

AstroSLAM: Autonomous monocular navigation in the vicinity of a celestial small body—Theory and experiments

The International Journal of
Robotics Research
2024, Vol. 43(11) 1770–1808
© The Author(s) 2024
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/02783649241234367
journals.sagepub.com/home/ijr



Mehregan Dor¹ , Travis Driver¹, Kenneth Getzandanner² and Panagiotis Tsiotras¹

Abstract

We propose *AstroSLAM*, a standalone vision-based solution for autonomous online navigation around an unknown celestial target small body. *AstroSLAM* is predicated on the formulation of the SLAM problem as an incrementally growing factor graph, facilitated by the use of the GTSAM library and the iSAM2 engine. By combining sensor fusion with orbital motion priors, we achieve improved performance over a baseline SLAM solution and outperform state-of-the-art methods predicated on pre-integrated inertial measurement unit factors. We incorporate orbital motion constraints into the factor graph by devising a novel relative dynamics—*RelDyn*—factor, which links the relative pose of the spacecraft to the problem of predicting trajectories stemming from the motion of the spacecraft in the vicinity of the small body. We demonstrate *AstroSLAM*'s performance and compare against the state-of-the-art methods using both real legacy mission imagery and trajectory data courtesy of NASA's Planetary Data System, as well as real in-lab imagery data produced on a 3 degree-of-freedom spacecraft simulator test-bed.

Keywords

Navigation, SLAM, sensor fusion, small body, relative dynamics, factor graph

Received 16 September 2022; Revised 04 November 2023; Accepted 24 January 2024

Senior Editor: José Luis Blanco Caraco

Associate Editor: David Rosen

1. Introduction

Precise relative navigation techniques, incorporating increased levels of autonomy, will be a key enabling element of future small-body orbiter missions (Christian and Lightsey, 2012; Delpech et al., 2015; Nesnas et al., 2021). Firstly, good navigation can inform safe and efficient path planning, control execution, and maneuvering (Bhaskaran and Kennedy, 2014). In near-small-body deep space missions, achieving fuel-efficiency during non-critical maneuvers and guaranteeing execution of safety-critical maneuvers requires precise knowledge of the relative position and orientation of the spacecraft with respect to the small-body. Secondly, precise navigation situates the acquired science data. Indeed, scientists and mission planners design science acquisition phases based on the expected scientific value of instrument data acquired at predetermined times, on specific orbits and with specific spacecraft orientations (Miller, 2002). Thirdly, precise navigation facilitates the detailed mapping and shape reconstruction of the target small-body, since good knowledge of the spacecraft relative position and orientation with respect to the target, as well as a good knowledge of the Sun light direction, are crucial in commonly used shape reconstruction solutions (Gaskell et al., 2008). Finally, good estimates of the spacecraft state enable precise characterization of the target

small-body's spin state, mass moment values and gravitational model (Miller, 2002).

In recent years, with ever-improving navigation solutions, space missions have successfully performed daring firsts in navigation around small celestial bodies. Orbiter Near-Earth Asteroid Rendezvous (NEAR) Shoemaker's controlled asteroid touchdown (1996) (Prockter et al., 2002), Hayabusa I & II's touchdown and successful sample return (2003) (Terui et al., 2020; Yoshikawa et al., 2015), Dawn's orbiting of two celestial bodies in a single mission (2007) (Konopliv et al., 2014) and the recent Origins, Spectral Interpretation, Resource Identification, Security—Regolith Explorer's (OSIRIS-REx's) Touch-and-Go (TAG) operation leveraging Natural Feature Tracking (NFT) relying on high navigation solution accuracy during descent (Berry et al., 2022; Laurretta 2021), are only few of the most notable feats accomplished thanks to autonomous navigation.

¹Georgia Institute of Technology, Atlanta, GA, USA

²NASA Goddard Space Flight Center, Greenbelt, MD, USA

Corresponding author:

Mehregan Dor, Dynamics and Control Systems Laboratory, Georgia Institute of Technology, Atlanta, GA 30332, USA.

Email: mehregan.dor@gatech.edu

The OSIRIS-REx proximity operations at the near-Earth asteroid (101955) Bennu, in particular, pushed the boundaries of what can be accomplished using primarily ground-in-the-loop navigation techniques (Antreasian et al., 2022). Like similar small-body missions, OSIRIS-REx’s proximity to Bennu, as well as the asteroid’s small size and low gravitational attraction relative to perturbing forces, drove the need for frequent and timely navigation updates in order to achieve mission objectives (Leonard et al., 2022b). These updates drove operation complexity and cadence, challenging the flight team, and heavily utilizing Deep Space Network (DSN) assets. OSIRIS-REx proximity operations navigation and TAG also relied on detailed local and global topographic maps constructed from image and LiDAR data, with ground sample distances ranging from 75 cm down to 8 mm (Barnouin et al., 2020). Building these maps required dedicated, months-long observation and data collection campaigns and a substantial amount of effort by the Altimetry Working Group on the ground, as well as multiple iterations with the navigation team (Leonard et al., 2020). It also required downlinking tens of thousands of images and hundreds of gigabytes of LiDAR data from the spacecraft through the DSN.

It is recognized that the high-risk nature of missions in the proximity of small celestial bodies, along with a lack of autonomy in current mission procedures, severely limits the possibilities in mission design (Starek et al., 2016). Indeed, ground-segment operators are intimately involved in all in-situ tasks, which ultimately rely on extensive human-in-the-loop verification, as well as ground-based computations for estimation, guidance, and control (Nesnas et al., 2021; Williams, 2002). In addition, long round-trip light times and severely limited bit-rate in communications render ground-in-the-loop processes extremely tedious. In tandem, we expect that the incorporation of autonomous capabilities has the potential to improve navigation performance and reduce operational complexity for future missions (Getzandanner et al., 2022; Nesnas et al., 2021).

In this paper, we build on our previous work (Dor et al., 2021), and present a holistic application of a factor graph-based incremental smoothing solution for monocular visual SLAM around a small-body, referred to as AstroSLAM. Using the GTSAM library (Dellaert and Kaess, 2017) and the iSAM2 solver (Kaess et al., 2012), we perform multi-sensor fusion and constraint enforcement on-the-fly. We incorporate inertial attitude measurements from a star tracker and Earth-relative DSN radiometric data to obtain an initial pose prior, and then we leverage image-based measurements and dynamic motion constraints for subsequent trajectory estimation. Crucially, we incorporate the equations of the relative motion between the spacecraft and the small-body into the problem, given that these are strong odometric constraints. We do so by writing and implementing a new factor node called RelDyn, which encodes the constraints derived from the relative dynamics and kinematics.

In order to demonstrate the improvement in performance brought about by incorporating dynamics in the SLAM

problem, which usually suffers from incorrect data association in real-world applications, we tested AstroSLAM using actual 2D-2D and 2D-3D correspondence mappings from real-world imagery.

Specifically, we tested our algorithm on an image sequence from NASA’s Dawn mission, as well as on a sequence of realistic asteroid images produced in the lab. We compare the estimated solution against archived navigational data from the Dawn mission for the first demonstration and also compare the estimated solution to ground-truth data from an in-lab produce sequence. It is shown that AstroSLAM surpasses the state-of-the-art performance in both cases in terms of navigational error and velocity estimates.

1.1. Related work

Filter-based methods (Nakath et al., 2018), such as the Extended Kalman Filter (EKF), have traditionally been applied to perform on-the-fly multi-sensor fusion for precise navigation purposes. Bercovici and McMahon (2019b) proposed a Flash-LiDAR-based pose estimation and shape reconstruction approach, by solving a maximum likelihood estimation problem via particle-swarm optimization, followed by a least-squares filter providing measurements for the spacecraft position and orientation in the small-body frame coordinates. Other recent works in the field have established proof-of-concepts for online implementation of batch optimization and graph-based approaches for precise near-small-body navigation, like real-time SLAM. Notably, Nakath et al. (2020) presents an active SLAM framework which also employs Flash-LiDAR as the base measurement of the SLAM formulation, with sensor fusion of data from an inertial measurement unit and star tracker, tested with simulated data. However, the limited range of Flash-LiDAR instruments restricts the spacecraft’s orbit to unrealistically small radii, reducing the use scenarios to either navigation near very small bodies or to the touchdown phase for larger target small-bodies. For example, the OSIRIS-REx Guidance, Navigation, and Control (GNC) Flash-LiDAR, which is mentioned by both Nakath and Bercovici, has a reliable maximum range of approximately 1 km (Church et al., 2020; Leonard et al., 2022a). In contrast, an approach that uses long-range optical imagery, like the one we propose in this paper, enables detailed characterization of the small-body early in the approach phase of the mission, at which point knowledge about the target small-body may still be poor.

