(Preprint) AAS 19-282

SIMULTANEOUS LOCALIZATION AND MAPPING FOR SATELLITE RENDEZVOUS AND PROXIMITY OPERATIONS USING RANDOM FINITE SETS

Lauren Schlenker,[*] Mark Moretto,[†] David Gaylor,[‡] and Richard Linares[§]

Future space missions require that spacecraft have the capability to autonomously navigate non-cooperative environments for rendezvous and proximity operations (RPO). Current relative navigation filters can have difficulty in these situations, 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. The goal of this work is to demonstrate the feasibility of random finite set (RFS) filters for spacecraft relative navigation and pose estimation. The approach is to formulate satellite relative navigation and pose estimation 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. This work utilizes a filter developed using the framework of RFS which are well suited to multi-target SLAM operations, avoiding data association entirely. Relevant RPO scenarios with simulated flash LIDAR measurements are tested with a Probability Hypothesis Density (PHD) RFS filter embedded in a particle filter to obtain a feature map of a target and a relative pose estimate between the target and observer. Preliminary results show that an RFS-based filter can successfully perform SLAM in a spacecraft relative navigation scenario with no *a priori* map of the target. These results demonstrate the feasibility of RFS filtering for spacecraft relative navigation and motivate future studies which may expand to tracking space objects for space situational awareness, as well as relative navigation around small bodies.

INTRODUCTION

There is a growing need for autonomous navigation solutions for spacecraft rendezvous and proximity operations (RPO) to support small body exploration, satellite servicing and on-orbit assembly and active orbital debris removal. However, reliable solutions are difficult to achieve due to the challenges of operating in space. Noisy measurements that are cluttered with additional false detections can make a state estimation filter diverge if association between measurements and state space cannot be performed quickly and correctly. Moreover, existing methods typically rely heavily on *a priori* information about the target or known features such as fiducials.[1–4] This information may not be available in situations such as RPO with non-cooperative satellites or initial mapping of a previously unexplored asteroid or space debris. A reliable relative navigation solution for spacecraft RPO must be able to perform despite these limitations.

[*] Graduate Student, Aerospace Engineering and Mechanics, University of Minnesota, Minneapolis, MN.

[†] Graduate Student, Aerospace Engineering Sciences, University of Colorado Boulder, Boulder, CO.

[‡] Aerospace Engineer, Navigation and Mission Design Branch, NASA Goddard Space Flight Center, Greenbelt, MD.

[§] Charles Stark Draper Assistant Professor, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA.

Satellite relative navigation can be formulated as a Simultaneous Localization and Mapping (SLAM) problem, in which a robotic agent attempts to estimate a map of its environment at the same time as its pose (position, velocity, and attitude) within that environment. With feature-based SLAM methods, the robot's sensors typically measure features in the environment, for RPO they would be the location of points or edges on the body of a rendezvous target using sensors such as cameras or LIDAR. In this way the mapping part of the SLAM problem becomes a multi-target tracking problem, as the observer is attempting to estimate the location of many features in order to construct a feature map of the target.

Many methods have been proposed to solve the multi-target SLAM problem and are summarized by surveys of current SLAM methods[5,6] These existing methods however tend to be limited by the fact that the underlying mathematics were originally intended for single-target tracking. In order to use the same methods for multi-target tracking, heuristic methods must be used to account for additional challenges introduced by multi-target tracking. Among the most significant of these challenges is the data association problem, where measurements must be matched to features in state space so that an estimation filter can be used for navigation. False sensor returns can make it difficult to assign a measurement to a feature prior to filtering; some sort of heuristic must be used to decide how to assign measurements to expected features despite having extraneous measurements available. Conversely, a filtering algorithm must be able to take into account the fact that a sensor may fail to detect a feature that is expected, which commonly occurs with optical measurements due to difficult lighting conditions such as those found in space. With a low number of features and measurements, heuristic methods may suffice for solving these problems on a case by case basis. However, as the number of features and measurements increases, as in a realistic environment, the data association problem becomes computationally challenging and the filters that rely on this problem being solved are prone to diverge if the association is performed incorrectly.[7]

One of the most widely used feature-based SLAM algorithms is FastSLAM, in which the feature map is estimated by an Extended Kalman Filter (EKF) and the corresponding trajectory and pose is estimated with a particle filter.[8] FastSLAM is a powerful approach because there is no Gaussian assumption on the pose probability distribution, allowing for highly non-linear problems to be solved; moreover, real-time implementations of FastSLAM have been achieved.[9]

Several past studies have focused on using SLAM methods for spacecraft rendezvous and proximity operations. Work by Augenstein as well as Sonnenburg et. al. has demonstrated that the spacecraft RPO problem can be appropriately handled by a SLAM formulation.[10,11] However, in both cases feature management and data association must be performed before measurements are passed to the filter, and no concept of realistic measurement situations such as features passing in and out of the field of view, clutter (extra measurements), or missed-detections exists in these formulations. These difficult aspects of a realistic SLAM problem are crucial to the success of the filter. Similarly, Cocaud and Kubota used an algorithm based on an improved particle filtering method intended for navigation and pinpoint landing on small celestial bodies.[12] Though these results address issues with drift common to particle filter based SLAM algorithms such as FastSLAM, the underlying approach is still heavily based on heuristic methods for data association of observed landmarks from frame to frame. From these studies, we see that although the formulation of multi-target tracking for spacecraft RPO as a SLAM problem is feasible, an efficient and reliable method which does not rely on heuristic methods for data association and map management is still needed.

Recently, a new family of filters has emerged that is formulated in such a way that the difficulties introduced by multi-target tracking can be directly incorporated into the mathematics of the filter, negating the need for an intermediate data association step or other heuristics for feature management. Random Finite Set (RFS) filters use a set-based mathematical framework, as compared to the random vector framework seen in traditional Kalman filter based methods. This is a powerful way of formulating the filter because it more naturally describes the kinds of realistic dynamic and measurement situations that occur in multi-target tracking problems.

The first RFS filter was proposed by Ronald Mahler in the early 2000's, called the Probability Hypothesis Density (PHD) filter.[13] Since the initial PHD filter, several variants of RFS filters have been proposed that address additional realistic implementation aspects of the PHD filter in ground-based and underwater robotics.[14] Previous studies by Mullane, et. al., have shown that a feature based map is fundamentally a finite set, and is thus better represented and manipulated with RFS methods.[7] Their results show that in the presence of uncertain dynamics and measurements and high degrees of clutter, the PHD filter significantly outperforms EKF-based mapping filters as well as FastSLAM.

These studies have shown that RFS-based methods such as the PHD filter are a promising approach for handling the difficulties inherent to multi-target tracking. However, the authors are not aware of any previous application of RFS-based methods to spacecraft relative navigation. Thus, the goal of this work is to demonstrate that an RFS-based Bayesian SLAM approach is appropriate and effective for the spacecraft RPO problem.

