Spacecraft Relative Navigation Using Random Finite Sets

A THESIS

SUBMITTED TO THE FACULTY OF THE

UNIVERSITY OF MINNESOTA

BY

LAUREN SCHLENKER

in partial fulfillment of the requirements

for the degree of Master of Science

DR. RICHARD LINARES

MAY 2019

Acknowledgements

First and foremost, I owe so much to the professors and faculty of the Aerospace Engineering and Mechanics department for being top notch. Seeing the quality of the teaching and research by the professors in the department gave me the courage to add Aerospace Engineering as my second major as an undergraduate student, and ultimately to pursue it further in graduate school. The faculty in the department have given me more help than I had any right to when it came to working with my overly ambitious undergraduate and post-graduate plans.

Second, I owe a lot to my adviser, Dr. Richard Linares. Deciding that I wasn't actually going to be a physicist was hard enough, much less deciding what area of engineering I fit into best. His passion for his research inspired me when I needed it most and helped me find where I fit in.

Thanks to my group of physics buddies for being there for me in undergrad while I struggled to sort out my priorities, and eventually encouraging me to pursue the things I find interesting. Thanks to my lifting buddies, old and new, for helping me become so much stronger mentally and physically, and for supporting me so enthusiastically. Thanks to my mother and sister for their consistent support, always keeping me grounded and reminding me how far I've come when things were more difficult than they had to be, even when you each had your own much more important struggles.

I've finally made it to where I wanted to be, and it wouldn't have happened without the help of everyone in the Navigation and Mission Design Branch at NASA Goddard Space Flight Center. Thanks for believing in me and helping me get here, even when I thought there was no chance. In particular, very special thanks to David Gaylor for supporting me the most every step of the way.

Thanks to Bryce Doerr and Jordan Larson for reading my drafts as I wrote them, even though they really didn't have to. There's a lot to be said for having smart people double check that you're not being completely stupid.

Finally of course, endless thanks to Eric Konitzer for being so many things to me for nearly a decade: an outstanding friend, boyfriend, future husband, engineer, and overall good human being. I very much would not have made it through school without your undying support and ability to help me pick myself up whenever a challenge knocked me down a little too far. I'm a better person and a much better engineer just by knowing you.

"I may not have gone where I intended to go, but I think I have ended up where I needed to be."

-Douglas Adams

Abstract

Spacecraft Relative Navigation Using Random Finite Sets

by Lauren SCHLENKER

Future space missions require that spacecraft have onboard capability to autonomously navigate non-cooperative environments for rendezvous and proximity operations (RPO). Current relative navigation filters can have difficulty in these situations when optical sensors are used, diverging due to complications with data association, high measurement uncertainty, and clutter, particularly when detailed *a priori* maps of the target object or spacecraft do not exist. This thesis demonstrates the feasibility of random finite set (RFS) filters for spacecraft relative navigation and pose estimation. A generalized RPO scenario is formulated as a simultaneous localization and mapping (SLAM) problem, in which an observer spacecraft seeks to simultaneously estimate the location of features on a target object or spacecraft as well as its relative position, velocity and attitude. An RFS-based filter called the Gaussian Mixture Probability Hypothesis Density (GMPHD) is used. Simulated flash LIDAR measurements are tested, using a GMPHD filter embedded in a particle filter to obtain a feature map of a target and a relative pose estimate between the target and observer over time. Results show that an RFS-based filter such as the one used can successfully perform SLAM in a spacecraft relative navigation scenario with no *a priori* map of the target, and that the formulation behind RFS-based filtering is potentially well suited to spacecraft relative navigation.

Contents

List of Tables

List of Figures

List of Algorithms

List of Abbreviations

CW	**C**lohessy-**W**iltshire
EAP	**E**xpected **A** **P**riori
EKF	**E**xtended **K**alman **F**ilter
GM-PHD	**G**aussian **M**ixture **P**robability **H**ypothesis **D**ensity
GNC	**G**uidance **N**avigation and **C**ontrol
LIDAR	**LI**ght **D**etection **A**nd **R**anging
LVR	**L**ow **V**ariance **R**esampling
MRP	**M**odified **R**odrigues **P**arameters
NASA	**N**ational **A**eronautics and **S**pace **A**dministration
PHD	**P**robability **H**ypothesis **D**ensity
RBGMPHD	**R**ao-**B**lackwellized **G**aussian **M**ixture **P**robability **H**ypothesis **D**ensity
RFS	**R**andom **F**inite **S**et
RPO	**R**endezvous and **P**roximity **O**perations
SC	**S**ingle **C**luster
SIFT	**S**cale **I**nvariant **F**eature **T**ransform
SLAM	**S**imultaneous **L**ocalization **A**nd **M**apping
SURF	**S**peeded-**U**p **R**obust **F**eatures

List of Symbols

a	semi-major axis of target orbit
B_k	set of targets birthed at time k
$\mathbf{C}$	optical center of the camera
f	focal length of the camera
$f_{k\|k-1}(\cdot\|\cdot)$	transition density
F	state transition matrix
$g_k(\cdot\|\cdot)$	probability density
h	linearized measurement model
H	Jacobian matrix
I_{3x3}	3x3 identity matrix
$J_{\cdot,k}$	number of Gaussian Mixture elements in a particular intensity function at time k
k	time index
K_k	set of clutter measurements at time k
m	mean of a distribution
m_u	horizontal inverse pixel size
m_v	vertical inverse pixel size
$M(k)$	number of elements in measurement space at time k
n	mean motion of an orbit
n_{col}	number of columns on focal plane
n_{row}	number of rows on focal plane
$n_{clutter}$	number of clutter measurements
$N(k)$	number of elements in state space at time k
N_{eff}	number of effective particles
N_p	number of particles
$p_{D,k}$	probability of detection at time k
$p_{S,k}$	probability of survival at time k
$p_{k\|k-1}(\cdot\|\cdot)$	posterior density prediction
$p_k(\cdot\|\cdot)$	updated posterior density
$\mathbf{p}$	point location in image space
P	covariance of a distribution
P_u	horizontal location of optical center
P_v	vertical location of optical center
$\mathbf{P}$	point location in real space
Q	process noise covariance

$\mathbf{r_{P/B}}$	position vector from frame P to frame B
R	measurement noise covariance
$\mathbf{r_{P/B}}$	position vector from frame P to frame B
$\mathbf{r_{\dot{P}/B}}$	velocity vector from frame P to frame B
s	optical skew factor
$S_{k\|k-1}$	set of surviving targets
t	time
T	feature pruning threshold
u	horizontal pixel coordinate
v	vertical pixel coordinate
$w_k^{(i)}$	weight of element i at time k
$x_{k,i}$	state i in state space at time k
X_k	set of elements in state space at time k
$z_{k,i}$	measurement i at time k
${}^{P}y_i$	measurement of feature i in the $\{P\}$ frame
$\mathbf{y}$	measurement vector
Z_k	set of elements in measurement space at time k
β_k	intensity function of newly birthed targets at time k
$\delta(\cdot\|\cdot)$	Dirac-delta function
$\eta^{[l]}$	weight of particle l
$\gamma_{k\|k-1}$	intensity function of newly spawned targets at time k
$\Gamma_{k\|k-1}$	set of newly spawned targets at time k
κ_k	clutter intensity function
Θ_k	set of real measurements at time k
$\nu(\cdot)$	intensity function
μ	Gravitational parameter of primary governing body
μ_s	reference measure
ω	angular velocity vector
ϕ	linearized state transition function
ρ	range
σ	variance
σ (bolded)	vector of Modified Rodrigues Parameters
ξ	placeholder state
ζ	predicted measurement
$\hat{x}$	predicted variable x
$\cup$	set union
$\mathcal{N}(\cdot;\cdot,\cdot)$	normal distribution
$\sum_i^j$	sum from i to j
$\{\cdot\}$	frame definition
$[BA]$	rotation from frame A to frame B
$\dot{x}$	time derivative of variable x

x^*	post resampled variable x

1 Introduction

1.1 Motivation

Technology that supports relative navigation for autonomous spacecraft is in need of improvement and innovation in order for space exploration and servicing missions to be technically and economically feasible in the near future. According to the NASA 2017 Strategic Technology Investment Plan, one critical potential technological advancement is the ability for autonomous robotic systems to enhance the gathering of scientific data as well as perform complex functions that are essential to missions that must operate independently from Earth ground systems [1]. The ability for spacecraft to autonomously explore increasingly distant, unknown, or hostile environments without direct human guidance will extend the reach of humanity far beyond current capabilities and is a major goal for NASA in the coming decade.

Examples of the direct need for this technology include on-board terrain mapping as well as general Guidance, Navigation, and Control (GNC) algorithms to alleviate the reliance on humans for missions that may be too distant for instant communication. This development in systems-level autonomy for these functions will directly enable missions to distant bodies such as moons, asteroids, comets, and one day a mission to Mars [2]. Autonomous relative navigation solutions also directly benefit missions which rely on precise rendezvous and docking capabilities such as the upcoming satellite servicing program Restore-L [3]. These future goals will only be made feasible if costs and risk can be reduced by decreasing the reliance on a human ground crew for critical navigation tasks.

1.2 Current State of the Art

Despite the clear need for innovation in the area of spacecraft relative navigation to achieve systems-level autonomy, much of the current research is focused on increasingly heuristic methods that address challenges with human-guided solutions rather than finding solutions which avoid these issues. Additionally, many of these methods are too computationally complex to be feasible on-board a spacecraft or require extensive guidance from humans on the ground. In this section, we will explore these current approaches and address why they are insufficient for the needs of future missions.

1.2.1 Simultaneous Localization and Mapping in Space

In general, relative navigation in space can be thought of as being made up of three important steps: sensing the environment, determining the spacecraft location relative to the environment from the sensor measurements, and deciding an action once the contents of the environment and the spacecraft location in the environment have been determined. The focus of this thesis is primarily in the second step, i.e. we assume that sensor measurements have been given, and we must determine what they mean in order for further action to be taken. Thus, we shall formulate this aspect of relative navigation as a Simultaneous Localization and Mapping (SLAM) problem, in which a robotic agent (e.g. the spacecraft) seeks to localize itself in its environment at the same time as mapping the details of the environment it is in. The goal is that the iterative improvement of the spacecraft's relative pose (position and attitude in the environment) will drive improvement in the estimate of features in the environment and vice versa. With this definition, the concept of an environment is completely general and can be a spacecraft localizing itself relative to another spacecraft or group of spacecraft, or in orbit around a small body such as an asteroid or comet.

There are many different formulations of SLAM that can be used depending on the context of the environment and several are widely employed with success [4, 5]. For example, autonomous driving applications lend themselves well to methods which create detailed 3D topological maps of the environment in order to identify obstacles in the path of the vehicle for planning purposes. For space applications, a less detailed approach is necessary due to limited on-board computational power as well as generally limited sensing and communication capabilities. For this reason, we will focus on feature-based SLAM methods, which tend to require computationally cheaper algorithms and can accommodate relatively simpler sensor schemes [5, 6].

When SLAM is formulated with feature-based maps, the mapping portion of the problem can be thought of as a multi-target tracking problem. Then, the features in the environment are the targets being tracked in order to create a model of the environment, and the pose of the observer in the environment is simultaneously being estimated. Many estimation methods in general are formulated as single-target tracking problems and heuristic methods must be employed to alter them to be suitable for multi-target tracking. The challenges introduced by this are typically non-trivial, and a poor solution can break an algorithm. For example, allowing the concept of multiple measurements and multiple targets automatically introduces the need for data association, i.e. a method must be used to determine which measurements must have originated from which targets before a navigation filter can be used. With a relatively low number of targets and measurements, heuristic methods can be successful in performing data association. However, as the dimension of the problem increases, these methods may no longer be computationally feasible, particularly on-board a spacecraft [7]. Despite this, data association for multi-target tracking is a crucially important problem, and navigation filters are likely to diverge entirely if this step is performed incorrectly [4, 6].

Another crucial aspect of the problem which must be considered is the fact that optical

navigation sensors can be difficult to use reliably in space environments, and an appropriate SLAM formulation must be able to work with potentially uncertain information effectively. Not only can sensor measurements in space be fairly noisy, they can also be corrupted with false detections. For example, when observing fiducials on a spacecraft, a sensor may detect more targets than are truly in view due to lighting artifacts and the context of the situation. This is particularly a problem when it comes to operation around small bodies such as comets, where it is expected that the environment will be filled with non-target objects that the sensor will inevitably detect. We shall refer to these extraneous measurements as clutter, and assume that they are in general indistinguishable from "true" measurements. Conversely if lighting conditions are poor, a sensor may fail to detect the targets that are expected, giving a navigation filter very little to work with for continued estimation.

Finally, many optical-based relative navigation methods rely on *a priori* knowledge of the target or of the spacecraft pose in order to compare an estimate to a model to refine the estimate [8–12]. Though these methods are certainly successful in situations in which this information is available, they are nearly unusable in situations where a spacecraft must navigate around something entirely new or is uncertain about its pose with respect to the target. This is particularly true when it comes to navigation around non-cooperative spacecraft or approach to and rendezvous with previously unexplored small bodies which have yet to be mapped.

In order for a SLAM method for spacecraft relative navigation to be successful, it must be able to correctly process measurements and produce estimates regardless of these challenges. Several studies in the past have addressed SLAM methods for spacecraft relative navigation purposes. Work by Augenstein as well as Sonnenburg et. al. has demonstrated that the problem can indeed be adequately handled by a SLAM formulation [13, 14]. However as expected, both studies required significant computation for feature management

and data association prior to the navigation filter; moreover, the formulations have no in-built concept of realistic measurement situations such as features passing in and out of the sensor field of view, (i.e. clutter and missed detections,) despite this being a critical aspect of a successful SLAM formulation.

One of the most popular general-use feature-based SLAM implementation is FastSLAM, a SLAM algorithm in which the feature map is estimated by an Extended Kalman Filter and the trajectory and pose of the observer is simultaneously estimated using a particle filter [15]. FastSLAM has enjoyed much success because of its lack of assumptions on the pose probability distribution, as well as the achievement of real time implementations, which is a crucial feature for limited computation on-board a spacecraft [16]. Though this is arguably the most successful approach in the literature, as well as most closely aligning with the goals of this thesis, the problem of data association must still be solved before a method such as FastSLAM can be performed. As previously stated, this can be a major drawback. A study by Cocaud and Kubota uses an algorithm based on an improved particle filtering method, intended for navigation and pinpoint landing on small bodies. [17] Their results do address issues with drift in the particle filter that is commonly seen in FastSLAM, however their underlying approach is still heavily reliant on heuristic methods for data association of features that are observed frame to frame.

Although solutions from these past studies exist for SLAM-based multi-target tracking for spacecraft relative navigation, it is clear that a reliable and efficient method which does not rely on heuristic methods for data association and map management has yet to be achieved.

1.2.2 Random Finite Sets for SLAM

Rather than developing heuristic methods for solving the problems inherent to feature-based SLAM in space, it may be preferable to avoid some of the issues entirely. Since it is well established that the problem of needing to associate measurements to known

locations in space makes up a significant portion of the computational load of most SLAM algorithms, it makes sense to attempt to avoid this portion of the problem in particular. Fortunately, a new family of filters has recently emerged that is formulated to specifically deal with many of the challenges introduced by multi-target tracking. The solution to these challenges are directly incorporated into the mathematics of the filter, negating the need for any intermediate heuristics for data association or map management. These filters, which are based off of Random Finite Set (RFS) statistics, use a mathematical framework based off of sets rather than the random vector framework commonly associated with traditional Kalman filtering methods. As shall be shown throughout this thesis, this is a powerful way of formulating a navigation filter for these purposes because it allows for a more natural mathematical description of the kinds of realistic dynamic and measurement scenarios that occur in multi-target tracking problems.

The first RFS filter was proposed for multi-target tracking by Ronald Mahler in the early 2000's – he called it the Probability Hypothesis Density (PHD) filter [18]. Since this initial PHD filter, several variants of other RFS filters have been used in studies that address realistic aspects of implementing the PHD filter for ground based and underwater robotics [19]. It is not surprising that the RFS based PHD filter has seen success in these areas – studies by Mullane, et. al. have demonstrated that a feature-based map used for SLAM is fundamentally a finite set, and is thus more easily represented and manipulated with RFS methods [6]. Their most significant results show that even in the presence of uncertain dynamics and measurements and high degrees of clutter, the PHD filter can significantly outperform EKF-based mapping filters, as well as FastSLAM.

Several recent studies have also leveraged the ability of RFS-based tracking methods for other space-based multi-target tracking applications. Work by McCabe and DeMars has used RFS-based filters in feature-based robotic mapping for planetary landing, successfully mapping the features of a lunar environment during a descent trajectory given a known spacecraft pose [20]. Their work compares the multi-target tracking performance

of several RFS-based filters for this purpose. Studies have also performed and evaluated the use of RFS-based filters in the realm of space object tracking, resulting in an integrated framework for successful identification and tracking of space objects [21, 22]. These past works have laid the groundwork for showing that RFS-based methods such as the PHD filter are a promising approach for fundamentally handling the challenges associated with multi-target tracking in space.

1.3 Thesis Contribution

Despite the success that RFS-based methods have had for SLAM in other navigation applications as well as other space-based applications, very little work has been done on applying the techniques to the challenges presented by spacecraft relative navigation. To date, the sole example is a conference paper by the author of this thesis [23]. Thus, the scope of this thesis is to expand upon the reasoning and background done in that initial paper, and further demonstrate the feasibility of RFS-based Bayesian methods for spacecraft relative navigation purposes. The situation to be studied is a generalized rendezvous and proximity operations (RPO) scenario with measurement and dynamics models based on existing research platforms.