Several prior works have applied visual SLAM solutions for spacecraft relative navigation. However, much fewer works have directly applied visual SLAM to the small-body navigation problem. Among the most interesting works in this area, we note Cocaud and Kubota (2010), which leverages SURF (Bay et al., 2006) visual cues and range measurements, and Cocaud and Kubota (2012), which focuses on image feature-only formulation, and solves the relative pose estimation problem using a Rao-Blackwellized particle filter. However, the latter works only tested the algorithm on simulated imagery of asteroid Itokawa.

Additionally, particle filters are notoriously compute-intensive, and not directly amenable to on-the-fly implementation. Similarly, Baldini et al. (2018) implemented OpenSFM on simulated images of comet 67P/Churyumov-Gerasimenko, while Takeishi et al. (2015) performed a particle filter minimization of the observation error and used both simulated landmarks and SIFT (Lowe 2004) features extracted and tracked across a sequence of real images of a simple asteroid mock-up, with albeit unrealistic motion. Most recently, Villa et al. (2022) successfully implemented an autonomous navigation and dense reconstruction predicated on visual-only landmark observation batch optimization and stereophotogrammetry. However, the implied batch optimization methods require intensive on-board compute power, which is not easily amenable to on-the-fly autonomous navigation.

The SLAM problem can be assimilated to a discrete-time sequential state estimation and static scene mapping problem, and it is known that the inclusion of odometric constraints allows for some filtering to be worked into the SLAM solution. However, the ability of the odometric constraint to improve the SLAM solution depends directly on the validity of the motion model in the application case and on the uncertainties associated with perturbations and un-modeled effects.

We postulate that an odometric constraint should indeed be incorporated into the near-small-body monocular SLAM navigation problem, but that a simplistic motion model is not enough to validate the algorithm on real data. The odometric constraint should, instead, be based on a high-fidelity description of the dynamics of the spacecraft-small-body system. Indeed, the orbital motion of the spacecraft in the vicinity of a small-body can be modeled with very high fidelity, owing to careful and accurate modeling of the perturbing forces at play. Additionally, the orbital motion of a typical small-body in the solar system targeted for further probing is typically estimated with high precision due to tracking and Orbit Determination (OD) throughout a long period via ground observations, as well as during the approach phase via Optical Navigation (OpNav). It follows that an accurate and highly certain relative motion model between the spacecraft and the small-body can be derived and used in formulating a *strong odometric constraint*. Nevertheless, parameters that affect the description of the relative position of the spacecraft with respect to the small-body, such as the true size of the small-body, the relative distance to its center of mass, its gravitational potential, its spin state, or the forces on the spacecraft due to its albedo radiation, are not well-known a priori, and must be estimated in-situ. We have incorporated the estimation of the spin state, gravitational parameter, and center-of-mass position into our estimation procedure. We note that since our formulation is strictly for monocular SLAM without the ability to incorporate typical visual-inertial cues which allow to estimate the scale, the well-known scale ambiguity issue remains. As a simplifying step, we assume the scale to be known a priori, but discuss later the impact of this

assumption on the obtained trajectory and parameter solution. Future work will incorporate additional measurement modalities and algorithms to estimate the scale and other parameters, such as inertia ratios, on-the-fly in an autonomous fashion.

Modern SLAM solutions are predicated on the formulation of the estimation or smoothing problems using a *factor graph* (Dellaert, 2021). As such, several works have incorporated dynamics-derived constraints into the SLAM factor graph problem. Most works incorporate a constant velocity with white-noise odometric constraint (Anderson and Barfoot, 2015; Matsuzaki et al., 2000) or a linear dynamical model with white-noise odometric constraint (Anderson et al., 2015), modeled as a Gaussian process. These approaches are often referred to as Simultaneous Trajectory Estimation And Mapping (STEAM) and fall within the realm of batch optimization methods, which for large problems can be compute-intensive. Notably, Yan et al. (2017) provides an extension by transforming the batch STEAM optimization into an incremental method using efficient variable re-ordering at every optimization step, while still exploiting a continuous-time Gaussian process odometric constraint. In these works, the estimated Gaussian process provides a time-based support to evaluate the trajectory at any desired query time within the sampling interval. However, the accuracy of the solution is predicated on interpolation of the estimated state and covariance between the selected optimization times using the Gaussian process model in an iterative fashion. Therefore, it is required that the time steps for optimization be chosen close to each other, thus increasing computations. This disadvantage offsets the advantage of the use of a continuous Gaussian process-based method for the purpose of small-body navigation, since optimization times can be significantly spread apart along the trajectory around the small-body given the limited on-board resources in a real mission. Instead, we propose to use a high-fidelity dynamical model, paired with an on-manifold integration method, to obtain accurate predictions of the state with large time steps, thus reducing the density of the selected optimization points in time.

As opposed to the aforementioned works, where the factor encodes the error of the dynamic prediction by exploiting the solution to the piecewise-constant input locally linearized model, Xie et al. (2020) formulates the dynamics factor using the non-linear differential equation directly. It follows that the resulting factor graph has “non-state” variables, such as linear and angular accelerations, as well as wrenches. To avoid an underconstrained problem, these dynamics-related accelerations each need an individual measurement or prior factor. This solution, although simple and useful for reference trajectory planning, is unrealistic for the purpose of estimation, since real-world measured acceleration and wrench signals are generally fraught with high-frequency noise and poor signal-to-noise ratio.

The factor graph-based formulation using stereo SLAM and smoothing approach first implemented and tested on the

SPHERES platforms (Tweddle, 2013) is most closely related to our work. Indeed, Tweddle (2013) incorporated non-specific dynamic constraints into the problem by implementing a factor that captures the residual of the integrated equations of motion. It is noteworthy that in the latter work, the observing agent has no inertial motion and has a static viewing direction. Several works have since extended Tweddle's procedure, and have tackled the problems of trajectory smoothing, object surface mapping, and inertial parameter estimation using factor graph formulations in spacecraft relative navigation scenarios. Chiefly, we denote the works by Setterfield et al., starting with Setterfield et al. (2017), in which inertial odometric factors in the form of the pre-integrated inertial measurement unit (IMU) factors of Forster et al. (2016) and spacecraft star tracker measurements were incorporated into a factor graph to complement visual odometry measurements obtained from a 3D visual sensor data, albeit in the vicinity of a passive target object. Later, in Setterfield et al. (2018a), additional kinematic factors were introduced on top of the inertial odometric factors to allow for the motion of the target and to estimate the center of mass of the target, along with visual odometry factors introduced by virtue of tracking and mapping surface features points on the spinning target object. In the latter approach, since the position of the center-of-mass of the target object is an unknown variable connected to all the introduced kinematic factors, to avoid an explosion of complexity during optimization, the factor graph was separated into two parallel procedures, one incrementally solved at every time step involving the inertial factors and visual odometry factors, and another one solved in batch-style sporadically involving the kinematic rotation factors.

Finally, in Setterfield et al. (2018b), the results of the previous work were exploited to estimate the target body's angular velocity by incorporating additional kinematic constraints, and then to deduce the target principal moments of inertia by polehode analysis. All these procedures were successfully demonstrated on simulated and real-life data, including demonstration aboard the International Space Station using the SPHERES-VERTIGO and Astrobees platforms (Oestreich et al., 2021). Later, in Terán Espinoza (2021), a Bayes tree representation was exploited, along with message passing techniques, to improve sparsity in the graph structure and reduce the computational load incurred when solving for global parameters, such as the target's center-of-mass location. Since introducing such a parameter significantly increases the size of cliques (fully connected sub-graphs), we may solve for them through a secondary and separate batch optimization using the time history of angular velocity measurements produced from the underlying visual-inertial odometry process, as in Setterfield et al. (2018a).