This paper begins by introducing the basics of RFS theory to explain why this type of mathematics is more naturally suited to SLAM for RPO. The original PHD filter is shown, and the results of a derivation for a tractable recursion called the Gaussian Mixture PHD filter is given. The RPO simulation setup and system dynamics are described, and a flash LIDAR measurement model is given. An approach for using the PHD filter with an outer Rao-Blackwellized particle filter is proposed to perform SLAM, and two relevant RPO scenarios are tested with the filter, processing simulated measurements.

THEORY

The Probability Hypothesis Density Filter

The goal of multi-target tracking is to jointly estimate the number of features present in state space as well as their states given a set of noisy, cluttered measurements. We assume that at time step k, there exists a set of $N(k)$ features and $M(k)$ measurements. Since in general there is no particular order in which we know these states and measurements must be associated, we represent them as finite sets:

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

$$Z_k = \{z_{k,1}, ..., z_{k,M(k)}\} \tag{2}$$

Thus in the RFS framework, the set of features X_k and set of measurements Z_k are a multi-target state and multi-target observation. Analogously to random vectors used for single-target tracking, X_k and Z_k are random finite sets, or finite-set-valued variables which are characterized by a probability distribution and a family of joint probability densities of the elements of X and Z.[15]

Formulating our state in this way allows for a more generalized set of options for the time evolution of states. For a given multi-target state X_{k-1}, each element (which shall hereon be referred to as a feature per the SLAM application intended) $x_{k-1} \in X_{k-1}$ either continues existing at the next time step k (the probability of this happening is $p_{S,k}$), or ceases to exist (with probability $1 - p_{S,k}$). Alternatively, a new feature may occur independently of existing features (i.e., be birthed into state space,) or appear by spawning from an existing feature. In terms of realistic scenarios, these options may occur because a feature is entering the field of view, or was previously obscured by an existing feature.

Thus at a time step k, we denote our set of feature states as:

$$X_k = \left[\bigcup_{\zeta \in X_{k-1}} S_{k|k-1}(\zeta) \right] \cup \left[\bigcup_{\zeta \in X_{k-1}} \Gamma_{k|k-1}(\zeta) \right] \cup B_k \tag{3}$$

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

Similarly, the RFS measurement model is able to take into account probabilities of detection and clutter. A feature $x_k \in X_k$ can either be detected (with probability $p_{D,k}$,) or 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 feature.

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

$$Z_k = K_k \cup \left[\bigcup_{x \in X_k} \Theta_k(x) \right] \tag{4}$$

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 the multi-target observation Z_k. The posterior density can be calculated using a Bayesian recursion given by:

$$p_{k|k-1}(X_k|Z_{1:k-1}) = \int f_{k|k-1}(X_k|X) p_{k-1}(X|Z_{1:k-1}) \mu_s(dX) \tag{5}$$

$$p_k(X_k|Z_{1:k}) = \frac{g_k(Z_k|X_k) p_{k|k-1}(X_k|Z_{1:k-1})}{\int g_k(Z_k|X) p_{k|k-1}(X|Z_{1:k-1}) \mu_s(dX)} \tag{6}$$

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.[16]

The recursion in equations 5-6 for calculating the multi-target posterior density is computationally intractable due to the set integrals required. In order to approximate this recursion, we will instead choose to propagate the posterior intensity, which is the first order statistical moment of the multi-target state. This is the basis of the Probability Hypothesis Density filter.[13] A key characteristic of this choice is that the integral of the intensity function over some region of state space is equal to the number of features expected in that region of state space, thus allowing for joint propagation of the number of expected features at each time step. In other words, $\hat{N}(k) = \int \nu(x)dx$, where the intensity function $\nu(x)$ is referred to as the probability hypothesis density.

From this point, a number of assumptions are made in order to obtain the PHD recursion from equations 5-6. Each target is assumed to evolve and generate observations independently of one another. Clutter is assumed to be Poisson and independent of target-originated measurements. The predicted multiple-target RFS governed by $p_{k|k-1}$ is assumed to be Poisson. Each target is assumed to follow a linear Gaussian dynamical model and the sensor is assumed to obey a linear Gaussian measurement model. The survival and detection probabilities are state independent and the intensities of the birth and spawn RFSs are modeled as Gaussian mixtures.[16]

The PHD recursion is then as follows:

$$v_{k|k-1}(x) = \int p_{S,k}(\zeta) f_{k|k-1}(x|\zeta) v_{k-1}(\zeta) d\zeta + \int \gamma_{k|k-1}(x|\zeta) v_{k-1}(\zeta) d\zeta + \beta_k(x) \tag{7}$$

$$v_k(x) = [1 - p_{D,k}(x)] v_{k|k-1}(x) + \sum_{z \in z_k} \frac{p_{D,k}(x) g_k(z|x) v_{k|k-1}(x)}{\kappa_k(z) + \int p_{D,k}(\xi) g_k(z|\xi) v_{k|k-1}(\xi) d\xi} \tag{8}$$

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 $p_{D,k}$ is now incorporated.

It is worth noting that the PHD recursion in equations 7 and 8 requires no data association between measurements and features, negating the need for expensive combinatorial computations. Unfortunately, no closed-form exists for the PHD recursion.[13, 14, 16]

Gaussian Mixture PHD

In order to produce a tractable, closed form recursion, linear Gaussian dynamics and measurement models are assumed. All subsequent theory presented here to enforce this assumption closely follows the derivation of Vo and Ma, and the interested reader is encouraged to reference their work.[16] A linear Gaussian multi-target model includes assumptions not only for existing features but also for birth, death, and detection of new features. Additionally, it assumes that individual features follow linear Gaussian dynamics and measurement model similar to a classic Kalman filter setup:

$$f_{k|k-1}(x|\zeta) = \mathcal{N}(x; F_{k-1}\zeta, Q_{k-1}) \tag{9}$$

$$g_k z|x = \mathcal{N}(z; H_k x, R_k) \tag{10}$$

where in general $\mathcal{N}(\cdot; m, P)$ represents a normal Gaussian distribution with mean m and covariance P, 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.

We also assume that the birth and spawn intensities are Gaussian mixtures:

$$\gamma_k(x) = \sum_{i=1}^{J_{\gamma,k}} w_{\gamma,k}^{(i)} \mathcal{N}(x; m_{\gamma,k}^{(i)}, P_{\gamma,k}^{(i)}) \tag{11}$$

$$\beta_{k|k-1}(x|\zeta) = \sum_{j=1}^{J_{\beta,k}} w_{\beta,k}^{(j)} \mathcal{N}(x; F_{\beta,k}^{(i)}\zeta + d_{\beta,k-1}^{(j)}, Q_{\beta,k}^{(j)}) \tag{12}$$

where $J_{\gamma,k}, w_{\gamma,k}^{(i)}, P_{\gamma,k}^{(i)}$ are given model parameters that determine the shape of the intensities and can be chosen depending on the scenario, allowing for the inclusion of *a priori* knowledge of the targets if desired or available.

