Spawn Model Derivations for Multi-Object Orbit Determination within a Random Finite Set Framework

by

Daniel S. Bryant

B.S., University of Texas at Arlington, 2011

M.S., University of Colorado at Boulder, 2013

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Aerospace Engineering Sciences

2017

This thesis entitled:
Spawn Model Derivations for Multi-Object Orbit Determination within a Random Finite Set Framework
written by Daniel S. Bryant
has been approved for the Department of Aerospace Engineering Sciences

_____

Dr. Brandon A. Jones

_____

Dr. Penina Axelrad

Date _____

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Bryant, Daniel S. (Ph.D., Aerospace Engineering Sciences)

Spawn Model Derivations for Multi-Object Orbit Determination within a Random Finite Set Framework

Thesis directed by Dr. Brandon A. Jones

Fragmentation events and large scale small-sat deployments are a significant threat; because of the sudden creation of many new objects posing potential risks to existing satellites, and the fact that current surveillance systems require laborious human intervention to identify and catalog these new objects. In the tracking community, the spontaneous appearance of new objects is referred to as birth, whereas spawning refers to the appearance of new objects generated by previously existing ones, such as a fragmentation event or small-sat deployment. In this dissertation, two-well known random finite set (RFS) filters are extended via mathematical derivation, aimed at performing initial orbit determination (IOD) of objects generated by spawning events. A Zero-Inflated Poisson (ZIP) spawn model is presented and a predicted cardinality expression for general spawn model configuration, capable of implementation via Partial Bell Polynomials, is derived for the Cardinalized Probability Hypothesis Density (CPHD) filter using a measure-theoretic approach. Generalized Labeled Multi-Bernoulli (GLMB) filter developments achieve a closed-form solution to the multi-object Bayes recursion capable of jointly estimating a spawned object's state and ancestry. Linear simulations demonstrate fundamental filter developments; the ZIP spawn model is shown to outperform other conventional models with the CPHD filter and multiple generations of spawn object ancestry are accurately estimated with the GLMB filter. Finally, non-linear simulations specific to Space Situational Awareness (SSA) IOD demonstrate the filters' efficacy, which include: fragmentation event and small-sat deployment scenarios, homogeneous and heterogeneous radar network observations, and spawning events that occur in and out of sensor field of view. This research shows that on-line multi-object IOD in the presence of spawning is possible within the RFS paradigm, and establishes a foundation upon which further SSA improvements can be investigated.

Dedication

To my wife Kathelin, without whom this would have never been possible. Thank you.

To my siblings, Phillip, Katie, and Ethan, in hope that I have inspired you to reach for the stars. Madi, may you rest in peace.

To those who did not make it home. Semper fidelis.

Acknowledgements

I would first like to thank my thesis advisor, Dr. Brandon Jones. Thank you for patiently guiding me through this adventure. Thank you for inspiring me to undertake such challenging and rewarding work. I would also like to thank Dr. George Born and Dr. Penina Axelrad, who have both served as advisors during my graduate career. I am very grateful for Dr. Penina Axelrad, Dr. Jay McMahon, Dr. Nisar Ahmed, and Dr. Vanja Dukic for serving on my thesis committee. I am grateful to Dr. Daniel Clark from Heriot-Watt University, Dr. Emmanuel Delande from the University of Texas, Austin, Dr. Jérémie Houssineau at the National University of Singapore, and Dr. Steve Gehly at the Royal Melbourne Institute of Technology, for their invaluable contributions to this work in CPHD filter developments. I am especially grateful to Dr. Ba-Tuong Vo and Dr. Ba-Ngu Vo at Curtin University for their contributions to this work in GLMB filter developments.

Thanks to my wife, parents, family, and friends for their support, encouragment, and patience. Thanks to my fellow students at the Colorado Center for Astrodynamics Research (CCAR) for your friendship. Special thanks to Ryan Mcgranaghan and McArthur Jones Jr. for your friendship and inspiration. Finally, thanks to the CCAR the Department of Aerospace Engineering Sciences faculty and staff for their years of support, with special thanks to Steve Hart and Sarah Melssen.

This research was sponsored in part by the Science, Mathematics, and Research for Transformation (SMART) Scholarship Program.

Contents

Chapter

1 Introduction 1

1.1 Motivation 1

1.2 State of the Art 3

1.2.1 Spawn Event IOD 3

1.2.2 Advanced Tracking Algorithms 4

1.3 Overview 10

1.4 Summary of Contributions 13

2 Multi-Object Filtering and Tracking with Random Finite Sets 15

2.1 Random Finite Set Filtering 15

2.1.1 Random Finite Sets 15

2.1.2 Multi-Object Bayes Filter 17

2.1.3 First-Order Approximation Filters 17

2.1.4 Multi-Bernoulli Approximation Filters 21

2.2 Labeled Random Finite Set Tracking 22

2.2.1 Label Random Finite Sets 23

2.2.2 Generalized Labeled Multi-Bernoulli 24

2.2.3 GLMB Tracking Filter 24

2.3 Gaussian Mixture Implementations 29

2.3.1 GM-CPHD 29
2.3.2 GM-GLMB 32
2.4 Tractability 34
2.4.1 Gaussian Mixture Components 34
2.4.2 GLMB Hypothesis Components 34
2.5 Metrics 36
2.6 Orbit Problem Overview 37
2.7 Uncertainty Propagation 39
2.8 Detection Probability Modeling 40
2.9 Breakup Modeling 41

3 Measure-Theoretic Background 44
3.1 Point processes 44
3.2 Probability generating functionals 47
3.3 Functional differentiation 48
3.4 Probability generating functionals and differentiation 50
3.5 Point process models 51
3.5.1 Bernoulli process 51
3.5.2 Poisson process 52
3.5.3 Zero-inflated Poisson process 52
3.5.4 I.i.d. process 52
3.6 CPHD Filter prediction derivation via p.g.fl. differentiation 53
3.6.1 Cardinality Prediction 53
3.6.2 Intensity Prediction 55

4 The CPHD Filter with Object-Spawning 56
4.1 The CPHD filter with spawning 56
4.1.1 Multi-Object Bayes Recursion 56

4.1.2 Prediction step . . . 57
4.2 Simulation . . . 60
4.2.1 The GM-CPHD filter with spawning . . . 60
4.2.2 Scenario and filter setup . . . 63
4.2.3 Simulation results . . . 66

5 Spawned Object Orbit Determination Simulations: GM-CPHD 71
5.1 Scenario Descriptions . . . 72
5.1.1 Dynamics . . . 73
5.1.2 Measurements . . . 75
5.1.3 Filter Configuration . . . 76
5.2 Results . . . 78
5.2.1 QB50 . . . 78
5.2.2 Rocket Body Explosion . . . 79

6 The GLMB Tracking Filter with Object-Spawning 82
6.1 Multi-object Labeled Spawning Model . . . 82
6.2 Multi-object prediction with spawning . . . 85
6.3 Multi-Object Update with Spawning . . . 86
6.4 Efficient Implementation . . . 90
6.5 Simulation . . . 93

7 Spawned Object Orbit Determination Simulations: GM-GLMB 103
7.1 Scenario Description . . . 103
7.1.1 Measurements . . . 108
7.1.2 Dynamics . . . 111
7.1.3 Filter Configuration . . . 112
7.2 Results . . . 113

7.2.1 PSLV-deploy-1 . . . 113
7.2.2 PSLV-deploy-2 . . . 116
7.2.3 Discussion . . . 120

8 Conclusions and Future Work 122
8.1 Summary of Results . . . 122
8.2 Contributions . . . 123
8.3 Future Work . . . 125

Bibliography 128

Appendix

A Derivations: CPHD with Spawning 135
A.1 Proof of theorem 1 . . . 135
A.1.1 Predicted probability generating functional (p.g.fl.) . . . 135
A.1.2 Predicted intensity . . . 136
A.1.3 Predicted cardinality . . . 137
A.2 Proof of corollary 1 . . . 140
A.2.1 Predicted intensity . . . 140
A.2.2 Predicted cardinality . . . 141
A.3 Comparison of expressions [59, (A.15)] and Eq. (4.4) . . . 142

B CPHD with Spawning Algorithm 144

C GLMB with Spawning Algorithm 145

Tables

Table

1.1 RFS Target Dynamics Terminology 7

2.1 Notation Definitions [89] 18

2.2 Keplerian Elements Description 38

4.1 Spawn model parameters. 66

5.1 RSO Initial Conditions 73

5.2 Propagation for Observation Simulation and Filter Dynamics 74

5.3 GM-CPHD filter configurations 77

6.1 Label Ground Truth at Time $k = 100$ 101

7.1 PSLV-C37 Initial Conditions 104

7.2 Phased Array Radar Locations and Physical Limits [88] 109

7.3 Phased Array Radar Error and Resolution Characteristics 110

7.4 Propagation for Observation Simulation and Filter Dynamics 112

7.5 GLMB filter configurations 113

Figures

Figure

1.1 Number of rideshares launched per year, 1960-2013 (Image credit: [83]). 6

1.2 Over 7500 Gaussian mixture (GM) components, each with a standard deviation of 100 km, required to fill a radar surveillance region [49]. 7

2.1 "An example of label assignments. The two tracks born at time 1 are given labels (1,1) and (1,2), while the only track born at time 4 is given label (4,1). Notice also the difference between multi-target states and target tracks." [94] 25

2.2 Caption for LOF 38

2.3 NSBM flow chart. Adapted from [54]. 42

4.1 Object trajectories. A circle "○" indicates where a trajectory begins, and a square "□" indicates where a trajectory ends. The large square indicates the limits of the sensor's field of view (FOV) and the large dashed circle represents the 90% confidence region of the Gaussian component of the spontaneous birth model. 64

4.2 Collected measurements (gray crosses) and object positions (black lines). 64

4.3 Maximum A Posteriori (MAP) estimate of the number of objects (averaged on 500 runs). 67

4.4 Optimal Sub-Pattern Assignment (OSPA) distance (averaged on 500 runs). 68

4.5 Hellinger distances (averaged on 500 runs). 69

5.1 Histogram of RB fragment Δv's generated with the NASA breakup model. 75

5.2 Each ■ indicates the location of a sensor within a homogeneous network. 76

5.3 QB50 cardinality estimates (top) and posterior number of GM components (bottom). 79

5.4 QB50 OSPA position (top) and velocity (bottom) errors. 79

5.5 RB cardinality estimates (top) and posterior number of GM components (bottom). . 80

5.6 RB OSPA position (top) and velocity (bottom) errors. 81

6.1 An example of label assignment for birth and spawn tracks. Two tracks are born at time 1 and are assigned labels $(1, 1)$ and $(1, 2)$. At time 5, a track is spawned from track $(1, 2)$ and is assigned label $(1, 2, 5, 1)$. 83

6.2 Object trajectories in the xy plane. A circle "○" indicates where an object is born, a square "□" indicates where a spawned object may be detected, and a triangle "△" indicates where an object dies. 94

6.3 Cardinality statistics for GLMB filter (100 Monte Carlo trials). 96

6.4 Cardinality statistics for CPHD filter (100 Monte Carlo trials). 96

6.5 OSPA distance for GLMB and CPHD filters (100 Monte Carlo trials). 96

6.6 OSPA components for GLMB and CPHD filters (100 Monte Carlo trials). 97

6.7 Birth region ancestry truth and estimates. Each region's true ancestry tree is at the far left. A red circle at the top indicates birth, while the first and second generation spawn times are denoted by a green square and yellow diamond, respectively. The birth track's time of death is marked by a pink diamond. Markers within the gridded area indicate GLMB filter estimated birth, death, and spawn times for 100 Monte Carlo runs. 98

6.8 Birth region ancestry truth and estimates. Each region's true ancestry tree is at the far left. A red circle at the top indicates birth, while the first and second generation spawn times are denoted by a green square and yellow diamond, respectively. The birth track's time of death is marked by a pink diamond. Markers within the gridded area indicate GLMB filter estimated birth, death, and spawn times for 100 Monte Carlo runs. 99

6.9 Birth region ancestry truth and estimates. Each region's true ancestry tree is at the far left. A red circle at the top indicates birth, while the first and second generation spawn times are denoted by a green square and yellow diamond, respectively. The birth track's time of death is marked by a pink diamond. Markers within the gridded area indicate GLMB filter estimated birth, death, and spawn times for 100 Monte Carlo runs. 100

7.1 PSLV-deploy-1 model depicted in the NTW frame. (A) View from aft of LV, looking forward (T axis points into the page). (B) Side view (N-axis points into the page). (C) Perspective view. (D) W-axis points out of page. 105

7.2 PSLV-deploy-2 model depicted in the NTW frame. (A) View from aft of LV, looking forward (T axis points into the page). (B) Side view (N-axis points into the page). (C) Perspective view. (D) W-axis points out of page. 107

7.3 Sensor locations and first 4 launch vehicle passes. 109

7.4 Phased array radar field of view volume. 111

7.5 [PSLV-deploy-1] Cardinality statistics vs. time (single run). 113

7.6 [PSLV-deploy-1] OSPA vs. time (single run). 114

7.7 [PSLV-deploy-1] OSPA, localization, and cardinality errors vs. filter step (single run). 115

7.8 [PSLV-deploy-1] Localization errors plotted on the left vertical axis with the time intervals between filter steps plotted on the right vertical axis. Both are plotted over filter step (single run). 115

7.9 [PSLV-deploy-1] Localization errors plotted on the left vertical axis with the number of missed detections plotted on the right vertical axis. Both are plotted over filter step (single run). 116

7.10 [PSLV-deploy-2] Cardinality statistics vs. time (single run). 117

7.11 [PSLV-deploy-2] OSPA vs. time (single run). 117

7.12 [PSLV-deploy-2] OSPA, localization, and cardinality errors vs. filter step (single run). 118

7.13 [PSLV-deploy-2] Localization errors plotted on the left vertical axis with the time intervals between filter steps plotted on the right vertical axis. Both are plotted over filter step (single run). 118

7.14 [PSLV-deploy-2] Localization errors plotted on the left vertical axis with the number of missed detections plotted on the right vertical axis. Both are plotted over filter step (single run). 119

7.15 [PSLV-deploy-2] Localization errors plotted on the left vertical axis with Range error standard deviations plotted on the right vertical axis. Both are plotted over filter step (single run). 120

Chapter 1

Introduction

1.1 Motivation

Sputnik 1, launched in 1957, was the first artificial satellite to orbit Earth. By today's standards, maintaining a space catalog for one object seems a trivial task as now, nearly sixty years later, tens of thousands of Earth orbiting artificial satellites, or resident space objects (RSOs), are tracked and cataloged by the Joint Space Operations Center (JSpOC)[1] in support of its SSA missions [2]. In general, Space Sitational Awareness (SSA) refers to the knowledge of the current and predicted states of the RSO population, however, depending on its specific application, definitions of SSA can vary. For instance, the end result of SSA for some operators is RSO collision avoidance[3], often referred to as conjunction assessment (CA), whereas [22] outlines applications of SSA that include CA, but also more militaristic applications such as determining a satellite's type and the purpose of its mission. As it pertains to the work presented in this paper, any reference to SSA hereafter will simply refer to the tracking of the RSO population and maintaining the catalog thereof.

All cataloged RSOs can be referred to as persistent objects, i.e., a priori information is

[1] JSpOC is a center within the Joint Functional Component Command for Space (JFCC Space), which is a component of the United States Strategic Command (USSTRATCOM).

[2] stratcom.mil. [online] Available at: https://www.stratcom.mil/factsheets/11/Space Control and Space Surveillance [Accessed 21 Jan. 2014]

[3] spacefoundation.org. [online] Available at: http://www.spacefoundation.org/programs/public-policy-and-government-affairs/introduction-space-activities/space-situational [13 Jun. 2015]

available for the propagation of their states, hence, initial orbit determination (IOD) is not required. However, new RSOs regularly appear due to launches and fragmentation events that do require some form of IOD or track initiation. Fragmentation events, e.g., when an RSO explodes or RSOs collide, are particularly distressing as they have the potential of producing several hundred new trackable objects [47], or more.

Another significant contributor to the RSO population is the small-sat industry. Small-sat refers to a category of satellites that have gained popularity over the last several years. They are typically small enough to fit on your desk and are categorized by a variety of names, e.g., nanosat, picosat, and, of course, CubeSat. CubeSats were first developed in 2000 and introduced as a means of low-cost space experimentation [39]. According to [84], more than seventy CubeSats were flown in the eight year span after 2003, when the first CubeSats were launched. Since then, seemingly every university has a space program, and launch vehicle (LV) operators have honed their "rideshare" [84] capabilities. For example, note that one-hundred rideshares were performed in 2013, as illustrated in the bar chart presented in Fig. 1.1, [83]. Only four years later, that same number of small satellites were deployed from a single LV[4,5].

The majority of small-sats are deployed with relatively low altitudes, which promotes their inevitable re-entry into Earth's atmosphere due to the effects of air drag. Still, prior to re-entry, they pose a threat to space operations like any other RSO; a threat exacerbated by the level of uncertainty associated with determining their orbits. Most small-sat programs use Two-Line Element (TLE) sets release by JSpOC for orbit determination. Unfortunately, it is increasingly the case where large numbers of small-sats are deployed from a launch vehicle and TLE sets get confused [33].

[4] "India launches more than 100 satellites into orbit." *USA Today*, February 15, 2017. Web. June 8, 2017.

[5] *PSLV-C37 Brochure-ISRO.* Web. June 8, 2017.

1.2 State of the Art

1.2.1 Spawn Event IOD

In terms of tracking, objects launched into orbit are referred to as birth objects, whereas those originating from small-sat deployments or fragmentation events are referred to as spawn objects. For example, a LV is born and goes on to spawn small-sats. Hereafter, such events are generally referred to as spawning events. Birth and spawning objects are categorized as newly appearing objects and, when they are observed by conventional space surveillance systems, they cause the generation of uncorrelated tracks (UCTs). UCTs are generally considered to be observations that automated processes fail to associate with known objects. Ultimately, if manual data association attempts fail, IOD is required to instantiate a new track for entry into the space catalog.

In [76], Schumacher detailed the then Naval Space Command's approach to UCT processing in the 1990's. Upon generation of UCTs, an analyst stepped in to make the appropriate data associations by drawing on their "expertise built up through long experience" and using a sophisticated set of tools which included the following:

- Satellite Identification (SID), which identifies position observations via an all-versus-all comparison of observations for a known set of objects selected by the analyst, and
- Search and Determine (SAD), which attempts correlation by considering all possible candidate element sets in the entire UCT database.

When a new association was made with tools like those listed above, analysts updated the space catalog using a tool called Manual Differential Correction (MANDC). Schumacher noted that with this framework, spawning events were potentially disruptive to catalog maintenance efforts due to the generation of a large number of UCTs.

Fast-forward to 2014, when Schumacher et al. presented a software package called Search and Determine Integrated Environment (SADIE) [77], the result of a joint effort by the Air Force Research Laboratory (AFRL) and the Naval Research Laboratory (NRL) under the High Perfor-

mance Computing Software Applications Institute for Space Situational Awareness (HSAI-SSA). SADIE was developed to address the expected increase in observation flow due to advancements in sensor technology, while meeting JSpOC mission requirements. Using "legacy" applications such as SID and SAD in conjunction with advance computing techniques, SADIE automatically adds to and maintains a space catalog; however, it is not a fully automated application as it was designed for analysts to use. Additionally, the fact remains that the tracking of persistent objects and the initialization of new objects via processing UCTs is performed disjointly.

1.2.2 Advanced Tracking Algorithms

Recent events such as the Fengyun-1C breakup [46] and the collision of Cosmos 2251 and Iridium 33 [1] prompted the SSA community to reevaluate their standards and practices. This included an assessment of the United States (US) Air Force's astrodynamics standards [22] which posited that advanced techniques such as Multiple Hypotheses Tracking (MHT) would produce significant improvement to SSA capabilities.

MHT was originally introduced in the late 1970's [72] as a data association approach to multi-target tracking (MTT) that considers multiple association hypotheses over several observation scans, i.e., it defers decisions about data associations, e.g., track correlations, by maintaining association hypotheses from multiple scans until sufficient information is collected to improve said decisions using probabilities computed via Bayes' rule . This specific approach is commonly referred to as hypothesis-oriented MHT whereas another commonly referenced approach is track-oriented MHT where, instead of maintaining hypotheses from scan to scan, tracks are maintained and used to form new hypotheses at each scan [11]. Since its initial presentation, MHT has been the focus of a substantial amount of research to include space surveillance applications [78].

Another approach to MTT is Joint Probabilistic Data Association (JPDA). Originally presented in 1980 [6], and the focus of a large volume of research since [5], JPDA is very similar to track-oriented MHT in that it maintains a set of tracks while forming new data association hypotheses at each scan. The main difference is that, unlike MHT where hypotheses can be maintained

for a number of scans, hypotheses are merged before moving on to the next scan. This similarity is likely why JPDA has been referred to as a special case of MHT [11].

A relatively new approach to MTT uses random finite sets (RFSs). Presented by Mahler as Finite Set Statistics (FISST) [61,63,64], RFSs are used to represent multi-object states. The multi-object Bayes filter is the theoretically optimal multi-object filter, however, for practical applications its implementation is intractable due to combinatorial complexity [63]. Hence, several RFS based approximations of the multi-object Bayes filter were developed to circumvent this intractability that include but are not limited to: the Probability Hypothesis Density (PHD) [89], Cardinalized PHD (CPHD) [96], Multi-target Multi-Bernoulli (MeMBer) [63], Cardinality Balanced MeMBer (CBMeMBer) [97], and Generalized Labeled Multi-Bernoulli (GLMB) [94, 95] filters.

Methods for multi-object tracking like MHT and JPDA are formulated using a bottom-up approach. That is, by way of data associations, they reduce the multi-object tracking problem into multiple single-object tracking problems [64]. RFS filtering, on the other hand, takes a top-down approach. Formulated on the principles of point process theory, RFS filtering uses the multi-object analog to Bayes' recursion to propagate multi-object densities through time. In this framework, the multi-object densities characterize the distributions of RFSs, which are the multi-object analog to the random variable representations in single-object filtering, e.g., in Kalman filtering.

Random finite set multi-object dynamics models can account for object survival, birth, spawning, and death: examples of each are given in Table 1.1. Both spontaneous birth and spawn models can account for the appearance of new objects in a given scene[6], however, for the presented work

[6] Spawn models can be considered *birth* models since they account for *new* objects, however, the term spontaneous is used to make a clearer distinction between objects that enter a scene on their own and objects generated by those already in the scene.

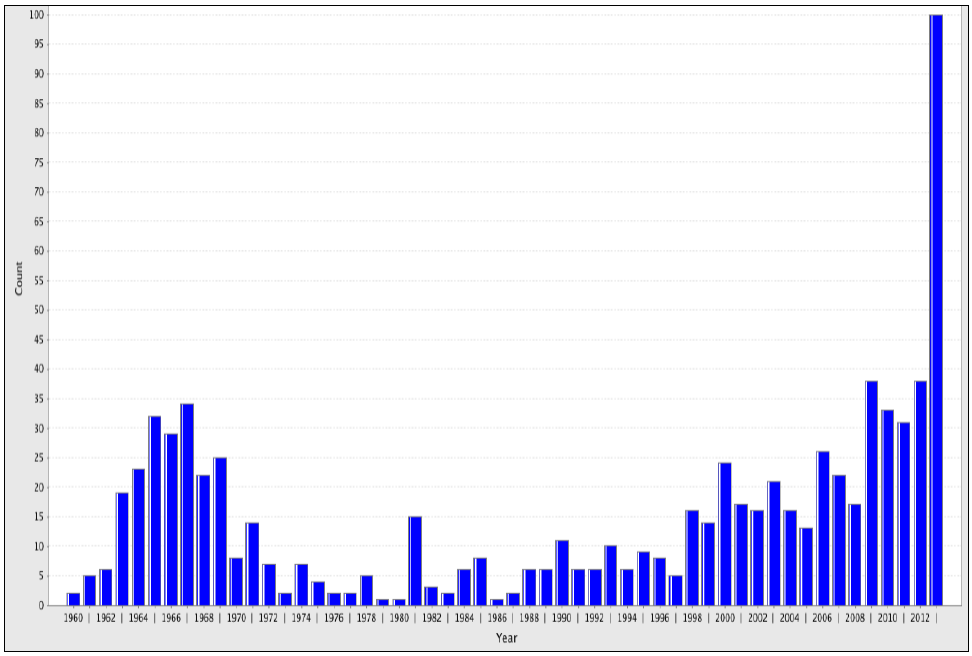


Figure 1.1: Number of rideshares launched per year, 1960-2013 (Image credit: [83]).

Table 1.1: RFS Target Dynamics Terminology

Nomenclature	Example
survival	continued existence of an object from one time to the next
spontaneous birth	first appearance of an object in a surveillance region, regardless of surviving object locations
spawning	first appearance of an object, statistically conditional on an existing object
death	departure or disappearance of an object from a surveillance region

spawn models are a more attractive approach. Not only do spawn models more accurately represent the generation of new objects for small-sat deployments and fragmentation events, they are potentially more efficient than birth models commonly used in the general tracking community, with respect to (w.r.t.) the same application. For example, some birth models populate a sensor's entire surveillance region with diffuse Gaussian distributions in an effort to detect object appearance [7]. For SSA applications, this would require an intractable number of GM components. Figure 1.2 from [49] illustrates this for a radar surveillance region large enough to encapsulate a single arc of observations for a low-Earth orbit (LEO) object. Assuming GM component means are equidis-

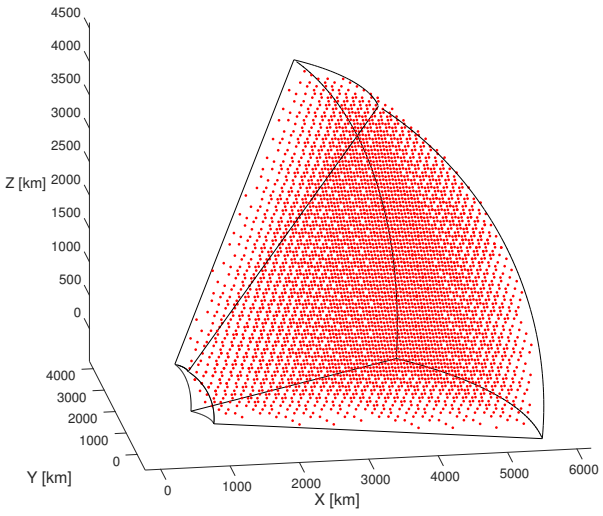


Figure 1.2: Over 7500 GM components, each with a standard deviation of 100 km, required to fill a radar surveillance region [49].

tantly separated by two standard deviations σ, where $\sigma = \sigma_{\mathrm{X}} = \sigma_{\mathrm{Y}} = \sigma_{\mathrm{Z}} = 100\,\mathrm{km}$, over 7500 GM

components are required to fill the region. Fortunately, more efficient SSA specific birth models are being developed [48, 50].

The PHD filter readily accounts for spawn objects, so in that regard, it makes sense to start with it for spawning event IOD. This filter was applied in [27] to the breakup of the Landsat 2 rocket body, though spontaneous birth models were used instead of spawn models. Notwithstanding, the PHD filter is a relatively poor choice for scenarios with a high variability in the number of objects (details provided in Chapter 2). The CPHD filter, whose formulation is designed to circumvent challenges that varying object number can present, is a better option for the given application. However, as the CPHD filter is originally derived, it does not account for the presence of spawned objects, nor does it account for object identity.

Though the CPHD filter's model for birth targets has the potential to address spawning targets [64], there may be cases where specific spawning models are more applicable. In the context of tracking RSO spawning events, without a spawn model, the best option may be the use of diffuse birth regions, however, as mentioned before the volume of space to be filled requires a potentially intractable number of birth regions [49]. To improve the CPHD filter's performance for space-object tracking, [50] presented a measurement-based birth model that leverages an astrodynamics approach to track initialization for Resident Space Objects (RSOs). While such an approach may be effective for tracking spawned RSOs, a multi-object filter that correctly models the birth process for a given object is expected to provide better accuracy and faster confirmation of new objects.

Note the distinction between filtering and tracking. In this document, the term *filtering* is used to denote the estimation of the number of objects in a surveillance region and their states, but it does not imply knowledge of a given object's identity. The term *tracking* implies that a given object's identity is known or estimated, along with its state. Therefore, to be effective, a multi-object tracker must provide object number and individual object state estimates, along with performing data associations to distinguish objects. The CPHD filter's capabilities could be extended to tracking as was done similarly in [69]. However, such a labeling scheme is more an algorithmic extension than an integral feature resulting from principled derivations; a cause

for concern since it may be difficult to characterize the behavior of track labels. An RFS filter that rigorously accounts for track labels has recently been gaining traction within the tracking community. The GLMB tracking filter [94, 95] accounts for track labeling mathematically via labeled random finite sets (LRFSs).

Information on lineage or ancestry is an important aspect of tracking multiple spawn objects. For example, in (biological) cell tracking, information on a cell's lineage is important to the analyis of cell behavior [4, 9, 57, 66]. For SSA, information on the ancestry of RSOs is important for cataloging and analysis, e.g., that of fragmentation events [13, 27, 76]; moreover, country of origin and launch site, information required to add a space object to the United States Strategic Command (USSTRATCOM) catalog[7], can be derived from ancestry information. Even with spawn models, the PHD/CPHD filters [14,60,61] only provide estimates of spawned objects' states, but no information on their ancestries. Further, in applications where ancestry information is not required, it is also possible to estimate spawned objects using RFS-based multi-object filters with measurement-driven (spontaneous) birth models [74], [65]. Hence, a complete treatment of modeling and estimation for spawning objects should address the issue of ancestry.

Labeled RFSs enable ancestry information to be incorporated into the modeling and estimation of spawn objects. Approximate multi-object Bayes filters such as the PHD [61], CPHD [62], and multi-Bernoulli [63, 93, 97] filters were not formulated to estimate tracks, without which the ancestries of the objects are, conceptually, not traceable. On the other hand, labeled RFSs provide the means for identifying and estimating individual object tracks [95], thereby making it possible, conceptually, to trace their ancestors. Furthermore, as demonstrated in this thesis, the labels used to identify individual tracks can also be encoded with ancestry information, which can be assimilated by RFS spawning models, and subsequently inferred from the labels obtained using labeled RFS estimation techniques.

Under the labeled RFS formulation, the multi-object Bayes recursion (without spawning) admits an analytic solution known as the GLMB filter [94, 95], which can be implemented with

[7] space-track.org. [online] Available at: https://www.space-track.org/ [12 Jun. 2016]

linear complexity in the number of measurements and quadratic in the number of hypothesized tracks [92]. This on-line multi-object tracker is based on the GLMB family of conjugate priors that enjoys a number of nice analytical properties, e.g., the void probability functional–a necessary and sufficient statistic–of a GLMB, the Cauchy-Schwarz divergence between two GLMBs [8], the L_1-distance between a GLMB and its truncation, can all be computed in closed form [94]. Of direct relevance to this thesis is the fact that the GLMB family is flexible enough to approximate any labeled RFS density with matching intensity function and cardinality distribution [70].

1.3 Overview

This thesis develops new RFS multi-object filtering/tracking methods that jointly estimate the states and number of objects that appear due to spawning events, with intended applications to IOD. There are multiple RFS filters presented in the literature and choosing the best options for extending their capabilities to meet the demands of IOD requires special considerations. These include, but are not limited to, the multi-object state models and track labeling methods available for a given approach.

Chapter 2 of this thesis provides relevant background on multi-object filtering and tracking within an RFS framework. Section 2.1 covers RFS multi-object filtering methods while Section 2.2 presents labeled RFS multi-object tracking methods. Gaussian mixture implementations of two filters central to the presented research are presented in Section 2.3. Their tractability is discussed in Section 2.4 and the performance metrics used to evaluate them are defined in Section 2.5. An overview of the orbit problem is provided in Section 2.6. Uncertainty propagation methods are discussed in Section 2.7, with detection probability and breakup modeling detailed in Section 2.8 and Section 2.9, respectively.

Chapter 3 of this thesis provides additional background material particularly relevant to the CPHD derivations presented in Chapter 4. Random finite sets are "simplified" versions of point processes [64], which are formulated in the more general theory of measures. According to Mahler, a point process must be a RFS to be of practical use, and the use of a point process

theory approach to multi-object filtering developments "unnecessarily increase[s] notational and theoretical complexity" [64, pg. 52]. While the addition of complexity cannot be denied, in the case of the research presented in Chapter 4, the point process theory approach, i.e., measure-theoretic approach, is key in facilitating a cohesive formulation of the spawn model inclusive CPHD filter, regardless of the specific spawn model used. Hence, Chapter 3 includes the relevant background for a measure-theoretic approach to CPHD derivations. Section 3.1 discusses point processes and their relationship to RFSs. Sections 3.2 and 3.3 define probability generating functionals and the mechanics of functional differentiation, respectively, while a few properties from the application of differentiation in the context of point processes are shown in Section 3.4.

In Chapter 4, models are proposed that allow for the development of CPHD implementations used for RSO filtering applications with spawning. The incorporation of spawn models in the context of CPHD filtering was previously explored in [60], relying on an intuitive construction of the filtering equations related to the spawning models considered (Bernoulli or Poisson process) through a non-standard derivation procedure. This thesis proposes expressions for the CPHD filter enhanced with various target spawning models through a standard derivation procedure within an RFS framework specific to the considered spawning model (Bernoulli, Poisson, or zero-inflated Poisson process). The derivation of the spawning terms in [60] relies on additional approximations and the approach does not lead to the same results as those presented here. Section 4.1 provides a detailed construction of the CPHD filter with target spawning, considering several models of spawning processes. Section 4.2 demonstrates the proposed concepts through simulation example. The proofs of the results in Section 4.1 are given in Appendix A.

Chapter 5 of this thesis presents multi-object filtering simulations for two spawning event orbit determination scenarios: small-sat deployments and a rocket body (RB) explosion. CubeSat deployments offer a unique opportunity to test spawn model filter implementations because they are typically scheduled, well defined, and controlled events. The QB50 mission [87] in which fifty CubeSats were planned for deployment in LEO is a good example of this as its mission design is well documented, including a study of deployment strategies [53]. Important to note, in the time since

the research presented in Chapter 5 was performed, according to the QB-50 mission's website[8], only thirty-six QB50 CubeSats have been launched: twenty-eight from the international space station (ISS) and eight from an Indian PSLV rocket. The presented simulations were performed in early 2016 at which time the QB50 mission was reportedly preparing to launch fifty CubeSats from the ISS that summer[9]. Rocket body explosions are a well known fragmentation event type and Space Surveillance Network (SSN) catalog entries for their fragments are periodically reported, e.g., a Delta 2 second-stage (SSN# 25637, International Designator 1999-008D) explosion yielded the addition of six fragments into the SSN catalog[10]. Scenario descriptions are provided in Section 5.1 which includes information pertaining to dynamics, measurements, and filter configuration. Results are presented in Section 5.2.

In Chapter 6 of this thesis, a new GLMB based filter that formally incorporates spawning, in addition to birth, is proposed. Using labeled RFSs ancestry information is encoded into the labels of individual object states and a labeled RFS spawn model is proposed. When a track is instantiated by spontaneous birth, its label contains information pertaining to when an object is born and from which birth region [95]. Similarly, for a track instantiated by spawning, its label contains information pertaining to when and from which parent it originated. Under such a spawning model, the multi-object prediction and filtering densities are no longer GLMBs, even if the initial prior is a GLMB. To derive a tractable filter, following [92] the prediction and update are combined into a single step and the labeled multi-object filtering density is approximated by a GLMB with matching first moment and cardinality using the technique in [70]. The result is a recursion that propagates the GLMB approximation of the labeled multi-object filtering density, from which the states of the spawned objects and their labels, hence lineage, can be jointly inferred. The derivation, approximation, and joint prediction and update of object spawning inclusive GLMB

[8] [online] Available at: https://www.qb50.eu/. [Accessed 10 Oct. 2017].

[9] [online] Available at: http://amsat-uk.org/2015/09/08/qb50-cubesats-to-be-deployed-from-iss/ [Accessed 10 Oct. 2017].

[10] orbitaldebris.jsc.nasa.gov. odqnv18i3.pdf (pdf). [online] Available at: https://orbitaldebris.jsc.nasa.gov/quarterly-news/pdfs/odqnv18i3.pdf [Accessed 11 Sep. 2014].

densities is developed in Section 6.1. Simulation results are presented in Section 6.5.

Chapter 7 of this thesis demonstrates that the filter advancements made in Chapter 6 are capable of performing IOD for objects generated by space based spawning events. CubeSat deployment is revisited, but with a more challenging scenario in which the simulated sensor network more closely resembles real-world capabilities. Multiple simulations are run that highlight the need for more versatile spawn models for SSA applications.

The contributions and significant results of the presented research are reviewed in Chapter 8. Future work is proposed for further filter advancements and improved SSA capabilities.

1.4 Summary of Contributions

Derivations were performed so that spawn modeling could be incorporated with the CPHD filter, at a fundamental level. The resultant expressions for predicted cardinality distribution were in general form, w.r.t. input spawn models, i.e., special considerations for spawn modeling prior to derivation were not required. A tractable implementation method was presented using Partial Bell Polynomials, and the Zero-Inflated Poisson spawn model was introduced. Filter capabilities were demonstrated via linear simulations.

- Results published in the peer reviewed journal IEEE, Transactions on Signal Processing [14]

The Gaussian mixture implementation of the CPHD filter, configured with a Zero-Inflated Poisson spawn model, was shown to be effective in IOD of objects originating from spawning events. When high observation density is achievable, the GM-CPHD filter can be a useful tool in estimating the number of objects generated by CubeSat deployments and/or fragmentation events and their states.

- Results presented at the 2016 AAS/AIAA Spaceflight Mechanics Meeting [13]

A GLMB filter implementation was derived that includes spawning; the first multi-object filter ever derived capable of providing ancestry information estimates, along with number of objects and individual object state estimates. Presented in a fundamental form, the derived results

are accessible to any sub-discipline of the multi-object filtering/tracking field where spawning is applicable.

- Results currently under review for publication in the peer reviewed journal IEEE, Transactions on Signal Processing
- Mathematical derivations presented at the 2017 International Conference on Control, Automation and Information Sciences

Additionally, this research has shown that on-line multi-object orbit determination in the presence of spawning is possible within the RFS filtering paradigm. Having shown that the GLMB filter can perform initial orbit determination for spawned RSOs, this research has established a foundation upon which further SSA improvements can be investigated.

Chapter 2

Multi-Object Filtering and Tracking with Random Finite Sets

This chapter provides background material on multi-object filtering and multi-object tracking within an RFS framework relevant to the presentation and discussion of advancements in this area later on. Section 2.1 covers RFS multi-object filtering methods while Section 2.2 presents labeled RFS multi-object tracking methods. Gaussian mixture implementations of two filters central to the presented research are presented in Section 2.3, their tractability is discussed in Section 2.4, and the performance metrics used to evaluate them are defined in Section 2.5. Uncertainty propagation methods are discussed in Section 2.7, with detection probability and breakup modeling detailed in Section 2.8 and Section 2.9, respectively.

2.1 Random Finite Set Filtering

2.1.1 Random Finite Sets

The content of this section closely follows that in [89].

Instead of a single system state x in the state space $\mathbb{X} \subseteq \mathbb{R}^n$ or observation z in observation space $\mathbb{Z} \subseteq \mathbb{R}^m$, a multi-object system considers finite sets $X \subset \mathbb{X}$ and $Z \subset \mathbb{Z}$ as the multi-object state and multi-object observation, respectively. Furthermore, in a Bayesian framework the multi-object state and multi-object observation are modeled as RFSs, i.e., finite-set-valued random variables [61]. An RFS, also known as a simple finite point process, consists of a random number of points that are, themselves, random and unordered. An RFS can be described by the multi-object density-defined to be the set derivative of its belief functional [63]-shown to be equivalent to a

probability density in [90].

Assume that at time k there is a collection of unordered single object states $x_k^{(i)}$ with $i = 1, 2, \ldots, J_k^{(x)}$ and a collection of unordered measurements $z_k^{(i)}$ with $i = 1, 2, \ldots, J_k^{(z)}$. Then, the multi-object state and multi-object observation at time k are

$$X_k = \left\{ x_k^{(1)}, \ldots, x_k^{(J_k^{(x)})} \right\} \in \mathcal{F}(\mathbb{X}), \tag{2.1}$$

$$Z_k = \left\{ z_k^{(1)}, \ldots, z_k^{(J_k^{(z)})} \right\} \in \mathcal{F}(\mathbb{Z}), \tag{2.2}$$

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

Depending on the intended application, the multi-object state can take on several different forms [89, 90, 96, 97]. Letting X_{k-1} denote the multi-object state at time $k-1$, each $x_{k-1} \in X_{k-1}$ will either survive to the next time k with probability of survival $p_{\mathrm{S},k}(x_{k-1})$ or die with probability $1 - p_{\mathrm{S},k}(x_{k-1})$. Define a surviving RFS $S_{k|k-1}$ that models the behavior of X_{k-1} at the next time. Also, define a spontaneous birth RFS B_k at time k that accounts for newly appearing objects, e.g., objects that recently entered a surveillance region, and a spawn RFS $T_{k|k-1}(x_{k-1})$ at time k that accounts for objects spawned from an object with previous state x_{k-1}. Then, the predicted multi-object state can be formulated in the following ways:

$$X_k = \left[\bigcup_{x_{k-1} \in X_{k-1}} S_{k|k-1}(x_{k-1}) \right], \text{ surviving only,} \tag{2.3}$$

$$X_k = \left[\bigcup_{x_{k-1} \in X_{k-1}} S_{k|k-1}(x_{k-1}) \right] \cup B_k, \text{ surviving and birth,} \tag{2.4}$$

$$X_k = \left[\bigcup_{x_{k-1} \in X_{k-1}} S_{k|k-1}(x_{k-1}) \right] \cup \left[\bigcup_{x_{k-1} \in X_{k-1}} T_{k|k-1}(x_{k-1}) \right], \text{ surviving and spawn,} \tag{2.5}$$

$$X_k = \left[\bigcup_{x_{k-1} \in X_{k-1}} S_{k|k-1}(x_{k-1}) \right] \cup \left[\bigcup_{x_{k-1} \in X_{k-1}} T_{k|k-1}(x_{k-1}) \right] \cup B_k, \text{ surviving, spawn and birth.} \tag{2.6}$$

The multi-object observation Z_k is the superposition of detections and false alarms, i.e., clutter returns. Each state $x_k \in X_k$ is either detected with probability of detection $p_{\mathrm{D},k}(x_k)$ and generates an observation $z_k \in Z_k$ or missed with probability $q_{\mathrm{D},k}(x_k) = 1 - p_{\mathrm{D},k}(x_k)$. Clutter models typically follow a Poisson distribution and their statistics are usually denoted by the Greek

letter κ.

2.1.2 Multi-Object Bayes Filter

The content of this section closely follows that in [14].

The multi-object Bayes filter [63] is the natural extension of the usual single-object Bayesian paradigm to the multi-object case. Letting X_k and Z_k denote the multi-object state and multi-object measurement at time k, respectively, the multi-object filtering density is recursively propagated by the multi-object Bayes filter [61, 63] via

$$\pi_{k|k-1}(X_k|Z_{1:k-1}) = \int f_{k|k-1}(X_k|X)\pi_{k-1}(X|Z_{1:k-1})\delta X, \tag{2.7}$$

$$\pi_k(X_k|Z_{1:k}) = \frac{g_k(Z_k|X_k)\pi_{k|k-1}(X_k|Z_{1:k-1})}{\int g_k(Z_k|X)\pi_{k|k-1}(X|Z_{1:k-1})\delta X}, \tag{2.8}$$

where the integral is a set integral defined for any function $f : \mathcal{F}(\mathbb{X}) \to \mathbb{R}$ by

$$\int f(X)\delta X = \sum_{i=0}^{\infty} \frac{1}{i!} \int f(\{x_1, \ldots, x_i\})\mathrm{d}(x_1, \ldots, x_i), \tag{2.9}$$

$Z_{1:k}$ denotes the measurement history up to time k, $\pi_{k|k-1}$ is the predicted multi-object density, and π_k is the multi-object posterior density. The multi-object transition density $f_{k|k-1}$ describes the time evolution of the population of objects since time step $k-1$ and encapsulates the underlying models of object birth, motion, spawning, and death. The multi-object likelihood g_k describes the sensor observation process and encapsulates the underlying models of object detection, object-generated measurements, and false alarms (i.e., clutter).

Practical applications of the recursion in (2.7) and (2.8) are typically deemed computationally intractable due to the involvement of multiple integrals on the space $\mathcal{F}(\mathbb{X})$ [89]; however, several of its approximations circumvent this intractability and are detailed next.

2.1.3 First-Order Approximation Filters

Instead of propagating the multi-object posterior density, first-order approximation filters propagate the first-order multi-object moment, or intensity, commonly known as the Probability

Hypothesis Density (PHD). Such an approximation reduces computational complexity by allowing for integration of terms over the single-object state space $\mathbb{X}$, as opposed to over the space $\mathcal{F}(\mathbb{X})$ [89]. Note that first-order approximation filters provide number of objects and object state estimates; however, they do not provide object tracks, i.e., they do not provide object identity information.

The PHD [63] and CPHD [96] filters are first-order approximations of the multi-object Bayes recursion that have been substantially researched in the past decade. Their details are provided below, but first, define the following terms in Table 2.1 (as they are presented in [89]):

Table 2.1: Notation Definitions [89]

Notation	Definition
$\mu_{\mathrm{B},k}(\cdot)$	intensity of the birth RFS B_k at time k
$\mu_{\mathrm{T},k\|k-1}(\cdot\|x_{k-1})$	intensity of the RFS $T_{k\|k-1}(x_{k-1})$ spawned at time k by an object with previous state x_{k-1}
$p_{\mathrm{S},k}(x_{k-1})$	probability of survival, i.e., probability that an object still exists at time k given that that its previous state is x_{k-1}
$p_{\mathrm{D},k}(x)$	probability of detection, i.e., probability of an object at time k given state x being detected by a sensor
κ_k	intensity of clutter RFS K_k at time k

2.1.3.1 PHD Filter

The content of this section closely follows that in [89].

The original PHD filter derivation assumes the multi-object state is given by (2.6). The following are also assumed:

A.2.1: objects evolve and generate observations independently from one another,

A.2.2: the clutter RFS is Poisson and independent of object-originated observations,

A.2.3: the predicted multi-object RFS (Eq. (2.7)) follows a Poisson distribution.

Remark 1: As noted in [89], assumptions A.2.1 and A.2.2 are fairly standard for tracking applications.

Remark 2: The PHD filter can have trouble with scenarios where the number of objects vary over time, e.g., fragmentation events. This is due to the "linearizing effect" the Poisson approximation in assumption A.2.3 has on the estimated number of objects [63, 64].

Letting $\mu_{k|k-1}$ and μ_k denote the intensities associated with $\pi_{k|k-1}$ in (2.7) and π_k in (2.8), respectively, the PHD recursion is defined as [61, 89],

$$\mu_{k|k-1}(x) = \int p_{\mathrm{S},k}(\varsigma) f_{k|k-1} \mu_{k-1}(\varsigma) \mathrm{d}\varsigma + \int \mu_{\mathrm{T},k|k-1}(x|\varsigma) \mu_{k-1}(\varsigma) \mathrm{d}\varsigma + \mu_{\mathrm{B},k}(x), \tag{2.10}$$

$$\mu_k(x) = \left[1 - p_{\mathrm{D},k}(x)\right] \mu_{k|k-1}(x) + \sum_{z \in Z_k} \frac{p_{\mathrm{D},k}(x) g_k(z|x) \mu_{k|k-1}(x)}{\kappa_k(z) + \int_{\mathbb{X}} p_{\mathrm{D},k}(\xi) g_k(z|\xi) \mu_{k|k-1}(\xi) \mathrm{d}\xi}. \tag{2.11}$$

2.1.3.2 CPHD Filter

The content of this section closely follows that in [96].

The original derivation of the CPHD filter did not account for spawned objects [62, 96], hence it assumed the multi-object state model (2.4). Noting that A.2.1 still holds, the remaining CPHD filter assumptions are:

A.2.4: birth and surviving RFSs are independent of each other,

A.2.5: the clutter RFS is an independent and identically distributed (i.i.d.) process, independent of the measurement RFSs,

A.2.6: the prior and predicted multi-object RFSs are i.i.d. processes.

Remark 3: A.2.5 and A.2.6 include the assumption of an i.i.d. process which is characterized by a cardinality distribution ρ and a spatial distribution. It describes a population whose size is distributed according to ρ and whose individual states are i.i.d.. A Poisson process is a special case of an i.i.d. process in which the cardinality distribution ρ is Poisson.

