{A.2})$$

The multi-target FISST PDF is uniquely defined by a cardinality distribution $\rho(n)$ and a family of symmetric joint distributions $p_n(x_1, \dots, x_n)$ as:

$$p(X) = p(\{x_1, \dots, x_n\}) := n! \underbrace{\rho(n)}_{\text{cardinality}} \underbrace{p_n(x_1, \dots, x_n)}_{\text{joint spatial density}}. \quad (\text{A.3})$$

The $n!$ factorial term accounts for the fact we need to consider all permutations in the joint distribution. Both the likelihood and Markov densities are multi-target probability density functions:

$$\int \vartheta_k(Z|X) \delta Z = 1, \quad \int \Pi_{k|k-1}(X|X') \delta X = 1.$$

A.3 Set Integral

Being a PDF, $p(X)$ integrates to one using the set integral. The set integral over multi-target functions is defined as:

$$\int p(X) \delta X := p(\emptyset) + \sum_{n=1}^{\infty} \frac{1}{n!} \int p(\{x_1, \dots, x_n\}) dx_1 \dots dx_n. \quad (\text{A.4})$$

Now it is straightforward to verify that indeed $p(X)$ integrates to 1:

$$\int p(X) \delta X = \rho(0) + \sum_{n=1}^{\infty} \frac{1}{n!} n! \rho(n) \underbrace{\int p_n(x_1, \dots, x_n) dx_1 \dots dx_n}_{=1, \text{ being a standard PDF}} \quad (\text{A.5})$$

$$= \sum_{n=0}^{\infty} \rho(n) = 1 \quad (\text{since } \rho(n) \text{ is a discrete distribution}) \quad (\text{A.6})$$

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

$$\rho(n) = \frac{1}{n!} \int p(\{x_1, \dots, x_n\}) dx_1 \dots dx_n. \quad (\text{A.7})$$

A.4 Probability Generating Functionals (p.g.fl.)

Let $\mathcal{F}(\mathcal{X})$ denote the space of finite subsets of $\mathcal{X} \subseteq \mathbb{R}^n$. Suppose X is an RFS on \mathcal{X} , i.e. X is a random variable taking values in $\mathcal{F}(\mathcal{X})$.

The probability generating functional (p.g.fl.) is a fundamental and versatile descriptor of an RFS in multi-target filtering. The role of a probability generating functional in an RFS is analogous to the role of a probability generating function in a discrete random variable.

The p.g.fl. $G[h]$ of an RFS X on \mathcal{X} is defined by:

$$G_X[h] := \mathbb{E}(h^X) = \int h^X p(X) \delta X \quad (\text{A.8})$$

$$= \sum_{n=0}^{\infty} \frac{1}{n!} \int \left(\prod_{i=1}^n h(x_i) dx_i \right) p(x_1, \dots, x_n) \quad (\text{A.9})$$

$$= p(\emptyset) + \int h(x) p(\{x\}) dx + \int h(x_1) h(x_2) p(\{x_1, x_2\}) dx_1 dx_2 + \dots \quad (\text{A.10})$$

where h is any real-valued function on \mathcal{X} , such that $0 \leq h(x) \leq 1$ is a test function, and $h^X := \prod_{x \in X} h(x)$ with $h^\emptyset = 1$ by convention.

The p.g.fl. is a convenient summary of the statistical properties of an RFS which is analogous to the idea of transform methods. The p.g.fl. can also be used to derive useful properties of an RFS. The Janossy measures, probability density, moments and cardinality distribution can be recovered from the p.g.fl..

Relation to Janossy Measures and Densities

The p.g.fl. G can be expanded in terms of the Janossy measures $\{J^{(n)} : n \in \mathbb{N}\}$

$$G[h] = \sum_{n=0}^{\infty} \frac{1}{n!} J^{(n)}[h, \dots, h] \quad (\text{A.11})$$

where

$$J^{(n)}[h, \dots, h] = \int \dots \int h_1(x_1) \dots h_n(x_n) J^{(n)}(dx_1, \dots, dx_n) \quad (\text{A.12})$$

for real-valued $h_1(x_1) \dots h_n(x_n)$ functions on \mathcal{X} such that $0 \leq h_i(x) \leq 1$.

Relation to Factorial Moment Measures and Densities

The factorial moment measures can be obtained by differentiating G at $h = 1$.

Relation to Cardinality Distributions

The univariate probability generating function of the cardinality distribution of the RFS, i.e. the probability generating function $G_{|X|}(\cdot)$ of the discrete random variable $|X|$ can be obtained by substituting the constant function $h = y$ for $0 < y < 1$ into its probability generating functional $G_{|X|}(y) = G[y]$.

In the context of multi-target tracking, an alternative interpretation for the p.g.fl. is: let X be an RFS and $0 \leq h(x) \leq 1$ a function representing the probability of detection or FoV of some sensor, then $G_X[h]$ is the probability that X is contained in the FoV.

A.5 Functional Derivatives and Set Derivatives

The functional derivative of a functional $F[h]$ is given by

$$\frac{\delta F}{\delta \varphi}[h] = \lim_{\varepsilon \rightarrow 0} \frac{F[h + \varepsilon \varphi] - F[h]}{\varepsilon}. \quad (\text{A.13})$$

e.g. the functional derivative of a linear functional $f[h] = \int h(x)f(x)dx$ in direction $\varphi(x)$ is given by

$$\begin{aligned} \frac{\delta f}{\delta \varphi}[h] &= \lim_{\varepsilon \rightarrow 0} \frac{\int (h(x) + \varepsilon \varphi(x))f(x)dx - \int h(x)f(x)}{\varepsilon} \\ &= \int \varphi(x)f(x)dx \end{aligned}$$

now set $\varphi(x) = \delta_y(x)$, then $\frac{\delta f}{\delta \varphi}[h] = f(y)$. The FISST generalizes the functional derivative to sets.

Let $X = \{x_1, \dots, x_n\}$ be a set. The set derivative of $F[h]$ w.r.t. X is defined as:

$$\frac{\delta F}{\delta X}[h] := \frac{\delta F}{\delta x_1 \dots \delta x_n}[h] \quad (\text{A.14})$$

An important point to note is that $\frac{\delta F}{\delta X}[h]$ has unit of $K^{-|X|}$, hence for a fixed $S \subseteq \mathcal{X}$ the set-derivatives $\frac{\delta F}{\delta X}[h]$ and $\frac{\delta F}{\delta Y}[h]$ have different units if $|X| \neq |Y|$.

The fundamental theorem of multi-target calculus shows that the set integral and set derivative are inverse operations.

$$f(X) = (\delta F)_X(\emptyset) \text{ if and only if } F(S) = \int_S f(X) \delta X \quad (\text{A.15})$$

In p.g.fl. form, the theorem can be written

$$f(Y) = \left[\frac{\delta}{\delta Y} \int h^W . f(W) . \delta(W) \right]_{h=0}, \quad F[h] = \int h^W \frac{\delta F}{\delta W}[0] \delta W \quad (\text{A.16})$$

FISST provides simple formulae for computing set and functional derivatives, such as constant rule, power rule, sum rule, product rule, chain rule and so on.

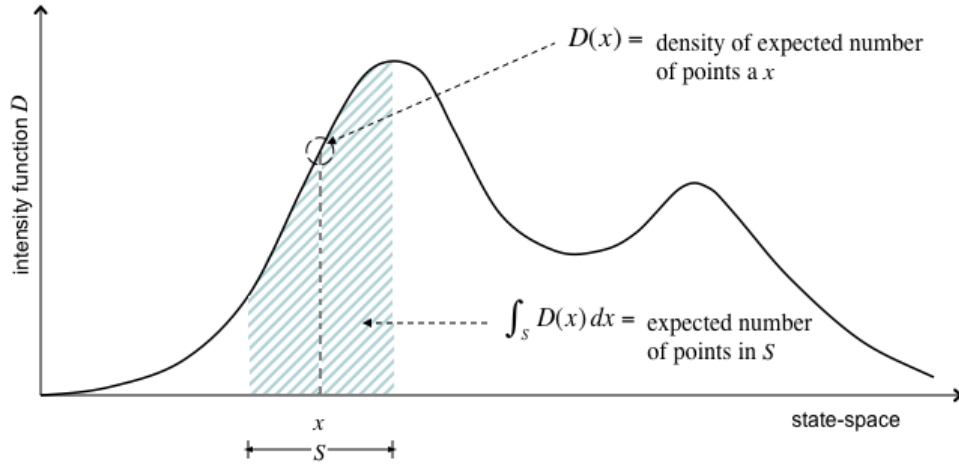
CONSTANT RULE	if $Y \neq \emptyset$, then $\frac{\delta}{\delta Y} K = 0$
POWER RULE	$\frac{\delta}{\delta Y} F[h]^N = N.F[h]^{N-1} \cdot \frac{\delta}{\delta Y} F[h]$
SUM RULE	$\frac{\delta}{\delta Y} (a_1.F_1[h] + a_2.F_2[h]) = a_1 \cdot \frac{\delta F_1}{\delta Y}[h] + a_2 \cdot \frac{\delta F_2}{\delta Y}[h]$
PRODUCT RULE	$\frac{\delta}{\delta Y} (F_1[h].F_2[h]) = \sum_{W \subseteq Y} \frac{\delta F_1}{\delta W}[h] \cdot \frac{\delta F_2}{\delta(Y-W)}[h]$
CHAIN RULE	$\frac{\delta}{\delta y} F_1[F_2[h]] = \delta(F_1 \circ F_2)(h; \varphi) = \delta F_1(F_2[h]; \delta F_2(h; \varphi))$
GENERAL CHAIN RULE [45]	$\delta^n (F_1 \circ F_2)(h; \varphi_1, \dots, \varphi_n) = \sum_{\pi \in \Pi(\varphi_{1:n})} \delta^{ \pi } F_1(F_2[h]; \xi_{\pi_1}(h), \dots, \xi_{\pi_{ \pi }}(h))$ where, $\xi_{\pi_i}(h)$ denotes the $ \pi_i ^{th}$ chain differential of F_2 along directions $\{\pi_i^1, \dots, \pi_i^{ \pi_i }\} \subseteq \{\varphi_1, \dots, \varphi_n\}$, i.e. $\xi_{\pi_i}(h) = \delta^{ \pi_i } F_2(h; \pi_i^1, \dots, \pi_i^{ \pi_i })$, $\Pi(\varphi_{1:n})$ represents the set of all partitions of the set $\{\varphi_1, \dots, \varphi_n\}$, and $ \pi $ denotes the cardinality of the set π .

A.6 Probability Hypothesis Density (PHD)

The intensity function, also known as the Probability Hypothesis Density (PHD) of an RFS X is defined as its first statistical moment:

$$D(x) = \mathbb{E}[\delta_X(x)] = \int \delta_X(x) p(X) \delta X, \quad (\text{A.17})$$

where $\delta_X(x) = \sum_{w \in X} \delta_w(x)$ is the set Dirac delta function, while $\delta_w(x)$ is the standard Dirac delta function concentrated at w . The intensity function of a point process is the average density of points (expected number of points per unit area).