Additionally, we assume that the posterior intensity at a previous time step $k-1$ is also a Gaussian mixture:

$$\nu_{k-1}(x) = \sum_{i=1}^{J_{k-1}} w_{k-1}^{(i)} \mathcal{N}(x; m_{k-1}^{(i)}, P_{k-1}^{(i)}) \tag{13}$$

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 features:

$$v_{k|k-1}(x) = p_{S,k} \sum_{j=1}^{J_{k-1}} w_{k-1}^{(j)} \mathcal{N}(x; m_{S,k|k-1}^{(j)}, P_{S,k|k-1}^{(j)}) + v_{\beta,k|k-1}(x) + \gamma_k(x) \tag{14}$$

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

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

$$v_{\beta,k|k-1}(x) = \sum_{j=1}^{J_k-1} \sum_{\ell}^{J_{\beta,k}} w_{k-1}^{(j)} w_{\beta,k}^{(\ell)} \mathcal{N}(x; m_{\beta,k|k-1}^{(j,\ell)}, P_{\beta,k|k-1}^{(j,\ell)}) \tag{15}$$

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

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

$$v_k(x) = (1 - p_{D,k}) v_{k|k-1}(x) + \sum_{z \in Z_k} v_{D,k}(x; z) \tag{16}$$

where the intensity of detected features is:

$$v_{D,k}(x; z) = \sum_{j=1}^{J_{k|k-1}} w_k^{(j)}(z) \mathcal{N}(x; m_{k|k}^{(j)}(z), P_{k|k}^{(j)}) \tag{17}$$

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

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

$$w_k^{(j)}(z) = \frac{p_{D,k} w_{k|k-1}^{(j)} q_k^{(j)}(z)}{\kappa_k(z) + p_{D,k} \sum_{\ell=1}^{J_{k|k-1}} w_{k|k-1}^{(\ell)} q_k^{(\ell)}(z)}$$

$$q_k^{(j)}(z) = \mathcal{N}(z; H_k m_{k|k-1}^{(j)}, R_k + H_k P_{k|k-1}^{(j)} H_k^T)$$

At this point, equations 14-17 give a tractable closed-form solution for the GM-PHD filter for taking noisy cluttered measurements and extracting an intensity function which represents likely feature positions in state space. It is very similar to the form of the Kalman filter, but is more general due to the RFS formulation. A thorough example of pseudocode for an implementation of this recursion can be found in the book by Mullane et. al.[7]

A note on implementation: at time k, the Gaussian Mixture produced by the recursion has $\mathcal{O}(J_{k-1}|Z_k|)$ terms, and this number increases without bound. A good approximation of the GM-PHD can still be obtained by pruning out elements of the GM whose weight falls below a certain tunable threshold. A simple method for performing this pruning is given by Vo and Ma.[16] This step is simple but crucial to achieving good filter performance.

Extended Kalman GM-PHD

Similar to how the Kalman filter can be extended to nonlinear dynamics and measurement models, Vo and Ma have shown that we can also use nonlinear dynamics and measurements by locally linearizing the state transition matrix and observation matrix. Thus, assuming dynamics and measurement models of the form:

$$x_k = \phi_k(x_{k-1}, \nu_{k-1}) \tag{18}$$

$$z_k = h_k(x_k, \epsilon_k) \tag{19}$$

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. Then, the state transition matrix can be calculated as:

$$F_{k-1}^{(j)} = \left.\frac{\partial \phi_k(x_{k-1}, 0)}{\partial x_{k-1}}\right|_{x_{k-1} = m_{k-1}^{(j)}} \tag{20}$$

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

$$H_k^{(j)} = \left.\frac{\partial h_k(x_k, 0)}{\partial x_k}\right|_{x_k = m_{k|k-1}^{(j)}} \tag{21}$$

Particle Filter

In order to demonstrate the usefulness of an RFS-based filter for spacecraft RPO, the work in this paper replaces the EKF portion of FastSLAM with a PHD filter. The GM-PHD filter in equations 14-17 is used to perform the Mapping part of SLAM, with a similar approach to that of Vo and Ma.[16] In order to perform Localization and obtain a relative pose estimate of the observer, the map from the PHD filter is used. Based on the work by Mullane et. al., the localization step is performed with a particle filter outer loop.[7] A basic flowchart of the algorithm showing the relationship between the GM-PHD filter and the particle filter is presented in Figure 1.

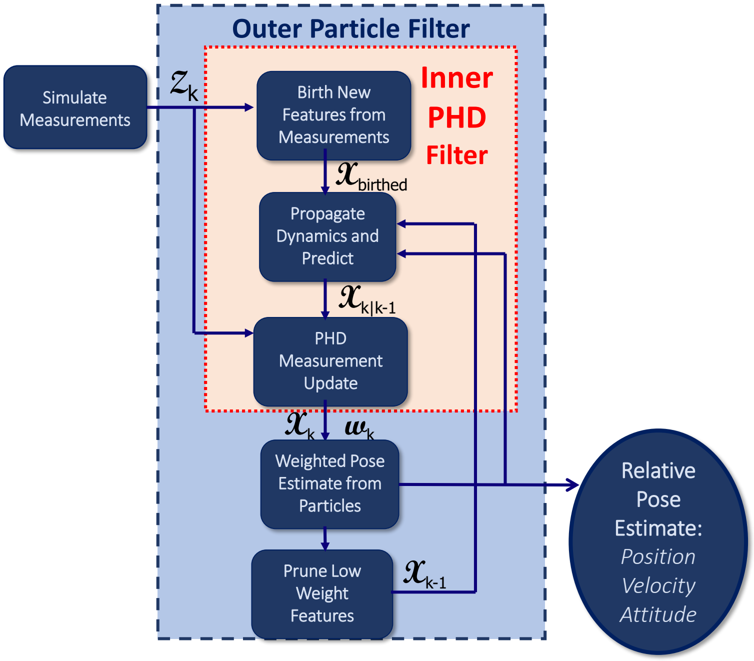


Figure 1: Filter Structure Overview: a PHD filter is wrapped inside a particle filter. Each particle has a PHD filter that processes a map based on a pose hypothesis.

Each particle carries a separate hypothesis of the pose and a corresponding map estimate from the PHD filter that is conditioned on the pose hypothesis. Thus, it is logical to weight each particle based on how closely the map and pose estimate match with the set of measurements. To do this, an importance weight $p(Z_k|X_k, Z_{1:k-1})$ must be calculated for each particle. There are many different methods for performing particle importance weighting, each with varying levels of computation required.[17] Single Cluster importance weighting is used in this approach for its low computational complexity and because the setup of the multi-target problem can be loosely approximated as a single-cluster Poisson process.[18] The updated weight $\eta_k^{[l]}$ of the l^{th} particle is then calculated from the previous map PHD $\nu_k^{-[l]}$ as:

$$\eta_k^{[l]} = \exp\left(\sum_{r=1}^{N(k)^{-[l]}} w_k^{r,[l]}\right) \times \prod_{z \in Z_k} \left(\kappa(z) + p_D \sum_{r=1}^{N(k)^{-[l]}} p\left(z|\mathcal{N}_k^{r,[l]}, x_{0:k}^{[l]}\right) w_k^{r,[l]}\right) \eta_{k-1}^{[l]} \tag{22}$$

The relative pose estimate is then updated by extracting the pose of the particle with the highest weight prior to resampling. Alternatively, a weighted average pose estimate could be obtained using all available particles.