The prediction step of the CPHD filter is similar to that of the PHD, except that the predicted intensity $\mu_{k|k-1}$ does not include spawning terms, and, there is an equation for the predicted cardinality distribution $\rho_{k|k-1}$. The prediction step of the classical CPHD recursion is given by [96]

$$\mu_{k|k-1}(x) = \int p_{\mathrm{S},k}(\varsigma) f_{k|k-1}\mu_{k-1}(\varsigma)\mathrm{d}\varsigma + \mu_{\mathrm{B},k}(x), \tag{2.12}$$

$$\rho_{k|k-1}(n) = \sum_{j=0}^{n} \rho_{\mathrm{B},k}(n-j) \sum_{\ell=j}^{\infty} C_j^{\ell} \frac{\langle p_{\mathrm{S},k}(\varsigma), \mu_{k-1}\rangle^{j} \langle 1 - p_{\mathrm{S},k}(\varsigma), \mu_{k-1}\rangle^{\ell-j}}{\langle 1, \mu_{k-1}\rangle^{\ell}} \rho_{k-1}(\ell), \tag{2.13}$$

where

$$\rho_{\mathrm{B},k}(\cdot) = \text{cardinality distribution of births at time } k,$$

$$\langle \lambda, \epsilon \rangle = \text{the inner product defined between two real-valued functions } \lambda \text{ and } \epsilon,$$

$$C_j^{\ell} = \text{the binomial coefficient} \frac{\ell!}{j!(\ell-j)!}.$$

The update step of the CPHD filter is given by [96]

$$\begin{aligned}\mu_k =& \frac{\langle \Upsilon_k^1 \left[\mu_{k|k-1}, Z_k\right], \rho_{k|k-1}\rangle}{\langle \Upsilon_k^0 \left[\mu_{k|k-1}, Z_k\right], \rho_{k|k-1}\rangle} \times \left[1 - p_{\mathrm{D},k}(x)\right] \mu_{k|k-1}(x) \\ &+ \sum_{z \in Z_k} \frac{\langle \Upsilon_k^1 \left[\mu_{k|k-1}, Z_k \setminus \{z\}\right], \rho_{k|k-1}\rangle}{\langle \Upsilon_k^0 \left[\mu_{k|k-1}, Z_k\right], \rho_{k|k-1}\rangle} \frac{\langle 1, \kappa_k\rangle}{\kappa_k(z)} g_k(z|x) p_{\mathrm{D},k}(x) \mu_{k|k-1}(x),\end{aligned} \tag{2.14}$$

$$\rho_k(n) = \frac{\Upsilon_k^0 \left[\mu_{k|k-1}, Z_k\right](n), \rho_{k|k-1}(n)}{\langle \Upsilon_k^0 \left[\mu_{k|k-1}, Z_k\right], \rho_{k|k-1}\rangle}, \tag{2.15}$$

where

$$\begin{aligned}\Upsilon_k^u \left[\mu, Z\right](n) = & \sum_{j=0}^{\min(|Z|,n)} \left(|Z| - j\right) \rho_{K,k}\left(|Z| - j\right) P_{j+u}^{n} \\ & \times \frac{\langle 1 - p_{\mathrm{D},k}(x), \mu\rangle^{n-(j+u)}}{\langle 1, \mu\rangle^{n}} e_j\big(\Lambda_k(\mu, Z)\big),\end{aligned} \tag{2.16}$$

$$\Lambda(\mu, Z) = \left\{ \langle \mu, \frac{\langle 1, \kappa_k\rangle}{\kappa_k(z)} g_k(z|x) p_{\mathrm{D},k}(x)\rangle : z \in Z \right\}, \tag{2.17}$$

$$\rho_{K,k}(\cdot) = \text{cardinality distribution of clutter at time } k,$$

$$P_j^n = \text{the permutation coefficient} \frac{n!}{(n-j)!},$$

$$e_j(\Lambda) = \text{the elementary symmetric function of order } j \text{ defined}$$

$$\text{for a finite set } \Lambda \text{ of real numbers.}$$

As can be seen in (2.12)-(2.17), the CPHD filter is more complex than the PHD filter, though it has been shown that the former can provide a better cardinality estimate than the latter [96]. Notwithstanding, both filters are known to exhibit a behavior referred to as "spooky action at a distance" [32], or the "spooky effect" [94], whereupon a missed detection the PHD *mass* is shifted away from undetected tracks to detected tracks, regardless of the distance between tracks. An example given in [32] using only two tracks illustrates how a portion of the updated weight of the undetected track is transferred to the detected track. A proposed remedy to this issue is to partition a multi-object scene into "statistically noninteracting target-clusters" and apply a single CPHD filter to each cluster [64]. For example, if a multi-object scenario involved tracking RSOs in both LEO and geosynchronous-Earth orbit (GEO), an individual CPHD filter would be applied to tracking LEO objects, with another filter applied to tracking GEO objects.

2.1.4 Multi-Bernoulli Approximation Filters

The content of this section closely follows that in [97].

Instead of approximating the multi-object density π of the multi-object Bayes recursion as a Poisson process (PHD filter) or an i.i.d. process (CPHD filter), approaches discussed in this section make a "less restrictive assumption" [63] of multi-object densities as approximately multi-Bernoulli [63]. According to Mahler, analogous to the way closed form solutions of the single-object Kalman filter are made possible when the relevant distributions are linear Gaussian, closed form solutions for the multi-object Bayes filter are possible when the relevant distributions are at least approximately multi-Bernoulli [63].

The first proposed multi-Bernoulli filter is the Multi-target Multi-Bernoulli (MeMBer) filter [63], which was determined to have a problem with over estimating the number of objects, i.e., cardinality, hence the proposal of the Cardinality Balanced MeMBer (CBMeMBer) filter [97]. These filters are not discussed here in detail, as they are not the focus of the presented work. However, both filters make use of multi-Bernoulli RFSs which are relevant in later sections.

2.1.4.1 Multi-Bernoulli RFSs

A Bernoulli RFS X on $\mathbb{X}$ describes a single object and is parameterized by a probability of existence $r \in (0,1)$ and a spatial distribution $p(x)$ that describes the state $x \in \mathbb{X}$. The probability density function (p.d.f.) of X is given by

$$\pi(X) = \begin{cases} 1-r, & X = \emptyset, \\ r \cdot p(x), & X = \{x\}. \end{cases} \tag{2.18}$$

A multi-Bernoulli RFS X on $\mathbb{X}$ is a union of M independent RFSs $X^{(i)}$, each with *existence* probability $r^{(i)} \in (0,1)$ and probability density $p^{(i)}(x)$ (defined on $\mathbb{X}$), where $i = 1, \ldots, M$. The probability density π of a multi-Bernoulli RFS is given as [63]

$$\pi(X) = \pi(\emptyset) \sum_{1 \leq i_1 \neq \cdots \neq i_n \leq M} \prod_{j=1}^{n} \frac{r^{(i_j)} p^{(i_j)}(x_j)}{1 - r^{(i_j)}}, \tag{2.19}$$

which is typically expressed more compactly as a set of parameters by [94, 95, 97]

$$\pi \triangleq \{(r^{(i)}, p^{(i)})\}_{i=1}^{M}. \tag{2.20}$$

2.2 Labeled Random Finite Set Tracking

The content of this section and its subsections closely follows that in [94, 95].

The previously detailed multi-object filters are capable of estimating the number of objects and their states, but provide no information on object identity. Labeled implementations of these filters do exist [21, 58]; however, the track labels themselves do not result from top down derivation, but are algorithmically added to the filters. Therefore, fundamentally, the previously discussed filters perform multi-object filtering, but not tracking.

A new approach to RFS multi-object tracking was recently developed that centers around the labeled RFS. An analytic solution to the labeled multi-object Bayes filter (2.29), known as the Generalized Labeled Multi-Bernoulli (GLMB) filter, was derived in [95], and is the focus of this section.

2.2.1 Label Random Finite Sets

A labeled RFS is a marked simple finite point process with state space $\mathbb{X}$ and discrete mark space $\mathbb{L}$, such that each realization has distinct marks [95], [94]. The distinct marks, or labels, provide the means to identify trajectories or tracks of individual objects since a trajectory is a time-sequence of states with the same label. Let $\mathcal{L} : \mathbb{X} \times \mathbb{L} \to \mathbb{L}$ be the projection $\mathcal{L}((x, \ell)) = \ell$, then the labels of realization $\mathbf{X} \subset \mathbb{X} \times \mathbb{L}$ are $\mathcal{L}(\mathbf{X}) = \{\mathcal{L}(\mathbf{x}) : \mathbf{x} \in \mathbf{X}\}$. The realization $\mathbf{X}$ is said to have *distinct labels* if and only if it has the the same cardinality as its labels $\mathcal{L}(\mathbf{X})$. This concept is compactly formulated by the *distinct label indicator* defined by [95], [94]

$$\Delta(\mathbf{X}) = \delta_{|\mathbf{X}|}\left(|\mathcal{L}(\mathbf{X})|\right),$$

where $|X|$ denotes the cardinality of a finite set X, and

$$\delta_Y(X) \triangleq \begin{cases} 1, & \text{if } X = Y, \\ 0, & \text{otherwise,} \end{cases}$$

denotes a generalization of the Kroneker delta that takes arbitrary arguments such as integers, sets, vectors etc.

Throughout this document the following notation conventions are observed. Lower case letters represent single-object states, e.g., $x, \mathbf{x}$, while upper case letters represent multi-object states, e.g., $X, \mathbf{X}$. Bold symbols represent labeled states and their distributions/statistics, e.g., $\mathbf{x}, \mathbf{X}, \boldsymbol{\pi}$, etc., to distinguish them from unlabeled ones. Blackboard letters represent spaces, e.g., $\mathbb{X}, \mathbb{Z}, \mathbb{L}, \mathbb{N}$. Multi-object exponential notation $h^X \triangleq \prod_{x \in X} h(x)$, where h is a real-valued function and $h^{\emptyset} = 1$ by convention, and the inclusion function notation, a generalization of the indicator function,

$$1_Y(X) \triangleq \begin{cases} 1, & \text{if } X \subseteq Y, \\ 0, & \text{otherwise,} \end{cases}$$

are also used.

Additionally, the list of variables $X_m, X_{m+1}, ..., X_n$ is abbreviated as $X_{m:n}$ and where it is convenient we let the symbol + denote the time index at the *next time* and its absence denote the time index at the *current time*, e.g., the state x_k at the current time and the state x_{k+1} at the next time can equivalently be denoted as x and x_+, respectively.

2.2.2 Generalized Labeled Multi-Bernoulli

A GLMB is a labeled RFS with state space $\mathbb{X}$ and label space $\mathbb{L}$ distributed according to [94, 95]

$$\boldsymbol{\pi}(\mathbf{X}) = \Delta(\mathbf{X}) \sum_{(I,\xi)\in\mathcal{F}(\mathbb{L})\times\Xi} w^{(I,\xi)} \delta_I(\mathcal{L}(\mathbf{X})) \left[p^{(\xi)}\right]^{\mathbf{X}} \tag{2.21}$$

where Ξ is a given discrete space, each $p^{(\xi)}(\cdot,\ell)$ is a (single-object) probability density on $\mathbb{X}$ (i.e., $\int p^{(\xi)}(x,\ell)dx = 1$ with each $x \in \mathbb{X}$ denoting a single-object state and each $\ell \in \mathbb{L}$ denoting a distinct label), and each $w^{(I,\xi)}$ is non-negative such that

$$\sum_{I\in\mathcal{F}(\mathbb{L})} \sum_{\xi\in\Xi} w^{(I,\xi)}(L) = 1. \tag{2.22}$$

Each GLMB density component (I,ξ) in (2.21) consists of a weight $w^{(I,\xi)}$ that depends solely on the labels of the multi-object state $\mathbf{X}$ and the multi-object exponential $\left[p^{(\xi)}\right]^{\mathbf{X}}$, which is a product of single-object probability densities.

Also relevant to this work is the Labeled Multi-Bernoulli (LMB). An LMB $\mathbf{X}$ defined on $\mathbb{X}\times\mathbb{L}$ is an RFS with parameter set $\{(r^{(\varsigma)}, p^{(\varsigma)}) : \varsigma \in \Psi\}$ distributed according to [95]

$$\boldsymbol{\pi}(\mathbf{X}) = \Delta(\mathbf{X}) 1_{\alpha(\Psi)}(\mathcal{L}(\mathbf{X})) \left[\Phi(\mathbf{X},\cdot)\right]^{\Psi} \tag{2.23}$$

where $\alpha : \Psi \to \mathbb{L}$ is a 1-1 mapping (usually an identity mapping) and

$$\Phi(\mathbf{X},\varsigma) = \sum_{(x,\ell)\in\mathbf{X}} \delta_{\alpha(\varsigma)}(\ell) r^{(\varsigma)} p^{(\varsigma)}(x) + \left(1 - 1_{\mathcal{L}(\mathbf{X})}(\alpha(\varsigma))\right)\left(1 - r^{(\varsigma)}\right). \tag{2.24}$$

2.2.3 GLMB Tracking Filter

Pertinent details of the GLMB tracking filter are presented in this subsection. There is some overlap between the filter recursion development presented here and that presented in Section 2.1.2,

and it may seem a bit repetitive. However, the multi-object filtering problem is now formulated using labeled RFSs and they, along with their statistical descriptors, use different notation. So, for the sake of clarity, some expressions are presented again using labeled notation.

2.2.3.1 Multi-object Bayes Filter

Using the convention detailed in [95], a label $\ell = (t, i)$ in the space $\mathbb{L}$ of labels at the current time k is an ordered pair, where the first term $t \leq k$ denotes time of birth, and the second term $i \in \mathbb{N}$ is a unique index distinguishing objects born at the same time. Birth labels at the next time belong to the space $\mathbb{B}_+ = \{(k+1, i) : i \in \mathbb{N}\}$, hence $\mathbb{L} \cap \mathbb{B}_+ = \emptyset$ and the label space at the next time is $\mathbb{L}_+ = \mathbb{L} \cup \mathbb{B}_+$. The assignment of labels to object trajectories is illustrated in Fig. 2.1 [94].

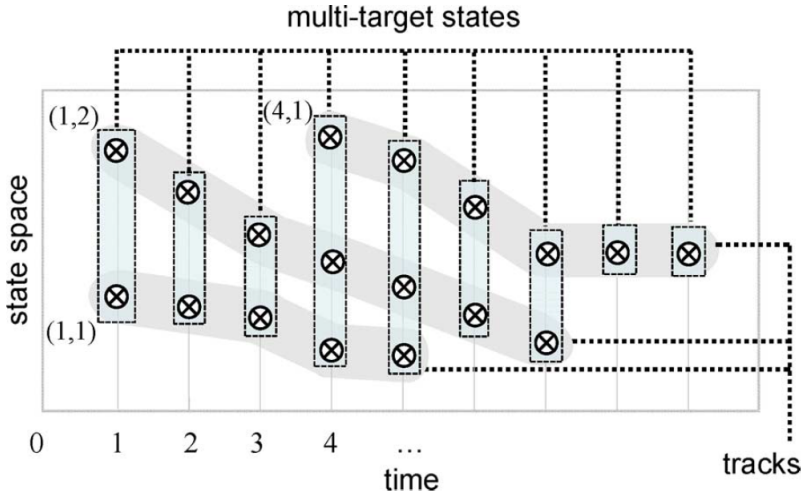


Figure 2.1: "An example of label assignments. The two tracks born at time 1 are given labels (1,1) and (1,2), while the only track born at time 4 is given label (4,1). Notice also the difference between multi-target states and target tracks." [94]

The history $\mathbf{X}_{0:k}$ of labeled multi-object states contains the set of all trajectories up to time k. All information on the set of trajectories conditioned on the observation history $Z_{1:k}$, is captured in the multi-object posterior density $\boldsymbol{\pi}_{0:k}(\cdot|Z_{1:k})$, which incorporates the evolution of the multi-object state via the multi-object transition density, as well as the observed data via the *multi-object likelihood* [63].

2.2.3.2 Multi-Object Transition Model

The multi-object transition density $\mathbf{f}_+(\cdot|\mathbf{X})$ models the evolution of a given multi-object state $\mathbf{X}$ to the next time and encapsulates all information pertaining to loss of objects via thinning, movement of surviving objects via Markov shifts[1], and appearance of new objects via superposition.

Given a single-object state $\mathbf{x} \in \mathbf{X}$ at the current time, an object either survives to the next time with probability $p_{\mathrm{S}}(x, \ell)$ and moves to a new state (x_+, ℓ_+) with probability density $f_{\mathrm{S},+}(x_+|x, \ell)\delta_\ell(\ell_+)$, or dies with probability $q_{\mathrm{S}}(x, \ell) = 1 - p_{\mathrm{S}}(x, \ell)$. Assuming that the transition of kinematic states are mutually independent and conditional on $\mathbf{X}$, we model the set $\mathbf{X}_{\mathrm{S},+}$ of surviving objects at the next time as a conditional LMB RFS distributed according to [95]

$$\mathbf{f}_{\mathrm{S},+}(\mathbf{X}_{\mathrm{S},+}|\mathbf{X}) = \Delta(\mathbf{X})\Delta(\mathbf{X}_{\mathrm{S},+})1_{\mathcal{L}(\mathbf{X})}(\mathcal{L}(\mathbf{X}_{\mathrm{S},+}))\left[\Phi_{\mathrm{S},+}(\mathbf{X}_{\mathrm{S},+}|\cdot)\right]^{\mathbf{X}} \tag{2.25}$$

where

$$\Phi_{\mathrm{S},+}(\mathbf{X}_{\mathrm{S},+}|x, \ell) = \sum_{(x_+,\ell_+)\in\mathbf{X}_{\mathrm{S},+}} \delta_\ell(\ell_+)p_{\mathrm{S}}(x, \ell)f_{\mathrm{S},+}(x_+|x, \ell) + \left[1 - 1_{\mathcal{L}(\mathbf{X}_{\mathrm{S},+})}(\ell)\right] q_{\mathrm{S}}(x, \ell).$$

A new object with state (x_+, ℓ_+) appears at the next time with probability $r_{\mathrm{B},+}(\ell_+)$ and probability density $p_{\mathrm{B},+}(x_+, \ell_+)$, or does not with probability $1 - r_{\mathrm{B},+}(\ell_+)$. Modeling object birth as an LMB RFS, the set $\mathbf{X}_{\mathrm{B},+}$ of new objects born at the next time is distributed according to [95]

$$\mathbf{f}_{\mathrm{B},+}(\mathbf{X}_{\mathrm{B},+}) = \Delta(\mathbf{X}_{\mathrm{B},+})w_{\mathrm{B},+}(\mathcal{L}(\mathbf{X}_{\mathrm{B},+}))\left[p_{\mathrm{B},+}\right]^{\mathbf{X}_{\mathrm{B},+}} \tag{2.26}$$

where $w_{\mathrm{B},+}(L) = 1_{\mathbb{B}_+}(L)\left[1 - r_{\mathrm{B},+}\right]^{\mathbb{B}_+ - L}\left[r_{\mathrm{B},+}\right]^{L}$.

The multi-object state at the next time is the superposition of birth and surviving objects, i.e., $\mathbf{X}_+ = \mathbf{X}_{\mathrm{S},+} \cup \mathbf{X}_{\mathrm{B},+}$, and since $\mathbb{L} \cap \mathbb{B}_+ = \emptyset$, *labeled* birth and surviving objects are mutually independent. Thus, the multi-object transition kernel ultimately reduces to the product of birth and survival transition densities [95]

$$\mathbf{f}_+(\mathbf{X}_+|\mathbf{X}) = \mathbf{f}_{\mathrm{S},+}(\mathbf{X}_+ \cap (\mathbb{X} \times \mathbb{L})|\mathbf{X})\mathbf{f}_{\mathrm{B},+}(\mathbf{X}_+ - (\mathbb{X} \times \mathbb{L})), \tag{2.27}$$

where $\mathbf{X}_+ \cap (\mathbb{X} \times \mathbb{L})$ is the subset of $\mathbf{X}_+$ consisting of surviving objects.

[1] Markov shift refers to the Markov state transition density function, which is a probability density characterizing the state x_k of an object at time k, assuming it had state x_{k-1} at time $k-1$ [64].

2.2.3.3 Multi-object Measurement Model

The multi-object likelihood is a multi-object density $g(\cdot|\mathbf{X})$ that models the multi-object observation generated by a given multi-object state $\mathbf{X}$, and encapsulates all information pertaining to missed detections via thinning, detections (observations of detected objects) via Markov shifts and clutter (false observations) via superposition.

The multi-object observation $Z = \{z_1, \ldots, z_{|Z|}\}$ is the superposition of detections and clutter. Each state $(x, \ell) \in \mathbf{X}$ is either detected with probability $p_{\mathrm{D}}(x, \ell)$ and generates an observation $z \in Z$ with likelihood $g(z|x, \ell)$ or missed with probability $q_{\mathrm{D}}(x, \ell) = 1 - p_{\mathrm{D}}(x, \ell)$. The multi-object likelihood is given by [94, 95]

$$g(Z|\mathbf{X}) \propto \sum_{\theta \in \Theta(\mathcal{L}(\mathbf{X}))} \prod_{(x,\ell)\in\mathbf{X}} \psi^{(\theta(\ell))}(x, \ell|Z) \tag{2.28}$$

where

$$\psi^{(j)}\left(x, \ell| \left\{Z_{1:|Z|}\right\}\right) = \delta_0(j) q_{\mathrm{D}}(x, \ell) + (1 - \delta_0(j)) \frac{p_{\mathrm{D}}(x, \ell) g(z_j|x, \ell)}{\kappa(z_j)},$$

$\kappa(\cdot)$ is the intensity of Poisson clutter, and $\Theta(L)$ denotes the space of mappings $\theta : L \to \{0 : |Z|\}$ that are 1-1 when restricting the range to the positive integers, i.e., $\theta(i) = \theta(j) > 0$ implies $i = j$.

2.2.3.4 Multi-object Bayes recursion

While the multi-object posterior density can be approximated by Markov Chain Monte Carlo [23, 98], these techniques are still expensive and not suitable for on-line applications. For real-time tracking, a more tractable alternative is the marginal $\boldsymbol{\pi}(\cdot) \triangleq \boldsymbol{\pi}_k(\cdot|Z_{1:k})$ called the multi-object filtering density, which can be recursively propagated by the multi-object Bayes filter [61], [63]

$$\boldsymbol{\pi}_+(\mathbf{X}_+) \propto g_+(Z_+|\mathbf{X}_+) \int \mathbf{f}_+(\mathbf{X}_+|\mathbf{X}) \boldsymbol{\pi}(\mathbf{X}) \delta\mathbf{X}, \tag{2.29}$$

where the integral is a set integral defined for any function $\mathbf{f} : \mathcal{F}(\mathbb{X} \times \mathbb{L}_k) \to \mathbb{R}$ by

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

Note that Bayes optimal multi-object estimators can be formulated by minimizing the Bayes risk, e.g., the marginal multi-object estimator [63].

If the multi-object prior is a GLMB of the form (2.21), then the predicted multi-object density is also a GLMB given by [95]

$$\boldsymbol{\pi}_+(\mathbf{X}_+) = \Delta(\mathbf{X}_+) \sum_{(I_+,\xi)\in\mathcal{F}(\mathbb{L}_+)\times\Xi} w_+^{(I_+,\xi)} \delta_{I_+}(\mathcal{L}(\mathbf{X}_+))[p_+^{(\xi)}]^{\mathbf{X}_+}, \tag{2.30}$$

where

$$w_+^{(I_+,\xi)} = w_{\mathrm{B},+}(I_+ \cap \mathbb{B}_+) w_{\mathrm{S},+}^{(\xi)}(I_+ \cap \mathbb{L}), \tag{2.31}$$

$$p_+^{(\xi)}(x,\ell) = 1_{\mathbb{L}}(\ell) p_{\mathrm{S},+}^{(\xi)}(x,\ell) + (1 - 1_{\mathbb{L}}(\ell)) p_{\mathrm{B},+}(x,\ell), \tag{2.32}$$

$$p_{\mathrm{S},+}^{(\xi)}(x,\ell) = \frac{\langle p_{\mathrm{S},+}(\cdot,\ell) f_{\mathrm{S},+}(x|\cdot,\ell), p^{(\xi)}(\cdot,\ell)\rangle}{\eta_{\mathrm{S},+}^{(\xi)}(\ell)}, \tag{2.33}$$

$$\eta_{\mathrm{S},+}^{(\xi)}(\ell) = \int \langle p_{\mathrm{S},+}(\cdot,\ell) f_{\mathrm{S},+}(x|\cdot,\ell), p^{(\xi)}(\cdot,\ell)\rangle \mathrm{d}x, \tag{2.34}$$

$$w_{\mathrm{S},+}^{(\xi)}(L) = [\eta_{\mathrm{S},+}^{(\xi)}]^L \sum_{I\subseteq\mathbb{L}} 1_I(L) [q_{\mathrm{S},+}^{(\xi)}]^{I-L} w^{(I,\xi)}, \tag{2.35}$$

$$q_{\mathrm{S},+}^{(\xi)}(\ell) = \langle q_{\mathrm{S},+}(\cdot,\ell), p^{(\xi)}(\cdot,\ell)\rangle. \tag{2.36}$$

If the predicted multi-object density if a GLMB of (2.21), then the multi-object posterior is also a GLMB with form [95]

$$\boldsymbol{\pi}(\mathbf{X}|Z) = \Delta(\mathbf{X}) \sum_{(I_+,\xi)\in\mathcal{F}(\mathbb{L}_+)\times\Xi} \sum_{\theta\in\Theta} w^{(I,\xi,\theta)}(Z) \delta_I(\mathcal{L}(\mathbf{X}))[p^{(\xi,\theta)}(\cdot|Z)]^{\mathbf{X}}, \tag{2.37}$$

where

$$w^{(I,\xi,\theta)}(Z) = \frac{\delta_{\theta^{-1}(\{0:|Z|\})}(I) w^{(I,\xi)} [\eta_Z^{(\xi,\theta)}]^I}{\sum_{(I_+,\xi)\in\mathcal{F}(\mathbb{L}_+)\times\Xi} \sum_{\theta\in\Theta} \delta_{\theta^{-1}(\{0:|Z|\})}(I) w^{(I,\xi)} [\eta_Z^{(\xi,\theta)}]^I}, \tag{2.38}$$

$$p^{(\xi,\theta)}(x,\ell|Z) = \frac{p^{(\xi)}(x,\ell)\psi^{(\theta(\ell))}(x,\ell|Z)}{\eta^{(\xi,\theta)}(\ell|Z)}, \tag{2.39}$$

$$\eta^{(\xi,\theta)}(\ell|Z) = \langle p^{(\xi)}(\cdot,\ell), \psi^{(\theta(\ell))}(x,\ell|Z)\rangle, \tag{2.40}$$

$$\psi^{(\theta(\ell))}(x,\ell|Z) = \delta_0(\theta(\ell)) q_{\mathrm{D}}(x,\ell) + (1 - \delta_0(\theta(\ell))) \frac{p_{\mathrm{D}}(x,\ell) g(z_{\theta(\ell)}|x,\ell)}{\kappa(z_{\theta(\ell)})}. \tag{2.41}$$

2.3 Gaussian Mixture Implementations

Random finite set filters typically require further approximations to admit a closed form solution, the CPHD and GLMB included. As is usually done, these filters are implemented with either Gaussian mixture (GM) or Sequential Monte Carlo (SMC) models, the former typically more computationally expensive than the latter. Solutions to astrodynamics problems naturally tend to be computationally expensive due to complex dynamics, hence GM implementations are favored in presented work and SMC implementations are not considered.

In this section key assumptions and the associated equations are presented for GM implementation of both CPHD and GLMB filters, see [96] and [94] for more details, respectively. Note the following assumptions:

A.2.7: single-target dynamics and observation models are linear Gaussian,

A.2.8: survival and detection probabilities are state independent, i.e.,

$$p_{\mathrm{S},k}(x) \equiv p_{\mathrm{S},k}, \tag{2.42}$$

$$p_{\mathrm{D},k}(x) \equiv p_{\mathrm{D},k}. \tag{2.43}$$

2.3.1 GM-CPHD

The content of this section closely follows that in [96].

Assumption A.2.7 is mathematically represented as

$$f_{k|k-1}(x|\varsigma) = \mathcal{N}(x; F_{k-1}\varsigma, Q_{k-1}), \tag{2.44}$$

$$g_k(z|x) = \mathcal{N}(z; H_k x, R_k), \tag{2.45}$$

where $\mathcal{N}(\cdot; m, P)$ denotes a Gaussian density with mean m and covariance P, F_{k-1} is the state transition matrix, Q_{k-1} is the process noise covariance, H_k is the observation mapping matrix and R_k is the observation error covariance.

2.3.1.1 Prediction

Recall that the CPHD filter *classically* accounts for newly appearing targets with only a birth model, hence, the birth RFS is assumed a Gaussian mixture with form [96]

$$\mu_{\mathrm{B},k}(x) = \sum_{i=1}^{J_{\mathrm{B},k}} \omega_{\mathrm{B},k}^{(i)} \mathcal{N}\Big(x; m_{\mathrm{B},k}^{(i)}, P_{\mathrm{B},k}^{(i)}\Big), \tag{2.46}$$

where $\omega_{\mathrm{B},k}^{(i)}, m_{\mathrm{B},k}^{(i)},$ and $P_{\mathrm{B},k}^{(i)}$ are the weights, means and covariances of the mixture birth intensity, respectively, and where $J_{\mathrm{B},k}$ denotes the number of birth mixture components at time k. Assuming that the posterior intensity μ_{k-1} at time $k-1$ is a Gaussian mixture given by

$$\mu_{k-1}(x) = \sum_{i=1}^{J_{k-1}} \omega_{k-1}^{(i)} \mathcal{N}\Big(x; m_{k-1}^{(i)}, P_{k-1}^{(i)}\Big), \tag{2.47}$$

then the predicted intensity $\mu_{k|k-1}$ is also a Gaussian mixture of the form

$$\mu_{k|k-1}(x) = \mu_{\mathrm{S},k|k-1}(x) + \mu_{\mathrm{B},k}(x), \tag{2.48}$$

where

$$\mu_{\mathrm{S},k|k-1}(x) = p_{\mathrm{S},k} \sum_{j=1}^{J_{k-1}} \omega_{k-1}^{(j)} \mathcal{N}\Big(x; m_{\mathrm{S},k|k-1}^{(j)}, P_{\mathrm{S},k|k-1}^{(j)}\Big), \tag{2.49}$$

$$m_{\mathrm{S},k|k-1}^{(j)} = F_{k-1} m_{k-1}^{(j)}, \tag{2.50}$$

$$P_{\mathrm{S},k|k-1}^{(j)} = Q_{k-1} + F_{k-1} P_{k-1}^{(j)} F_{k-1}^{T}. \tag{2.51}$$

Given a posterior cardinality distribution ρ_{k-1} the predicted cardinality distribution is given by

$$\rho_{k|k-1}(n) = \sum_{j=0}^{n} \rho_{\mathrm{B},k}(n-j) \sum_{\ell=j}^{\infty} C_j^{\ell} \rho_{k-1}(\ell) p_{\mathrm{S},k}^{j} (1-p_{\mathrm{S},k})^{\ell-j}, \tag{2.52}$$

where $\rho_{\mathrm{B},k}(n-j)$ and C_j^{ℓ} are defined in Section 2.1.3.2.

2.3.1.2 Update

The presented research includes changes to the GM-CPHD prediction to account for spawn models, but no changes were required for the GM-CPHD update. Nevertheless, for completeness,

the equations are presented here. Assuming at time k the predicted intensity $\mu_{k|k-1}$ and predicted cardinality distribution $\rho_{k|k-1}$ are given and that the predicted intensity is a Gaussian mixture given by

$$\mu_{k|k-1}(x) = \sum_{i=1}^{J_{k|k-1}} \omega_{k|k-1}^{(i)} \mathcal{N}\left(x; m_{k|k-1}^{(i)}, P_{k|k-1}^{(i)}\right), \tag{2.53}$$

then the updated intensity and cardinality distributions are given by

$$\mu_k = \frac{\langle \Psi_k^1 \left[\omega_{k|k-1}, Z_k\right], \rho_{k|k-1}\rangle}{\langle \Psi_k^0 \left[\omega_{k|k-1}, Z_k\right], \rho_{k|k-1}\rangle} (1 - p_{\mathrm{D},k})\, \mu_{k|k-1}(x) + \sum_{z \in Z_k} \sum_{j=1}^{J_{k|k-1}} \omega_k^{(j)}(z) \mathcal{N}\left(x; m_k^{(j)}, P_k^{(j)}\right), \tag{2.54}$$

$$\rho_k(n) = \frac{\Psi_k^0 \left[\omega_{k|k-1}, Z_k\right](n), \rho_{k|k-1}(n)}{\langle \Psi_k^0 \left[\omega_{k|k-1}, Z_k\right], \rho_{k|k-1}\rangle}, \tag{2.55}$$

where

$$\Psi_k^u \left[\omega, Z\right](n) = \sum_{j=0}^{\min(|Z|,n)} (|Z| - j)\, \rho_{K,k}(|Z| - j)\, P_{j+u}^n \frac{(1 - p_{\mathrm{D},k})^{n-(j+u)}}{\langle 1, \omega\rangle^{j+u}} e_j\big(\Lambda_k(\omega, Z)\big), \tag{2.56}$$

$$\Lambda(\omega, Z) = \left\{ \frac{\langle 1, \kappa_k\rangle}{\kappa_k(z)} p_{\mathrm{D},k} \omega^T q_k(z) : z \in Z \right\}, \tag{2.57}$$

$$\omega_{k|k-1} = \left[\omega_{k|k-1}^{(1)}, \ldots, \omega_{k|k-1}^{(J_{k|k-1})}\right]^T, \tag{2.58}$$

$$q_k(z) = \left[q_k^{(1)}(z), \ldots, q_k^{(J_{k|k-1})}(z)\right]^T, \tag{2.59}$$

$$q_k^{(j)}(z) = \mathcal{N}\left(z; \eta_{k|k-1}^{(j)}, S_{k|k-1}^{(j)}\right), \tag{2.60}$$

$$\eta_{k|k-1}^{(j)} = H_k m_{k|k-1}^{(j)}, \tag{2.61}$$

$$S_{k|k-1}^{(j)} = H_k P_{k|k-1}^{(j)} H_k^T + R_k, \tag{2.62}$$

$$\omega_k^{(j)}(z) = p_{\mathrm{D},k} \omega_{k|k-1}^{(j)} q_k^{(j)}(z) \frac{\langle \Psi_k^1 \left[\omega_{k|k-1}, Z_k \setminus \{z\}\right], \rho_{k|k-1}\rangle}{\langle \Psi_k^0 \left[\omega_{k|k-1}, Z_k\right], \rho_{k|k-1}\rangle} \frac{\langle 1, \kappa_k\rangle}{\kappa_k(z)}, \tag{2.63}$$

$$m_k^{(j)}(z) = m_{k|k-1}^{(j)} + K_k^{(j)}\left(z - \eta_{k|k-1}^{(j)}\right), \tag{2.64}$$

$$P_k^{(j)}(z) = \left[I - K_k^{(j)} H_k\right] P_{k|k-1}^{(j)}, \tag{2.65}$$

$$K_k^{(j)} = P_{k|k-1}^{(j)} H_k^T \left[S_{k|k-1}^{(j)}\right]^{-1}, \tag{2.66}$$

$$\tag{2.67}$$

and where $\rho_{K,k}, P_j^n,$ and $e_j(\Lambda)$ are defined in Section 2.1.3.2. See [96] for details on implementing elementary symmetric functions and circumventing infinitely tailed cardinality distributions (the summation to ∞ in Eq. (2.52)).

2.3.2 GM-GLMB

The content of this section closely follows that in [94].

This section presents the GM implementation of the GLMB filter, which is fairly straightforward, as it is only a matter of substituting Gaussian distributions into (2.30) and (2.37), then carrying out simplifications. However, it is instructive to present the steps here as they will support simplifications performed in the presented work. First, note [89, Lemmas 1 and 2], transcribed here for convenience.

Lemma 1. *Given F, d, Q, m, and P of appropriate dimensions and that Q and P are positive definite [89]*

$$\int \mathcal{N}(x; F\zeta + d, Q)\mathcal{N}(\zeta; m, P)\mathrm{d}\zeta = \mathcal{N}(x; Fm + d, Q + FPF^T). \tag{2.68}$$

Lemma 2. *Given H, R, m, and P of appropriate dimensions and that R and P are positive definite*

$$\mathcal{N}(z; Hx, R)\mathcal{N}(x; m, p) = q(z)\mathcal{N}(x; \tilde{m}, \tilde{P}), \tag{2.69}$$

where

$$q(z) = \mathcal{N}(z; Hm, R + HPH^T), \tag{2.70}$$

$$\tilde{m} = m + K(z - Hm), \tag{2.71}$$

$$\tilde{P} = (I - KH)P, \tag{2.72}$$

$$K = PH^T(HPH^T + R)^{-1}. \tag{2.73}$$

2.3.2.1 Prediction

Assume each single-object probability density $p^{(\xi)}(x,\ell)$ in (2.21) is a Gaussian mixture with form

$$p^{(\xi)}(x,\ell) = \sum_{i=1}^{J^{(\xi)}(\ell)} w^{(\xi,i)}(\ell)\mathcal{N}(x; m^{(\xi,i)}(\ell), P^{(\xi,i)}(\ell)), \tag{2.74}$$

and let

$$w_{\mathrm{B},+}(L) = 1_{\mathbb{B}_+}(L)\left[1-r_{\mathrm{B},+}\right]^{\mathbb{B}_+-L}\left[r_{\mathrm{B},+}\right]^{L}, \tag{2.75}$$

$$p_{\mathrm{B},+}(x,\ell) = \sum_{j=1}^{J_{\mathrm{B},+}(\ell)} w_{\mathrm{B},+}^{(j)}(\ell)\mathcal{N}(x; m_{\mathrm{B},+}^{(j)}(\ell), P_{\mathrm{B},+}^{(j)}(\ell)), \tag{2.76}$$

$$f_{\mathrm{S},+}(y|x,\ell) = \mathcal{N}(y; Fx(\ell), Q). \tag{2.77}$$

First, note that with a constant probability of survival and Lemma 1, that (2.34) simplifies to $\eta_{\mathrm{S},+}^{(\xi)}(\ell) = p_{\mathrm{S},+}$. Similarly, (2.36) simplifies to $q_{\mathrm{S},+}^{(\xi)}(\ell) = q_{\mathrm{S},+}$. Then, by substituting in terms, (2.31) becomes

$$w_+^{(I_+,\xi)} = 1_{\mathbb{B}_+}(L-\mathbb{L})\left[1-r_{\mathrm{B},+}\right]^{\mathbb{B}_+-(L-\mathbb{L})}\left[r_{\mathrm{B},+}\right]^{L-\mathbb{L}} p_{\mathrm{S},+}^{|L\cap\mathbb{L}|}\sum_{I\subseteq\mathbb{L}} 1_I(L\cap\mathbb{L}) q_{\mathrm{S},+}^{|I-(L\cap\mathbb{L})|} w^{(I,\xi)}(I). \tag{2.78}$$

Now, (2.33) becomes

$$p_{\mathrm{S},+}^{(\xi)}(x_+,\ell_+) = \sum_{i=1}^{J^{(\xi)}(\ell)} w^{(\xi,i)}(\ell)\mathcal{N}(x_+; Fm^{(\xi,i)}(\ell), Q+FP^{(\xi,i)}(\ell)F^T) \tag{2.79}$$

via Lemma 1.

2.3.2.2 Update

With constant probability of detection, note that (2.40) simplifies to

$$\eta^{(\xi,\theta)}(\ell|Z) = \begin{cases} q_{\mathrm{D}} & \text{if } \theta(\ell)=0, \\ \frac{p_{\mathrm{D}}}{\kappa(z_{\theta(\ell)})} & \text{otherwise,} \end{cases} \tag{2.80}$$

and, via Lemma 2, (2.39) becomes

$$p^{(\xi,\theta)}(x,\ell|Z) = \begin{cases} p^{(\xi)}(x,\ell) & \text{if } \theta(\ell)=0, \\ q(z_{\theta(\ell)})\mathcal{N}(z_{\theta(\ell)}; Hx(\ell), R) & \text{otherwise,} \end{cases} \tag{2.81}$$

where

$$q(z_{\theta(\ell)}) = \mathcal{N}(z; Hm(\ell), R + HP(\ell)H^T), \tag{2.82}$$

$$\tilde{m}(\ell) = m(\ell) + K(z_{\theta(\ell)} - Hm(\ell)), \tag{2.83}$$

$$\tilde{P}(\ell) = (I - KH)P(\ell), \tag{2.84}$$

$$K = P(\ell)H^T(HP(\ell)H^T + R)^{-1}. \tag{2.85}$$

2.4 Tractability

2.4.1 Gaussian Mixture Components

Both the CPHD and GLMB filter implementations discussed in this paper employ pruning methods to combat combinatorial increase of Gaussian mixture components. They include truncation methods based on component weight and maximum component number thresholds and component merging. In this work, merging is performed on the basis of Mahalanobis distance. For more details on these methods, see [89].

2.4.2 GLMB Hypothesis Components

The content of this section closely follows that in [92].

The first GLMB filter implementation consists of prediction and update stages, each requiring independent truncations of GLMB densities [94, 95]. Alternatively, a substantially more efficient implementation of the GLMB filter [92], hereafter referred to as the *fast* GLMB implementation, employs a single joint prediction/update stage requiring only one truncation procedure. This work employs the fast GLMB implementation, thus for convenience, we introduce pertinent expressions and conventions for GLMB joint prediction/update and formulation of the GLMB truncation problem originally presented in [92]. We expand on this material in Section 6.4 to incorporate spawning.

Given the GLMB filtering density (2.21) at the current time, the GLMB filtering density at

the next time is given by [92][2]

$$\boldsymbol{\pi}_+(\mathbf{X}_+) \propto \Delta(\mathbf{X}_+) \sum_{I,\xi,I_+,\theta_+} w^{(I,\xi)} w^{(I,\xi,I_+,\theta_+)}(Z_+) \delta_{I_+}(\mathcal{L}(\mathbf{X}_+)) \left[p_+^{(\xi,\theta_+)}(\cdot|Z_+) \right]^{\mathbf{X}_+}, \tag{2.86}$$

where $I \in \mathcal{F}(\mathbb{L})$, $\xi \in \Xi$, $I_+ \in \mathcal{F}(\mathbb{L}_+)$, $\theta_+ \in \Theta_+(I_+)$, and

$$w^{(I,\xi,I_+,\theta_+)}(Z_+) = \left[1 - \bar{p}_{\mathrm{S}}^{(\xi)}\right]^{I-I_+} \left[\bar{p}_{\mathrm{S}}^{(\xi)}\right]^{I\cap I_+} [1 - r_{\mathrm{B},+}]^{\mathbb{B}_+ - I_+} [r_{\mathrm{B},+}]^{\mathbb{B}_+ \cap I_+} \left[\bar{\psi}_+^{(\xi,\theta_+)}(\cdot|Z_+)\right]^{I_+}, \tag{2.87}$$

$$\bar{p}_{\mathrm{S}}^{(\xi)}(\ell) = \left\langle p_{\mathrm{S}}(\cdot,\ell), p^{(\xi)}(\cdot,\ell) \right\rangle, \tag{2.88}$$

$$\bar{\psi}_+^{(\xi,\theta_+)}(\ell_+|Z_+) = \left\langle \bar{p}_+^{(\xi)}(\cdot,\ell_+), \psi_+^{(\theta_+(\ell_+))}(\cdot,\ell_+|Z_+) \right\rangle, \tag{2.89}$$

$$p_+^{(\xi,\theta_+)}(x_+,\ell_+|Z_+) = \frac{\bar{p}_+^{(\xi)}(x_+,\ell_+)\psi_+^{(\theta_+(\ell_+))}(x_+,\ell_+|Z_+)}{\bar{\psi}_+^{(\xi,\theta_+)}(\ell_+|Z_+)}, \tag{2.90}$$

$$\bar{p}_+^{(\xi)}(x_+,\ell_+) = 1_{\mathbb{B}_+}(\ell_+) p_{\mathrm{B},+}(x_+,\ell_+) + 1_{\mathbb{L}}(\ell_+) \frac{\left\langle p_{\mathrm{S}}(\cdot,\ell_+) f_{\mathrm{S},+}(x_+|\cdot,\ell_+), p^{(\xi)}(\cdot,\ell_+) \right\rangle}{\bar{p}_{\mathrm{S}}^{(\xi)}(\ell_+)}. \tag{2.91}$$

Though (2.86) is not strictly GLMB, it does take on GLMB form when rewritten as a sum over I_+, ξ, θ_+ with weights [92]

$$w_+^{(I_+,\xi,\theta_+)}(Z_+) \propto \sum_I w^{(I,\xi)} w^{(I,\xi,I_+,\theta_+)}(Z_+). \tag{2.92}$$

Efficient implementation of the GLMB recursion (2.86) is achieved by propagating only the components with significant $w^{(I,\xi,I_+,\theta_+)}(Z_+)$ through time, i.e., for each component (I,ξ) from the GLMB density at the current time and a multi-object observation Z_+ at the next time, the set of pairs $(I_+,\theta_+) \in \mathcal{F}(\mathbb{L}) \times \Theta_+(I_+)$ with significant $w^{(I,\xi,I_+,\theta_+)}(Z_+)$ are retained while the rest are discarded. The truncation procedure is described as follows.

Consider a fixed component (I,ξ), and enumerate $Z_+ = \{z_{1:|Z_+|}\}$, $\mathbb{B}_+ = \{\ell_{1:K}\}$, and $I = \{\ell_{K+1:P}\}$. For each pair $(I_+,\theta_+) \in \mathcal{F}(\mathbb{L}) \times \Theta_+(I_+)$, an equivalent P-dimensional vector representation $\gamma = (\gamma_{1:P}) \in \{-1 : |Z_+|\}^P$ is defined as

$$\gamma_i = \begin{cases} \theta_+(\ell_i), & \text{if } \ell_i \in I_+, \\ -1, & \text{otherwise.} \end{cases} \tag{2.93}$$

[2] In the interest of simplifying notation, note that $\sum_{(I,\xi)\in\mathcal{F}(\mathbb{L})\times\Xi} a^{(I,\xi)} = \sum_{I,\xi} a^{(I,\xi)}$ when the definitions $I \in \mathcal{F}(\mathbb{L})$ and $\xi \in \Xi$ are provided.

Note that γ inherits the positive 1-1 property from θ_+, and that I_+ and $\theta_+ : I_+ \to \{0 : |Z_+|\}$ can be recovered by

$$I_+ = \{\ell_i \in \mathbb{B}_+ \cup\ I : \gamma_i \geq 0\}, \qquad \theta_+(\ell_i) = \gamma_i. \tag{2.94}$$

Assuming that, for all $i \in \{1 : P\}$, $\bar{p}_{\mathrm{S}}^{(\xi)}(\ell_i) \in (0,1)$ and $\bar{p}_{\mathrm{D},+}^{(\xi)}(\ell_i) \triangleq \left\langle p_{\mathrm{D},+}(\cdot,\ell_i), \bar{p}_+^{(\xi)}(\cdot,\ell_i) \right\rangle \in (0,1)$, we define for each $j \in \{-1 : |Z_+|\}$

$$\eta_i(j) = \begin{cases} 1 - r_{\mathrm{B},+}(\ell_i), & 1 \leq i \leq K, j < 0, \\ r_{\mathrm{B},+}(\ell_i)\bar{\psi}_+^{(\xi,j)}(\ell_i|Z_+), & 1 \leq i \leq K, j \geq 0, \\ 1 - \bar{p}_{\mathrm{S}}^{(\xi)}(\ell_i), & K+1 \leq i \leq P, j < 0, \\ \bar{p}_{\mathrm{S}}^{(\xi)}(\ell_i)\bar{\psi}_+^{(\xi,j)}(\ell_i|Z_+), & K+1 \leq i \leq P, j \geq 0, \end{cases} \tag{2.95}$$

where $\bar{\psi}_+^{(\xi,j)}(\ell_i|Z_+) = \left\langle \bar{p}_+^{(\xi)}(\cdot,\ell_i), \psi_+^{(j)}(\cdot,\ell_i|Z_+) \right\rangle$. Then $w^{(I,\xi,I_+,\theta_+)}(Z_+) = \prod_{i=1}^{P} \eta_i(\gamma_i)$, if the positive 1-1 vectors γ are equivalent representations of (I_+,θ_+). Hence generating significant GLMB children components of (I,ξ) translates to generating positive 1-1 vectors with significant weights [92]. Methods for obtaining a set of positive 1-1 vectors include:

- solving a ranked assignment problem using Murty's algorithm [68], which finds the N best 1-1 vectors in non-increasing order; and
- a more efficient method using the Gibbs sampler to simulate an unordered set of significant positive 1-1 vectors [92].

2.5 Metrics

The content of this section was presented in [14].

To compare the multi-target state representing the true targets in a scene – the "ground truth" – and a collection of targets extracted from a filter's output, the OSPA metric [75] is used for assessing the accuracy of multi-object filters. Given two sets $X = \{x_1, \ldots, x_m\}$, $x_i \in \mathbb{X}$, $1 \leq i \leq m$, and $Y = \{y_1, \ldots, y_n\}$, $y_j \in \mathbb{X}$, $1 \leq j \leq n$, the second-order OSPA distance $d_2^{(c)}(X,Y)$

between X and Y is defined as

$$d_2^{(c)}(X,Y) = \begin{cases} 0, & m = n = 0, \\ \left[\frac{1}{n}\left(\min_{\pi\in\Pi_n}\sum_{i=1}^{m} d^{(c)}(x_i, y_{\pi(i)})^2 + c^2(n-m)\right)\right]^{1/2}, & m \leq n, \\ d_2^{(c)}(Y,X), & \text{otherwise}, \end{cases} \tag{2.96}$$

with

$$d^{(c)}(x_i, y_j) = \min(c, ||x_i - y_j||), \tag{2.97}$$

where c is the cutoff parameter, and $||\cdot||$ is the usual norm on $\mathbb{X}$. The OSPA distance is such that $0 \leq d_2^{(c)}(X,Y) \leq c$; $d_2^{(c)}(X,Y) = 0$ indicates that X and Y are identical, while $d_2^{(c)}(X,Y)$ increases with the discrepancies between X and Y, taking into account mismatches in number of elements and element states.

The Hellinger distance [36] is used to compare the true number of targets in a scene and a estimated cardinality distribution extracted from the filter's output. Given two finite cardinality distributions $P = (p_1, \ldots, p_k)$ and $Q = (q_1, \ldots, q_k)$, the Hellinger distance $d_H(P,Q)$ is

$$d_H(P,Q) = \frac{1}{\sqrt{2}}\sqrt{\sum_{i=1}^{k}(\sqrt{p_i} - \sqrt{q_i})^2}. \tag{2.98}$$

Note that in (2.98), the coefficient $1/\sqrt{2}$ is included in order to scale the Hellinger distance such that it is bounded as $0 \leq d_H(P,Q) \leq 1$; $d_H(P,Q) = 0$ indicates that P and Q are equivalent, where as $d_H(P,Q) \to 1$, P and Q become increasingly dissimilar.