The outline of this thesis is as follows: Chapter 3 discusses the specific scenario to be studied, including the measurement model, dynamics, and simplifying assumptions used to make the problem feasible. Chapter 2 discusses the theory behind the RFS-based SLAM methods which shall be applied to the problem. Chapter 4 details the exact implementation of the theory developed in Chapter 2. Chapter 5 then presents and analyzes the results of the algorithm for several different test cases. Finally, Chapter 6 summarizes the outcomes of these studies and gives suggestions for further research into the area of RFS-SLAM for spacecraft relative navigation.

2 Theory

2.1 Random Finite Sets for Multi-Target Tracking

In multi-target tracking, we wish to jointly estimate the number of targets which are present in state space, as well as their states. We are given a set of noisy, potentially cluttered measurements, meaning that our sensors may be measuring artifacts that we do not wish to track. Assume that at time step k, there exists a set of $N(k)$ targets and $M(k)$ measurements. In general there is no specific order in which it is known that these measurements and states are associated, thus we represent them as random finite sets given by:

$$X_k = \{x_{k,1}, ..., x_{k,N(k)}\} \tag{2.1}$$

$$Z_k = \{z_{k,1}, ..., z_{k,M(k)}\} \tag{2.2}$$

which remain identical regardless of the order in which elements are presented. In the RFS framework, the sets X_k and Z_k are a multi-target state and multi-target observation, respectively. Somewhat analogous to random vectors used for single-target tracking, X_k and Z_k are random finite sets, variables that can be characterized by a probability distribution and a family of joint probability densities of all the elements of X_k and Z_k [24]. Additionally, the cardinality $|\cdot|$, or number of elements in a random finite set becomes a variable that can be estimated, and does not have to remain the same over time.

This formulation allows more generalized possibilities for the time evolution of the states contained in X_k. For a given multi-target state X_{k-1}, each element $x_{k-1} \in X_{k-1}$

either continues to exist at the next time step k with probability $p_{S,k}$ or ceases to exist with probability $1 - p_{S,k}$. Alternatively, a target which does not exist at time $k-1$ may begin to exist at time step k independently of the targets existing at time step $k-1$. We say that this target has been birthed into state space. Similarly, a new target may begin to exist at time step k by being "spawned" from a target which existed at time step $k-1$. In real physical scenarios, both these options may occur because a target is entering the region of state space of interest after previously being outside it, or a target that was initially estimated to be one target becomes better resolved and is actually more likely to be several targets that are very closely spaced. These possibilities can likely account for nearly every imaginable physical scenario that may be encountered in multi-target tracking.

Thus at a time step k, a general target state is represented as

$$X_k = \left[\bigcup_{\zeta \in X_{k-1}} S_{k|k-1}(\zeta) \right] \cup \left[\bigcup_{\zeta \in X_{k-1}} \Gamma_{k|k-1}(\zeta) \right] \cup B_k \tag{2.3}$$

where X_k is composed of a union of a set of surviving targets $S_{k|k-1}$ each with transition density $f_{k|k-1}(x_k|x_{k-1})$, a set of newly birthed targets B_k, and a set of spawned targets $\Gamma_{k|k-1}$. Note that Γ and B are entirely general and their forms are determined by the specific scenario in which they are used.

Similarly, the RFS measurement model is fundamentally able to take into account probabilities of detection and the possibility of clutter in the environment. A target $x_k \in X_k$ can either be detected with probability $p_{D,k}$, or not detected (missed) with probability $1 - p_{D,k}$. Additionally, the framework of RFS allows for the concept of clutter measurements, which we shall represent as an additional RFS K_k of false detections which do not originate from a target.

Thus, our set of measurements at a time step k is denoted by

$$Z_k = K_k \cup \left[\bigcup_{x \in X_k} \Theta_k(x) \right] \tag{2.4}$$

where Z_k is composed of a union of sets of actual measurements $\Theta_k(x)$ which occur with probability density $g_k(z_k|x_k)$ and clutter measurements K_k. K_k is also entirely general and determined by the specific scenario involved.

The goal of multi-target filtering is to obtain a posterior density of the multi-target state X_k given a multi-target observation Z_k. The posterior density can be calculated using a Bayesian recursion given by

$$p_{k|k-1}(X_k|Z_{1:k-1}) = \int f_{k|k-1}(X_k|X)p_{k-1}(X|Z_{1:k-1})\mu_s(dX) \tag{2.5}$$

$$p_k(X_k|Z_{1:k}) = \frac{g_k(Z_k|X_k)p_{k|k-1}(X_k|Z_{1:k-1})}{\int g_k(Z_k|X)p_{k|k-1}(X|Z_{1:k-1})\mu_s(dX)} \tag{2.6}$$

where $f_{k|k-1}(\cdot|\cdot)$ is a transition density from one state to another, $g_k(\cdot|\cdot)$ is the observation likelihood of a measurement given a state; μ_s is an appropriate reference measure on the collection of all finite subsets of state space [25]. Unfortunately as with most Bayesian recursions, this recursion is computationally intractable due to the set integrals required.

2.2 The Probability Hypothesis Density Filter

In order to approximate the multi-target posterior density in equations 2.5 and 2.6, we will choose to instead propagate the posterior intensity, which is simply the first order statistical moment of the multi-target state. This simplification forms the basis of the Probability Hypothesis Density (PHD) filter [18]. A key result of this choice is that the integral of the intensity function over a particular region of state space is equal to the number of targets expected to exist in that region of state space. This means that the number of expected target at each time step can be propagated jointly. In other words, we find that $\hat{N}(k) = \int v(x)dx$, where the intensity function $v(x)$ is referred to as the probability hypothesis density and is integrated over the whole RFS space. This allows for estimation of

the cardinality of the multi-target state set.

From here, a number of additional assumptions are made in order to obtain the full tractable PHD recursion from equations 2.5-2.6. At this point in the discussion, the rigorous derivations and proof are detailed by Vo and Ma in [25] and will not be fully reproduced here as they do not necessarily add to the fundamental understanding needed to work with the PHD filter. Important assumptions and key steps will however be shown. These assumptions are summarized below:

- Each target evolves and generates observations independently of one another.
- Clutter is Poisson distributed, and independent of measurements which originate from targets.
- The predicted multi-target RFS governed by $p_{k|k-1}$ is Poisson distributed.
- Survival and detection probabilities are state independent.

Clearly, these assumptions come with varying degrees of congruence to the relevant problem – regardless, these are common assumptions to make. With these assumptions in place, the full PHD recursion is then as follows:

$$v_{k|k-1}(x) = \int p_{S,k}(\zeta) f_{k|k-1}(x|\zeta) v_{k-1}(\zeta) d\zeta + \int \gamma_{k|k-1}(x|\zeta) v_{k-1}(\zeta) d\zeta + \beta_k(x) \tag{2.7}$$

$$v_k(x) = [1 - p_{D,k}(x)] v_{k|k-1}(x) + \sum_{z \in z_k} \frac{p_{D,k}(x) g_k(z|x) v_{k|k-1}(x)}{\kappa_k(z) + \int p_{D,k}(\xi) g_k(z|\xi) v_{k|k-1}(\xi) d\xi} \tag{2.8}$$

where $\gamma_{k|k-1}$ is the intensity function of the RFS of spawned states $\Gamma_{k|k-1}$, β_k is the intensity function of the birth RFS B_k, and κ_k is the intensity function of the clutter RFS K_k. Additionally, the probability of detection $p_{D,k}$ has now been directly incorporated into the update equation.

It is worth noting that the PHD recursion in equations 2.7 and 2.8 indirectly performs a simple form of data association between measurements and targets, negating the need for

expensive combinatorial computations in post-processing. Unfortunately, no closed-form solution exists for the PHD recursion [18, 19, 25].

2.2.1 Gaussian Mixture PHD Filter

In order to use the PHD recursion in a tractable, closed form manner, several further assumptions must be made:

- Each target follows a linear Gaussian dynamics model.
- Each sensor follows a linear Gaussian measurement model.
- The intensities of the birth and spawn sets are modeled as Gaussian mixtures.

Mathematically, this means that we assume each individual target follows a similar model as is typically seen with a classic Kalman filter:

$$f_{k|k-1}(x|\zeta) = \mathcal{N}(x; F_{k-1}\zeta, Q_{k-1}) \tag{2.9}$$

$$g_k z|x = \mathcal{N}(z; H_k x, R_k) \tag{2.10}$$

where in general $\mathcal{N}(\cdot; m, P)$ represents a normal Gaussian distribution with mean m and covariance P; Q_{k-1} is process noise covariance on the dynamics, F_{k-1} is the state transition matrix, H_k is the measurement model observation matrix, and R_k is the covariance of the measurement noise.

Additionally, the Gaussian mixture birth and spawn intensities become:

$$\beta_k(x) = \sum_{i=1}^{J_{\beta,k}} w_{\beta,k}^{(i)} \mathcal{N}(x; m_{\beta,k}^{(i)}, P_{\beta,k}^{(i)}) \tag{2.11}$$

$$\gamma_{k|k-1}(x|\zeta) = \sum_{j=1}^{J_{\gamma,k}} w_{\gamma,k}^{(j)} \mathcal{N}(x; F_{\gamma,k}^{(i)}\zeta + d_{\gamma,k-1}^{(j)}, Q_{\gamma,k}^{(j)}) \tag{2.12}$$

where J, w, P are given model parameters that determine the shape of the corresponding intensities and can be chosen depending on the scenario, allowing for the inclusion of *a priori* knowledge of the targets if desired or available.

Finally, we can reasonably assume that the posterior intensity at $k-1$ is also a Gaussian mixture, which continues to be a Gaussian mixture at the next time step:

$$v_{k-1}(x) = \sum_{i=1}^{J_{k-1}} w_{k-1}^{(i)} \mathcal{N}(x; m_{k-1}^{(i)}, P_{k-1}^{(i)}) \tag{2.13}$$

The end result of the use of these assumptions is the following computationally tractable recursion, where the predicted intensity at time k is also a Gaussian mixture consisting of the sum of surviving, spawned, and birthed targets:

$$v_{k|k-1}(x) = p_{S,k} \sum_{j=1}^{J_{k-1}} w_{k-1}^{(j)} \mathcal{N}(x; m_{S,k|k-1}^{(j)}, P_{S,k|k-1}^{(j)}) + v_{\gamma,k|k-1}(x) + \beta_k(x) \tag{2.14}$$

$$\begin{aligned} m_{S,k|k-1}^{(j)} &= F_{k-1} m_{k-1}^{(j)} \\ P_{S,k|k-1}^{(j)} &= Q_{k-1} + F_{k-1} P_{k-1}^{(j)} F_{k-1}^T \end{aligned}$$

where the intensity of spawned targets depends on the set of previously existing targets:

$$v_{\gamma,k|k-1}(x) = \sum_{j=1}^{J_k-1} \sum_{\ell}^{J_{\beta,k}} w_{k-1}^{(j)} w_{\gamma,k}^{(\ell)} \mathcal{N}(x; m_{\gamma,k|k-1}^{(j,\ell)}, P_{\gamma,k|k-1}^{(j,\ell)}) \tag{2.15}$$

$$\begin{aligned} m_{\gamma,k|k-1}^{(j,\ell)} &= F_{\gamma,k-1}^{(\ell)} m_{k-1}^{(j)} + d_{\gamma,k-1}^{(\ell)} \\ P_{\gamma,k|k-1}^{(j,\ell)} &= Q_{\gamma,k-1}^{(\ell)} + F_{\gamma,k-1}^{(\ell)} P_{\gamma,k-1}^{(j)} (F_{\gamma,k-1}^{(\ell)})^T \end{aligned}$$

Then, the measurement updated posterior is also a Gaussian mixture:

$$v_k(x) = (1 - p_{D,k}) v_{k|k-1}(x) + \sum_{z \in Z_k} v_{D,k}(x; z) \tag{2.16}$$

where the intensity of detected targets is:

$$v_{D,k}(x;z) = \sum_{j=1}^{J_{k|k-1}} w_k^{(j)}(z)\mathcal{N}(x; m_{k|k}^{(j)}(z), P_{k|k}^{(j)}) \tag{2.17}$$

$$\begin{aligned} m_{k|k}^{(j)}(z) &= m_{k|k-1}^{(j)} + K_k^{(j)}(z - H_k m_{k|k-1}^{(j)}) \\ P_{k|k}^{(j)} &= [I - K_k^{(j)} H_k] P_{k|k-1}^{(j)} \\ K_k^{(j)} &= P_{k|k-1}^{(j)} H_k^T (H_k P_{k|k-1}^{(j)} H_k^T + R_k)^{-1} \end{aligned}$$

and where the weight of each element of the Gaussian Mixture is given by:

$$\begin{aligned} w_k^{(j)}(z) &= \frac{p_{D,k} w_{k|k-1}^{(j)} q_k^{(j)}(z)}{\kappa_k(z) + p_{D,k} \sum_{\ell=1}^{J_{k|k-1}} w_{k|k-1}^{(\ell)} q_k^{(\ell)}(z)} \\ q_k^{(j)}(z) &= \mathcal{N}(z; H_k m_{k|k-1}^{(j)}, R_k + H_k P_{k|k-1}^{(j)} H_k^T) \end{aligned}$$

Thus, equations 2.14-2.17 combined give a tractable closed-form solution for the Gaussian Mixture Probability Hypothesis Density (GM-PHD) filter. This recursion takes in noisy, cluttered measurements and extracts an intensity function which represents the location of likely target positions in state space. The form presented here is visually and theoretically similar to the Kalman filter, but it is more general due to the nuances introduced by the RFS formulation.

2.2.2 Extended Kalman Filter Approximations

Similar to how the Kalman filter can be modified to incorporate nonlinear dynamics and measurement models, Vo and Ma have shown that the PHD filter can also use nonlinear dynamics and measurements by locally linearizing the state transition matrix and observation matrix. Assuming dynamics and measurement models are given in the form:

$$x_k = \phi_k(x_{k-1}, \nu_{k-1}) \tag{2.18}$$

$$z_k = h_k(x_k, \epsilon_k) \tag{2.19}$$

where ϕ_k and h_k are known nonlinear dynamics and measurement models respectively, allowing for zero-mean Gaussian process noise ν_k and measurement noise ϵ_k, with covariances Q_{k-1} and R_k respectively. Additionally, the state transition matrix can be calculated as:

$$F_{k-1}^{(j)} = \left.\frac{\partial \phi_k(x_{k-1}, 0)}{\partial x_{k-1}}\right|_{x_{k-1}=m_{k-1}^{(j)}} \tag{2.20}$$

Similarly, the observation model derivative matrix can be calculated as:

$$H_k^{(j)} = \left.\frac{\partial h_k(x_k, 0)}{\partial x_k}\right|_{x_k=m_{k|k-1}^{(j)}} \tag{2.21}$$

These approximations for the state transition and observation model derivative matrices are used for this work, as the problem defined in Chapter 3 utilizes highly nonlinear dynamics and measurement models.

2.3 Particle Filters

The GM-PHD filter in equations 2.14-2.17 is used primarily to perform the mapping duties of SLAM, similar to the approach outlined by Vo and Ma [25]. Using the context of a SLAM application, the targets tracked by the filter are thus the features which have been identified and extracted from images of the target spacecraft taken by the observer. Thus in order to perform the localization aspect of SLAM, a method must be used to obtain a relative pose estimate of the observer based on the feature map multi-target state provided by the PHD filter. Work by Mullane et. al. has successfully achieved this by using an outer particle filter loop [6]. To fully perform SLAM with an RFS-based filter, we also employ

the use of a particle filter (also commonly referred to as Sequential Monte Carlo filter) to recursively obtain a maximum *a posteriori* pose estimate.

In this formulation, each particle of the particle filter represents an individual estimate of the pose of the observer as well as a corresponding estimate of the map. The map comes from the PHD filter which each particle utilizes independently, thus the feature map belonging to each particle is conditioned on an independent estimate of the pose.

2.3.1 Importance Weighting

For this method to succeed, there needs to be a way to determine which particle has the "best" joint estimate of feature map and observer pose. To achieve this, an importance weight $p_l(Z_k|X_k^{[l]}, Z_{1:k-1})$ must be calculated for each l_{th} particle. In other words, a weight can be strategically assigned to each particle based on how well the observations in Z_k match with the map conditioned pose associated with each particle. The basis of importance weighting begins by assuming that a posterior density of states x and measurements z can generally be approximated with N_p particles as

$$p(x_k|z_{1:k}) \approx \sum_{l=1}^{N_p} \eta_k^{[l]} \delta(x_k|x_k^l) \tag{2.22}$$

where the weight $\eta_k^{[l]}$ from Baye's Rule of the l^{th} particle is calculated recursively as

$$\eta_k^{[l]} \propto \eta_{k-1}^{[l]} \frac{p(z_k|x_k^{[l]}) p(x_k^{[l]}|x_{k-1}^{[l]})}{\pi(x_k|x_{k-1}, z_k)} \tag{2.23}$$

This approximation of the posterior approaches the true posterior density as $N_p \to \infty$. Thus, the weights of the particles in equation 2.23 must be calculated in order to sequentially update the weight of a particle based on newly obtained measurements at each time step. As usual, further assumptions must be made in order to make this a tractable computation. Many different methods of calculating this quantity for performing particle filter

importance weighting have been developed, each with varying levels of applicability and computation required [26]. For this work, Single Cluster (SC) importance weighting is used.