Similarly, in our work, both the observing spacecraft and the target have inertial translational and rotational motion, and are subject to specific dynamics as a consequence of the driving forces. We capture these constraints in the form of

relative kinematics and relative dynamics yielding equations describing the relative motion between the spacecraft and the small-body. We have incorporated many of the ideas from Setterfield et al. (2018a) into our own, notably the kinematic rotation factor, allowing us to estimate the center-of-mass of the target small-body. Several deviations exist between our work and that of Setterfield et al. (2018a): (1) we opt for a dynamic modeling of the gravitational interactions instead of exploiting inertial pre-integrated IMU factors, with the rationale of this choice further detailed in Section 3. Another deviation is that we do not estimate the target body's principal moments of inertia (more specifically, its inertia ratios). As stated in Setterfield et al. (2018b), in scenarios where the target body's motion is torque free and is single-axis, the inertia ratios are not observable. This is indeed the case for many small-bodies (Russell and Raymond, 2012a; Scheeres, 2016). Rather than motion observation, most methods currently used for estimating the inertia ratios of the target small-body, such as space carving (Driver et al., 2020), stereophotoclinometry (Gaskell et al., 2008), or geometric representation as Bezier curves (Bercovici and McMahon, 2019a), are instead based on shape reconstruction, from which inertias may be inferred. For the scope of this work, we therefore make the simplifying assumption that the rotational motion of the target is in single-axis configuration. Future work will amend this to allow for multiple-axis rotation scenarios to also be included.

Several works incorporate some form of dynamics-based modeling in SLAM for the purpose of improving the baseline SLAM solution in near-small-body celestial navigation. For example, Delpech et al. (2015) propose to perform an EKF step integrating the inertial equations of motion in between steps involving bundle adjustment for SLAM, in an alternating fashion. Most notably, the work by Rathinam and Dempster (2017), which has important parallels to our own, incorporates orbital motion priors as factors into the SLAM smoothing problem factor graph directly, arguably for the purpose of estimating the relative pose with respect to a small-body. Nevertheless, there are several key differences and shortcomings with respect to our work, which we detail next.

The first difference lies in the fidelity and accuracy of the modeled spacecraft-small-body system dynamics with respect to the real-mission setting. In Rathinam and Dempster (2017) the inertial equations of motion of the spacecraft and of the small-body are implemented separately, whereby the spacecraft is subject to a massive central body gravitational force and the small-body is subject to only a zero-mean perturbation force. However, it is generally known that, due to the relatively small gravitational force of the small-body, the solar gravitational force and the solar radiation pressure forces intervene in a significant way to affect the trajectory of the spacecraft (Scheeres, 2016) around the small-body, and therefore these forces should not be neglected during modeling. This is true for most missions to small celestial bodies of interest in the solar system, especially smaller asteroids, such as Itokawa, which Rathinam and Dempster

(2017) specifically consider in their work as an illustrative example. In addition, Rathinam and Dempster (2017) omit to incorporate spacecraft control forces in the modeling, as well as any spacecraft inertial attitude and angular velocity measurements. Such a choice severely restricts the possible use cases of their formulation. We chose, instead, to model the *relative* kinematics and dynamics between the spacecraft and the small-body, predicated on the fact that in the SLAM problem, without added dynamics-related factors, the measurement is innately a function of the relative position and attitude of the camera (here the spacecraft) with respect to the static scene (here the rigid small-body surface) through the 3D point projection measurement function. It then suffices to propagate the relative position and orientation in the kinematics and dynamics. In addition, we also include spacecraft control forces, spacecraft attitude and angular velocity measurements and the above-mentioned Sun-related perturbing forces in our modeling framework.

The second difference stems from the approach to encode the chosen dynamics in factors to be included in the factor-graph. Specifically, Rathinam and Dempster (2017) encode the dynamics into two separate factors without any specific discussion about the identification or quantification of the disturbances on which the factors' residual error function is predicated. Yet, it is obvious from their formulation that the dynamics of the spacecraft and of the small-body are actually coupled through the noise. Indeed, there is a link through the dynamics between the disturbance considered in the inertial motion model of the small-body and the small-body gravitational force affecting the spacecraft, which is itself a function of the relative position vector between the spacecraft and the small-body. Hence, we postulate that the equations of motion of the spacecraft-small-body system cannot be decoupled and, consequently, should not be encoded by two separate factors. As the quantification of the noise sources and the way they enter the equations of motion impact the feasibility of smoothing, based on the concept of *smoothability* (Gelb et al., 1974), we expose the effect of these noise sources in a full development of the stochastic differential equations in Section 4. We discuss the factor error function in Section 4.3 and assess their impact on smoothability in Section 4.7.

The third difference pertains to our use of a “front-end” system processing actual image data towards the SLAM solution, and their lack thereof. In Rathinam and Dempster (2017), randomly sampled idealized 3D points from a shape model are used to simulate camera feature point measurements. However, this simulation fails to mimic real-life effects encountered in small-body surveying missions. These effects include, among others, small-body surface shadowing, landmark visibility restricted by view-cone or occlusions, local image quality variations and image blurring, etc, which all affect the number of tracked surface features, the reliability of the matched features and, in turn, the overall error in the SLAM solution. Crucially, the use of a real front-end system in our work allows us to quantify the real improvement brought

about by the incorporation of motion priors. Indeed, the ability to appropriately match features directly depends on knowledge of the relative pose, which, in turn, is improved by the use of motion priors, a result we detail in Section 6.1.

Finally, we use real-mission imagery and trajectory data to validate our overall system, demonstrating impressive performance, further supporting our choice to model the forces missing from Rathinam and Dempster (2017) and to use dynamical vehicle modeling, instead of implementing inertial sensor-based constraints as proposed by Setterfield et al. (2017, 2018a).

1.2. Contributions

In contrast to the traditional ground-in-the-loop mindset, this paper proposes AstroSLAM, a viable autonomous navigation approach for near-small-body operations based on small-body imagery collected by the spacecraft on-board cameras and on inertial sensor data, such as an inertial star tracker and a rate gyro.

The assumptions made in the modeling in the works by Setterfield et al. (2017, 2018a), although well-suited for short-duration spacecraft proximity operations navigation, are not ideal for small-body circumnavigation, as we further argue in Section 3. We have consequently adapted Setterfield's kinematic rotation factor—a simple motion model—and incorporated it into our procedure. We call this adapted factor RelKin, since it models the relative kinematics of the spacecraft-small-body pair, and works in conjunction with the RelDyn factor, which we discuss in Section 4. Since the IMU preintegration factors considered in Setterfield et al. (2017, 2018a) are not well-suited for our scenario we instead leverage explicit vehicle dynamical modeling predicated on the gravitational interaction. We further discuss this discrepancy with the work in Setterfield et al. (2017, 2018a) in Section 3. We also compare our RelDyn results to the original procedure from Setterfield et al. (2017) running on the same datasets. This comparison demonstrates improved navigation performance, further supporting our approach, as demonstrated experimentally in Section 7.

Specifically, the contributions of this work are the following: (a) we adapt the kinematic factor described in Setterfield et al. (2018a) to the situation where the target object undergoes significant motion between time instances; (b) we model and incorporate orbital motion constraints specific to the small-body circumnavigation problem to make the SLAM solution more robust to outliers and drift in the form of the RelDyn factor, predicated on the relative motion constraints, as opposed to the typical inertial approach; (c) we demonstrate our algorithm on real imagery obtained in-situ from a previously flown small-body orbiter mission, as well as using imagery and trajectory data produced in our experimental lab facility; such a theoretical procedure and experimental validation for the small-body navigation problem is, to our knowledge, a first among works concerning near-small-body SLAM algorithms and constitutes a significant novelty of this work.