2.6 Orbit Problem Overview

An object's orbit can be characterized by its six Keplerian Elements, which are described in Table 2.2 and some of which are illustrated in Fig. 2.2.

Table 2.2: Keplerian Elements Description

Element	Description
Semimajor Axis	A geometric quantity that describes the size of a particular orbit
Eccentricity	Determines the shape of an orbit, e.g., circular, elliptical, etc.
Inclination	Angle formed between the orbital and equatorial planes, denoted as i in Fig. 2.2.
Right Ascension of the Ascending Node	Positive angle Ω in the equatorial plane measured from the X unit vector to the ascending node in Fig. 2.2.
Argument of Perigee	Angle ω in the orbital plane measured from the ascending node that locates the point of the orbit that is closest to Earth.
True Anomaly	Angle ν that determines a satellite's position relative to perigee.

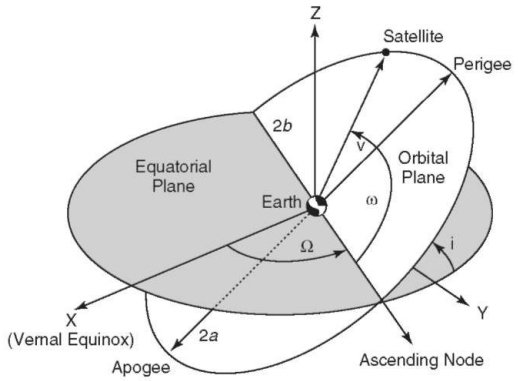


Figure 2.2: Keplerian elements and the orbital plane[3].

Using the Keplerian elements, a satellite's position $\mathbf{r}$ and velocity $\dot{\mathbf{r}}$ vectors can be computed for a given reference frame. The differential equations of motion for a perturbed satellite are given by [86]

$$\ddot{\mathbf{r}} = -\frac{\mu \mathbf{r}}{r^3} + \mathbf{a}, \tag{2.99}$$

where $\ddot{\mathbf{r}}$ is the satellite's acceleration vector, μ is a gravitational constant, r denotes the norm of $\mathbf{r}$, and $\mathbf{a}$ represents perturbing accelerations. There are many sources of perturbations; however, only

[3] Lh4.ggpht.com. tmp12402.jpg (image). [online] Available at: http://lh4.ggpht.com/_1wtadqGaaPs/TF5SS7_qEdI/AAAAAAAAOls/MCGLdsknqdQ/s1600-h/tmp12402.jpg [Accessed 18 Dec. 2015].

accelerations that account for non-spherical Earth $\mathbf{a}_{NS}$ and atmospheric drag $\mathbf{a}_{drag}$ perturbations are considered here, i.e.,

$$\mathbf{a} = \mathbf{a}_{NS} + \mathbf{a}_{drag}. \tag{2.100}$$

Perturbations due to a non-spherical Earth can be modeled by the use of the geopotential expansion given by [88]

$$U = \frac{\mu}{r}\left[1 + \sum_{\ell=2}^{\infty}\sum_{m=0}^{\ell}\left(\frac{R_\oplus}{r}\right)^{\ell} P_{\ell,m}(\sin(\phi))(C_{\ell m}\cos(m\lambda) + S_{\ell m}\sin(m\lambda))\right] \tag{2.101}$$

where ϕ is the geocentric latitude, λ is the longitude, $R_\oplus$ is Earth's mean equatorial radius, P_n^m are *associated Legendre polynomials* of degree n and order m, and where C_{nm} and S_{nm} are empirically derived coefficients that can be found in a table. Other forms of Eq. 2.101 are sometimes used that include the "J" notation for zonal harmonics.[4] The work presented here only considers perturbations due to the J_2, $(\ell = 2, m = 0)$, and J_3, $(\ell = 3, m = 0)$, zonal harmonics.

The perturbing acceleration $\mathbf{a}_{NS}$ is given by [86]

$$\mathbf{a}_{NS} = T_{ns} \cdot \nabla U, \tag{2.102}$$

where ∇ denotes the gradient operator and T_{ns} denotes the appropriate coordinate system transformation matrix required to rotate components (u_r, u_ϕ, u_λ) in spherical coordinates, to components (x, y, z) in inertial frame coordinates.

Atmospheric drag perturbation acceleration $\mathbf{a}_{drag}$ can be modeled by [86]

$$\mathbf{a}_{drag} = -\frac{1}{2}\rho\left(C_D\frac{A}{m}\right)V_r\mathbf{V}_r, \tag{2.103}$$

where ρ is the atmospheric density, C_D is the drag coefficient, A/m is the area-to-mass ratio, and $\mathbf{V}_r$ is the satellite velocity relative to the atmosphere.

2.7 Uncertainty Propagation

An issue commonly encountered in orbit propagation is that of covariance degradation due to non-linearities [22]. The linear Gaussian assumptions above promote tractable real-time tracking,

[4] "Zonal harmonics are defined by the zeroth order ($m = 0$), where the dependencies of the potential on longitude vanishes and the field is symmetrical about the polar axis." [88]

but this conflicts with the nature of RSO tracking due to inevitable observation gaps which translate into high integration times and non-linear single-target densities [49].

To address this issue, the presented work uses unscented Kalman filter implementations of the proposed filters. Details for the unscented Kalman filter (UKF) for single-target tracking can be found in [51, 52] while details for UKF implementation of a PHD filter can be found in [89]. To avoid numerical issues, the square-root formulation of the UKF (SR-UKF) is used. Appropriate modifications to the UKF implementation to PHD and GLMB filtering are made.

Another tool used to address uncertainty propagation issues is the adaptive entropy-based Gaussian-mixture information synthesis method (AEGIS) [26]. Given a GM representation of a p.d.f., this method monitors non-linearity during the propagation of the p.d.f. and through entropy-based detection, can increase the number of Gaussian mixture components used to represent the p.d.f. online by splitting a Gaussian distribution into a representative Gaussian mixture so as to reduce the effects of non-linearity on any given component. AEGIS is seamlessly integrated with GM filter implementations due to the GM nature of its formulation.

AEGIS detects non-linearity via the deviation between linearized and non-linearized based entropy measures. Once the deviation surpasses some pre-defined threshold, propagation is stopped and the process of splitting Gaussian distributions is carried out. A multi-variate distribution is split along the columns of the square-root factor of its covariance in accordance to pre-defined splitting library parameters: they determine the number of GM components used to represent the component being split. A square-root implementation of AEGIS is used in the preliminary research and is achievable by splitting distributions along the columns of the Cholesky decomposition of their covariance, thereby avoiding the square-root operation.

2.8 Detection Probability Modeling

Due to the GM approximation, assumption A.2.8 in Section 2.3 introduces a constraint that is problematic for RSO tracking applications, namely a probability of detection $p_{\mathrm{D},k}$ that is constant across the entire single-target state space for a given time k. There are several considerations to be

made for accurate modeling of $p_{\mathrm{D},k}$ [49]; however, of concern to the presented work is the tendency for track loss due to observation gaps and GM component *down-weighting*: in the case of the GM-CPHD filter, this is illustrated by (2.54) in Section 2.3.1. If an RSO is not detected, then over a period of time its representative GM component weight in the second term of (2.54) is reduced and eventually discarded in the pruning process. Its component weight in the first term may increase over time, though its state is never corrected by measurement updates, so when the object is observable again, its state is poorly estimated and its components are potentially discarded.

The presented work uses the approach in [34] to address this issue by augmenting the probability of detection using an indicator function. Given a constant probability of detection $p_{\mathrm{D},filter}$ defined for a given filter implementation, the probability of detection $p_{\mathrm{D},k}$ is given by

$$p_{\mathrm{D},k} = p_{\mathrm{D},FOV}(z_k) p_{\mathrm{D},filter}, \tag{2.104}$$

where

$$p_{\mathrm{D},FOV}(z_k) = \begin{cases} 1, & \text{if } z_k \in \text{ FOV}, \\ 0, & \text{otherwise.} \end{cases} \tag{2.105}$$

However, as noted in [35], this method's performance potentially degrades when an object is near the edge of the FOV.

2.9 Breakup Modeling

RSO breakups are simulated using the NASA standard breakup model (NSBM) [45]. Only rocket body (RB) explosions are simulated in the presented work when fragmentation events are considered, hence, only NSBM components relating to RB explosions are presented here for the sake of brevity. See [45] and [54] for more details on satellite collision modeling.

Break up model steps are depicted in Fig. 2.3 where N_d denotes the cumulative number of fragments with diameter d greater than characteristic length l_c, A, m, and A/m denote area, mass and area-to-mass ratio, respectively, and where Δv denotes change in velocity.

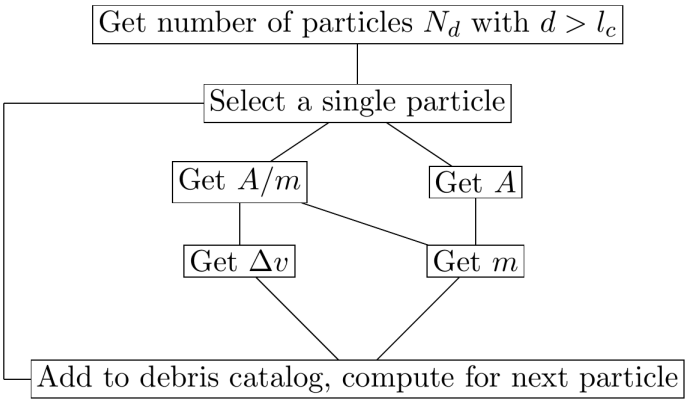


Figure 2.3: NSBM flow chart. Adapted from [54].

The cumulative number of explosion particles (fragments) is determined by [54]

$$N_d(d \geq l_c) = 6 c_s l_c^{-1.6}, \tag{2.106}$$

where c_s is an empirical calibration factor used to account for space catalog incompleteness and is given by

$$c_s = c_{TLE} = 10^{0.5 \exp\left(-2.464\left(\log_{10}\left(\frac{d_{SSN}}{[\mathrm{m}]}\right)+1.22\right)^2\right)}, \tag{2.107}$$

where

$$d_{SSN}/[\mathrm{m}] = \begin{cases} 0.089, & \text{for } H \leq 620\,\mathrm{km}, \\ 10^{-2.737+0.604\log_{10}\left(\frac{H}{[\mathrm{km}]}\right)}, & \text{for } 620\ \mathrm{km} < H \leq 1300\ \mathrm{km}, \\ 10^{-6.517+1.819\log_{10}\left(\frac{H}{[\mathrm{km}]}\right)}, & \text{for } 1300\ \mathrm{km} < H \leq 3800\ \mathrm{km}, \\ 1.0, & \text{for } H > 3800\ \mathrm{km}, \end{cases} \tag{2.108}$$

where H denotes orbital altitude.

Noting that $\lambda_c(l_c) = \log_{10}(l_c)$, $\chi = \log_{10}(A/m)$, and $\mathcal{N}$ = denotes the normal distribution function $\mathcal{N}(\chi; \mu, \sigma) = \left[1/(\sigma(2\pi)^{0.5})\right] e^{-(\chi-\mu)^2/2\sigma^2}$, area-to-mass ratios are assigned to RB fragments

with $l_c > 11\,\text{cm}$ using the bi-modal distribution[5]

$$D_{R/B}^{(A/m)}(\lambda_c, \chi) = \alpha_{R/B}(\lambda_c)\mathcal{N}\Big(\chi; \mu_{R/B}^{(1)}(\lambda_c), \sigma_{R/B}^{(1)}(\lambda_c)\Big) + \big(1 - \alpha_{R/B}(\lambda_c)\big)\mathcal{N}\Big(\chi; \mu_{R/B}^{(2)}(\lambda_c), \sigma_{R/B}^{(2)}(\lambda_c)\Big), \tag{2.109}$$

where

$$\alpha_{R/B}(\lambda_c) = \begin{cases} 1, & \lambda_c \leq -1.4, \\ 1 - 0.3571(\lambda_c + 1.4), & -1.4 < \lambda_c < 0, \\ 0.5, & \lambda_c \geq 0, \end{cases} \tag{2.110}$$

$$\mu_{R/B}^{(1)}(\lambda_c) = \begin{cases} -0.45, & \lambda_c \leq -0.5, \\ -0.45 - 0.9(\lambda_c + 0.5), & -0.5 < \lambda_c < 0, \\ -0.9, & \lambda_c \geq 0, \end{cases} \tag{2.111}$$

$$\sigma_{R/B}^{(1)} = 0.55, \tag{2.112}$$

$$\mu_{R/B}^{(2)} = -0.9, \tag{2.113}$$

$$\sigma_{R/B}^{(2)}(\lambda_c) = \begin{cases} 0.28, & \lambda_c \leq -1.0, \\ 0.28 - 0.1636(\lambda_c + 1), & -1.0 < \lambda_c < 0.1, \\ 0.1, & \lambda_c \geq 0.1. \end{cases} \tag{2.114}$$

The inverse cumulative distribution function is used to solve for values of A/m.

Similarly, explosion fragments are assigned Δv's via the distribution

$$D_{EXP}^{(\Delta v)}(\chi, \nu) = \mathcal{N}\Big(\nu; \mu_{EXP}(\chi), \sigma_{EXP}(\chi)\Big), \tag{2.115}$$

where $\chi = \log_{10}(A/m)$, $\nu = \log_{10}(\Delta v)$, $\mu_{EXP} = 0.2\chi + 1.85$ and $\sigma_{EXP} = 0.4$. Though the NSBM can be used to simulate Δv magnitudes, it does not account for the direction in which they are applied, thus directions are generated by randomly selecting points on a unit sphere.

[5] The presented work only considers $l_c > 11\,\text{cm}$. For distributions with $l_c \leq 11\,\text{cm}$ see [45].

Chapter 3

Measure-Theoretic Background

Some of the advancements presented in this work are developed within a more generalized measure-theoretic framework, as described in Chapter 1. Therefore, this chapter introduces the necessary background on point processes (Section 3.1), probability generating functionals (p.g.fl.s) (Section 3.2), functional differentiation (Section 3.3), and a few properties from the application of differentiation in the context of point processes (Section 3.4). The content of this chapter, excluding the derivations at the end, was presented in [14].

3.1 Point processes

A point process on some space $\mathbb{X}$ is a random variable whose number of elements and element states, belonging to $\mathbb{X}$, are random. In the context of multi-object filtering, the population of objects is represented by a point process Ω, on a single-object state space $\mathbb{X} \subseteq \mathbb{R}^n$, whose elements describe individual object states. A realization of Ω is a set of points $X = (x_1, \ldots, x_N)$ depicting a specific multi-object configuration, where $x_i \in \mathbb{X}$ describes the n-component state of an individual object (position, velocity, etc.).

Fundamental to probabilistic characterizations of a point process is the measurable space $(\mathcal{X}, \mathcal{B}_{\mathcal{X}})$, where $\mathcal{X} = \bigcup_{n \geq 0} \mathbb{X}^n$ is the point process state space, i.e., the space of all the finite sets of points in $\mathbb{X}$, and $\mathcal{B}_{\mathcal{X}}$ is the Borel σ-algebra on $\mathcal{X}$ [82]. Think of $\mathcal{X}$ as analogous to $\mathcal{F}(\mathbb{X})$, the collection of finite subsets of $\mathbb{X}$, from Section 2.1.1. A σ-algebra is a system, or collection, of sets that are closed under basic set operations that include union, intersection, and complementation,

e.g., if A_1 and A_2 are disjoint subsets of A such that $A = A_1 \cup A_2$ and given a set function f, then $f(A) = f(A_1 \cup A_2) = f(A_1) + f(A_2)$ [82]. The Borel σ-algebra on $\mathcal{X}$ is the σ-algebra generated by the open sets of $\mathcal{X}$ [42]. For more on measurable spaces and σ-algebras, see [25, 42, 82].

A point process Ω is characterized by its probability distribution P_Ω on the measurable space $(\mathcal{X}, \mathcal{B}_\mathcal{X})$. The probability distribution of a point process is defined as a symmetric function, so that the order of points in a realization is irrelevant for statistical purposes – for example, realizations (x_1, x_2) and (x_2, x_1) are equally probable. In addition, if the probability distribution is such that the realizations are sets of points that are pairwise distinct almost surely, then the point process is called simple. For the presented research, all considered point processes are assumed simple.

The probability distribution P_Ω is characterized by its projection measures $P_\Omega^{(n)}$, for any $n \geq 0$. The n^{th}-order projection measure $P_\Omega^{(n)}$, for any $n \geq 1$, is defined on the Borel σ-algebra of $\mathbb{X}^n$ and gives the probability for the point process to be composed of n points, and the probability distribution of these points. By extension, $P_\Omega^{(0)}$ is the probability for the point process to be empty. For any $n \geq 0$, $J_\Omega^{(n)}$ denotes the n^{th}-order Janossy measure [25, p. 124], and is defined as

$$J_\Omega^{(n)}(B_1 \times \ldots \times B_n) = \sum_{\sigma(n)} P_\Omega^{(n)}(B_{\sigma_1} \times \ldots \times B_{\sigma_n}) \tag{3.1a}$$

$$= n! P_\Omega^{(n)}(B_1 \times \ldots \times B_n), \tag{3.1b}$$

where B_i is in $\mathcal{B}_\mathbb{X}$, the Borel σ-algebra of $\mathbb{X}$, $1 \leq i \leq n$, and where $\sigma(n)$ denotes the set of all permutations $(\sigma_1, \ldots, \sigma_n)$ of $(1, \ldots, n)$. The probability density p_Ω, the n^{th}-order projection density $p_\Omega^{(n)}$, and the n^{th}-order Janossy density $j_\Omega^{(n)}$ are the Radon-Nikodym derivatives of the probability distribution P_Ω, the n^{th}-order projection measure $P_\Omega^{(n)}$, and the n^{th}-order Janossy measure $J_\Omega^{(n)}$, w.r.t. some reference measure.

All these quantities provide equivalent ways to describe the point process Ω; furthermore, note that a simple point process is equivalent to an RFS and a Janossy density is a multi-object probability density [64]. Accordingly, a measure-theoretical formulation of the multi-object filtering problem is in many aspects equivalent to Mahler's RFS formulation, but with different notation and terminology. However, a measure-theoretical formulation provides a more general framework

for constructing certain statistical properties on point processes that can be exploited for practical applications. This is the case for the mathematical developments presented in Chapter 4 and the reason for the material presented in this chapter.

Assuming that f is a non-negative measurable function on $\mathcal{X}$, then the integral of f w.r.t. to the measure P_Ω can be written in the following ways:

$$P_\Omega(f) = \int_{\mathcal{X}} f(X) P_\Omega(\mathrm{d}X) \tag{3.2a}$$

$$= \int_{\mathcal{X}} f(X) p_\Omega(X) \mathrm{d}X \tag{3.2b}$$

$$= \sum_{n\geq 0} \int_{\mathbb{X}^n} f(x_1, \ldots, x_n) P_\Omega^{(n)}(\mathrm{d}(x_1, \ldots, x_n)) \tag{3.2c}$$

$$= \sum_{n\geq 0} \int_{\mathbb{X}^n} f(x_1, \ldots, x_n) p_\Omega^{(n)}(x_1, \ldots, x_n) \mathrm{d}x_1 \ldots \mathrm{d}x_n \tag{3.2d}$$

$$= \sum_{n\geq 0} \frac{1}{n!} \int_{\mathbb{X}^n} f(x_1, \ldots, x_n) J_\Omega^{(n)}(\mathrm{d}(x_1, \ldots, x_n)) \tag{3.2e}$$

$$= \sum_{n\geq 0} \frac{1}{n!} \int_{\mathbb{X}^n} f(x_1, \ldots, x_n) j_\Omega^{(n)}(x_1, \ldots, x_n) \mathrm{d}x_1 \ldots \mathrm{d}x_n. \tag{3.2f}$$

Throughout this chapter and Chapter 4, the exploitation of the Janossy measures will be preferred, for they are convenient tools in the context of functional differentiation (see Section 3.3). For the sake of simplicity, domains of integration will be omitted when they refer to the full object state space $\mathbb{X}$.

The Janossy measures can also be used directly to exploit meaningful information on the point process Ω. For example, central to the presented work is the extraction of the cardinality distribution measure-theoretical formulation ρ_Ω of the point process, that describes the number of elements in the realizations of Ω (see Section 4.1):

Example 1 (Cardinality distribution)**.** *Consider the function f_n defined as*

$$f_n(X) = \begin{cases} 1, & |X| = n, \\ 0, & \textit{otherwise}, \end{cases} \tag{3.3}$$

where $|X|$ denotes the number of elements in the set X. The integral of f_n w.r.t. to P_Ω yields the probability $\rho_\Omega(n)$ that a realization X of the point process Ω has size n and we have, using Eq. (3.2)

(see [81, p.28]):

$$\rho_\Omega(n) = P_\Omega(f_n) \tag{3.4a}$$

$$= \int_{\mathbb{X}^n} P_\Omega^{(n)}(\mathrm{d}(x_1, \ldots, x_n)) \tag{3.4b}$$

$$= \frac{1}{n!} \int_{\mathbb{X}^n} J_\Omega^{(n)}(\mathrm{d}(x_1, \ldots, x_n)). \tag{3.4c}$$

The function ρ_Ω is called the cardinality distribution of the point process Ω. Note that, in the general case, the n^{th}-order projection measure $P_\Omega^{(n)}$ and the n^{th}-order Janossy measure $J_\Omega^{(n)}$ are not probability measures since their integrals over $\mathbb{X}^n$ yield $\rho_\Omega(n)$ and $n!\rho_\Omega(n)$, respectively.

3.2 Probability generating functionals

The p.g.fl. provides a useful characterization for point process theory [67] and is defined as follows.

Definition 3.1 (Probability generating functional [25])**.** *The probability generating functional G_Ω of a point process Ω on $\mathbb{X}$ can be written for any test function $h \in \mathcal{U}(\mathbb{X})$ as*[1]

$$G_\Omega(h) = \int_{\mathcal{X}} \Big[\prod_{x \in X} h(x) \Big] P_\Omega(\mathrm{d}X) \tag{3.5a}$$

$$= J_\Omega^{(0)} + \sum_{n \geq 1} \frac{1}{n!} \int_{\mathbb{X}^n} h(x_1) \ldots h(x_n) J_\Omega^{(n)}(\mathrm{d}(x_1, \ldots, x_n)). \tag{3.5b}$$

The p.g.fl. G_Ω fully characterizes the point process Ω, and is a very convenient tool for the extraction of statistical information on Ω through functional differentiation (see Section 3.3). From Eq. (3.5) we can immediately write

$$G_\Omega(0) = J_\Omega^{(0)} \ (= P_\Omega^{(0)}), \tag{3.6}$$

$$G_\Omega(1) = 1. \tag{3.7}$$

Operations on point processes (e.g., superposition of two populations) can be translated into operations on their corresponding p.g.fl.s. In the context of multi-object tracking, p.g.fl.s provide a

[1] $\mathcal{U}(\mathbb{X})$ is the space of bounded measurable functions u on $\mathbb{X}$ satisfying $||u||_\infty \leq 1$.

convenient description of the compound population (objects or measurements) resulting from an operation on elementary populations.

The superposition operation for point processes describes the union of two populations Ω_1, Ω_2 into a compound population $\Omega_1 \cup \Omega_2$, during which the information about the origin population of each individual is lost.

Proposition 1 (Superposition of independent processes [67]). *Let Ω_1 and Ω_2 be two independent point processes defined on the same space, with respective p.g.fl.s G_{Ω_1} and G_{Ω_2}. The p.g.fl. of the superposition process $\Omega_1 \cup \Omega_2$ is given by the product*

$$G_{\Omega_1 \cup \Omega_2}(h) = G_{\Omega_1}(h) G_{\Omega_2}(h). \tag{3.8}$$

The Galton-Watson recursion for point processes [67, 99] describes the evolution of each individual x from a parent population Ω_{S} into a population of spawned individuals, independently of the other parent individuals but following a common evolution model described by a process Ω_{e}. The resulting spawn population Ω_{T} is then the superposition of all the populations of spawn individuals.

Proposition 2 (The Galton-Watson recursion [99]). *Let $G_{\Omega_{\mathrm{S}}}$ be the p.g.fl. of a parent process Ω_{S} on $\mathbb{X}$, and let $G_{\Omega_{\mathrm{e}}}(\cdot|x)$ be the conditional p.g.fl. of an evolution process Ω_{e}, defined for every $x \in \mathbb{X}$. The p.g.fl. of the spawn process Ω_{T} is given by the composition*

$$G_{\Omega_{\mathrm{T}}}(h) = G_{\Omega_{\mathrm{S}}}\left(G_{\Omega_{\mathrm{e}}}(h|\cdot)\right). \tag{3.9}$$

3.3 Functional differentiation

To make use of functionals in the derivations presented in Chapter 4.1, we require the notion of differentials on functional spaces. We adopt a restricted form of the Gâteaux differential, known as the chain differential [10], so that a general chain rule can be determined [18,19]. Following this, we describe the general higher-order chain rule.

Definition 3.2 (Chain differential [10]). *Under the conditions detailed in [10], the function f on some set X has a chain differential $\delta f(x;\eta)$ at $x \in X$ in the direction η if, for any sequence $\eta_n \to \eta \in X$, and any sequence of real numbers $\theta_n \to 0$, it holds that*

$$\delta f(x;\eta) = \lim_{n\to\infty} \frac{1}{\theta_n} \left(f(x+\theta_n\eta_n) - f(x)\right). \tag{3.10}$$

The n^{th}-order chain differential can be defined recursively as

$$\delta^n f\left(x;\eta_1,\ldots,\eta_n\right) = \delta\left(\delta^{n-1} f\left(x;\eta_1,\ldots,\eta_{n-1}\right);\eta_n\right). \tag{3.11}$$

Applying n^{th}-order chain differentials on composite functions can be an extremely laborious process since it involves determining the result for each choice of function and proving the result by induction. For ordinary derivatives, the general higher-order chain rule is normally attributed to Faà di Bruno [28]. The following result generalizes Faà di Bruno's formula to chain differentials and allows for a systematic derivation of composite functions (see [18] for an example of exploitation in the context of Bayesian estimation).

Proposition 3 (General higher-order chain rule, from [19, 20]). *Under the differentiability and continuity conditions detailed in [20], the n^{th}-order variation of composition $f \circ g$ in the sequence of directions $(\eta_i)_{i=1}^n$ at point x is given by*

$$\delta^n(f\circ g)(x;(\eta_i)_{i=1}^n) = \sum_{\nu\in N_n} \delta^{|\nu|} f\left(g(x);\left(\delta^{|\omega|} g\big(x;(\eta_i)_{i\in\omega}\big)\right)_{\omega\in\nu}\right), \tag{3.12}$$

where $N_n = N(\{1,\ldots,n\})$ represents the set of partitions of the index set $\{1,\ldots,n\}$, and $|\nu|$ denotes the cardinality of the set ν.

Example 2 (General higher-order chain rule).

$$\delta^2(f\circ g)(x;\eta_1,\eta_2) = \underbrace{\delta^2 f\left(g(x);\delta g(x;\eta_1),\delta g(x;\eta_2)\right)}_{\nu=\{\{1\},\{2\}\}} + \underbrace{\delta f\left(g(x);\delta^2 g(x;\eta_1,\eta_2)\right)}_{\nu=\{\{1,2\}\}}. \tag{3.13}$$

Applying n^{th}-order chain differentials on a product of functions follows a more straightforward approach, similar to Leibniz' rule for ordinary derivatives.

Proposition 4 (Leibniz' rule, from [20])**.** *Under the differentiability conditions detailed in [20], the n^{th}-order variation of the product $f \cdot g$ in the sequence of directions $(\eta_i)_{i=1}^n$ at point x is given by*

$$\delta^n(f \cdot g)(x; (\eta_i)_{i=1}^n) = \sum_{\nu \subseteq \{1,\dots,n\}} \delta^{|\nu|} f(x; (\eta_i)_{i \in \nu}) \delta^{n-|\nu|} g(x; (\eta_i)_{i \in \nu^c}), \tag{3.14}$$

where $\nu^c = \{1, \dots, n\} \setminus \nu$ denotes the complement of ν in $\{1, \dots, n\}$.

Example 3 (Leibniz' rule)**.**

$$\delta^2(f \cdot g)(x; \eta_1, \eta_2) = \underbrace{\delta^2 f(x; \eta_1, \eta_2) g(x)}_{\nu=\{1,2\}} + \underbrace{\delta f(x; \eta_1) \delta g(x; \eta_2)}_{\nu=\{1\}} + \underbrace{\delta f(x; \eta_2) \delta g(x; \eta_1)}_{\nu=\{2\}} + \underbrace{f(x) \delta g(x; \eta_1, \eta_2)}_{\nu=\{\emptyset\}}. \tag{3.15}$$

3.4 Probability generating functionals and differentiation

Key properties of a point process can be recovered from the functional differentiation of its p.g.fl. Taking the k^{th}-order variation of $G_\Omega(h)$ in the directions $\eta_1, \dots, \eta_k$, we have (see, for example [80, p. 21]),

$$\delta^k G_\Omega(h; \eta_1, \dots, \eta_k) = \sum_{n \geq k} \frac{1}{(n-k)!} \int_{\mathbb{X}^n} \prod_{i=1}^k \eta_i(x_i) \prod_{i=k+1}^n h(x_i) \; J_\Omega^{(n)}(\mathrm{d}(x_1, \dots, x_n)). \tag{3.16}$$

It is then useful to consider the cases when we set $h = 1$ or $h = 0$, i.e.

$$\delta^k G_\Omega(0; \eta_1, \dots, \eta_k) = \int_{\mathbb{X}^k} \eta_1(x_1) \dots \eta_k(x_k) J_\Omega^{(k)}(\mathrm{d}(x_1, \dots, x_k)), \tag{3.17}$$

$$\delta^k G_\Omega(1; \eta_1, \dots, \eta_k) = \int_{\mathbb{X}^k} \eta_1(x_1) \dots \eta_k(x_k) M_\Omega^{(k)}(\mathrm{d}(x_1, \dots, x_k)), \tag{3.18}$$

Assuming that one wishes to evaluate the Janossy and factorial moment measures in some measurable subsets $B_i \in \mathcal{B}_\mathbb{X}$, $1 \leq i \leq k$, then they can be recovered from Eqs (3.17), (3.18) by setting the directions to be indicator functions $\eta_i = 1_{B_i}$, $1 \leq i \leq k$, so that

$$\delta^k G_\Omega(h; 1_{B_1}, \dots, 1_{B_k})\Big|_{h=0} = J_\Omega^{(k)}(B_1 \times \dots \times B_k), \tag{3.19}$$

$$\delta^k G_\Omega(h; 1_{B_1}, \dots, 1_{B_k})\Big|_{h=1} = M_\Omega^{(k)}(B_1 \times \dots \times B_k). \tag{3.20}$$

The propagation of the first-order factorial moment measure $M_{\Omega}^{(1)}$ – also called the intensity measure μ_{Ω} – of the multi-object point process Ω, in a Bayesian context, is a key component of the construction of both the PHD filter [61] and the CPHD filter [62]. The density of the intensity measure is the Probability Hypothesis Density [61].

3.5 Point process models

Note that in what follows here, and in Chapter 4 and Appendix A, there are some notational differences with previously presented background material. For example, variables r and p are used to denote probability of existence and probability density, respectively, in (2.18). Here, variables p and s are used to denote probability of existence and probability density, respectively. The two major filtering developments made in this research were performed using approaches that share a common origin, but are different enough that maintaining consistent notation between the two is challenging, to say the least, hence the change in variables.

3.5.1 Bernoulli process

A Bernoulli process Ω is characterized by a parameter $0 \leq p \leq 1$ and a spatial distribution s. It describes the situation where 1) either there is no object in the scene, or 2) there is a single object in the scene, with state distributed according to s. Its projection measures are given by

$$P_{\Omega}^{(n)}(B_1 \times \ldots \times B_n) = \begin{cases} 1-p, & n = 0, \\ ps(B_1), & n = 1, \\ 0, & \text{otherwise.} \end{cases} \tag{3.21}$$

Proposition 5 (p.g.fl. of a Bernoulli process [64])**.** *The p.g.fl. of a Bernoulli process Ω with parameter p and spatial distribution s is given by*

$$G_{\Omega}(h) = 1 - p + p \int h(x) s(\mathrm{d}x). \tag{3.22}$$

3.5.2 Poisson process

A Poisson process Ω is characterized by a rate $\lambda \geq 0$ and a spatial distribution s. It describes a population whose size follows a Poisson distribution and whose individual states are i.i.d. according to s. Its projection measure is given by

$$P_{\Omega}^{(n)}(B_1 \times \ldots \times B_n) = e^{-\lambda} \frac{\lambda^n}{n!} \prod_{i=1}^{n} s(B_i). \tag{3.23}$$

Proposition 6 (p.g.fl. of a Poisson process [64])**.** *The p.g.fl. of a Poisson process Ω with rate λ and spatial distribution s is given by*

$$G_{\Omega}(h) = \exp\left[\lambda \left(\int h(x)s(\mathrm{d}x) - 1\right)\right]. \tag{3.24}$$

3.5.3 Zero-inflated Poisson process

A zero-inflated Poisson process Ω (from [55]) is characterized by a parameter $0 \leq p \leq 1$, a rate $\lambda \geq 0$, and a spatial distribution s. It describes a population that is 1) either empty, or 2) non-empty, with size following a Poisson distribution and whose individual states are i.i.d. according to s. Its projection measures are given by

$$P_{\Omega}^{(n)}(B_1 \times \ldots \times B_n) = \begin{cases} 1 - p + pe^{-\lambda}, & n = 0, \\ pe^{-\lambda} \frac{\lambda^n}{n!} \prod_{i=1}^{n} s(B_i), & \text{otherwise.} \end{cases} \tag{3.25}$$

Note that a Poisson process is a special case of a zero-inflated Poisson process in which the parameter p is set to one.

Proposition 7 (p.g.fl. of a zero-inflated Poisson process)**.** *The p.g.fl. of a zero-inflated Poisson process Ω with parameter p, rate λ, and spatial distribution s is given by*

$$G_{\Omega}(h) = 1 - p + p \exp\left[\lambda \left(\int h(x)s(\mathrm{d}x) - 1\right)\right]. \tag{3.26}$$

3.5.4 I.i.d. process

An i.i.d. process Ω is characterized by a cardinality distribution ρ and a spatial distribution s. It describes a population whose size is distributed according to ρ, and whose individual states

are i.i.d. according to s. Its Janossy measures are given by

$$J_{\Omega}^{(n)}(B_1 \times \ldots \times B_n) = n!\rho(n)\prod_{i=1}^{n} s(B_i). \tag{3.27}$$

Note that a Poisson process is a special case of i.i.d. process in which the cardinality distribution ρ is Poisson.

3.6 CPHD Filter prediction derivation via p.g.fl. differentiation

The original CPHD predicted intensity and cardinality distribution are derived in this section to demonstrate use of the preceding material in this chapter, which is also used in the new developments presented in Chapter 4.

Assume the survival process is a Bernoulli point process, characterized by the p.g.fl. $G_{\mathrm{S}}(h|\cdot)$ given by

$$G_{\mathrm{S}}(h|x) = 1 - p_{\mathrm{S},k} + p_{\mathrm{S},k} \int h(x)s(\mathrm{d}x), \tag{3.28}$$

and that the birth process is characterized by a p.g.fl. $G_{\mathrm{B}}(h)$.

3.6.1 Cardinality Prediction

Using (3.19), let

$$J_{k|k-1}^{(n)}(\mathrm{d}(y_1, \ldots, y_n)) = \delta^n G_{k|k-1}(h; 1_{\mathrm{d}y_1}, \ldots, 1_{\mathrm{d}y_n})\big|_{h=0}, \tag{3.29}$$

$$= \delta^n (G_{k-1}(G_{\mathrm{S}}(h|\cdot))G_{\mathrm{B}}(h); 1_{\mathrm{d}y_1}, \ldots, 1_{\mathrm{d}y_n})|_{h=0}, \tag{3.30}$$

where

$$G_{k|k-1}(h) = G_{k-1}(G_{\mathrm{S}}(h|\cdot))G_{\mathrm{B}}(h), \tag{3.31}$$

and $G_{k-1}(h)$ is given in (3.5b). Applying the product rule (3.14) then gives

$$J_{k|k-1}^{(n)}(\mathrm{d}(y_1, \ldots, y_n)) = \sum_{\tau \subseteq \{1,\ldots,n\}} \delta^{|\tau|}(G_{k-1}(G_{\mathrm{S}}(h|\cdot)); (1_{\mathrm{d}y_i})_{i\in\tau})\big|_{h=0} \delta^{n-|\tau|} G_{\mathrm{B}}(h; (1_{\mathrm{d}y_i})_{i\in\tau^c})\big|_{h=0}. \tag{3.32}$$

Using Eq. (3.19) on the instantaneous birth process then gives

$$J_{k|k-1}^{(n)}(\mathrm{d}(y_1, \ldots, y_n)) = \sum_{\tau \subseteq \{1,\ldots,n\}} C_{|\tau|}(\mathrm{d}(y_i)_{i\in\tau}) J_{\mathrm{B}}^{(n-|\tau|)}(\mathrm{d}(y_i)_{i\in\tau^c}), \tag{3.33}$$

where

$$C_{|\tau|}(\mathrm{d}(y_i)_{i\in\tau}) = \delta^{|\tau|}(G_{k-1}(G_{\mathrm{S}}(h|\cdot)); (1_{\mathrm{d}y_i})_{i\in\tau})\big|_{h=0}, \tag{3.34}$$

and where $J_{\mathrm{B}}^{(n-|\tau|)}$ is the $(n-|\tau|)^{th}$-order Janossy measure of the instantaneous birth process. Using a change of variable for notational convenience and applying the general chain rule (3.12), let

$$C_q(\mathrm{d}(z_1,\ldots,z_q)) = \delta^q(G_{k-1}(G_{\mathrm{S}}(h|\cdot)); 1_{\mathrm{d}z_1},\ldots,1_{\mathrm{d}z_q})\big|_{h=0}, \tag{3.35}$$

$$= \sum_{\pi\in\Pi_q} \delta^{|\pi|} G_{k-1}\left(G_{\mathrm{S}}(h|\cdot); \left(\delta^{|\omega|}G_{\mathrm{S}}(h|\cdot; (1_{\mathrm{d}z_i})_{i\in\omega})\right)_{\omega\in\pi}\right)\Big|_{h=0}. \tag{3.36}$$

Developing the predicted p.g.fl. G_{k-1} through Janossy measures with Eq. (3.2) then gives

$$C_q(\mathrm{d}(z_1,\ldots,z_q)) = \sum_{\pi\in\Pi_q}\sum_{m\geq|\pi|} \frac{1}{(m-|\pi|)!}\int_{\mathbb{X}^m}\prod_{i=1}^{|\pi|}\delta^{|\omega_i|}G_{\mathrm{S}}(h|x_i; (1_{\mathrm{d}z_j})_{j\in\omega_i})\Big|_{h=0},$$
$$\times \prod_{i=|\pi|+1}^{m} G_{\mathrm{S}}(0|x_i) J_{k-1}^{(m)}(\mathrm{d}(x_1,\ldots,x_m)). \tag{3.37}$$

Since the prior process is assumed i.i.d., we can substitute the expression given by Eq. (3.27) to the prior Janossy measures $J_{k-1}^{(m)}$ and obtain

$$C_q(\mathrm{d}(z_1,\ldots,z_q)) = \sum_{\pi\in\Pi_q}\sum_{m\geq|\pi|}\frac{m!}{(m-|\pi|)!}\rho_{k-1}(m)C_\pi(\mathrm{d}(z_1,\ldots,z_q)), \tag{3.38}$$

where

$$C_\pi(\mathrm{d}(z_1,\ldots,z_q)) = \int\cdots\int\prod_{i=1}^{|\pi|}\delta^{|\omega_i|}G_{\mathrm{S}}(h|x_i; (1_{\mathrm{d}z_j})_{j\in\omega_i})\big|_{h=0}\prod_{i=|\pi|+1}^{m}G_{\mathrm{S}}(0|x_i)\prod_{i=1}^{m}s(\mathrm{d}x_i) \tag{3.39a}$$

$$= \left(\int G_{\mathrm{S}}(0|x)s(\mathrm{d}x)\right)^{m-|\pi|}\prod_{\omega\in\pi}\left(\int\delta^{|\omega|}G_{\mathrm{S}}(h|x; (1_{\mathrm{d}z_i})_{i\in\omega})\big|_{h=0}s(\mathrm{d}x)\right). \tag{3.39b}$$

Note that $\delta^{|\omega|}G_{\mathrm{S}}(h|x; 1_{\mathrm{d}z}) = 0$ when $|\omega| > 1$, and the only partition of π that does not result in a an inner summation of zeros is the one in which $|\pi| = q$, therefore, (3.38) can be simplified as

$$C_q(\mathrm{d}(z_1,\ldots,z_q)) = \sum_{m\geq q}\frac{m!}{(m-q)!}\rho_{k-1}(m)\langle 1-p_{\mathrm{S},k}, s\rangle^{m-q}\langle p_{\mathrm{S},k}, s\rangle^q. \tag{3.40}$$

Using (3.19), the predicted cardinality is given by

$$\rho_{k|k-1}(n) = \frac{1}{n!}\int_{\mathbb{X}^n} J^{(n)}_{k|k-1}(\mathrm{d}(y_1,\ldots,y_n)) \tag{3.41a}$$

$$= \sum_{\nu\subseteq\{1,\ldots,n\}} \frac{1}{(n-|\nu|)!}\int_{\mathbb{X}^{n-|\nu|}} J_{\mathrm{B}}^{(n-|\nu|)}(\mathrm{d}(y_i)_{i\in\nu^c})\frac{(n-|\nu|)!}{n!}\int_{\mathbb{X}^{|\nu|}} C_{|\nu|}(\mathrm{d}(y_i)_{i\in\nu}) \tag{3.41b}$$

$$= \sum_{q=0}^{n}\binom{n}{q}\rho_{\mathrm{B}}(n-q)\frac{(n-q)!}{n!}\sum_{m\geq q}\frac{m!}{(m-q)!}\rho_{k-1}(m)\langle 1-p_{\mathrm{S},k},s\rangle^{m-q}\langle p_{\mathrm{S},k},s\rangle^{q} \tag{3.41c}$$

$$= \sum_{q=0}^{n}\rho_{\mathrm{B}}(n-q)\sum_{m\geq q}\frac{m!}{q!(m-q)!}\rho_{k-1}(m)\langle 1-p_{\mathrm{S},k},s\rangle^{m-q}\langle p_{\mathrm{S},k},s\rangle^{q}. \tag{3.41d}$$

3.6.2 Intensity Prediction

Using Eq. (3.20) let

$$\mu_{k|k-1}(y) = \delta G_{k|k-1}(h;1_y)\big|_{h=1}, \tag{3.42}$$

$$= \delta\big(G_{k-1}(G_{\mathrm{S}}(h|\cdot))G_{\mathrm{B}}(h);1_y\big)\big|_{h=1}. \tag{3.43}$$

Using the product rule (3.14) it becomes

$$\mu_{k|k-1}(y) = \delta\big(G_{k-1}(G_{\mathrm{S}}(h|\cdot));1_y\big)\big|_{h=1}\underbrace{G_{\mathrm{B}}(1)}_{=1} + \underbrace{G_{k-1}(G_{\mathrm{S}}(1|\cdot))}_{=1}\delta G_{\mathrm{B}}(h;1_y)|_{h=1}. \tag{3.44}$$

Using the definition of the p.g.fl. (3.5a) then yields

$$\mu_{k|k-1}(y) = \delta\bigg(\int_{\mathcal{X}}\Big[\prod_{x\in\varphi}G_{\mathrm{S}}(h|x)\Big]P_{k-1}(\mathrm{d}\varphi);1_y\bigg)\bigg|_{h=1} + \delta G_{\mathrm{B}}(h;1_y)|_{h=1}, \tag{3.45}$$

$$= \int_{\mathcal{X}}\delta\bigg(\prod_{x\in\varphi}G_{\mathrm{S}}(h|x);1_y\bigg)\bigg|_{h=1}P_{k-1}(\mathrm{d}\varphi) + \delta G_{\mathrm{B}}(h;1_y)|_{h=1}. \tag{3.46}$$

Using Eq. (3.20), introduce the intensities μ_{S} and μ_{B} of the survival and spontaneous birth processes and obtain:

$$\mu_{k|k-1}(y) = \int_{\mathcal{X}}\sum_{x\in\varphi}\mu_{\mathrm{S}}(y|x)P_{k-1}(\mathrm{d}\varphi) + \mu_{\mathrm{B}}(y), \tag{3.47}$$

which becomes, using Campbell's theorem [25, p. 271]:

$$\mu_{k|k-1}(y) = \int \mu_{\mathrm{S}}(y|x)\mu_{k-1}(\mathrm{d}x) + \mu_{\mathrm{B}}(y). \tag{3.48}$$

Chapter 4

The CPHD Filter with Object-Spawning

This chapter presents results for the development of spawn model implementations of the CPHD filter. The contents of this chapter were formed into a paper and published in the IEEE Transactions on Signal Processing [14]. The derivations in this chapter take a measure-theoretic approach as opposed to the RFS approach presented in Chapter 2. Nonetheless, definitions and other forms of supporting material required to follow the derivations presented here are contained within Chapter 3, this chapter, and the references thereof.

4.1 The CPHD filter with spawning

This section covers the derivation of the filtering equations for the CPHD filter for various object spawning processes. Section 4.1.1 provides a brief description of the general multi-object Bayes filter [63], and the principled approximation leading to the construction of the original CPHD filter [62]. Section 3.5 then presents the various models of point processes that will be necessary for the construction of the CPHD filter with spawning in Section 4.1.2.

4.1.1 Multi-Object Bayes Recursion

The multi-object Bayes recursion, shown in (4.1) and (4.2) with measure-theoretic notation, is used to propagate the posterior distribution $P_k(\cdot|Z_{1:k})$ that describes the current object population

based on all the measurements $Z_1, \ldots, Z_k$ collected so far.

$$P_{k|k-1}(\mathrm{d}X|Z_{1:k-1}) = \int_{\mathcal{X}} f_{k|k-1}(X|\bar{X}) P_{k-1}(\mathrm{d}\bar{X}|Z_{1:k-1}), \tag{4.1}$$

$$P_k(\mathrm{d}X|Z_{1:k}) = \frac{g_k(Z_k|X) P_{k|k-1}(\mathrm{d}X|Z_{1:k-1})}{\int_{\mathcal{X}} g_k(Z_k|\bar{X}) P_{k|k-1}(\mathrm{d}\bar{X}|Z_{1:k-1})}, \tag{4.2}$$

The CPHD Bayes recursion aims at simplifying the multi-object Bayes recursion by approximating the predicted and posterior multi-object processes as i.i.d. processes[1], a class of point processes fully characterized by their cardinality distribution ρ_Ω and their first-order moment measure μ_Ω [62]. The CPHD filter thus focuses on the propagation of the posterior cardinality distribution ρ_k and the posterior first-order moment measure μ_k, rather than the full posterior probability distribution P_k.

The original construction of the CPHD filter [62] does not consider an object spawning mechanism, and the key contribution of this work is to propose the integration of several object spawning models in the CPHD time prediction equation (see Section 4.1.2). Note that the measurement update step does not involve the object spawning mechanism, therefore its presentation here is unnecessary. Update step details can be found in Section 2.1.3.2, Section 2.3.1, and [96].

4.1.2 Prediction step

In this section, an alternative expression of the original CPHD time prediction step [62] is proposed in which newborn objects may originate from either a spawning mechanism or spontaneous birth. Note that the assumptions on the posterior multi-object process from the previous time step, the object survival mechanism, and the object evolution mechanism are identical to the original assumptions in [62].

Theorem 1 (CPHD with spawning: prediction step)**.** *Assuming that, at step k:*

- *The posterior multi-object process Φ_{k-1} is an i.i.d. process with intensity measure μ_{k-1}, with cardinality distribution ρ_{k-1}, and spatial distribution s_{k-1},*

[1] The definition of an i.i.d. process is given in Section 3.5.

- *An object in state x at time $k-1$ survives to time k with probability $p_{\mathrm{S},k}(x)$,*
- *A surviving object in state x at time $k-1$ evolves since time $k-1$ according to a Markov transition $f_{\mathrm{S},k}(\cdot|x)$,*
- *Newborn objects are born, independently from prior objects, following a process with intensity measure $\mu_{\mathrm{B},k}$, and cardinality distribution $\rho_{\mathrm{B},k}$,*
- *Newborn objects are spawned from a prior object, in state x at time $k-1$, following a process with intensity measure $\mu_{\mathrm{T},k}(\cdot|x)$, and cardinality distribution $\rho_{\mathrm{T},k}(\cdot|x)$,*

then the intensity measure $\mu_{k|k-1}$ and cardinality distribution $\rho_{k|k-1}$ of the predicted multi-object process $\Phi_{k|k-1}$ are given by

$$\mu_{k|k-1}(\cdot) = \int \left[p_{\mathrm{S},k}(x) f_{\mathrm{S},k}(\cdot|x) + \mu_{\mathrm{T},k}(\cdot|x)\right] \mu_{k-1}(\mathrm{d}x) + \mu_{\mathrm{B},k}(\cdot), \tag{4.3}$$

$$\rho_{k|k-1}(n) = \sum_{\ell=0}^{n} \rho_{\mathrm{B},k}(n-\ell) \sum_{j=0}^{\ell} B_{\ell,j}(b_1, \ldots, b_\ell) \left[\sum_{m \geq j} \frac{m!}{\ell!(m-j)!} \rho_{k-1}(m) {b_0}^{m-j}\right], \tag{4.4}$$

where $B_{\ell,j}$ is the partial Bell polynomial [16, p. 412] given by

$$B_{\ell,k}(x_1, x_2, \cdots, x_\ell) = \sum_{\substack{k_1+2k_2+\cdots+\ell k_\ell=\ell \\ k_1+k_2+\cdots+k_\ell=k}} \frac{\ell!}{k_1!(1!)^{k_1} k_2!(2!)^{k_2} \cdots k_\ell!(\ell!)^{k_\ell}} x_1^{k_1} x_2^{k_2} \cdots x_\ell^{k_\ell}, \tag{4.5}$$

where the coefficients b_i are given by

$$b_i = \begin{cases} \int q_{\mathrm{S},k}(x) \rho_{\mathrm{T},k}(0|x) s_{k-1}(\mathrm{d}x), & i = 0, \\ i! \int \left[q_{\mathrm{S},k}(x) \rho_{\mathrm{T},k}(i|x) + p_{\mathrm{S},k}(x) \rho_{\mathrm{T},k}(i-1|x)\right] s_{k-1}(\mathrm{d}x), & i > 0, \end{cases} \tag{4.6}$$

where $q_{\mathrm{S},k}(\cdot) \equiv 1 - p_{\mathrm{S},k}(\cdot)$.