Mullane et. al. have shown that a SC importance weighting method works demonstrably better than an empty strategy or single-feature based method, with a modest increase in computation required compared to these simpler methods [27]. Additionally, they show that relative to additional methods such as multi-feature importance weighting strategies, SC importance weighting is lower complexity with only a slight decrease in filter performance.

The single cluster approximation leverages the fact that SLAM can be thought of as a hierarchical cluster process, in which a daughter process is dependent on a parent process – when the cardinality of the parent process is one, the process is referred to as "single cluster." Lee argues that SLAM is fundamentally a single cluster process, where the daughter process of the map is conditioned on the parent process of the vehicle pose. Thus by assuming the measurements obtained are a result of Poisson point processes, a single cluster update for the PHD filter can be obtained. This argument and subsequent derivation of how it affects the calculation of the multi-target likelihood is thoroughly laid out in [28] and will not be reproduced here as it would require reproducing nearly the whole scope of the work.

Using these arguments, the right hand side of equation 2.23 becomes computable and the updated weight $\eta_k^{[l]}$ of the l^{th} particle can be calculated from the previous map PHD $\nu_k^{-[l]}$ as:

$$\eta_k^{[l]} = \exp\left(\sum_{r=1}^{N(k)^{-[l]}} w_k^{r,[l]}\right) \times \prod_{z\in Z_k}\left(\kappa(z) + p_D \sum_{r=1}^{N(k)^{-[l]}} p\left(z|\mathcal{N}_k^{r,[l]}, x_{0:k}^{[l]}\right) w_k^{r,[l]}\right) \eta_{k-1}^{[l]} \tag{2.24}$$

where $\mathcal{N}_k^{r,[l]}$ is the r^{th} Gaussian in the intensity $\nu_k^{-[l]}$ of targets in the PHD filter associated

with particle l. This is a computationally tractable form of equation 2.23, and it represents a multi-target likelihood, or how likely the measurements “agree" with each particle’s pose-conditioned feature map. Moreover, this equation is relatively simple compared to methods which would require further data association for computation of the multi-target likelihood.

2.3.2 Resampling

In addition to importance weighting, a particle filter requires some amount of resampling computation to maintain particle hypothesis in regions of the posterior that have higher probability. The primary reasons for this are to prevent degenerate particle estimates with unreasonably high variance and to improve the exploration of state space, enabling more efficient filtering. Choosing a resampling method requires several considerations: a choice of distribution from which to resample, a strategy for deciding which particles must be resampled and how, and a decision for how often to resample the distribution of particles. Many solutions to the decisions required for resampling exist, ranging from simple to complex, and are well surveyed [29].

For the purpose of this work, Low Variance Resampling (LVR) is chosen as the primary resampling technique because it is relatively simple yet rigorously systematic and thus lends itself well to applications in which the “best" sampling distribution isn’t necessarily known or may be very complicated and difficult to resolve. Moreover, LVR schemes are of complexity $\mathcal{O}(N_p)$, as compared to naive approaches which are typically of complexity $\mathcal{O}(N_p \log N_p)$. In LVR, particle hypotheses have a certain probability of being deleted and reset to be more similar to higher weighted particles, based on their original weight [30].

A qualitative summary of the properties of LVR is as follows:

- The probability that a particle gets resampled is directly proportional to its weight.
- If resampling does not take place, the particle maintains the same weight.

- Particles may be “roughened" after being resampled, meaning that their properties are varied slightly from the particles from which they were resampled in an attempt to maintain particle diversity and efficient exploration of state space. The magnitude of roughening which occurs is determined by the user.

For this work, particles are resampled at a given time step only if the “effective" number of particles, defined as $\sum \eta_i^2$, falls below a chosen threshold which is tunable by the user. Resampling too often increases the risk that particles do not adequately explore state space, but resampling too infrequently is inefficient and risks wasting particles in regions of low probability, thus the user may tune the threshold such that particles are resampled at an appropriate rate. Additionally, the weights of all particles are normalized prior to resampling, i.e. the sum of all particles weights is set to equal 1. This is common practice in particle filtering.

A complete workflow of how these properties are implemented in the full code of the filter design is shown in Chapter 4.

2.3.3 Pose Estimation

The overall relative pose estimate at each time step is updated by extracting the pose of the particle with the highest weight prior to resampling, i.e. the maximum *a posteriori* estimate. Alternatively, a weighted average pose estimate could be obtained using all available particles depending on how well distributed the particle hypotheses tend to be.

With the inclusion of a particle filter for pose estimation the overall method employed in this thesis closely resembles that of FastSLAM, where the EKF portion for mapping has been replaced with a GM-PHD filter. A more thorough discussion on the implementation and interplay between the GM-PHD and particle filters is given in Chapter 4.

3 Spacecraft Relative Navigation Problem Setup

The methods used in this thesis have been developed to be completely general and transferable to any definable spacecraft relative navigation problem, as long as appropriate dynamics and measurement models can be specified. This allows for flexibility in future applications. For the purposes of narrowing the scope of this thesis, a spacecraft rendezvous/proximity operations (RPO) scenario is discussed in order to obtain usable dynamics and measurements models, but the methods and discussion remain largely unchanged if the specific scenario being studied were to change. This section details how the dynamics and measurement models for a spacecraft RPO scenario have been developed and specified.

3.1 Discussion of Simplifying Assumptions

In an effort to be completely transparent and reproducible in regards to the method in which dynamics and measurements are being simulated, the following is a list of assumptions that are being made in order to simplify the models used for this work. In general, assumptions were made in order to focus more on the design and implementation of the filter, rather than spending time developing a fully robust or high fidelity simulation platform, as such platforms already exist. The intention is that once the filter in this work has been developed, it can be implemented in one such platform for further testing.

- The measurement frame (i.e. the camera frame) is the same as the observer body fixed frame in order to avoid including another rotation from an external measurement frame to the observer body frame.

- Modified Rodrigues Parameters are used to represent rotations, as they have just one easily avoidable singularity at infinity. This is dealt with using a simple switching procedure once the norm of the MRP vector becomes larger than a chosen threshold, and only affects the simulation of the measurements and not the filtering as switched vs. non-switched MRPs contain the same attitude representation.

- A spherical target was chosen to negate the need for complicated occlusion computations. This does not detract from the algorithms to be presented in later sections; as long as a measurement model is specified, the logic on whether or not a feature is visible at any point in time does not matter. The motion of the spherical target still includes the general dynamics of how features will rotate in and out of view of the observer.

- Clutter measurements are drawn from a uniform distribution in the field of view of the sensor independently at each time step. In other words, the clutter measurements being used are non-persistent. Persistent clutter could be used instead without loss of generality in the algorithms used, and it has been demonstrated by past studies. [27]

- The dynamics presented are in their simplest form, and no perturbations due to non-spherical gravity or solar radiation pressure are added. If these terms were added to the dynamics simulation, this would not be prohibitive as the filter to be presented is easily modifiable to include process noise terms.

- The target spacecraft is assumed to be nadir pointing in order to constrain its attitude dynamics in the Hill frame. Once again, this is not necessary to the functioning of

the filter. More complicated attitude motion could be simulated and the filter to be presented could handle it, as long as the dynamics are able to be specified in a similar way as has been done above.

- Linear Gaussian noise on the measurements has been assumed for simplicity and as common practice. However, it is not necessarily the case that feature extraction algorithms such as SIFT and SURF would produce measurement noise with precisely Gaussian measurement statistics. This would be an area of potential research in the future if a front-end feature extractor were to be used.

- The target spacecraft is assumed to be located in the center of the image frame of the observer spacecraft. This is a reasonable assumption given that modern control systems could easily achieve this kind of pointing. However once again as long as another kind of pointing model could be specified, the filter to be used in this work would remain unchanged.

3.2 Test Case Scenario

Figure 3.1 shows a general spacecraft RPO scenario, including relevant coordinate frames and defining vectors. The observer spacecraft (also commonly referred to as the chaser or deputy spacecraft) is collecting measurements of features which are located on a target spacecraft (also commonly referred to as the chief spacecraft.) The features which are being observed are physical details which show up as "optically significant" when images are collected, such as edges or corners of a solar panel, or craters and rocks on an asteroid. These definitions are used throughout this thesis.

We define a body-fixed frame $\{B\}$ centered on the target, which shall be defined the same as for the relative orbital dynamics, the Hill frame, used in following sections. Similarly, we define a primary body-fixed frame $\{P\}$ centered on the observer. This is the frame

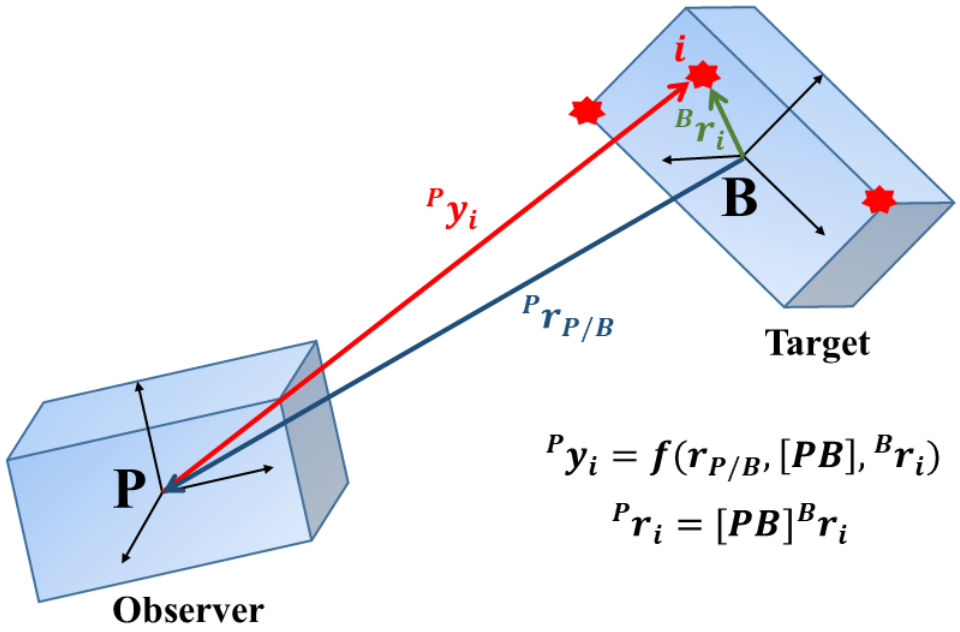

FIGURE 3.1: Relationship between observer and target coordinate frames and vector definitions.

in which measurements are to be taken. The optically significant features located on the target are defined in the rigid body frame $\{B\}$ – in this way, the ith feature located on the target can be located by a vector from the origin of the $\{B\}$ frame to the feature.

Each measurement of a feature is collected relative to the $\{P\}$ frame fixed to the observer. A measurement of feature i is defined as $\mathbf{y}_i$ and is related to the relative position of the $\{P\}$ frame with respect to the $\{B\}$ frame, as well as the relative orientation of the $\{P\}$ frame with respect to the $\{B\}$ frame, represented by the notation [PB]. These relationships are defined in Section 3.3.

The state we shall propagate using the system dynamics is defined as $\mathbf{x} = [\mathbf{r_{P/B}}, \dot{\mathbf{r}}_{\mathbf{P/B}}, \sigma]^T$,

where $\mathbf{r_{P/B}}$ denotes the relative location of {P} with respect to {B}, $\dot{\mathbf{r}}_{\mathbf{P/B}}$ is the relative velocity of {P} with respect to {B}, and σ contains the Modified Rodrigues Parameters (MRPs) used to describe the rotation [PB] between any vector defined in $\{P\}$ to be represented as a vector in $\{B\}$.

It is traditional to formulate navigation with respect to the target. Thus, the outcome of the filter developed in this thesis is an estimate of the position, velocity, and orientation of the target relative to the observer represented by $\mathbf{r_{P/B}}$, $\dot{\mathbf{r}}_{\mathbf{P/B}}$ and a rotation matrix [BP] = $[PB]^T$ which is calculated from the MRPs contained in σ.

3.2.1 Target Properties

For initial simulations the target is assumed to be spherical and covered with randomly distributed features. The target has a radius of 50 meters and has 20 randomly distributed features on its surface. This size was chosen to be a middle-ground value of size between a typical satellite or a very small asteroid. The features are chosen from a uniform distribution constrained to the surface of the sphere – however, the filter used in this thesis is not made directly aware of this constraint.

Note that a spherical shape was chosen to significantly simplify the problem of determining whether or not a simulated feature is visible to the sensor. This is achieved by a simple dot product between the measurement and relative position vectors – if the dot product is positive, the feature is then known to be visible to the sensor. Though this test would also be approximately true for any general body, it is not true in general if the body has additional physical protuberances such as solar panels. Solving the occlusion problem for a general simulated target shape and measurement model is a non-trivial feat, and is thus considered outside of the scope of this thesis.

3.3 System Dynamics

The target is assumed to be in a circular orbit about the Earth, and the observer is operating in close proximity to the target, i.e. within less than 1km range. The dynamics of the observer relative to the target can then be closely approximated by the linear Clohessy-Wiltshire (CW) equations. [31] Thus in the target body-fixed Hill frame $\{B\}$, the elements of $\mathbf{r_{P/B}} = [x \quad y \quad z]$ are defined by the following dynamics:

$$\begin{aligned} \ddot{x} &= 3n^2x + 2n\dot{y} \\ \ddot{y} &= -2n\dot{x} \\ \ddot{z} &= -n^2z \end{aligned} \tag{3.1}$$

where n is the target's mean motion defined as $n = \sqrt{\mu/a^3}$, where μ is the central planet's gravitational parameter and a is the semi-major axis of the target's orbit.

To simplify the dynamic equations, the motion is constrained such that the attitude of the target is fixed in the Hill frame, similar to a nadir pointing spacecraft. We also assume that the observer is perfectly tracking the target in its field of view, which is a reasonable assumption given an appropriate control system for pointing. In other words, the center of the target is assumed to be located in the center of the images taken by the observer. This negates the need to estimate angular velocity, since the angular velocity of the observer body frame with respect to the Hill body frame can then be directly calculated as:

$$\boldsymbol{\omega} = \frac{\mathbf{r} \times \dot{\mathbf{r}}}{|\mathbf{r}|^2} \tag{3.2}$$

The relative attitude of the observer and target are expressed using MRPs as they only have one easily avoidable singularity, compared to multiple singularities common with using a Euler angle or difficulties with normalization for a quaternion description of the

attitude dynamics. The MRPs obey the following kinematic differential equation [32]:

$$\dot{\sigma} = \frac{1}{4}\left[(1-|\sigma|^2)[I_{3\times 3}] + 2[\tilde{\sigma}] + \sigma\sigma^T\right]\omega \tag{3.3}$$

where $[\tilde{\sigma}]$ is the skew symmetric matrix composed of the elements of σ. Thus, equations 3.1 and 3.3 fully describe the system dynamics needed for basic simulation.

Though the CW equations are linear and have an analytical solution, for this work they are integrated numerically with the MRPs, which are nonlinear and cannot be analytically integrated, using Matlab's ode45 integrator to generate the truth trajectory and to propagate trajectories within the filter in order to easily increase the fidelity of the dynamics model in future studies.

3.4 Measurement Model

In the scenarios studied in this thesis, the observer spacecraft collects flash LIDAR measurements of the target spacecraft. These measurements are corrupted with Gaussian-distributed noise after being simulated from the truth trajectory. Additionally, extraneous measurements which do not originate from features on the target are appended to the noisy measurements which did originate from features on the target. Although the algorithms used in this thesis are fully capable of handling missed detections, this was not simulated.

In order to properly narrow the scope of this thesis to the implementation of the navigation filter, it is assumed that the optically significant features have been obtained and processed from a front end feature extraction algorithm such as Scale-Invariant Feature Transform (SIFT) or Speeded-Up Robust Features (SURF), which would provide these feature locations from image data [33, 34] or LIDAR data. [35] Thus, we define that a measurement of an extracted feature consists of angular position in pixel coordinates on an image as well as an associated range for that pixel position. It is important to note that

the measurements have no other identifying information, i.e. the filter doesn't receive any information about which feature a particular measurement originated from.

The flash LIDAR measurements are simulated from the simulated feature locations using a pinhole projection camera model for the pixel coordinates. This model is depicted in Figure 3.2. The range measurement of a feature is determined by the Euclidean distance from the optical center **C** of the camera to the feature located on the target at point **P**. The camera frame is defined such that the u and v axes are aligned with the rows and columns of the detector and the third axis points along the boresight of the optics. The pixel coordinates of the feature image **p** are defined in the camera frame as $\{u, v, w\}$. The measurements of a feature are thus defined as $\mathbf{y} = [u \quad v \quad \rho]$, where ρ is defined as:

$$\rho = \|\mathbf{X} - \mathbf{X}_c\| \tag{3.4}$$

where $\mathbf{X}$ is the feature location and $\mathbf{X}_c$ is the camera location in the same frame, in this case the Hill frame previously defined as {B}.