The PHD can be found from the p.g.fl. as

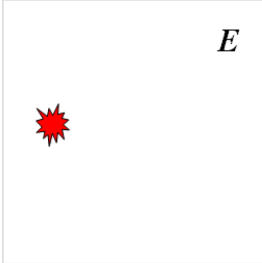
$$D(x) = \left. \frac{\delta}{\delta x} G[h] \right|_{h=1} \quad (\text{A.18})$$

Common Point Processes

We now present some specific examples of point processes, namely

- The Bernoulli Point Process
- The I.I.D. cluster process
- The Poisson Point Process
- The Poisson cluster process
- The multi-Bernoulli Point Process

B.1 Bernoulli RFS



Let assume that a point exists with probability r . If the point exists, the location of the point is distributed according to some spatial distribution $s(x)$. Then, we can define a Bernoulli RFS on \mathcal{X} with probability $(1-r)$ of being empty, and probability r of being a singleton whose only element is distributed according to s defined on \mathcal{X} .

Given (r, s) , a realization of the BERNOULLI RFS is obtained as:

- SAMPLE $u \sim U[0, 1]$;
- IF $u < r$ SAMPLE $x \sim s(\cdot)$ END;

The cardinality distribution of a Bernoulli RFS is a Bernoulli distribution with parameter r .

The intensity function of a Bernoulli point process is given by $D(x) = r s(x)$.

The multi-target density function for a Bernoulli process of random finite set variable X is

$$p(X) = \begin{cases} 1 - r, & X = \emptyset; \\ r s(x), & X = \{x\}; \\ 0, & |X| > 1. \end{cases} \quad (\text{B.1})$$

This density integrates to 1 by explicitly evaluating the set-integral,

$$\int p(X) \delta X := p(\emptyset) + \int p(\{x\}) dx \quad (\text{B.2})$$

$$= (1 - r) + r \int s(x) dx = 1. \quad (\text{B.3})$$

The p.g.fl. of a Bernoulli RFS is:

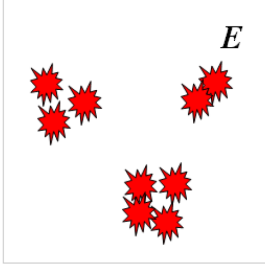
$$G[h] = \int p(X) \delta X = p(\emptyset) + \int h(x) p(\{x\}) dx \quad (\text{B.4})$$

$$= (1 - r) + r \int h(x) s(x) dx. \quad (\text{B.5})$$

while its probability density:

$$p(X) = \begin{cases} 1 - r, & X = \emptyset; \\ K r s(x), & X = \{x\}; \\ 0, & \text{Otherwise.} \end{cases} \quad (\text{B.6})$$

B.2 I.I.D. Cluster process



Let assume a number of points distributed according to the cardinality distribution $\rho(n)$. Each of the points is i.i.d. according to some spatial distribution $s(x)$. Then we can define an independent and identically distributed cluster RFS on \mathcal{X} given a matching intensity function $D(x)$ if its cardinality satisfies $\bar{N} = \sum_{n=0}^{\infty} n \rho(n) = \int D(x) dx$, and for any finite cardinality, the elements x of X are i.i.d. according to the probability density $s(x) := D(x)/\bar{N}$.

Given (ρ, s) , a realization of the I.I.D. CLUSTER RFS is obtained as:

- SAMPLE $n \sim \rho(\cdot)$
- FOR $i = 1 : n$ SAMPLE $x_i \sim p(\cdot)$ END

The intensity function of a i.i.d. Cluster process $D(x) = s(x) \cdot (\sum_{n>0} n \cdot \rho(n))$.

For any random set $X = \{x_1, \dots, x_n\}$ with cardinality $|X| = n$, the i.i.d. cluster process distribution is

$$p(X) := n! \cdot \underbrace{\rho(n)}_{\text{cardinality distribution}} \cdot \underbrace{\prod_{x \in X} s(x)}_{\text{joint spatial density}} \quad (\text{B.7})$$

The set-integral over this process is evaluated as follows

$$\int p(X) \delta X := \rho(0) + \sum_{n=1}^{\infty} \frac{1}{n!} n! \rho(n) \left(\int s(x) dx \right)^n \quad (\text{B.8})$$

$$= \sum_{n=0}^{\infty} \rho(n) = 1 \quad (\text{B.9})$$

The p.g.fl. of an i.i.d. cluster process is:

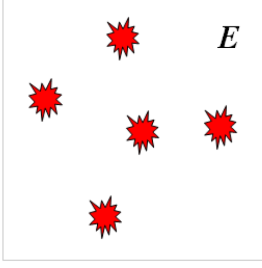
$$G[h] = \int p(X) \delta X = p(\emptyset) + \int h(x) p(\{x\}) dx \quad (\text{B.10})$$

$$= \sum_{n=0}^{\infty} \rho(n) \left(\int h(x) s(x) dx \right)^n \quad (\text{B.11})$$

The probability density of an i.i.d. cluster RFS can be written as:

$$p(\{x_1, \dots, x_n\}) := K^n n! \rho(n) \prod_{i=1}^n \frac{D(x_i)}{\bar{N}} \quad (\text{B.12})$$

B.3 Poisson RFS with Poisson rate $\lambda > 0$



Let assume that the expected number of points in the region S is λ . The locations of the points are i.i.d. according to some spatial distribution $s(x)$. Then, we can define a Poisson RFS on \mathcal{X} given an intensity function $D(x)$ if its cardinality is Poisson distributed, with mean $\bar{N} = \int D(x)dx$, and for any finite cardinality, the elements x of X are i.i.d. according to the probability density $s(x) := D(x)/\bar{N}$.

Given (λ, s) , a realization of the POISSON RFS is obtained as:

- SAMPLE $n \sim \text{Pois}(\lambda)$
- FOR $i = 1 : n$ SAMPLE $x_i \sim s(\cdot)$ END

The cardinality distribution of a Poisson RFS is specified by $\rho(n) = \frac{e^{-\lambda} \lambda^n}{n!}$.

The intensity function of a Poisson Point process $D(x) = \lambda s(x)$.

The multi-target distribution for a Poisson Point process of random finite set variable X is

$$p(X) := n! \frac{e^{-\lambda} \lambda^n}{n!} \prod_{x \in X} s(x) \quad (\text{B.13})$$

$$= e^{-\lambda} \lambda^n \prod_{x \in X} s(x) \quad (\text{B.14})$$

$$= e^{-\bar{N}} D^X. \quad (\text{B.15})$$

The p.g.fl. of a Poisson RFS is:

$$G[h] = \int h^X p(X) \delta X \quad (\text{B.16})$$

$$= e^{-\lambda} \sum_{n=0}^{\infty} \frac{1}{n!} \left(\int \lambda h(x) s(x) dx \right)^n \quad (\text{B.17})$$

$$= e^{\lambda \int h(x) s(x) dx - \lambda} \quad (\text{B.18})$$

$$= e^{\langle D, h-1 \rangle}. \quad (\text{B.19})$$

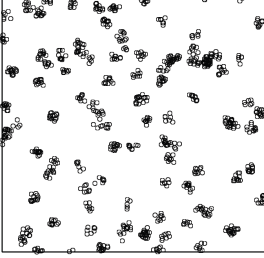
where $\langle D, h \rangle = \int D(x) h(x) dx$ is the standard notation for inner product. The probability density of a Poisson RFS can be written as:

$$p(\{x_1, \dots, x_n\}) := K^n e^{-N} \prod_{i=1}^n D(x_i) \quad (\text{B.20})$$

with K the unit of volume on \mathcal{X} .

B.4 Poisson cluster process

A doubly-stochastic (or hierarchical) point process:



- PARENT points are distributed according to a Poisson point process with rate λ and spatial distribution $s(c)$;
INTENSITY FUNCTION $D(c) = \lambda.s(c)$
- DAUGHTER points are distributed according to a Poisson point process with rate $\mu(c)$ and spatial distribution $u(x|c)$;
CONDITIONAL INTENSITY $D(x|c) = \mu(c).u(x|c)$.

The p.g.fl. of a Poisson cluster process is:

$$G[h] := G_c[\underbrace{G_m[h|\cdot]}_{\text{conditional daughter p.g.fl.}}] \quad (\text{B.21})$$

Parent p.g.fl.

$$= \exp \left(\lambda \int G_m[h|c] s(c) dc - \lambda \right) \quad (\text{B.22})$$

$$= \exp \left(\lambda \int s(c) \underbrace{\exp \left(\mu(c) \int u(x|c).h(x|c) dx - \mu(c) \right)}_{\text{conditional daughter Poisson p.g.fl.}} dc - \lambda \right). \quad (\text{B.23})$$

Parent Poisson p.g.fl.

B.5 Multi-Bernoulli RFS

A multi-Bernoulli RFS on \mathcal{X} is a union of a fixed number of independent Bernoulli RFSs $X^{(i)}$ with existence probability $r^{(i)} \in (0, 1)$ and probability density $s^{(i)}$ defined on \mathcal{X} for $i = 1, \dots, M$, i.e.

$$X = \bigcup_{i=1}^M X^{(i)}. \quad (\text{B.24})$$

It follows that the mean cardinality of a multi-Bernoulli RFS is $\sum_{i=1}^M r^{(i)}$.

Using the independence of the $X^{(i)}$, the p.g.fl. of a multi-Bernoulli RFS is

$$G[h] = \prod_{i=1}^M (1 - r^{(i)} + r^{(i)} \langle s^{(i)}, h \rangle) \quad (\text{B.25})$$

A multi-Bernoulli RFS is thus completely described by the corresponding multi-Bernoulli parameter set $\{(r^{(i)}, s^{(i)})\}_{i=1}^M$ equivalently $\{q^{(i)} = \frac{r^{(i)} s^{(i)}}{1-s^{(i)}}\}_{i=1}^M$. Its probability density is:

$$p(\{x_1, \dots, x_n\}) := \begin{cases} \prod_{j=1}^M (1 - r^{(j)}), & n = 0; \\ K^n \prod_{j=1}^M (1 - r^{(j)}) \sum_{1 \leq i_1 \neq \dots \neq i_n \leq M} \prod_{j=1}^n \frac{r^{(i_j)} s^{(i_j)}(x_j)}{1 - r^{(i_j)}}, & n \leq M \\ 0, & n > M. \end{cases} \quad (\text{B.26})$$

Alternatively, (B.26) can be written more compactly in terms of the parameters $\{q^{(i)}\}_{i=1}^M$ as follows

$$p(\{X\}) := K^{|X|} \left(\frac{1}{1 - \langle q^{(\cdot)}, 1 \rangle} \right)^{\{1, \dots, M\}} \sum_{\tau \in \mathcal{T}(X, \{1, \dots, M\})} q_{\tau}^X, \quad (\text{B.27})$$

where $q_{\tau}(x) = q^{(\tau(x))}(x)$ and $\mathcal{T}(X, Y)$ denotes the set of all one-to-one functions taking a finite set X to a finite set Y , with the convention that the summation over $\mathcal{T}(X, Y)$ is zero when $|X| > |Y|$ and unity when $X = \emptyset$.