The proof is given in the Appendix A. Note that the structure of the predicted cardinality (4.4) allows for its efficient computation through an algorithm dedicated to the computation of partial Bell polynomials. Exploiting the recursive formula [16, (11.11)], we propose an algorithm in Appendix B for implementation of the predicted cardinality (4.4) with a computational cost of $\mathcal{O}(n_{\max}^3)$, where $n_{\max}$ is the maximum number of objects considered for the support of the

cardinality distributions. Also, note that the construction of the predicted intensity (4.3) is identical to that of the original PHD filter (see [89], for example).

Corollary 1. *The intensity measure $\mu_{\mathrm{T},k}$ and the coefficients b_i, describing the spawning process in the CPHD prediction step* (4.3), (4.4)*, depend on the modeling choices. Denoting $q_{\mathrm{T},k}(\cdot) \equiv 1 - p_{\mathrm{T},k}(\cdot)$, they are given as follows for various spawning processes:*

a) Bernoulli process, with parameter $p_{\mathrm{T},k}$ and spatial distribution $s_{\mathrm{T},k}$:

$$\mu_{\mathrm{T},k}(\cdot|x) = p_{\mathrm{T},k}(x) s_{\mathrm{T},k}(\cdot|x), \tag{4.7}$$

and

$$b_i = \begin{cases} \int q_{\mathrm{S},k}(x) q_{\mathrm{T},k}(x) s_{k-1}(\mathrm{d}x), & i = 0, \\ \int \big[p_{\mathrm{S},k}(x) q_{\mathrm{T},k}(x) + q_{\mathrm{S},k}(x) p_{\mathrm{T},k}(x)\big] s_{k-1}(\mathrm{d}x), & i = 1, \\ 2\int p_{\mathrm{S},k}(x) p_{\mathrm{T},k}(x) s_{k-1}(\mathrm{d}x), & i = 2, \\ 0, & i > 2. \end{cases} \tag{4.8}$$

b) Poisson process, with rate $\lambda_{\mathrm{T},k}$ and spatial distribution $s_{\mathrm{T},k}$:

$$\mu_{\mathrm{T},k}(\cdot|x) = \lambda_{\mathrm{T},k}(x) s_{\mathrm{T},k}(\cdot|x), \tag{4.9}$$

and

$$b_i = \int \lambda_{\mathrm{T},k}^{i-1}(x) e^{-\lambda_{\mathrm{T},k}(x)} \big[q_{\mathrm{S},k}(x)\lambda_{\mathrm{T},k}(x) + i p_{\mathrm{S},k}(x)\big] s_{k-1}(\mathrm{d}x), \tag{4.10}$$

for $i \geq 0$.

c) zero-inflated Poisson process, with parameter $p_{\mathrm{T},k}$, rate $\lambda_{\mathrm{T},k}$, and spatial distribution $s_{\mathrm{T},k}$:

$$\mu_{\mathrm{T},k}(\cdot|x) = p_{\mathrm{T},k}(x)\lambda_{\mathrm{T},k}(x) s_{\mathrm{T},k}(\cdot|x), \tag{4.11}$$

and,

$$b_i = \begin{cases} \int q_{\mathrm{S},k}(x)\big[q_{\mathrm{T},k}(x) + p_{\mathrm{T},k}(x) e^{-\lambda_{\mathrm{T},k}(x)}\big] s_{k-1}(\mathrm{d}x), & i = 0, \\ \int \big[q_{\mathrm{S},k}(x) p_{\mathrm{T},k}(x) e^{-\lambda_{\mathrm{T},k}(x)} \lambda_{\mathrm{T},k}(x) + p_{\mathrm{S},k}(x)\big[q_{\mathrm{T},k}(x) + p_{\mathrm{T},k}(x) e^{-\lambda_{\mathrm{T},k}(x)}\big]\big] s_{k-1}(\mathrm{d}x), & i = 1, \\ \int p_{\mathrm{T},k}(x)\lambda_{\mathrm{T},k}^{i-1}(x) e^{-\lambda_{\mathrm{T},k}(x)} \big[q_{\mathrm{S},k}(x)\lambda_{\mathrm{T},k}(x) + i p_{\mathrm{S},k}(x)\big] s_{k-1}(\mathrm{d}x), & i \geq 2. \end{cases} \tag{4.12}$$

The proof is given in Appendix A. Note that the construction of the predicted cardinality for a CPHD filter with Bernoulli and Poisson spawning processes was previously explored in [60], following traditional Bayesian statistics. The results provided in [60] are not supported by a detailed construction; however, an earlier work by the same authors [59] proposed a more principled derivation of the predicted cardinality through the exploitation of Probability Generating Functions (p.g.f.s). It can be shown (a sketch is given in Appendix A) that the general expression of the predicted cardinality [59, (A.15)] is equivalent to the presentation here through the Bell polynomials in Eq. (4.4). However, the latter facilitates practical implementation of the CPHD filter with spawning, and allows for a clear presentation of the exact time prediction equation (i.e., without requiring additional approximations) for specific models of spawning through the coefficients b_i, as illustrated in Corollary 1.

4.2 Simulation

The CPHD filter with spawning models is illustrated in this section through a simulation-based scenario. The GM implementation of the CPHD filter is briefly described in Section 4.2.1. The scenario and the selection of the filter parameters are detailed in Section 4.2.2, and the results are discussed in Section 4.2.3.

4.2.1 The GM-CPHD filter with spawning

Since the incorporation of spawning in the CPHD filtering process does not affect the data update step, this section focuses only on the specifics of the prediction step for the GM-CPHD filter with spawning. A description of the usual GM-CPHD, including the implementation of the spontaneous birth term, can be found in Section 2.1.3.2, Section 2.3.1, and [96].

4.2.1.1 Filtering assumptions

The usual assumptions of the GM-CPHD filter [96] are followed regarding the transition process from time $k-1$ to time k, namely, that the probability of survival $p_{\mathrm{S},k}$ is uniform over the

state space $\mathbb{X}$ and the transition $f_{\mathrm{S},k}$ follows a linear Gaussian dynamical model:

$$p_{\mathrm{S},k}(\cdot) \equiv p_{\mathrm{S},k}, \tag{4.13}$$

$$f_{\mathrm{S},k|k-1}(\cdot|x) = \mathcal{N}(\cdot\,; F_k x, Q_k), \tag{4.14}$$

where $\mathcal{N}(\cdot\,; m, P)$ denotes a Gaussian distribution with mean m and covariance P, F_k is a state transition matrix, and Q_k is a process noise covariance matrix.

Regardless of the chosen spawning model (see Theorem 1), it is further assumed that the spatial distribution of each spawned object $s_{\mathrm{T},k}$ can be described as the Gaussian mixture

$$s_{\mathrm{T},k}(\cdot|x) = \sum_{j=1}^{J_{\mathrm{T},k}} w_{\mathrm{T},k}^{(j)} \mathcal{N}(\cdot\,; F_{\mathrm{T},k}^{(j)} x + d_{\mathrm{T},k}^{(j)}, Q_{\mathrm{T},k}^{(j)}), \tag{4.15}$$

where $d_{\mathrm{T},k}^{(j)}$ is a deviation vector, $F_{\mathrm{T},k}^{(j)}$ is a spawning transition matrix, and $Q_{\mathrm{T},k}^{(j)}$ is a spawning noise covariance matrix, for $1 \leq j \leq J_{\mathrm{T},k}$, and $\sum_{j=1}^{J_{\mathrm{T},k}} w_{\mathrm{T},k}^{(j)} = 1$. Also, assume that the model parameters $p_{\mathrm{T},k}$, $\lambda_{\mathrm{T},k}$, when applicable, are uniform over the state space $\mathbb{X}$:

$$\begin{aligned} p_{\mathrm{T},k}(\cdot) &\equiv p_{\mathrm{T},k}, \\ \lambda_{\mathrm{T},k}(\cdot) &\equiv \lambda_{\mathrm{T},k}. \end{aligned} \tag{4.16}$$

4.2.1.2 Predicted intensity

The construction of the predicted intensity $\mu_{k|k-1}$ in Eq. (4.3) follows a similar structure as for the usual GM-CPHD filter [89]. Assume that the posterior intensity μ_{k-1} can be written as a Gaussian mixture of the form

$$\mu_{k-1}(\cdot) = \sum_{j=1}^{J_{k-1}} w_{k-1}^{(j)} \mathcal{N}(\cdot\,; m_{k-1}^{(j)}, P_{k-1}^{(j)}), \tag{4.17}$$

where $m_{k-1}^{(j)}$ and $P_{k-1}^{(j)}$ are the posterior mean and covariance of the j-th component of the mixture, respectively. Then the predicted intensity $\mu_{k|k-1}$ can also be written as a Gaussian mixture of the form

$$\mu_{k|k-1}(\cdot) = \mu_{\mathrm{S},k|k-1}(\cdot) + \mu_{\mathrm{T},k|k-1}(\cdot), \tag{4.18}$$

where the surviving component $\mu_{\mathrm{S},k|k-1}$ is the Gaussian mixture

$$\mu_{\mathrm{S},k|k-1}(\cdot) = p_{\mathrm{S},k} \sum_{j=1}^{J_{k-1}} w_{k-1}^{(j)} \mathcal{N}(\cdot\,; m_{\mathrm{S},k|k-1}^{(j)}, P_{\mathrm{S},k|k-1}^{(j)}), \tag{4.19}$$

with

$$m_{\mathrm{S},k|k-1}^{(j)} = F_k m_{k-1}^{(j)}, \tag{4.20}$$

$$P_{\mathrm{S},k|k-1}^{(j)} = Q_k + F_k P_{k-1}^{(j)} F_k^T, \tag{4.21}$$

for $1 \leq j \leq J_{k-1}$, and the spawning component $\mu_{\mathrm{T},k|k-1}$ is the Gaussian mixture

$$\mu_{\mathrm{T},k|k-1}(\cdot) = \alpha_{\mathrm{T},k} \sum_{j=1}^{J_{k-1}} w_{k-1}^{(j)} \sum_{i=1}^{J_{\mathrm{T},k}} w_{\mathrm{T},k}^{(i)} \mathcal{N}(\cdot\,; m_{\mathrm{T},k|k-1}^{(j,i)}, P_{\mathrm{T},k|k-1}^{(j,i)}), \tag{4.22}$$

with

$$m_{\mathrm{T},k|k-1}^{(j,i)} = F_{\mathrm{T},k}^{(i)} m_{k-1}^{(j)} + d_{\mathrm{T},k}^{(i)}, \tag{4.23}$$

$$P_{\mathrm{T},k|k-1}^{(j,i)} = Q_{\mathrm{T},k}^{(i)} + F_{\mathrm{T},k}^{(i)} P_{k-1}^{(j)} (F_{\mathrm{T},k}^{(i)})^T, \tag{4.24}$$

for $1 \leq j \leq J_{k-1}$, $1 \leq i \leq J_{\mathrm{T},k}$, and the scalar $\alpha_{\mathrm{T},k}$ depends on the spawning model:

$$\alpha_{\mathrm{T},k} = \begin{cases} p_{\mathrm{T},k}, & \text{Bernoulli process,} \\ \lambda_{\mathrm{T},k}, & \text{Poisson process,} \\ p_{\mathrm{T},k}\lambda_{\mathrm{T},k}, & \text{zero-inflated Poisson process.} \end{cases} \tag{4.25}$$

4.2.1.3 Predicted cardinality distribution

Due to the assumptions presented in Section 4.2.1.1, the coefficients of the Bell polynomial in Eq. (4.4) have the simpler form

a) Bernoulli process:

$$b_i = \begin{cases} (1 - p_{\mathrm{S},k})(1 - p_{\mathrm{T},k}), & i = 0, \\ p_{\mathrm{S},k}(1 - p_{\mathrm{T},k}) + (1 - p_{\mathrm{S},k}) p_{\mathrm{T},k}, & i = 1, \\ 2 p_{\mathrm{S},k} p_{\mathrm{T},k}, & i = 2, \\ 0, & i > 2. \end{cases} \tag{4.26}$$

b) Poisson process:

$$b_i = \lambda_{\mathrm{T},k}^{i-1} e^{-\lambda_{\mathrm{T},k}} \left[(1 - p_{\mathrm{S},k}) \lambda_{\mathrm{T},k} + i p_{\mathrm{S},k} \right], \quad i \geq 0. \tag{4.27}$$

c) zero-inflated Poisson process:

$$b_i = \begin{cases} (1 - p_{\mathrm{S},k}) \left(1 - p_{\mathrm{T},k} + p_{\mathrm{T},k} e^{-\lambda_{\mathrm{T},k}} \right), & i = 0, \\ (1 - p_{\mathrm{S},k}) p_{\mathrm{T},k} e^{-\lambda_{\mathrm{T},k}} \lambda_{\mathrm{T},k} + p_{\mathrm{S},k} \left(1 - p_{\mathrm{T},k} + p_{\mathrm{T},k} e^{-\lambda_{\mathrm{T},k}} \right), & i = 1, \\ p_{\mathrm{T},k} \lambda_{\mathrm{T},k}^{i-1} e^{-\lambda_{\mathrm{T},k}} \left[(1 - p_{\mathrm{S},k}) \lambda_{\mathrm{T},k} + i p_{\mathrm{S},k} \right], & i \geq 2. \end{cases} \tag{4.28}$$

The predicted cardinality distribution is then computed by the appropriate substitution of Eqs. (4.26)-(4.28) into Eq. (4.4).

4.2.2 Scenario and filter setup

A point $[x, y, \dot{x}, \dot{y}]$ of the single-object state space $\mathbb{X} \subset \mathbb{R}^4$ describes the position and velocity coordinates of an object in a square surveillance region of size $2000\,\mathrm{m} \times 2000\,\mathrm{m}$. The simulated multi-object tracking scenario consists of one scan per second for 100 s, and up to seven objects evolving in the region with constant velocity. Two objects are present at the beginning of the scenario and each spawns objects at different times: object 1 spawns two additional objects at $t = 15\,\mathrm{s}$ and object 2 spawns three additional objects at $t = 25\,\mathrm{s}$. All spawned objects have a lifespan of 60 s. Fig. 4.1 shows the trajectories of the objects cumulated over time, while Fig. 4.2 illustrates these trajectories and the collected measurements across time.

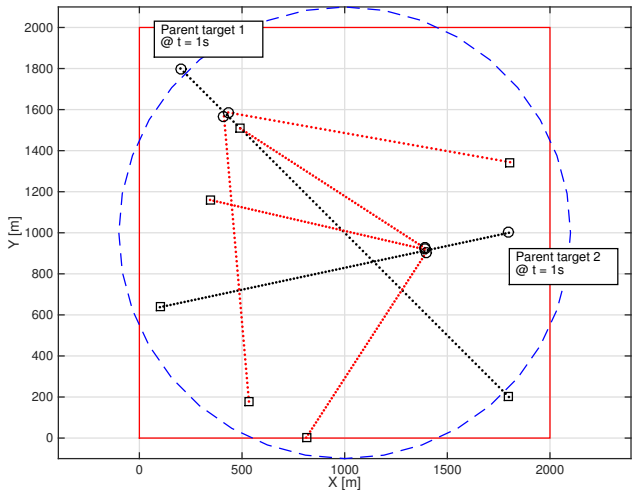
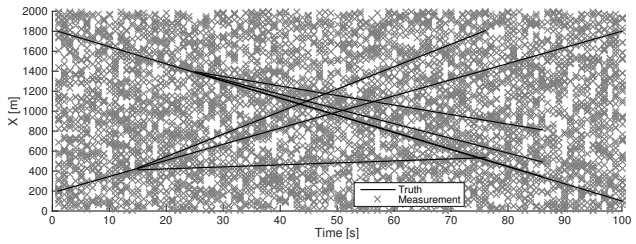
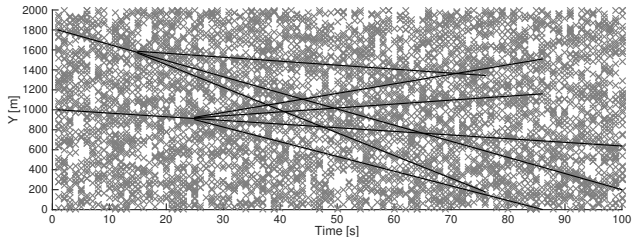


Figure 4.1: Object trajectories. A circle "○" indicates where a trajectory begins, and a square "□" indicates where a trajectory ends. The large square indicates the limits of the sensor's FOV and the large dashed circle represents the 90% confidence region of the Gaussian component of the spontaneous birth model.



(a) x-axis



(b) y-axis

Figure 4.2: Collected measurements (gray crosses) and object positions (black lines).

The probability of survival $p_{\mathrm{S},k}$ (4.13) is constant throughout the scenario, and set to $p_{\mathrm{S},k} =$

0.99. The object motion model $f_{\mathrm{S},k|k-1}$ (4.14) is set as follows:

$$F_k = \begin{bmatrix} \mathbf{1}_2 & \Delta\mathbf{1}_2 \\ \mathbf{0}_2 & \mathbf{1}_2 \end{bmatrix}, \qquad Q_k = \sigma_\nu^2 \begin{bmatrix} \frac{\Delta^4}{4}\mathbf{1}_2 & \frac{\Delta^3}{2}\mathbf{1}_2 \\ \frac{\Delta^3}{2}\mathbf{1}_2 & \Delta^2\mathbf{1}_2 \end{bmatrix}, \tag{4.29}$$

where $\Delta = 1\,\mathrm{s}$, $\sigma_\nu = 5\,\mathrm{m\,s^{-2}}$, $\mathbf{1}_n$ denotes the $n \times n$ identity matrix, and $\mathbf{0}_n$ denotes the $n \times n$ zero matrix.

The sensor's probability of detection is uniform over the sensor's FOV, and set at a constant value of 0.95 throughout the scenario. Each object-generated measurement consists of the object's coordinate position with an independent Gaussian white noise on each component, with a standard deviation of 10 m. Spurious measurements are modeled as a Poisson point process with uniform spatial distribution over the state space and an average number of clutter per unit volume of $12.5 \times 10^{-6}\,\mathrm{m^{-2}}$, that is, an average of 50 clutter returns per scan over the surveillance region.

For the sake of comparison, the usual GM-CPHD filter [96] with spontaneous birth and no spawning is implemented as well. The spontaneous birth model is Poisson, with a constant rate of 0.025 per time step (which yields, over the 100 s of the scenario, an average of 2.5 newborn objects). The spatial distribution is modeled with a single Gaussian component, centered on the sensor's FOV as illustrated in Fig. 4.1.

The spatial distribution of the spawning (4.15) is identical for the three considered models. Assume no spawned object deviation vectors, and a standard deviation of 12 units is set on each component of the spawning noise covariance, i.e.

$$F_{\mathrm{T},k} = \begin{bmatrix} \mathbf{1}_2 & \mathbf{0}_2 \\ \mathbf{0}_2 & \mathbf{1}_2 \end{bmatrix}, \; d_{\mathrm{T},k} = \mathbf{0}, \; Q_{\mathrm{T},k} = \begin{bmatrix} \sigma_{\mathrm{T}}^2\mathbf{1}_2 & \mathbf{0}_2 \\ \mathbf{0}_2 & \dot{\sigma}_{\mathrm{T}}^2\mathbf{1}_2 \end{bmatrix}, \tag{4.30}$$

where $\mathbf{0}$ denotes the null vector in $\mathbb{X}$, $\sigma_{\mathrm{T}} = 12\,\mathrm{m}$, and $\dot{\sigma}_{\mathrm{T}} = 12\,\mathrm{m\,s^{-1}}$.

The parameters of the three spawning models are set as follows. The zero-inflated Poisson model assumes one spawning per parent object during the scenario with an average of 2.5 spawned objects per spawning event, thus $p_{\mathrm{T},k}$ and λ_{T} are set to 0.01 and 2.5, respectively. Relative to the zero-inflated Poisson model, the Poisson model is set to yield a similar spawning intensity thus its

$\lambda_{\mathrm{T},k}$ is set to 0.025, whereas the Bernoulli model is set to yield a similar spawning frequency so its $p_{\mathrm{T},k}$ is set to 0.01. These parameters are also presented in Table 4.1.

Table 4.1: Spawn model parameters.

Model	$p_{\mathrm{T},k}$	$\lambda_{\mathrm{T},k}$	$\mu_{\mathrm{T},k}(\cdot\|x)$
Bernoulli	0.01	-	$0.01 \cdot \mathcal{N}(\cdot\,; x, Q_{\mathrm{T},k})$
Poisson	-	0.025	$0.025 \cdot \mathcal{N}(\cdot\,; x, Q_{\mathrm{T},k})$
zero-inflated Poisson	0.01	2.5	$0.025 \cdot \mathcal{N}(\cdot\,; x, Q_{\mathrm{T},k})$

It is interesting to note that neither the Poisson nor the Bernoulli models are equipped to capture the nature of the spawning events occurring in this scenario, since, per construction, the Poisson model is a poor match for spawning events occurring at unknown times and the Bernoulli model is a poor match for spawning events creating more than one spawned object. The zero-inflated Poisson model possesses a greater flexibility and should be able to cope with a wider range of spawning situations; in any case, it is expected to yield better performances on the scenario presented in this paper.

To maintain tractability, GM components are truncated with threshold $T = 10^{-5}$, pruned with maximum number of components $J_{\max} = 100$, and merged with threshold $U = 4$ (see [89] for more details on the pruning and merging mechanisms). Additionally, the maximum number of objects is set to $N_{\max} = 20$ to circumvent issues with infinitely tailed cardinality distributions [96].

4.2.3 Simulation results

The proposed spawning models and the birth model are implemented with the GM-CPHD filter, and compared over 500 Monte Carlo (MC) runs of the multi-object scenario decribed in Section 4.2.2.

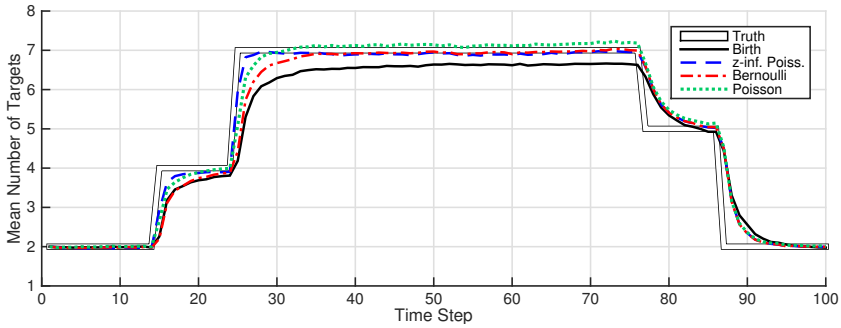


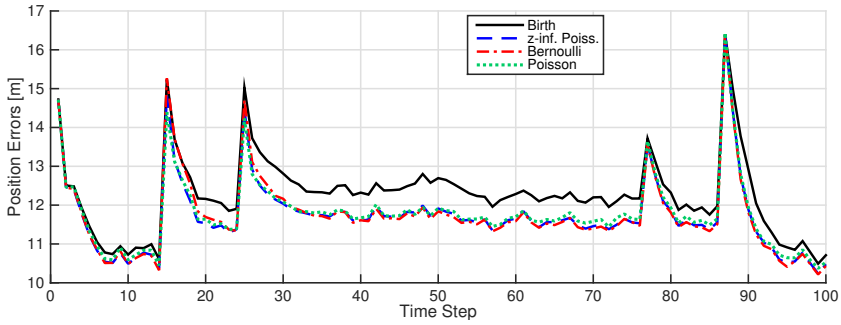
Figure 4.3: MAP estimate of the number of objects (averaged on 500 runs).

The MAP estimate of the number of objects is plotted in Fig. 4.3, along with the true number of objects in the scene. The results suggest that the spawning models provide a better estimate of the number of objects and, in particular, converge faster to the true number of objects following the appearance of new objects in the scene. This is expected, because the scenario does not feature any spontaneous but only spawning-related births, and thus in this context spawning models are a better match than the birth model.

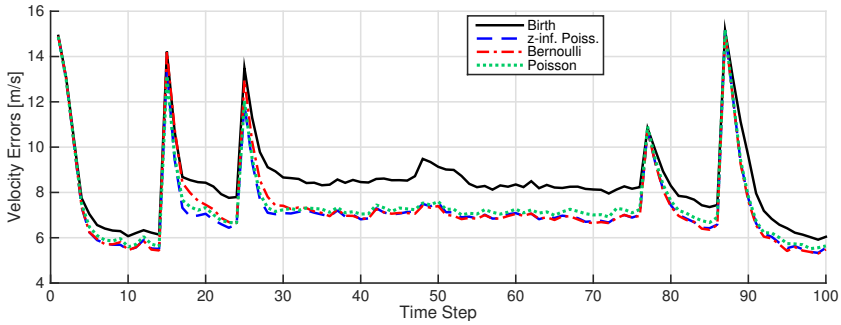
Among the three spawning models, the zero-inflated Poisson converges the fastest following the appearance of new objects, while the Bernoulli model converges the slowest. This is expected, for the zero-inflated Poisson model provides the best match to the spawning events occurring in this scenario. Note in particular that the Bernoulli model may not consider the appearance of more than one spawned object per spawning event, and must therefore stage the multiple-object appearances across several successive time steps; in other words, the Bernoulli is ill-adapted to "busy" events where objects appear simultaneously. Note also the slight overestimation shown by the Poisson model when the true number of object is stable. Per construction, the Poisson model is well-equipped for the simultaneous appearance of an arbitrary number of spawned objects at any time step, but it fails at coping with "quiet" periods where no spawning occurs because, unlike the zero-inflated Poisson model, it does not temper the Poisson-driven spawning with a probability of spawning. In other words, the Poisson model is ill-adapted to the spawning events shown in this scenario.

Note that all models – spawning and birth – follow the same mechanism for object deaths

and yield much closer performances when object disappearances occur.



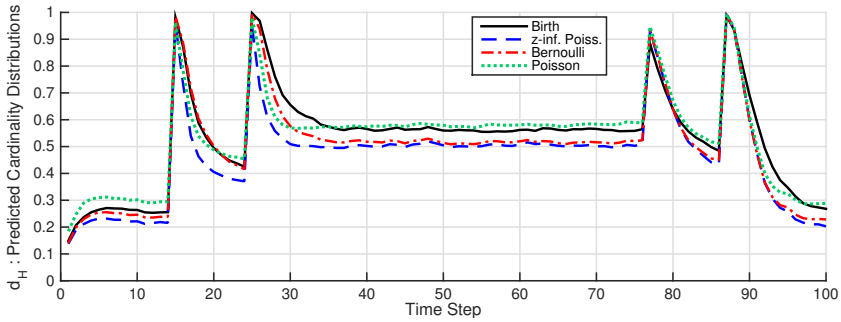
(a) Position



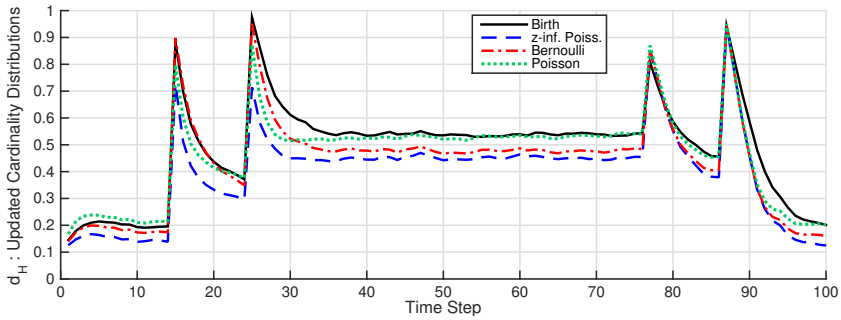
(b) Velocity

Figure 4.4: OSPA distance (averaged on 500 runs).

Similar conclusions can be drawn from the comparisons of the OSPA distances shown in Fig. 4.4. All models show error spikes at times of spawning ($t = 15\,\mathrm{s}$, $t = 25\,\mathrm{s}$) and death ($t = 76\,\mathrm{s}$, $t = 86\,\mathrm{s}$), however, the spawning models recover more quickly than the birth model, and have consistently lower errors.



(a) Predicted cardinality



(b) Updated cardinality

Figure 4.5: Hellinger distances (averaged on 500 runs).

The quality of the estimation of the number of objects proposed by the four models is further illustrated in Fig. 4.5, where the Hellinger distance between the cardinality distribution propagated by each model and the "ideal" cardinality distribution (i.e., a distribution in which all the mass is concentrated on the true number of objects).

The results in Fig. 4.5 allow a more refined analysis of the proposed models. All the models yield poor estimates immediately after a change in the true number of objects[2], but the zero-inflated Poisson model converges the fastest following an object birth/death and it converges to the best estimate during periods where the number of object is stable. The Poisson model converges faster than the Bernoulli model, but to a worse estimate: this is expected, since the Poisson model is ill-adapted to "quiet" periods while the Bernoulli model is ill-adapted to "busy" events (see discussion above on Fig. 4.3).

[2] Recall from Eq. (2.98) that the Hellinger distance d_H is such that $0 \leq d_H \leq 1$.

As expected, the updated cardinality distributions are consistently more accurate than the predicted cardinality distributions since they benefit from the processing of an additional measurement batch.

Chapter 5

Spawned Object Orbit Determination Simulations: GM-CPHD

CubeSat deployments offer a unique opportunity to test spawn model filter implementations because they are typically scheduled, well defined, and controlled events. The QB50 mission [87] in which fifty CubeSats were planned for deployment in LEO is a good example of this as its mission design is well documented, including a study of deployment strategies [53]. Important to note, in the time since the research presented here was performed, according to the QB-50 mission's website[1], only thirty-six QB50 CubeSats have been launched: twenty-eight from the ISS and eight from an Indian PSLV rocket. The simulations presented in this chapter were performed in early 2016 at which time the QB50 mission was reportedly preparing to launch fifty CubeSats from the ISS that summer[2].

TLE sets released by the JSpOC are used by most CubeSat programs for orbit determination, however, it is increasingly the case where large numbers of CubeSats are deployed from a launch vehicle and TLE's get confused [33]. Random finite set filters, formulated with spawn models, could potentially alleviate such confusion by leveraging launch vehicle state information to instantiate new states for CubeSats.

CubeSat deployments simulated in this chapter are modeled after the QB50 mission where fifty CubeSats were planned to deploy from the ISS. Excluding implementation considering ballistic coefficients, the "optimum scenario" proposed in [53], hereafter referred to as the QB50 deployment model, is used for simulation with the following details:

[1] https://www.qb50.eu/. Accessed 10 Oct. 2017.

[2] http://amsat-uk.org/2015/09/08/qb50-cubesats-to-be-deployed-from-iss/. Accessed 10 Oct. 2017.

- CubeSats are deployed one at a time,
- 4-6 minute interval between consecutive deployments,
- deployment directions determined for each deployment by random rotation angle θ about the In-track axis of the Radial In-track Cross-track (RIC) reference frame w.r.t. the launch vehicle, where $135^\circ \pm 10^\circ < \theta < 225^\circ \pm 10^\circ$.

Though its primary goals are to generate a "string of pearls" constellation and prevent collisions, the model also promotes CubeSat observability via spatial and temporal distributions.

Orbit determination for objects generated by a fragmentation event is significantly more challenging due primarily to uncertainties regarding event time, number of fragments, and an infinite number of possible directions a fragment can deviate from its parent trajectory. As a proof of concept, the fragmentation event simulated in this chapter is a RB explosion modeled after a Delta 2 explosion where six new RSOs entered the SSN catalog[3].

The event is simulated using the NASA breakup model (NBM) detailed in Section 2.9. Using characteristic lengths $l_c > 11\,\text{cm}$, a Δv magnitude distribution is generated using (2.115). Six Δv's are drawn from the distribution, and, since the NBM does not account for the direction in which Δv's are applied, directions are generated by randomly selecting points on a unit sphere.

5.1 Scenario Descriptions

Each simulation begins with only one object at epoch, i.e., the RB or QB50 launch vehicle (LV), the initial conditions for which are presented in Table 5.1. The RB and QB50 LV trajectories are based on that of SSN# 25637 and the ISS, respectively. For both simulations, the scan interval is $\Delta t = 1\,\text{min}$ and scenario duration is 1000 scans (just over 16.5 hrs).

The spawning event for each scenario type is initiated at the 399^{th} scan ($t = 6.65\,\text{hrs}$). For the RB explosion, this means that at scan 399, each new fragment shares the same position as the

[3] orbitaldebris.jsc.nasa.gov. odqnv18i3.pdf (pdf). [online] Available at: https://orbitaldebris.jsc.nasa.gov/quarterly-news/pdfs/odqnv18i3.pdf [Accessed 11 Sep. 2014].

Table 5.1: RSO Initial Conditions

Parameter	RB	QB50 LV
Epoch Time (Julian Date)	2457410.344	2457323.882
Semimajor Axis (km)	7115.626	6781.278
Eccentricity	1.4457×10^{-2}	6.833×10^{-4}
Inclination (deg)	96.449	51.644
Right Asc. of Node (deg)	37.231	139.310
Arg. of Perigee (deg)	107.633	88.039
True Anomaly (deg)	304.686	340.332

RB, yet varying changes in velocity are instantaneously imparted upon each fragment. It is only at scan 400 ($t = 6.67$ hrs) that the fragments become observable. Similarly for QB50 deployment, the first CubeSat is deployed at scan 399, but is not observable until scan 400. This method is sequentially implemented until all CubeSats have been deployed, i.e., it is not until scan 694 that all CubeSats are observable[4]. Furthermore, it is assumed that the trajectory and probability of survival of the original RB and QB50 launch vehicle are unaffected by explosion or the deployment of CubeSats, respectively. Hence, the number of objects increases from 1 to 7 for the RB scenario and from 1 to 51 for the QB50 sceanrio.

5.1.1 Dynamics

Model error is simulated via discrepancy between truth simulation and filter dynamics models. All models use two-body dynamics that include Earth gravity potential and drag perturbations. Essentially, the only difference is that the truth simulation's gravity perturbations model includes up to the J_3 zonal harmonic, whereas the filter's gravity perturbations model only includes up to the J_2 zonal harmonic. Single-object states are defined as $\mathrm{X} = [x, y, z, \dot{x}, \dot{y}, \dot{z}]^T$. Dynamics model parameters are presented in Table 5.2.

Both truth simulation and filtering uses the CU-TurboProp orbit propagation package [40] with an embedded Runge-Kutta 8(7) integrator with a relative tolerance of 10^{-12}. Process noise is modeled as state noise compensation [86] where the process noise covariance Q_{b} and process noise

[4] Note that CubeSats are not deployed at every scan time between 399 and 693.

Table 5.2: Propagation for Observation Simulation and Filter Dynamics

Parameter	Truth Model	Filter Model
Gravity Perturbation	GGM02C [85]	GGM02C [85]
Spherical Harmonic Degree/Order	3×0	2×0
Area-to-Mass Ratio (m^2/kg)	0.01	0.01
Drag Coefficient C_d	2.0	2.0
Process Noise σ_Q (km/s^2)	N/A	10^{-8}(RB), 10^{-9}(QB50)

transition matrix Γ are given by

$$Q_{\mathrm{b}} = \sigma_Q^2 \Gamma(t_k, t_{k-1}) \Gamma^T(t_k, t_{k-1}), \tag{5.1}$$

$$\Gamma(t_k, t_{k-1}) = \begin{bmatrix} 0.5(t_k - t_{k-1})^2 \mathbf{1}_3 & (t_k - t_{k-1}) \mathbf{1}_3 \end{bmatrix}^T. \tag{5.2}$$

Also, position and velocity errors are placed on initial a priori objects states with standard deviations $\sigma_{\mathrm{pos}} = 100\,\mathrm{m}$ and $\sigma_{\mathrm{vel}} = 1\,\mathrm{m/s}$, respectively, for both scenarios.

For the CubeSat scenario, a spawn component covariance $Q_{\mathrm{T},k-1}^{(QB50)}$, for filter use in (4.24), is developed by running one-hundred Monte Carlo simulations of the full QB50 deployment and compiling deviation vectors $d_{\mathrm{X}}^{(i)}$ formed as

$$d_{\mathrm{X}}^{(i)} = [X_{iss,k} - X_{CubeSat,k}^{(i)}]^T, \tag{5.3}$$

where $X_{iss,k}$ denotes the state of the ISS at the first time k the i^{th} deployed CubeSat is observable, $X_{CubeSat,k}^{(i)}$ denotes the state of the i^{th} deployed CubeSat at the same time, and states consist of 3-dimensional position and velocity vectors, i.e., $X = [x, y, z, \dot{x}, \dot{y}, \dot{z}]^T$. The deviation vectors are compiled to form a $(50 \cdot 100) \times 6$ array d_{X} for which the column-wise variance is taken and used to form a diagonal spawn component covariance as

$$Q_{\mathrm{T},k-1}^{(QB50)} = \mathrm{diag}\big(\mathrm{var}(d_{\mathrm{X}})\big). \tag{5.4}$$

For the RB scenario, a spawn component covariance $Q_{\mathrm{T},k-1}^{(RB)}$, for filter use in (4.24), is constructed by Monte Carlo simulations in which the NBM is used one hundred times to yield a large sample of Δv's. A Δv histogram using one-hundred bins is presented in Figure 5.1 where the vertical axis (number of objects in each bin) is log-scaled to reveal more detail. The standard deviation

of Δv's, $\sigma_{\Delta v}$, is used to construct $Q^{(RB)}_{\mathrm{T},k-1}$ as

$$Q^{(RB)}_{\mathrm{T},k-1} = \begin{bmatrix} (\sigma_{\Delta v} \cdot \Delta t)^2 \mathbf{1}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \sigma^2_{\Delta v} \mathbf{1}_3 \end{bmatrix}, \tag{5.5}$$

where Δt is the simulated scan interval, $\mathbf{1}_n$ denotes the $n \times n$ identity matrix, and $\mathbf{0}_n$ denotes the $n \times n$ zero matrix. It is noted that the standard deviation $\sigma_{\Delta v}$ is taken from a highly non-Gaussian distribution. For potential follow study, it is suggested that a more appropriate spawn model that will include a Gaussian mixtures approximation of the distribution represented in Figure 5.1.

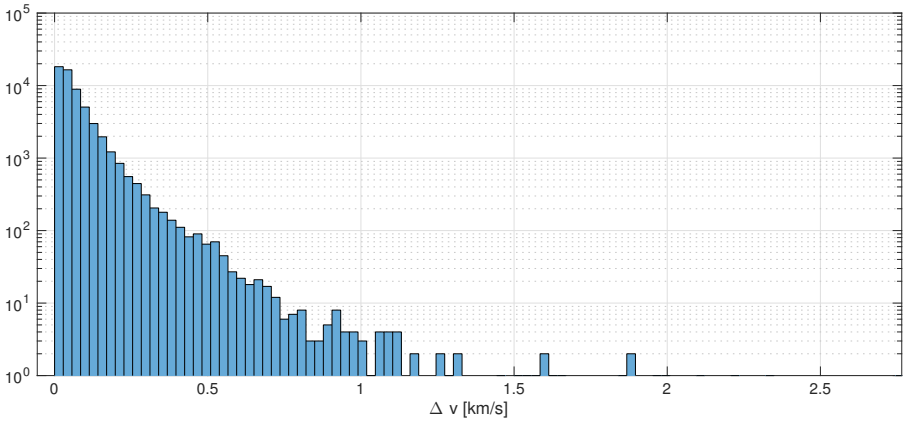


Figure 5.1: Histogram of RB fragment Δv's generated with the NASA breakup model.

5.1.2 Measurements

Simulations in this work assume a homogeneous sensor network consisting of seventy-two identical sensors distributed about the surface of the Earth as illustrated in Figure 5.2, providing full coverage of both scenarios with no missed detections or clutter returns. This is done as a prototyping step, as future work will move towards a more realistic sensor network model.

At each scan k, only one sensor collects range ρ, range-rate $\dot{\rho}$, azimuth β, and elevation $e\ell$ measurements of the observable objects, generating a set of measurements $Z_k = \{z_1, \ldots, z_m\}$, where each $z \in Z_k$ is given as

$$z = [\rho, \dot{\rho}, \beta, e\ell]^T. \tag{5.6}$$

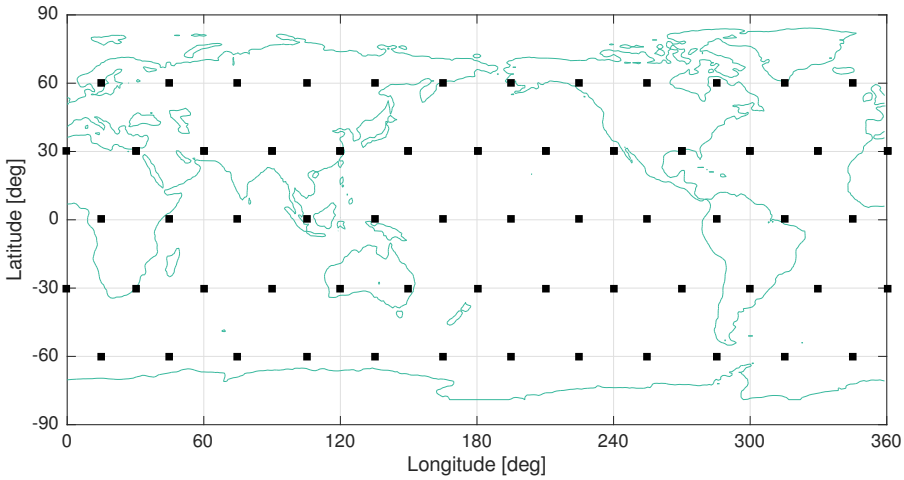


Figure 5.2: Each ■ indicates the location of a sensor within a homogeneous network.

Each $z \in Z_k$ includes independent Gaussian white noise on each component with standard deviations $\sigma_\rho = 15\,\text{m}, \sigma_{\dot{\rho}} = 1\,\text{mm/s}$, and $\sigma_\beta = \sigma_{e\ell} = 1^\circ$.

Each sensor is assumed to have an azimuth interval of $0^\circ \leq \beta \leq 360^\circ$, an elevation interval of $0^\circ \leq e\ell \leq 90^\circ$, a maximum range $\rho_{max} \approx 4500\,\text{km}$, and a FOV volume V_{FOV} computed via

$$V_{\text{FOV}} = \frac{2}{3}\pi\rho_{max}^3. \tag{5.7}$$

5.1.3 Filter Configuration

Filter configurations for the RB and QB50 simulations are presented in Table 5.3. Pruning ε and merging δ thresholds and the maximum number of (posterior) GM components n_{max} are considered tuning parameters, allowing for some control over tractability for a given implementation: see [89] for details on their implementation. The maximum number of (allowable) objects N_{max} is used to circumvent infinitely tailed cardinality distribution issues [96], i.e., the inner summation of Eq. (4.4) becomes a sum from $m \geq j$ to N_{max}, $\sum_{m\geq j}^{N_{max}}$. Though object death is not simulated, and there are no missed detections, the probabilities of survival and detection are set to 0.99 as opposed to 1.0, to avoid numerical issues.

In a real-world application of the proposed spawn model configured filter, pre-analysis for a given scenario would be required to determine appropriate values for spawn rates and probabilities.

Table 5.3: GM-CPHD filter configurations

Parameter	RB	QB50
Prune Threshold (ε)	10^{-5}	10^{-5}
Merge Threshold (δ)	4	4
Maximum Number of GM components (n_{max})	500	500
Maximum Number of Targets (N_{max})	20	100
Survival Probability (p_{S})	0.99	0.99
Detection Probability (p_{D})	0.99	0.99
Spawn Probability (p_{T})	3.115×10^{-3}	1.042×10^{-2}
Spawn Rate (λ_{T})	6	1
Position/Velocity OSPA Cutoff (c)	100 (m) / 1 (m/s)	100 (m) / 1 (m/s)

For instance, in the case of the anticipated collision of two RSOs, a probability of spawning could be determined by performing a conjunction assessment and a spawn rate could be deduced by leveraging trajectory and physical characteristic information for each RSO. Developing such methods for determining spawn model parameters is beyond the scope of the presented work, hence, it is assumed that such prior-analysis has been performed. For the presented simulations, Poisson spawn rate λ_{T} is determined by the true average number of spawned objects per each spawning event in a given simulation. The probability of spawning p_{T} is determined by a method that takes into consideration the total number of filter steps per truly existing objects and the true number of spawning events.

Example 1: Consider a scenario that includes only 3 time steps. At step 1, only 1 object truly exists. At step 2, a second object is spawned, so 2 objects truly exist at steps 2 and 3. Assuming a spawned object that cannot spawn new objects itself at the time of its spawning, then the total number of time steps that spawning can occur is $N_{\mathrm{steps}} = 4$. Assuming the number of true spawning events is $N_{\mathrm{spawn}} = 1$, then the probability of spawning for this example scenario is given by $p_{\mathrm{T}} = N_{\mathrm{spawn}}/N_{\mathrm{steps}} = 0.25$.

5.2 Results

5.2.1 QB50

MAP cardinality estimates and the number of posterior GM components (i.e., the number of mixture components used to represent the measurement updated intensity) for the QB50 simulation are presented in Figure 5.3. The cardinality estimates, plotted over the true number of objects for the scenario, indicate that the filter correctly estimated the number of objects at each time step throughout the simulation with no lag. It is expected that with a more realistically modeled simulation, the cardinality estimate plot will potentially lag behind the true cardinality plot due to missed detections (particularly if they occur at the time when CubeSats are first observable), clutter returns, observation gaps, increase in CubeSat number, etc., however, convergence on the true number of objects is anticipated with the development of a more robust spawn model designed to address such issues: the focus of future work.

The number of posterior GM components plot (bottom of Figure 5.3) serves to indicate computational effort for the given scenario. Until CubeSat deployment is initiated, only one component is passed from one time to the next, reflecting the presence of only one object. As expected, component number increases with time in the deployment phase. This increase is attributed to the proximity of CubeSats to each other and the LV shortly after deployments. Component number exhibits a downward trend after the deployment phase due to the CubeSats spreading out over time and the filter converging on more accurate single-object state estimates.

QB50 scenario OSPA position and velocity error plots are presented in Figure 5.4. Both plots indicate large errors at the beginning of the simulation due to the initial error placed on a priori object states. The filter converges on state error levels commensurate with observation errors until the CubeSat deployment phase begins, as expected. The OSPA errors follow a downward trend throughout the deployment phase as the filter converges on more accurate cardinality and single-object state estimates.

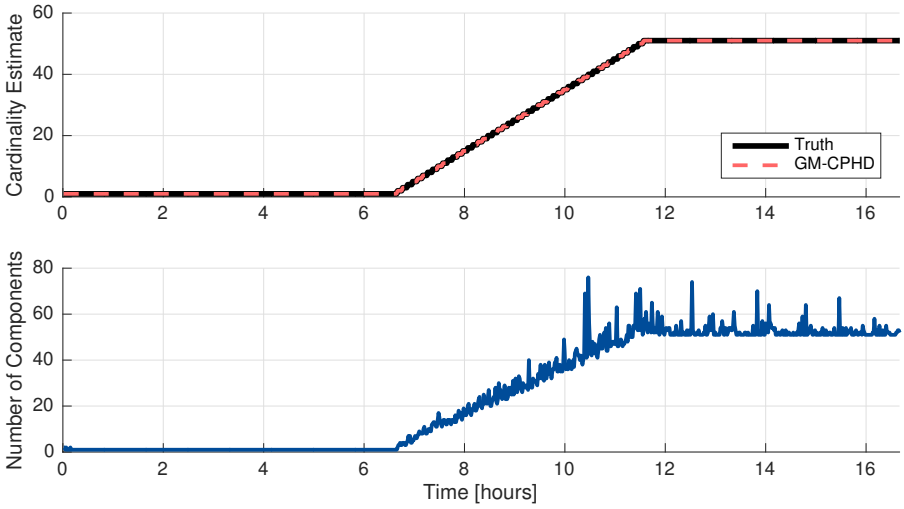


Figure 5.3: QB50 cardinality estimates (top) and posterior number of GM components (bottom).

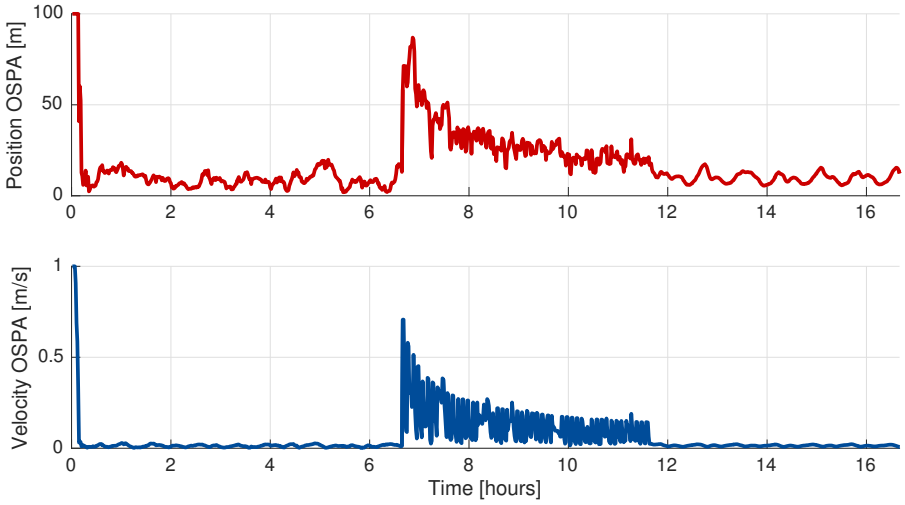


Figure 5.4: QB50 OSPA position (top) and velocity (bottom) errors.