To calculate u and v a pinhole camera projection is defined with the following standard equations:

$$\mathbf{x} = w \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} wu \\ wv \\ w \end{bmatrix} \tag{3.5}$$

where $\mathbf{x}$ is calculated as:

$$\mathbf{x} = \begin{bmatrix} fm_u & s & P_u \\ 0 & fm_v & P_v \\ 0 & 0 & 1 \end{bmatrix} [R][[I_{3\times 3}, -\mathbf{X}_c] \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} \tag{3.6}$$

Here, $[R]$ is the rotation matrix from the Hill frame to the camera frame, f is the focal

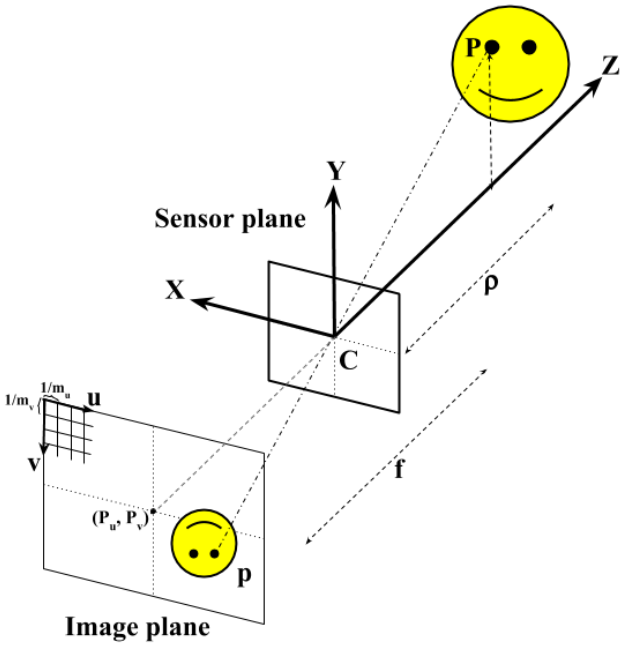

FIGURE 3.2: Depiction of a simplified pinhole camera model.

length of the detector or the distance between the sensor plane and image plane, s can be used to account for any skew in the optics if desired, P_u and P_v are the location in pixels of the optical center of the camera projected onto the image plane, and m_u and m_v are the inverse sizes of the pixels in each dimension. [36] These relationships fully dictate the base measurement model used to generate non-corrupted measurements.

Noise and $n_{clutter}$ uniformly distributed clutter measurements are added after truth measurements are calculated. The noise is generated with a zero-mean Gaussian distribution with covariance $[\sigma_u^2, \sigma_v^2, \sigma_\rho^2]$. If a measurement with noise is outside the bounds of the camera, then it is disregarded prior to the filter being run. The constants used to simulate these data are included in Table 3.1. These values were chosen to produce an angle of view of approximately 14°, and to closely mimic the specifications of the flash LIDAR on-board Raven, a module on the International Space Station which performs autonomous tracking of rendezvous targets. [37].

TABLE 3.1: Measurement model parameters.

Constant	Value	Units
f	25	mm
P_u, P_v	128	[]
s	0	[]
m_u, m_v	20480	m^{-1}
n_{col}, n_{row}	256	[]
σ_u, σ_v	1	[]
σ_ρ	10	m
$n_{clutter}$	10	[]

An example of the simulated measurements for a chosen test case is shown in Figure 3.3. At each time step, 10 clutter measurements are appended to the true measurements. The plot on the left shows the cluttered, noisy measurements given to the filter, while the plot on the right is of the non-corrupted measurements corresponding to the true location of visible features on the target body.

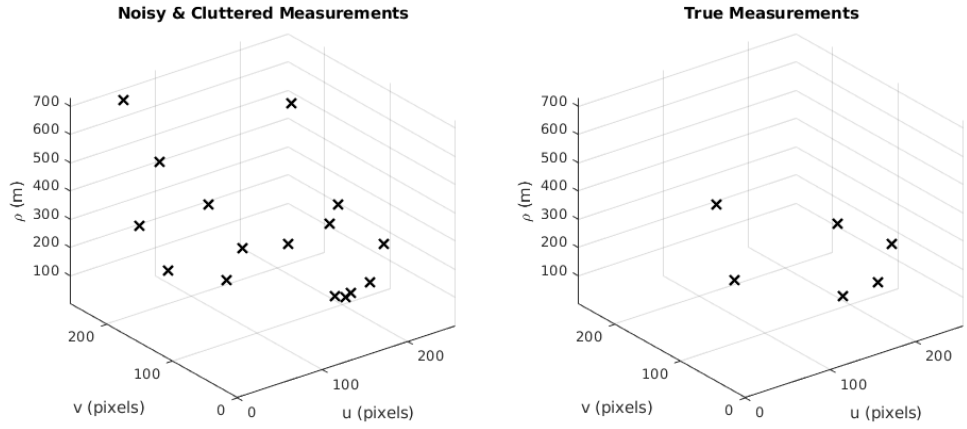

FIGURE 3.3: A simulated example of features extracted from flash LIDAR observations.

4 Algorithm Setup

The full filter used to perform SLAM for this thesis is composed of two nested filters: an inner PHD filter which performs mapping and an outer particle filter which performs localization based on the map estimation of the interior PHD filter. This chapter explicitly discusses the implementation of these filters, clarifying where the inputs go and where the outputs come from for each.

4.1 Feature Birth

As seen in Algorithm 1, the full filter is run one time step at a time. At the beginning of each time step, new features are birthed from the current set of measurements Z_k. When a feature is birthed, it is initialized with a pre-selected weight and covariance, specified by the variables "birthWeight" and "birthCovariance" respectively. These variables are tunable by the user – for our purposes we select a constant birth covariance, and the birth weight is sampled from a birth distribution specified by the user. However, these could potentially be set as a function of the measurement or pose if *a priori* information is available. The position of the feature is specified by m_{birth}, and is a function of the measurement and pose associated with the particle at that time step. This position is calculated from the user specified function "invSensorModel", which is based on the measurement model described in Chapter 2, where the equations are inverted such that the function returns a location in physical space given a measurement and a pose as inputs.

Algorithm 1 RB-GMPHD Filter

for $k = startTime : endTime$ **do**
  for $ii = 1 : numParticles$ **do** ▷ Birth new features into the map
    for $j = 1 : |Z_k|$ **do**
      $m_{birth,j}^{[ii]} = \text{invSensorModel}(PoseState_{k-1}^{[ii]}, Z_k(j))$;
      $P_{birth,j}^{[ii]} = \text{birthCovariance}$;
      $w_{birth,j}^{[ii]} = \text{birthWeight}$; ▷ Normal distribution or uniform distribution
    end for
  end for
  for $ii = 1 : numParticles$ **do**
    RB-GMPHD Prediction ▷ Propagate existing and birthed targets
    RB-GMPHD Update ▷ Update based on measurements
    Importance Weighting ▷ Reweight particles
  end for
  Resampling
  $PoseEstimate = \frac{1}{\sum \eta_k^{[ii]}} \sum(\eta_k^{[ii]} \cdot PoseState^{[ii]})$ ▷ Get the EAP pose estimate
  for $ii = 1 : numParticles$ **do** ▷ Prune out low-weight map features
    for $j = 1 : J_k$ **do**
      if $w_k(j)^{[ii]} <= T$ **then** ▷ If weight is below threshold, delete feature
        $w_k^{[ii]}(j) = []$;
        $m_k^{[ii]}(j) = []$;
        $P_k^{[ii]}(j) = []$;
      end if
    end for
  end for
end for

4.2 Prediction

The prediction step of the filter is fundamentally similar to the familiar EKF prediction step, using the dynamics models outlined in Chapter 3. This step is shown in Algorithm 2. One key difference is that the weights which exist from the previous time step are augmented by the probability of survival at that time step, $p_{S,k}$. Additionally, newly birthed features are appended to the sets of features which exist from the previous time step. Then, the means and covariances of the full set of features get propagated forward in time. The

state transition matrix F is specified by the user based on the dynamics as defined in Chapter 3, as well as the linearization in Equation 2.20. The measurement covariance, R, and process noise covariance, Q, are also specified by the user. Note that the $[ii]$ superscripts indicating that a parameter belongs to a specific particle have been omitted for ease of reading after a few steps, as it is understood that this algorithm is performed individually for each particle.

Algorithm 2 RB-GMPHD Prediction

$w_{k-1}^{[ii]} = p_{S,k} w_k^{[ii]}$ ▷ Move the time index forward for existing features
$m_{k-1}^{[ii]} = m_k^{[ii]}$
$P_{k-1}^{[ii]} = P_k^{[ii]}$
$w_{k-1}^{[ii]} = [w_{k-1}^{[ii]}, w_{birth,j}^{[ii]}]$ ▷ Append birthed features to existing features
$m_{k-1}^{[ii]} = [m_{k-1}^{[ii]}, m_{birth,j}^{[ii]}]$
$P_{k-1}^{[ii]} = [P_{k-1}^{[ii]}, P_{birth,j}^{[ii]}]$
for $j = 1 : J_{k|k-1}$ **do** ▷ Propagate dynamics as in Equations 3.1 and 3.3
  $m_{k|k-1}(j) = \mathrm{F} m_{k-1}(j)$
  $P_{k|k-1}(j) = \mathrm{Q} + \mathrm{F} P_{k-1}(j) \mathrm{F}^T$
end for

4.3 Update

The update step of the filter is also fundamentally similar to the familiar EKF update step, but it is augmented to be able to update the Gaussian mixture weights of each feature as well. This step is shown in Algorithm 3. In the update step, all possibilities of detection are accounted for, including the possibility that the predicted features were not get observed. In this case, the weight of the feature gets augmented by a factor of $(1 - p_D)$, or the probability that the feature does not get detected.

Additionally, all possibilities of each of the $J_{k|k-1}$ features being associated with each

measurement in Z_k is considered, because no information is given about which measurement originated from which feature. For each feature, a predicted measurement ξ is calculated using the measurement model, which is outlined in Chapter 3 and depends on the pose of the current particle. The measurement Jacobian H is calculated in a similar way, using the linearization given in Equation 2.21. Then, the Kalman gain K is computed using the covariances and Jacobian associated with the feature. These quantities are used for each pairing of measurement and feature to compute the weight, mean, and covariance updates given in Equation 2.17. The clutter intensity κ is a tuning parameter that is specified by the user. Once again, the $[ii]$ superscripts indicating that a parameter belongs to a specific particle have been omitted for ease of reading, as it is understood that this algorithm is performed individually for each particle.

4.4 Importance Weighting

After the prediction and update steps, each particle contains a feature map estimate which is conditioned on the pose used to calculate the predicted measurements in the update step. Prior to resampling, each particle needs an updated weight assigned to it in order to quantify how well the estimated feature map from the PHD update matches with the measurements used for the update. This is achieved with the single cluster weighting described in Chapter 2, an implementation of which is shown in Algorithm 4. The quantity $M_{k|k-1}$ is the "mass" of the PHD intensity, which in the case of a Gaussian mixture is simply the sum of all the feature weights.

4.5 Resampling

After all the particles have been updated and reweighted, they must be resampled in order to maintain diversity in the exploration of state space. This is implemented in Algorithm 5,

Algorithm 3 RB-GMPHD Update

for $j = 1 : J_{k|k-1}$ **do**
  $w_k(j) = (1 - p_D) w_{k|k-1}(j)$ ▷ Assume features were not detected
  $m_k(j) = m_{k|k-1}$
  $P_k(j) = P_{k|k-1}(j)$

end for

for $zz = 1 : |Z_k|$ **do** ▷ Assume features were detected

  $W(zz) = \sum^{J_{k|k-1}} w_{k|k-1} \mathcal{N}(Z(zz); Hm_{k|k-1}(j), R + HP_{k|k-1}H^T)$

  for $j = 1 : J_{k|k-1}$ **do** ▷ Calculate each measurement matched to each feature

    $\xi = \text{measurementModel}(PoseState_{k-1}^{[ii]}, m_{k|k-1}(j))$ ▷ From Chapter 3

    $H(j) = \text{Jacobian}(PoseState_{k-1}^{[ii]}, m_{k|k-1}(j))$ ▷ Calculate from Equation 2.19

    $K = P_{k|k-1}H(j)^T (HP_{k|k-1}H(j) + R_k)^{-1}$ ▷ Updates from Equations 2.17

    $w_k^{[jj]} = \dfrac{p_D w_{k|k-1}(j) \mathcal{N}(Z(zz); H(j)m_{k|k-1}(j), R + H(j)P_{k|k-1}H(j)^T)}{\kappa + p_D W(zz)}$

    $m_k(zz \cdot J_{k|k-1} + j) = m_{k|k-1} + K(Z(zz) - \xi)$

    $P_k(zz \cdot J_{k|k-1} + j) = [I - KH(j)]P_{k|k-1}]$

  end for

end for

Algorithm 4 Importance Weighting

$M_{k|k-1}^{[ii]} = \sum w_{k|k-1}^{[ii]}$ ▷ Mass of the PHD pre-update
$\gamma = 1; \quad \lambda = 0$
for $zz = 1 : |Z_k|$ **do**
  for $j = 1 : J_{k|k-1}^{[ii]}$ **do**
    $\lambda = \lambda + \mathcal{N}(Z(zz); \xi(zz)), (R + H(j)P_{k|k-1}(j)H(j)^T) * w_{k|k-1})(j)^{[ii]}$
  end for
  $\gamma = \gamma \cdot (\kappa + p_D * \lambda)$
end for
$\eta_k^{[ii]} = \exp(M_{k|k-1}^{[ii]}) \cdot \gamma \cdot \eta_{k-1}^{[ii]}$ ▷ Particle weight update; Equation 2.24

using the Low Variance Resampling technique discussed in Chapter 2. Note that quantities with an asterisk index * are understood to be the post-resampling quantities. Additionally,

particle weights are normalized at each time step, as is customary.

The quantity *jitterCov* is used to slightly vary the pose of the resampled particles in order to diversify the exploration of the particles at the next time step. Thus, the final resampled pose is drawn from a normal distribution centered at the pose to be selected, with covariance given by *jitterCov*. The selection of the jitter covariance is an extremely crucial aspect of the resampling process. If the covariance is too large, the particles do not adequately represent the underlying probability distribution and may begin to diverge over time. If the covariance is too small, there is no other mechanism through which the pose distribution can be explored, and the particles will be tightly clustered around a pose that may not be the best estimate.

The calculation of the jitter covariance here is computed by taking the Euclidean distance between the pre- and post- resampled pose state, scaled by a chosen factor (in this case, a factor of 0.5.) This method was chosen through a large amount of trial and error. The intuition is that if a particle is resampled such that its pose is going to change by a large amount, there's a chance that perhaps it is being resampled with a poor estimate of the pose. In order to offset this possibility, the covariance should be larger to allow the particle to have more of a chance to have a pose that is similar to the pre-resampled pose it had before. Conversely, if the pose difference is smaller, chances are the particles are converging on a good estimate. In other words, this method of choosing a jitter covariance allows particles to not be too permanently influenced by other particles which may be outliers, while still allowing for some exploration of the underlying distribution.

4.6 Pose Estimate

After the particles have been resampled, an estimate of the vehicle pose can be obtained. This is shown after the **Resampling** step in Algorithm 1. In this case, the expected *a posteriori* (EAP) or weighted average of all the hypothesized pose states is used as the output

Algorithm 5 Resampling

for $ii = 1 : numParticles$ **do**

$\alpha = \sum_{ii} \eta_k^{[ii]}$

$\eta_k^{[ii]} = \frac{1}{\alpha} \eta_k^{[ii]}$ ▷ Normalize the particle weights prior to resampling

end for

$N_{eff} = \frac{1}{\sum_i (\eta_{k_i}^2)}$ ▷ Calculate the number of effective particles

if $N_eff \leq N_a$ **then** ▷ If the number of effective particles is low, resample

$r = rand(); \quad jj = 1; \quad ii = 1; \quad \beta = \eta_k^{[1]};$

for $m = 1 : numParticles$ **do**

$U = r + (m-1)/numParticles$

while $U \geq \beta$ **do** ▷ Find the particle ii which has higher weight than jj

$ii = ii + 1$

$\beta = \beta + \eta_k^{[ii]}$

end while

$w_k^{*[jj]} = w_k^{[ii]}$ ▷ Set all the map properties of particle jj to those of particle ii

$m_k^{*[jj]} = m_k^{[ii]}$

$P_k^{*[jj]} = P_k^{[ii]}$

$jitterCov = 0.5(PoseState^{[ii]} - PoseState^{[jj]})^2$

$PoseState^{*[jj]} \sim \mathcal{N}(PoseState^{[ii]}, jitterCov)$ ▷ Resample the particle's pose

$jj = jj + 1$

end for

else

Don't resample. Move on to next step.

end if=0

estimate of the filter. Note that this estimate is not directly fed back into the filter at any point, as the filter operates based on the individual estimates of each particle. This estimate is however taken as the "answer" from the filter. This method assumes that the pose probability distribution is not highly non-normal, and that the weighted average is thus a reasonable representation. If this was found to not be the case, the maximum *a posteriori* estimate could also be used by simply extracting the pose of the particle with the highest weight at each time step.

4.7 Feature Pruning

At time k, the Gaussian mixture produced by the recursion has $\mathcal{O}(J_{k-1}|Z_k|)$ terms, and this number is able to increase without bound. A good approximation can still be obtained by pruning out elements of the Gaussian mixture whose weight falls below a certain tunable threshold. A simple method for performing this pruning is given by Vo and Ma, and the same method is used for this thesis [25]. This step is simple but crucial to achieving good filter performance, and is performed for each particle after resampling occurs, shown at the bottom of Algorithm 1.