Note that $1 - r^{(i)} = \frac{1}{1 - \langle q^{(\cdot)}, 1 \rangle}$ and recall that $q^X \equiv \prod_{x \in X} q(x)$.

Markov Chain Monte Carlo (MCMC)

In recent years there has been an increasing interest towards Markov Chain Monte Carlo (MCMC) methods to simulate complex, nonstandard multivariate distributions. This appendix is organized as follows. Section C.1 introduces the fundamental concepts of Monte Carlo methods, rejection sampling and importance sampling. Section C.2 recalls the basic of Markov Chain Monte Carlo process, and describes two basic MCMC techniques: the Metropolis-Hastings (M-H) [75,101], and the Gibbs sampler [40,61].

C.1 Monte Carlo Integration

C.1.1 The Monte Carlo Principle

Monte Carlo simulation consists to draw an i.i.d. set of samples $\{x^{(i)}\}_{i=1}^N$ from a target density p defined on a high-dimensional space \mathcal{X} . These N samples can be used to approximate the target density with the following empirical point-mass function

$$p_N(x) = \frac{1}{N} \sum_{i=1}^N \delta_{x^{(i)}}(x), \quad (\text{C.1})$$

where $\delta_{x^{(i)}}(x)$ denotes the delta-Dirac mass located at $x^{(i)}$. Thus, one can approximate the integrals of the form

$$\mathbb{E}_p[f(x)] = \int_{\mathcal{X}} f(x) p(x) dx \quad (\text{C.2})$$

with tractable sums \bar{f}_N that converge as follows:

$$\bar{f}_N = \frac{1}{N} \sum_{i=1}^N f(x^{(i)}) \xrightarrow[N \rightarrow \infty]{\text{a.s.}} \int_{\mathcal{X}} f(x) p(x) dx = \mathbb{E}_p[f(x)]. \quad (\text{C.3})$$

That is, the estimate \bar{f}_N is unbiased and by the Strong Law of Large Numbers (SLLN), it will almost surely (a.s.) converge to $\mathbb{E}_p[f(x)]$.

If the variance of $f(x)$ satisfies

$$\sigma_f^2 \triangleq \mathbb{E}_p[f^2(x)] - \mathbb{E}_p[f(x)]^2 < \infty, \quad (\text{C.4})$$

then the variance of the estimate \bar{f}_N is equal to

$$\text{var}[\bar{f}_N] = \frac{\sigma_f^2}{N} \quad (\text{C.5})$$

and the Central Limit theorem yields convergence in distribution of the error

$$\sqrt{N} (\bar{f}_N - \mathbb{E}_p[f(x)]) \xrightarrow[N \rightarrow \infty]{} \mathcal{N}(0, \sigma_f^2), \quad (\text{C.6})$$

where \Rightarrow denotes convergence in distribution [121].

The advantage of Monte Carlo integration over deterministic integration arises from the fact that the former positions the integration grid (samples) in regions of high probability. When $p(x)$ has standard form, e.g. Gaussian, it is straightforward to sample from it using easily available routines. However, in most cases, more advanced techniques based on Rejection sampling, Importance sampling or even Markov Chain Monte Carlo techniques.

C.1.2 Rejection Sampling

We can sample from a distribution $p(x)$, which is known up to a proportionality constant, by sampling from another easy-to-sample proposal distribution $q(x)$ that satisfies $p(x) \leq M q(x)$, $1 < M < \infty$, using the ACCEPT/REJECT procedure describe in Algorithm 21 and Fig. C.1.

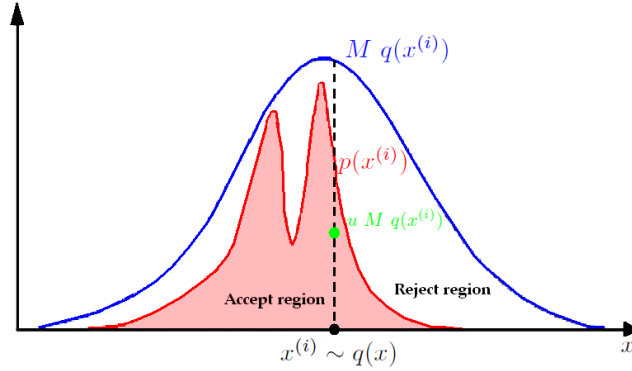


Figure C.1: Rejection sampling: Sample a candidate $x^{(i)}$ and a uniform variable u . Accept the candidate sample if $u M q(x^{(i)}) < p(x^{(i)})$, otherwise reject it.

Algorithm 21: Rejection-Sampling algorithm

```

Set  $i = 1$ 
while  $i < N$  do
    Sample  $x^{(i)} \sim q(x)$  and  $u \sim \mathcal{U}(0, 1)$ 
    if  $u < \frac{p(x^{(i)})}{M q(x^{(i)})}$  then
        | Accept  $x^{(i)}$  and  $i = i + 1$ 
    else
        | Reject
    end
end
end

```

This simple method suffers from severe limitations. First, it requires to bound $\frac{p(x)}{q(x)}$ with a reasonable constant M over the whole space \mathcal{X} . Second, if M is too large, the acceptance probability α given by the proportion of accepted samples

$$\alpha = Pr \left(u < \frac{p(x)}{M q(x)} \right) = \frac{1}{M}, \quad (\text{C.7})$$

will be too small. This makes the method impractical in high-dimensional scenarios. In high dimensions, it is necessary to use a different approach, typically an MCMC method such as M-H sampling or Gibbs sampling.

C.1.3 Importance Sampling

Let us consider an arbitrary importance proposal distribution $q(x)$ such that its support includes the support of $p(x)$. Then $\mathbb{E}_p[f(x)]$ can be rewritten as follows:

$$\mathbb{E}_p[f(x)] = \int_{\mathcal{X}} f(x) w(x) q(x) dx = \mathbb{E}_q \left[\frac{f(x) p(x)}{q(x)} \right], \quad (\text{C.8})$$

where $w(x) \triangleq \frac{p(x)}{q(x)}$ known as importance weight.

So if one can sample N i.i.d draws $\{x^{(i)}\}_{i=1}^N$ from q instead of p and evaluate $w(x^{(i)})$, a possible estimate of $\mathbb{E}_p[f(x)]$ is given by:

$$\hat{f}_N = \sum_{i=1}^N f(x^{(i)}) w(x^{(i)}), \quad (\text{C.9})$$

as long as $\text{var}[f(x)w(x)] < \infty$ and $\text{support}(f \times p) \subset \text{support}(q)$. This estimate is unbiased and, under weak assumptions, the SLLN applies, that is

$$\hat{f}_N \xrightarrow[N \rightarrow \infty]{\text{a.s.}} \mathbb{E}_p[f(x)] \quad (\text{C.10})$$

This integration method can be interpreted as a sampling method where the posterior density $p(x)$ is approximated by the empirical measure:

$$\hat{p}_N(x) = \sum_{i=1}^N w(x^{(i)}) \delta_{x^{(i)}}(x) \quad (\text{C.11})$$

A suitably designed proposal distribution $q(x)$ is fundamental. Such distributions should be easy to sample from and should minimize the variance of the estimate \hat{f}_N given by

$$\text{var}[\hat{f}_N] = \frac{\sigma_f^2}{N} = \frac{1}{N} \left(\mathbb{E}_q[f^2(x) w^2(x)] - \mathbb{E}_p[f(x)]^2 \right), \quad (\text{C.12})$$

The second term does not depend on $q(x)$ and hence only the first term need to be minimized, which according to Jensen's inequality has the following lower bound

$$\mathbb{E}_q[f^2(x) w^2(x)] \geq (\mathbb{E}_q[|f(x)| w(x)])^2 = \left(\int |f(x)| p(x) dx \right)^2. \quad (\text{C.13})$$

This lower bound is attained when the optimal importance distribution $q^*(x)$ is adopted

$$q^*(x) = \frac{|f(x)| p(x)}{\int |f(x)| p(x) dx} \quad (\text{C.14})$$

Although this not easy-to sample optimal proposal can not be retained, this tells us that high sampling efficiency can be achieved by sampling from $p(x)$ in region where $|f(x)| p(x)$ is large.

When the normalizing constant of $p(x)$ is unknown, it is still possible to apply the importance sampling method by rewriting $\mathbb{E}_p[f(x)]$ as follows:

$$\mathbb{E}_p[f(x)] = \frac{\int_{\mathcal{X}} f(x) w(x) q(x) dx}{\int_{\mathcal{X}} w(x) q(x) dx}, \quad (\text{C.15})$$

where $w(x) \propto \frac{p(x)}{q(x)}$ is now only known up to a normalizing constant. The Monte Carlo approximation of $\mathbb{E}_p[f(x)]$ becomes

$$\tilde{f}_N = \frac{\frac{1}{N} \sum_{i=1}^N f(x^{(i)}) w(x^{(i)})}{\frac{1}{N} \sum_{i=1}^N w(x^{(i)})} = \sum_{i=1}^N f(x^{(i)}) \tilde{w}(x^{(i)}) \quad (\text{C.16})$$

where $\tilde{w}(x^{(i)})$ is a normalized importance weight.

For N finite, \tilde{f}_N is biased but asymptotically, under weak assumptions, the SLLN applies, that is

$$\tilde{f}_N \xrightarrow[N \rightarrow \infty]{\text{a.s.}} \mathbb{E}_p[f(x)]. \quad (\text{C.17})$$

If one is interested in obtaining M i.i.d. samples from $\hat{p}_N(x)$, then an asymptotically valid method consists of resampling M times according to the discrete distribution $\hat{p}_N(x)$. This procedure results in M samples $\tilde{x}^{(i)}$ with replicates. After resampling, the approximation of the target density is

$$\check{p}_M(x) = \sum_{i=1}^M \delta_{x^{(i)}}(x) \quad (\text{C.18})$$

There are several ways of implementing this basic idea, the most obvious approach being sampling with replacement, with the probability of sampling each $x^{(i)}$ set equal to the normalized importance weight $\tilde{w}(x^{(i)})$. The resampling scheme introduces some additional Monte Carlo variation.

Remark 1: The standard Particle Filter approach resorts to a sequential version of the importance sampling and resampling procedure outlined above. Although the resampling step increases the Monte Carlo variance, in a sequential scheme this step is crucial to avoid degeneracy of the importance weights over time. Resampling however introduces other practical problems. First, it limits parallel processing since all particles have to be combined. Second, by resampling the posterior distribution, the heavily weighted particles are statistically selected several times. This leads to a loss of diversity among the particles, the sample impoverishment problem. In the extreme case, all the particles collapse to the same location. The sample impoverishment leads to failure in tracking since less diverse particles are used to represent the uncertain dynamics of the moving object.