5.2.2 Rocket Body Explosion

Cardinality estimate, component number, and OSPA position and velocity error plots for the QB50 simulation are presented in Figures 5.5 and 5.6, respectively, indicating filter behavior that follows suit with that presented by their QB50 simulation counterparts: the major differences stemming from the nature of the simulated scenarios.

In contrast to the QB50 scenario where a number of new objects are generated over a period

of time, the appearance of new objects occurs instantaneously, hence the sharp cardinality jump in the top of Figure 5.5. Cardinality estimates, again plotted over the true number of objects for the scenario, indicate that the filter correctly estimated the number of objects at each time step. Component number increases with cardinality, illustrating the filter's increased computational effort as more objects enter the scene. The large spike in component number and OSPA errors lasts for a short time, which is an anticipated behavior due to fragments starting off in close proximity, then separating quickly due to larger Δv's (relative to the CubeSat scenario).

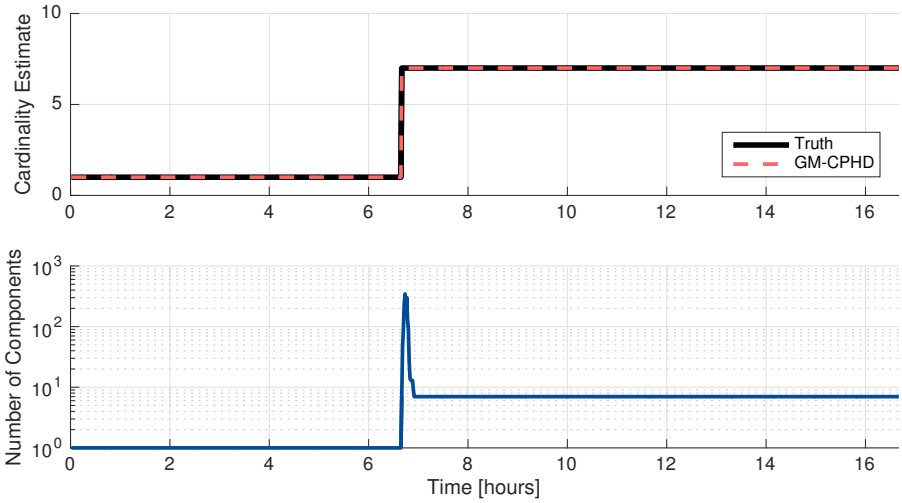


Figure 5.5: RB cardinality estimates (top) and posterior number of GM components (bottom).

Both plots in Figure 5.6 indicate large errors at the beginning of the simulation, again, due to the initial error placed on a priori object states. The filter converges on noise level errors until the time of breakup, as expected. The duration of increased error follows the period of increased component number in Figure 5.5, indicating the filter's need for more measurements before converging on noise level errors again. It is anticipated that the duration of increased error will increase with fragmentation event complexity, which future work will consider.

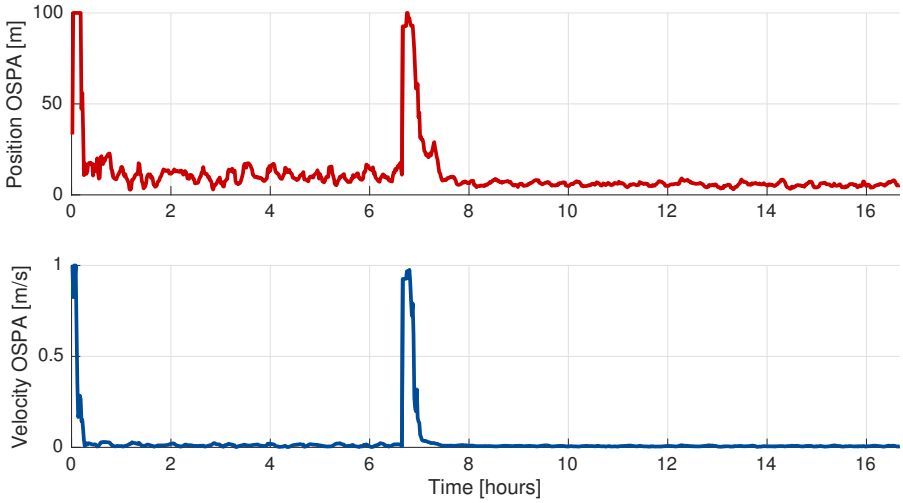


Figure 5.6: RB OSPA position (top) and velocity (bottom) errors.

Chapter 6

The GLMB Tracking Filter with Object-Spawning

This chapter presents a labeled RFS spawning model and a tractable multi-object filter for it. The labeled RFS spawning model is detailed in Section 6.1. In Section 6.2 the resulting prediction multi-object density at the next time, for a given GLMB at the current time, is derived. In Section 6.3, the method of GLMB approximation which matches first moment and cardinality is discussed, then utilized in the derivation of the multi-object filtering density at the next time. Implementation of the resulting GLMB recursion is detailed in Subsection 6.4. Simulations are presented in Section 6.5. The contents of this chapter are currently under review for publication in a journal.

6.1 Multi-object Labeled Spawning Model

Recall the labeling convention described in Section 2.2.3.1, where a label $\ell = (t, i)$ in the space $\mathbb{L}$ of labels at the current time k is an ordered pair, where the first term $t \leq k$ denotes time of birth, and the second term $i \in \mathbb{N}$ is a unique index distinguishing objects born at the same time. The label space at the next time was defined as $\mathbb{L}_+ = \mathbb{L} \cup \mathbb{B}_+$, where birth labels at the next time belong to the space $\mathbb{B}_+ = \{(k+1, i) : i \in \mathbb{N}\}$.

To encode ancestry information in the labels, the following labeling convention for spawned tracks is observed. An object spawned from a parent with label ℓ, at time $k+1$, has label $\varsigma = (\ell, k+1, i)$, where i is an index that distinguishes multiple objects simultaneously spawned by the same parent. As a result, spawn labels consist of an ancestral element, i.e., the parent's label, and a

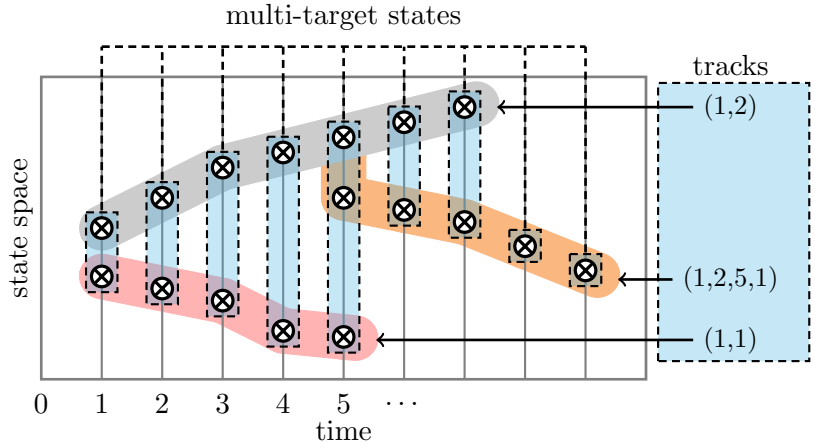


Figure 6.1: An example of label assignment for birth and spawn tracks. Two tracks are born at time 1 and are assigned labels $(1,1)$ and $(1,2)$. At time 5, a track is spawned from track $(1,2)$ and is assigned label $(1,2,5,1)$.

non-ancestral element that distinguishes multiple spawned objects originating simultaneously from the same parent. Hence, given the label space $\mathbb{L}_k$ for objects at the current time, the label space $\mathbb{T}_{k+1}$ for tracks spawned at the next time is given by $\mathbb{T}_{k+1} = \mathbb{L}_k \times \{k+1\} \times \mathbb{N}$.

Reverting to the convention of letting the symbol + denote the "next time" index, the same construction in [95] is followed by letting $\mathbb{T}_+$ denote the label space for objects spawned at the next time such that the label space at the next time is $\mathbb{L}_+ = \mathbb{L} \cup \mathbb{T}_+ \cup \mathbb{B}_+$. Note that $\mathbb{L}$, $\mathbb{T}_+$, and $\mathbb{B}_+$ are mutually disjoint, i.e., $\mathbb{L} \cap \mathbb{T}_+ = \mathbb{L} \cap \mathbb{B}_+ = \mathbb{T}_+ \cap \mathbb{B}_+ = \emptyset$. Hence, surviving, spawn, and birth objects can be distinguished from their labels. Fig. 6.1, modeled after [94, Fig.1] in the interest of consistency, illustrates label assignment to birth and spawn tracks.

The elements of a given current multi-object state $\mathbf{X}$ spawn new objects independently of each other. In addition, the set $\mathbf{U}$ of objects spawned at the next time by a single-object state $\mathbf{x} \triangleq (x, \ell)$, is an LMB[1] with parameter set $\{(p_{\mathrm{T}}(\mathbf{x};\varsigma), f_{\mathrm{T},+}(\cdot|\mathbf{x};\varsigma)) : \varsigma \in \mathbb{T}_+(\mathcal{L}(\mathbf{x}))\}$, where

$$\mathbb{T}_+(\ell) \triangleq \{(\ell, k+1)\} \times \{1 : M_\ell\} \tag{6.1}$$

is a finite subset set of $\mathbb{T}_+$. In other words, for each LMB component $\varsigma \in \mathbb{T}_+(\mathcal{L}(\mathbf{x}))$, the state $\mathbf{x}$ either spawns a state (x_+, ς) with probability $p_{\mathrm{T}}(\mathbf{x};\varsigma)$ and probability density $f_{\mathrm{T},+}(x_+|\mathbf{x};\varsigma)$, or it

[1] It is possible to derive a GLMB based spawning model, but an LMB is presented for compactness

does not with probability $q_{\mathrm{T}}(\mathbf{x};\varsigma) = 1 - p_{\mathrm{T}}(\mathbf{x};\varsigma)$. The density of the set of objects spawned from $\mathbf{x}$ can be written as

$$\mathbf{f}_{\mathrm{T},+}(\mathbf{U}|\mathbf{x}) = \Delta(\mathbf{U})1_{\mathbb{T}_+(\mathcal{L}(\mathbf{x}))}(\mathcal{L}(\mathbf{U}))[\Phi_{\mathrm{T},+}(\mathbf{U}|\mathbf{x};\cdot)]^{\mathbb{T}_+(\mathcal{L}(\mathbf{x}))}, \tag{6.2}$$

where

$$\Phi_{\mathrm{T},+}(\mathbf{U}|\mathbf{x};\varsigma) = \sum_{(x_+,\ell_+)\in\mathbf{U}} \delta_\varsigma(\ell_+)p_{\mathrm{T}}(\mathbf{x};\varsigma)f_{\mathrm{T},+}(x_+|\mathbf{x};\varsigma) + [1 - 1_{\mathcal{L}(\mathbf{U})}(\varsigma)]q_{\mathrm{T}}(\mathbf{x};\varsigma). \tag{6.3}$$

Since $\mathbf{X}$ has distinct labels, the LMB label sets $\mathbb{T}_+(\mathcal{L}(\mathbf{x}))$ for all $\mathbf{x} \in \mathbf{X}$ are mutually disjoint, and the set of possible labels spawned from $\mathbf{X}$ is the disjoint union

$$\mathbb{T}_+(\mathcal{L}(\mathbf{X})) = \biguplus_{\mathbf{x}\in\mathbf{X}} \mathbb{T}_+(\mathcal{L}(\mathbf{x})).$$

Note that when a labeled set $\mathbf{V}$ is spawned from $\mathbf{X}$, $\mathcal{L}(\mathbf{V}) \subseteq \mathbb{T}_+(\mathcal{L}(\mathbf{X}))$, i.e., $1_{\mathbb{T}_+(\mathcal{L}(\mathbf{X}))}(\mathcal{L}(\mathbf{V})) = 1$, otherwise $\mathcal{L}(\mathbf{V}) \nsubseteq \mathbb{T}_+(\mathcal{L}(\mathbf{X}))$, i.e., $1_{\mathbb{T}_+(\mathcal{L}(\mathbf{X}))}(\mathcal{L}(\mathbf{V})) = 0$. Hence, the inclusion $1_{\mathbb{T}_+(\mathcal{L}(\mathbf{X}))}(\mathcal{L}(\mathbf{V}))$ is an indicator of whether $\mathbf{V}$ is spawned by $\mathbf{X}$ or not. Moreover, if $\mathbf{V}$ is not spawned from $\mathbf{X}$**,** then the spawning density $\mathbf{f}_{\mathrm{T},+}(\mathbf{V}|\mathbf{X}) = 0$. On the other hand if $\mathbf{V}$ is spawned from $\mathbf{X}$, then

$$\mathbf{V} = \biguplus_{\mathbf{x}\in\mathbf{X}} \mathbf{V} \cap (\mathbb{X} \times \mathbb{T}_+(\mathcal{L}(\mathbf{x})),$$

and since $\mathbf{V} \cap (\mathbb{X} \times \mathbb{T}_+(\mathcal{L}(\mathbf{x}))$ is the set of objects spawned by $\mathbf{x}$, it follows from the FISST fundamental convolution theorem [63], and arguments presented in [95] that the spawning density

$$\mathbf{f}_{\mathrm{T},+}(\mathbf{V}|\mathbf{X}) = \prod_{\mathbf{x}\in\mathbf{X}} \mathbf{f}_{\mathrm{T},+}(\mathbf{V} \cap (\mathbb{X} \times \mathbb{T}_+(\mathcal{L}(\mathbf{x}))|\mathbf{x}).$$

Hence, $\mathbf{f}_{\mathrm{T},+}(\mathbf{V}|\mathbf{X})$ can be written as

$$\mathbf{f}_{\mathrm{T},+}(\mathbf{V}|\mathbf{X}) = 1_{\mathbb{T}_+(\mathcal{L}(\mathbf{X}))}(\mathcal{L}(\mathbf{V})) \prod_{\mathbf{x}\in\mathbf{X}} \mathbf{f}_{\mathrm{T},+}(\mathbf{V} \cap (\mathbb{X} \times \mathbb{T}_+(\mathcal{L}(\mathbf{x}))|\mathbf{x}). \tag{6.4}$$

Substituting (6.2) into the above equation and noting that when $1_{\mathbb{T}_+(\mathcal{L}(\mathbf{X}))}(\mathcal{L}(\mathbf{V})) = 1$,

$$\begin{aligned} 1_{\mathbb{T}_+(\mathcal{L}(\mathbf{x}))}(\mathcal{L}(\mathbf{V} \cap (\mathbb{X} \times \mathbb{T}_+(\mathcal{L}(\mathbf{x}))))) &= 1, \\ \prod_{\mathbf{x}\in\mathbf{X}} \Delta(\mathbf{V} \cap (\mathbb{X} \times \mathbb{T}_+(\mathcal{L}(\mathbf{x}))) &= \Delta(\mathbf{V}), \end{aligned}$$

we have

$$\mathbf{f}_{\mathrm{T},+}(\mathbf{V}|\mathbf{X}) = \Delta(\mathbf{V})1_{\mathbb{T}_+(\mathcal{L}(\mathbf{X}))}(\mathcal{L}(\mathbf{V}))\left[\Phi_{\mathrm{T},+}(\mathbf{V}|\cdot)\right]^{\mathbf{X}}, \tag{6.5}$$

where

$$\Phi_{\mathrm{T},+}(\mathbf{V}|\mathbf{x}) = \left[\Phi_{\mathrm{T},+}(\mathbf{V}\cap(\mathbb{X}\times\mathbb{T}_+(\mathcal{L}(\mathbf{x}))|\mathbf{x};\cdot)\right]^{\mathbb{T}_+(\mathcal{L}(\mathbf{x}))}. \tag{6.6}$$

The multi-object state at the next time $\mathbf{X}_+ = \mathbf{X}_{\mathrm{S},+} \cup \mathbf{X}_{\mathrm{T},+} \cup \mathbf{X}_{\mathrm{B},+}$ is the superposition of surviving objects $\mathbf{X}_{\mathrm{S},+} = \mathbf{X}_+ \cap (\mathbb{X}\times\mathbb{L})$, birth objects $\mathbf{X}_{\mathrm{B},+} = \mathbf{X}_+ \cap (\mathbb{X}\times\mathbb{B}_+)$ and spawned objects $\mathbf{X}_{\mathrm{T},+} = \mathbf{X}_+ \cap (\mathbb{X}\times\mathbb{T}_+)$. Since the label spaces $\mathbb{L}$, $\mathbb{T}_+$, and $\mathbb{B}_+$ are mutually disjoint, it follows that $\mathbf{X}_{\mathrm{S},+}$, $\mathbf{X}_{\mathrm{T},+}$, and $\mathbf{X}_{\mathrm{B},+}$ are also mutually disjoint. Further, using the conditional independence of $\mathbf{X}_{\mathrm{S},+}$, $\mathbf{X}_{\mathrm{T},+}$, and $\mathbf{X}_{\mathrm{B},+}$, it follows from the FISST fundamental convolution theorem [63], [95] that the multi-object transition kernel is given by

$$\mathbf{f}_+(\mathbf{X}_+|\mathbf{X}) = \mathbf{f}_{\mathrm{S},+}(\mathbf{X}_{\mathrm{S},+}|\mathbf{X})\mathbf{f}_{\mathrm{T},+}(\mathbf{X}_{\mathrm{T},+}|\mathbf{X})\mathbf{f}_{\mathrm{B},+}(\mathbf{X}_{\mathrm{B},+}). \tag{6.7}$$

6.2 Multi-object prediction with spawning

In general, for a multi-object transition density with spawning (6.7), the GLMB family is not necessarily closed under the Chapman-Kolmogorov equation

$$\boldsymbol{\pi}(\mathbf{X}_+) = \int \mathbf{f}_+(\mathbf{X}_+|\mathbf{X})\boldsymbol{\pi}(\mathbf{X})\delta\mathbf{X}. \tag{6.8}$$

Proposition 8. *If the current multi-object filtering density is GLMB of the form* (2.21)*, then the multi-object prediction density formed by surviving, birth and spawning processes is given by*

$$\boldsymbol{\pi}(\mathbf{X}_+) = \Delta(\mathbf{X}_+)\sum_{I,\xi} w_+^{(I,\xi)}(\mathcal{L}(\mathbf{X}_+))p^{(I,\xi)}(\mathbf{X}_+), \tag{6.9}$$

where

$$\begin{aligned} w_+^{(I,\xi)}(\mathcal{L}(\mathbf{X}_+)) &= w^{(I,\xi)}1_I(\mathcal{L}(\mathbf{X}_{\mathrm{S},+}))1_{\mathbb{T}_+(\mathcal{L}(\mathbf{X}))}(\mathcal{L}(\mathbf{X}_{\mathrm{T},+}))1_{\mathbb{B}_+}(\mathcal{L}(\mathbf{X}_{\mathrm{B},+})) \\ &\quad\times[1-r_{\mathrm{B},+}]^{\mathbb{B}_+-\mathcal{L}(\mathbf{X}_{\mathrm{B},+})}[r_{\mathrm{B},+}]^{\mathcal{L}(\mathbf{X}_{\mathrm{B},+})}, \end{aligned} \tag{6.10}$$

$$p^{(I,\xi)}(\mathbf{X}_+) = [p_{\mathrm{B},+}]^{\mathbf{X}_{\mathrm{B},+}}\prod_{\ell\in I}\left\langle \Phi_{\mathrm{S},+}(\mathbf{X}_{\mathrm{S},+}|\cdot,\ell)\Phi_{\mathrm{T},+}(\mathbf{X}_{\mathrm{T},+}|\cdot,\ell), p^{(\xi)}(\cdot,\ell)\right\rangle, \tag{6.11}$$

$\mathbf{X}_{\mathrm{S},+} = \mathbf{X}_+ \cap (\mathbb{X}\times\mathbb{L})$, $\mathbf{X}_{\mathrm{T},+} = \mathbf{X}_+ \cap (\mathbb{X}\times\mathbb{T}_+)$, *and* $\mathbf{X}_{\mathrm{B},+} = \mathbf{X}_+ \cap (\mathbb{X}\times\mathbb{B}_+)$.

Proof. Using the Chapman-Kolmogorov equation (6.8) and (6.7) with $\mathbf{f}_{\mathrm{S},+}(\mathbf{X}_{\mathrm{S},+}|\mathbf{X})$ from (2.25) and $\mathbf{f}_{\mathrm{T},+}(\mathbf{X}_{\mathrm{T},+}|\mathbf{X})$ from (6.2), we have

$$\begin{aligned}\boldsymbol{\pi}(\mathbf{X}_+) &= \mathbf{f}_{\mathrm{B},+}(\mathbf{X}_{\mathrm{B},+}) \int \Delta(\mathbf{X}_{\mathrm{S},+}) 1_{\mathcal{L}(\mathbf{X})}(\mathcal{L}(\mathbf{X}_{\mathrm{S},+})) \left[\Phi_{\mathrm{S},+}(\mathbf{X}_{\mathrm{S},+}|\cdot)\right]^{\mathbf{X}} \\ &\quad \times \Delta(\mathbf{X}_{\mathrm{T},+}) 1_{\mathbb{T}_+(\mathcal{L}(\mathbf{X}))}(\mathcal{L}(\mathbf{X}_{\mathrm{T},+})) \left[\Phi_{\mathrm{T},+}(\mathbf{X}_{\mathrm{T},+}|\cdot)\right]^{\mathbf{X}} \\ &\quad \times \Delta(\mathbf{X}) \sum_{I,\xi} w^{(I,\xi)} \delta_I(\mathcal{L}(\mathbf{X})) \left[p^{(\xi)}\right]^{\mathbf{X}} \delta\mathbf{X}, \qquad (6.12) \\ &= \Delta(\mathbf{X}_{\mathrm{S},+}) \Delta(\mathbf{X}_{\mathrm{T},+}) \mathbf{f}_{\mathrm{B},+}(\mathbf{X}_{\mathrm{B},+}) \\ &\quad \times \sum_{I,\xi} \int \Delta(\mathbf{X}) w^{(I,\xi)} \delta_I(\mathcal{L}(\mathbf{X})) 1_{\mathcal{L}(\mathbf{X})}(\mathcal{L}(\mathbf{X}_{\mathrm{S},+})) 1_{\mathbb{T}_+(\mathcal{L}(\mathbf{X}))}(\mathcal{L}(\mathbf{X}_{\mathrm{T},+})) \\ &\quad \times \left[\Phi_{\mathrm{S},+}(\mathbf{X}_{\mathrm{S},+}|\cdot) \Phi_{\mathrm{T},+}(\mathbf{X}_{\mathrm{T},+}|\cdot) p^{(\xi)}\right]^{\mathbf{X}} \delta\mathbf{X}, \qquad (6.13) \\ &= \Delta(\mathbf{X}_{\mathrm{S},+}) \Delta(\mathbf{X}_{\mathrm{T},+}) \mathbf{f}_{\mathrm{B},+}(\mathbf{X}_{\mathrm{B},+}) \sum_{I,\xi} \sum_{J \in \mathcal{F}(\mathbb{L})} w^{(I,\xi)} \delta_I(J) 1_J(\mathcal{L}(\mathbf{X}_{\mathrm{S},+})) 1_{\mathbb{T}_+(\mathcal{L}(\mathbf{X}))}(\mathcal{L}(\mathbf{X}_{\mathrm{T},+})) \\ &\quad \times \prod_{\ell \in I} \left\langle \Phi_{\mathrm{S},+}(\mathbf{X}_{\mathrm{S},+}|\cdot,\ell) \Phi_{\mathrm{T},+}(\mathbf{X}_{\mathrm{T},+}|\cdot,\ell), p^{(\xi)}(\cdot,\ell) \right\rangle, \qquad (6.14)\end{aligned}$$

where the last line follows from [95, Lemma 3]. Using $\Delta(\mathbf{X}_+) = \Delta(\mathbf{X}_{\mathrm{S},+})\Delta(\mathbf{X}_{\mathrm{T},+})\Delta(\mathbf{X}_{\mathrm{B},+})$, $\mathbf{f}_{\mathrm{B},+}(\mathbf{X}_{\mathrm{B},+})$ from (2.26), and noting that the only non-zero inner summand occurs when $I = J$, we have (6.9). □

Implicit in (6.9) is that, even though the survival and spawn RFSs are mutually disjoint due to their labels (see (6.7)), they are both conditioned on the same multi-object state $\mathbf{X}$. Further, the objects spawned by a state $\mathbf{x} \in \mathbf{X}$ and its state at the next time (if survived) are all distinct, but conditioned on $\mathbf{x}$.

6.3 Multi-Object Update with Spawning

Since the predicted multi-object density is not a GLMB, the updated multi-object density

$$\boldsymbol{\pi}_+(\mathbf{X}_+|Z_+) = \frac{\boldsymbol{\pi}(\mathbf{X}_+) g(Z_+|\mathbf{X}_+)}{\int \boldsymbol{\pi}(\mathbf{X}) g(Z_+|\mathbf{X}) \delta\mathbf{X}}. \qquad (6.15)$$

with the standard multi-object likelihood, is also not a GLMB. One strategy of using the GLMB filter to track with spawnings is to approximate the multi-object prediction density (6.9) by a

GLMB prior to performing a measurement update. A more prudent approach is preferred in this work whereby the GLMB approximation is performed on the updated multi-object density to reduce information loss, albeit at the cost of increased complexity. Such approximation can be achieved by finding a GLMB that matches the multi-object filtering density in the first moment and cardinality, which is done as follows.

An arbitrary labeled multi-object density on $\mathcal{F}(\mathbb{X}\times\mathbb{L})$ can be writen in the form [70, Proposition 3]

$$\boldsymbol{\pi}(\mathbf{X}) = \Delta(\mathbf{X})\sum_{c\in\mathbb{C}} w^{(c)}(\mathcal{L}(\mathbf{X}))p^{(c)}(\mathbf{X}), \tag{6.16}$$

where $\mathbb{C}$ is a discrete index set, the weights $w^{(c)}(\cdot)$ satisfy (2.22), and with $n = |\mathbf{X}|$,

$$\int p^{(c)}\left(\{(x_1,\ell_1),\ldots,(x_n,\ell_n)\}\right) d(x_1,\ldots,x_n) = 1.$$

Moreover, it was shown in [70, Proposition 3], that such a labeled multi-object density can be approximated by the GLMB

$$\hat{\boldsymbol{\pi}}(\mathbf{X}) = \Delta(\mathbf{X}) \sum_{(c,I)\in\mathbb{C}\times\mathcal{F}(\mathbb{L})} \delta_I(\mathcal{L}(\mathbf{X}))\hat{w}^{(c,I)}\left[\hat{p}^{(c,I)}\right]^{\mathbf{X}}, \tag{6.17}$$

where

$$\hat{w}^{(c,I)} = w^{(c)}(I), \tag{6.18}$$

$$\hat{p}^{(c,I)}(x,\ell) = 1_I(\ell)p^{(c)}_{I-\{\ell\}}(x,\ell), \tag{6.19}$$

$$p^{(c)}_{\{\ell_1,\ldots,\ell_n\}}(x,\ell) = \int p^{(c)}(\{(x,\ell),(x_1,\ell_1),\ldots,(x_n,\ell_n)\})d(x_1,\ldots,x_n). \tag{6.20}$$

A salient feature of this approximation method is that both the cardinality distribution and PHD of $\boldsymbol{\pi}$ are preserved. Additionally, note that $\mathbb{C}$ can take the form of any discrete index set, including the set of indices for the Cartesian product of a collection of finites subsets of some label space and an association history space, i.e., letting $\mathbb{C} = \mathcal{F}(\mathbb{L})\times\Xi$ is possible.

The exact form of the multi-object filtering density at the next time, and its GLMB approximation as per the result above, is given below in Proposition 2.

Proposition 9. *If the current filtering density is GLMB of form* (2.21) *and given the multi-object likelihood* (2.28)*, then the multi-object filtering density at the next time is given by*

$$\boldsymbol{\pi}_+(\mathbf{X}_+|Z_+) \propto \Delta(\mathbf{X}_+) \sum_{I,\xi,\theta_+} w_+^{(I,\xi)}(\mathcal{L}(\mathbf{X}_+)) \left[p_{\mathrm{B},+}\psi_+^{(\theta_+)}(\cdot|Z_+)\right]^{\mathbf{X}_{\mathrm{B},+}} p_+^{(I,\xi,\theta_+)}(\mathbf{X}_{\mathrm{S},+} \cup \mathbf{X}_{\mathrm{T},+}|Z_+), \quad (6.21)$$

where $I \in \mathcal{F}(\mathbb{L})$*,* $\xi \in \Xi$*,* $\theta_+ \in \Theta_+(\mathcal{L}(\mathbf{X}_+))$*,* $\mathbf{X}_{\mathrm{B},+} = \mathbf{X}_+ \cap (\mathbb{X} \times \mathbb{B}_+)$*,* $\mathbf{X}_{\mathrm{S},+} = \mathbf{X}_+ \cap (\mathbb{X} \times \mathbb{L})$*,* $\mathbf{X}_{\mathrm{T},+} = \mathbf{X}_+ \cap (\mathbb{X} \times \mathbb{T}_+)$*, and*

$$\begin{aligned} p_+^{(I,\xi,\theta_+)}(\mathbf{X}_{\mathrm{S},+} \cup \mathbf{X}_{\mathrm{T},+}|Z_+) = &\left[\psi_+^{(\theta_+)}(\cdot|Z_+)\right]^{\mathbf{X}_{\mathrm{S},+}\cup\mathbf{X}_{\mathrm{T},+}} \\ &\times \prod_{\ell\in I} \left\langle \Phi_{\mathrm{S},+}(\mathbf{X}_{\mathrm{S},+}|\cdot,\ell)\Phi_{\mathrm{T},+}(\mathbf{X}_{\mathrm{T},+}|\cdot,\ell), p^{(\xi)}(\cdot,\ell)\right\rangle. \end{aligned} \quad (6.22)$$

Moreover, it can be approximated by the GLMB given by

$$\begin{aligned} \hat{\boldsymbol{\pi}}_+(\mathbf{X}_+|Z_+) = \Delta(\mathbf{X}_+) &\sum_{I,\xi,I_+,\theta_+} \delta_{I_+}(\mathcal{L}(\mathbf{X}_+))\hat{w}_+^{(I,\xi,I_+,\theta_+)}(Z_+) \left[p_{\mathrm{B},+}\psi_+^{(\theta_+)}(\cdot|Z_+)\right]^{\mathbf{X}_{\mathrm{B},+}} \\ &\times \frac{\left[\hat{p}_+^{(I,\xi,I_+,\theta_+)}(\cdot|Z_+)\right]^{\mathbf{X}_{\mathrm{S},+}\cup\mathbf{X}_{\mathrm{T},+}}}{\left[\bar{p}_+^{(I,\xi,I_+,\theta_+)}(\cdot|Z_+)\right]^{I_+}}, \end{aligned} \quad (6.23)$$

$$\hat{w}_+^{(I,\xi,I_+,\theta_+)}(Z_+) = \frac{w_+^{(I,\xi)}(I_+)\left[\bar{p}_+^{(I,\xi,I_+,\theta_+)}(\cdot|Z_+)\right]^{I_+}}{\sum_{I,\xi,I_+,\theta_+} w_+^{(I,\xi)}(I_+)\left[\bar{p}_+^{(I,\xi,I_+,\theta_+)}(\cdot|Z_+)\right]^{I_+}}, \quad (6.24)$$

$$\hat{p}_+^{(I,\xi,I_+,\theta_+)}(x_+,\ell_+|Z_+) = 1_{I_+}(\ell_+)\int p_+^{(I,\xi,\theta_+)}\left(\{(x_+,\ell_+),(x_{1:n,+},\ell_{1:n,+})\}|Z_+\right) d(x_{1:n,+}), \quad (6.25)$$

$$\begin{aligned} \bar{p}_+^{(I,\xi,I_+,\theta_+)}(\ell_+|Z_+) = &1_{\mathbb{B}_+}(\ell_+)\left\langle p_{\mathrm{B},+}(\cdot,\ell_+), \psi_+^{(\theta_+)}(\cdot|Z_+)\right\rangle \\ &+ (1-1_{\mathbb{B}_+}(\ell_+))\left\langle \hat{p}_+^{(I,\xi,I_+,\theta_+)}(\cdot,\ell_+|Z_+), 1\right\rangle, \end{aligned} \quad (6.26)$$

which preserves the first moment and cardinality distribution, where $I_+ \in \mathbb{L}_+$*.*

Proof. With $g(Z_+|\mathbf{X}_+)$ from (2.28) and $\boldsymbol{\pi}(\mathbf{X}_+)$ from (6.9), we have

$$\boldsymbol{\pi}_+(\mathbf{X}_+|Z_+) \propto \boldsymbol{\pi}(\mathbf{X}_+)g(Z_+|\mathbf{X}_+), \tag{6.46}$$

$$= \Delta(\mathbf{X}_+)\sum_{I,\xi,\theta_+} w_+^{(I,\xi)}(\mathcal{L}(\mathbf{X}_+))p_+^{(I,\xi)}(\mathbf{X}_+)\left[\psi_+^{(\theta_+)}(\cdot|Z_+)\right]^{\mathbf{X}_+}, \tag{6.47}$$

$$= \Delta(\mathbf{X}_+)\sum_{I,\xi,\theta_+} w_+^{(I,\xi)}(\mathcal{L}(\mathbf{X}_+))\left[p_{\mathrm{B},+}\psi_+^{(\theta_+)}(\cdot|Z_+)\right]^{\mathbf{X}_{\mathrm{B},+}} \times \prod_{\ell\in I}\left\langle \Phi_{\mathrm{S},+}(\mathbf{X}_{\mathrm{S},+}|\cdot,\ell)\Phi_{\mathrm{T},+}(\mathbf{X}_{\mathrm{T},+}|\cdot,\ell), p^{(\xi)}(\cdot,\ell)\right\rangle \left[\psi_+^{(\theta_+)}(\cdot|Z_+)\right]^{\mathbf{X}_{\mathrm{S},+}\cup\mathbf{X}_{\mathrm{T},+}}, \tag{6.48}$$

$$= \Delta(\mathbf{X}_+)\sum_{I,\xi,\theta_+} w_+^{(I,\xi)}(\mathcal{L}(\mathbf{X}_+))\left[p_{\mathrm{B},+}\psi_+^{(\theta_+)}(\cdot|Z_+)\right]^{\mathbf{X}_{\mathrm{B},+}} p_+^{(I,\xi,\theta_+)}(\mathbf{X}_{\mathrm{S},+}\cup\mathbf{X}_{\mathrm{T},+}|Z_+). \tag{6.49}$$

Now apply the GLMB approximation (6.17) to (6.49), and note that determining the product of the marginals of the single-object birth densities encapsulated in $[p_{\mathrm{B},+}\psi_+^{(\theta_+)}(\cdot|Z_+)]^{\mathbf{X}_{\mathrm{B},+}}$ is redundant. Only the single-object densities encapsulated in $p_+^{(I,\xi,\theta_+)}(\mathbf{X}_{\mathrm{S},+}\cup\mathbf{X}_{\mathrm{T},+}|Z_+)$ require marginalization. Hence, applying the GLMB approximation from (6.17) gives

$$\hat{\boldsymbol{\pi}}_+(\mathbf{X}_+|Z_+) = \hat{C}\Delta(\mathbf{X}_+)\sum_{I,\xi,I_+,\theta_+}\delta_{I_+}(\mathcal{L}(\mathbf{X}_+))w_+^{(I,\xi)}(\mathcal{L}(\mathbf{X}_+))\left[p_{\mathrm{B},+}\psi_+^{(\theta_+)}(\cdot|Z_+)\right]^{\mathbf{X}_{\mathrm{B},+}} \times \left[\hat{p}^{(I,\xi,I_+,\theta_+)}(\cdot|Z_+)\right]^{\mathbf{X}_{\mathrm{S},+}\cup\mathbf{X}_{\mathrm{T},+}}, \tag{6.50}$$

where $I_+ \in \mathcal{F}(\mathbb{L}_+)$, $\hat{p}^{(I,\xi,I_+,\theta_+)}(\cdot|Z_+)$ is defined in (6.25), and

$$\hat{C}^{-1} = \sum_{I,\xi,I_+,\theta_+}\int\Delta(\mathbf{X}_+)\delta_{I_+}(\mathcal{L}(\mathbf{X}_+))w^{(I,\xi)}(\mathcal{L}(\mathbf{X}_+))\left[p_{\mathrm{B},+}\psi_+^{(\theta_+)}(\cdot|Z_+)\right]^{\mathbf{X}_{\mathrm{B},+}} \times \left[\hat{p}_+^{(I,\xi,I_+,\theta_+)}(\cdot|Z_+)\right]^{\mathbf{X}_{\mathrm{S},+}\cup\mathbf{X}_{\mathrm{T},+}}\delta\mathbf{X}_+, \tag{6.51}$$

$$= \sum_{I,\xi,I_+,\theta_+}\sum_{L\subseteq\mathbb{L}_+}\delta_{I_+}(L)w_+^{(I,\xi)}(L)\prod_{\ell_+\in L}\bar{p}_+^{(I,\xi,I_+,\theta_+)}(\ell_+|Z_+), \tag{6.52}$$

$$= \sum_{I,\xi,I_+,\theta_+} w_+^{(I,\xi)}(I_+)\left[\bar{p}_+^{(I,\xi,I_+,\theta_+)}(\cdot|Z_+)\right]^{I_+}, \tag{6.53}$$

where (6.52) follows from [95, Lemma 3], which simplifies in (6.53) since the only non-zero inner summand occurs when $L = I_+$, and $\bar{p}_+^{(I,\xi,I_+,\theta_+)}(\ell_+|Z_+)$ is defined in (6.26). Substituting (6.53) into (6.50) we have (6.23).

□

6.4 Efficient Implementation

Expanding on the material discussed in Section 2.4.2, in this section we leverage the joint prediction and update of the fast GLMB implementation to truncate the number of components used to generate the updated multi-object density in (6.23). Note that a GLMB density of form (2.21) at the current time can be represented by

$$\{(I^{(h)}, \xi^{(h)}, w^{(h)}, p^{(h)})\}_{h=1}^{H}, \tag{6.54}$$

which is an enumeration of the set of density parameters $\{(w^{(I,\xi)}, p^{(\xi)}) : (I,\xi) \in \mathcal{F}(\mathbb{L}) \times \Xi\}$ where $w^{(h)} \triangleq w^{(I^{(h)},\xi^{(h)})}$ and $p^{(h)} \triangleq p^{(\xi^{(h)})}$. The objective is to generate a parameter set

$$\{(I_+^{(h_+)}, \xi_+^{(h_+)}, w_+^{(h_+)}, p_+^{(h_+)})\}_{h_+=1}^{H_+} \tag{6.55}$$

that represents the GLMB density at the next time given by (6.23). Following the development in [92, Section III-E], the first step toward doing so requires drawing $H_+^{\max}$ samples from the distribution π given as

$$\pi(I,\xi) \propto w^{(I,\xi)}, \tag{6.56}$$

noting that $T_+^{(h)}$ duplicates of a distinct sample $(I^{(h)}, \xi^{(h)})$ can be drawn. Then, we determine a set of $\tilde{T}_+^{(h)}$ candidate components of the form

$$\{(I_+^{(h,t)}, \theta_+^{(h,t)})\}_{t=1}^{\tilde{T}_+^{(h)}} \tag{6.57}$$

from each (distinct) $(I^{(h)}, \xi^{(h)})$ that together yield significant weights, as in $\hat{w}_+^{(I,\xi,I_+,\theta_+)}(Z_+)$ in (6.23).

Recall from Section 2.4.2 that a set of positive 1-1 vectors γ specifies a significant weight $\hat{w}_+^{(I,\xi,I_+,\theta_+)}(Z_+)$, if γ generates a significant $\omega^{(\gamma)} = \prod_{i=1}^{P} \eta_i(\gamma_i)$. Hence, determining a set of candidate components amounts to finding sets of γ's that yield $\omega^{(\gamma)}$ above a given threshold; it follows that such vector sets can be generated using (6.23). However, complexity of the GLMB filter is naturally increased with the inclusion of spawn modeling and the involvement of marginalization in the truncation procedure is inefficient, especially in cases where many GLMB components are

ultimately discarded. Therefore, the following proposal density is exploited for the purpose of generating candidate components (6.57) in an effort to offset complexity and minimize inefficiency.

Definition 6.1. *Given a GLMB density* (2.21) *at the current time, let the proposal density* $\tilde{\boldsymbol{\pi}}$ *at the next time be of form* (2.86) *such that*

$$\tilde{\boldsymbol{\pi}}_+(\mathbf{X}_+|Z_+) \propto \Delta(\mathbf{X}_+) \sum_{h=1}^{H} \sum_{t=1}^{\tilde{T}_+^{(h)}} w^{(h)} \tilde{w}_+^{(h,t)}(Z_+) \delta_{I_+^{(h)}}(\mathcal{L}(\mathbf{X}_+)) \left[\tilde{p}^{(h,t)}(Z_+)\right]^{\mathbf{X}_+}, \tag{6.58}$$

where $\psi_+^{(h,t)} \triangleq \psi_+^{(\theta_+^{(h,t)})}$, $\tilde{\psi}_+^{(h,t)} \triangleq \tilde{\psi}_+^{(\theta_+^{(h,t)})}$,

$$\begin{aligned}\tilde{w}_+^{(h,t)}(Z_+) = {} & [r_{\mathrm{B},+}]^{\mathbb{B}_+ \cap I_+^{(h,t)}} [1 - r_{\mathrm{B},+}]^{\mathbb{B}_+ - I_+^{(h,t)}} \left[\bar{p}_{\mathrm{S}}^{(h)}\right]^{I^{(h)} \cap I_+^{(h,t)}} \left[1 - \bar{p}_{\mathrm{S}}^{(h)}\right]^{I^{(h)} - I_+^{(h,t)}} \\ & \times \left[\bar{p}_{\mathrm{T}}^{(h)}\right]^{\mathbb{T}_+ \cap I_+^{(h,t)}} \left[1 - \bar{p}_{\mathrm{T}}^{(h)}\right]^{\mathbb{T}_+ - I_+^{(h,t)}} \left[\tilde{\psi}_+^{(h,t)}(\cdot|Z_+)\right]^{I_+^{(h,t)}},\end{aligned} \tag{6.59}$$

$$\tilde{p}^{(h,t)}(x_+, \ell_+|Z_+) = \frac{\tilde{p}_+^{(h)}(x_+, \ell_+) \psi^{(h,t)}(x_+, \ell_+|Z_+)}{\tilde{\psi}^{(h,t)}(x_+, \ell_+|Z_+)}, \tag{6.60}$$

$$\tilde{p}_+^{(h)}(x_+, \ell_+) = 1_{\mathbb{B}_+}(\ell_+) p_{\mathrm{B},+}(x_+, \ell_+) + 1_{\mathbb{L}}(\ell_+) \tilde{p}_{\mathrm{S}}^{(h)}(x_+, \ell_+) + 1_{\mathbb{T}_+}(\ell_+) \tilde{p}_{\mathrm{T}}^{(h)}(x_+, \ell_+), \tag{6.61}$$

$$\tilde{p}_{\mathrm{S}}^{(h)}(x_+, \ell_+) = \frac{\langle p_{\mathrm{S}}(\cdot, \ell_+) f_{\mathrm{S},+}(x_+|\cdot, \ell), p^{(h)}(\cdot, \ell_+) \rangle}{\bar{p}_{\mathrm{S}}^{(h)}(\ell_+)}, \tag{6.62}$$

$$\tilde{p}_{\mathrm{T}}^{(h)}(x_+, \ell_+) = \frac{\langle p_{\mathrm{T}}(\ell_+) f_{\mathrm{T},+}(x_+|\cdot, \ell), p^{(h)}(\cdot, \ell) \rangle}{\bar{p}_{\mathrm{T}}^{(h)}(\ell_+)}, \tag{6.63}$$

$$\bar{p}_{\mathrm{S}}^{(h)}(\ell_+) = \langle p^{(h)}(\cdot, \ell), p_{\mathrm{S}}(\ell_+) \rangle, \tag{6.64}$$

$$\bar{p}_{\mathrm{T}}^{(h)}(\ell_+) = \langle p^{(h)}(\cdot, \ell), p_{\mathrm{T}}(\ell_+) \rangle, \tag{6.65}$$

$$\tilde{\psi}^{(h,t)}(x_+, \ell_+|Z_+) = \langle \tilde{p}_+^{(h)}(x_+, \ell_+), \psi^{(h,t)}(x_+, \ell_+|Z_+) \rangle. \tag{6.66}$$

Enumerate $Z_+ = \{z_{1:|Z_+|}\}$, $\mathbb{B}_+ = \{\ell_{1:K}\}$, $I^{(h)} = \{\ell_{K+1:L}\}$, along with the additional set of spawn labels at the next time $\mathbb{T}_+(I^{(h)}) = \{\ell_{L+1:P}\}$. Next, define a P dimensional vector $\gamma^{(h,t)}$ that inherits the 1-1 mapping from $\theta_+^{(h,t)}$ (see (2.93)), then is recovered $(I_+^{(h,t)}, \theta_+^{(h,t)})$ via

$$I_+^{(h,t)} = \{\ell_i \in \mathbb{B}_+ \cup I^{(h)} \cup \mathbb{T}_+(I^{(h)}) : \gamma_i^{(h,t)} \geq 0\}, \tag{6.67a}$$

$$\theta_+^{(h,t)}(\ell_i) = \gamma_i^{(h,t)}, \tag{6.67b}$$

such that $\theta_+^{(h,t)} : I_+^{(h,t)} \to \{0 : |Z_+|\}$. Then, for each $j \in \{-1 : |Z_+|\}$ define

$$\eta_i^{(h)}(j) = \begin{cases} 1 - r_{\mathrm{B},+}(\ell_i), & \ell_i \in \mathbb{B}_+, j < 0, \\ r_{\mathrm{B},+}(\ell_i)\tilde{\psi}_+^{(h,j)}(\ell_i|Z_+), & \ell_i \in \mathbb{B}_+, j \geq 0, \\ 1 - \bar{p}_{\mathrm{S}}^{(h)}(\ell_i), & \ell_i \in I^{(h)}, j < 0, \\ \bar{p}_{\mathrm{S}}^{(h)}(\ell_i)\tilde{\psi}_+^{(h,j)}(\ell_i|Z_+), & \ell_i \in I^{(h)}, j \geq 0, \\ 1 - \bar{p}_{\mathrm{T}}^{(h)}(\ell_i), & \ell_i \in \mathbb{T}_+(I^{(h)}), j < 0, \\ \bar{p}_{\mathrm{T}}^{(h)}(\ell_i)\tilde{\psi}_+^{(h,j)}(\ell_i|Z_+), & \ell_i \in \mathbb{T}_+(I^{(h)}), j \geq 0, \end{cases} \tag{6.68}$$

assuming that, for all $i \in \{1 : P\}$, $\bar{p}_{\mathrm{D},+}^{(h)}(\ell_i) \triangleq \left\langle \tilde{p}_+^{(h)}(\cdot, \ell_i), p_{\mathrm{D},+}(\cdot, \ell_i) \right\rangle \in (0,1)$, $\bar{p}_{\mathrm{S}}^{(h)}(\ell_i) \in (0,1)$, and $\bar{p}_{\mathrm{T}}^{(h)}(\ell_i) \in (0,1)$. Equation (6.68) is used in conjunction with the Gibbs sampler to yield *mostly* high-weighted positive 1-1 vectors $\gamma^{(h,t)}$, then each $\gamma^{(h,t)}$ is converted to candidate component $(I_+^{(h,t)}, \theta_+^{(h,t)})$ using (6.67).

Moving forward, the candidate components determined using the proposal density (6.58)-(6.68) are subsequently used to generate the GLMB density in (6.23). For each sample component $(I^{(h)}, \xi^{(h)})$ and each of its candidate components (6.57) formed using (6.67) and (6.68), generate intermediate components of form $(I_+^{(h,t)}, \xi_+^{(h,t)}, w_+^{(h,t)}, p_+^{(h,t)})$. Letting

$$\hat{p}_+^{(h,t)} \triangleq \hat{p}_+^{(I^{(h)}, \xi^{(h)}, I_+^{(h,t)}, \theta_+^{(h,t)})},$$

and

$$\bar{p}_+^{(h,t)} \triangleq \bar{p}_+^{(I^{(h)}, \xi^{(h)}, I_+^{(h,t)}, \theta_+^{(h,t)})},$$

compute

$$\bar{w}_+^{(h,t)} = w^{(h)} \left[r_{\mathrm{B},+}\right]^{\mathbb{B}_+ \cap I_+^{(h,t)}} \left[1 - r_{\mathrm{B},+}\right]^{\mathbb{B}_+ - I_+^{(h,t)}} \left[\bar{p}_+^{(h,t)}(\cdot|Z_+)\right]^{I_+^{(h,t)}}, \tag{6.69}$$

$$p_+^{(h,t)}(\cdot, \ell_i) \propto 1_{\mathbb{B}_+}(\ell_i) p_{\mathrm{B},+}(\cdot, \ell_i) \psi_+^{(h,t)}(\cdot, \ell_i|Z_+) + (1 - 1_{\mathbb{B}_+}(\ell_i)) \hat{p}_+^{(h,t)}(\cdot, \ell_i|Z_+), \tag{6.70}$$

define $\xi_+^{(h,t)} = (\xi^{(h)}, \theta_+^{(h,t)})$, and let

$$\hat{C} = \sum_{h,t} \bar{w}_+^{(h,t)} \left[\bar{p}_+^{(h,t)}(\cdot|Z_+)\right]^{I_+^{(h,t)}} \tag{6.71}$$

be a normalizing constant. Equations (6.69)-(6.71) follow directly from (6.23)-(6.26). Algorithm 1 (see Appendix C) summarizes the joint prediction and update procedure for one iteration, including how the intermediate component set $\{(I_+^{(h,t)}, \xi_+^{(h,t)}, w_+^{(h,t)}, p_+^{(h,t)})\}_{h,t=1,1}^{H,\tilde{T}_+^{(h)}}$ is formed.