In this procedure, a feature is simply discarded if the weight of its Gaussian mixture element is below a certain scalar threshold, T, after the update. This threshold is a crucial tuning parameter, and must be carefully chosen for each case. If T is chosen to be too high, no features will be chosen as "feasible" and the feature map will be empty. If T is chosen to be too low, many poor features will be chosen as feasible, and will not adequately represent the relationship between the measurements and the dynamics of the features from the previous time step. In general, T must be chosen by eye. However, a good initial estimate of an adequate value for T can be found by observing the typical values of the elements in w_k prior to pruning – if the filter is working well there is a clear distinction of features which have much higher weights than other features. A cutoff value can be chosen based on this, and iterated more tightly in order to produce desired results.

In general for this work, a conservative value of T was used, i.e. the value used was lower than may be optimal for computational performance. This typically overestimates the number of features and in theory helps the filter maintain a populated map estimate, even if it is not a perfectly accurate one, in the event that measurements become very sparse for a range of time steps. Though it was observed that the filter does in general recover the pose and map estimate after a period of sparse measurements, the robustness of that has not been analyzed sufficiently to justify a more tight pruning threshold.

A better logical heuristic could in the future be used in order to avoid this conservative thresholding. For example, the filter could be told to not update the feature map at all if the number of measurements at a certain time step is below a certain amount, e.g. the expected number of clutter measurements, if that quantity is known. For now, biasing the pruning threshold and allowing more features in the map seems to accomplish the same result with only a slight observed increase in computation required. The result of this is that a few "clutter" features sometimes exist at each time step and are not associated with an actual physical feature, however in general these additional "false" features do not survive pruning for more than one time step as additional measurements do not support their existence and their weights get decremented accordingly.

Note also that the pruning step is not performed prior to importance weighting and particle resampling. From a theoretical standpoint, even the low weight elements of the Gaussian mixture are a solution and thus must contribute to the weighting of the particle. The low weight features are pruned purely for computational purposes, and not as a means towards improving the estimates of the filter. Using feature pruning as a potentially rigorous way of improving the filter estimates has not been investigated in this thesis, nor has it been investigated by other studies.

4.8 Full Filter

Figure 4.1 graphically depicts the relationships between the GMPHD filter and the particle filter in a flow chart.

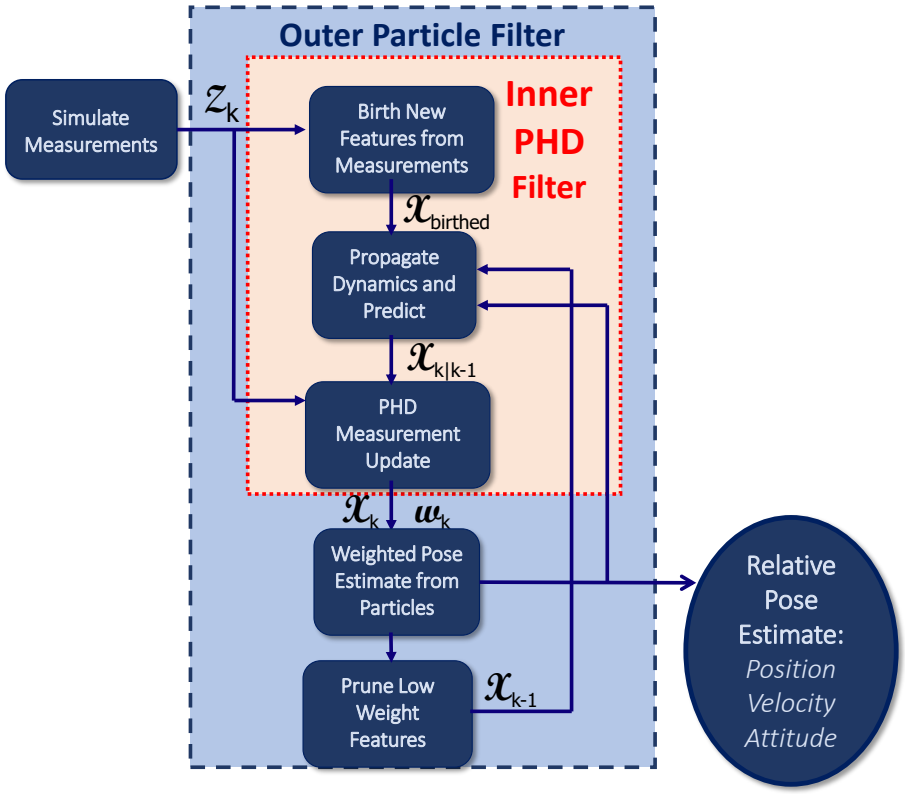

FIGURE 4.1: Simplified diagram depicting the flow of data between different parts of the filter.

5 Results

The primary purpose of this thesis is to demonstrate that an RFS-based filter can successfully perform SLAM for relative spaceflight purposes. The approach to achieve this, as discussed in previous sections, involves an inner PHD filter which is used inside a particle filter to perform both the mapping and localization aspects of SLAM. Thus in order to understand if the RFS-based part of the overall filter is performing as expected, it is useful to first investigate the PHD filter independently of the particle filter. This is essentially equivalent to using the full filter described in the previous sections, but given perfect knowledge of the pose and only 1 particle. By giving the filter perfect knowledge of the pose, SLAM is not truly being performed, but a good understanding of how the PHD filter is performing the mapping aspect of SLAM can be achieved prior to moving on to a full SLAM implementation.

Unfortunately, it is not possible to only test the localization part of the full SLAM filter. The relative weights of the particles, which are determined by the disparity between the pose-conditioned map and the measurements, are what primarily drive the estimate of the pose. Thus if the map estimate is held the same across all the particles, then the particles will all have equal weight; essentially, no pose distribution exists. Therefore, it is impossible to isolate the localization aspect of the full SLAM filter. Despite this, the following section will demonstrate that the mapping portion of the filter is performing well, so further sections are justified in moving on to exploring the full SLAM filter.

The methods used for evaluating the following results are a mix of qualitative and

quantitative. The mapping results of the PHD filter are analyzed by observing how closely the estimated map matches the actual map of visible features at each time step. This is approached in terms of how many features are tracked and their general proximity to the actual visible feature locations. When pose is being estimated, this can be analyzed in the more traditional sense, as it is not a set-based quantity.

5.1 Test Case Descriptions

The measurement models and dynamics used to generate the test cases are described in Chapters 2 and 3. The different test cases are chosen by simply varying the initial conditions of the dynamics to achieve a desired orbit; features are then randomly distributed onto the target body and measurements are simulated. Two primary test cases are used to demonstrate the results of the filter. The initial conditions relative to the target body used to generate these trajectories are given in Table 5.1.

TABLE 5.1: Test case initial conditions.

Test Case	**x** (m)	**y** (m)	**z** (m)	$\dot{\mathbf{x}}$ (m/s)	$\dot{\mathbf{y}}$ (m/s)	$\dot{\mathbf{z}}$ (m/s)
1 - Periodic Orbit	-200	0	0	0	0.4241	0
2 - Walking Safety Ellipse	0	0	-500	0.1	-0.01	0.1

The orbit for the first test case is shown in Figure 5.1; the blue curve is the trajectory, the sphere is the target body, and the red x's on the target body are the locations of the features to be tracked. This is a periodic orbit around the target body, mimicking the behavior that might for example be used during an inspection phase of a servicing mission or the mapping phase of a small body exploration mission. This orbit was chosen to investigate the long term behavior of the filter, on the order of several hours. In this case, the observer periodically encounters the same features as ones it has observed in the past. Currently, no mechanism is in place to have the filter store features that have been tracked in the past; features are simply deleted from the tracked set once they become infeasible. However

in the future, this test case would be ideal for implementing a loop-closure method, i.e. storing old features and identifying if they are encountered later in the trajectory.

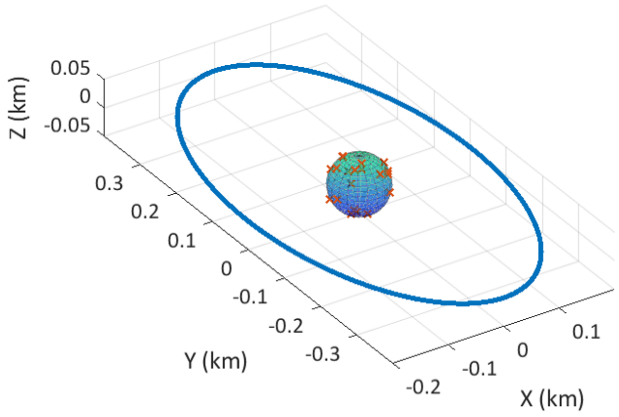

FIGURE 5.1: Trajectory for Case 1, the periodic orbit.

It is widely established in the literature that a fundamental problem with SLAM implementations is dealing with very sparse maps. Thus, a second test case has been chosen that investigates how the filter performs in the event that very few or none of the measurements originate from a true feature for a period of time, i.e. no features on the target body are visible to the sensor. The relative orbit of the observer with respect to the target for this case is shown in Figure 5.2 and is referred to as a walking safety ellipse. This type of trajectory may also be used for the inspection and rendezvous phase of a servicing mission [38].

The walking safety ellipse trajectory brings the observer to within 30m of the surface of the target and as far away as 650m. The trajectory shown is over a time period of 15,000 seconds in order to see the behavior of the orbit, though this full time span is not used for the results. As will be seen in the results, the sensor periodically sees no features on the target body for at least 1 time step, leading to a sparse feature map.

In both test cases, the birth model $\gamma_k(x)$ for the PHD filter is given to be a Gaussian

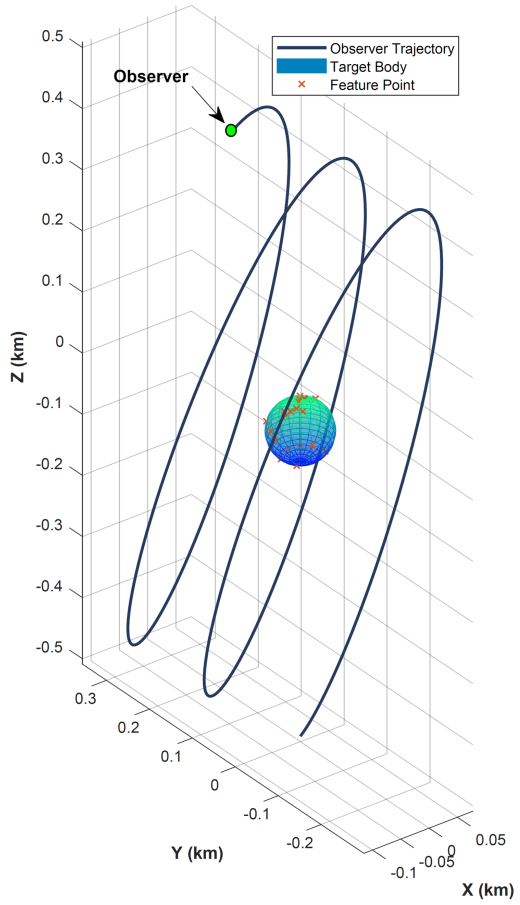

FIGURE 5.2: Trajectory for Case 2, the walking safety ellipse.

Mixture with means located at the true feature locations and covariance of 10 meters. The birth model could also be chosen to be a uniform distribution such that all tracked features are birthed with the same weight, a weight which does not depend on the location in space. This would give the filter no *a priori* knowledge; some work towards this is shown in later sections. Since feature spawning is not expected to occur in this particular scenario (though it is allowed to occur in general,) no spawn intensity function is specified.

All time steps are in units of 50 seconds. Each test case is given 10 uniformly distributed clutter measurements at each time step, in addition to the noisy measurements which originate from true features.

5.2 Standalone GMPHD Results

This section tests the mapping portion of the overall SLAM filter independently of the localization aspect.

Test Case 1: Periodic Orbit

The results of the standalone GMPHD filter applied to the periodic orbit test case are shown in Figures 5.3 and 5.4. Figure 5.3 shows the results in regards to the mapping of the location of the features on the target body over time. The red x's indicate the "true" location of the features, and the blue circles indicate the estimated locations of the features. The periodic nature of the orbit can be seen in these results.

The plot in the upper left shows that early on in the filtering, most of the visible features have been identified by the filter. After many time steps, all the visible features are being tracked, in addition to some extraneous features being estimated. Figure 5.4 shows the difference in the number of visible features vs. the number of estimated features over time. For a well-tuned filter, these two curves should be close over time. From this we see that in general, slightly more features are being estimated than are visible. This makes sense, as seen by the presence of a few extra features in Figure 5.3, which depicts the results of the mapping portion of the filter for a few selected time steps over the trajectory. Occasionally, extra features aren't pruned out, but they were rarely observed to persist through more than one time step; in other words, their existence was not supported by further measurements, causing their weights to fall below the pruning threshold at later time steps. This could be mitigated by further tuning of the pruning threshold variable, but a conservative threshold was purposely chosen here in order to ensure that all visible features are being tracked immediately. In general, as the observer orbits around the target over a long period of time, the filter continues to track features as they enter and exit the sensor field of view.

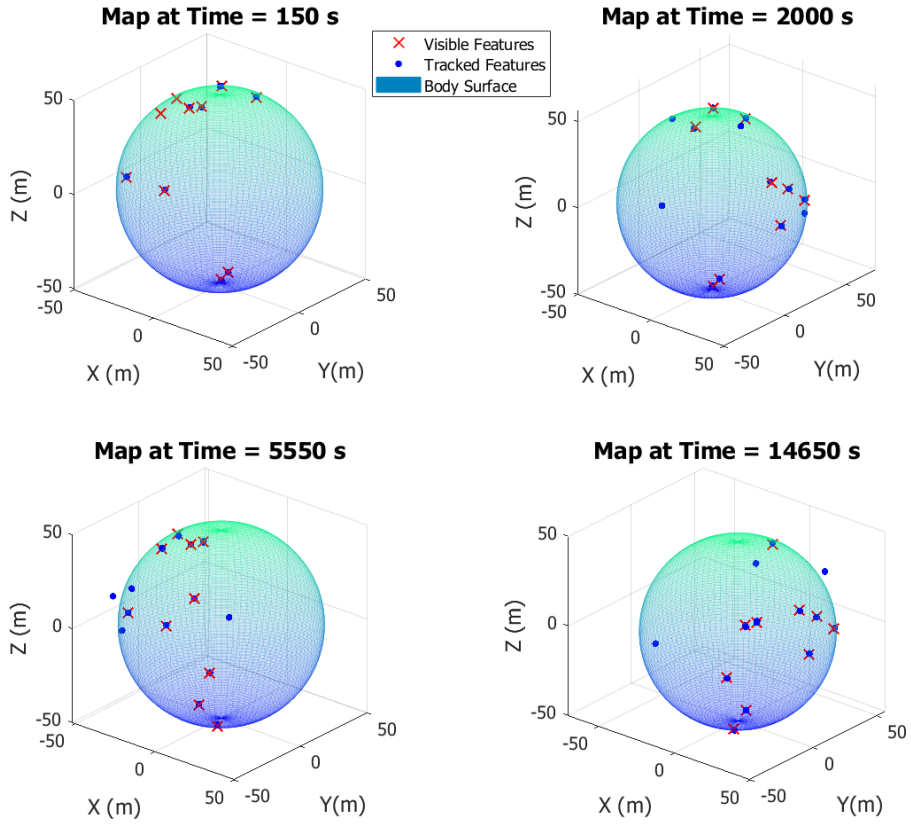

FIGURE 5.3: Features tracked over time using the standalone GMPHD filter for the periodic orbit trajectory.

These results show that the standalone GMPHD filter can successfully perform mapping in the presence of a high amount of clutter over a long period of time when given good information about the relative pose of the observer.

Test Case 2: Walking Safety Ellipse

Figures 5.5 and 5.6 show the results of the standalone GMPHD filter for the walking safety ellipse test case. The upper right plot in Figure 5.5 shows that after just a few time steps, most of the features have been estimated, but not all. This is acceptable, as depending on

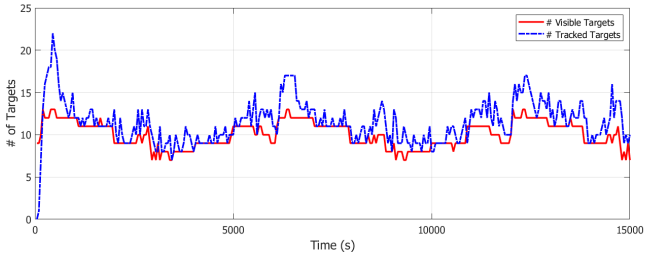

FIGURE 5.4: Number of features tracked using the standalone GMPHD filter for the periodic orbit trajectory.

the tuning of the filter, it can take time for the weight of a feature to grow from birth weight to above the pruning threshold by successive measurement updates. By $t = 900s$, shown in the upper right, it is clear that the filter is tracking all of the visible features.