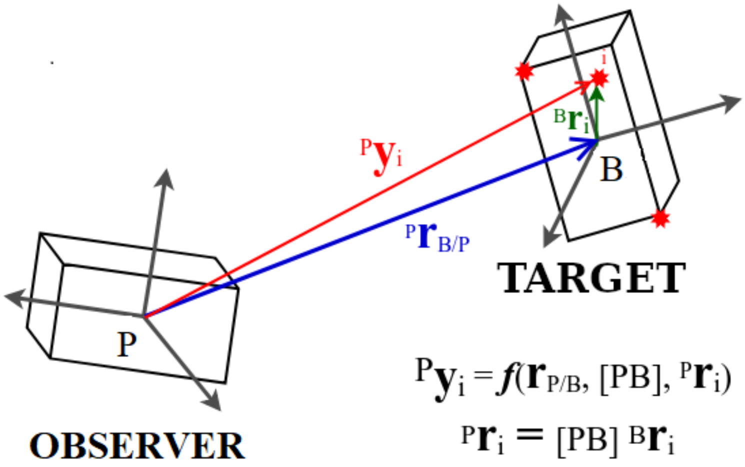
Along with importance weighting, any particle filter implementation requires some amount of computation to keep the particle hypotheses in high probability regions of the posterior. In order to accomplish this simply, we use Low Variance Resampling to resample our particle hypotheses whenever the number of effective particles (calculated with the sum of the squared particle weights) falls below a user-specified threshold.[19]

SIMULATIONS

This section describes the simulation used to provide an initial demonstration of the feasibility of RFS filtering for spacecraft RPO including the problem setup, system dynamics, and measurement model.

Problem Setup

Figure 2 depicts the spacecraft RPO scenario to be studied and defines relevant coordinate frames and vectors. In general, the observer spacecraft is taking measurements of features located on a target which could be a cooperative or non-cooperative spacecraft, or a small body such as an asteroid. These features would typically be “optically significant” features such as edges or corners of a solar panel, or craters on an asteroid.



$^{P}\mathbf{y}_i$
$^{B}\mathbf{r}_i$
i
B
$^{P}\mathbf{r}_{B/P}$
TARGET
P
OBSERVER
$^{P}\mathbf{y}_i = f(\mathbf{r}_{P/B}, [PB], {}^{P}\mathbf{r}_i)$
$^{P}\mathbf{r}_i = [PB]\, {}^{B}\mathbf{r}_i$

Figure 2: Relationship between observer and target coordinate frames and vector definitions.

The target has a body-fixed frame $\{B\}$ which we have defined to be the same as the Hill frame used to later define the relative orbital dynamics. The observer has a primary body-fixed frame $\{P\}$, which is defined to be the same as the frame in which measurements are taken in order to simplify the measurement equations. Various features are located in the rigid body frame $\{B\}$ – the ith feature located on the target can be located by a vector from the origin of the B frame to the feature.

A measurement of the ith feature is obtained in the $\{P\}$ frame. Each measurement $\mathbf{y}_i$ 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 [PB].

In order to propagate the dynamics, the state of interest is defined to be $\mathbf{x} = [\mathbf{r}_{\mathbf{P/B}}, \dot{\mathbf{r}}_{\mathbf{P/B}}, \boldsymbol{\sigma}]^T$, where $\mathbf{r}_{\mathbf{P/B}}$ is 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 $\boldsymbol{\sigma}$ contains the Modified Rodrigues Parameters required to describe the rotation [PB] between a vector in {P} and a vector in {B}.

The output of the filter is an estimate of the position, velocity, and orientation of the observer relative to the target represented by $\mathbf{r_{B/P}}$, $\dot{\mathbf{r}}_{\mathbf{B/P}}$ and a rotation matrix [BP] = [PB]^T which is calculated from the Modified Rodrigues Parameters.

System Dynamics

The target body is assumed to be in a circular orbit about the Earth. The observer is operating in close proximity to the target. Therefore, the dynamics of the observer relative to the target can be described by the Clohessy-Wiltshire (CW) equations.[20]

In the Hill frame $\{B\}$, the elements of $\mathbf{r_{P/B}}$ are defined by the following dynamics:

$$\begin{aligned} \ddot{x} &= 3n^2x + 2n\dot{y} \\ \ddot{y} &= -2n\dot{x} \\ \ddot{z} &= -n^2z \end{aligned} \tag{23}$$

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

To simplify the dynamic equations needed to calculate equations 18 and 20, the motion is constrained such that the attitude of the rigid-body target is fixed in the Hill frame, similar to a nadir pointing spacecraft. We also assume that the observer is perfectly tracking the target with its sensor pointed at the target, which is a reasonable assumption given an appropriate control system for pointing. This negates the need to estimate angular velocity, since the angular velocity of the observer body frame with respect to the Hill body frame is defined as:

$$\boldsymbol{\omega} = \frac{\mathbf{r} \times \dot{\mathbf{r}}}{|\mathbf{r}|^2} \tag{24}$$

The relative attitude of the observer and target is expressed using Modified Rodrigues Parameters (MRPs) because they only have one easily avoidable singularity. The MRPs obey the kinematic differential equation:[21]

$$\dot{\boldsymbol{\sigma}} = \frac{1}{4}\left[(1 - |\boldsymbol{\sigma}|^2)[I_{3\times3}] + 2[\tilde{\boldsymbol{\sigma}}] + \boldsymbol{\sigma}\boldsymbol{\sigma}^T\right]\boldsymbol{\omega} \tag{25}$$

where $[\tilde{\boldsymbol{\sigma}}]$ is the skew symmetric matrix composed of the elements of $\boldsymbol{\sigma}$. Equations 23 and 25 dictate the dynamics used for calculating equations 18 and 20.

The target is assumed to be spherical with a radius of 50 meters and covered with 20 randomly distributed features on its surface. Note that though the true features are constrained to be on the surface of a sphere, this information is not given directly to the filter. A spherical shape was chosen to simplify determining whether or not a simulated feature is visible to the sensor.

Though the CW equations are linear and have an analytical solution, they are integrated numerically along with the MRPs 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.

Measurement Model

In this scenario, the observer spacecraft takes measurements of the target spacecraft with a flash LIDAR. We simulate noisy flash LIDAR measurements from the truth trajectory to feed to the filter, as well as extraneous measurements which do not originate from features on the target. Although the filter is capable of handling missed detections, this was not simulated for this study.