Note that a given set $\{(I_+^{(h,t)}, \xi_+^{(h,t)}, p_+^{(h,t)})\}_{h,t=1,1}^{H,\tilde{T}_+^{(h)}}$ may not be unique, accordingly, the parameter set (6.55) is determined by summing all $w_+^{(h,t)}$ with $(I_+^{(h,t)}, \xi_+^{(h,t)}) = (I_+^{(h_+)}, \xi_+^{(h_+)})$, then normalizing the weights $w_+^{(h,t)}$. This procedure follows from the relationship presented in (2.92) and is summarized in Algorithm 2 , which comes from the bottom portion of [92, Algorithm 2] and is replicated here for convenience. All algorithms are relegated to Appendix C and follow the format of those presented in [92] in the interest of consistency. Additionally, we use the Gibbs and Unique functions as described in [92, Algorithm 1] and [92, Section III-E], respectively.

6.5 Simulation

A linear Gaussian example is used to verify the proposed GLMB filter and compare its performance to the CPHD filter; both filters incorporate object spawning. Fig. 6.2 illustrates the multiple trajectories in a $[-1000, 1000]\,\mathrm{m} \times [-1000, 1000]\,\mathrm{m}$ surveillance region considered in this scenario. Over the 100 s scenario duration, the number of objects varies due to birth, spawning, and death. In total, there are 6 spontaneous births and 6 spawning events.

At the start, an object is born in each of the three birth regions. Each birth object goes on to generate a single first generation spawn, then dies. After crossing at the origin at time $k = 45$, each spawn object generates a single second generation spawn. Towards the end, an object appears in each birth region that goes on to cross paths with a second generation spawn; crossings occur at times $k = 82, k = 84,$ and $k = 86$ at positions $(-250, -433)$, $(-260, 430)$, and $(507, 26)$, respectively.

The single-object state describing an object's planar position and velocity coordinates is $x_+ = [p_{x,+}, p_{y,+}, \dot{p}_{x,+}, \dot{p}_{y,+}]^T$. Each object has a probability of survival $p_S = 0.99$ and follows linear

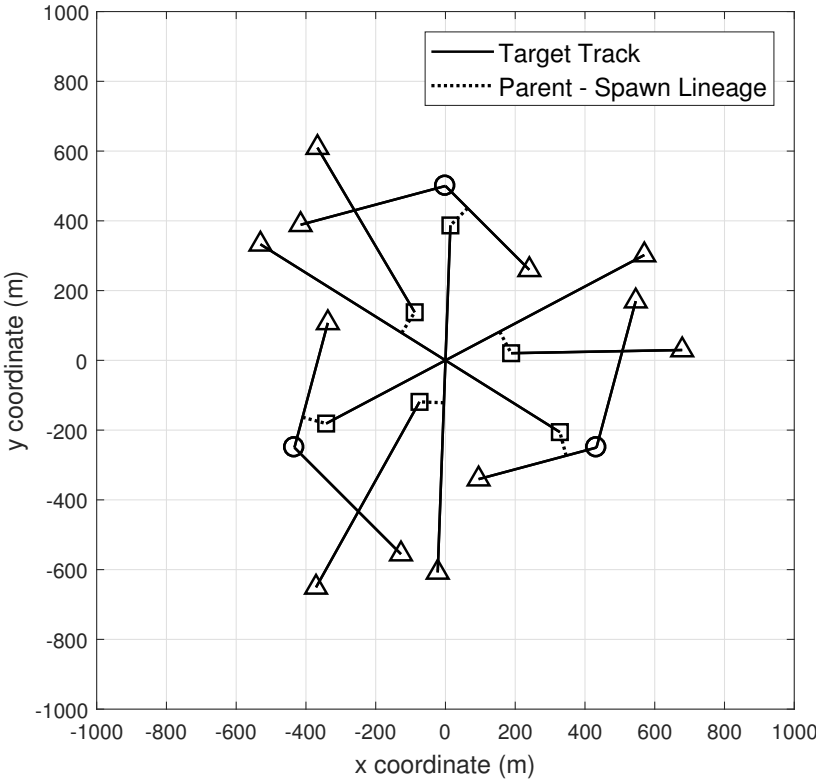


Figure 6.2: Object trajectories in the xy plane. A circle "○" indicates where an object is born, a square "□" indicates where a spawned object may be detected, and a triangle "△" indicates where an object dies.

Gaussian dynamics with transition density $f_{\mathrm{S},+}(x_+, \ell_+|x, \ell) = \mathcal{N}(x_+; Fx, Q)$ such that

$$F = \begin{bmatrix} I_2 & \Delta I_2 \\ 0_2 & I_2 \end{bmatrix}, \qquad Q = \sigma_\nu^2 \begin{bmatrix} \frac{\Delta^4}{4} I_2 & \frac{\Delta^3}{2} I_2 \\ \frac{\Delta^3}{2} I_2 & \Delta^2 I_2 \end{bmatrix},$$

where $\Delta = 1\,\mathrm{s}$, $\sigma_\nu = 1\,\mathrm{m\,s^{-2}}$, and I_n and 0_n denote the $n \times n$ identity and zero matrices, respectively.

Each object is detected with probability $p_{\mathrm{D},+} = 0.88$ and each object generated measurement $z_+ = [z_{x,+}, z_{y,+}]^T$ consists of the object's position with noise added to each component. Measurements follow the linear Gaussian measurement model $g_+(z_+|x_+, \ell_+) = \mathcal{N}(z_+; Hx_+, R)$ such that

$$H = [I_2 \;\; 0_2], \qquad R = \sigma_\epsilon^2 I_2,$$

where $\sigma_\epsilon = 10\,\mathrm{m}$. Clutter is modeled as a Poisson RFS with an average intensity of $\lambda_c = 1.65 \times 10^{-5}\mathrm{m}^{-2}$ yielding an average of 66 clutter returns per scan.

Objects can appear either by birth or spawning. The birth model is an LMB RFS with parameters $\pi_{\mathrm{B},+} = \{r_{\mathrm{B},+}(\ell_i), p_{\mathrm{B},+}(\ell_i)\}_{i=1}^{3}$ where $r_{\mathrm{B},+}(\ell_i) = 0.02$ and $p_{\mathrm{B},+}(\ell_i) = \mathcal{N}(x; m_{\mathrm{B}}^{(i)}, P_{\mathrm{B}})$ with $m_{\mathrm{B}}^{(1)} = [0, 500, 0, 0]^T$, $m_{\mathrm{B}}^{(2)} = [433, -250, 0, 0]^T$, $m_{\mathrm{B}}^{(3)} = [-433, -250, 0, 0]^T$, and $P_{\mathrm{B}} = \sigma_{\mathrm{B}}^2 I_4$ where $\sigma_{\mathrm{B}} = 10$.

Given a parent state $\mathbf{x} = (x, \ell)$ at the current time k and setting $M_\ell = 1$, from (6.1), the set of spawn labels at the next time $k+1$ is defined as

$$\mathbb{T}_+(\ell) = \{(\ell, k+1) \times \{1\}\} = \{(\ell, k+1, 1)\}.$$

Additionally, the probability of detection is set constant, i.e., $p_{\mathrm{T}} \triangleq p_{\mathrm{T}}(\mathbf{x}; \varsigma)$. Then, the spawn model is a conditional LMB RFS with parameters $\{(p_{\mathrm{T}}, f_{\mathrm{T},+}(\cdot|\mathbf{x};\varsigma)) : \varsigma \in \mathbb{T}_+(\ell)\}$ where $p_{\mathrm{T}} = 0.01$ and $f_{\mathrm{T},+}(\cdot|x, \ell; \varsigma) = \sum_{i=1}^{3} \mathcal{N}(\cdot; Fx + d_{\mathrm{T}}^{(i)}, Q_{\mathrm{T}})$ with $Q_{\mathrm{T}} = \sigma_{\mathrm{T}}^2 I_4$, $\sigma_{\mathrm{T}} = 5$. Each $d_{\mathrm{T}}^{(i)}$ is configured such that a spawn track with zero velocity is generated at a distance of 70 m from a parent state x_k and in a direction relative to the parent's bearing θ, i.e.,

$$d_{\mathrm{T}}^{(i)} = [70\cos(\theta + \phi^{(i)}), 70\sin(\theta + \phi^{(i)}), -\dot{p}_{x,k}, -\dot{p}_{y,k}]^T,$$

where $\phi^{(1)} = -80\,\mathrm{deg}$, $\phi^{(2)} = -90\,\mathrm{deg}$, and $\phi^{(3)} = -100\,\mathrm{deg}$.

The maximum number of GLMB filter components is capped at 1000. Using the Gibbs sampler to randomly generate hypotheses, the probabilities of survival and detection are tempered with values set to $\breve{p}_{\mathrm{S},k} = 0.90 p_{\mathrm{S},k}$ and $\breve{p}_{\mathrm{D},k} = 0.90 p_{\mathrm{D},k}$, respectively. This induces the sampler to yield more track termination and miss detection hypotheses, which expedites the termination of truly dead tracks while reducing the occurrence of dropped tracks. For more details on tempering techniques, see [92]. The CPHD filter is configured with a Bernoulli spawn model following the presentation in [14] and is capped at 1000 Gaussian mixture components.

Results are presented for 100 Monte Carlo simulations. The mean and standard deviation of cardinality estimates over time are shown in Figs. 6.3 and 6.4, while mean OSPA [75] distances are shown in Fig. 6.5. Mean OSPA localization and cardinality components are shown in Fig. 6.6.

Similar to the results presented in [94], both filters accurately estimate cardinality with the GLMB filter providing a better cardinality variance estimate. The GLMB also exhibits better miss

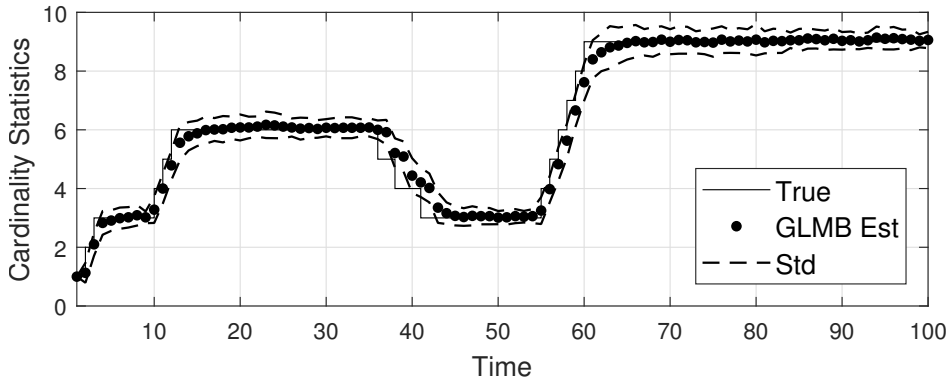


Figure 6.3: Cardinality statistics for GLMB filter (100 Monte Carlo trials).

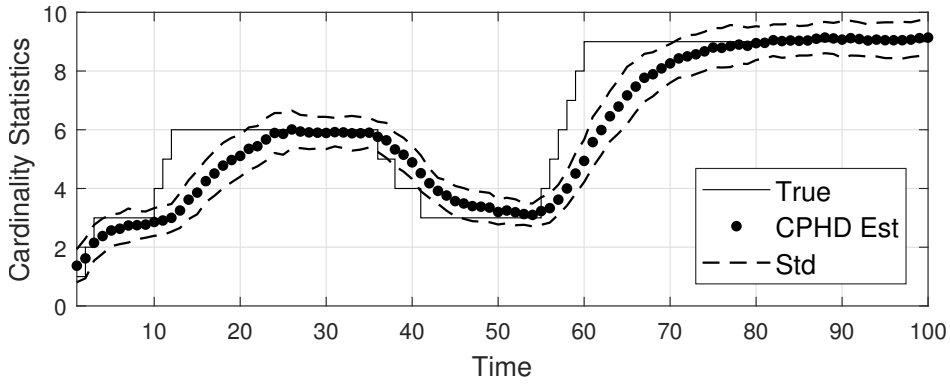


Figure 6.4: Cardinality statistics for CPHD filter (100 Monte Carlo trials).

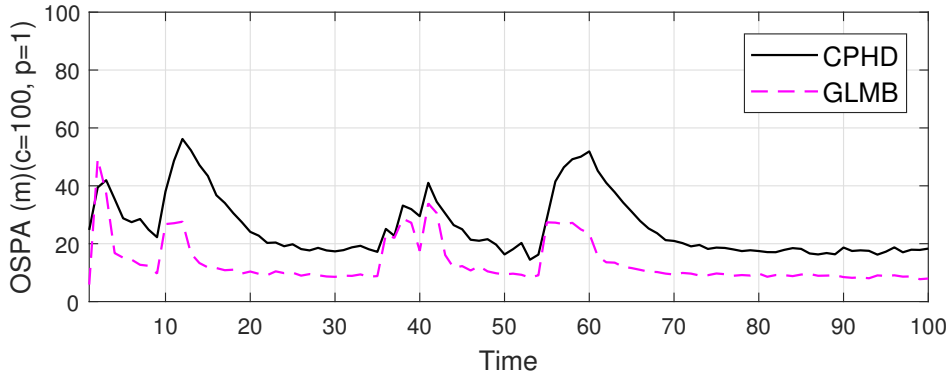


Figure 6.5: OSPA distance for GLMB and CPHD filters (100 Monte Carlo trials).

distance performance throughout the majority of the simulation. From Fig. 6.6 we see that the GLMB outperforms the CPHD in both cardinality and localization components overall. Consistent with the assessment in [94], lower estimated cardinality variance promotes improved cardinality

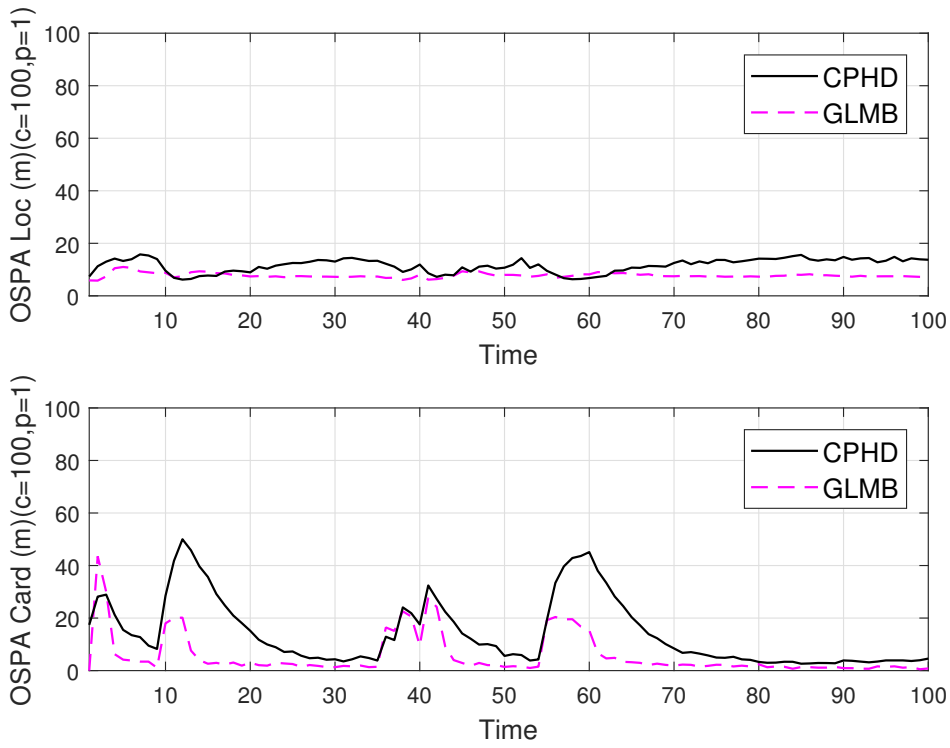


Figure 6.6: OSPA components for GLMB and CPHD filters (100 Monte Carlo trials).

performance. Improved localization performance is attributed to the GLMB's ability to propagate the filtering density more accurately and its immunity to the "spooky-effect". The CPHD filter is susceptible to this effect where in the event of a miss detection the PHD *mass* shifts away from undetected components to detected ones, regardless of the distance between them [32, 94].

The proposed GLMB filter's ability to capture ancestry information is demonstrated in Figs. 6.7-6.9. For each Monte Carlo run, the final label estimates at time $k = 100$ are compared to the set of true labels of the same time which are presented in Table 6.1. The true labels are presented horizontally and are enumerated by the far left column. Shading is used to divide labels 1-6 into segments correlating with birth and spawning events in Figs. 6.7-6.9, where red, green, and yellow represent a birth track from which spawn originate, first generation spawn, and second generation spawn, respectively. Labels 7-9 are not considered for this analysis since they have no ancestry.

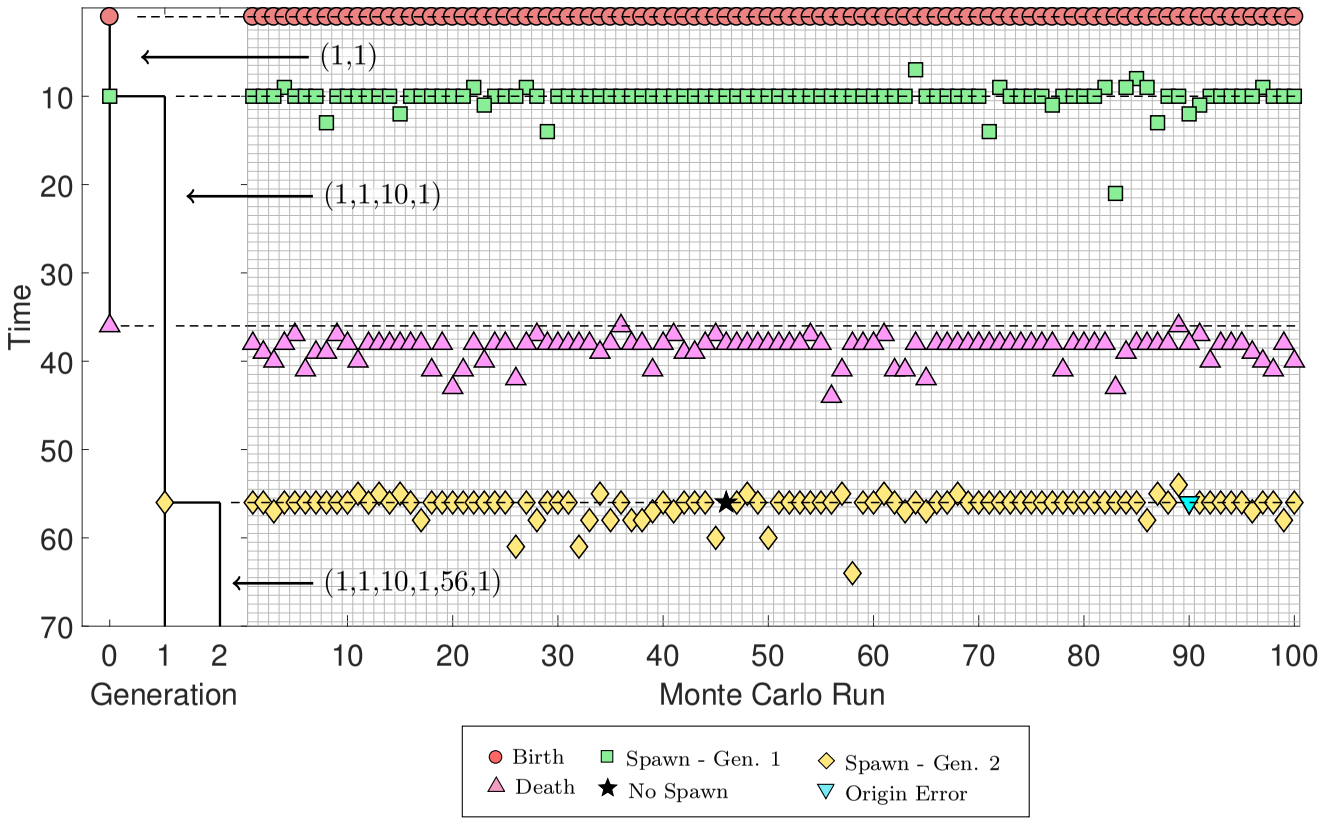


Figure 6.7: Birth region ancestry truth and estimates. Each region's true ancestry tree is at the far left. A red circle at the top indicates birth, while the first and second generation spawn times are denoted by a green square and yellow diamond, respectively. The birth track's time of death is marked by a pink diamond. Markers within the gridded area indicate GLMB filter estimated birth, death, and spawn times for 100 Monte Carlo runs.

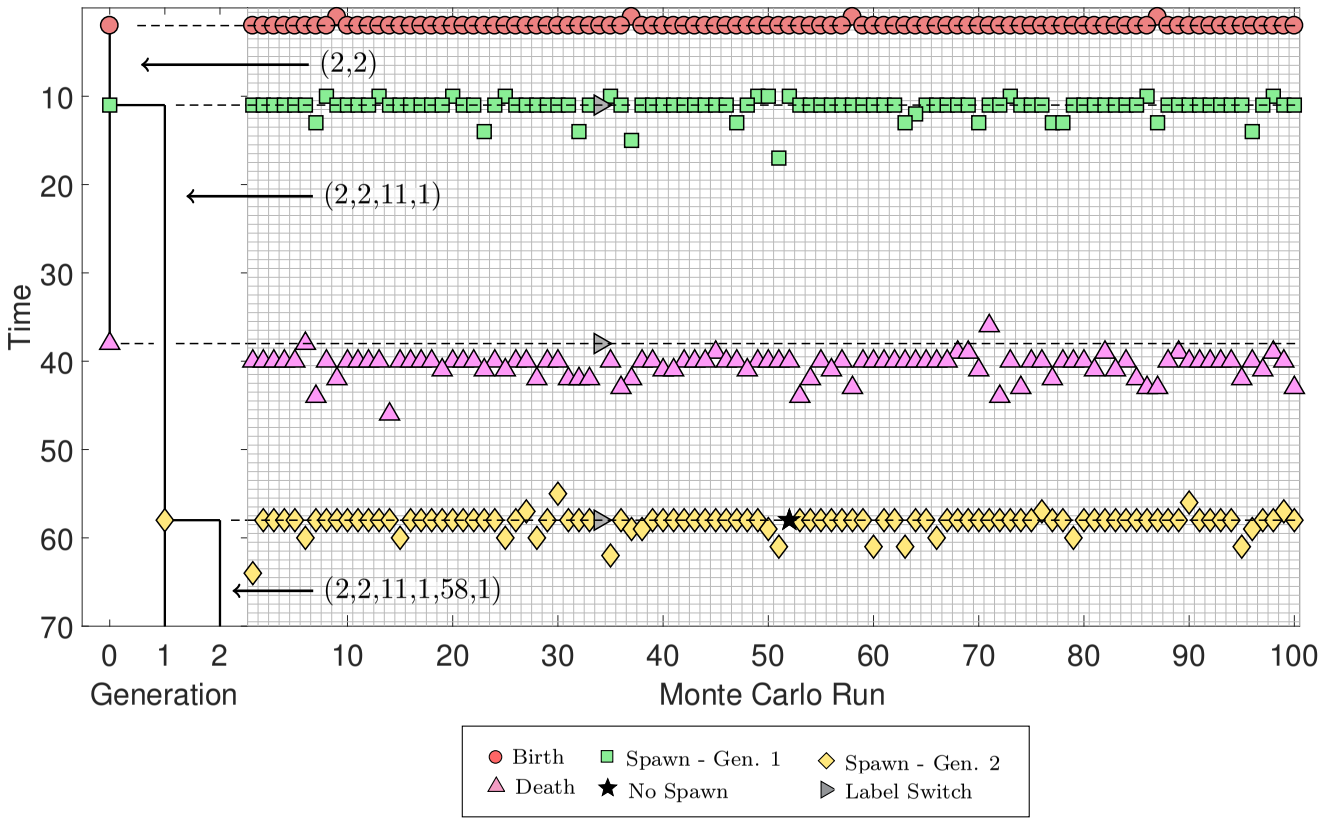


Figure 6.8: Birth region ancestry truth and estimates. Each region's true ancestry tree is at the far left. A red circle at the top indicates birth, while the first and second generation spawn times are denoted by a green square and yellow diamond, respectively. The birth track's time of death is marked by a pink diamond. Markers within the gridded area indicate GLMB filter estimated birth, death, and spawn times for 100 Monte Carlo runs.

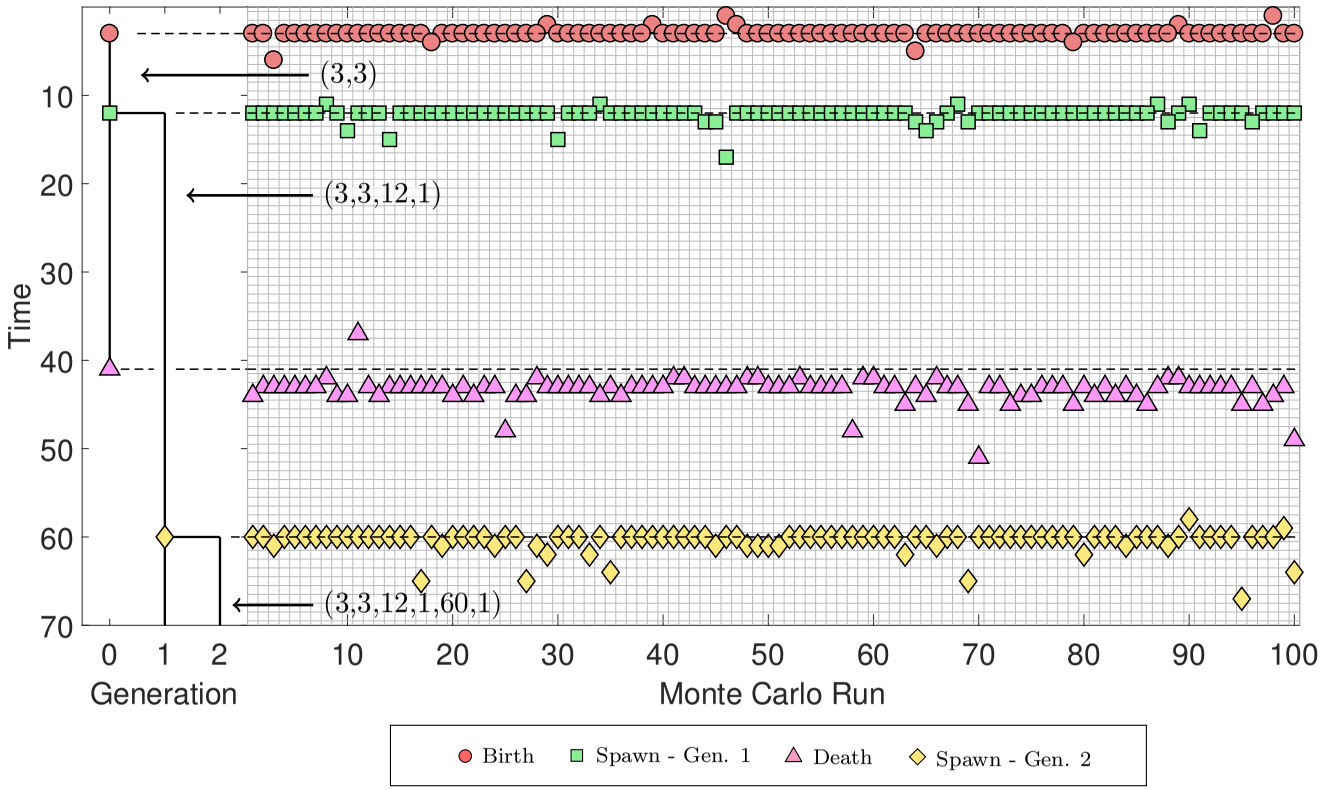


Figure 6.9: Birth region ancestry truth and estimates. Each region's true ancestry tree is at the far left. A red circle at the top indicates birth, while the first and second generation spawn times are denoted by a green square and yellow diamond, respectively. The birth track's time of death is marked by a pink diamond. Markers within the gridded area indicate GLMB filter estimated birth, death, and spawn times for 100 Monte Carlo runs.

When comparing the final label estimates at time $k = 100$ of each Monte Carlo run with the truth, estimated labels are segregated by common ancestry, e.g., from Table 6.1, labels 1 and 4 belong to one group, labels 2 and 5 belong to another, etc. For a given label group, the time of birth for their common birth ancestor track may differ from the truth, therefore it is necessary to trace the common ancestor track's label and state estimates toward the beginning of the given Monte Carlo run. Comparing this time history of state estimates with the truth, the originating birth region is determined. Then, the birth track's time of death is found by tracing its label from the beginning of the given run to the point in time when its label is no longer present in the set of label estimates. The remaining event times represented in Figs. 6.7-6.9, i.e., times of birth and spawning, are extracted from the labels of a final label estimates at time $k = 100$ of the given Monte Carlo run.

Table 6.1: Label Ground Truth at Time $k = 100$

Label #	Label					
1	1	1	10	1	56	1
2	2	2	11	1	58	1
3	3	3	12	1	60	1
4	1	1	10	1		
5	2	2	11	1		
6	3	3	12	1		
7	55	3				
8	57	1				
9	59	2				

The formats of Figs. 6.7-6.9 are generally the same. Each birth region's true ancestry tree is at the far left and is aligned with a gridded area to the right with markers indicating GLMB filter estimated birth, death, and spawn times for 100 Monte Carlo runs. Each figure corresponds to one of the three modeled birth regions as indicated by the true track labels. Each birth region's true ancestry tree is at the far left and begins with a red circle at the top. Moving down the vertical axis in time, the first and second generation spawn times are denoted by a green square and yellow diamond, respectively, while the birth track's time of death is marked by a pink diamond. Moving

from left to right across the gridded area, markers indicate GLMB filter estimated birth, death, and spawn times for 100 Monte Carlo runs. Each figure corresponds to one of the three modeled birth regions as indicated by the true track labels.

We see from Figs. 6.7-6.9 that the GLMB filter accurately captures ancestry information overall, where the results in Fig. 6.9 exhibit the best performance. Fig. 6.7 indicates that the second generation spawn track $(1, 1, 10, 1, 56, 1)$ was not spawned during run 46 and that the same generation track was estimated as originating from a different parent during run 90. In this specific case, track $(1, 1, 10, 1)$ dropped prior to tracks crossing at the origin (see Fig. 6.2) and was later estimated as having spawned from one of the remaining first generation spawn tracks, subsequently spawning an object at time $k = 56$. Similarly, Fig. 6.8 shows that track $(2, 2, 11, 1, 58, 1)$ did not spawn during run 52, while during run 34, a label switch occurred when track $(2, 2)$ essentially took the place of track $(2, 2, 11, 1)$, going on to spawn track $(2, 2, 58, 1)$ at time $k = 58$. These inaccurate ancestry estimates are due to missed detections of either the parent track, spawn track, or both.

Chapter 7

Spawned Object Orbit Determination Simulations: GM-GLMB

This chapter includes the first application of the GLMB filter, with spawning, to an orbit determination problem. The CubeSat deployment scenario is revisited, similar to that in Chapter 5, but not identical. Since the 50 CubeSat deployment simulations presented in Chapter 5 were performed, the Indian Space Research Organisation (ISRO) has successfully executed a more ambitious real-world mission in which a total of 104 objects, 100 of them CubeSats, were deployed from a single LV. The simulations presented in this chapter are modeled after the ISRO's mission, details of which are provided in Section 7.1. The most significant difference between the scenario presented in this chapter is that the GLMB filter does not require the same high observation density as the CPHD filter did in Chapter 5. Hence, a more realistic surveillance scheme is implemented, which is detailed in Section 7.1.1. The simulated dynamics in this chapter, however, are quite similar to those of Chapter 5, and are discussed in Section 7.1.2. Section 7.1.3 details filter configuration and results are presented in Section 7.2. Section 7.2.3 discusses what remains to be done for the material presented in this chapter to become a more well-rounded exploration of appying the GLMB filter to spawning event orbit determination.

7.1 Scenario Description

Simulation begins with only one object at epoch, i.e., the PSLV-C37 launch vehicle, initial conditions for which are presented in Table 7.1. The models for LV trajectory and CubeSat deploy-

ment are based on information available at the time of analysis[1,2]. The scan interval is $\Delta t = 60\,\text{sec}$ and scenario duration is just over 25 hours.

Table 7.1: PSLV-C37 Initial Conditions

Parameter	Setting
Epoch Time (Julian Date)	2457799.645833
Semimajor Axis (km)	6862.340
Eccentricity	2.4234×10^{-3}
Inclination (deg)	97.604
Right Asc. of Node (deg)	134.959
Arg. of Perigee (deg)	80.761
True Anomaly (deg)	274.512

CubeSat deployment is initiated 295 min after epoch, at which time two CubeSats are ejected from the LV. Then, two more every 10 s until 50 CubeSats are deployed. For analysis and troubleshooting purposes, only half of the full 100 CubeSat deployments were executed for analysis presented in this document.

With the absence of elaborate CubeSat deployment details, much of the deployment model is assumed. In part, the QB50 deployment model [53] from Chapter 5 is used, but with some adjustment. Two different deployment models are used, i.e., the scenario is run twice, once with each of the deployment models. This is done to demonstrate filter efficacy when the disparity between (filter) spawn and (true) deployment models varies, and to highlight the need for expanding spawn model versatility. Both models are used to simulate ground truth and are implemented in the NTW frame, a coordinate system that moves with the LV where the N axis lies in the orbital plane, the T axis is always parallel to the LV's velocity vector, and the W axis completes the right-handed system.

Like the QB50 deployment model, CubeSats are deployed in the anti-velocity direction of PSLV-C37's motion. The first deployment model, hereafter referred to as the PSLV-deploy-1 model, assumes very precise control over the LV's motion such that rotation about any axis in the NTW

[1] "India launches more than 100 satellites into orbit." *USA Today*, February 15, 2017. Web. June 8, 2017.

[2] *PSLV-C37 Brochure-ISRO.* Web. June 8, 2017.

frame is negligible. CubeSats are ejected with a speed of 1.5 m/s such that, after one second elapses, they appear precisely at the two points indicated by red and blue squares in Fig 7.1.

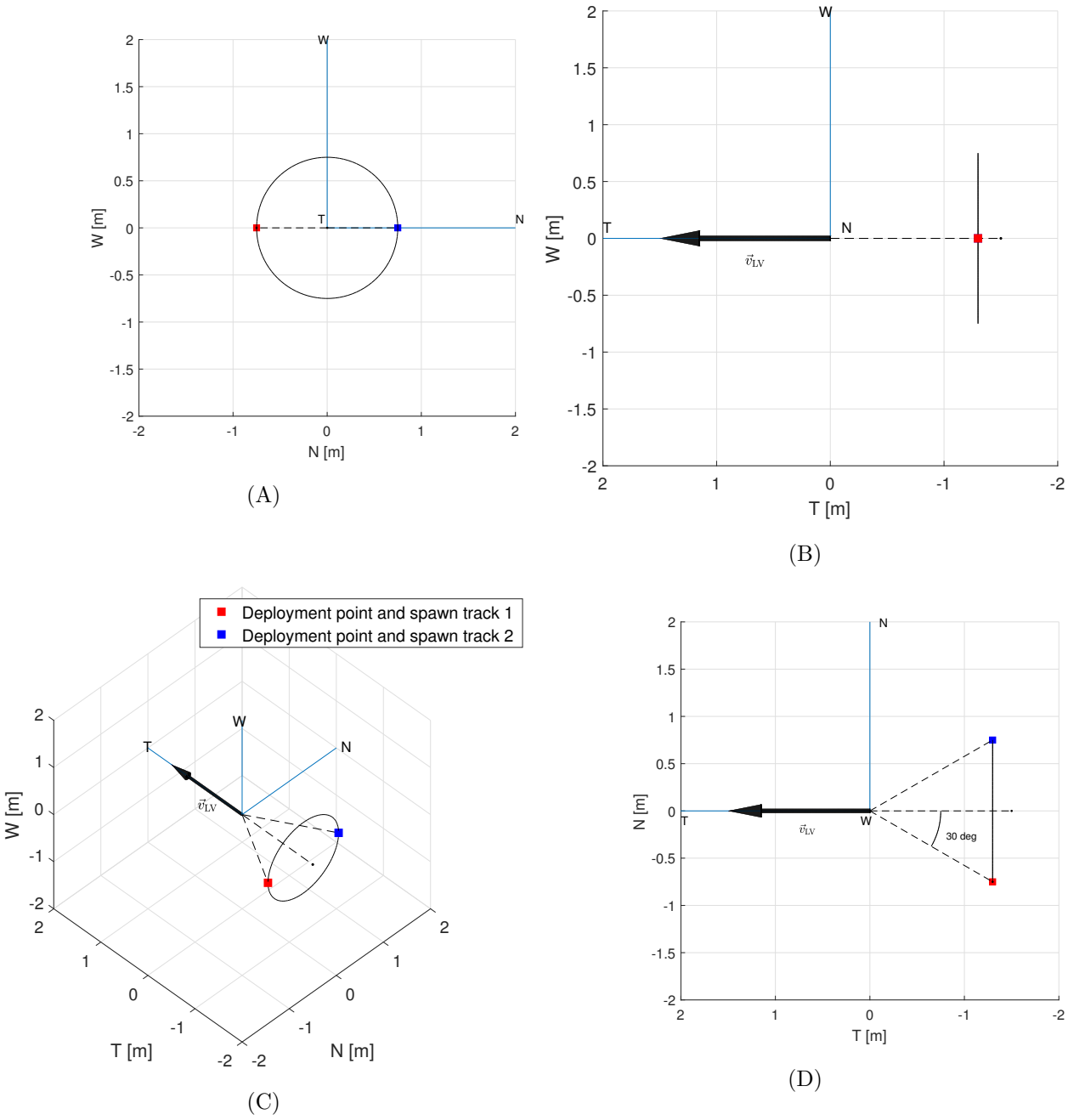


Figure 7.1: PSLV-deploy-1 model depicted in the NTW frame. (A) View from aft of LV, looking forward (T axis points into the page). (B) Side view (N-axis points into the page). (C) Perspective view. (D) W-axis points out of page.

The second deployment model, PSLV-deploy-2, assumes that less accurate knowledge of CubeSat ejection dynamics is available. Instead of deploying at two rigid points, slight directional variations are assumed for CubeSat ejection such that objects can randomly appear in the arc-shaped regions illustrated in Fig. 7.2.

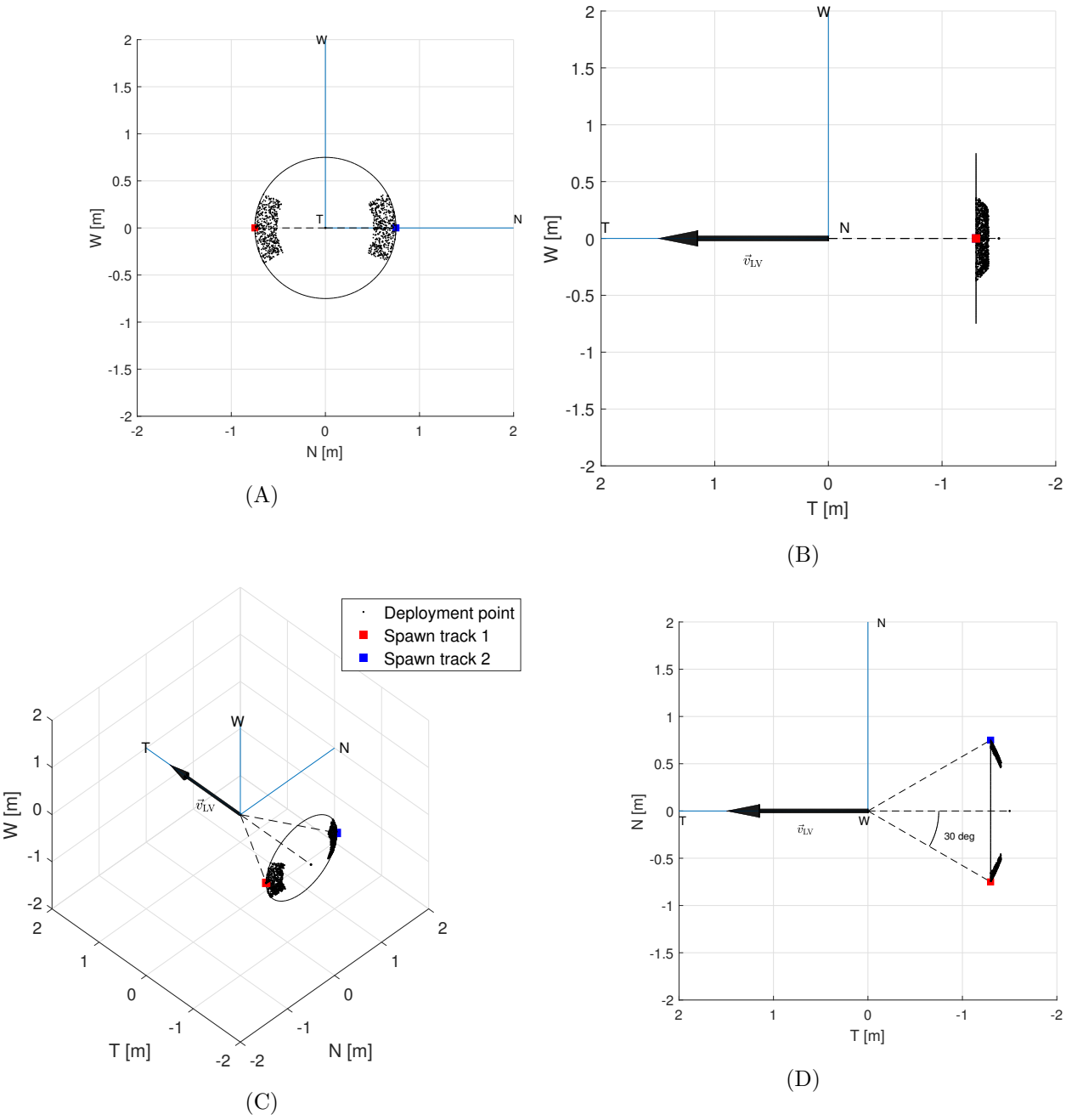


Figure 7.2: PSLV-deploy-2 model depicted in the NTW frame. (A) View from aft of LV, looking forward (T axis points into the page). (B) Side view (N-axis points into the page). (C) Perspective view. (D) W-axis points out of page.

7.1.1 Measurements

A surveillance network consisting of 8 phased array radars is simulated. Figure 7.3 illustrates radar locations and groundtracks for the LV's first four passes. Simulating observations consists of generating *ground truth*[3] states in the Geocentric Celestial Reference Frame (GCRF) for all considered objects at every point in time, as per the modeled scenario period and time interval Δt. At each point in time, each GCRF state is converted to an observation vector consisting of range ρ, range-rate $\dot{\rho}$, azimuth β, and elevations $e\ell$, i.e., $z_k = [\rho, \dot{\rho}, \beta, e\ell]^T$, for each sensor. The parameters in Table 7.2, transcribed here from [88] for convenience, are used to determine sensor-to-RSO access, i.e., whether the sensor has line of sight (LOS) to the RSO. If no access is determined for all sensor-RSO pairings, then data for the current time are discarded, thus forming an observation gap. When access is determined for only a single sensor, the m available observation vectors form a set $Z_k = \{z_1, \ldots, z_m\}$.

Tractability issues can arise when more than one sensor is used to form a multi-object observation at the same point in time. Such issues are addressed in [91], specifically for GLMB filter implementations, however, doing so here is beyond the scope of this thesis. Hence, only one sensor at each time is selected when simulating observations. When access is determined for multiple sensors, the sensor with access to the most objects is selected. Furthermore, in the event that multiple sensors have access to the same number of objects, the sensor with the shortest range to the LV is selected.

[3] The term ground truth refers to a data set that is unaltered by bias or noise, which serves as a means of determining a filter's efficacy in providing estimates when provided a biased and/or noisy version of the same data set. The term is commonly used in tracking literature, e.g., see [5].

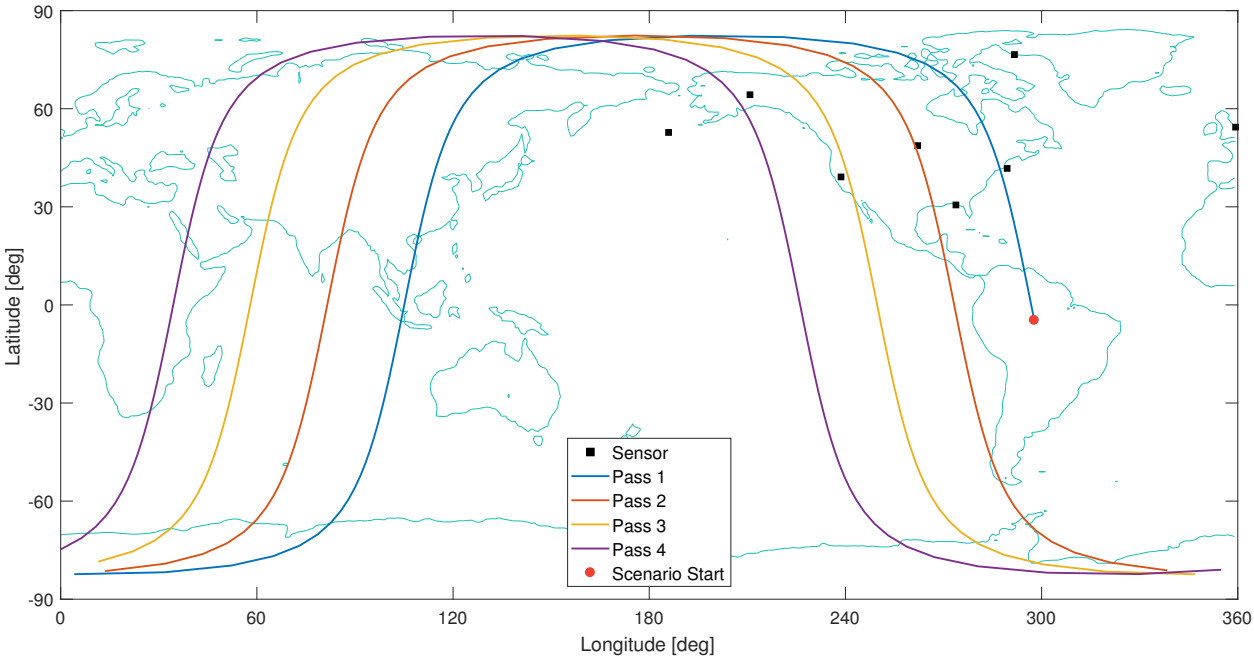


Figure 7.3: Sensor locations and first 4 launch vehicle passes.

Table 7.2: Phased Array Radar Locations and Physical Limits [88]

				Physical Limits				
	Location			Range (km)	Azimuth (deg)		Elevation (deg)	
Name	Lat. (deg)	Lon. (deg)	Alt. (m)	Max	Min	Max	Min	Max
Eglin	30.57	273.79	34.7	13,210	145	215	1	90
Fylingdales	54.37	359.33	338.9	4,820	285	189	4	70
Clear	64.29	210.81	213.3	4,910	170	110	1	90
Cape Cod	41.75	289.46	80.3	5,555	347	227	3	80
Beale	39.14	238.65	115.7	5,555	126	6	3	80
Shemya	52.74	185.91	89.8	4,910	259	379	1	30
Thule	76.57	291.7	424.7	5,555	297	177	3	80
Cavalier	48.72	262.1	347.3	3,300	313	62	2	45

Zero-mean Gaussian white noise is added to each $z_k \in Z_k$ with covariance Cholesky factor $D = \text{diag}(\sigma_\rho, \sigma_{\dot{\rho}}, \sigma_\beta, \sigma_{e\ell})$, where σ_ρ, σ_β, and $\sigma_{e\ell}$ are defined in Table 7.3 [88] for all sensors and $\sigma_{\dot{\rho}} = \sigma_\rho \times 10^{-4}$ is assumed.

Table 7.3: Phased Array Radar Error and Resolution Characteristics

	Noise				
Name	Range [σ_ρ] (m)	Azimuth [σ_β] (deg)	Elevation [$\sigma_{e\ell}$] (deg)	# Elements	$\theta_{3\text{dB}}$ (deg)
Eglin	32.1	0.0154	0.0147	5,184	1.4
Fylingdales	50.0	0.0220	0.0200	2,560	2.0
Clear	62.5	0.0791	0.0240	4,910	1.4
Cape Cod	26.0	0.0260	0.0220	1,792	2.4
Beale	35.0	0.0320	0.0330	1,792	2.4
Shemya	2.9	0.0540	0.0530	15,000	0.8
Thule	26.0	0.0260	0.0220	2,560	2.0
Cavalier	28.0	0.0125	0.0086	6,888	1.2

The last two columns of Table 7.3 include characteristics used for determining the volume of each sensor's FOV, and ultimately its clutter intensity κ_k. Each phased array radar consists of N radiating elements, tabulated in the fifth column [12, 54, 71]. The number of elements of a given radar is related to its 3 dB (decibel) angular beamwidth[4] $\theta_{3_{\text{dB}}}$ via [79]

$$\theta_{3_{\text{dB}}} = \frac{100}{\sqrt{N}}. \tag{7.1}$$

Angular beamwidth values are in the sixth column of Table 7.3. For each observation set, the predicted LV state is used to determine a reference range ρ_{ref} between it and the radar for use with track gating. For a given radar and ρ_{ref}, range gates are set at $\rho_{\text{ref}} \pm \sigma_\rho$, thus defining range boundaries for the radar's FOV. Using the formula for the volume of a cone, the FOV volume V_{FOV} is approximated by

$$V_{\text{FOV}} = \frac{\pi}{3}(\tan\theta_{3_{\text{dB}}})^2[(\rho_{\text{ref}} + \sigma_\rho)^3 - (\rho_{\text{ref}} - \sigma_\rho)^3]. \tag{7.2}$$

See Fig. 7.4 for a depiction of $\theta_{3_{\text{dB}}}$ and $\rho_{\text{ref}} \pm \sigma_\rho$.