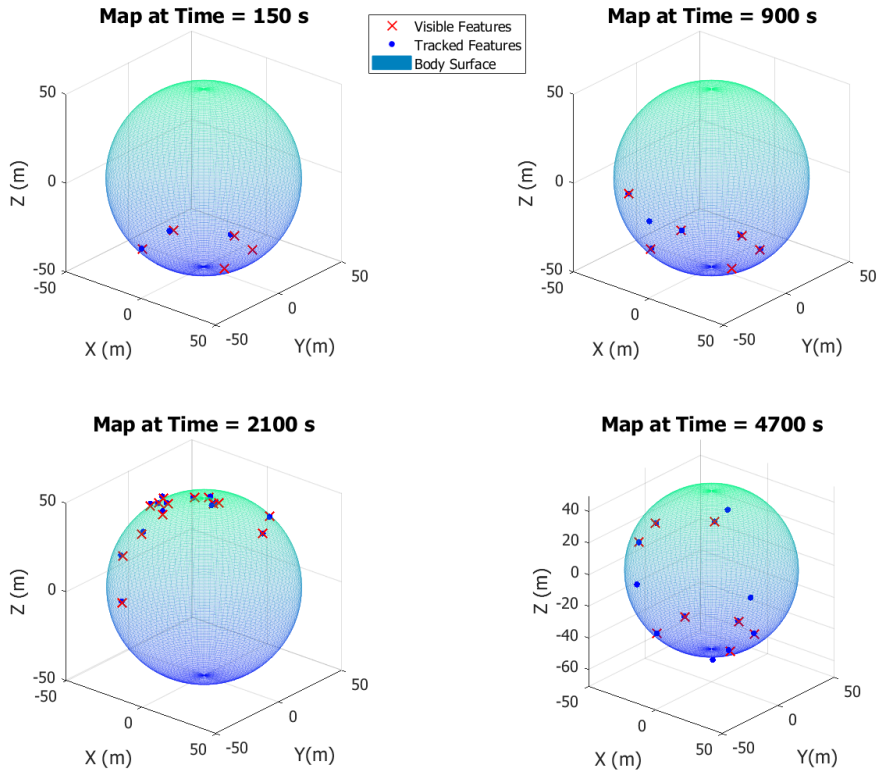

FIGURE 5.5: Features tracked over time using the standalone GMPHD filter for the walking safety ellipse trajectory.

Based on the timeline shown in Figure 5.6, it is seen that the number of visible targets briefly drops to zero as the observer orbits around the target body and features pass in and out of the field of view of the sensor. The plot in the bottom of Figure 5.5 shows that shortly after this completely sparse map, at $t = 2100s$, the filter is able to successfully identify the features once more even though the currently visible features are now on a completely different location on the sphere as compared to $t = 900s$. This map sparsity occurs once more just before $t = 4500s$; the plot on the bottom right shows that by $t = 4700s$, the filter has once again recovered the map in a different location on the sphere.

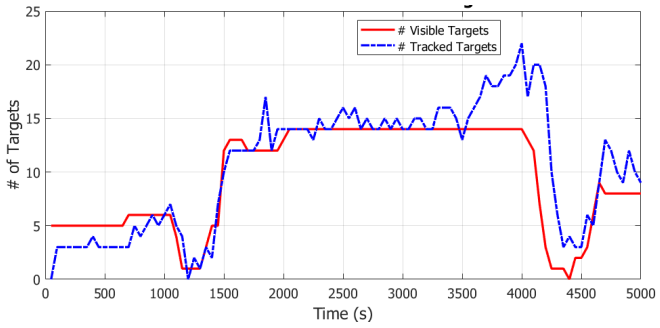

FIGURE 5.6: Number of features tracked using the standalone GMPHD filter for the walking safety ellipse trajectory.

These results show that the standalone GMPHD filter is able to successfully perform the mapping portion SLAM even when the set of tracked features becomes briefly very sparse, and features move in and out of the sensor field of view.

5.3 Rao-Blackwellized GMPHD for SLAM

For the particle filter implementation, 100 particles are used. The pose hypotheses of the particles are initially distributed slightly offset from the true pose state with a covariance corresponding to a standard deviation of 0.1 meters in the position coordinates, 0.1 cm/s

in the velocity coordinates, and 1e-3 in the dimensionless attitude coordinates. This covariance was chosen in order to allow for a relatively small initial particle dispersion on the order of what might be given from an external estimate of the pose from other sensors.

Test Case 1: Periodic Orbit

Figure 5.7 shows the results of the full RBGMPHD filter mapping step for the first case, the periodic orbit. This case was tuned to be slightly less strict about allowing features to be accepted past the pruning step. From the first time step, the map is fully tracked; as time progresses, the features continue to be observed over a long period of time.

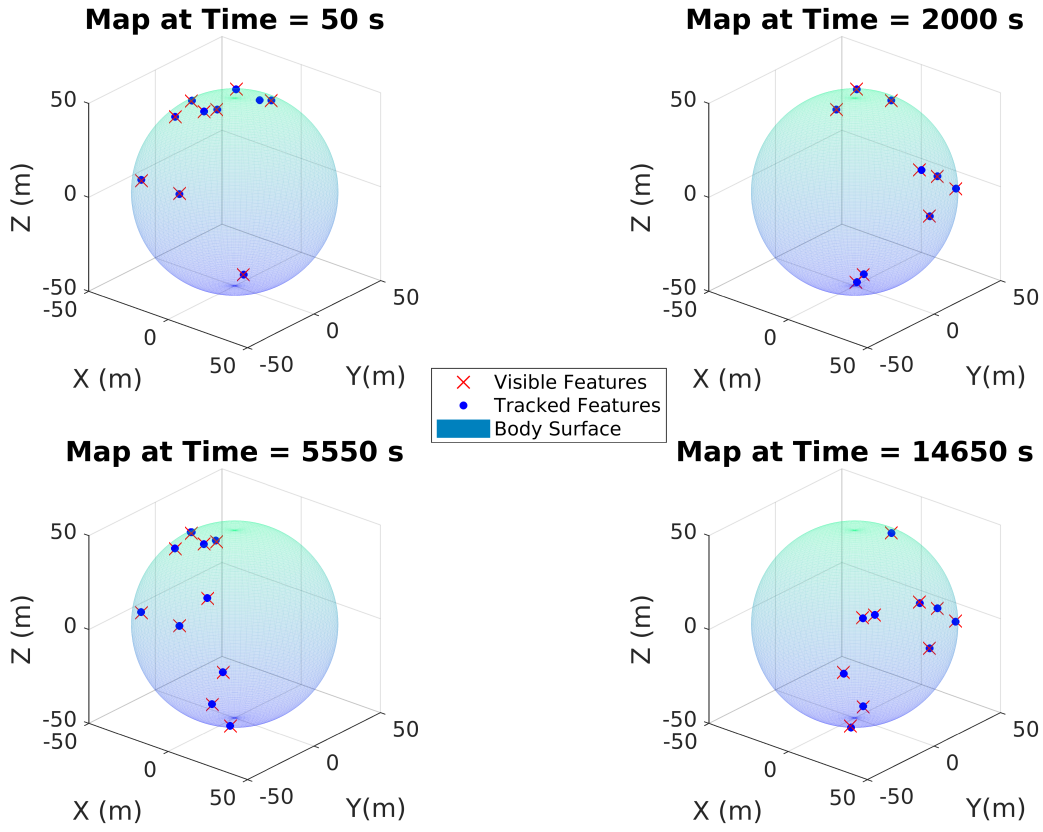

FIGURE 5.7: Sequential snapshots of the estimated feature map for the full RBGMPHD filter used on Case 1.

Figure 5.8 depicts the number of true features visible to the observer compared to the number of features obtained by the PHD filter. Additionally, the less strict pruning of the features is seen as the filter consistently overestimates how many features are visible. Further tuning would make it possible to improve these estimates, as well as implementing the merging procedure specified by Vo and Ma. [25]

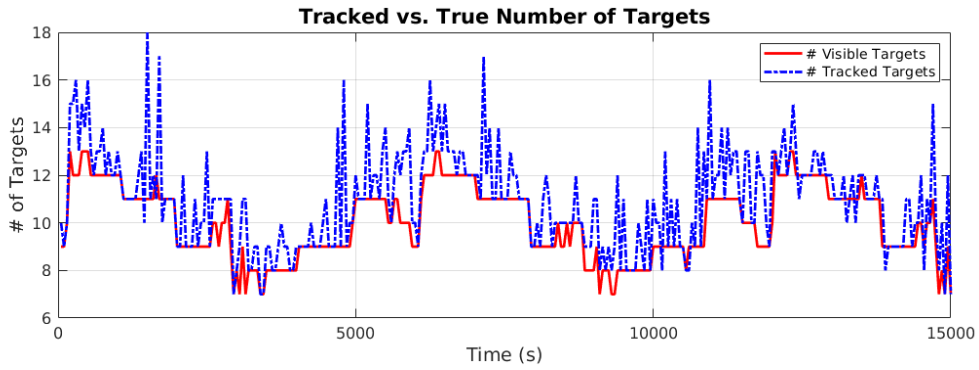

FIGURE 5.8: Comparison between the true number of visible features vs. how many features the RBGMPHD filter has estimated for Case 1.

Figure 5.9 shows the position and velocity errors of the highest weight particle compared to the 3σ bounds, which are calculated from the sample covariance of all the particle poses. The black line shows the difference between the true and estimated pose, and the red dashed line shows the 3σ bounds of the 100 pose hypotheses. Note that there is no sample covariance calculated for the Euler angles, as these have been converted from MRPs. In the MRP formulation, the concept of an additive sample covariance is not valid. The periodic nature of the orbit is seen in the covariance bounds, as the sample covariance periodically increases and decreases, particularly in the Z coordinate which may be slightly less observable due to the planar nature of the orbit. In some dimensions, the filter appears to converge on the correct pose as time progresses.

These results show that the particle filter approach in conjunction with the PHD filter has potential for adequate and stable pose estimation over a long period of time, leading to a successful SLAM formulation.

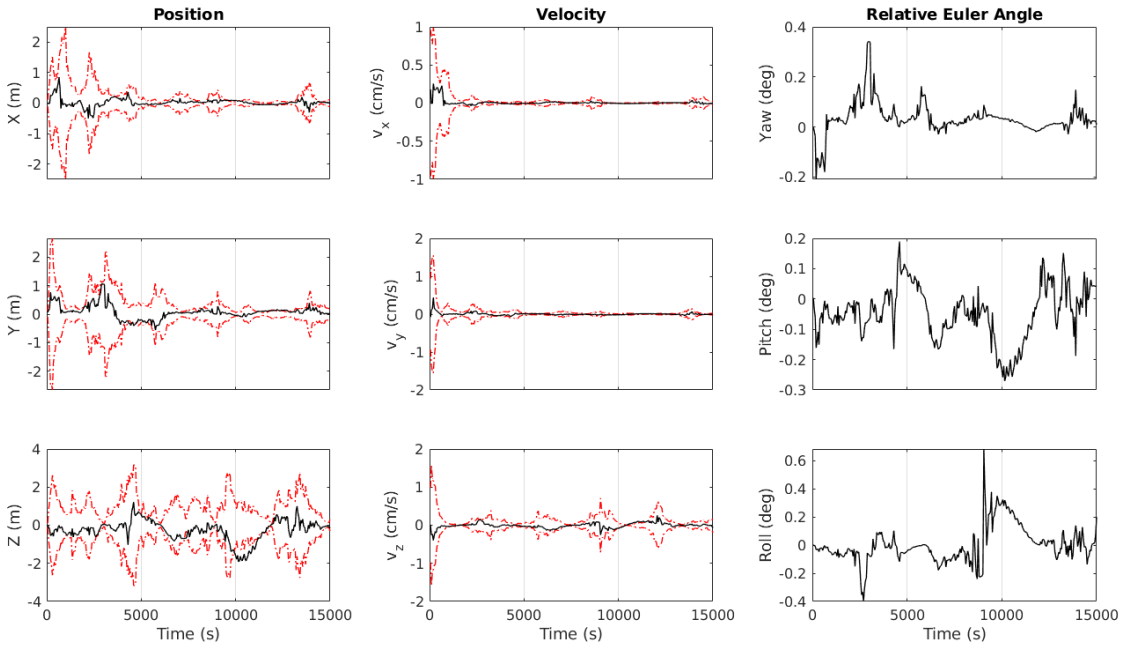

FIGURE 5.9: Comparison between sample covariance 3σ bounds vs. the estimate error for Case 1.

Test Case 2: Walking Safety Ellipse

Figure 5.10 shows the results of the PHD filter for the highest weight particle at 4 selected time steps. From these results, it is clear that the PHD filter is not only determining the feature map from the very first time step, but the map continues to be updated as features move in and out of the field of view as the observer sees different sides of the target. Moreover, when measurements become sparse such as at 1400 seconds, the filter is able to recover and maintain a stable map estimate.

Figure 5.11 depicts the number of true features visible to the observer compared to the number of features obtained by the PHD filter. This figure shows that the estimated number of features roughly tracks the truth.

Figure 5.12 shows the resulting position and velocity errors compared to the sample covariance 3σ bounds of all the particle poses, as well as the Euler angle differences between

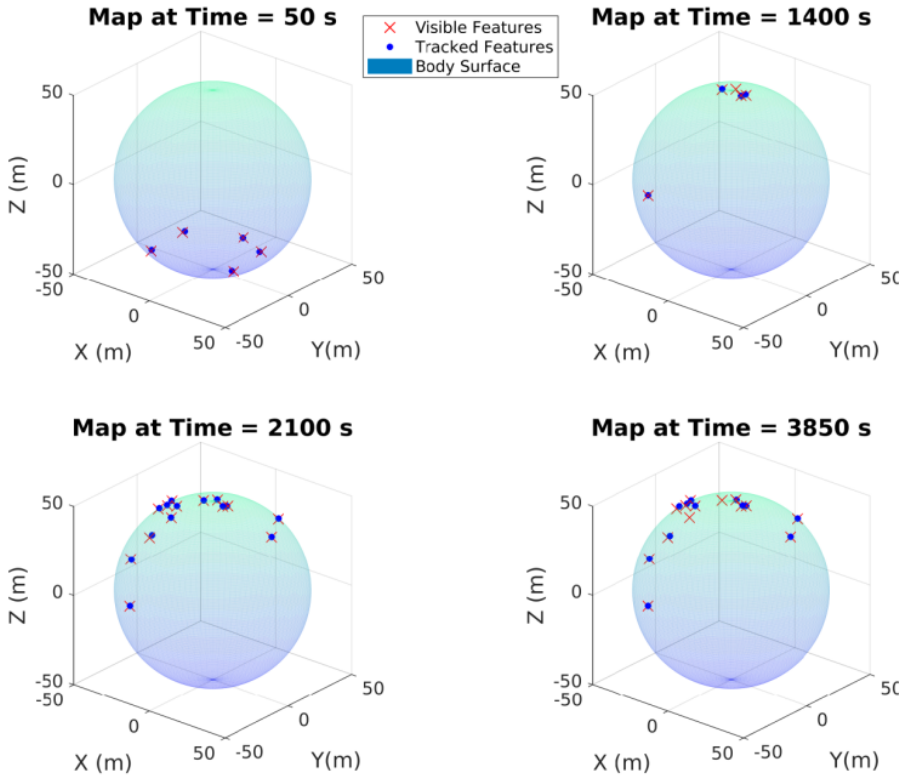

FIGURE 5.10: Sequential snapshots of the estimated feature map from the RBGMPHD filter over time for Case 2.

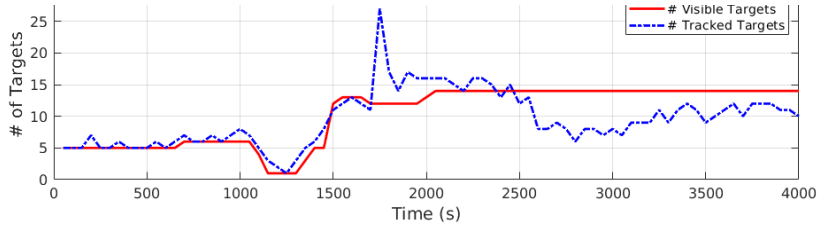

FIGURE 5.11: Comparison between the true number of visible features vs. how many features the RBGMPHD filter has estimated for Case 2.

the true and estimated relative attitude. These results show that the covariance bounds are reasonable for the amount of estimation error, and the pose is being tracked to within a few meters in relative position and cm/s in relative velocity, and generally less than one

degree of error in relative attitude.

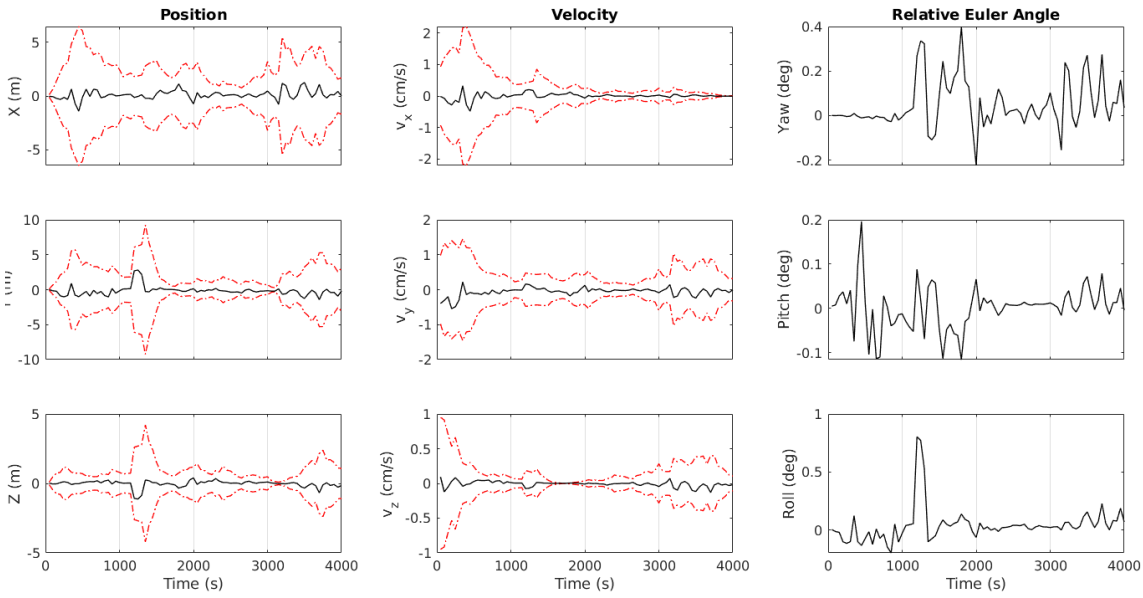

FIGURE 5.12: Comparison between sample covariance 3σ bounds vs. the position and velocity errors for Case 2.