The paper is organized as follows: [Section 2](#) introduces the problem and summarizes the notation used. [Section 3](#) discusses in more detail why the use of pre-integrated IMU measurements is inadequate for our purposes, and [Section 4](#) focuses on the novel relative orbital odometric constraint calculations utilized in the factor graph estimation engine. Details of the technical approach in terms of implementation are given in [Section 5](#). [Section 6.1](#) validates the algorithm against real asteroid imagery from prior NASA missions, and [Section 7](#) presents the results from experiments carried out in a realistic laboratory facility where the ground truth relative pose is available. Lastly, [Section 8](#) provides some conclusions along with possible avenues of future work.

2. Problem statement

In this section, we discuss the relevant theory, and establish the problem statement pertaining to the asteroid relative navigation problem incorporating monocular SLAM, an appropriate motion prior and sensor fusion.

Firstly, we summarize the notation conventions used throughout the paper in [Section 2.1](#). Secondly, we contextualize the defined notation within the problem of a spacecraft navigating around an asteroid in [Section 2.2](#).

2.1. Notation

Given the affine space $(\mathbb{E}^3, \mathbb{R}^3)$, the translation vector between any two points $X, Y \in \mathbb{E}^3$ is denoted $\mathbf{r}_{YX} \triangleq (Y - X) \in \mathbb{R}^3$, read “from X to Y .” Any frame $\mathcal{X}$ is a tuple $(X, \{\vec{\mathbf{x}}_i\}_{i=1}^3)$ where the point $X \in \mathbb{E}^3$ denotes the origin of the frame and the set of unit directions $\{\vec{\mathbf{x}}_i\}_{i=1}^3$, where $\vec{\mathbf{x}}_i \in \mathbb{S}^2$, $i = 1, \dots, 3$, constitutes the right-handed orthonormal basis of the frame.

The expression of any vector $\mathbf{v} \in \mathbb{R}^3$ in a given frame $\mathcal{X} = (\cdot, \{\vec{\mathbf{x}}_i\}_{i=1}^3)$ is denoted by $\mathbf{v}^{\mathcal{X}} \triangleq [\mathbf{v} \cdot \vec{\mathbf{x}}_1 \quad \mathbf{v} \cdot \vec{\mathbf{x}}_2 \quad \mathbf{v} \cdot \vec{\mathbf{x}}_3]^\top \in \mathbb{R}^3$. For any two frames $\mathcal{X}$ and $\mathcal{Y}$, we denote the rotation $R_{\mathcal{X}\mathcal{Y}} \triangleq [\vec{\mathbf{y}}_1^{\mathcal{X}} \quad \vec{\mathbf{y}}_2^{\mathcal{X}} \quad \vec{\mathbf{y}}_3^{\mathcal{X}}] \in \text{SO}(3)$ such that $\mathbf{v}^{\mathcal{X}} = R_{\mathcal{X}\mathcal{Y}} \mathbf{v}^{\mathcal{Y}}$ for any $\mathbf{v} \in \mathbb{R}^3$. This notation is consistent with the composition rule $R_{\mathcal{X}\mathcal{Z}} = R_{\mathcal{X}\mathcal{Y}} R_{\mathcal{Y}\mathcal{Z}}$ for any three frames $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$. Given a coordinate vector $\mathbf{v}^{\mathcal{X}} \in \mathbb{R}^3$ expressed in frame $\mathcal{X}$, we denote the corresponding homogeneous coordinates $\mathbf{v}^{\mathcal{X}} \triangleq [(\mathbf{v}^{\mathcal{X}})^\top \quad 1]^\top \in \mathbb{P}^3$. For any two frames $\mathcal{X} = (X, \{\vec{\mathbf{x}}_i\}_{i=1}^3)$ and $\mathcal{Y} = (Y, \{\vec{\mathbf{y}}_i\}_{i=1}^3)$, we denote the homogeneous transformation $T_{\mathcal{X}\mathcal{Y}} \in \text{SE}(3)$ by

$$T_{\mathcal{X}\mathcal{Y}} \triangleq \begin{bmatrix} R_{\mathcal{X}\mathcal{Y}} & \mathbf{r}_{YX}^{\mathcal{X}} \\ 0_{1 \times 3} & 1 \end{bmatrix}, \quad (1)$$

such that $\mathbf{r}_{PX}^{\mathcal{X}} = T_{\mathcal{X}\mathcal{Y}} \mathbf{r}_{PY}^{\mathcal{Y}}$ for any point $P \in \mathbb{E}^3$. This notation is consistent with the composition rule $T_{\mathcal{X}\mathcal{Z}} = T_{\mathcal{X}\mathcal{Y}} T_{\mathcal{Y}\mathcal{Z}}$ for any three frames $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$.

Given a Lie algebra $\mathfrak{g}$ of dimension n associated (at the identity) to a matrix Lie group $\mathcal{G}$, and given a set of basis vectors $\{E_i\}_{i=1}^n$ of the matrix Lie algebra $\mathfrak{g}(n)$, we denote the *hat* operator $[\cdot]^\wedge : \mathbb{R}^n \rightarrow \mathfrak{g}$ which maps any n -vector $\mathbf{x} = [x_1 \dots x_n]^\top \in \mathbb{R}^n$ to an element $[\mathbf{x}]^\wedge \in \mathfrak{g}$, by $[\mathbf{x}]^\wedge = \sum_{i=1}^n x_i E_i$. We denote its inverse *vee* operator $[\cdot]^\vee : \mathfrak{g} \rightarrow \mathbb{R}^n$, which extracts the coordinates $\mathbf{x} = [x_1 \dots x_n]^\top$ from $[\mathbf{x}]^\wedge \in \mathfrak{g}(n)$ in terms of E_i . We denote the exponential map $\exp : \mathfrak{g} \rightarrow \mathcal{G}$, mapping an element $[\mathbf{x}]^\wedge \in \mathfrak{g}$ to the element $\exp([\mathbf{x}]^\wedge) \in \mathcal{G}$ in the neighborhood of the identity element. Specifically, for the $\text{SO}(3)$ group of rotations, the exponential map at the identity $\exp : \mathfrak{so}(3) \rightarrow \text{SO}(3)$ which associates any tangent vector (skew-symmetric matrix) $[\mathbf{x}]^\wedge \in \mathfrak{so}(3)$ to a 3D rotation as the matrix exponential, is given by

$$\begin{aligned} \exp([\mathbf{x}]^\wedge) &= \mathbf{I}_3 + [\mathbf{x}]^\wedge + \frac{1}{2}([\mathbf{x}]^\wedge)^2 + \frac{1}{6}([\mathbf{x}]^\wedge)^3 + \dots \\ &= \mathbf{I}_3 + \frac{\sin\|\mathbf{x}\|}{\|\mathbf{x}\|} [\mathbf{x}]^\wedge + \frac{1 - \cos\|\mathbf{x}\|}{\|\mathbf{x}\|^2} ([\mathbf{x}]^\wedge)^2. \end{aligned} \quad (2)$$

Finally, we denote the logarithm map (at the identity) by $\log : \mathcal{G} \rightarrow \mathfrak{g}$, which maps an element in the neighborhood of the identity element of group $\mathcal{G}$ to an element in the associated Lie algebra $\mathfrak{g}$. In the case when $\mathcal{G} = \text{SO}(3)$ and $\mathfrak{g} = \mathfrak{so}(3)$, the logarithm map $\log : \text{SO}(3) \rightarrow \mathfrak{so}(3)$ is the inverse of the operation given in (2), and is a bijective mapping as long as $\|\mathbf{x}\| < \pi$ ([Chirikjian, 2011](#)).

The following facts are useful in deriving first-order approximations for the effect of noise on the stochastic differential equations presented in [Section 4.6](#), as well as for deriving partial derivatives of functions of elements of $\text{SO}(3)$. Further information may be found in [Forster et al. \(2016\)](#); [Wang and Chirikjian \(2008\)](#).

Fact 1. Given the hat operator $[\cdot]^\wedge : \mathbb{R}^3 \rightarrow \mathfrak{so}(3)$, for any $R \in \text{SO}(3)$, and coordinate vector $\mathbf{x} \in \mathbb{R}^3$,

- (1) $R[\mathbf{x}]^\wedge R^\top = [R\mathbf{x}]^\wedge$,
- (2) $R[\mathbf{x}]^\wedge = [R\mathbf{x}]^\wedge R$,
- (3) $[\mathbf{x}]^\wedge R = R[R^\top \mathbf{x}]^\wedge$.