Remark 2: As the dimension of the x increases, it becomes harder to obtain a suitable $q(x)$ from which to draw samples. A poor selection of the proposal distribution $q(x)$ will lead to a high-variance estimate

\hat{f}_N . A sensible strategy is to adopt a parameterized $q(x, \theta)$ and to adapt θ during the simulation [49]. However, even with adaptation, it is often impossible to obtain proposal distributions that are both easy to sample from and good approximations. To deal with multimodal posterior probability density functions in large dimensional state spaces, more advanced sampling algorithms based on Markov chains need to be considered.

C.2 Markov Chain Monte Carlo

In Figure C.2, side-by-side simulations of the SIR sampler and the M-H sampler is provided, for comparison. Each filter jointly tracks two one-dimensional targets, using $N = 200$ particles. The target dynamics are governed by a AutoRegressive (AR)(1)-process, i.e $\mathbf{x}_k = \mathbf{x}_{k-1} + \varepsilon_k$, where $\varepsilon_k \sim \mathcal{N}(0, 0.5)$. The likelihood model is defined as $p(\mathbf{z}_k | \mathbf{x}_k) = \exp(-3|\mathbf{z}_k - \mathbf{x}_k|)$.

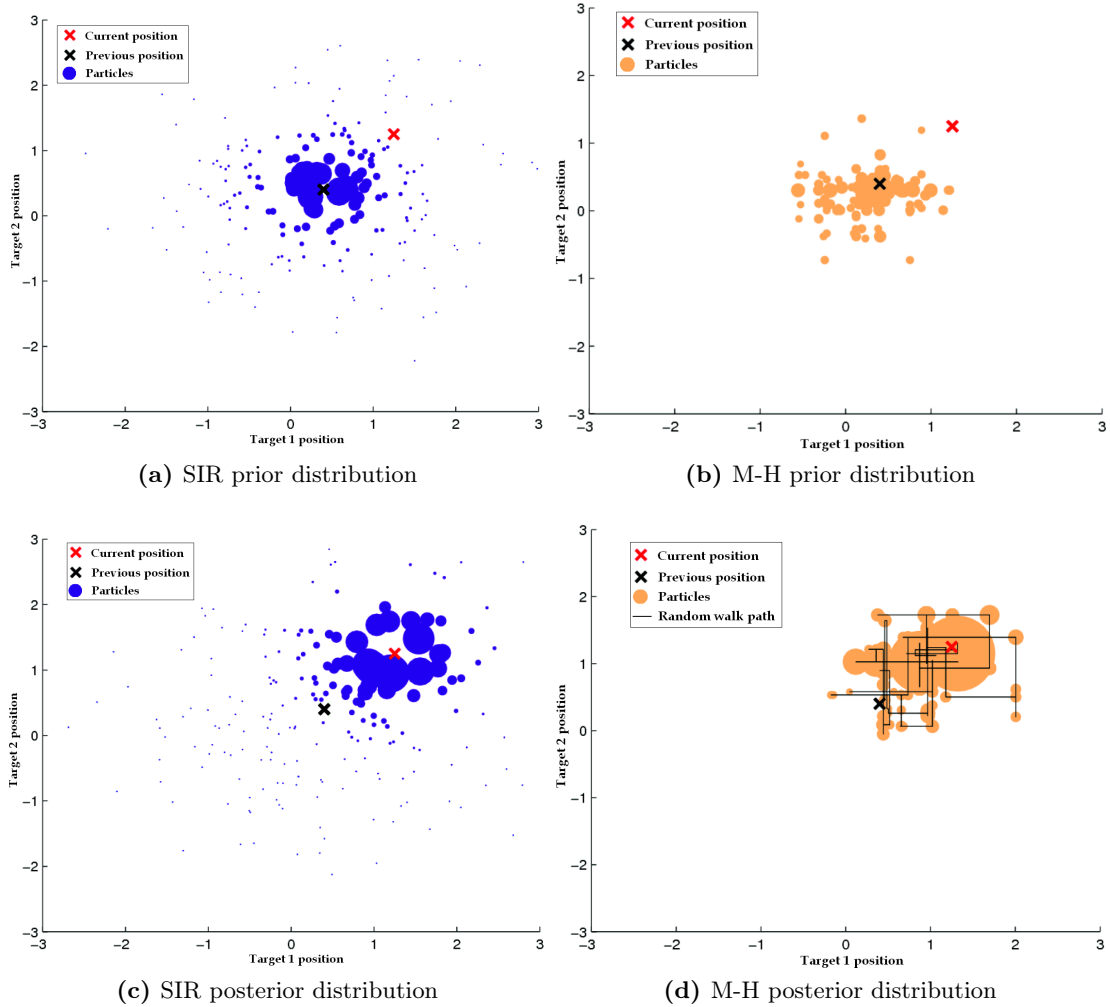


Figure C.2: SIR vs M-H

C.2.1 Basic of Markov Chain Monte Carlo

Markov Chain Monte Carlo (MCMC) algorithms turn around the Markov chain theory. Consider a state vector $x \in \mathcal{X} \subseteq \mathbb{R}^{N_x}$ a compact set, $K(\cdot, \cdot)$ is assumed to be a transition kernel in the state space, which represents the probability of moving from x to a point in a set $S \in \mathcal{B}(\mathcal{X})$ the Borel σ -field on \mathcal{X} , a Markov chain is a sequence of random variable $\{x^{(i)}\}_{i \geq 1}$ such that

$$Pr(x^{(i)} \in \mathcal{B}|x^{(1)}, \dots, x^{(i-1)}) = Pr(x^{(i)} \in \mathcal{B}|x^{(i-1)}) \quad (\text{C.19})$$

and $K(x^{(i-1)}, x^{(i)}) = p(x^{(i)}|x^{(i-1)})$. In finite state space the transition kernel k becomes the matrix of transition probabilities T .

Suppose that $p(x)$ is a probability distribution on \mathcal{X} . Denote by $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$ the associated measure space. Then, an MCMC method for the simulation of the distribution f is any method which generates an ergodic, reversible Markov chain $\{x^{(i)}\}_{i \geq 1}$ on \mathcal{X} , according to a kernel $K(\cdot, \cdot)$ defined on $(\mathcal{X} \times \mathcal{B}(\mathcal{X}))$, with f as stationary distribution.

Let us recall three basic concepts:

1. f is the stationary distribution of the Markov chain, i.e., for all $x' \in \mathcal{B}(\mathcal{X})$,

$$p(x') = \int_{\mathcal{X}} K(x, x')p(x)dx. \quad (\text{C.20})$$

2. The Markov chain is ergodic, i.e. for any integrable function $f : \mathcal{X} \rightarrow \mathbb{R}$

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N f(x^{(i)}) \rightarrow \mathbb{E}_p[f] = \int_{\mathcal{X}} f(x)p(x)dx \quad (\text{C.21})$$

with probability 1, $\forall x^{(i)} \in \mathcal{X}$, where N is the number of iterations of the MCMC algorithm.

3. The Markov chain is reversible, i.e. satisfies the detailed balance which states that $\forall x, x' \in \mathcal{B}(\mathcal{X})$

$$p(x) K(x, x') = p(x') K(x', x) \quad (\text{C.22})$$

So, in order to generate samples from p , the MCMC methods attempt to find a transition kernel $K(\cdot, \cdot)$ whose iterations converges to f given an arbitrary starting point. MCMC Markov chains are irreducible¹ and aperiodic².

C.2.2 Metropolis-Hastings and Gibbs Samplers

The Metropolis-Hastings (M-H) algorithm is a well known procedure based on a Markovian process which fulfills the requirement of ergodicity [4]. The M-H algorithm employs a conditional density $q(x'|x)$, also known as proposal distribution, to generate a Markov chain with an invariant distribution $p(x)$. Assume $q(x'|x)$ does not satisfy the reversibility condition, without loss of generality, suppose

¹A Markov chain is called irreducible if any state can be reached from any other state in a finite number of iterations

²A Markov chain is called aperiodic if there exists an $n < \infty$ such that $Pr(x^{(n')} = i | x^{(1)} = i) > 0 \forall n' \geq n$

$p(x) q(x'|x) > p(x') q(x|x')$, then to compensate a probability of move acceptance $\alpha(x, x')$ is introduced. Hence, the transition kernel for the M-H algorithm becomes

$$p_{M-H}(x'|x) = q(x'|x) \alpha(x, x'). \quad (\text{C.23})$$

In order to satisfy reversibility condition, $\alpha(x, x')$ is set to

$$\alpha(x, x') = \begin{cases} \min\left(\frac{p(x') q(x|x')}{p(x) q(x'|x)}, 1\right), & \text{if } p(x) q(x'|x) > 0; \\ 1, & \text{Otherwise.} \end{cases} \quad (\text{C.24})$$

Hence the probability that Markov process stays at x is given by $1 - \int_{\mathcal{X}} q(y|x) \alpha(x, y) dy$ and the transition kernel is given by

$$K_{M-H}(x'|x) = q(x'|x) \alpha(x, x') + \underbrace{\delta_x(x') \int_{\mathcal{X}} q(y|x) (1 - \alpha(x, y)) dy}_{\text{Rejection term}}. \quad (\text{C.25})$$

The M-H algorithm is summarized in Algorithm 22 [65]. At each MCMC iteration i , a move $x' \sim q(x'|x^{(i-1)})$ is proposed and accepted with a probability $0 < \alpha(x^{(i-1)}, x') < 1$.

Algorithm 22: M-H algorithm

input : $x^{(0)}$ initial configuration, B burn-in period, N required number of samples.

output: $\{x^{(i)}\}_{i=1}^N$.

for $i \leftarrow 1$ **to** $B + N$ **do**

Given current state $x^{(i-1)}$, draw x' from the proposal density function $q(x'|x^{(i-1)})$.

Sample $u \sim \mathcal{U}_{[0,1]}$, where $\mathcal{U}_{[0,1]}$ a uniform distribution in $[0, 1]$.

Calculate the acceptance ratio : $\alpha = \frac{p(x') q(x^{(i-1)}|x')}{p(x^{(i-1)}) q(x'|x^{(i-1)})}$

if $u \leq \min(1, \alpha)$ **then**

| Accept move: $x^{(i)} = x'$,

else

| Reject move: $x^{(i)} = x^{(i-1)}$.

end

end

Discard B initial burn-in samples, store the other samples.

Only the samples that are drawn after the Markov chain approaches the equilibrium, i.e. converges to the stationary distribution, are regarded as the representative draws from the posterior. The initial burn-in samples strongly influenced by the initial configuration are discarded. In addition, to reduce the correlation between samples, it is customary to thin out the samples by only keeping a subset of them, taken at regular intervals.