These results show that even in the presence of a sparse feature map with features frequently moving in and out of the field of view of the sensor, the RBGMPHD formulation can successfully perform SLAM.

5.4 Towards True Freedom From *a priori* Reliance

One strength of the RB-GMPHD formulation is that it has the potential to perform SLAM in the complete absence of *a priori* information about the environment. Though the previous results are relatively sparse in terms of *a priori* knowledge given, some information was technically given to the filter in the form of the birth distribution. The birth distribution was specified as a Gaussian mixture with means located at the true positions of the

features on the target body, and covariance of 10 meters. This means that when a measurement is mapped back to a region near a feature location, it is birthed in the filter with a higher probability than a measurement which corresponds to a location farther away from a true feature. Although this is still relatively sparse information compared to the full detailed maps that are frequently used in optical navigation approaches, and a relatively wide covariance was used, this information would be unavailable in situations in which the target has never been explored, and subsequently no knowledge of the feature locations exists.

Thus, these results explore the possibility of an RFS-based filter for SLAM which can function with absolutely no *a priori* information – in other words, a uniform birth distribution such that all measurements are introduced with the same probability of being feasible, and the strengthening or weakening of this probability must come solely from the mechanics of the filter.

5.4.1 Standalone GMPHD

First the standalone GMPHD filter is once again tested, but with a uniform birth density. The walking safety ellipse case is tested as it is the more difficult case in terms of mapping. These results are shown in Figures 5.13 and 5.14.

Compared to the results in Section 5.2 for the same test case with a Gaussian Mixture birth model, not much has changed. In general the filter tends to overestimate the number of features; this was a purposeful tuning choice made to allow a better chance that a feature becomes tracked, since it is more difficult for the filter to initially distinguish clutter from real features with just one measurement update. Despite this, the mapping appears to be comparably successful, and feature measurements which correspond to clutter in general do not persist beyond one time step as no further measurements support their existence.

These results support the possibility that the GMPHD filter can successfully perform

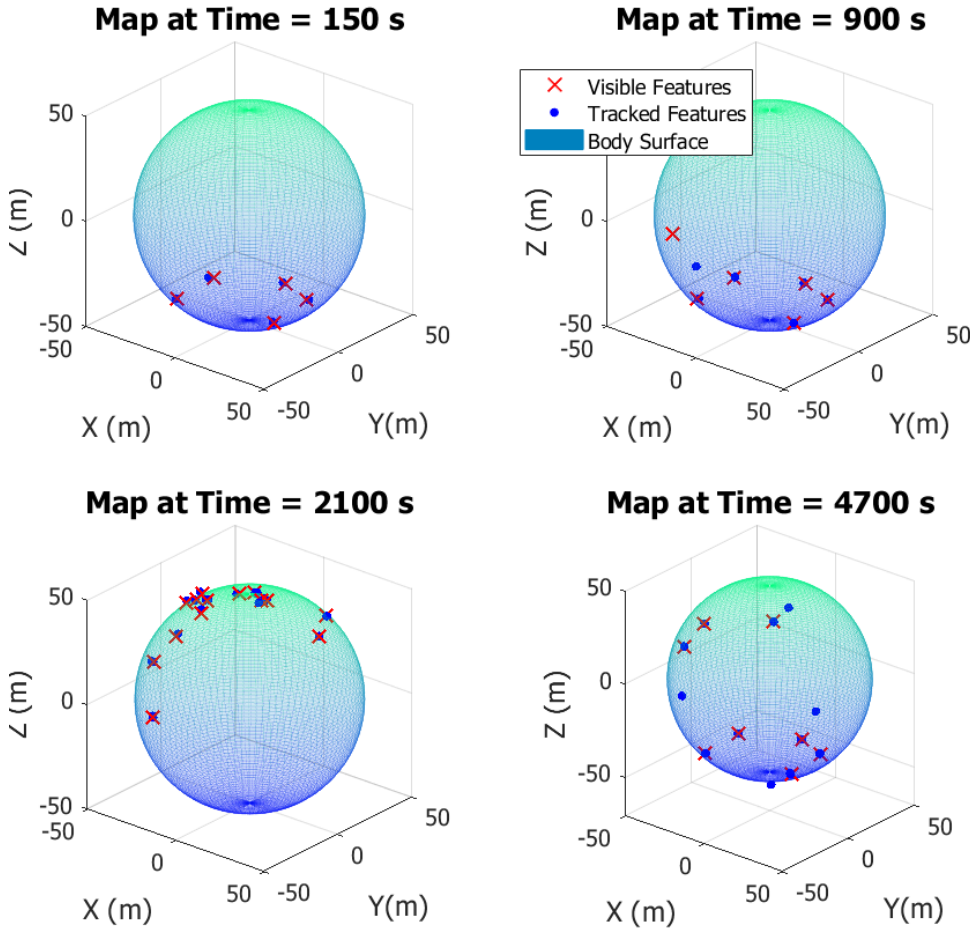

FIGURE 5.13: Sequential snapshots of the estimated feature map from the PHD filter over time for the standalone GMPHD filter given a uniform birth density.

mapping given no *a priori* knowledge about the environment, only dynamics and measurement model information.

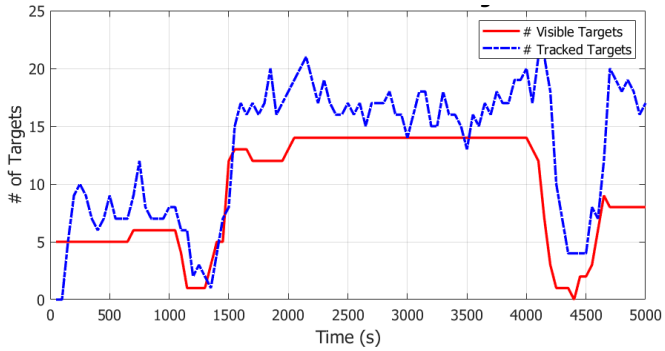

FIGURE 5.14: Comparison between the true number of visible features vs. how many features the standalone GMPHD filter has estimated for Case 2 given a uniform birth model.

5.4.2 Full RBGMPHD

Walking Safety Ellipse

The full RBGMPHD filter is once again tested, but with a uniform birth density. The walking safety ellipse case is again used as it is the more difficult case in terms of mapping. These results are shown in Figures 5.15, 5.16, and 5.17.

It is clear that the filter is having a much more difficult time with weighting clutter measurements, even when the clutter is physically very distant from the target body. In the previous test with the standalone GMPHD filter, a few extra features were typically estimated, however they were usually in close proximity to the target body. Thus, the number of features tracked does not closely follow the number of features visible to the sensor. Despite this, the real features are tracked.

Additionally, given the disparity between the expected feature map and the predicted feature map, the filter is roughly able to estimate the pose of the observer, as seen in estimate errors and sample covariances in Figure 5.17. Compared to previous results of course, these covariances are poor. The sample covariance does not always adequately bound the estimate error for very long. However, the filter does seem to be able to make corrections when the estimate becomes too far off. It appears that when the estimate becomes poor,

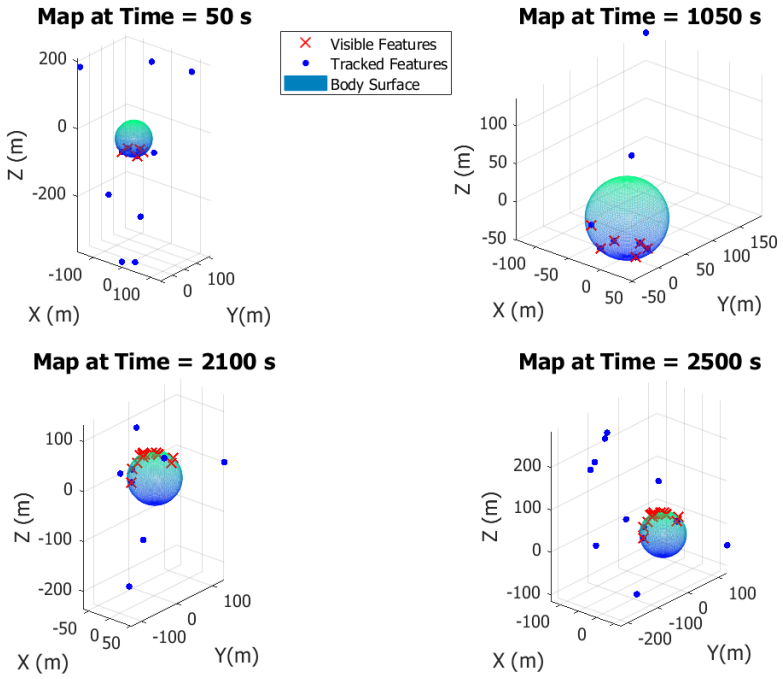

FIGURE 5.15: Sequential snapshots of the estimated feature map from the PHD filter over time for the RBGMPHD filter given a uniform birth model.

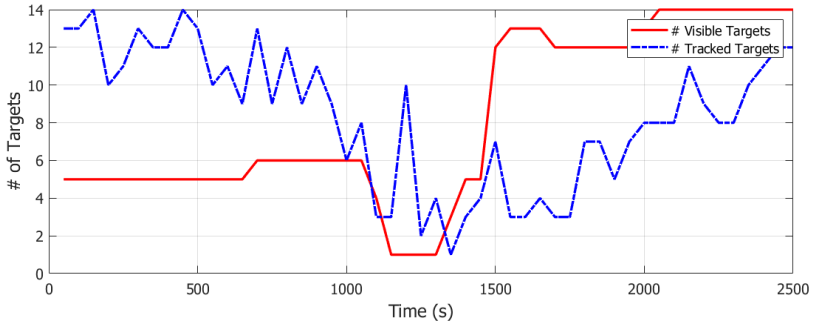

FIGURE 5.16: Comparison between the true number of visible features vs. how many features have been estimated using the RBGMPHD filter given a uniform birth model.

the sample covariance subsequently increases, meaning that the particle pose hypotheses are becoming more distributed in an attempt to rediscover the region where the best estimate is. This is also reflected in the relative Euler angles between the truth and estimate pose. This behavior is exactly what was intended with the resampling scheme chosen, as

described in Chapter 4.

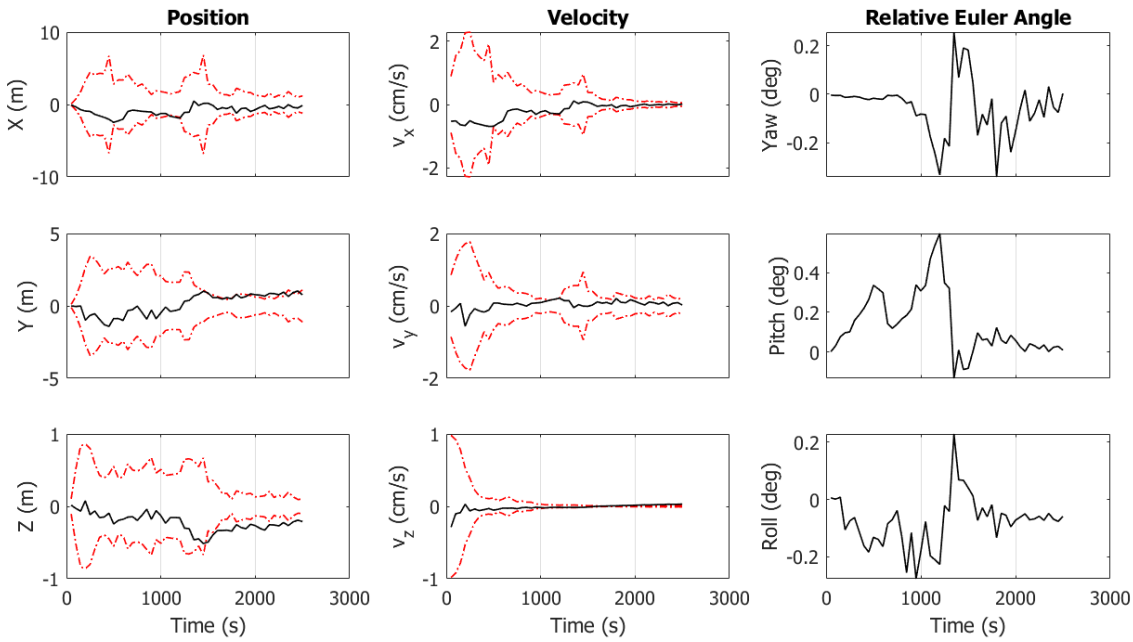

FIGURE 5.17: Comparison between sample covariance 3σ bounds vs. the position and velocity errors for the RBGMPHD filter given a uniform birth model.

These results support the assertion that the RBGMPHD filter may be able to successfully perform SLAM given no *a priori* information about the environment. Though the mapping portion of the filter did not perform as well as previous test cases, the pose estimates show potential regardless. In an environment in which few clutter measurements are likely the mapping portion would likely perform far better, therefore driving even better pose estimation and potentially more robustness to map sparsity.

Periodic Orbit

Unfortunately, the previous walking safety ellipse test case was difficult to simulate for longer than the time steps shown, as the sparsity of the map made it difficult for the filter to recover a pose estimate through the resampling mechanism. Therefore, the periodic

orbit is used to investigate if this recovery behavior is seen over a longer period of time. These results are shown in Figures 5.18, 5.19, and 5.20.

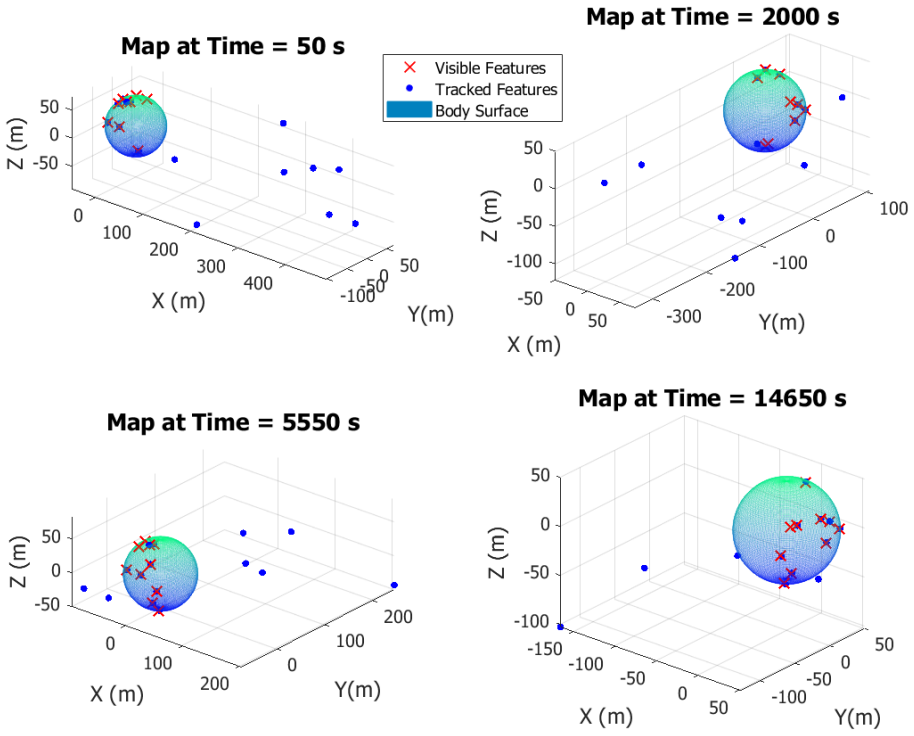

FIGURE 5.18: Sequential snapshots of the estimated feature map from the PHD filter over time for the RBGMPHD filter given a uniform birth model for a periodic orbit.

In general, the mapping portion of these results do not differ much from the walking safety ellipse results; the filter had a difficult time with clutter measurements, but the real features are being tracked.

Figure 5.20 on the other hand is slightly more interesting than the covariance plots for the walking safety ellipse. Though the sample covariances do not seem to reliably bound the estimate error, the correction mechanism is clearly seen, particularly in the Y coordinate and the relative Euler angles. Again, it appears that if the estimate error (i.e. the estimate of the highest weight particle) becomes too high, the sample covariance opens up and the

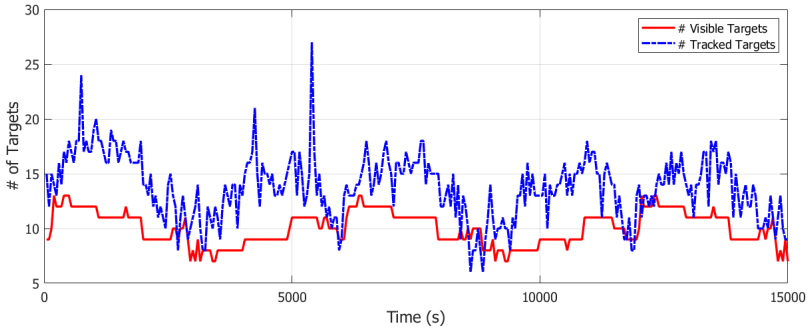

FIGURE 5.19: Comparison between the true number of visible features vs. how many features have been estimated using the RBGMPHD filter given a uniform birth model for a periodic orbit.

particles sample a larger region of state space in an attempt to recover a better estimate. This is particularly seen in the rapid changes in relative Euler angle. Though these results are not necessarily stable, they have not totally diverged after nearly 4 hours of tracking. As long as the sample covariance stays in a reasonable region of state space, it is reasonable to suspect that the filter could continue correcting the pose indefinitely.