Note that Fact 1 is a consequence of the rotational invariance of the cross product in $\mathbb{R}^3$.

Fact 2. Given the exponential map $\exp : \mathfrak{so}(3) \rightarrow \text{SO}(3)$, for any $R \in \text{SO}(3)$ and any coordinate vector $\mathbf{x} \in \mathbb{R}^3$,

- (1) $R \exp([\mathbf{x}]^\wedge) R^\top = \exp([R\mathbf{x}]^\wedge)$,
- (2) $R \exp([\mathbf{x}]^\wedge) = \exp([R\mathbf{x}]^\wedge) R$,
- (3) $\exp([\mathbf{x}]^\wedge) R = R \exp([R^\top \mathbf{x}]^\wedge)$.

Note that Fact 2 is a direct consequence of Fact 1 applied to the matrix exponential expansion in (2).

Fact 3. Given the exponential map $\exp: \mathfrak{so}(3) \rightarrow \text{SO}(3)$, for any vector $\mathbf{x} \in \mathbb{R}^3$ having sufficiently small length, it follows that

$$\exp([\mathbf{x}]^\wedge) \simeq \mathbf{I}_3 + [\mathbf{x}]^\wedge. \quad (3)$$

Note that Fact 3 is a consequence of dropping higher order terms in the matrix exponential expansion of equation (2).

Fact 4. Let $T: \text{SO}(3) \times \mathbb{R}^3 \rightarrow \text{SE}(3)$ be a function that produces the pose corresponding to a rotation matrix and a translation vector pair, such that

$$T(R, \mathbf{r}) \triangleq \begin{bmatrix} R & \mathbf{r} \\ 0 & 1 \end{bmatrix}, R \in \text{SO}(3), \mathbf{r} \in \mathbb{R}^3.$$

Define the perturbation vectors, $\delta\Phi, \delta\mathbf{r} \in \mathbb{R}^3$. We may write the perturbed pose

$$T(R \exp([\delta\Phi]^\wedge), \mathbf{r} + \delta\mathbf{r}) = \begin{bmatrix} R \exp([\delta\Phi]^\wedge) & \mathbf{r} + \delta\mathbf{r} \\ 0 & 1 \end{bmatrix}.$$

Multiplying the inverse of the pose $T(R, \mathbf{r})$ on the left, we obtain the perturbation pose

$$\begin{aligned} \delta T &= (T(R, \mathbf{r}))^{-1} T(R \exp([\delta\Phi]^\wedge), \mathbf{r} + \delta\mathbf{r}) \\ &= \begin{bmatrix} R^\top & -R^\top \mathbf{r} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R \exp([\delta\Phi]^\wedge) & \mathbf{r} + \delta\mathbf{r} \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \exp([\delta\Phi]^\wedge) & R^\top \delta\mathbf{r} \\ 0 & 1 \end{bmatrix} \end{aligned}$$

Taking the logarithm map of the $\text{SE}(3)$, followed by the vee operator, we obtain the corresponding pose perturbation vector

$$\delta\mathbf{p} = \log(\delta T)^\vee = \begin{bmatrix} \delta\Phi \\ R^\top \delta\mathbf{r} \end{bmatrix},$$

thus yielding the Jacobian

$$J_R = \begin{bmatrix} \mathbf{I}_3 \\ 0_{3 \times 3} \end{bmatrix}, \text{ and } J_r = \begin{bmatrix} 0_{3 \times 3} \\ R^\top \end{bmatrix}.$$

It then follows that

$$\begin{bmatrix} \delta\Phi \\ \delta\mathbf{r} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_3 & 0 \\ 0 & R \end{bmatrix} \delta\mathbf{p}$$

2.2. Problem definitions

Let $A, S, O, W \in \mathbb{E}^3$ and assume that the point A corresponds to the center of mass of the small-body, S corresponds to the center of mass of the spacecraft, O corresponds to the center-of-mass of the Sun, and W corresponds to an inertial point in space, for example, non-accelerating, constant velocity. Initially, we distinguish three frames of interest: the inertial frame

$\mathcal{I} \triangleq (W, \{\vec{\mathbf{n}}_i\}_{i=1}^3)$, and the spacecraft body-fixed frame $\mathcal{S} \triangleq (S, \{\vec{\mathbf{s}}_i\}_{i=1}^3)$, as well as an arbitrarily chosen small body-fixed frame $\mathcal{G} \triangleq (G, \{\vec{\mathbf{g}}_i\}_{i=1}^3)$, useful for the description of geometry pertaining to the small-body. Note that since frame $\mathcal{G}$ is fixed with respect to small-body, and assuming that the small body is a rigid body, the position of the small-body's center of mass point A with respect to the point G in frame $\mathcal{G}$ coordinates, denoted $\mathbf{r}_{AG}^{\mathcal{G}}$, is fixed.

Given an inertial point W, the inertial absolute position vector of the observing spacecraft is denoted $\mathbf{r}_{SW}$ and that of the small-body $\mathbf{r}_{AW}$. The relative position vector of the observing spacecraft is then given as $\mathbf{r}_{SA} = \mathbf{r}_{SW} - \mathbf{r}_{AW}$. The inertial relative velocity vector of the observing spacecraft is denoted by $\mathbf{v}_{SA} \triangleq \frac{d}{dt}(\mathbf{r}_{SA})$. In the same fashion, the inertial relative acceleration is denoted $\mathbf{a}_{SA} \triangleq \frac{d}{dt}(\mathbf{v}_{SA})$.

To later exploit knowledge of the relative orbital pose, we decompose the relative position vector $\mathbf{r}_{SA}$ using the intermediary point G fixed in the $\mathcal{G}$ -frame coordinates, such that $\mathbf{r}_{SA} = \mathbf{r}_{SG} + \mathbf{r}_{GA} = \mathbf{r}_{SG} - \mathbf{r}_{AG}$. Expressing in the body-fixed $\mathcal{G}$ -frame coordinates and writing as a function of the relative position vector, we get the relative pose translation vector

$$\mathbf{r}_{SG}^{\mathcal{G}} = R_{GS} R_{SI} \mathbf{r}_{SA}^{\mathcal{I}} + \mathbf{r}_{AG}^{\mathcal{G}}. \quad (4)$$

Any reference hereafter to the relative pose of the spacecraft, for a given spacecraft frame $\mathcal{S} = (S, \{\vec{\mathbf{s}}_i\}_{i=1}^3)$, designates the transformation

$$T_{GS} = \begin{bmatrix} R_{GS} & \mathbf{r}_{SG}^{\mathcal{G}} \\ 0_{1 \times 3} & 1 \end{bmatrix}, \quad (5)$$

which encodes the relative rotation $R_{GS} = [\vec{\mathbf{s}}_1^{\mathcal{G}} \ \vec{\mathbf{s}}_2^{\mathcal{G}} \ \vec{\mathbf{s}}_3^{\mathcal{G}}]^\top \in \text{SO}(3)$ of the spacecraft with respect to the $\mathcal{G}$ frame and the coordinates $\mathbf{r}_{SG}^{\mathcal{G}} = [\mathbf{r}_{SG} \cdot \vec{\mathbf{g}}_1 \ \mathbf{r}_{SG} \cdot \vec{\mathbf{g}}_2 \ \mathbf{r}_{SG} \cdot \vec{\mathbf{g}}_3] \in \mathbb{R}^3$ of the spacecraft position vector relative to the point G as expressed in the $\mathcal{G}$ frame. Figure 1 provides an illustration of the problem's relevant points, vectors, and frames, as defined above.