Some properties of the M-H algorithm are worth highlighting. Firstly, the normalizing constant of the target distribution is not required. We only need to know the target distribution up to a constant of

proportionality. Secondly, although the pseudo-code makes use of a single chain, it is easy to simulate several independent chains in parallel. Lastly, the success or failure of the algorithm often hinges on the choice of proposal distribution. A suitably designed proposal distribution is fundamental to guarantee a well mixing Markov chain. Such distributions should both be easy to sample from and allow the Markov chain to explore all the high density regions of \mathcal{X} under p freely. In particular, the design of the proposal distribution becomes central when dealing with high dimensional and interdependent state vectors. In high dimensions, local strategies, focusing on some of the subcomponents of p , e.g. propose to move one target at each time, will be used in practice, to break up the original sampling problem into simpler ones. Nevertheless, these may be prone to poor performance, as local strategies inevitably ignore some of the global features of p .

Gibbs sampling can be viewed a special form of M-H algorithm in which the proposal distribution is defined in terms of the conditional distributions of the joint distribution and every proposal is always accepted. That is, suppose we have an n -dimensional vector $x = [x_1, x_2, \dots, x_n]$ and the expressions for the full conditionals $p(x_j | x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n)$. In this case, the following proposal distribution can be used for $j = 1, \dots, n$,

$$q(x' | x^{(i-1)}) = \begin{cases} p(x'_j | x_{-j}^{(i-1)}), & \text{if } x'_{-j} = x_{-j}^{(i-1)}; \\ 0, & \text{Otherwise.} \end{cases} \quad (\text{C.26})$$

Thus the corresponding acceptance probability is

$$\alpha(x^{(i-1)}, x') = \min \left(1, \frac{p(x') q(x^{(i-1)} | x')}{p(x^{(i-1)}) q(x' | x^{(i-1)})} \right) = 1. \quad (\text{C.27})$$

The Gibbs sampler is summarized in Algorithm 23 [62].

Algorithm 23: Gibbs sampler

$x_{1:n}^{(0)}$ initial configuration.

for $i \leftarrow 1$ **to** $B + N$ **do**

Sample $x_1^{(i)} \sim p(x_1 | x_2^{(i-1)}, x_3^{(i-1)}, \dots, x_n^{(i-1)})$,

Sample $x_2^{(i)} \sim p(x_2 | x_1^{(i-1)}, x_3^{(i-1)}, \dots, x_n^{(i-1)})$,

.

.

.

Sample $x_j^{(i)} \sim p(x_j | x_1^{(i-1)}, \dots, x_{j-1}^{(i-1)}, x_{j+1}^{(i-1)}, \dots, x_n^{(i-1)})$,

.

.

.

Sample $x_n^{(i)} \sim p(x_n | x_1^{(i-1)}, x_2^{(i-1)}, \dots, x_{n-1}^{(i-1)})$,

end

C.2.3 MCMC Convergence Diagnostics

Markov Chain Monte Carlo (MCMC) methods are convenient and flexible, but compared to simpler methods of statistical computation the following conditions must be met:

- Running the Markov chains long enough for convergence, only the samples that are drawn after this burn-in period are regarded as the representative draws from the posterior. Burn-in allows to correct for starting value bias.
- Having enough simulation draws for a suitably accurate inference for estimating the stationary distribution p .

The latter is called the problem of monitoring convergence of the sampler. The convergence diagnostics of Gelman and Rubin [62] [35] and of Raftery and Lewis [116] currently are the most popular amongst the statistical community. In this appendix let us focus on Gelman and Rubin approach. Convergence of Markov chain simulations can be monitored by measuring the diffusion and mixing of multiple independently-simulated chains. Typically the convergence of all the parameters are monitored separately. The approach supposes that M chains have been simulated in parallel, each with different starting points which are overdispersed with respect to the target distribution. Given any individual chain, and if approximate convergence has been reached, inferences about any parameter are assumed made by computing the sample mean and variance from the simulated draws. Let $x_m^{(i)}$ denote the i^{th} of the N samples in chain m . The approach consists, for each parameter, first to compute the variance of the samples from each chain (after discarding burn-in), to average these within-chain variances, and compare this to the variances of all the chains mixed together. Then compute the Potential Scale Reduction Factor (PSRF) \hat{R} defined as the square root of the mixture variance divided by the average within-chain variance. In particular, the average of the m within-chain variances, W , can be expressed as:

$$W = \frac{1}{M-1} \sum_{m=1}^M \left(\frac{1}{N-1} \sum_{i=1}^N \left(x_m^{(i)} - \bar{x}_m \right)^2 \right), \text{ where } \bar{x}_m = \frac{1}{N-1} \sum_{i=1}^N x_m^{(i)}. \quad (\text{C.28})$$

The empirical variance from all chains combined B is given by:

$$B = \frac{N}{M-1} \sum_{m=1}^M (\bar{x}_m - \bar{x})^2, \text{ where } \bar{x} = \frac{1}{M-1} \sum_{m=1}^M \bar{x}_m. \quad (\text{C.29})$$

Note that the ANalysis Of VAriance (ANOVA) assumption of pooled within-chain variances holds, under convergence, the M within-chain variances will be equal. Thus, having observed these estimates, let us estimate the variance of the stationary distribution by a weighted average of B and W , as follows:

$$\hat{\sigma}^2 = \left(1 - \frac{1}{N} \right) W + \frac{1}{N} B. \quad (\text{C.30})$$

Note that $\hat{\sigma}^2$ would be an unbiased estimate of the true variance, if the starting points of the chains were drawn from the target distribution, but overestimates the true variance if the starting points are patchy across the state space. Accounting for the sampling variability of the estimate of the sample mean yields a pooled posterior variance estimate

$$\hat{\text{var}}(x) = \hat{\sigma}^2 + \frac{B}{M N}. \quad (\text{C.31})$$

The comparison of pooled and within-chain inferences is expressed as the square root of the variance ratio

$$\hat{R} = \sqrt{\frac{\widehat{\text{var}}(x)}{W}} \quad (\text{C.32})$$

which is called PSRF, and can be interpreted as a convergence diagnostic as follows. If \hat{R} is greater than 1, this implies that the chains have not fully mixed and that further simulation might increase the precision of inferences. Alternatively, if the PSRF is close to 1, we can conclude that each of the M sets of N simulated observations is close to the target distribution. In practice we typically go until \hat{R} is less than 1.1 for all parameters; however this rule can declare convergence prematurely.

C.2.4 Reversible Jump MCMC

Given a family of M models $\{\mathcal{M}_m; m = 1, \dots, N\}$, we focus on constructing ergodic Markov chains admitting $p(m, x_m)$ as the invariant distribution. Up to this section, we have been comparing densities in the acceptance ratio. However, if we are carrying out model selection, then comparing the densities of objects in different dimensions has no meaning. Instead, we have to compare distributions $P(dx) = \text{Pr}(x \in dx)$ under a common measure of volume. The acceptance ratio will now include the ratio of the densities and the ratio of the measures (Radon Nikodym derivative). The latter gives rise to a Jacobian term. To compare densities point-wise, we need, therefore, to map the two models to a common dimension. The parameters $x_m \in \mathcal{X}_m$ are model dependent. RJMCMC method allows the sampler to jump between the different subspaces. To ensure a common measure, it requires the extension of each pair of communicating spaces, \mathcal{X}_m and \mathcal{X}_n , to $\bar{\mathcal{X}}_{m,n} \triangleq \mathcal{X}_m \times \mathcal{U}_{m,n}$ and $\bar{\mathcal{X}}_{n,m} \triangleq \mathcal{X}_n \times \mathcal{U}_{n,m}$. It also requires the definition of a deterministic, differentiable, invertible dimension matching function $f_{n \rightarrow m}$ between $\bar{\mathcal{X}}_{m,n}$ and $\bar{\mathcal{X}}_{n,m}$,

$$(x_m, u_{m,n}) = f_{n \rightarrow m}(x_n, u_{n,m}) = (f_{n \rightarrow m}^x(x_n, u_{n,m}), f_{n \rightarrow m}^u(x_n, u_{n,m})). \quad (\text{C.33})$$

We define $f_{m \rightarrow n}$ such that $f_{m \rightarrow n}(f_{n \rightarrow m}(x_n, u_{n,m})) = (x_n, u_{n,m})$. The choice of the extended spaces, deterministic transformation $f_{m \rightarrow n}$ and proposal distributions for $q_{n \rightarrow m}(\cdot | n, x_n)$ and $q_{m \rightarrow n}(\cdot | m, x_m)$ is problem dependent and needs to be addressed on a case by case basis. If the current state of the chain is (n, x_n) , we move to (m, x_m) by generating $u_{n,m} \sim q_{n \rightarrow m}(\cdot | n, x_n)$, ensuring that we have reversibility $(x_m, u_{m,n}) = f_{n \rightarrow m}(x_n, u_{n,m})$, and accepting the move according to the probability ratio

$$\alpha_{n,m} = \min \left\{ 1, \frac{p(m, x'_m)}{p(n, x_n)} \times \frac{q(n|m)}{q(m|n)} \times \frac{q_{m \rightarrow n}(u_{m,n}|m, x'_m)}{q_{n \rightarrow m}(u_{n,m}|n, x_n)} \times \mathcal{J}_{f_{n \rightarrow m}} \right\}, \quad (\text{C.34})$$

where $x'_m = f_{n \rightarrow m}^x(x_n, u_{n,m})$ and $\mathcal{J}_{f_{n \rightarrow m}}$ is the Jacobian of the transformation $f_{n \rightarrow m}$

$$\mathcal{J}_{f_{n \rightarrow m}} = \left| \det \frac{\partial f_{n \rightarrow m}^x(x_n, u_{n,m})}{\partial (x_n, u_{n,m})} \right|. \quad (\text{C.35})$$

Reversible jump is a mixture of MCMC kernels (moves).