[4] The 3 dB (decibel) angular beamwidth is also known as the half power beamwidth. For more information, see [54,79].

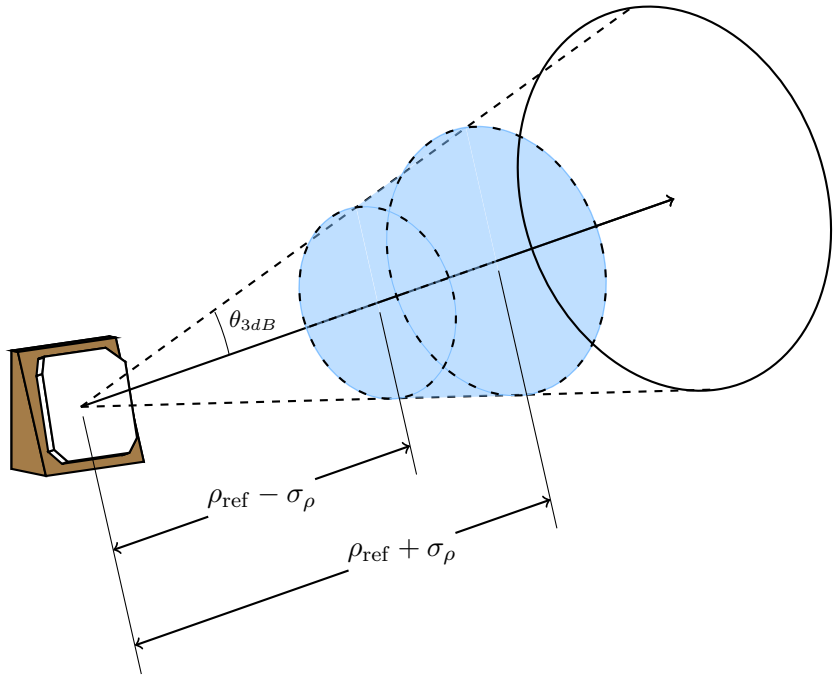


Figure 7.4: Phased array radar field of view volume.

7.1.2 Dynamics

As in Chapter 5, two-body dynamics with Earth gravity and drag perturbations are used, and model error is simulated by including up to the J_3 zonal harmonic in the gravity perturbations model when simulating the ground truth, but only up to the J_2 zonal harmonic is used in the filter's model. Dynamics model parameters for simulations in this chapter are presented in Table 7.4.

For both truth simulation and filtering, the CU-TurboProp orbit propagation package [40] was used, with an embedded Runge-Kutta 8(7) integrator [29] with a relative tolerance of 10^{-10}. Process noise is modeled as state noise compensation [86] where the process noise covariance Q is given by

$$Q = \begin{bmatrix} (5 \times 10^{-6})^2 \mathbf{1}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & (10^{-5})^2 \mathbf{1}_3. \end{bmatrix} \tag{7.3}$$

Position and velocity errors are placed on initial a priori objects states with standard deviations $\sigma_{\text{pos}} = 100\,\text{m}$ and $\sigma_{\text{vel}} = 0.5\,\text{m/s}$, respectively.

Table 7.4: Propagation for Observation Simulation and Filter Dynamics

Parameter	Truth Model	Filter Model
Gravity Perturbation	GGM02C [85]	GGM02C [85]
Spherical Harmonic Degree/Order	3×0	2×0
Area-to-Mass Ratio (m^2/kg)	0.01	0.01
Drag Coefficient C_d	2.0	2.0
Process Noise σ_Q (km/s^2)	N/A	10^{-10}

7.1.3 Filter Configuration

A single spawn model is used with both CubeSat deployment models. Spawn track position deviation vectors are depicted by red and blue squares inf Figs. 7.1-7.2. Spawn track velocity deviation vectors are defined as equivalent to their position deviation counterparts. Each track is represented as a single Gaussian component, with a covariance Q_{T} given as

$$Q_{\mathrm{T}} = \begin{bmatrix} (10^{-4})^2\mathbf{1}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & (10^{-8})^2\mathbf{1}_3 \end{bmatrix}. \tag{7.4}$$

Like the CPHD filter, special considerations must be made for the probability of detection $p_{\mathrm{D},k}$, due to the nature in which RSOs move in and out of a sensor's FOV. Detection probability augmentation is handled in the same way, as discussed in Section 2.8. Note, however, that additional considerations must be made for the GLMB filter regarding hypotheses updating. Care must be taken to ensure that when objects are out of the FOV, that the assignment costs of objects in the FOV are not improperly down-weighted. Essentially, the indexing scheme within the filter code must be partitioned to account for varying RSO access.

Pertinent filter parameters are presented in Table 7.5.

Table 7.5: GLMB filter configurations

Parameter	Setting
Survival Probability $p_{\mathrm{S}}/p_{\mathrm{S,tempd}}$	0.99 / $1-10^{-6}$
Detection Probability $p_{\mathrm{D}}/p_{\mathrm{D,tempd}}$	0.95 / $0.75 \times p_{\mathrm{D}}$
Spawn Probability $p_{\mathrm{T}}/p_{\mathrm{T,tempd}}$	0.07 / $1-10^{-4}$
Clutter Rate λ_c	10
Prune Threshold (Hypotheses)	10^{-19}
Maximum Number of Hypotheses	2000
Maximum Number of Tracks	1000
Prune Threshold (GM)	10^{-12}
Merge Threshold (GM)	4
Maximum Number of Components (GM)	5
OSPA Cutoff/Order (c/p)	10 (km)/2

7.2 Results

7.2.1 PSLV-deploy-1

Results from the PSLV-deploy-1 simulations are presented in Figs. 7.5-7.9. Cardinality statistics and OSPA errors are plotted over time in Figs. 7.5 and 7.6, respectively, showing that the filter accurately estimated the number of objects and their states throughout the entire scenario. Spikes in OSPA error occur many times, which is primarily due to long time intervals between observations as discussed below, however, the overall converging trend toward lower error is very apparent.

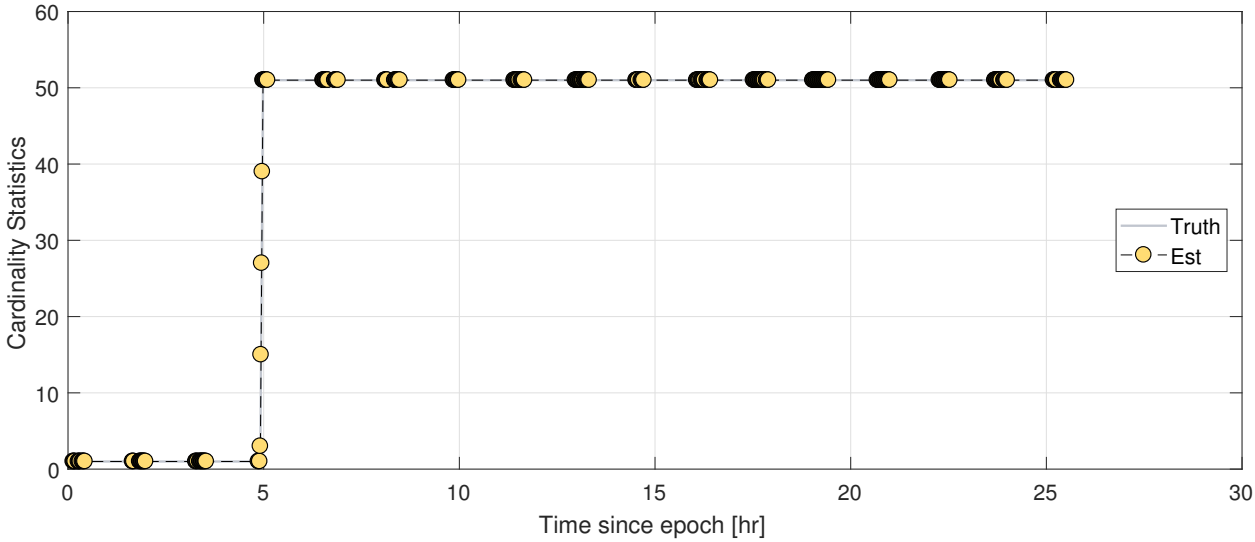


Figure 7.5: [PSLV-deploy-1] Cardinality statistics vs. time (single run).

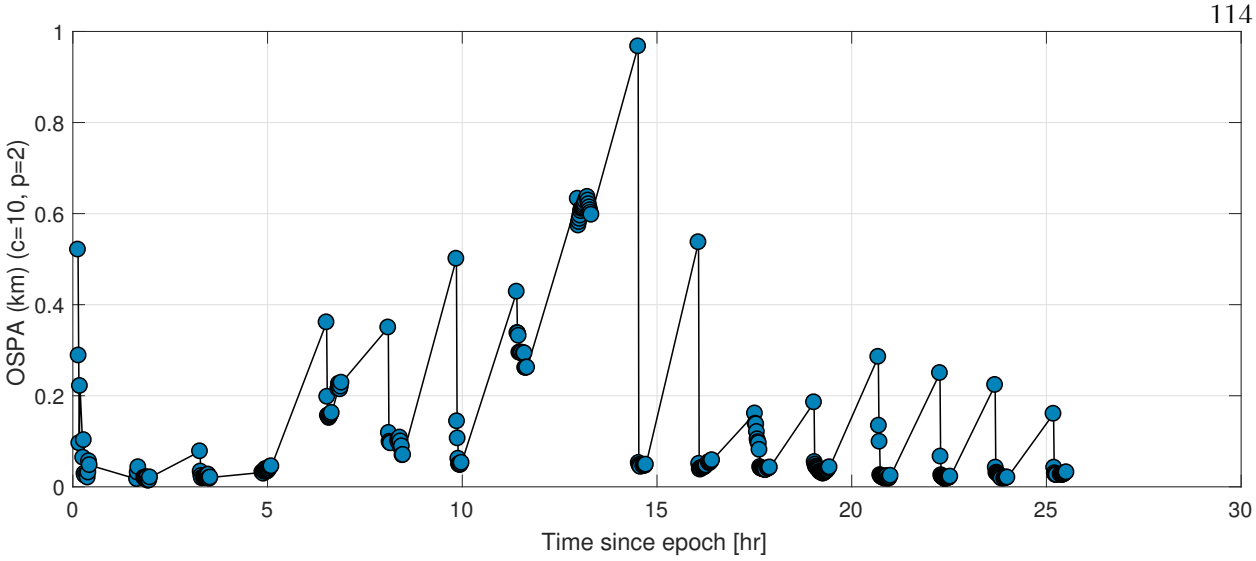


Figure 7.6: [PSLV-deploy-1] OSPA vs. time (single run).

Figs. 7.5 and 7.6 included plots over time, which gives a sense of overall scenario duration and the lengthy periods of time without observation. However, many times it is useful to investigate tracking results that are plotted over filter step, as it allows us to more closely examine behaviors by evenly distributing plot points horizontally, as opposed the how many of them get bunched together and mask variability in the plots above. Hence, the remainder of figures in this section include plots over filter step.

Figure 7.7 includes three subplots; the top is (total) OSPA error, OSPA localization errors in the middle, and OSPA cardinality errors at the bottom. Since the filter correctly estimated the number of objects throughout the entire scenario, both the top and middle plots are identical. As expected, at the beginning of CubeSat deployments, and for a short duration after, localization errors are increased, then eventually converge on noise level errors. The most prominent features of these plots are the intermittent spikes, which are primarily due to observation gaps. In Fig. 7.8, scenario time intervals vs. filter step are plotted over localization errors vs. filter step. The figure shows that the majority of peak localization errors occur just after a long observation gap, most of which are over an hour in duration.

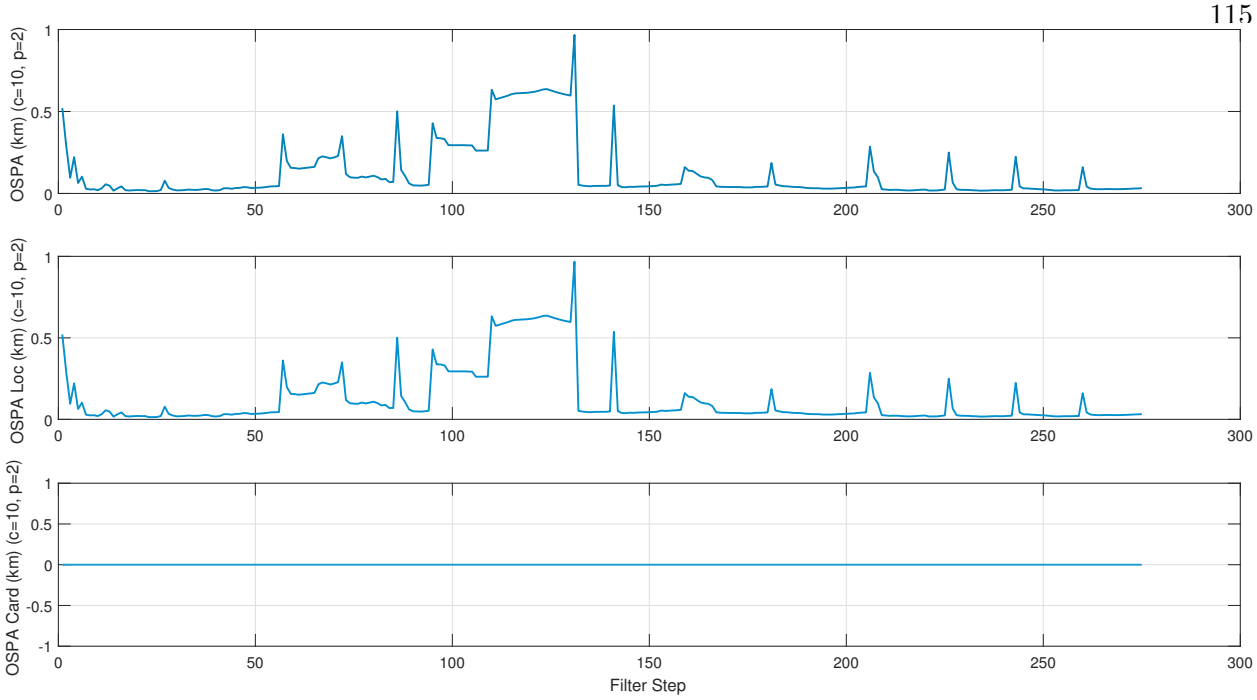


Figure 7.7: [PSLV-deploy-1] OSPA, localization, and cardinality errors vs. filter step (single run).

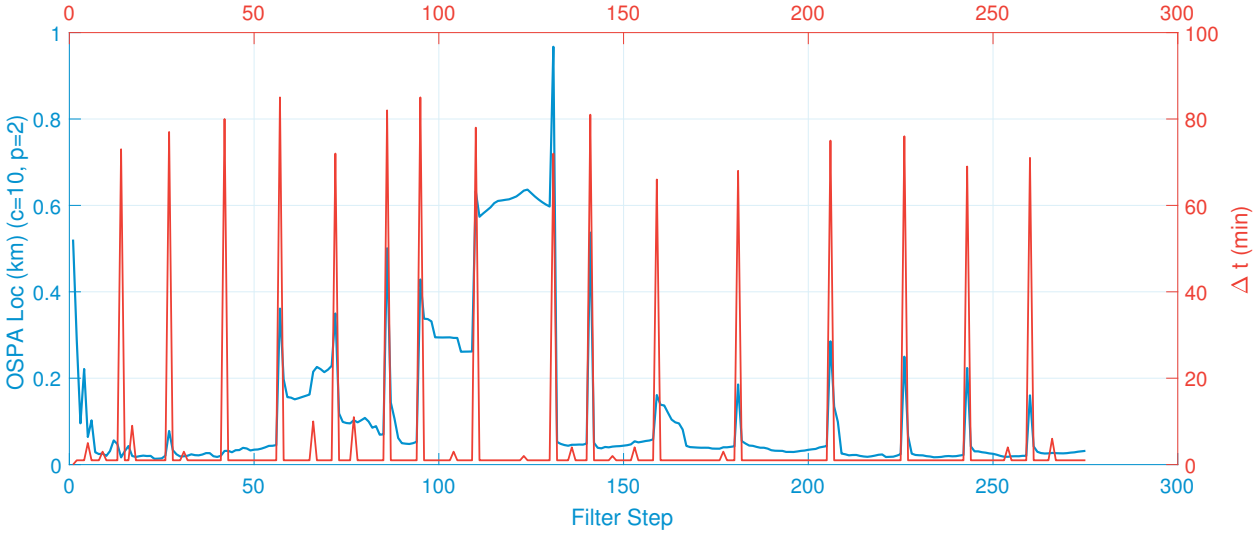


Figure 7.8: [PSLV-deploy-1] Localization errors plotted on the left vertical axis with the time intervals between filter steps plotted on the right vertical axis. Both are plotted over filter step (single run).

Another prominent feature of the localization error plots is the region of increased error centered at about the 120th filter step. Reviewing the number of missed detections vs. filter step plotted over localization errors vs. filter step in Figure 7.9, we see that two of the largest peaks in the number of missed detections occur at the beginning of this region, indicating that, at a

minimum, missed detections are a contributing factor to this period of increased error.

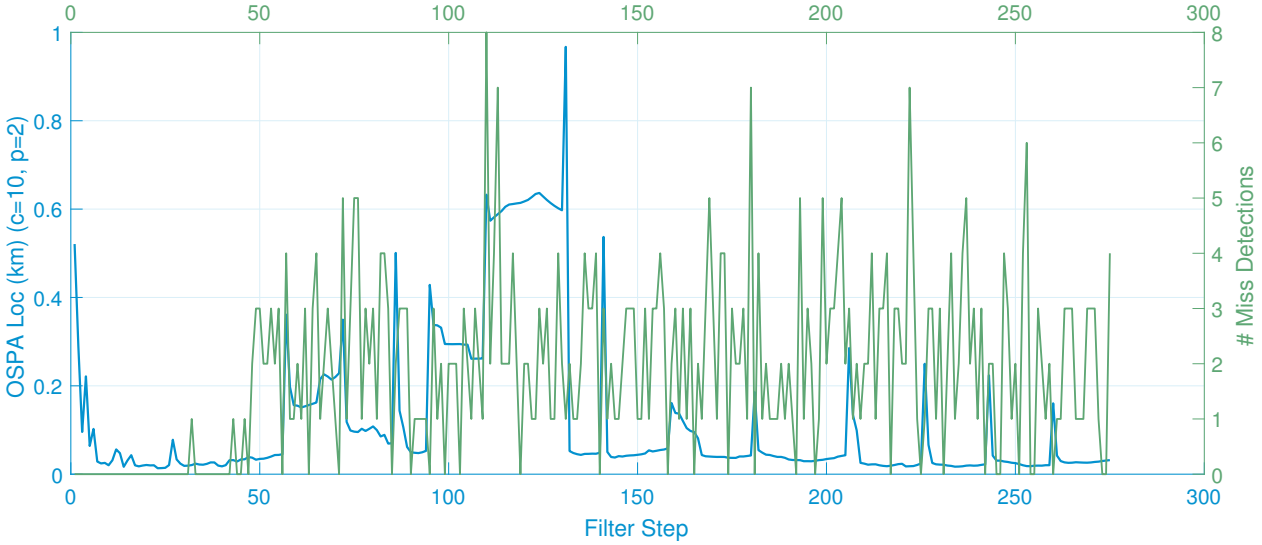


Figure 7.9: [PSLV-deploy-1] Localization errors plotted on the left vertical axis with the number of missed detections plotted on the right vertical axis. Both are plotted over filter step (single run).

7.2.2 PSLV-deploy-2

Results from the PSLV-deploy-2 simulations are presented in Figs. 7.10-7.14. Recall that a discrepancy between CubeSat deployment and filter spawn models was imposed in this scenario, so it is no surprise that the filter converged on a lower cardinality estimate, evidenced in Fig 7.10. Two CubeSat tracks were lost during simulation; one approximately 11.5 hours past epoch, the other at approximately 13 hours past epoch. The elevated OSPA error plotted in Fig 7.11 is due to the cardinality error, which is more easily seen in Fig. 7.12 where the OSPA error is broken into its components, plotted over filter step.

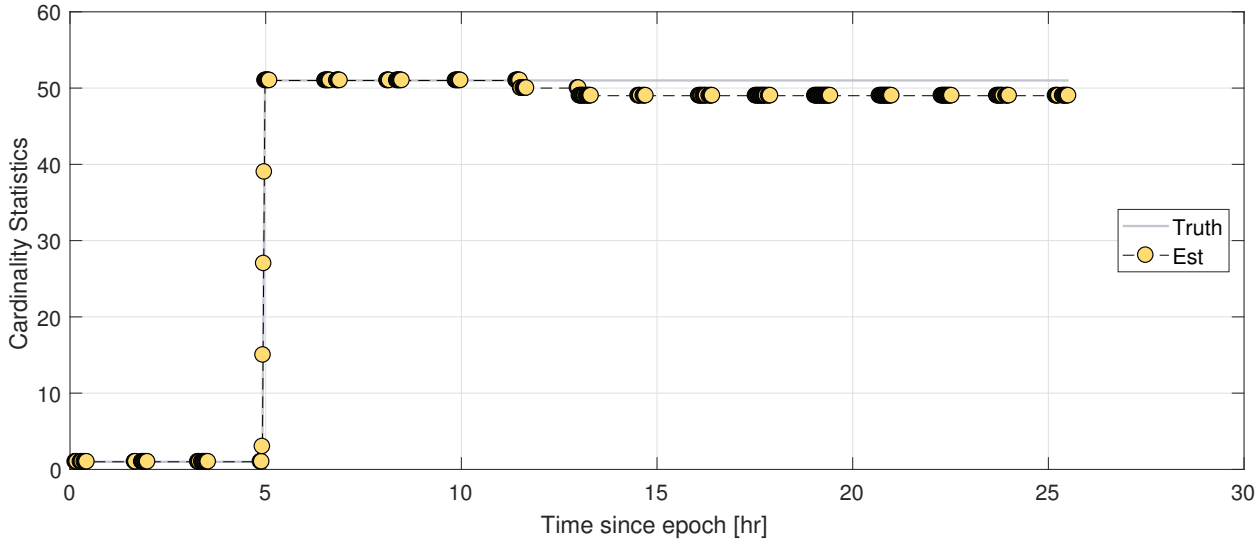


Figure 7.10: [PSLV-deploy-2] Cardinality statistics vs. time (single run).

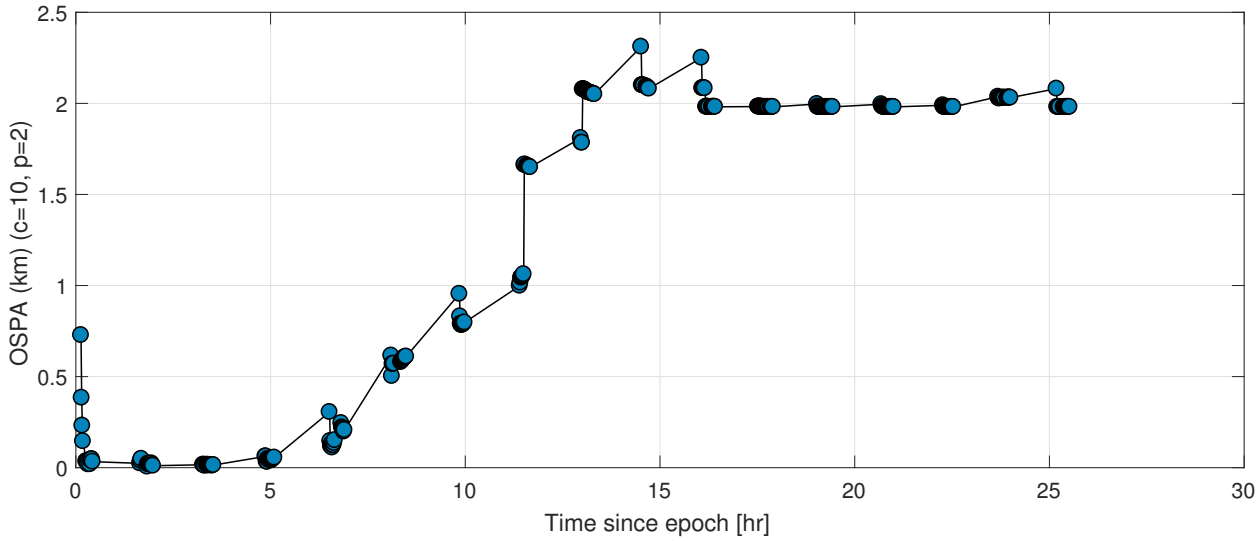


Figure 7.11: [PSLV-deploy-2] OSPA vs. time (single run).

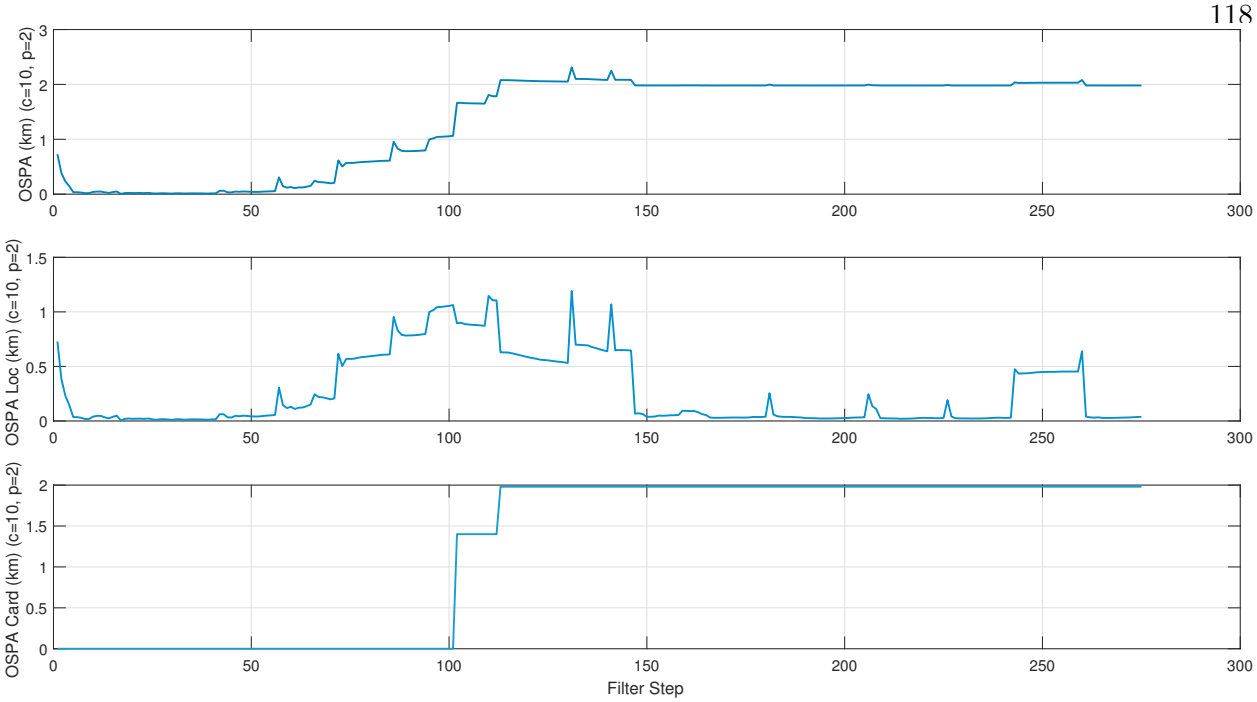


Figure 7.12: [PSLV-deploy-2] OSPA, localization, and cardinality errors vs. filter step (single run).

As with the first scenario, we see that peak observation gaps align perfectly with the majority of peak localization errors in Fig. 7.13. Similarly, the two more prominent number of miss detection peaks had a similar influence on the filter in this scenario as seen in Fig. 7.14, though for a shorter period of time as the miss detections occur near times of track loss.

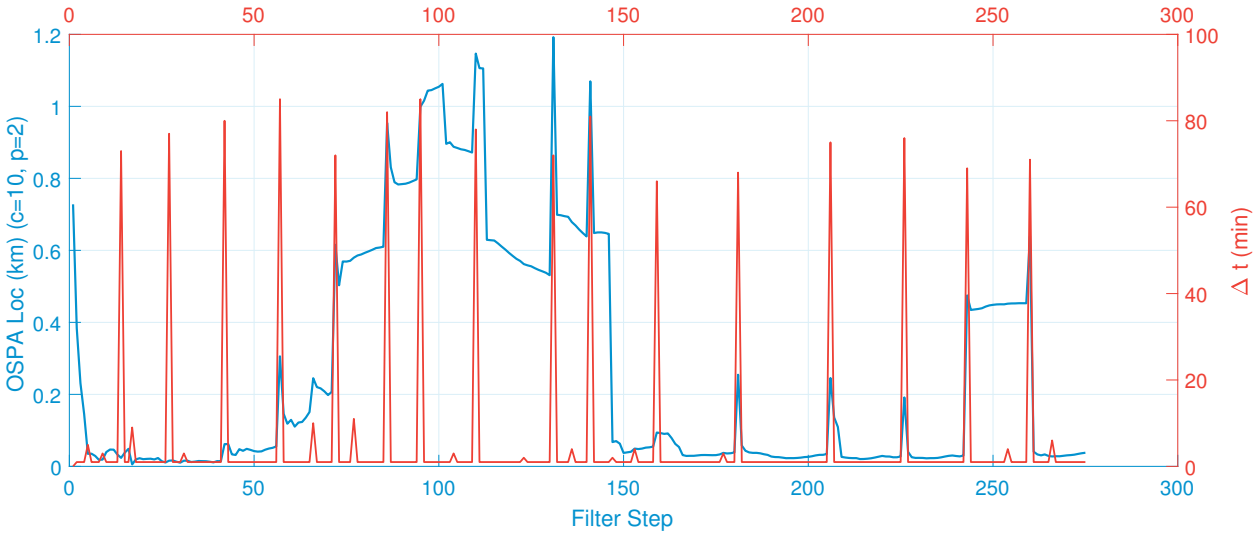


Figure 7.13: [PSLV-deploy-2] Localization errors plotted on the left vertical axis with the time intervals between filter steps plotted on the right vertical axis. Both are plotted over filter step (single run).

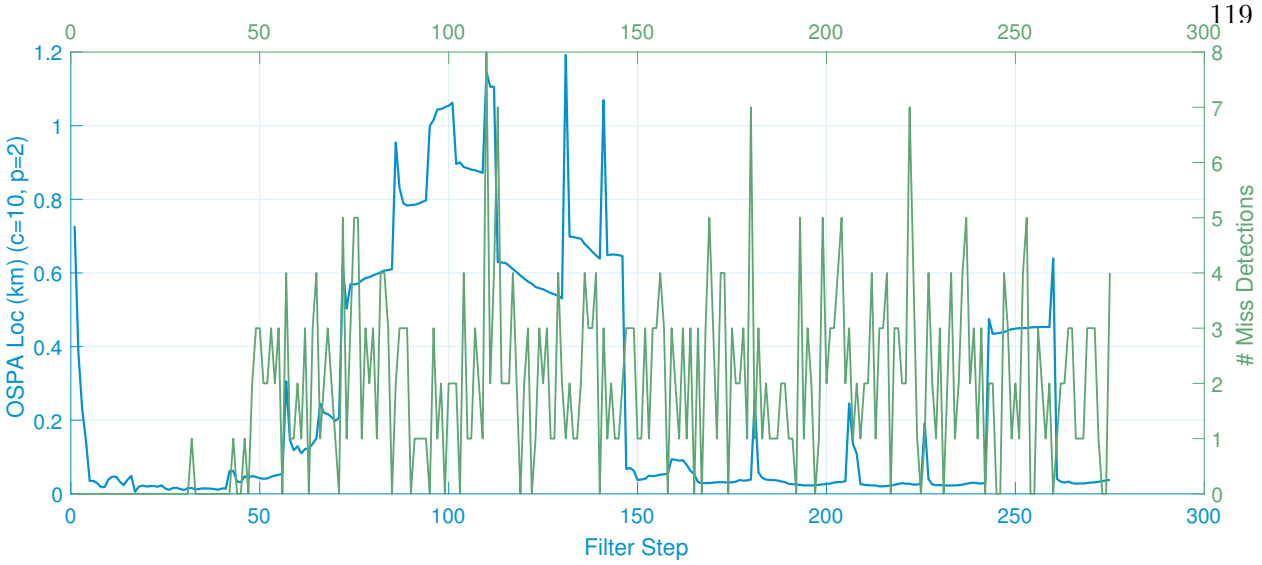


Figure 7.14: [PSLV-deploy-2] Localization errors plotted on the left vertical axis with the number of missed detections plotted on the right vertical axis. Both are plotted over filter step (single run).

Perhaps the most prominent feature of the localization error plot is the elevated region centered at approximately the 252nd filter step, a feature not present in the plots for the first scenario. Note that in Fig. 7.13, this region is bounded by two large observation gaps. Additionally, seen in Fig. 7.15, is that the region is also bounded by a sharp increase in observed range error, coinciding with the increase in localization error, and a sharp decrease in range error that coincides with the decrease in localization error. So it makes sense that there is an elevated error region in this case, because the filter is operating with a significant portion of its observation error increasing throughout a period of time that is bounded by two relatively lengthy periods where it is deprived of information.

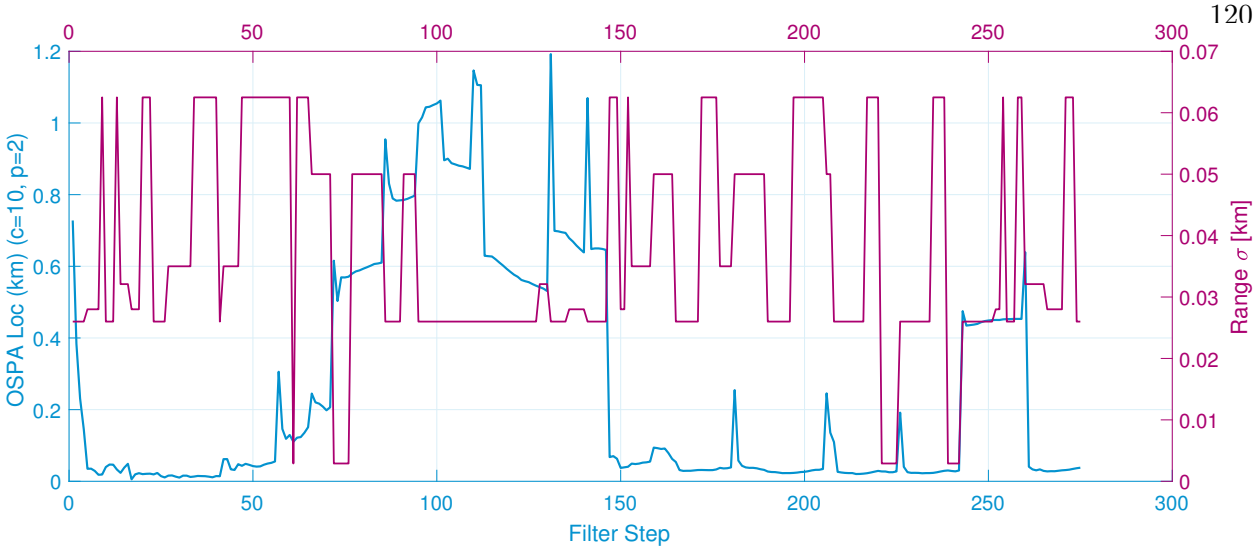


Figure 7.15: [PSLV-deploy-2] Localization errors plotted on the left vertical axis with Range error standard deviations plotted on the right vertical axis. Both are plotted over filter step (single run).

7.2.3 Discussion

These results have shown that the spawn model inclusive GLMB filter performs as expected. When sufficient information is available to model a spawning event with relatively high accuracy, the filter performs exactly as it was intended to. Furthermore, it was demonstrated that with a less accurate spawn model the filter does lose tracks, a behavior which was anticipated. However, in the given case, the filter was able to instantiate tracks for the majority of objects and provide comparable estimates for individual states, which demonstrates the filter's robustness.

One of the goals of this chapter is to demonstrate a proof of concept, in that the GLMB filter, configured with spawning, is capable of initial orbit determination. However, for the sake of demonstration, simplifying assumptions were made to accommodate the lack of available information pertaining to, e.g., the actual dynamics of small-sat deployments. In general, it is unlikely that prior information such as the timing of a spawning event will be available, especially if the event is a rocket body explosion, or the like. Hence, an additional goal of this chapter is to demonstrate to the SSA community that IOD within an RFS tracking framework, as applied to space based spawning events, is possible, and, to indicate areas in need of further investigation. The exploration

of more versatile spawn models for the IOD of a variety of space based spawning events is one such area.

Chapter 8

Conclusions and Future Work

This dissertation presented derivations of new, spawn model inclusive, multi-object filtering implementations and demonstrations of their ability to perform initial orbit determination for objects originating from a spawning event. This chapter summarizes the main results, discusses their contributions, and suggests future work with the potential for further improvement of multi-object filter implementations for SSA applications.

8.1 Summary of Results

The motivation for the work presented in Chapter 4 is the resolution of multi-object detection and tracking problems in which newborn objects are spawned from preexisting ones. To this end, the construction of a CPHD filter in which the appearance of newborn objects is modeled with a spawning mechanism rather than spontaneous birth was proposed, based on a principled derivation procedure within the FISST framework.

A GM implementation of the CPHD filter with spawning was presented, considering three different models for the spawning mechanism based on a Bernoulli, a Poisson, or a zero-inflated Poisson process. The three resulting filters were then illustrated, analyzed, and compared to a usual CPHD filter with spontaneous birth but no spawning, on the same simulated scenario involving two parent objects spawning a total of five new objects. Results demonstrated that a spawning model, appropriately chosen for a given application, can provide better estimates than a spontaneous birth model.

Spawning event orbit determination capabilities of the GM-CPHD filter were demonstrated in Chapter 5. CubeSat deployment and RB explosion models for the two different scenarios were detailed. Simulations involved the use of a full coverage, homogeneous sensor network capable of observing all objects at all times, and results demonstrated the implemented filter's efficacy of tracking the two event types.

Chapter 6 presented the first GLMB filter to consider object spawning. Using a top-down formulation, a general labeled RFS density characterizing the predicted multi-object density of surviving, birth and spawn objects was derived. A joint prediction-update was performed yielding a density that was then approximated to form a posterior GLMB density while preserving its cardinality and PHD. A key innovation of the proposed filter is the capacity of spawn track labels to encapsulate their ancestry. The filter's ability to instantiate new tracks originating from previously known objects was verified by simulation.

The first ever application of the GLMB multi-object tracking filter to the IOD of space based spawning events was presented in Chapter 7. It was shown that, even with discrepancy between spawn model and ground truth, the filter is capable of performing IOD for the majority of objects in a demanding scenario. The need for further exploration of spawn models for SSA applications was highlighted, and what remains to complete the presented research was discussed.

8.2 Contributions

Prior to this research, spawn model inclusive CPHD filter derivations existed, however, they were presented by a disjoint combination of FISST and non-FISST based derivations requiring special considerations for different spawn models, and, without a concise method for implementation. This research used a more fundamental approach to CPHD spawning derivations that not only yielded a highly configurable filter with an explicit means of implementation, but demonstrated a derivation method useful for extending first order approximate RFS filter implementations. Exploiting the p.g.fl. representation of point processes, i.e., RFSs, one can formulate a multi-object filtering problem of interest and develop a filter via differentiation. Furthermore, this research introduced

the Zero-Inflated Poisson spawn model whose development was directly inspired by the nature of fragmentation events. Tractability is a key concern with multi-object filtering, and maintaining it typically comes down to managing the number of filter components propagated from one time to the next. Fragmentation events occur infrequently, so it does not make sense to use a spawn model that consistently introduces components that persist through multiple filter steps, e.g., the Poisson model. Assuming the same Poisson rate is applied, the Zero-Inflated Poisson model will introduce the same number of components, however, they will be tempered by a probability of spawning that promotes more efficient component number management. The Zero-Inflated Poisson model was shown to be effective in initial orbit determination of objects originating from spawning events via simulation. When high observation density is achievable, the GM-CPHD filter can be a useful tool in estimating the number of objects generated by CubeSat deployments and/or fragmentation events and their states.

Significant contribution to the multi-object tracking community was also made by deriving a GLMB filter implementation that includes spawning. It is the first multi-object filter ever derived capable of providing ancestry information estimates, along with number of objects and individual object state estimates. The filter implementation itself has applications beyond that of the immediate goal of tracking space based spawning events. For instance, an imminent and exciting application involves the tracking of cancer cells, where the goal is to gain insight on their behavior via ancestry information. With the filter derivations presented in a fundamental form, the results of this research are accessible to any sub-discipline of the multi-object filtering/tracking field where spawning is applicable.

Further, this research has shown that on-line multi-object orbit determination in the presence of spawning is possible within the RFS filtering paradigm. Prior to this work, the only known means of determining the source of spawned RSOs and instantiating their tracks was by complex brute force methods with a strong potential need for analyst intervention. Having shown that the GLMB filter can perform initial orbit determination for spawned RSOs, this research has established a foundation upon which further SSA improvements can be investigated.

8.3 Future Work

There are several areas where further investigation can extend the presented research, at both fundamental and specific application levels. This section recommends lines of research for expanding filter versatility and implementation methods for improved SSA tracking.

There are at least two opportunities for exploring fundamental extensions to the GLMB tracking filter. The first is development of a measurement-based spawn model implementation. This thesis presented cases in which prior knowledge of spawning events was assumed, including the time at which they occur and the dynamics of spawned objects. In reality, such prior information is likely unavailable and configuring a filter to anticipate the spawning of multiple objects at every scan time is bound to be computationally cost prohibitive. A more elegant and potentially tractable approach would be to leverage information gathered from each observation scan to form spawn models. The caveat is, if special considerations are not made when using measurements in a prediction model, the filter can become biased and yield poor estimates.

Another potential area of fundamental filter research is exploring the implementation of Rao-Blackwellization with the GLMB filter. Rao-Blackwellization is a technique often used in particle filtering to maintain tractability when sampling in high-dimensional spaces, i.e., long state vectors, is required [30]. The state-vector is partitioned into two components, e.g., a conditionally linear Gaussian state-space model component that can be processed by Kalman filter and a non-linear non-Gaussian component processed by particle filter [73]. Use of this technique is prevalent in the Simultaneous Localization and Mapping (SLAM) community [15]; for example, a robot's trajectory is represented by the non-linear component of the state-vector estimated by a particle filter, and the robot's pose (orientation, attitude) is represented by the other state-vector component estimated by more tractable means [37,38]. Another application is group tracking, as described in [73], where individual object trajectories are estimated via particle filter and the behavior of the group as a whole is estimated by Kalman filter. Future work could explore a Rao-Blackwellization technique implemented with a GLMB filter instead of a particle filter.

The fundamental advancements discussed above would provide the opportunity to explore SSA specific applications. For example, a measurement based spawn model could be used in conjunction with a Lambert Solver for spawning event initial orbit determination. Determining the orbit between two known position vectors is typically referred to as Lambert's problem, solutions for which are provided by a Lambert Solver. Several variants exist, with the most commonly known solvers considering only two-body dynamics, i.e., perturbations, gravity, drag, or otherwise, are not considered, which would undoubtedly lead to issues with accuracy. However, a perturbation inclusive Lambert Solver was developed in [31] and is also the focus of more recent work [100]. With an object's estimated position at a given time and assuming that position information can be extracted from an observation at the next time, spawn model tracks could be instantiated by solving Lambert's problem.

Another area of active research is space debris cloud behavior [2, 3, 17, 24, 41, 44, 56]. It may be possible to leverage this wealth of information by using it to model the common behavior of a group of objects generated by fragmentation event, and subsequently using that model in a Rao-Blackwellized GLMB implementation. For example, studies have shown that there are multiple phases of debris cloud behavior, the first of which is a pulsating ellipsoid that lasts only a matter of hours [43]. Treating the debris fragments as if they were moving in an ellipsoidal formation, the tracking problem can be formulated such that a given state vector is partitioned into a component representing, e.g., an objects relative position in formation, and another component representing an individual object's state.

Finally, one line of research is the development of more robust small-sat deployment and fragmentation event spawn models for use with the currently available filter configurations. The small-sat deployment spawn models presented in this research worked for their given scenarios, but are by no means robust enough to handle any conceivable small-sat deployment scenario. Further investigation into the real-world mechanics of small-sat deployment from various types of launch vehicles and in various types of orbits is necessary for more robust spawn model development. The research presented here has provided a strong starting point for further investigation of the

challenging problem of fragmentation event filtering/tracking within the RFS framework; when coupled with models like the NASA breakup model, or with methods such as those used to solve Lambert's Problem, it could provide a fully automated means of performing data association for fragmentation event generated UCTs.

Bibliography

[1] Luciano Anselmo and Carmen Pardini. Analysis of the consequences in low Earth orbit of the collision between Cosmos 2251 and Iridium 33. In Proc. 21st Int. Symp. Sp. Flight Dyn., Toulouse, France, 2009.

[2] J. Ashenberg. On the short-term spread of space debris. In Proc. AIAA Astrodyn. Spec. Conf., 1992.

[3] J. Ashenberg. Formulas for the phase characteristics in the problem of low-Earth-orbital debris. J. Spacecr. Rockets, 31(6):1044–1049, Nov. 1994.

[4] Richard M. Baker, Megan E. Brasch, M. Lisa Manning, and James H. Henderson. Automated, contour-based tracking and analysis of cell behaviour over long time scales in environments of varying complexity and cell density. J. R. Soc. Interface, 11(97), 2014.

[5] Y. Bar-Shalom. Tracking and Data Fusion: A Handbook of Algorithms. YBS Publishing, Bloomfield, CT, USA, 2011.

[6] Y. Bar-Shalom, T. E. Fortmann, and M. Scheffe. Joint probabilistic data association for multiple targets in clutter. In Proc. Conf. Inf. Sci. Syst., pages 404–409, 1980.

[7] Michael Beard, Ba-Ngu Vo, Ba-Tuong Vo, and Sanjeev Arulampalam. A partially uniform target birth model for Gaussian mixture PHD/CPHD Filtering. IEEE Trans. Aerosp. Electron. Syst., 49(4):2835–2844, 2013.

[8] Michael Beard, Ba-Tuong Vo, Ba-Ngu Vo, and Sanjeev Arulampalam. Void probabilities and Cauchy-Schwarz divergence for generalized labeled multi-Bernoulli models. IEEE Trans. Signal Process., 65(19):5047–5061, Oct. 2017.

[9] T. Becker and A. Madany. Morphology-based features for adaptive mitosis detection of in vitro stem cell tracking data. Methods Inf. Med., 51(5):449, 2012.

[10] Pierre Bernhard. Chain differentials with an application to the mathematical fear operator. Nonlinear Anal. Theory, Methods Appl., 62(7):1225–1233, Sep. 2005.

[11] Samuel Blackman and Robert Popoli. Design and Analysis of Modern Tracking Systems. Artech House, Norwood, MA, 1999.

[12] Eli Brookner. Large phased-array radars. In AIP Conf. Proc., pages 165–205. AIP, 1988.

[13] D. S. Bryant and B. A. Jones. Fragmentation event tracking with the GM-CPHD filter. In AAS/AIAA Spacefl. Mech. Meet., 2016.

[14] Daniel S. Bryant, Emmanuel D. Delande, Steven Gehly, Jérémie Houssineau, Daniel E. Clark, and Brandon A. Jones. The CPHD filter with target spawning. IEEE Trans. Signal Process., 65(5):13124–13138, Mar. 2017.

[15] Luca Carlone, Jingjing Du, Miguel Kaouk Ng, Basilio Bona, and Marina Indri. Active SLAM and Exploration with Particle Filters Using Kullback-Leibler Divergence. J. Intell. Robot. Syst., 75(2):291–311, Aug. 2014.

[16] C. A. Charalambides. Enumerative combinatorics. CRC Press, 2002.

[17] Vladimir A. Chobotov, D. B. Spencer, D. L. Schmitt, R. P. Gupta, D. T. Knapp, and R. G. Hopkins. Dynamics of Debris Motion and the Colllision Hazard to Spacecraft Resulting from an Orbital Breakup. Technical report, The Aerospace Corp., 1988.

[18] Daniel E. Clark and Jérémie Houssineau. Faà di Bruno's formula and spatial cluster modelling. Spat. Stat., 6:109–117, 2013.

[19] Daniel E. Clark and Jérémie Houssineau. Faà di Bruno's formula for chain differentials. arXiv:1310.2833, 2013.

[20] Daniel E. Clark, Jérémie Houssineau, and Emmanuel D. Delande. A few calculus rules for chain differentials. arXiv:1506.08626, 2015.

[21] Daniel E. Clark, Kusha Panta, and Ba-Ngu Vo. The GM-PHD filter multiple target tracker. 2006 9th Int. Conf. Inf. Fusion, 2006.

[22] Committee for the Assessment of the US Air Force's Astrodynamic Standards and Aeronautics and Space Engineering Board and Division on Engineering and Physical Sciences and National Research Council (US). Continuing Kepler's Quest: Assessing Air Force Space Command's Astrodynamics Standards. National Academies Press, 2012.

[23] Paula Craciun, Mathias Ortner, and Josiane Zerubia. Joint detection and tracking of moving objects using spatio-temporal marked point processes. In 2015 IEEE Winter Conf. Appl. Comput. Vis., pages 177–184. IEEE, Jan. 2015.

[24] R. Crowther. Modeling the short-term evolution of orbital debris clouds in circular orbits. J. Spacecr. Rockets, 31(4):709–711, Jul. 1994.

[25] D. J. Daley and D. Vere-Jones. An Introduction to the Theory of Point Processes, Volume I. Springer-Verlag, New York, 2 edition, 2003.