2.3. Scene mapping considerations

Let t_0 be the initial time, let $t \geq t_0$, let $(t_k)_{k=0}^n \subset [t_0, t]$ be the sequence of sensor acquisition times, and let $T_{GS}(t_k)$ describe the pose of the spacecraft as expressed in the $\mathcal{G}$ frame at each time index $0 \leq k \leq n$. Then, the sequence $(T_{GS}(t_k))_{k=0}^n \in \Pi_{k=1}^n \text{SE}(3)$ describes the discrete trajectory of the relative pose of the spacecraft. We define the camera sensor frame $\mathcal{C} = (S, \{\vec{\mathbf{c}}_i\}_{i=1}^3)$, with fixed pose T_{SC} with respect to the spacecraft frame $\mathcal{S}$, and obtain the sequence $(T_{GC}(t_k))_{k=1}^n$ of all camera poses, also known as frames,

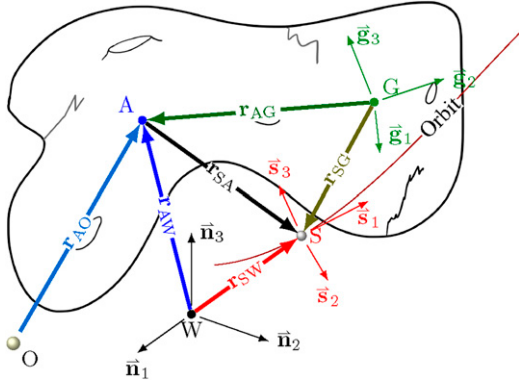


Figure 1. Relative navigation problem frame definitions and vector quantities.

where the relative camera pose is $T_{G_k C_k} = T_{G_k S_k} T_{S_k C_k} = T_{G_k S_k} T_{SC}$, ($k = 0, \dots, n$).

A *landmark* $L \in \mathbb{E}^3$ is defined as a notable 3D point in the scene, which is potentially triangulated during SLAM using camera observations. We denote by $\Psi_k = \{L_i \in \mathbb{E}^3, i = 1, \dots, m_k\}$ the set of all landmarks accumulated up until time index $k = 0, \dots, n$, also called the *map* at time index k . To each landmark $L \in \Psi_k$ corresponds a position vector $\mathbf{r}_{LG} \in \mathbb{R}^3$, whose coordinates in the $\mathcal{G}$ frame, denoted $\mathcal{G}$, are fixed since the small body is presumed to be a rigid body. Let r_k be the total number of image feature points detected in the camera image captured at time t_k , $k = 0, \dots, n$. We collect all detected feature points in the set $\mathcal{I}_k = \{P_i \in \mathbb{P}^2, i = 1, \dots, r_k\}$ where, to each $P_i \in \mathcal{I}_k$ are associated the ideal 2D image coordinates $\mathbf{y}_{ik} \in \mathbb{R}^2$.

Assume a point $P_i \in \mathcal{I}_k$ corresponds to the image projection of a scene landmark $L_i \in \Psi_k$, as captured at time index k . The 2D image coordinates $\mathbf{y}_{ik}$ relate to the 3D position coordinates $\mathbf{r}_{L_i G}^G$ through the pinhole camera model relationship, given by

$$\begin{bmatrix} \lambda \mathbf{y}_{ik} \\ \lambda \end{bmatrix} = \mathbf{K} T_{C_k G_k} \mathbf{r}_{L_i G}^G, \quad (6)$$

where $\lambda > 0$ is a scaling factor, and where

$$\mathbf{K} \triangleq \begin{bmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

is the camera intrinsic matrix, with f_x, f_y, c_x, c_y scalars corresponding to the known camera focal lengths and optical center offsets along the two image dimensions. The real measured feature point coordinates $\mathbf{y}_{ik}^m$ are then defined such that

$$\mathbf{y}_{ik}^m = \mathbf{y}_{ik} + \mathbf{v}_y, \quad (7)$$

where $\mathbf{v}_y \sim \mathcal{N}(\mathbf{0}, \Sigma_y^m)$ corresponds to the feature point measurement noise, with associated 2×2 covariance matrix Σ_y^m . We collect all real camera frame-landmark

observations up until time index $k = 0, \dots, n$ in the set $\mathcal{Y}_k \triangleq \{\mathbf{y}_{ij}^m \in \mathbb{R}^2 : i = 1, \dots, r_j, j = 0, \dots, k\}$. We also define $\mathcal{X}_k \triangleq \{T_{G_i S_i} \in \text{SE}(3) : i = 0, \dots, k\}$ as the set of all possible spacecraft relative poses discretized at times $(t_k)_{k=0}^n$, and define $\mathcal{L}_k \triangleq \{\mathbf{r}_{LG}^G \in \mathbb{R}^3 : L \in \Psi_k\}$ as the set of all landmark coordinates mapped up until time index $k = 0, \dots, n$.

By exploiting the multi-view geometry constraints derived by capturing observations $\mathcal{Y}_k$ of the landmarks Ψ_k at poses $\mathcal{X}_k$, as well as the constraints derived from the intrinsic motion of the spacecraft around the small body and other sensor measurements, further detailed in Section 4, we wish to find a solution to the trajectory $\mathcal{X}_k$ along with the set of mapped landmark coordinates $\mathcal{L}_k$, on-the-fly, for $k = 0, \dots, n$ in a sequential and incremental manner. We detail the method employed to solve this problem in Section 5. First, we provide a brief primer on the Bayesian estimation, the SLAM problem formulated in the Bayesian framework, and the relevance of representing the structure of the problem using a factor graph.

3. Estimating relative motion using IMU measurements

We now further explain the discrepancies between our work and Setterfield's procedure in Setterfield et al. (2018a), specifically: (a) why the IMU preintegration scheme is inadequate and why relative dynamics need to be considered; and (b) why the kinematic rotation factor introduced in Setterfield et al. (2018a) needs to be modified to account for non-inertial and large motion of the target center-of-mass point.

3.1. Using IMU accelerometer measurements to perform dead-reckoning in the presence of unknown gravity vector

As stated in Setterfield et al. (2017) “[a]lthough the IMU preintegration factor was intended to accept gyroscope and accelerometer inputs and operate in the influence of Earth’s gravity field, the mathematics and open source implementation accept any valid angular rate and acceleration measurements in any desired gravitational field. [...] Herein, gyroscope measurements are used for the angular rate measurements, the gravitational field strength is set to zero, and thruster forces $\mathbf{u}$ divided by vehicle mass m are used in lieu of accelerometer measurements.” This may be true in the specific context of spacecraft relative navigation in Earth orbit, but would not hold if the gravitational effect is not known a priori, as we explain below.

Specifically, in our scenario: (1) in terms of gravitational forces, the observing spacecraft is affected by the combined pull of the target small-body and of the Sun, while the small-body is only affected by the gravitational acceleration of the Sun. This results in a net non-negligible relative gravitational interaction between the “chief” (the small-body) and the “deputy” (the observer S/C), which, in turn, affects the

relative motion of the observer spacecraft. However, an IMU accelerometer is insensitive to gravitational forces. The double integrator dynamics assumed in [Setterfield et al. \(2018a\)](#) do not hold over extended arcs of time, and a full non-linear treatment provides higher precision; (2) there are persistent external non-gravitational forces acting on the observer spacecraft which are small in magnitude, and consequently may not be detectable given the signal-to-noise ratio (SNR) of an IMU measurement signal, but which may still affect the long-term trajectory of the observing spacecraft over an inspection arc. The IMU measurements required for the IMU factors incorporated in [Setterfield et al. \(2017, 2018a\)](#) may not capture the accumulated effect of such small-scale non-gravitational forces; (3) unlike spacecraft inspection arcs, the duration of small-body inspection arcs is typically extended—that is, their length is a non-negligible portion of the period of revolution of the spacecraft around the small-body. Assuming, similarly to [Setterfield et al. \(2018a\)](#), that the center-of-mass of the target body only moves negligibly between time instances, or that its motion is inertial throughout the inspection arc, may induce considerable relative navigation errors during estimation.