Bibliography

- [1] C. Agate and K. Sullivan. Road-Constrained Target Tracking and Identification Using a Particle Filter. *Signal and Data Processing of Small Targets, San Diego, CA*, pages 532–543, 2003.
- [2] B.D.O. Anderson and J.B. Moore. *Optimal Filtering*. Prentice-Hall, Englewood Cliffs, NJ, 1979.
- [3] C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov Chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B*, 72(3):269–342, 2010.
- [4] C. Andrieu and E. Moulines. On the ergodicity properties of some adaptive MCMC algorithms. *Annals of Applied Probability*, 16(3):1462–1505, 2006.
- [5] C. Andrieu and G. O. Roberts. The Pseudo-marginal Approach for Efficient Monte Carlo Computations. *Annals of Statistics*, 37(2):697–725, 2009.
- [6] E.H. Aoki, A. Bagchi, P.K. Mandal, and Y. Boers. A Theoretical Analysis of Bayes-optimal Multi-target Tracking and Labelling. *Memorandum 1953, Department of Applied Mathematics, University of Twente, Enschede*, 2011.
- [7] E.H. Aoki, Y. Boers, L. Svensson, P.K. Mandal, and A. Bagchi. A Bayesian Analysis of the Mixed Labelling Phenomenon in Two-target Tracking. *Memorandum 1986, Department of Applied Mathematics, University of Twente, Enschede*, 2012.
- [8] M. Arulampalam, S. Maskell, N. Gordon, and G. Clapp. A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.
- [9] S. Arulampalam, N. Gordon, and R. Ristic. *Beyond the Kalman Filter - Particle Filters for Tracking Applications*. Boston - London: Artech House, 2004.
- [10] A. Bagchi. *Optimal Control of Stochastic Systems*. Prentice Hall International, Englewood Cliffs, 1993.
- [11] A. Bain and D. Crisan. *Fundamentals of Stochastic Filtering, Stochastic Modelling and Applied Probability*, volume 60. Springer, New York, NY, USA, 2009.
- [12] Y. Bar-Shalom and T.E. Fortmann. *Tracking and Data Association*, volume 179 of *Mathematics in Science and Engineering*. Academic Press Professional, Inc., 1987.
- [13] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*. John Wiley and Sons, 2001.
- [14] Y. Bar-Shalom and X.R. Li. *Multitarget-Multisensor Tracking: Principles, Techniques*. Storrs, YBS Publishing, 1995.

- [15] Y. Barniv. *Dynamic Programming Algorithm for Detecting Dim Moving Targets*, chapter 4. Artech House, multitarget-multisensor tracking: advanced applications edition, 1990.
- [16] L.E. Baum, T.P. Petrie, G. Soules, and N. Weiss. A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains. *Annals of the Institute of Statistical Mathematics*, 41:164–171, 1970.
- [17] D. Bertsekas. The Auction Algorithm for Assignment and other Network Flow Problems: A Tutorial. *Interfaces*, 20:133–149, 1990.
- [18] D.P. Bertsekas. *Linear Network Optimization: Algorithms and Codes*. The MIT Press, 1991.
- [19] H. Bhaskar, L. Mihaylova, and S.R. Maskell. Population based Particle Filtering. *IET Seminar on Target Tracking and Data Fusion: Algorithms and Applications*, pages 31–38, 2008.
- [20] S. Blackman. Multiple Hypothesis Tracking for Multiple Target Tracking. *IEEE Aerospace & Electronic Systems Magazine*, 19(1):5–18, January 2004.
- [21] S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. New York: Artech House, 1999.
- [22] H.A.P. Blom, E.A. Bloem, Y. Boers, and J. N. Driessen. Tracking Closely Spaced Targets: Bayes outperformed by an approximation? *Proc. of the 11th Int. Conf. on Information Fusion*, pages 1–8, Cologne, Germany, July 2008.
- [23] R. Bock and M. Aitkin. Marginal Maximum Likelihood Estimation of Item Parameters: Application of an EM Algorithm. *Psychometrika*, 46(4):443–459, 1981.
- [24] M. Bocquel, H. Driessen, and A. Bagchi. Multitarget Tracking with Interacting Population-based MCMC-PF. *Proc. of the 15th Int. Conf. on Information Fusion*, pages 74–81, Singapore, July 2012.
- [25] M. Bocquel, H. Driessen, and A. Bagchi. Multitarget Tracking with IP Reversible Jump MCMC-PF. *Proc. of the 16th Int. Conf. on Information Fusion*, Istanbul, July 2013.
- [26] M. Bocquel, F. Papi, M. Podt, and H. Driessen. Multitarget Tracking with Multiscan Knowledge Exploitation using Sequential MCMC sampling. *IEEE Journal of Selected Topics In Signal Processing*, 7(3):532–542, June 2013.
- [27] Y. Boers and H. Driessen. Particle Filter Track-Before-Detect Application Using Hard Inequality Constraints. *IEEE Trans. on Aerospace and Electronic Systems*, 41(4):1483–1489, 2005.
- [28] Y. Boers and J. N. Driessen. Multitarget Particle Filter Track-Before-Detect Application. *IEE Proc. on Radar, Sonar and Navigation*, 151:351–357, 2004.
- [29] Y. Boers and J.N. Driessen. Particle Filter based Detection for Tracking. *Proc. of the 2001 American Control Conference*, pages 4393–4397, 2001.
- [30] Y. Boers and J.N. Driessen. Interacting Multiple Model Particle Filter. *IEE Proceedings - Radar, Sonar and Navigation*, 150(5):344–349, 2003.

- [31] Y. Boers, F. Ehlers, W. Koch, T. Luginbuhl, L.D. Stone, and R.L. Streit. Track Before Detect Algorithms. *Eurasip Journal on applied Signal Processing*, 2008.
- [32] Y. Boers, E. Sviestins, and H. Driessen. Mixed Labelling in Multitarget Particle Filtering. *IEEE Trans. on Aerospace and Electronic Systems*, 46(2):792–802, 2010.
- [33] F. Bourgeois and J. Lassalle. An extension of the Munkres Algorithm for the Assignment Problem to Rectangular Matrices. *Commun. ACM*, 14(12):802–804, 1971.
- [34] D. Brigo, B. Hanzon, and F. Le Gland. Approximate Nonlinear Filtering by Projection on Exponential Manifolds of Densities. *Bernoulli*, 5(3):495–534, 1999.
- [35] S. P. Brooks and A. Gelman. General Methods for Monitoring Convergence of Iterative Simulations. *Journal of Computational and Graphical Statistics*, 7(4):434–455, 1998.
- [36] J.V. Candy. *Bayesian Signal Processing: Classical, Modern and Particle Filtering Methods*. Wiley-Interscience, New York, NY, USA, 2009.
- [37] O. Cappé, S. Godsill, and E. Moulines. An Overview of Existing Methods and Recent Advances in Sequential Monte Carlo. *Proceedings of the IEEE*, 95(5):899–924, May 2007.
- [38] O. Cappé, E. Moulines, and T. Ryden. *Inference in Hidden Markov Models*. Springer Series in Statistics, 2005.
- [39] B.D. Carlson, E.D. Evans, and S.L. Wilson. Search Radar Detection and Track with the Hough Transform. Part I: System Concept. *IEEE Trans. Aerospace & Electronic Systems*, 30:102–108, 1994.
- [40] G. Casella and E. I. George. Explaining the Gibbs Sampler. *The American Statistician*, 46(3):167–174, August 1992.
- [41] M. Chaleyat-Maurel and D. Michel. Des résultats de non existence de filtre de dimension finie. *Stochastics*, 13(1-2):83–102, 1984.
- [42] S. Challa and N. Bergman. Target Tracking Incorporating Flight Envelope Information. *Proc. of the 3th Int. Conf. on Information Fusion*, 2:THC2/22–THC2/27, Paris, France, July 2000.
- [43] S. Chib and E. Greenberg. Understanding the Metropolis-Hastings Algorithm. *The American Statistician*, 49:327–335, 1995.
- [44] D.E. Clark and J. Bell. Multi-Target State Estimation and Track Continuity for the Particle PHD Filter. *IEEE Trans. on Aerospace & Electronic Systems*, 43(4):1441–1453, 2007.
- [45] D.E. Clark and J. Houssineau. Faà di Bruno’s formula for variational calculus. *Manuscripts submitted to ESAIM Control, Optimization and the Calculus of Variations*, 2013.
- [46] D.J. Daley and D. Vere-Jones. *An Introduction to the Theory of Point Processes*. Springer-Verlag, 1988.

- [47] R. Danchick and G.E. Newnam. Reformulating Reid's MHT method with Generalised Murty K-best Ranked Linear Assignment Algorithm. *IEE Proceedings on Radar, Sonar and Navigation*, 153(1):13–22, February 2006.
- [48] S. Davies, M. Rutten, and B. Cheung. A Comparison of Detection Performance for Several Track-before-Detect Algorithms. *EURASIP Journal on Advances in Signal Processing*, 2008:Article ID 428036, 2008.
- [49] N. de Freitas, M. Niranjan, A.H. Gee, and A. Doucet. Sequential Monte Carlo Methods to Train Neural Network Models. *Neural Computation*, 12(4):955–993, 2000.
- [50] P. Del Moral. *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*. Springer, New York, 2004.
- [51] T.M. Deserno, editor. *Biomedical Image Processing*. Springer-Verlag, New York, NY, USA, 2011.
- [52] R. Douc, E. Moulines, and O. Cappé. Comparison of Resampling Schemes for Particle Filtering. In *4th International Symposium on Image and Signal Processing and Analysis (ISPA)*, pages 64–69, 2005.
- [53] A. Doucet. On sequential simulation-based methods for bayesian filtering. Signal Processing Group CUED/F-INFENG/TR.310, Cambridge University Engineering Department, 1998.
- [54] A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, Berlin, 2001.
- [55] A. Doucet, N. de Freitas, and K. Murphy. Rao-blackwellised particle filtering for dynamic bayesian networks. Technical report, Stuart Russel, 2000.
- [56] A. Doucet, S. Godsill, and C. Andrieu. On Sequential Monte Carlo Sampling Methods for Bayesian Filtering. *Statistics and Computing*, 10:197–208, 2000.
- [57] A. Doucet and A. M. Johansen. A Tutorial on Particle Filtering and Smoothing: Fifteen Years Later. In *Handbook of Nonlinear Filtering*, University Press, 2009.
- [58] H. Driessen and Y. Boers. A Multiple Model Multiple Hypothesis Filter for Markovian Switching Systems. *Automatica*, 41(4):709–716, 2005.
- [59] S. Duane, A.D. Kennedy, B.J. Pendleton, and D. Roweth. Hybrid Monte Carlo. *Physics Letters*, B(195):216–222, 1987.
- [60] M. Fallon and S. Godsill. Acoustic Source Localization and Tracking Using Track Before Detect. *IEEE Transactions on Audio, Speech and Language Processing*, 18(6):1228–1242, 2010.
- [61] A. E. Gelfand and A. F. M. Smith. Sampling-Based Approaches to Calculaing Marginal Densities. *Journal of the American Statistical Association*, 85:398–409, 1990.
- [62] A. Gelman and D. B. Rubin. Inference from Iterative Simulation Using Multiple Sequences (with discussion). *Statistical Science*, 7:457–511, 1992.