These results support the RBGMPHD filter as a viable method for performing SLAM in a rendezvous and proximity operations scenario in which absolutely no *a priori* information about the environment is given to the filter.

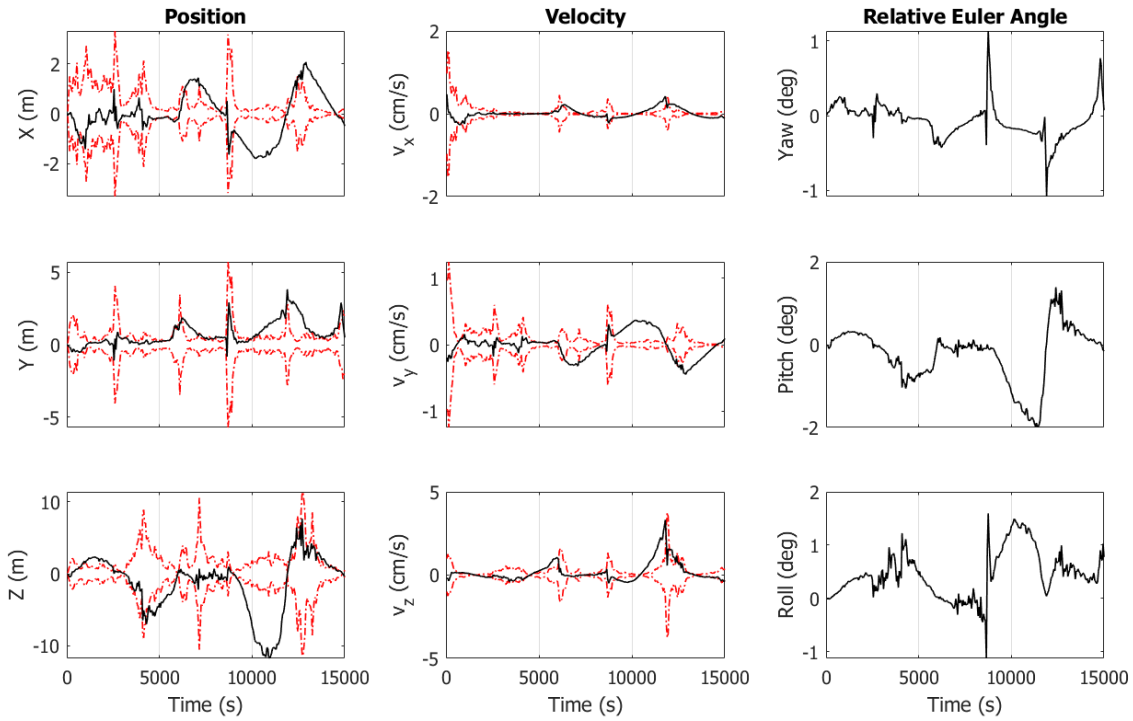

FIGURE 5.20: Comparison between sample covariance 3σ bounds vs. the position and velocity errors for the RBGMPHD filter given a uniform birth model for a periodic orbit.

6 Conclusions

It is clear that innovation is needed in order to achieve autonomous, on-board relative navigation for future space missions, both in orbit around the Earth and in deep space. While traditional methods are in general too computationally complex and reliant on heuristic methods to be seriously viable candidates for this goal, this thesis adds to the growing number of studies that show that the emerging techniques made possible by Random Finite Set based filters offer a rigorous and promising alternative.

By formulating the general relative navigation problem as a SLAM scenario, the notion of performing multi-target tracking as a means towards estimating relative position, velocity, and attitude supports the use of RFS based filters, which are specifically formulated to rigorously perform multi-target tracking. The addition of a particle filter to an RFS filter expands the range of possible parameters which may be estimated, leveraging the dynamics of the underlying probability distributions.

The strength of this RFS based formulation is that many of the significant problems with traditional multi-target tracking schemes are avoided entirely, or dealt with in the initial mathematics used to derive the filter. The primary benefit of this is the avoidance of the need for a separate heuristic data association algorithm to determine a mapping between measurements and features prior to filtering. With RFS based methods, this association is automatically accounted for in the prediction and update steps of the filter; this indirectly allows for much higher dimensional problems to be solved, moving closer towards eventual on-board feasibility.

Additionally, the RFS framework is built from the beginning with the ability to account

for realistic measurement possibilities which may occur with optical navigation filters in space, such as clutter measurements, and the possibility that features can move in and out of the field of view of a sensor over time. These characteristics together make RFS-based filters such as the GMPHD filter a more rigorous and robust framework for performing SLAM for relative navigation purposes.

These strengths have been demonstrated in a generalized rendezvous and proximity operations scenario in this thesis, as well as the prior work forming the basis of this thesis, using a RBGMPHD filter. Simplified but fundamentally realistic dynamics models were used along with a measurement model based on an existing research platform being used in a real space environment for a similar scenario.

In general, the test cases used to test the RBGMPHD filter strive to create a challenging dynamics and measurement situation for the filter; namely an abnormally low signal to noise ratio (i.e. many clutter measurements, at times a ratio of 1:10) attempting to give the filter absolutely no *a priori* knowledge, or poor initial pose knowledge in the particle filter. Realistically, the most expedient use of an RFS-based filter on-board a spacecraft may be less harsh. For example the initial pose of the particle filter may be initialized by another estimate of the spacecraft pose from other sensors. The results of the standalone GMPHD filter in this scenario demonstrate that the RFS-based portion of the filter is very well suited to the multi-target tracking formulation of SLAM for relative navigation, even when no *a priori* information about the environment is given. In fact, the majority of the challenges seen with the overall RBGMPHD filter came directly from the fact that the particle filter is very delicate. Despite this, careful selection of a resampling method can allow the particle filter to salvage estimates even when the estimation errors grow large.

Thus, the main conclusion of these results is that an RFS-based filter like the GMPHD filter is robustly and rigorously suited for multi-target tracking in a spacecraft relative navigation scenario. It would be uniquely suited to perform some of the most difficult aspects of traditional relative navigation when used in conjunction with other methods for

estimating pose given a map of the environment. The exact manner in which this tracking can be used to perform SLAM may not necessarily be a particle filter, though few examples of other solutions exist in the literature.

6.1 Suggested Future Work

This work has primarily been an initial demonstration that an RFS-based formulation is suitable for performing SLAM for spacecraft relative navigation. Further research is necessary in order to quantify the strengths and weaknesses of the approach, as well as how to improve details of the individual parts of the approach.

First and foremost, there must be a consistent way to quantify the mapping portion of the SLAM procedure in a rigorous way. There is a distinct lack of methodology in the literature for properly quantifying the results of an RFS-based filter in a way that is analogous with familiar methods (e.g. root-mean-square estimate error.) One such method does exist called the Optimal SubPattern Assignment (OSPA) method, as introduced in [39] and used in [25]. This method was attempted for these results, however the calculations involve matrices of combinatorial size; when attempted for the results presented here, the matrices required exceeded the memory capabilities of MATLAB. In other words, the OSPA metric used in other studies has been impossible to use for these results due to the large number of measurements used and features being tracked. In general, there are very few other ways to efficiently compare sets of estimated features with sets of actual features – i.e. there is no set-based equivalent to a Euclidean distance. With further time, it may be possible to adapt the OSPA metric to be compatible with a problem of this size.

Based on the process of obtaining the results presented, the most challenging aspect of the presented method is by far the tuning of the resampling step of the particle filter. This is a crucial step, and many different methods exist for performing it. The most common method, low variance resampling, was chosen, but it did not begin to perform well until

the variable resampling covariance was used. Even then, it could be argued that while the resampling behavior of the filter tends to be beneficial in these results for preventing divergence, it does not necessarily facilitate convergence in the traditional sense. Thus, further study into a stronger resampling method could potentially significantly improve both the efficiency and robustness of the filter.

Due to the difficulties with resampling leading to a lack of structured convergence in the pose estimate, one important aspect of the filter still remains to be studied in greater detail. While the initial pose covariance of the particle filter pose distribution was non-zero, the magnitude was kept near an expected covariance that might be provided by an external estimation filter. Testing the robustness of the filter to a much more poor pose estimate would be an important aspect for further study, as it would strengthen the notion that the filter could truly be used with no *a priori* information, both from the environment and from the observer itself. As discussed in the previous paragraph, this would likely not be achievable until a more robust resampling method is implemented.

Finally, the GMPHD filter is just one filter in a family of filters which are based on RFS mathematics. Given that the assumptions used to arrive at the tractable GMPHD recursion are so similar to the assumptions upon which the original Extended Kalman Filter is based on, it is a comfortable and easily understandable first choice for determining if the RFS-based filtering approach is suitable for an intended application, which was the primary purpose of this thesis. However, RFS filters which are based on other assumptions and subsequently have better performance do exist and have been studied extensively in the literature. For example, many of the works cited in this thesis use a variant called the Generalized Labelled Multi-Bernoulli filter for SLAM applications, with overall positive results. [20, 40] A Rao-Blackwellized version of this filter has been studied, similar to the RBGMPHD filter used in this thesis. [41] Moving towards a more extensively used RFS filter would be a wise next step to take to achieve better performance.

Bibliography

[1] NASA. "NASA Strategic Technology Investment Plan". In: (2017).

[2] NASA. "NASA Technology Roadmaps: Robotics and Autonomous Systems". In: 4 (2015).

[3] NASA Satellite Servicing Projects Division. *Restore-L Fact Sheet*. 2016. URL: https://www.nasa.gov/mission_pages/tdm/satellite-servicing.html (visited on 02/18/2019).

[4] Nitin Dhiman, Dipti Deodhare, and Deepak Khemani. "Where Am I? Creating Spatial Awareness in Unmanned Ground Robots Using SLAM: A Survey". In: *Sadhana Academy Proceedings* 40.5 (2015), pp. 1385–1433.

[5] Guilluame Bresson et al. "Simultaneous Localization and Mapping: A Survey of Current Trends in Autonomous Driving". In: *IEEE Transactions on Intelligent Vehicles* 2.3 (2017), pp. 194–220.

[6] John Mullane et al. *Random Finite Sets for Robot Mapping and SLAM: New Concepts in Autonomous Robotic Map Representations*. Berlin: Springer-Verlag, 2011.

[7] Marco Alexander Treiber. *Optimization for Computer Vision*. London: Springer-Verlag, 2013.

[8] Paul Besl and Neil McKay. "Method for Registration of 3-D Shapes". In: *Proceedings of SPIE* 1611.1 (1992), pp. 586–606.

[9] Qi-Xing Huang and Dragomir Anguelov. "High Quality Pose Estimation By Aligning Multiple Scans to a Latent Map". In: *IEEE International Conference on Robotics and Automation* (2010), pp. 1353–1360.

[10] Bo Naasz et al. "Flight Results from the HST SM4 Relative Navigation Sensor System". In: *Proceedings in Advances in the Astronautical Sciences* 137 (2010), pp. 723–744.

[11] Matthew Strube et al. "Raven: An On-Orbit Relative Navigation Demonstration Using International Space Station Visiting Vehicles". In: *Proceedings in Advances in the Astronautical Sciences* 154 (2015), pp. 659–672.

[12] Sumant Sharma and Simone D'Amico. "Pose Estimation for Non-Cooperative Spacecraft Rendezvous Using Neural Networks". In: *Proceedings of the American Astronautical Society Spaceflight Mechanics Meeting* 19.350 (2019).

[13] Sean Augenstein. "Monocular Pose and Shape Estimation of Moving Targets for Autonomous Rendezvous and Docking". In: *Doctoral Thesis, Stanford University* (2011).

[14] Arne Sonnenburg, Marcel Tkocz, and Klaus Janschek. "EKF-SLAM Based Approach for Spacecraft Rendezvous Navigation with Unknown Target Spacecraft". In: *International Federation of Automatic Control Proceedings* 43.15 (2010), pp. 339–344.

[15] Michael Montemerlo et al. "FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem". In: *Proceedings of the National Conference on Artificial Intelligence* 18 (2002), pp. 593–598.

[16] Ethan Eade and Tom Drummond. "Scalable Monocular SLAM". In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 1 (2006), pp. 469–476.

[17] Cedric Cocaud and Takashi Kubota. "SLAM-Based Navigation Scheme for Pinpoint Landing on Small Celestial Body". In: *Advanced Robotics* 26.15 (2012), pp. 1747–1770.

[18] R.P.S. Mahler. "Multitarget Bayes Filtering via First-Order Multitarget Moments". In: *IEEE Transactions on Aerospace and Electronic Systems* 39.4 (2003), pp. 1152–1178.

[19] Ronald Mahler. "A Survey of PHD Filter and CPHD Filter Implementations". In: *Proceedings of SPIE* 6567.1 (2007), pp. 0–12.

[20] James S. McCabe and Kyle J. DeMars. "Feature-Based Robotic Mapping with Generalized Labeled Multi-Bernoulli Filters for Planetary Landers". In: *AIAA/AAS Astrodynamics Specialist Conference* (2016).

[21] James S. McCabe, Kyle J. DeMars, and Carolin Frueh. "Integrated Detection and Tracking for Multiple Space Objects". In: *AAS/AIAA Spaceflight Mechanics Meeting* (2015).

[22] Y Cheng et al. "Gaussian Mixture PHD Filter for Space Object Tracking". In: *Proceedings of the American Astronautical Society Spaceflight Mechanics Meeting* 13 (2013).

[23] Lauren Schlenker et al. "Simultaneous Localization and Mapping for Spacecrfaft Rendezvous and Proximity Operations Using Random Finite Sets". In: *Proceedings of the American Astronautical Society Spaceflight Mechanics Meeting* 19.282 (2019).

[24] Ronald Mahler. *Statistical Multisource-Multitarget Information Fusion*. Boston, MA: Artech House, 2007.

[25] Ba-Ngu Vo and Wing-Kin Ma. "The Gaussian Mixture Probability Hypothesis Density Filter". In: *IEEE Transactions on Signal Processing* 54.11 (2006), pp. 4091–4104.

[26] Keith Leung, Felipe Inostroza, and Martin Adams. "Multifeature-Based Importance Weighting for the PHD SLAM Filter". In: *IEEE Transactions on Aerospace and Electronic Systems* 52.6 (2016), pp. 2697–2714.

[27] John Mullane et al. "Mobile Robotics in a Random Finite Set Framework". In: *International Conference in Swarm Intelligence* 6279.2 (2011), pp. 519–528.

[28] Chee Sing Lee, Daniel Clark, and Joaqium Salvi. "SLAM With Dynamic Targets via Single-Cluster PHD Filtering". In: *IEEE Selected Topics in Signal Processing* 7.3 (2013), pp. 543–552.

[29] Tiancheng Li, Miodrag Bolic, and Petar M. Djuric. "Resampling Methods for Particle Filtering". In: *IEEE Signal Processing Maganize* 5 (2015), pp. 70–86.

[30] Sebastian Thrun, Wolfram Burgard, and Deiter Fox. *Probabilistic Robotics*. Boston, MA: MIT Press, 2005.

[31] David Vallado and Wayne McClain. *Fundamentals of Astrodynamics and Applications*. 2nd. El Segundo, CA: Microcosm Press, 2001.

[32] Hanspeter Schaub. *Analytical Mechanics of Space Systems*. 2nd. Reston, VA: AIAA, 2009.

[33] David Lowe. "Object Recognition from Local Scale-Invariant Features". In: *International Conference on Computer Vision* 2 (1999), pp. 1150–1158.

[34] Herbert Bay et al. "Speeded-Up Robust Features". In: *Computer Vision and Image Understanding* 110.3 (2008), pp. 346–359.

[35] M .Jayendra-Lakshman and V. Devarajan. "A New Feature Descriptor for LIDAR Image Matching". In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* II-2/W1 (2013), pp. 157–162.

[36] "Fundamentals of Vision Based Navigation". In: *NASA Engineering and Safety Center Seminars* (2014).

[37] Matthew Strube et al. "Raven: An On-Orbit Relative Navigation Demonstration Using International Space Station Visiting Vehicles". In: *AAS GNC Conference Proceedings* 15.111 (2015).

[38] Matthew Vavrina et al. "Safe Rendezvous Tracjetory Design for the Restore-L Mission". In: *American Astronautical Society Spaceflight Mechanics Meeting* 19.410 (2019).

[39] Dominic Schuhmacher, Ba-Tuong Vo, and Ba-Ngu Vo. "A Consistent Metric for Performance Evaluation of Multi-Object Filters". In: *IEEE Transactions on Signal Processing* 56.8 (2008), pp. 3447–3457.

[40] James S. McCabe. "Multitarget Tracking and Terrain-Aided Navigation Using Square-Root Consider Filters". In: *PhD Dissertation* (2018).

[41] Hendrik Deusch, Stephan Reuter, and Klaus Dietmayer. "The Labeled Multi-Bernoulli SLAM Filter". In: *IEEE Signal Processing Letters* 22.10 (2015), pp. 1561–1565.