[26] Kyle J. DeMars, Robert H. Bishop, and Moriba K. Jah. Entropy-Based Approach for Uncertainty Propagation of Nonlinear Dynamical Systems. J. Guid. Control. Dyn., 36(4):1047–1057, 2013.

[27] Kyle J. DeMars and James S McCabe. Tracking of the LANDSAT 2 rocket body primary breakup. In AAS/AIAA Spacefl. Mech. Meet., number AAS:15-420, Williamsburg, VA, Jan. 2015.

[28] Faà di Bruno. Sullo Sviluppo delle Funzioni. Ann. di Sci. Mat. e Fis., 6:479–480, 1885.

[29] J. R. Dormand and P. J. Prince. A family of embedded Runge-Kutta formulae. J. Comput. Appl. Math., 6(1):19–26, Mar. 1980.

[30] Arnaud Doucet, J. de Freitas, K. Murphy, and S. Russel. Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks. In Conf. Uncertain. Artif. Intell., pages 176–183, 2000.

[31] R. C. Engels and J. L. Junkins. The gravity-perturbed Lambert problem: A KS variation of parameters approach. Celest. Mech., 24(1):3–21, 1981.

[32] D. Fränken, M. Schmidt, and M. Ulmke. "Spooky action at a distance" in the cardinalized probability hypothesis density filter. IEEE Trans. Aerosp. Electron. Syst., 45(4):1657–1664, 2009.

[33] Joseph W Gangestad, Brian S Hardy, and David a Hinkley. Operations , Orbit Determination, and Formation Control of the AeroCube-4 CubeSats. 27th Annu. AIAA/USU Conf. Small Satell., 2013.

[34] Steven Gehly, Brandon Jones, and Penina Axelrad. Sensor Allocation for Tracking Geosynchronous Space Objects. J. Guid. Control. Dyn., In Press, Aug. 2016.

[35] Steven Gehly, Brandon A. Jones, and Penina Axelrad. An AEGIS-CPHD Filter to Maintain Custody of GEO Space Objects with Limited Tracking Data. In Proc. 2014 Adv. Maui Opt. Sp. Surveill. Technol. Conf., pages 25–34, 2014.

[36] Alison L. Gibbs and Francis Edward Su. On Choosing and Bounding Probability Metrics. Int. Stat. Rev., 70(3):419–435, 2002.

[37] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters. IEEE Trans. Robot., 23(1):34–46, Feb. 2007.

[38] Brian Hampton, Akram Al-Hourani, Branko Ristic, and Bill Moran. RFS-SLAM robot: An experimental platform for RFS based occupancy-grid SLAM. In 2017 20th Int. Conf. Inf. Fusion, pages 1–8. IEEE, Jul. 2017.

[39] Hank Heidt, Jordi Puig-Suari, Augustus S. Moore, Shinichi Nakasuka, and Robert J. Twiggs. CubeSat: A new Generation of Picosatellite for Education and Industry Low-Cost Space Experimentation. AIAA/USU Conf. Small Satell., pages 1–19, 2000.

[40] K. Hill and Brandon A. Jones. TurboProp Version 4.0. Colorado Center for Astrodynamics Research (CCAR), 2009.

[41] F. R. Hoots and B. W. Hansen. Satellite breakup debris cloud characterization. In 24th AAS/AIAA Sp. Flight Mech. Meet., 2014.

[42] Jean Jacod and Philip Protter. Probability Essentials. Springer, Berlin, 2004.

[43] Ruediger Jehn. Dispersion of debris clouds from on-orbit fragmentation events. In Dresden Int. Astronaut. Fed. Congr., volume 1, 1990.

[44] Alan Jenkin and Marlon Sorge. Debris clouds in eccentric orbits. In Sp. Programs Technol. Conf., Reston, Virigina, Sep. 1990. American Institute of Aeronautics and Astronautics.

[45] Nicholas L. Johnson, P. H. Krisko, J. C. Liou, and P. D. Anz-Meador. NASA's new breakup model of EVOLVE 4.0. Adv. Sp. Res., 28(9):1377–1384, 2001.

[46] Nicholas L. Johnson, E. Stansbery, J. C. Liou, M. Horstman, C. Stokely, and D. Whitlock. The characteristics and consequences of the break-up of the Fengyun-1C spacecraft. Acta Astronaut., 63(1-4):128–135, 2008.

[47] Nicholas L. Johnson, David O. Whitlock, P. Anz-Meador, M. Eleanor Cizek, S. A. Portman, Eugene Stansbery, Kira J. Abercromby, and Debra Shoots. History of on-orbit satellite fragmentations 14th edition. Houst. NASA, (June), 2008.

[48] Brandon A. Jones. Modeling Birth in a Space-Object CPHD Filter Using the Probabilistic Admissible Region. In Inf. Fusion (FUSION), 2016 Proc. 19th Int. Conf., pages 162–169, 2016.

[49] Brandon A. Jones, Daniel S. Bryant, Ba-Ngu Vo, and Ba-Tuong Vo. Challenges of multi-target tracking for space situational awareness. 18th Int. Conf. Inf. Fusion, pages 1278–1285, 2015.

[50] Brandon A. Jones, Steven Gehly, and Penina Axelrad. Measurement-based birth model for a space object cardinalized probability hypothesis density filter. In Proc. 2014 AIAA/AAS Astrodyn. Spec. Conf., 2014.

[51] S. J. Julier and J. K. Uhlmann. Unscented filtering and nonlinear estimation. Proc. IEEE, 92 Issue 3:401–422, Mar. 2004.

[52] Simon J. Julier and Jeffrey K. Uhlmann. A new extension of the Kalman filter to nonlinear systems. In Int. symp. aerospace/defense sensing, simul. Control., volume 3, pages 2–3. Orlando, FL, 1997.

[53] C. Kilic, T. Scholz, and C. Asma. Deployment strategy study of QB50 network of CubeSats. In Recent Adv. Sp. Technol. (RAST), 2013 6th Int. Conf., pages 935–939, Jun. 2013.

[54] Heiner Klinkrad. Space Debris: Models and Risk Analysis. Springer-Verlag, Berlin, 2006.

[55] Diane Lambert. Zero-inflated Poisson regression, with an application to defects in manufacturing. Technometrics, 34(1):1–14, 1992.

[56] Francesca Letizia, Camilla Colombo, and Hugh G. Lewis. Analytical Model for the Propagation of Small-Debris-Object Clouds After Fragmentations. J. Guid. Control. Dyn., 38(8):1478–1491, Aug. 2015.

[57] Kang Li, Eric D. Miller, Mei Chen, Takeo Kanade, Lee E. Weiss, and Phil G. Campbell. Cell population tracking and lineage construction with spatiotemporal context. Med. Image Anal., 12(5):546–566, 2008.

[58] Lin Lin, Yaakov Bar-Shalom, and T. Kirubarajan. Track labeling and PHD filter for multi-target tracking. Aerosp. Electron. Syst. IEEE Trans., 42(3):778–793, 2006.

[59] Malin Lundgren, Lennart Svensson, and Lars Hammarstrand. A CPHD Filter for Tracking with Spawning Models - inlcuding a FISST based derivation. Technical Report R006/2012, Department of Signals and Systems, Chalmers University of Technology, 2012.

[60] Malin Lundgren, Lennart Svensson, and Lars Hammarstrand. A CPHD filter for tracking with spawning models. IEEE J. Sel. Top. Signal Process., 7(3):496–507, 2013.

[61] Ronald Mahler. Multitarget Bayes filtering via first-order multi-target moments. IEEE Trans. Aerosp. Electron. Syst., 39(4):1152–1178, 2003.

[62] Ronald Mahler. PHD filters of higher order in target number. IEEE Trans. Aerosp. Electron. Syst., 43(4):1523–1543, 2007.

[63] Ronald Mahler. Statistical Multisource-Multitarget Information Fusion. Artech House, Norwood, MA, 2007.

[64] Ronald Mahler. Advances in Statistical Multisource-Multitarget Information Fusion. Artech House, Norwood, MA, 2014.

[65] Vasileios Maroulas and Ronald Mahler. Tracking spawning objects. Radar, Sonar Navig. IET, 7(3):321–331, 2013.

[66] Erik Meijering, Oleh Dzyubachyk, and Ihor Smal. Methods for cell and particle tracking. In P Michael Conn, editor, Methods Enzymol., volume 504, pages 183–200. Academic Press, 2012.

[67] J. E. Moyal. The general theory of stochastic population processes. Acta Math., 108(1):1–31, 1962.

[68] K. G. Murty. An algorithm for ranking all the assignments in order of increasing cost. Oper. Res., 16(3):682–687, 1968.

[69] Kusha Panta, Ba-Ngu-Vo, and Daniel E. Clark. An efficient track management scheme for the Gaussian-mixture probability hypothesis density tracker. In Intell. Sens. Inf. Process. 2006. ICISIP 2006. Fourth Int. Conf., pages 230–235. IEEE, 2006.

[70] Francesco Papi, Ba-Ngu Vo, Ba-Tuong Vo, Claudio Fantacci, and Michael Beard. Generalized labeled multi-Bernoulli approximation of multi-object densities. IEEE Trans. Signal Process., 63(20):5487–5497, Oct. 2015.

[71] J.E. Reed. The AN/FPS-85 radar system. Proc. IEEE, 57(3):324–335, 1969.

[72] Donald Reid. An algorithm for tracking multiple targets. IEEE Trans. Automat. Contr., 24(6):843–854, 1979.

[73] B. Ristic, S. Arulampalam, and Neil J. Gordon. Beyond the Kalman filter. Artech House, 2004.

[74] Branko Ristic, Daniel E. Clark, Ba-Ngu Vo, and Ba-Tuong Vo. Adaptive target birth intensity for PHD and CPHD filters. IEEE Trans. Aerosp. Electron. Syst., 48(2):1656–1668, Apr. 2012.

[75] Dominic Schuhmacher, Ba-Tuong Vo, and Ba-Ngu Vo. A consistent metric for performance evaluation of multi-object filters. IEEE Trans. Signal Process., 56(8):3447–3457, 2008.

[76] Paul W. Schumacher Jr. Tracking satellite break-ups. In US-Russian 2nd Sp. Surveill. Work. Conf. Proc. DTIC Document, 1996.

[77] Paul W. Schumacher Jr., Chris Sabol, Alan Segerman, Aaron Hoskins, Shannon Coffey, Kathy Borelli, Jason Addison, Keric Hill, Kevin Roe, and Bruce Duncan. Search and Determine Integrated Environment (SADIE) for automated processing of space surveillance observations. In AIAA/AAS Astrodyn. Spec. Conf. AIAA, 2014.

[78] Navraj Singh, Joshua T. Horwood, Jeffrey M. Aristoff, Aubrey B. Poore, Carolyn Sheaff, and Moriba K. Jah. Multiple Hypothesis Tracking (MHT) for Space Surveillance: Results and Simulation Studies. Technical report, DTIC Document, Retrieved from http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA591392, 2013.

[79] Merrill Ivan Skolnik. Radar handbook. 1970.

[80] S. K. Srinivasan. Stochastic Point Processes and Their Applications. Griffin, 1974.

[81] S. K. Srinivasan and A. Vijayakumar. Point Processes and Product Densities. Alpha Science International, 2003.

[82] D. Stoyan, W. S. Kendall, and J. Mecke. Stochastic geometry and its applications. Akademie-Verlag, Berlin, 2 edition, 1995.

[83] M. Swartwout. Cheaper by the dozen: The avalanche of rideshares in the 21st century. In 2013 IEEE Aerosp. Conf., pages 1–12. IEEE, Mar. 2013.

[84] Michael A. Swartwout. A brief history of rideshares (and attack of the CubeSats). Aerosp. Conf. 2011 IEEE, pages 1–15, Mar. 2011.

[85] B. Tapley, J. Ries, S. Bettadpur, D. Chambers, M. Cheng, F. Condi, B. Gunter, Z. Kang, P. Nagel, and R. Pastor. GGM02–An improved Earth gravity field model from GRACE. J. Geod., 79(8):467–478, 2005.

[86] Byron D. Tapley, Bob E. Schutz, and George H. Born. Statistical Orbit Determination. Elsevier Academic Press, Burlington, MA, 2004.

[87] J. Thoemel, F. Singarayar, T. Scholz, D. Masutti, P. Testani, C. Asma, R. Reinhard, and J. Muylaert. Status of The QB50 CubeSat Constellation Mission. In Proc. 65th Int. Astronaut. Congr. Toronto, Canada, 2014.

[88] David A. Vallado. Fundamentals of astrodynamics and applications. Springer, 3rd edition, 2007.

[89] Ba-Ngu Vo and Wing-Kin Ma. The Gaussian Mixture Probability Hypothesis Density Filter. IEEE Trans. Signal Process., 54(11):4091–4104, 2006.

[90] Ba-Ngu Vo, S. Singh, and A. Boucet. Sequential Monte Carlo methods for multi-target filtering with random finite sets. IEEE Trans. Aerosp. Electron. Syst., 41(4):1224–1245, Oct. 2005.

[91] Ba-Ngu Vo and Ba-Tuong Vo. Multi-sensor multi-object tracking with the generalized labeled multi-Bernoulli filter. arXiv:1702.08849, 2017.

[92] Ba-Ngu Vo, Ba-Tuong Vo, and H. G. Hoang. An efficient implementation of the generalized labeled multi-Bernoulli filter. IEEE Trans. Signal Process., 65(8):1975–1987, Apr. 2017.

[93] Ba-Ngu Vo, Ba-Tuong Vo, Nam-Trung Pham, and David Suter. Joint detection and estimation of multiple objects from image observations. IEEE Trans. Signal Process., 58(10):5129–5141, Oct. 2010.

[94] Ba-Ngu Vo, Ba-Tuong Vo, and Dinh Phung. Labeled random finite sets and the Bayes multi-target tracking filter. IEEE Trans. Aerosp. Electron. Syst., 62(24):6554–6567, Dec. 2014.

[95] Ba-Tuong Vo and Ba-Ngu Vo. Labeled random finite sets and multi-object conjugate priors. IEEE Trans. Signal Process., 61(13):3460–3475, Jul. 2013.

[96] Ba-Tuong Vo, Ba-Ngu Vo, and Antonio Cantoni. Analytic Implementations of the Cardinalized Probability Hypothesis Density Filter. IEEE Trans. Signal Process., 55(7):3553–3567, 2007.

[97] Ba-Tuong Vo, Ba-Ngu Vo, and Antonio Cantoni. The cardinality balanced multi-target multi-Bernoulli filter and its implementations. IEEE Trans. Signal Process., 57(2):409–423, 2009.

[98] Tuyet Vu, Ba-Ngu Vo, and Rob Evans. A particle marginal Metropolis-Hastings multi-target tracker. IEEE Trans. Signal Process., 62(15):3953–3964, Aug. 2014.

[99] H. W. Watson and Francis Galton. On the probability of the extinction of families. J. Anthropol. Inst. Gt. Britain Irel., 4:138, 1875.

[100] Robyn M. Woollands, Ahmad Bani Younes, and John L. Junkins. New Solutions for the Perturbed Lambert Problem Using Regularization and Picard Iteration. J. Guid. Control. Dyn., 38(9):1548–1562, 2015.

Appendix A

Derivations: CPHD with Spawning

A.1 Proof of theorem 1

For the sake of simplicity, the time subscripts will be omitted throughout the proof when there is no ambiguity. Additionally, the π's used notationally in this appendix are not related to the multi-object densities in other chapters. Here they are simply variables used to denote sets.

A.1.1 Predicted p.g.fl.

Let us focus first on the p.g.fl. $G_{k|k-1}$ of the predicted multi-object point process $\Omega_{k|k-1}$. Each parent object in the population, represented by the prior point process Ω_{k-1}, generates spawned objects in the predicted population in two ways:

- a spawned object stemming from the (eventual) survival of the parent object, represented by a survival point process Ω_{S},
- a population of spawned objects from the parent object, represented by a spawning point process Ω_{T}.

Using Eq. (3.8), and denoting by G_{S} and G_{T} the p.g.fl.s of the survival and spawning point processes, respectively, we can describe the evolution of a parent object with state $x \in \mathbb{X}$ with a compound process with p.g.fl.

$$G_{\mathrm{c}}(h|x) = G_{\mathrm{S}}(h|x)G_{\mathrm{T}}(h|x). \tag{A.1}$$

In addition, a population of newborn objects is generated independently of the prior objects, represented by a spontaneous birth process Ω_{B} whose p.g.fl. is denoted by G_{B}. Exploiting the Galton-Watson equation (3.9), we may finally write

$$G_{k|k-1}(h) = G_{k-1}(G_{\mathrm{c}}(h|\cdot))G_{\mathrm{B}}(h) \tag{A.2a}$$

$$= G_{k-1}(G_{\mathrm{S}}(h|\cdot)G_{\mathrm{T}}(h|\cdot))G_{\mathrm{B}}(h). \tag{A.2b}$$

A.1.2 Predicted intensity

Let us now focus on the expression of the predicted intensity $\mu_{k|k-1}$. For that, let us fix an arbitrary measurable subset $B \in \mathcal{B}_{\mathbb{X}}$. The expression of the intensity evaluated in B can be recovered from the first derivative of the p.g.fl. $G_{k|k-1}$ using Eq. (3.20):

$$\mu_{k|k-1}(B) = \delta G_{k|k-1}(h; 1_B)\big|_{h=1}, \tag{A.3}$$

$$= \delta\big(G_{k-1}(G_{\mathrm{c}}(h|\cdot))G_{\mathrm{B}}(h); 1_B\big)\big|_{h=1}. \tag{A.4}$$

Using the product rule (3.14) it becomes

$$\mu_{k|k-1}(B) = \delta\big(G_{k-1}(G_{\mathrm{c}}(h|\cdot)); 1_B\big)\big|_{h=1} \underbrace{G_{\mathrm{B}}(1)}_{=1} + \underbrace{G_{k-1}(G_{\mathrm{c}}(1|\cdot))}_{=1} \delta G_{\mathrm{B}}(h; 1_B)|_{h=1}. \tag{A.5}$$

Using the definition of the p.g.fl. (3.5a) then yields

$$\mu_{k|k-1}(B) = \delta\bigg(\int_{\mathcal{X}} \bigg[\prod_{x\in\varphi} G_{\mathrm{c}}(h|x) \bigg] P_{k-1}(\mathrm{d}\varphi); 1_B \bigg)\bigg|_{h=1} + \delta G_{\mathrm{B}}(h; 1_B)|_{h=1}, \tag{A.6}$$

$$= \int_{\mathcal{X}} \delta\bigg(\prod_{x\in\varphi} G_{\mathrm{c}}(h|x); 1_B \bigg)\bigg|_{h=1} P_{k-1}(\mathrm{d}\varphi) + \delta G_{\mathrm{B}}(h; 1_B)|_{h=1}. \tag{A.7}$$

From the product rule (3.14) it follows that

$$\mu_{k|k-1}(B) = \int_{\mathcal{X}} \sum_{x\in\varphi} \bigg[\delta G_{\mathrm{c}}(h|x; 1_B)\bigg|_{h=1} \prod_{\substack{\bar{x}\in\varphi \\ \bar{x}\neq x}} \underbrace{G_{\mathrm{c}}(1|\bar{x})}_{=1} \bigg] P_{k-1}(\mathrm{d}\varphi) + \delta G_{\mathrm{B}}(h; 1_B)|_{h=1}. \tag{A.8}$$

Using the product rule (3.14) on $G_{\mathrm{c}}(\cdot|x) = G_{\mathrm{S}}(\cdot|x)G_{\mathrm{T}}(\cdot|x)$ then yields

$$\mu_{k|k-1}(B) = \int_{\mathcal{X}} \sum_{x\in\varphi} \bigg[\delta G_{\mathrm{S}}(h|x; 1_B)\big|_{h=1} \underbrace{G_{\mathrm{T}}(1|x)}_{=1} \underbrace{G_{\mathrm{S}}(1|x)}_{=1} \delta G_{\mathrm{T}}(h|x; 1_B)\big|_{h=1} \bigg] P_{k-1}(\mathrm{d}\varphi)$$

$$+ \delta G_{\mathrm{B}}(h; 1_B)|_{h=1}. \tag{A.9}$$

Using Eq. (3.20) we introduce the intensities μ_{S}, μ_{T}, and μ_{B} of the survival, spawning, and spontaneous birth processes and we obtain:

$$\mu_{k|k-1}(B) = \int_{\mathcal{X}} \sum_{x\in\varphi} [\mu_{\mathrm{S}}(B|x)+\mu_{\mathrm{T}}(B|x)] P_{k-1}(\mathrm{d}\varphi) + \mu_{\mathrm{B}}(B), \tag{A.10}$$

which becomes, using Campbell's theorem [25, p. 271]:

$$\mu_{k|k-1}(B) = \int [\mu_{\mathrm{S}}(B|x) + \mu_{\mathrm{T}}(B|x)]\, \mu_{k-1}(\mathrm{d}x) + \mu_{\mathrm{B}}(B). \tag{A.11}$$

Note that the validity of the expression of the predicted intensity above is not restricted to specific models for the prior process Ω_{k-1}. As such, the construction of the predicted intensity is identical in the case of the PHD filter with spawning (see Mahler's original proof in [61]). Let us now focus on the explicit expression of the intensity measure μ_{S}. Since the survival process is assumed Bernoulli with parameter $p_{\mathrm{S}}(\cdot)$ and spatial distribution $f_{\mathrm{S}}(\cdot|\cdot)$, we can exploit Eq. (3.20) to retrieve the intensity μ_{S} through the expression of the p.g.fl. G_{S} given by Eq. (3.22):

$$\begin{aligned}
\mu_{\mathrm{S}}(B|\cdot) &= \delta G_{\mathrm{S}}(h|\cdot; 1_B)|_{h=1} && \text{(A.12a)}\\
&= \delta\left(1 - p_{\mathrm{S}}(\cdot) + p_{\mathrm{S}}(\cdot) \int h(x) f_{\mathrm{S}}(\mathrm{d}x|\cdot); 1_B\right)\Big|_{h=1} && \text{(A.12b)}\\
&= p_{\mathrm{S}}(\cdot) f_{\mathrm{S}}(B|\cdot). && \text{(A.12c)}
\end{aligned}$$

A.1.3 Predicted cardinality

Let us now focus on the expression of the predicted cardinality $\rho_{k|k-1}$. From Eq. (3.4) the cardinality distribution of an arbitrary point process can be retrieved through its Janossy measures; let us then compute the predicted n^{th}-order Janossy measure $J^{(n)}_{k|k-1}$ evaluated at the neighborhood of a collection of n arbitrary points $y_1, \ldots, y_n$. Using Eq. (3.19) yields

$$\begin{aligned}
J^{(n)}_{k|k-1}(\mathrm{d}(y_1, \ldots, y_n)) &= \delta^n G_{k|k-1}(h; 1_{\mathrm{d}y_1}, \ldots, 1_{\mathrm{d}y_n})\big|_{h=0}\,, && \text{(A.13)}\\
&= \delta^n (G_{k-1}(G_{\mathrm{c}}(h|\cdot)) G_{\mathrm{B}}(h); 1_{\mathrm{d}y_1}, \ldots, 1_{\mathrm{d}y_n})|_{h=0}\,. && \text{(A.14)}
\end{aligned}$$

Applying the product rule (3.14) then gives

$$J^{(n)}_{k|k-1}(\mathrm{d}(y_1, \ldots, y_n)) = \sum_{\tau\subseteq\{1,\ldots,n\}} \delta^{|\tau|}(G_{k-1}(G_{\mathrm{c}}(h|\cdot)); (1_{\mathrm{d}y_i})_{i\in\tau})\big|_{h=0} \delta^{n-|\tau|} G_{\mathrm{B}}(h; (1_{\mathrm{d}y_i})_{i\in\tau^c})\big|_{h=0}. \tag{A.15}$$

Using Eq. (3.19) on the instantaneous birth process then gives

$$J^{(n)}_{k|k-1}(\mathrm{d}(y_1,\ldots,y_n)) = \sum_{\tau\subseteq\{1,\ldots,n\}} C_{|\tau|}(\mathrm{d}(y_i)_{i\in\tau}) J^{(n-|\tau|)}_{\mathrm{B}}(\mathrm{d}(y_i)_{i\in\tau^c}), \tag{A.16}$$

where

$$C_{|\tau|}(\mathrm{d}(y_i)_{i\in\tau}) = \delta^{|\tau|}(G_{k-1}(G_{\mathrm{c}}(h|\cdot)); (1_{\mathrm{d}y_i})_{i\in\tau})\big|_{h=0}, \tag{A.17}$$

and where $J^{(n-|\tau|)}_{\mathrm{B}}$ is the $(n-|\tau|)^{th}$-order Janossy measure of the instantaneous birth process.

We shall now detail the expression of the quantity C_q, evaluated at the neighborhood of a collection of q arbitrary points $z_1,\ldots,z_q$. Applying the general chain rule (3.12) yields

$$C_q(\mathrm{d}(z_1,\ldots,z_q)) = \delta^q(G_{k-1}(G_{\mathrm{c}}(h|\cdot)); 1_{\mathrm{d}z_1},\ldots,1_{\mathrm{d}z_q})\big|_{h=0}, \tag{A.18a}$$

$$= \sum_{\pi\in\Pi_q} \delta^{|\pi|} G_{k-1}\left(G_{\mathrm{c}}(h|\cdot); \left(\delta^{|\omega|} G_{\mathrm{c}}(h|\cdot; (1_{\mathrm{d}z_i})_{i\in\omega})\right)_{\omega\in\pi}\right)\Big|_{h=0}. \tag{A.18b}$$

Developing the predicted p.g.fl. G_{k-1} through Janossy measures with Eq. (3.2) then gives

$$C_q(\mathrm{d}(z_1,\ldots,z_q)) = \sum_{\pi\in\Pi_q}\sum_{m\geq|\pi|} \frac{1}{(m-|\pi|)!}\int_{\mathbb{X}^m} \prod_{i=1}^{|\pi|} \delta^{|\omega_i|} G_{\mathrm{c}}(h|x_i; (1_{\mathrm{d}z_j})_{j\in\omega_i})\Big|_{h=0},$$
$$\times \prod_{i=|\pi|+1}^{m} G_{\mathrm{c}}(0|x_i) J^{(m)}_{k-1}(\mathrm{d}(x_1,\ldots,x_m)). \tag{A.18c}$$

Since the prior process is assumed i.i.d., we can substitute the expression given by Eq. (3.27) to the prior Janossy measures $J^{(m)}_{k-1}$ and obtain

$$C_q(\mathrm{d}(z_1,\ldots,z_q)) = \sum_{\pi\in\Pi_q}\sum_{m\geq|\pi|} \frac{m!}{(m-|\pi|)!}\rho(m) C_\pi(\mathrm{d}(z_1,\ldots,z_q)), \tag{A.18d}$$

where

$$C_\pi(\mathrm{d}(z_1,\ldots,z_q)) = \int\cdots\int \prod_{i=1}^{|\pi|} \delta^{|\omega_i|} G_{\mathrm{c}}(h|x_i; (1_{\mathrm{d}z_j})_{j\in\omega_i})\big|_{h=0} \prod_{i=|\pi|+1}^{m} G_{\mathrm{c}}(0|x_i) \prod_{i=1}^{m} s(\mathrm{d}x_i) \tag{A.19a}$$

$$= \left(\int G_{\mathrm{c}}(0|x) s(\mathrm{d}x)\right)^{m-|\pi|} \prod_{\omega\in\pi}\left(\int \delta^{|\omega|} G_{\mathrm{c}}(h|x; (1_{\mathrm{d}z_i})_{i\in\omega})\big|_{h=0} s(\mathrm{d}x)\right). \tag{A.19b}$$

Recall from Eq. (A.1) that $G_{\mathrm{c}}(h|x) = G_{\mathrm{S}}(h|x) G_{\mathrm{T}}(h|x)$; using the product rule (3.14) on Eq. (A.19b)

then yields

$$
\begin{aligned}
&C_\pi(\mathrm{d}(z_1,\ldots,z_q)) \\
&= \Big(\int G_{\mathrm{S}}(0|x) G_{\mathrm{T}}(0|x) s(\mathrm{d}x) \Big)^{m-|\pi|} \\
&\quad \times \prod_{\omega\in\pi} \Big(\int \sum_{\nu\subseteq\omega} \delta^{|\nu|} G_{\mathrm{S}}(h|x;(1_{\mathrm{d}z_i})_{i\in\nu})\Big|_{h=0} \delta^{|\omega|-|\nu|} G_{\mathrm{T}}(h|x;(1_{\mathrm{d}z_i})_{i\in\omega\setminus\nu})\Big|_{h=0} s(\mathrm{d}x) \Big).
\end{aligned} \tag{A.20}
$$

Now, from the derivation shown in Eq. (A.12), we see that:

$$
\delta^{|\nu|} G_{\mathrm{S}}(h|x;(1_{\mathrm{d}z_i})_{i\in\nu})\big|_{h=0} = \begin{cases} 1-p_{\mathrm{S}}(x), & \nu=\emptyset, \\ p_{\mathrm{S}}(x) f_{\mathrm{S}}(\mathrm{d}z_i|x), & \nu=\{i\}, \\ 0, & |\nu|>1. \end{cases} \tag{A.21}
$$

Therefore, Eq. (A.20) simplifies as follows:

$$
\begin{aligned}
&C_\pi(\mathrm{d}(z_1,\ldots,z_q)) = \Big(\int q_{\mathrm{S}}(x) G_{\mathrm{T}}(0|x) s(\mathrm{d}x) \Big)^{m-|\pi|} \\
&\times \prod_{\omega\in\pi} \Big(\int q_{\mathrm{S}}(x) \delta^{|\omega|} G_{\mathrm{T}}(h|x;(1_{\mathrm{d}z_i})_{i\in\omega})\big|_{h=0} s(\mathrm{d}x) \\
&+ \int \sum_{i\in\omega} p_{\mathrm{S}}(x) f_{\mathrm{S}}(\mathrm{d}z_i|x) \delta^{|\omega|-1} G_{\mathrm{T}}(h|x;(1_{\mathrm{d}z_j})_{j\in\omega\setminus i})\big|_{h=0} s(\mathrm{d}x) \Big),
\end{aligned} \tag{A.22a}
$$

Which becomes, using Eq. (3.19):

$$
\begin{aligned}
&C_\pi(\mathrm{d}(z_1,\ldots,z_q)) = \Big(\int q_{\mathrm{S}}(x) J_{\mathrm{T}}^{(0)}(x) s(\mathrm{d}x) \Big)^{m-|\pi|} \\
&\times \prod_{\omega\in\pi} \Big(\int q_{\mathrm{S}}(x) J_{\mathrm{T}}^{(|\omega|)}(\mathrm{d}(z_i)_{i\in\omega}|x) s(\mathrm{d}x) \\
&+ \int \sum_{i\in\omega} p_{\mathrm{S}}(x) f_{\mathrm{S}}(\mathrm{d}z_i|x) J_{\mathrm{T}}^{(|\omega|-1)}(\mathrm{d}(z_j)_{j\in\omega\setminus i}|x) s(\mathrm{d}x) \Big),
\end{aligned} \tag{A.22b}
$$

where $J_{\mathrm{T}}^{(|\omega|)}$ is the $|\omega|^{th}$-order Janossy measure of the spawning process. Exploiting Eq. (3.4), it follows from Eq. (A.22b) that

$$
\begin{aligned}
\int_{\mathbb{X}^q} C_\pi(\mathrm{d}(z_1,\ldots,z_q)) &= \Big(\int q_{\mathrm{S}}(x) \rho_{\mathrm{T}}(0|x) s(\mathrm{d}x) \Big)^{m-|\pi|} \\
&\quad \times \prod_{\omega\in\pi} |\omega|! \int [q_{\mathrm{S}}(x) \rho_{\mathrm{T}}(|\omega||x) + p_{\mathrm{S}}(x) \rho_{\mathrm{T}}(|\omega|-1|x)] s(\mathrm{d}x) \qquad \text{(A.23a)} \\
&= b_0^{m-|\pi|} \prod_{\omega\in\pi} b_{|\omega|}, \qquad \text{(A.23b)}
\end{aligned}
$$

where the coefficients b_i are defined by

$$b_i = \begin{cases} \int q_{\mathrm{S}}(x)\rho_{\mathrm{T}}(0|x)s(\mathrm{d}x), & i = 0, \\ i! \int \left[q_{\mathrm{S}}(x)\rho_{\mathrm{T}}(i|x) + p_{\mathrm{S}}(x)\rho_{\mathrm{T}}(i-1|x)\right] s(\mathrm{d}x), & i > 0. \end{cases} \tag{A.24}$$

Exploiting Eq. (A.23b), it follows from Eq. (A.18d) that

$$\int_{\mathbb{X}^q} C_q(\mathrm{d}(z_1, \ldots, z_q)) = \sum_{\pi \in \Pi_q} \sum_{m \geq |\pi|} \frac{m!}{(m-|\pi|)!} \rho(m) b_0^{m-|\pi|} \prod_{\omega \in \pi} b_{|\pi|}. \tag{A.25}$$

We may finally retrieve the scalar $\rho_{k|k-1}(n)$ through Eq. (3.4):

$$\rho_{k|k-1}(n) = \frac{1}{n!} \int_{\mathbb{X}^n} J^{(n)}_{k|k-1}(\mathrm{d}(y_1, \ldots, y_n)) \tag{A.26a}$$

$$= \sum_{\nu \subseteq \{1,\ldots,n\}} \frac{1}{(n-|\nu|)!} \int_{\mathbb{X}^{n-|\nu|}} J_{\mathrm{B}}^{(n-|\nu|)}(\mathrm{d}(y_i)_{i \in \nu^c}) \frac{(n-|\nu|)!}{n!} \int_{\mathbb{X}^{|\nu|}} C_{|\nu|}(\mathrm{d}(y_i)_{i \in \nu}) \tag{A.26b}$$

$$= \sum_{q=0}^{n} \binom{n}{q} \rho_{\mathrm{B}}(n-q) \frac{(n-q)!}{n!} \sum_{\pi \in \Pi_q} \sum_{m \geq |\pi|} \frac{m!}{(m-|\pi|)!} \rho(m) b_0^{m-|\pi|} \prod_{\omega \in \pi} b_{|\omega|} \tag{A.26c}$$

$$= \sum_{q=0}^{n} \rho_{\mathrm{B}}(n-q) \sum_{\pi \in \Pi_q} \sum_{m \geq |\pi|} \frac{m!}{q!(m-|\pi|)!} \rho(m) b_0^{m-|\pi|} \prod_{\omega \in \pi} b_{|\omega|}. \tag{A.26d}$$

Using the definition of the Bell polynomial (4.5) then yields the desired result.

A.2 Proof of corollary 1

For the sake of simplicity, the time subscripts will be omitted throughout the proof when there is no ambiguity. Also, we will denote by q_{S} and q_{D} the functions $1 - p_{\mathrm{S},k}$ and $1 - p_{\mathrm{D},k}$, respectively.

A.2.1 Predicted intensity

Let us first focus on the explicit expression of the intensity measure μ_{T} of the spawning process in Eq. (4.3), depending on the modeling choices for the spawning process.

a) Bernoulli process with parameter $p_{\mathrm{T}}(\cdot)$ and spatial distribution $s_{\mathrm{T}}(\cdot|\cdot)$:

Using the same construction as in Eq. (A.12) we have immediately

$$\mu_{\mathrm{T}}(B|\cdot) = p_{\mathrm{T}}(\cdot) s_{\mathrm{T}}(B|\cdot). \tag{A.27}$$

b) zero-inflated Poisson process with parameter $p_{\mathrm{T}}(\cdot)$, rate $\lambda_{\mathrm{T}}(\cdot)$ and spatial distribution $s_{\mathrm{T}}(\cdot|\cdot)$:

Exploiting Eq. (3.26) yields

$$\mu_{\mathrm{T}}(B|\cdot) = \delta G_{\mathrm{T}}\left(h|\cdot;1_B\right)|_{h=1} \tag{A.28a}$$

$$= \delta\left(q_{\mathrm{T}}(\cdot)+p_{\mathrm{T}}(\cdot)\exp\left[\lambda_{\mathrm{T}}(\cdot)\left(\int h(x)s_{\mathrm{T}}(\mathrm{d}x|\cdot)-1\right)\right];1_B\right)\Big|_{h=1} \tag{A.28b}$$

$$= p_{\mathrm{T}}(\cdot)\lambda_{\mathrm{T}}(\cdot)\delta\left(\int h(x)s_{\mathrm{T}}(\mathrm{d}x|\cdot)-1;1_B\right)\Big|_{h=1}\exp\underbrace{\left[\lambda_{\mathrm{T}}(\cdot)\left(\int s_{\mathrm{T}}(\mathrm{d}x|\cdot)-1\right)\right]}_{=0} \tag{A.28c}$$

$$= p_{\mathrm{T}}(\cdot)\lambda_{\mathrm{T}}(\cdot)s_{\mathrm{T}}(B|\cdot). \tag{A.28d}$$

A.2.2 Predicted cardinality

Let us now detail the expression of the coefficients b_i of the Bell polynomial in Eq. (4.6), depending on the modeling choices for the spawning process.

a) Bernoulli process with parameter $p_{\mathrm{S}}(\cdot)$ and spatial distribution $f_{\mathrm{S}}(\cdot|\cdot)$:

From the description of the Bernoulli process (3.21) it follows that

$$\rho_{\mathrm{T}}(n|x) = \begin{cases} q_{\mathrm{T}}(x), & n=0, \\ p_{\mathrm{T}}(x), & n=1, \\ 0, & \text{otherwise.} \end{cases} \tag{A.29}$$

Thus, the coefficients b_i in Eq. (4.6) become

$$b_i = \begin{cases} \int q_{\mathrm{S}}(x)q_{\mathrm{T}}(x)s(\mathrm{d}x), & i=0, \\ \int \left[p_{\mathrm{S}}(x)q_{\mathrm{T}}(x)+q_{\mathrm{S}}(x)p_{\mathrm{T}}(x)\right]s(\mathrm{d}x), & i=1, \\ 2\int p_{\mathrm{S}}(x)p_{\mathrm{T}}(x)s(\mathrm{d}x), & i=2, \\ 0, & i>2. \end{cases} \tag{A.30}$$

b) zero-inflated Poisson process with parameter $p_{\mathrm{T}}(\cdot)$, rate $\lambda_{\mathrm{T}}(\cdot)$, and spatial distribution $s_{\mathrm{T}}(\cdot|\cdot)$:

From the description of the zero-inflated Poisson process (3.25) it follows that

$$\rho_{\mathrm{T}}(n|x) = \begin{cases} q_{\mathrm{T}}(x) + p_{\mathrm{T}}(x)e^{-\lambda_{\mathrm{T}}(x)}, & n = 0, \\ p_{\mathrm{T}}(x)e^{-\lambda_{\mathrm{T}}(x)}\frac{\lambda_{\mathrm{T}}(x)^n}{n!}, & \text{otherwise.} \end{cases} \tag{A.31}$$

Thus, the coefficients b_i in Eq. (4.6) become

$$b_i = \begin{cases} \int q_{\mathrm{S}}(x)\left[q_{\mathrm{T}}(x) + p_{\mathrm{T}}(x)e^{-\lambda_{\mathrm{T}}(x)}\right] s(\mathrm{d}x), & i = 0, \\ \int \left[q_{\mathrm{S}}(x)p_{\mathrm{T}}(x)e^{-\lambda_{\mathrm{T}}(x)}\lambda_{\mathrm{T}}(x) + p_{\mathrm{S}}(x)\left[q_{\mathrm{T}}(x) + p_{\mathrm{T}}(x)e^{-\lambda_{\mathrm{T}}(x)}\right]\right] s(\mathrm{d}x), & i = 1, \\ \int p_{\mathrm{T}}(x)\lambda_{\mathrm{T}}^{i-1}(x)e^{-\lambda_{\mathrm{T}}(x)}\left[q_{\mathrm{S}}(x)\lambda_{\mathrm{T}}(x) + ip_{\mathrm{S}}(x)\right] s(\mathrm{d}x), & i \geq 2. \end{cases} \tag{A.32}$$

A.3 Comparison of expressions [59, (A.15)] and Eq. (4.4)

The construction of the predicted cardinality in [59] relies on the derivation of p.g.f.s, describing the cardinality distribution of specific processes, through the Faà di Bruno's general chain formula for usual derivatives (more information on p.g.f.s in the context of multi-object filtering can be found in [61, 63]). The connection between the two expressions can be established through the version of Faà di Bruno's formula involving partial Bell polynomials [16, p.420], i.e.

$$\frac{\mathrm{d}^n}{\mathrm{d}\mathbf{x}^n}F\big(G(\mathbf{x})\big) = \sum_{k=0}^{n} F^{(k)}\big(G(\mathbf{x})\big)B_{n,k}\big(G^{(1)}(\mathbf{x}), \ldots, G^{(n)}(\mathbf{x})\big), \tag{A.33}$$

where F, G denote some suitable functions. Substituting Eq. (A.33) in [59, (A.9)] yields

$$\frac{1}{q!}\frac{\mathrm{d}^q}{\mathrm{d}\mathbf{x}^q}G_{k-1}(g(\mathbf{x})) = \frac{1}{q!}\sum_{j=0}^{q} G_{k-1}^{(j)}\big(g(\mathbf{x})\big)B_{q,j}\big(g^{(1)}(\mathbf{x}), \ldots, g^{(q)}(\mathbf{x})\big), \tag{A.34}$$

where

$$g(\mathbf{x}) = \int \left[q_{\mathrm{S},k}(x) + p_{\mathrm{S},k}(x)\mathbf{x}\right] G_{\mathrm{T},k}(\mathbf{x}|x)s_{k-1}(\mathrm{d}x), \tag{A.35}$$

$$g^{(i)}(\mathbf{x}) = \int \left[q_{\mathrm{S},k}(x)G_{\mathrm{T},k}^{(i)}(\mathbf{x}|x) + ip_{\mathrm{S},k}(x)G_{\mathrm{T},k}^{(i-1)}(\mathbf{x}|x)\right] s_{k-1}(\mathrm{d}x), \tag{A.36}$$

where $i \geq 1$, and where G_{k-1} and $G_{\mathrm{T},k}$ denote the p.g.f.s of the prior and spawning processes, respectively. (Note that, for the sake of simplicity, we use the same notation for the p.g.f. and

p.g.fl. of a process, though the quantities are different in nature.) Following [59, (A.7)], we then have

$$\frac{1}{n!}\frac{\mathrm{d}^n}{\mathrm{d}\mathbf{x}^n}G_{k|k-1}(\mathbf{x})\bigg|_{\mathbf{x}=0} = \sum_{q=0}^{n}\rho_{\mathrm{B},k}(n-q)\frac{1}{q!}\sum_{j=0}^{q}G_{k-1}^{(j)}\big(g(0)\big)B_{q,j}\big(g^{(1)}(0),\ldots,g^{(n)}(0)\big), \tag{A.37}$$

where $G_{k|k-1}$ denotes the p.g.f. of the predicted process. Using basic calculus properties on p.g.f.s, we have

$$G_{k-1}^{(j)}\big(g(0)\big) = \sum_{m\geq j}\frac{m!}{(m-j)!}\rho_{k-1}(m)\big(g(0)\big)^{m-j} \tag{A.38a}$$

$$= \sum_{m\geq j}\frac{m!}{(m-j)!}\rho_{k-1}(m)b_0^{m-j}. \tag{A.38b}$$

Also, from Eq. (A.36) we have

$$g^{(i)}(0) = \int\left[q_{\mathrm{S},k}(x)G_{\mathrm{T},k}^{(i)}(0|x) + ip_{\mathrm{S},k}(x)G_{\mathrm{T},k}^{(i-1)}(0|x)\right]s_{k-1}(\mathrm{d}x) \tag{A.39a}$$

$$= i!\int\left[q_{\mathrm{S},k}(x)\rho_{\mathrm{T},k}(i|x) + ip_{\mathrm{S},k}(x)\rho_{\mathrm{T},k}(i-1|x)\right]s_{k-1}(\mathrm{d}x) \tag{A.39b}$$

$$= b_i, \tag{A.39c}$$

where $i \geq 1$. Substituting Eq. (A.38b) and Eq. (A.39c) into Eq. (A.37), the predicted cardinality [59, (A.15)] then takes the form (4.4).

Appendix B

CPHD with Spawning Algorithm

Input
Posterior cardinality distribution: ρ_{k-1}
Posterior spatial distribution: s_{k-1}

Parameters
Cardinality distribution of spawning: $\rho_{\mathrm{T},k}$
Cardinality distribution of spontaneous birth: $\rho_{\gamma,k}$
Probability of survival: $p_{\mathrm{S},k}$

Initialisation
for $n \in [0\ n_{\max}]$ **do**
  Predicted card. dist.: $\rho_{k|k-1}(n) := 0$
  Mult. coefficient: $c_n := \sum_{m=n}^{n_{\max}} \frac{m!}{(m-n)!} \rho_{k-1}(m) b_0^{m-n}$
  Bell polynomial: $B_{n,0} := [n == 0]$
  Bell coefficient: $b_n :=$ (see Eq. (4.6))
  Bell sum: $B_n := B_{n,0} c_0$
end for

Prediction
for $n \in [0\ n_{\max}]$ **do**
  for $m \in [1\ n]$ **do**
    Bell pol.: $B_{n,m} := \sum_{q=m-1}^{n-1} \binom{n-1}{n-1-q} b_{n-q} B_{q,m-1}$
    Bell sum: $B_n \mathrel{+}= B_{n,m} c_m$
  end for
  Bell sum: $B_n \times= \frac{1}{n!}$
  for $m \in [0\ n_{\max} - n]$ **do**
    Predicted card. dist.: $\rho_{k|k-1}(n+m) \mathrel{+}= \rho_{\gamma,k}(m) B_n$
  end for
end for

Output
Predicted cardinality distribution: $\rho_{k|k-1}$

Appendix C

GLMB with Spawning Algorithm

Algorithm 1: Joint Prediction and Update with Spawning

- input: $\{(I^{(h)}, \xi^{(h)}, w^{(h)}, p^{(h)})\}_{h=1}^{H}$, Z_+, $H_+^{\max}$,
- input: $\{(r_{\mathrm{B},+}^{(\ell)}, p_{\mathrm{B},+}^{(\ell)})\}_{\ell \in \mathbb{B}_+}$, p_{S}, $f_{\mathrm{S},+}(\cdot|\cdot)$,
- input: p_{T}, $f_{\mathrm{T},+}(\cdot|\cdot)$, $\mathbb{T}_+(\cdot), \kappa_+$, $p_{\mathrm{D},+}$, $g_+(\cdot|\cdot)$
- output: $\{(I^{(h_+)}, \xi^{(h_+)}, w^{(h_+)}, p^{(h_+)})\}_{h_+=1}^{H_+}$

---

sample counts $[T_+^{(h)}]_{h=1}^{H}$ from a multinomial distribution with parameters $H_+^{\max}$ trials and weights $[w^{(h)}]_{h=1}^{H}$,
for $h = 1:H$
  generate $\mathbb{T}_+(I^{(h)}) = \{\{(\ell, k+1)\} \times \{1:M_\ell\} : \ell \in I^{(h)}\}$
  initialize $\gamma^{(h,1)}$
  compute $\eta^{(h)} = [\eta_i^{(h)}(j)]_{(i,j)=(1,-1)}^{(|\mathbb{B}_+ \cup I^{(h)} \cup \mathbb{T}_+(I^{(h)})|, |Z_+|)}$ using (6.68)
  $\{\gamma^{(h,t)}\}_{t=1}^{\tilde{T}_+^{(h)}} := \mathrm{Unique}(\mathrm{Gibbs}(\gamma^{(h,1)}, T_+^{(h)}, \eta^{(h)}))$
  for $t = 1:\tilde{T}_+^{(h)}$
    convert $\gamma^{(h,t)}$ to $(I_+^{(h,t)}, \theta_+^{(h,t)})$ using (6.67)
    compute $\bar{w}_+^{(h,t)}$ from $w^{(h)}$ and $I_+^{(h,t)}$ using (6.69)
    compute and normalize $p_+^{(h,t)}$ using (6.70)
    $\xi_+^{(h,t)} = (\xi^{(h)}, \theta_+^{(h,t)})$
  end
end
compute $\hat{C}$ given in (6.71)
compute $w_+^{(h,t)} = \bar{w}_+^{(h,t)} / \hat{C}$
$\{(I_+^{(h_+)}, \xi_+^{(h_+)}, w_+^{(h_+)}, p_+^{(h_+)})\}_{h_+=1}^{H_+} := \mathrm{Aggregate}(\{(I_+^{(h,t)}, \xi_+^{(h,t)}, w_+^{(h,t)}, p_+^{(h,t)})\}_{h,t=1,1}^{H,\tilde{T}_+^{(h)}})$

Algorithm 2: Aggregate

From the bottom portion of [92, Algorithm 2] replicated here for convenience, though with the addition of $\xi_+^{(h,t)}$ terms

- input:$\{(I_+^{(h,t)}, \xi_+^{(h,t)}, w_+^{(h,t)}, p_+^{(h,t)})\}_{h,t=1,1}^{H,\tilde{T}_+^{(h)}}$
- output: $\{(I_+^{(h_+)}, \xi_+^{(h_+)}, w_+^{(h_+)}, p_+^{(h_+)})\}_{h_+=1}^{H_+}$

1: $(\{(I_+^{(h_+)}, \xi_+^{(h_+)}, p_+^{(h_+)})\}_{h_+=1}^{H_+}, \sim, [U_{h,t}])$
$:=$ Unique$(\{(I_+^{(h,t)}, \xi_+^{(h,t)}, p_+^{(h,t)})\}_{h,t=1,1}^{H,\tilde{T}_+^{(h)}})$
2: for $h_+ = 1 : H_+$
3: $w_+^{(h_+)} = \sum_{h,t:U_{h,t}=h_+} w_+^{(h,t)}$
4: end
5: normalize weights $\{w_+^{(h_+)}\}_{h_+=1}^{H_+}$