In order to focus on the implementation of the filter, it is assumed that features are obtained 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[22,23] or LIDAR data.[24] Thus, each measurement of a feature consists of angular position in pixel coordinates and a range. Furthermore, the features 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 using a pinhole projection for the pixel coordinates and the Euclidean distance from the optical center to the feature for the range. For the camera projection the camera frame is defined where the x and y axes are aligned with the rows and columns of the detector and the third axis points along the boresight of the optics. Pixel coordinates are defined in the camera frame as $\{u, v, w\}$. The measurements passed to the filter are thus $\mathbf{y} = [u \quad v \quad \rho]$:

$$\tilde{x} = \begin{bmatrix} u \\ v \end{bmatrix} \tag{26}$$

$$\rho = \|\mathbf{X} - \mathbf{X}_c\| \tag{27}$$

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 $\{\mathrm{B}\}$.

To calculate $\tilde{x}$ a pinhole camera projection is defined with the following standard equations:

$$\mathbf{x} = w \begin{bmatrix} \tilde{x} \\ 1 \end{bmatrix} = \begin{bmatrix} wu \\ wv \\ w \end{bmatrix} \tag{28}$$

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

$$\mathbf{x} = \begin{bmatrix} fm_u & s & P_u \\ 0 & fm_v & P_v \\ 0 & 0 & 1 \end{bmatrix} [R][[I_{3\times3}, -\mathbf{X}_c] \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} \tag{29}$$

where $[R]$ is the rotation matrix from the Hill frame to the camera frame, f is the focal length of the optics, s is the skew of the camera, P_u and P_v are the location in pixels of the optical center on the detector, and m_u and m_v are the inverse sizes of the pixels in each dimension.[25] Equations 26-28 dictate the measurement model used for calculating equation 19 and the observation matrix in equation 21.

Noise and uniformly distributed clutter are added after truth measurements are calculated. If a measurement with noise is outside the bounds of the model then it is disregarded. The constants used to simulate these data are included in Table 1. These values were chosen to produce an angle of view of approximately 14°, and closely mimic the specifications of the flash LIDAR onboard Raven, a module on the International Space Station which performs autonomous tracking of rendezvous targets.[26]

Table 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	[]
ρ_{max}	750	m

An example of the measurements being passed to the filter is shown in Figure 3. At each time step, 10 clutter measurements are appended to the true measurements. Note that this clutter is not persistent, i.e. it does not remain in physical space across time steps, however that could be simulated for future work.

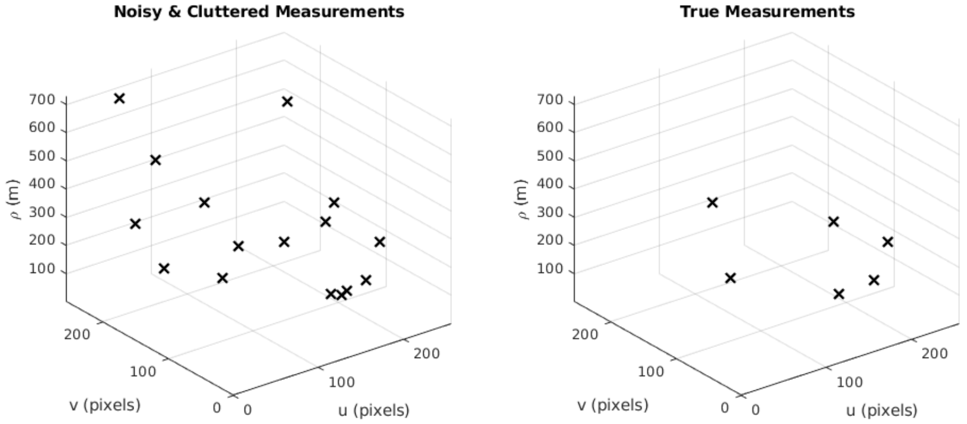


Figure 3: A simulated example of features extracted from flash LIDAR observations, given as measurements to the filter. The measurements on the left show the cluttered, noisy measurements given to the filter, while measurements on the right are the “true” measurements corresponding to the true location of features.

RESULTS

To test the filter, two test cases with relevant RPO dynamics were simulated. In both cases, the birth model $\gamma_k(x)$ for the PHD filter is given to be a Gaussian Mixture with means located at the true target locations and covariance of 1 meter. The birth model could also be chosen to be a uniform distribution such that all features are birthed with the same weight which does not depend on the location in space, giving the filter no *a priori* knowledge – simulations with a uniform birth model are in progress. 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.

For the particle filter, 100 particles are used. The pose hypotheses of the particles are initially distributed slightly offset from the true state with a small covariance in order to focus on primarily testing the mapping portion of the PHD filter but also allow for initial particle dispersion.

Case 1: Walking Safety Ellipse

The relative trajectory of the observer for the first case is shown in Figure 4. This trajectory brings the observer to within 30m of the surface of the target, and as far away as 650m.

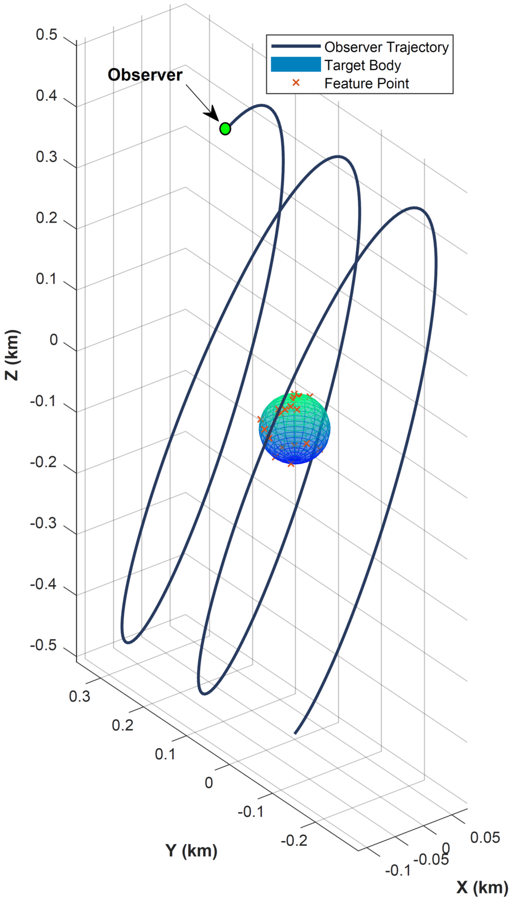


Figure 4: Observer trajectory in the Hill frame for Case 1. Red x's on the target body indicate the location of features to be estimated. Trajectory shown is over 15,000s.

Figure 5 shows the results of the PHD filter for the highest weight particle at 4 selected time steps. The red x's show the true feature locations on the target body, and the blue dots show the features that are estimated by the PHD filter. 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.