When we analyze the SPHERES-VERTIGO platform-inspired problem formulation described in [Setterfield \(2017\)](#); [Setterfield et al. \(2017, 2018a\)](#), we note that the inspector spacecraft (deputy) and the ISS habitat module in which the experiment is run (chief) are both subjected to the same Earth gravitational acceleration which maintains them in orbit, without any other gravitational interaction between the two. The ISS frame, being nadir-pointing, may be assimilated to the Hill frame, and consequently the relative motion of the observer spacecraft with respect to the Hill frame is fundamentally described by the Clohessy-Wiltshire (CW) equations of motion ([Schaub and Junkins, 2003](#)). We note that the target spacecraft (another deputy) has a similar relationship to the ISS frame (chief), but its coordinate is coincident with that of the Hill frame (called world frame in [Setterfield et al. \(2018a\)](#) and assumed to be inertial, as explained in [Setterfield et al. \(2017\)](#)). In this specific context, given the very small difference in Earth-centered inertial position between the chief (ISS module) and the deputy (observer) spacecraft, and the short time-frame of the considered experiment (60–160 s in [Setterfield et al. \(2018a\)](#)), the Hill frame may be considered as quasi-inertial and the motion of the observer may indeed be approximated by the solution to double-integrator dynamics. Only in this limited scenario, the simple accelerometer model discussed in [Setterfield et al. \(2017\)](#), where gravity is set to zero and acceleration measurements are simulated as thruster-specific forces, seems valid.

In the latter case, the accelerometer measurements may be used to perform “dead-reckoning,” that is, to compute the change in the inertial velocity and the inertial position of the observer spacecraft with respect to the quasi-inertial world frame due to the influence of external accelerations, such as

the action of thruster firing. Such dead-reckoning may be performed using the preintegration factor of [Forster et al. \(2016\)](#), which attempts to encode the residual between the current state estimate and the state obtained by preintegration of these accelerometer measurements starting from the previous navigational state.

However, as is generally known ([Goel et al., 2021](#)), an IMU’s accelerometer does not measure pure accelerations, but rather externally applied non-gravitational specific forces, that is, inertial accelerations sans gravity-induced accelerations.

The IMU measurements do not provide any information about the direction or the magnitude of the gravity vector in freefall (unforced orbital) motion. We can intuit this fact by considering a simple model for the measurement $\mathbf{n}_{\text{MO}}^{\mathcal{M},\text{m}}$ of an accelerometer centered at point M and with frame $\mathcal{M}$, given as

$$\mathbf{n}_{\text{MO}}^{\mathcal{M},\text{m}} = cR_{\mathcal{M}\mathcal{I}}(\mathbf{a}_{\text{MO}}^{\mathcal{I}} - \mathbf{g}^{\mathcal{I}}) + \mathbf{v}, \quad (8)$$

where O is an inertial point, $\mathcal{I}$ is an inertial frame, c is a measurement sensitivity constant, $R_{\mathcal{M}\mathcal{I}}$ is the rotation between frames $\mathcal{M}$ and $\mathcal{I}$, and $\mathbf{v}$ is a Gaussian noise term. It follows that, in an unforced (freefall) motion, while assuming the IMU frame $\mathcal{M}$ is coincident with the spacecraft body frame $\mathcal{S}$, only gravity affects the IMU, that is, $\mathbf{a}_{\text{MO}}^{\mathcal{I}} = \mathbf{g}^{\mathcal{I}}$, and thus the accelerometer measurement over time is null on average. In other words, the IMU accelerometer is unusable for the purpose of generating estimates of inertial velocity and change in inertial position by dead-reckoning in the presence of unknown gravitational accelerations. To obtain an accurate estimate of inertial motion, $\mathbf{g}^{\mathcal{I}}$ needs to be known.

3.2. Using IMU accelerometer measurements to perform dead-reckoning of motion owing to small-scale persistent perturbing forces

Admittedly, accelerometer measurements may be used in our work to obtain estimates of motion owing only to non-gravitational external forces being applied on the S/C, such as actuated thruster profiles. However, such forces need to be strong enough to generate a large enough signal-to-noise ratio to be useful. One common external force which does not meet the latter requirement, but whose effect on the trajectory is non-negligible over the length of an inspection arc, is solar radiation pressure (SRP). Its magnitude is typically orders of magnitude smaller than the gravitational effect ([Russell and Raymond, 2012a](#); [Scheeres, 2016](#)), and it does not meet the criterion of having enough SNR to be detectable by the accelerometer.

In light of these caveats, incorporating IMU factors to process such IMU measurements, given that we need to augment the state with the additional IMU bias states and incur additional computation time, may not be justifiable. Therefore, an admissible “odometric” factor for our scenario would then have to rely on explicitly modeling and

tracking the gravity vector, to ascertain the true relative motion between the observer spacecraft and the target small-body. Moreover, since the center-of-mass of the small-body may not be considered as inertial owing to the gravitational pull by the Sun over the inspection arc, and its inertial velocity may not be known with good precision during proximity operations, it is preferable to model the relative dynamics of the spacecraft-small-body pair, by avoiding the modeling of the inertial dynamics of both spacecraft and target small-body separately, as is done in Rathinam and Dempster (2017).

4. Relative kinematic and orbital odometric factors

In this section, we detail the theoretical approach for devising an adaptation of Setterfield's kinematic rotation factor (Setterfield et al., 2018a) as well as our new RelDyn factor, which enforces a strong odometric constraint, namely, the equations governing the motion of the spacecraft relative to the small-body, and which incorporates fusion of other sensor measurements. We also discuss the relevant sources of noise, and the propagated system of equations.

4.1. Factor graph encoding and iSAM2 algorithm

In modern renditions of SLAM (Dellaert and Kaess, 2017), the problem stated in Section 2.2 is formulated using a probabilistic inference framework, predicated on Bayesian estimation, which we evoke for our solution in this section. The use of the factor graph formulation for modeling the small body navigation problem is motivated by the fact that the typical Bayesian estimation method used to solve the SLAM estimation problem is amenable to a graph representation due to the sparsity in the structure of the cost. Factor graphs are un-directed bi-partite probabilistic graphical models constituted of *factor* nodes and *variable* nodes, with *edges* connecting variable-factor pairs (Dellaert, 2021). The structure captured by the edges and nodes of the factor graph encodes the structure of the estimation problem's posterior probability density function, by exploiting the fact that the latter can be factorized as a product of many functions, each depending on a subset of the variables of the problem. By exploiting sparsity in the structure of the joint density function, the factor graph formulation can render very large estimation problems tractable in terms of computation. Factor functions can be derived and emplaced in the factor graph based on the *problem-specific constraints* which we wish to include.

Given a constructed factor graph and a specific variable ordering, variable elimination is performed, transforming the factor graph into a chordal Bayes net. We note that, in theory, the chordal Bayes net may then be written as an equivalent square root information matrix, which may then

be solved using algebra (back-substitution) when the factors are linear, and by Gauss-Newton iteration when the factors are non-linear. The procedure creates the resulting Gauss-Newton iteration directly from the factor graph and then attempts to solve for updated values of the variable set.

However, in more recent and efficient renditions (Kaess et al., 2012); (Fourie et al., 2021) of factor graph solvers, Bayes trees are used to store internally the structure of the problem, while the factor graph is used principally as an intermediary for modeling. A Bayes tree is a type of directed junction tree in which the nodes store the cliques of the chordal Bayes net it represents and the edges are the separator set of variables which separate the cliques.

In iSAM2 (Kaess et al., 2012), specifically, after a first step variable elimination step is performed on the initial factor graph, a Bayes tree is constructed from the chordal Bayes net and stored for the next time step, contrary to the classical approach of storing the factor graph itself for the next stage. When a new subgraph relating to a set of new variables and measurements is available for insertion, iSAM2 updates the Bayes tree by (1) converting the part of the Bayes tree relating to this new subgraph into a factor graph, (2) appending the new factor subgraph—usually at the top of the tree, that is, affecting the most recent cliques—then (3) eliminating the factor subgraph to produce a new Bayes net, and (4) reassesses the affected cliques, (5) updates the appropriate variables. iSAM2 updates variables involved in non-linear factors by performing a non-linear optimization step at a current linearization point. To select which variables undergo optimization update, iSAM2 first marks all variables in a subset of the graph variables—that is, the ones being added to the graph—which clear the “wildfire” threshold to update, after which a new linearization point is computed and subsequent all cliques which involve the updated variable get marked for update as well.

These points of the procedure are important in that they dictate the effect on the iSAM2 runtime and complexity when we insert variables which may render the graph fully connected, greatly affecting the size of the cliques marked for update and optimization. We further discuss these in Section 5.1.