- [63] K. Gilholm and D. Salmond. A Spatial Distribution Model for Tracking Extended Objects. *IEE Proceedings on Radar, Sonar and Navigation*, 152(5), 2005.
- [64] W. R. Gilks and C. Berzuini. Following a moving target-Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society: Series B*, 63:127–146, 2001.
- [65] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Chapman & Hall, 1996.
- [66] M. Girolami and B. Calderhead. Riemann Manifold Langevin and Hamiltonian Monte Carlo Methods. *Journal of the Royal Statistical Society: Series B*, 73(2):123–214, 2011.
- [67] S.J. Godsill, A. Doucet, and M. West. Monte Carlo Smoothing for Nonlinear Time Series. *Journal of the American Statistical Association*, pages 438–449, 2004.
- [68] I. Goodman, R. Mahler, and H. Nguyen. *Mathematics of Data Fusion*. Kluwer Academic Publishers, 1997.
- [69] N. Gordon, S. Maskell, T. Kirubarajan, and M. Mallick. Littoral Tracking using Particle Filter. *Proc. of the 5th Int. Conf. on Information Fusion*, 2:935–942, Annapolis, USA, July 2002.
- [70] N. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to Nonlinear/Non-Gaussian Bayesian State Estimation. *IEE Proceedings on Radar and Signal Processing*, 14(2):107–113, 1993.
- [71] A.G. Gray. *Bringing Tractability to Generalized N-Body Problems in Statistical and Scientific Computing*. PhD thesis, Carnegie Mellon University, 2003.
- [72] P. J. Green. Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination. *Biometrika*, 82(4):711–732, December 1995.
- [73] F. Gustafsson and G. Hendeby. Some Relations Between Extended and Unscented Kalman Filters. *IEEE Transactions on Signal Processing*, 60(2):545–555, 2012.
- [74] A.C. Harvey. *Applications of the Kalman Filter in Econometrics*. Cambridge University Press, 1987.
- [75] W. K. Hastings. Monte Carlo Sampling Methods using Markov Chains and their Applications. *Biometrika*, 57:97–109, 1970.
- [76] D.M. Higdon. Auxiliary Variable Methods for Markov Chain Monte Carlo with Applications. *Journal of the American Statistical Association*, 93(442):585–595, June 1998.
- [77] J. Hoffman and R. Mahler. Multitarget Miss-distance and its Applications. *Proc. of the 5th Int. Conf. on Information Fusion*, 1:149–155, Annapolis, USA, July 2002.
- [78] J.H. Holland. *Adaptation in Natural and Artificial Systems: an Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. University of Michigan Press, 1975.

- [79] X.-L. Hu, T. Schon, and L. Ljung. A Basic Convergence Result for Particle Filtering. *IEEE Transactions on Signal Processing*, 56(4):1337–1348, April 2008.
- [80] R. Jonker and A. Volgenant. A Shortest Augmenting Path Algorithm for Dense and Sparse Linear Assignment Problems. *Computing*, 38(4):325–340, 1987.
- [81] S. Julier and J. Uhlmann. Unscented Filtering and Nonlinear Estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.
- [82] R. Kalman. New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME - Journal of Basic Engineering*, 82(Series D):35–45, 1960.
- [83] E. W. Kamen. Multiple Target Tracking based on Symmetric Measurement Equations. *IEEE Transaction on Automatic Control*, 37:371–374, 1992.
- [84] Z. Khan, T. Balch, and F. Dellaert. MCMC-Based Particle Filtering for Tracking a Variable Number of Interacting Targets. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(11):1805–1819, 2005.
- [85] G. Kitagawa. The Two-Filter Formula for Smoothing and an Implementation of the Gaussian-sum Smoother. *Annal Institute of Statistical Mathematics*, 46(4):605–623, 1994.
- [86] G. Kitagawa. Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models. *Journal of Computational and Graphical Statistics*, 5(1):1–25, March 1996.
- [87] M. Klaas, M. Briers, N. de Freitas, A. Doucet, S. Maskell, and D. Lang. Fast Particle Smoothing: If I Had a Million Particles. *Proc. of the 23rd International Conference on Machine Learning*, Pittsburg, PA, USA 2006.
- [88] H. Kuhn. The Hungarian Method for the Assignment Problem. *Naval Research Logistic Quarterly*, (2):83–97, 1955.
- [89] W. Leven and A. Lanterman. Unscented Kalman Filters for Multiple Target Tracking With Symmetric Measurement Equations. *IEEE Transactions on Automatic Control*, 54(2):370–375, February 2009.
- [90] J. Liu and R. Chen. Sequential Monte Carlo Methods for Dynamic Systems. *Journal of the Royal Statistical Society: Series B*, 93(443):1032–1044, 1998.
- [91] Vihola M. *Random Sets for Multitarget Tracking and Data Fusion*. PhD thesis, Licentiate Thesis, Tampere University of Technology, 2004.
- [92] R. Mahler. Multitarget Bayes Filtering via First-Order Multitarget Moments. *IEEE Trans. Aerospace & Electronic Systems*, 39(4):1152–1178, 2003.
- [93] R. Mahler. PHD Filters of Higher Order in Target Number. *IEEE Trans. Aerospace & Electronic Systems*, 43(3):1523–1543, July 2007.

- [94] R. Mahler and A. El-Fallah. The Random Set Approach to Nontraditional Measurements is Rigorously Bayesian. *Proc. of the SPIE, Signal Processing, Sensor Fusion, and Target Recognition XXI*, 8392, Baltimore, April 2012.
- [95] R.P.S. Mahler. *Statistical Multisource-Multitarget Information Fusion*. Artech House, 2007.
- [96] M. Majji, J. L. Junkins, and J. D. Turner. Jth Moment Extended Kalman Filtering for Estimation of Nonlinear Dynamical Systems. *Guidance Navigation and Control Conference*, 2008.
- [97] G. Mathéron. *Random sets and Integral Geometry*. Wiley series in Probability and Mathematical Statistics. Wiley, New York, 1975.
- [98] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan. Interacting multiple model methods in target tracking: a survey. *IEEE Trans. on Aerospace and Electronic Systems*, 35(1):103–123, 1998.
- [99] M. McDonald and B. Balaji. Track-before-Detect using Swerling 0,1, and 3 Target Models for Small Manoeuvring Maritime Targets. *EURASIP Journal on Advances in Signal Processing*, 2008:Article ID 326259, 2008.
- [100] E. Meijering, I. Smal, and Danuser G. Tracking in Molecular Bioimaging. *IEEE Signal Processing Magazine*, 23(3):46–53, June 2006.
- [101] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equations of State Calculations by Fast Computing Machines. *J. Chemical Physics*, 21:1087–1091, 1953.
- [102] B. Miasojedow, E. Moulines, and M. Vihola. An Adaptive parallel tempering algorithm. *Journal of Computational and Graphical Statistics*, March 2013.
- [103] R. Montenegro and P. Tetali. Mathematical Aspects of Mixing Times in Markov Chains. *Foundations and Trends in Theoretical Computer Science*, 1(3):237–354, 2006.
- [104] W.J. Morokoff and R.E. Caflisch. Quasi-Monte Carlo Integration. *Journal of Computational Physics*, 122:218–230, 1995.
- [105] E. Moulines. Adaptive and interacting mcmc algorithms, March 2010.
- [106] N. Nandakumaran, S. Sutharsan, R. Tharmarasa, T. Lang, M. McDonald, and T. Kirubarajan. Interacting multiple model forward filtering and backward smoothing for maneuvering target tracking. *Proc. of SPIE - Signal and Data Processing of Small Targets*, 7445:1–10, 2009.
- [107] F.E. Nathanson. *Radar Design Principles, Signal Processing and the Environment*. Mendham, NJ: Scitech Publishing, Inc., 1991.
- [108] R. Neal. *MCMC for Using Hamiltonian Dynamics*, chapter 5, pages 113–162. Chapman & Hall / CRC Press, handbook of markov chain monte carlo (eds s. brooks, a. gelman, g. jones, xl. meng) edition, 2011.

- [109] S. Oh, S. Russell, and S. Sastry. Markov chain monte carlo data association for multiple-target tracking. Communication UCB/ERL M05/19, Eng. College, University of California, Berkeley, USA, June, 2005.
- [110] A. B. Owen. *Monte Carlo, Quasi-Monte Carlo and Randomized Quasi-Monte Carlo*. H. Niederreiter and J. Spanier, Springer-Verlag, 2000.
- [111] F. Papi, M. Bocquel, M. Podt, and Y. Boers. Fixed-Lag Smoothing for Bayes Optimal Exploitation of External Knowledge. *Proc. of the 15th Int. Conf. on Information Fusion*, pages 463–470, Singapore, July 2012.
- [112] F. Papi, M. Podt, Y. Boers, G. Battistello, and M. Ulmke. Bayes Optimal Knowledge Exploitation for Hard-Constrained Target Tracking. *Proc. of the 9th IET Data Fusion & Target Tracking Conference*, London, UK 2012.
- [113] F. Papi, B.T. Vo, M. Bocquel, and B.N. Vo. Multi-target Track-Before-Detect using Labeled Random Finite Set.
- [114] S. Park, J. P. Hwang, E. Kim, and H. J. Kang. A new Evolutionary Particle Filter for the Prevention of Sample Impoverishment. *IEEE Transactions on Evolutionary Computation*, 13(4):801–809, August 2009.
- [115] M.K. Pitt and N. Shephard. Filtering via Simulation: Auxiliary Particle Filters. *Journal of the American Statistical Association*, 94(446):590–599, June 1999.
- [116] Adrian E. Raftery and Steven Lewis. How Many Iterations in the Gibbs Sampler? In *In Bayesian Statistics 4*, pages 763–773. Oxford University Press, 1992.
- [117] D.B. Reid. An algorithm for Tracking Multiple Targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, 1979.
- [118] B. Ristic, D. Clark, and Ba-Ngu Vo. Improved SMC Implementation of the PHD Filter. *Proc. of the 13th Int. Conf. on Information Fusion*, pages 1–8, Edinburgh, UK, July 2010.
- [119] B. Ristic, D. Clark, Ba-Ngu Vo, and Ba-Tuong Vo. Adaptive Target Birth Intensity for PHD and CPHD Filters. *IEEE Trans. on Aerospace and Electronic Systems*, 48(2):1656–1668, 2012.
- [120] B. Ristic, B.N. Vo, D. Clark, and B.T. Vo. A Metric for Performance Evaluation of Multi-Target Tracking Algorithms. *IEEE Trans. on Signal Processing*, 59(7):3452–3457, 2011.
- [121] C.P. Robert and G. Casella. *Monte Carlo Statistical Methods*. New York: Springer-Verlag, 1999.
- [122] G.O. Roberts and O. Stramer. Langevin diffusions and metropolis-hastings algorithms. *Methodology And Computing In Applied Probability*, 4(4):337–357, 2002.
- [123] J. Roecker. A class of near optimal JPDA algorithms. *IEEE Trans. Aerospace & Electronic Systems*, 30(2):504–510, April 1994.
- [124] D. J. Salmond and H. Birch. A Particle Filter for Track-Before-Detect. *Proc. of the 2001 American Control Conference*, pages 3755–3760, 2001.