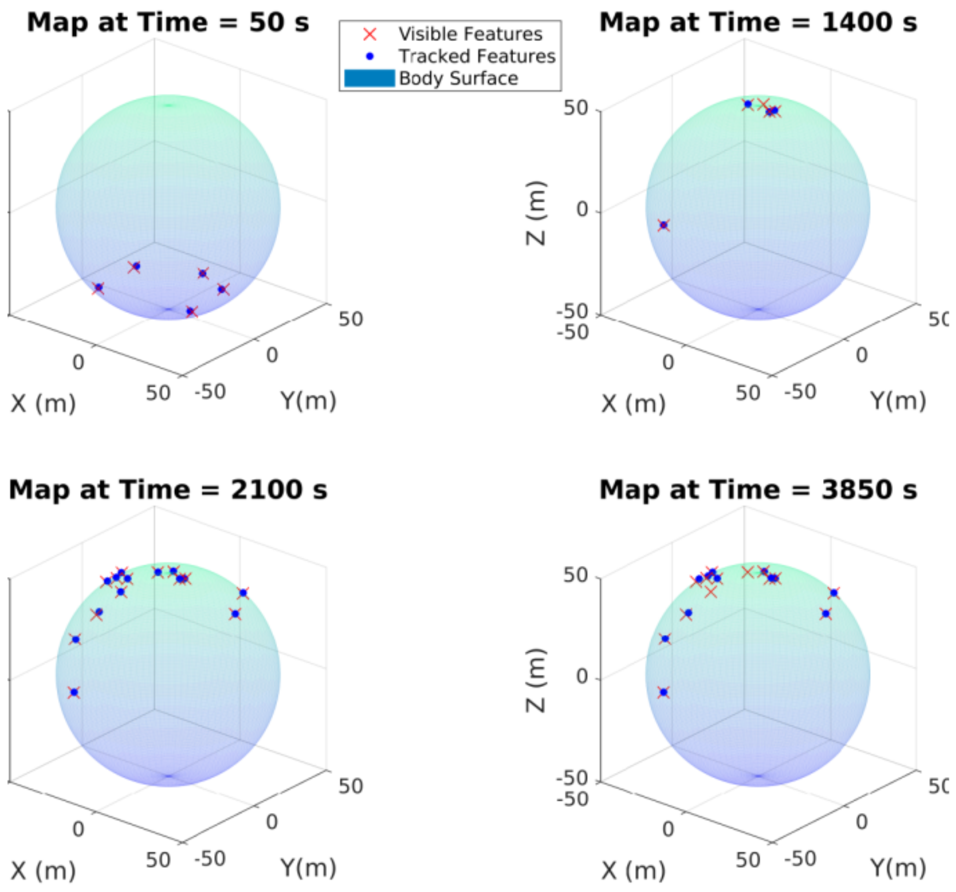


Figure 5: Sequential snapshots of the estimated feature map from the PHD filter over time for Case 1; the map shown is from the particle with the highest weight at that time step.

Figure 6 depicts the number of true features visible to the observer compared to the number of features obtained by the PHD filter. For a well-tuned filter, these two curves should be close over time. This figure shows that the estimated number of features closely tracks the truth. Additional features could be pruned out with further tuning, depending on the weight threshold chosen. The tuning chosen here was to favor more features in order to maintain a non-zero number of features at all time, as the filter currently discards features after they are no longer being tracked.

Figure 7 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 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

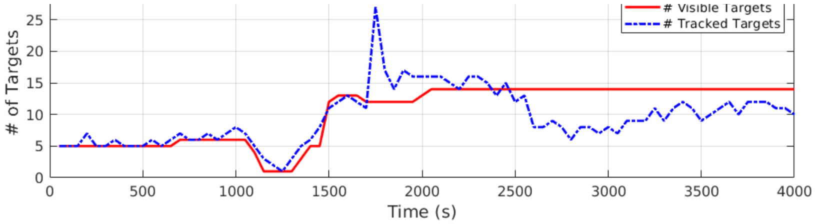


Figure 6: Comparison between the true number of visible features vs. how many features the PHD filter has estimated for Case 1.

in relative velocity, and generally less than one degree of error in relative attitude.

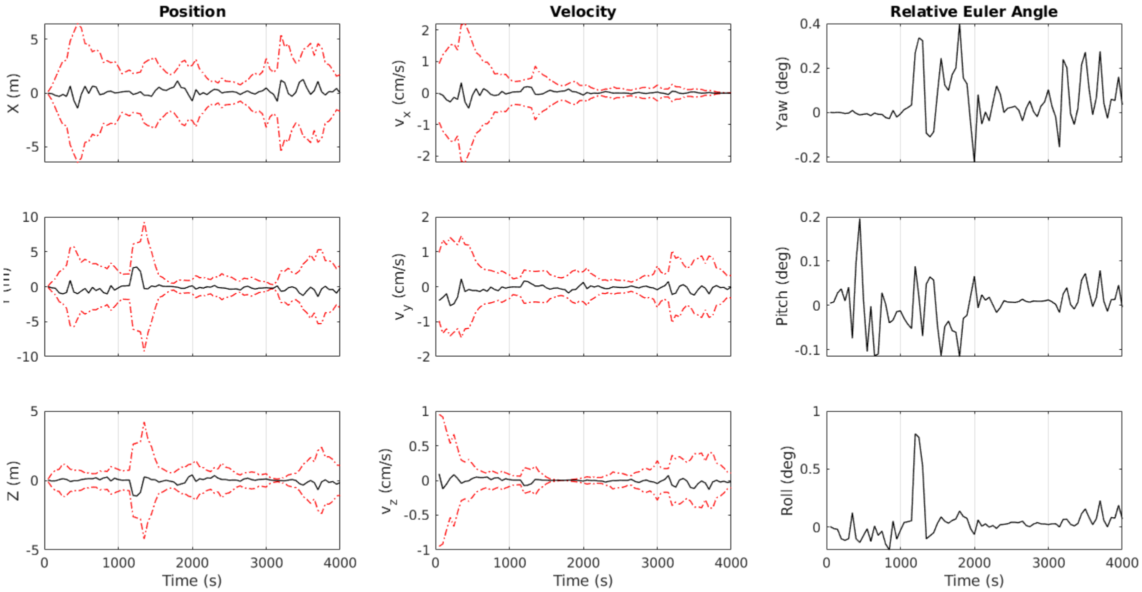


Figure 7: Comparison between sample covariance 3σ bounds vs. the position and velocity errors for Case 1. Note that there is no sample covariance for the Euler angles, as these have been converted from MRPs, where the concept of a sample covariance is nonexistent.

Case 2: Elliptical Orbit

Case 2 is a repeating elliptical orbit, shown in Figure 8. This case was chosen to see how the filter responds to periodic map information, as well as how the filter performs over a longer period of time.

Figure 9 shows the results of the PHD filter mapping step for Case 2. This case was tuned to be slightly less strict about allowing features to be accepted as “real.” 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 10 depicts the number of true features visible to the observer compared to the number of features obtained by the PHD filter. The periodic nature of the orbit can be seen in these results. Additionally, the less strict pruning of the features is seen as the filter consistently overestimates

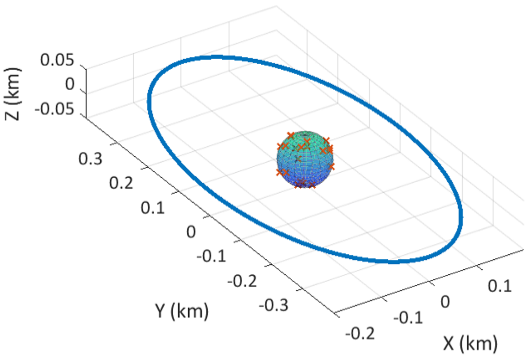
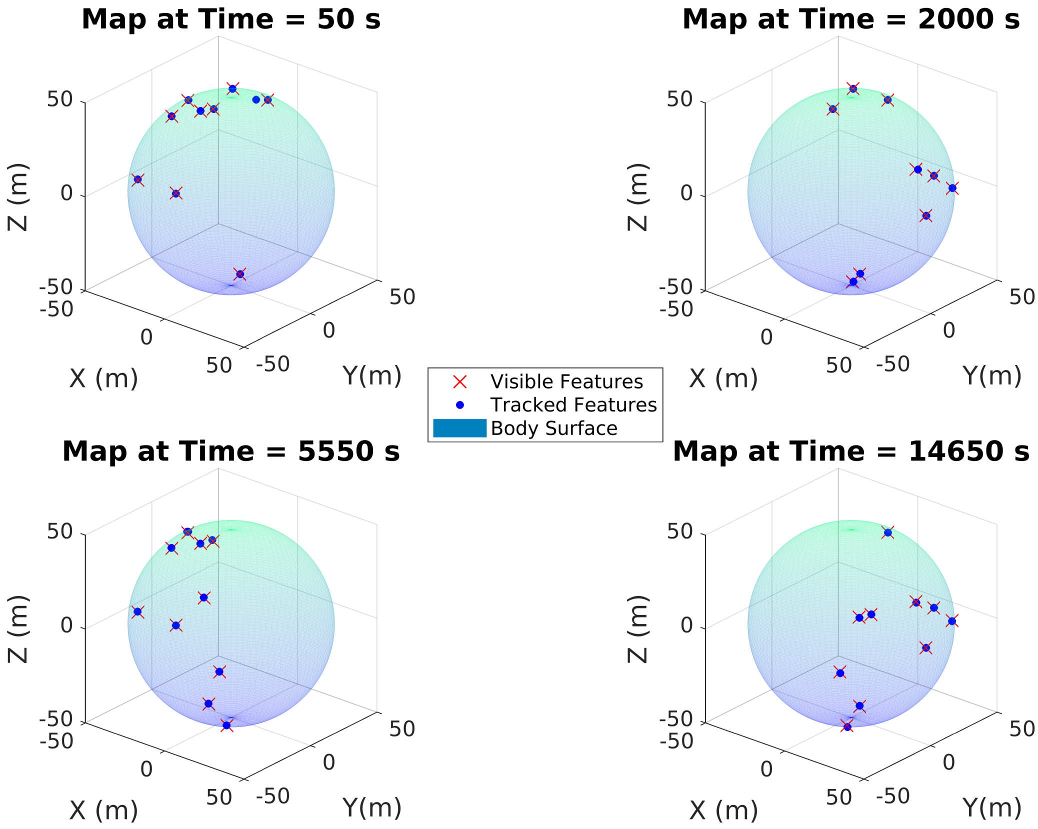


Figure 8: Snapshot of the observer trajectory in the Hill frame for Case 2. Red x's on the target body indicate the location of features to be estimated.


Map at Time = 50 s

Map at Time = 2000 s

Map at Time = 5550 s

Map at Time = 14650 s

Figure 9: Sequential snapshots of the estimated feature map from the PHD filter over time for Case 2; the map shown is from the particle with the highest weight at that time step.

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.[16]

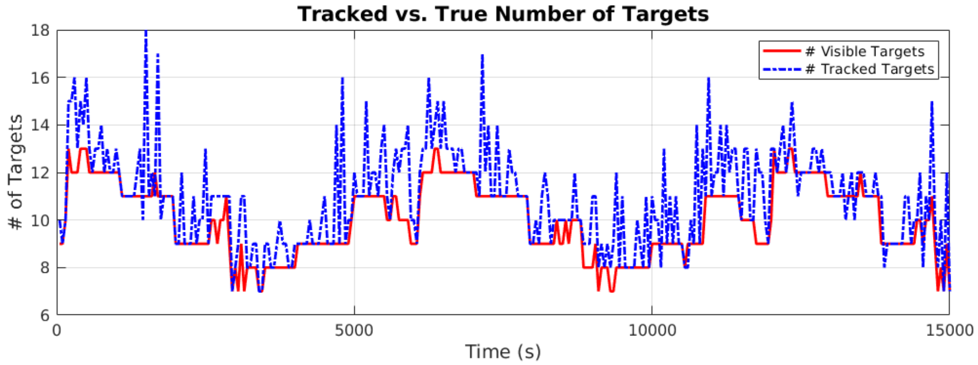


Figure 10: Comparison between the true number of visible features vs. how many features the PHD filter has estimated for Case 2.

Figure 11 shows the position and velocity errors compared to the sample covariance 3σ bounds of all the particle poses. Again, the black line shows the difference between the true and estimated pose, and the red dashed line shows the 3 σ bounds of the 100 pose hypotheses. The periodic nature of the orbit is seen in the covariance bounds. In some dimensions, the filter appears to converge on the correct pose as time progresses, showing that the particle filter approach has potential for good pose estimation.

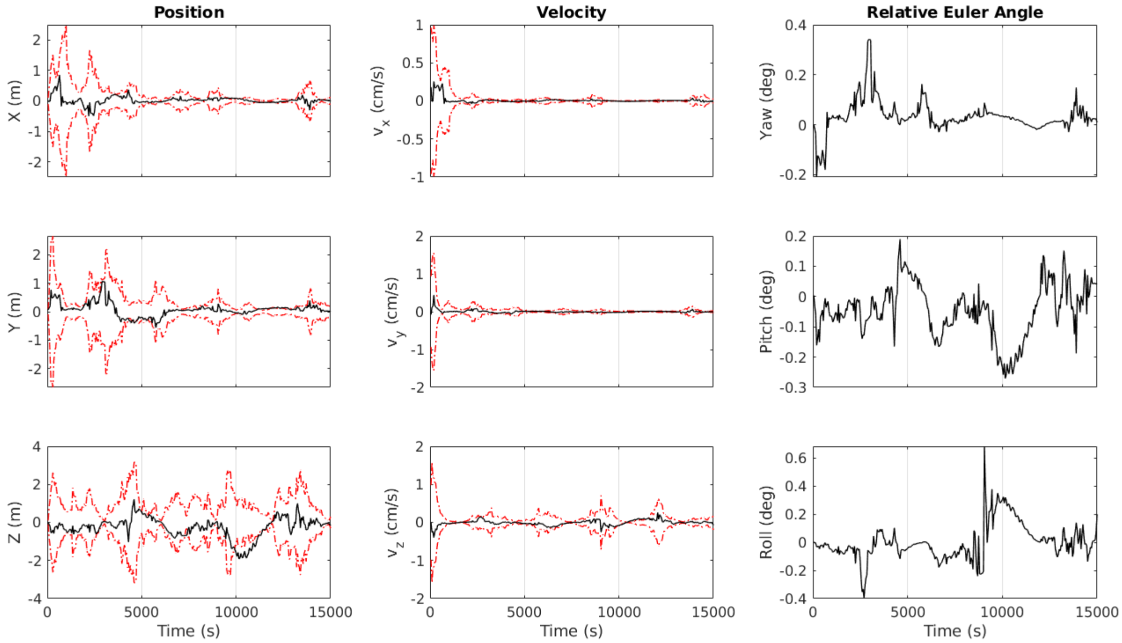


Figure 11: Comparison between sample covariance 3σ bounds vs. the estimate error for Case 2.