- [125] D. Schuhmacher, B.T. Vo, and B.N. Vo. A Consistent Metric for Performance Evaluation of Multi-Object Filters. *IEEE Transactions on Signal Processing*, 56(8):3447–3457, 1998.
- [126] F. Septier, S.K. Pang, A. Carmi, and S.J. Godsill. On MCMC-Based Particle Methods for Bayesian filtering: Application to Multitarget Tracking. *3rd IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 360–363, 2009.
- [127] R. Séroul. *Programming for Mathematicians*, chapter 10.12 Newton-Girard Formulas, pages 278–279. Berlin: Springer-Verlag, handbook of markov chain monte carlo edition, 2000.
- [128] K. Shafique and M. Shah. A Non-Iterative Greedy Algorithm for Multiframe Point Correspondence. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(1):55–65, 2005.
- [129] H. Sidenbladh. Multi-Target Particle Filtering for the Probability Hypothesis Density. *Proc. of the 6th Int. Conf. on Information Fusion*, pages 800–806, Cairns, Australia, July 2003.
- [130] M.I. Skolnik. *Introduction to Radar Systems*. McGraw-Hill, Inc, Singapore, 2nd ed., 1981.
- [131] A. F. M. Smith and A. E. Gelfand. Bayesian Statistics without Tears: A Sampling-Resampling Perspective. *The American Statistician*, 46:84–88, 1992.
- [132] G.L. Smith, S.F. Schmidt, and L.A. McGee. Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle. Technical report, NASA, Tech. Rep. TR R-135, 1962.
- [133] K. Smith, S. Ba, D. Gatica-Perez, and J.M. Odobez. Tracking the Visual Focus of Attention for a Varying Number of Wandering People. *IEEE Transaction Pattern Anal. Mach. Intell.*, 30:1212–1229, 2008.
- [134] P. Swerling. Probability of Detection for Fluctuating Targets. *Information Theory, IRE Transactions on*, 6(2):269–308, 1960.
- [135] M. Tahk and J. Speyer. Target Tracking Problems Subject to Kinematic Constraints. *IEEE Transaction on Automatic Control*, 35(3):324–326, 1990.
- [136] J. Vermaak, S. Godsill, and P. Perez. Monte Carlo Filtering for Multi-Target Tracking and Data Association. *IEEE Trans. on Aerospace and Electronic Systems*, 41(1):309–332, January 2005.
- [137] B.T. Vo. *Random Finite Sets in Multi-Object Filtering*. PhD thesis, School of Electrical, Electronic and Computer Engineering, The University of Western Australia, 2008.
- [138] B.T. Vo and B.N. Vo. A Random Finite Set Conjugate Prior and Application to Multi-Target Tracking. *Proc. 7th Int. Conf. Intelligent Sensors, Sensor Networks, and Information Processing (ISSNIP 2011)*, Adelaide, Australia, December 2011.
- [139] B.T. Vo and B.N. Vo. Labeled Random Finite Sets and Multi-Object Conjugate Priors. *IEEE Transactions on Signal Processing*, 61(13):3460–3475, July 2013.
- [140] B.T. Vo, B.N. Vo, and A. Cantoni. The Cardinality Balanced Multi-Target Multi-Bernoulli Filter and its Implementations. *IEEE Transactions on Signal Processing*, 57(2):409–423, 2009.

- [141] B.T. Vo, B.N. Vo, N.T. Pham, and D. Sutter. Joint Detection and Estimation of Multiple Objects From Image Observations. *IEEE Trans. on Signal Processing*, 58(10):5129–5141, 2010.
- [142] J.L. Williams. *Information Theoretic Sensor Management*. PhD thesis, Massachusetts Institute of Technology, 2007.

Abstract

Over the last few decades multi-target tracking (MTT) has proved to be a challenging and attractive research topic. MTT applications span a wide variety of disciplines, including robotics, radar/sonar surveillance, computer vision and biomedical research. The primary focus of this dissertation is to develop an effective and efficient multi-target tracking algorithm dealing with an unknown and time-varying number of targets.

The emerging and promising Random Finite Set (RFS) framework provides a rigorous foundation for optimal Bayes multi-target tracking. In contrast to traditional approaches, the collection of individual targets is treated as a set-valued state. The intent of this dissertation is two-fold; first to assert that the RFS framework not only is a natural, elegant and rigorous foundation, but also leads to practical, efficient and reliable algorithms for Bayesian multi-target tracking, and second to provide several novel RFS based tracking algorithms suitable for the specific Track-Before-Detect (TBD) surveillance application.

The TBD approach proposes to base the tracking on the raw non-thresholded measurement data, thereby exploiting all available information to update the multi-target state estimate, in the hope of achieving more accurate and robust performance in comparison with traditional point measurement based approaches. A highly nonlinear non-Gaussian tracking problem arises in TBD applications.

In recent years particle filters (PFs) have become a tremendously popular tool to perform tracking for non-linear and/or non-Gaussian models. Nevertheless, when the state space is very high dimensional, the performance of PFs depends to a large extent on the choice of the so-called proposal distribution. In order to tackle more general and more complex probability distributions, Markov Chain Monte Carlo (MCMC) methods are utilized. In the particle filtering framework, MCMC can be used for drawing the samples from an invariant distribution, either in sampling step or resampling step.

The first key contribution in this dissertation is a rigorous derivation and practical implementation of a novel algorithm well suited to deal with multi-target tracking problems for a given cardinality. The proposed Interacting Population-based MCMC-PF algorithm makes use of several Metropolis-Hastings samplers running in parallel, which interact through genetic variation. The performance of the proposed algorithm is demonstrated in simulated scenarios, and shown to outperform conventional filtering approaches.

The second key contribution concerns the design and implementation of two novel algorithms to handle a varying number of targets. The first approach exploits Reversible Jumps. The second approach is built upon the concepts of labeled RFSs and multiple cardinality hypotheses. The performance of the proposed algorithms is also demonstrated in practical scenarios, and shown to significantly outperform conventional multi-target PF in terms of track accuracy and consistency.

The third key contribution is a derivation and practical implementation of a novel multi-Bernoulli approximation to the multi-target Bayes recursion suitable for our specific TBD application. The performance of the proposed algorithm is compared with the full multi-target PF implementation, and shown to offer an attractive computationally cheaper alternative in certain situations.

The final contribution seeks to exploit external information to increase the performance of the surveillance system. In multi-target scenarios, kinematic constraints from the interaction of targets with their environment or other targets can restrict target motion. Such motion constraint information is integrated by using a fixed-lag smoothing procedure, named Knowledge-Based Fixed-Lag Smoother (KB-Smoother). The proposed combination IP-MCMC-PF/KB-Smoother yields enhanced tracking.

Further research planned and possible extensions to the initial approach are discussed.

Samenvatting

In de afgelopen decennia is multi-target tracking (MTT) als uitdagend en aantrekkelijk onderzoeksonderwerp gebleken te zijn. MTT toepassingen bestrijken een breed scala aan disciplines, waaronder robotica, radar/sonar surveillance, computer vision en biomedisch onderzoek. De primaire focus van dit proefschrift is het ontwikkelen van een effectief en efficiënt multi-target tracking algoritme voor een onbekend en tijdsvariërend aantal doelen.

Het opkomende en veelbelovende Random Finite Set (RFS) raamwerk biedt een stevig basis voor optimale Bayesiaanse MTT. In tegenstelling tot traditionele benaderingen, wordt de verzameling van individuele doelen behandeld als een *set-valued state*. De doelstelling van dit proefschrift is tweeledig: ten eerste het aantonen dat het RFS raamwerk niet alleen een natuurlijke, elegante en stevige basis is, maar ook leidt tot praktische, efficiënte en betrouwbare algoritmen voor Bayesiaanse MTT, en ten tweede het verstrekken van een aantal innovatieve RFS gebaseerde tracking algoritmen geschikt voor een Track-Before-Detect (TBD) Radar surveillance toepassing.

De TBD aanpak stelt voor om de tracking te baseren op de ruwe ongedrempelde meetgegevens, hiermee zorgend voor de exploitatie van alle beschikbare informatie voor het bijwerken van de multi-target state, ten einde nauwkeuriger, gevoeliger en meer robuuste prestaties te verkrijgen t.o.v. traditionele puntmeting gebaseerde aanpakken. TBD toepassingen leiden tot moeilijke niet lineair niet-Gaussisch tracking problemen.

De laatste jaren zijn particle filters (PFs) een populaire tracking techniek geworden voor niet-lineaire en/of niet-Gaussische modellen. Echter, wanneer de state space zeer hoogdimensional is, hangen de prestaties van PFs in grote mate af van de keuze van de zogenaamde proposal distributie Markov Chain Monte Carlo (MCMC) methoden worden gehanteerd voor het tackelen van algemenere en meer complexe kansverdelingen. MCMC kan in het particle filtering raamwerk gebruikt worden voor het nemen van de samples uit een invariante distributie, hetzij in sampling stap of resampling stap.

De eerste belangrijke bijdrage van dit proefschrift is een stevige afleiding en praktische implementatie van een innovatief algoritme dat zeer geschikt is voor de omgang met multi-target-tracking problemen voor een gegeven kardinaliteit. Het voorgestelde Interacting Population-based MCMC-PF (IP-MCMC-PF) algoritme maakt gebruik van meerdere parallel uitgevoerde Metropolis-Hastings samplers, die interactie hebben volgens genetische variatie. De prestaties van het voorgestelde algoritme wordt aangetoond in gesimuleerde scenarios en toont overtreffing van conventionele filtering aanpakken.

De tweede belangrijke bijdrage betreft het ontwerp en de implementatie van twee innovatieve algoritmen voor de omgang met een variërend aantal targets. De eerste aanpak maakt gebruik van Reversible Jumps. De tweede is gebaseerd op de concepten van gelabelde RFSs en multiple kardinaliteit hypotheses. De prestatie van de voorgestelde algoritmen wordt tevens aangetoond in typische praktische scenarios, waarbij conventionele multi-target PFs aanzienlijk worden overtroffen op het gebied van track nauwkeurigheid en consistentie.

De derde belangrijke bijdrage is het ontwerp en de implementatie van een innovatieve multi-Bernoulli benadering van de multi-target Bayes recursie, geschikt voor een TBD toepassing. De prestatie van het voorgestelde algoritme wordt vergeleken met de volledige multi-target PF implementatie en blijkt in bepaalde situaties een aantrekkelijk computationeel goedkoper alternatief te bieden.

De laatste bijdrage streeft naar het gebruik van externe informatie voor de verbetering van de prestaties van het surveillance systeem. In multi-target scenarios kunnen kinematische beperkingen, tijdens de interactie van targets met hun omgeving of andere targets, de target beweging begrenzen. Zulke informatie over bewegingsbeperking is geïntegreerd door middel van de fixed-lag smoothing procedure genaamd Knowledge-Based Fixed-Lag Smoother (KB-Smoother). De voorgesteld combinatie IP-MCMC-PF/KB-Smoother brengt verbeterde tracking.

Toekomstig onderzoek en mogelijke uitbreidingen van de algoritmes ontwikkelt in dit proefschrift worden besproken.