CONCLUSION

Our simulations have shown that RFS-based filters are feasible for performing relative navigation and pose estimation for RPO. The RFS formulation incorporates realistic dynamics and measurement scenarios directly into the mathematics. Our formulation required limited to no *a priori* knowledge of the environment, and no pre-filtering data association was required to match state estimates to measurements. Further work is needed to analyze filter robustness to initial conditions as well as various levels of initial uncertainty. Since the particle filter for pose estimation is susceptible to divergence, future work is also needed to improve or replace the particle filter.

ACKNOWLEDGMENTS

The authors would like to acknowledge the NASA Goddard Space Flight Center Internal Research and Development (IRAD) program for funding this work. Additional thanks to Dr. Martin Adams and Felipe Inostroza at the Universidad de Chile for their advice and recommendations for this work. Thanks also to members of the Satellite Servicing Projects Division at NASA Goddard Spaceflight Center for valuable conversations and the opportunity to observe a real-life spacecraft RPO.

REFERENCES

[1] P. Besl and N. McKay, “Method for Registration of 3-D Shapes,” *Proceedings of SPIE*, Vol. 1611, No. 1, 1992, pp. 586–606.

[2] Q.-X. Huang and D. Anguelov, “High Quality Pose Estimation By Aligning Multiple Scans to a Latent Map,” *IEEE International Conference on Robotics and Automation*, 2010, pp. 1353–1360.

[3] B. Naasz, J. V. Eepoel, S. Queen, C. M. Southward, and J. Hannah, “Flight Results from the HST SM4 Relative Navigation Sensor System,” *Proceedings in Advances in the Astronautical Sciences*, Vol. 137, 2010, pp. 723–744.

[4] M. Strube, R. Henry, E. Skelton, J. V. Eepoel, N. Gill, and R. McKenna, “Raven: An On-Orbit Relative Navigation Demonstration Using International Space Station Visiting Vehicles,” *Proceedings in Advances in the Astronautical Sciences*, Vol. 154, 2015, pp. 659–672.

[5] N. Dhiman, D. Deodhare, and D. Khemani, “Where Am I? Creating Spatial Awareness in Unmanned Ground Robots Using SLAM: A Survey,” *Sadhana Academy Proceedings*, Vol. 40, No. 5, 2015, pp. 1385–1433.

[6] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, “Simultaneous Localization and Mapping: A Survey of Current Trends in Autonomous Driving,” *IEEE Transactions on Intelligent Vehicles*, Vol. 2, No. 3, 2017, pp. 194–220.

[7] J. Mullane, B.-N. Vo, M. Adams, and B.-T. Vo, *Random Finite Sets for Robot Mapping and SLAM: New Concepts in Autonomous Robotic Map Representations*. Berlin: Springer-Verlag, 2011.

[8] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, “FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem,” *Proceedings of the National Conference on Artificial Intelligence*, Vol. 18, 2002, pp. 593–598.

[9] E. Eade and T. Drummond, “Scalable Monocular SLAM,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 1, 2006, pp. 469–476.

[10] S. Augenstein, “Monocular Pose and Shape Estimation of Moving Targets for Autonomous Rendezvous and Docking,” *Doctoral Thesis, Stanford University*, 2011.

[11] A. Sonnenburg, M. Tkocz, and K. Janschek, “EKF-SLAM Based Approach for Spacecraft Rendezvous Navigation with Unknown Target Spacecraft,” *International Federation of Automatic Control Proceedings*, Vol. 43, No. 15, 2010, pp. 339–344.

[12] C. Cocaud and T. Kubota, “SLAM-Based Navigation Scheme for Pinpoint Landing on Small Celestial Body,” *Advanced Robotics*, Vol. 26, No. 15, 2012, pp. 1747–1770.

[13] R. Mahler, “Multitarget Bayes Filtering via First-Order Multitarget Moments,” *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 39, No. 4, 2003, pp. 1152–1178.

[14] R. Mahler, “A Survey of PHD Filter and CPHD Filter Implementations,” *Proceedings of SPIE*, Vol. 6567, No. 1, 2007, pp. 0–12.

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

[16] B.-N. Vo and W.-K. Ma, “The Gaussian Mixture Probability Hypothesis Density Filter,” *IEEE Transactions on Signal Processing*, Vol. 54, No. 11, 2006, pp. 4091–4104.

[17] K. Leung, F. Inostroza, and M. Adams, “Multifeature-Based Importance Weighting for the PHD SLAM Filter,” *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 52, No. 6, 2016, pp. 2697–2714.

[18] C. S. Lee, D. Clark, and J. Salvi, “SLAM With Dynamic Targets via Single-Cluster PHD Filtering,” *IEEE Selected Topics in Signal Processing*, Vol. 7, No. 3, 2013, pp. 543–552.

[19] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Boston, MA: MIT Press, 2005.

[20] D. Vallado and W. McClain, *Fundamentals of Astrodynamics and Applications*. El Segundo, CA: Microcosm Press, 2nd ed., 2001.

[21] H. Schaub, *Analytical Mechanics of Space Systems*. Reston, VA: AIAA, 2nd ed., 2009.

[22] D. Lowe, “Object Recognition from Local Scale-Invariant Features,” *International Conference on Computer Vision*, Vol. 2, 1999, pp. 1150–1158.

[23] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, “Speeded-Up Robust Features,” *Computer Vision and Image Understanding*, Vol. 110, No. 3, 2008, pp. 346–359.

[24] M. .Jayendra-Lakshman and V. Devarajan, “A New Feature Descriptor for LIDAR Image Matching,” *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. II-2/W1, 2013, pp. 157–162.

[25] J. Christian, “Fundamentals of Vision Based Navigation,” *NASA Engineering and Safety Center Seminars*, 2014.

[26] M. Strube, R. Henry, E. Skelton, J. V. Eepoel, N. Gill, and R. McKenna, “Raven: An On-Orbit Relative Navigation Demonstration Using International Space Station Visiting Vehicles,” *AAS GNC Conference Proceedings*, Vol. 15, No. 111, 2015.