Next, we devise the factor which serves as a motion constraint in the small body circumnavigation problem.

4.2. The relative kinematic factor formulation

We modified the kinematic rotation factor introduced in Setterfield et al. (2018a) and incorporated it in the problem factor graph. Unlike the formulation of the kinematic rotation factor in Setterfield et al. (2018a), however, we do not assume that the target's center-of-mass point A is inertial or that the motion of the target's center-of-mass between two successive time indices is negligible. Therefore, for any two time indices i, j , $i \neq j$, $\mathbf{r}_{A_j A_i} \neq \mathbf{0}$, and we can write

$$\mathbf{r}_{S_j S_i} = \mathbf{r}_{S_j G_j} + \mathbf{r}_{G_j A_j} + \mathbf{r}_{A_j A_i} + \mathbf{r}_{A_i G_i} + \mathbf{r}_{G_i S_i}. \quad (9)$$

We rewrite the left-hand side of equation (9) by transiting through points A_j and A_i in the vector chain, such that

$$\mathbf{r}_{S_j S_i} = \mathbf{r}_{S_j A_j} + \mathbf{r}_{A_j A_i} + \mathbf{r}_{A_i S_i}, \quad (10)$$

We replace equation (10) into the equation (9), cancel out terms, and manipulate to obtain the kinematic relationship, stated as

$$\mathbf{r}_{S_j A_j} - \mathbf{r}_{S_i A_i} = \mathbf{r}_{S_j G_j} + \mathbf{r}_{G_j A_j} + \mathbf{r}_{A_i G_i} + \mathbf{r}_{G_i S_i}. \quad (11)$$

We note that in lieu of the inertial position vectors $\mathbf{r}_{S_i W}$ with respect to an inertial world frame origin W used for the kinematic factor in [Setterfield et al. \(2018a\)](#), we have the relative vector $\mathbf{r}_{S_i A_i}$. While acknowledging that the center-of-mass vector $\mathbf{r}_{AG}$ is fixed when expressed in the $\mathcal{G}$ -frame coordinates, such that $\mathbf{r}_{A_i G_i}^{\mathcal{G}} = \mathbf{r}_{A_j G_j}^{\mathcal{G}} = \mathbf{r}_{AG}^{\mathcal{G}}$, and expressing all vectors in the inertial frame, we get the kinematic relationship

$$\begin{aligned} \mathbf{r}_{S_j A_j}^{\mathcal{I}} - \mathbf{r}_{S_i A_i}^{\mathcal{I}} &= R_{IS_j} R_{S_j G_j} \left(\mathbf{r}_{S_j G_j}^{\mathcal{G}} - \mathbf{r}_{AG}^{\mathcal{G}} \right) \\ &\quad + R_{IS_i} R_{S_i G_i} \left(\mathbf{r}_{AG}^{\mathcal{G}} - \mathbf{r}_{S_i G_i}^{\mathcal{G}} \right). \end{aligned} \quad (12)$$

Further assuming that we have access to a filtered estimate $\hat{R}_{IS}$ of the inertial attitude $R_{IS} = \hat{R}_{IS} \exp([\mathbf{v}_R]^{\wedge})$, $\mathbf{v}_{R,i} \sim \mathcal{N}(\mathbf{0}, \Sigma_R)$ from the spacecraft's inertial star tracker at any desired time, we may rewrite equation (12) as

$$\begin{aligned} \mathbf{r}_{S_j A_j}^{\mathcal{I}} - \mathbf{r}_{S_i A_i}^{\mathcal{I}} &= \hat{R}_{IS_j} \exp([\mathbf{v}_{R,j}]^{\wedge}) R_{S_j G_j} \left(\mathbf{r}_{S_j G_j}^{\mathcal{G}} - \mathbf{r}_{AG}^{\mathcal{G}} \right) \\ &\quad + \hat{R}_{IS_i} \exp([\mathbf{v}_{R,i}]^{\wedge}) R_{S_i G_i} \left(\mathbf{r}_{AG}^{\mathcal{G}} - \mathbf{r}_{S_i G_i}^{\mathcal{G}} \right). \end{aligned} \quad (13)$$

Using Facts 1 and 3, and manipulating equation (13), we obtain the translation residual

$$\begin{aligned} \mathcal{E}_{ij}^{\text{RelKin}, \mathbf{r}} &\triangleq \mathbf{r}_{S_j A_j}^{\mathcal{I}} - \mathbf{r}_{S_i A_i}^{\mathcal{I}} \\ &\quad - \hat{R}_{IS_j} R_{S_j G_j} \left(\mathbf{r}_{S_j G_j}^{\mathcal{G}} - \mathbf{r}_{AG}^{\mathcal{G}} \right) \\ &\quad - \hat{R}_{IS_i} R_{S_i G_i} \left(\mathbf{r}_{AG}^{\mathcal{G}} - \mathbf{r}_{S_i G_i}^{\mathcal{G}} \right) \\ &= \left[\hat{R}_{IS_j} R_{S_j G_j} \left(\mathbf{r}_{S_j G_j}^{\mathcal{G}} - \mathbf{r}_{AG}^{\mathcal{G}} \right) \right]^{\wedge} \hat{R}_{IS_j} \mathbf{v}_{R,j} \\ &\quad + \left[\hat{R}_{IS_i} R_{S_i G_i} \left(\mathbf{r}_{AG}^{\mathcal{G}} - \mathbf{r}_{S_i G_i}^{\mathcal{G}} \right) \right]^{\wedge} \hat{R}_{IS_i} \mathbf{v}_{R,i}. \end{aligned} \quad (14)$$

While assuming that $\mathbb{E}[\mathbf{v}_{R,i} \mathbf{v}_{R,j}^{\top}] = \mathbf{0}_{3 \times 3}$, $i \neq j$, and using the shorthand $B_i \triangleq [\hat{R}_{IS} R_{S_i G_i} (\mathbf{r}_{AG}^{\mathcal{G}} - \mathbf{r}_{S_i G_i}^{\mathcal{G}})]^{\wedge}$, an expression for the covariance of $\mathcal{E}_{ij}^{\text{RelKin}, \mathbf{r}}$ may be found by computing

$$\mathbb{E}[\mathcal{E}_{ij}^{\text{RelKin}, \mathbf{r}} (\mathcal{E}_{ij}^{\text{RelKin}, \mathbf{r}})^{\top}] = B_i \Sigma_R B_i^{\top} + B_j \Sigma_R B_j^{\top} \quad (15)$$

In parallel, and similarly to [Setterfield et al. \(2018b\)](#), we can write the rotation kinematic relationship using the rotation composition rule, given as

$$R_{G_i G_j} = R_{G_i S_i} R_{S_i \mathcal{I}} R_{\mathcal{I} S_j} R_{S_j G_j} \quad (16)$$

We rewrite the left-hand side of the equation using the intermediary $\mathcal{I}$ frame, such that

$$R_{G_i G_j} = R_{G_i \mathcal{I}} R_{\mathcal{I} G_j}. \quad (17)$$

Using an Euler on-manifold integration scheme with the simplifying assumption of piecewise constant $\boldsymbol{\omega}_{G\mathcal{I}}(t) = \boldsymbol{\omega}_{G_i \mathcal{I}}$, $t \in [t_i, t_j]$, a kinematic relationship between $R_{\mathcal{I} G_i}$ and $R_{\mathcal{I} G_j}$ holds, such that

$$R_{\mathcal{I} G_j} = R_{\mathcal{I} G_i} \exp([\boldsymbol{\omega}_{G_i \mathcal{I}}(t_j - t_i)]^{\wedge}). \quad (18)$$

Substituting equation (18) into equation (17), and then substituting that result into equation (16), we obtain the kinematic relationship

$$\exp([\boldsymbol{\omega}_{G_i \mathcal{I}}(t_j - t_i)]^{\wedge}) = R_{G_i S_i} R_{S_i \mathcal{I}} R_{\mathcal{I} S_j} R_{S_j G_j}. \quad (19)$$

Further assuming that we have access to a filtered estimate $\hat{R}_{IS_j}$ of the inertial attitude $R_{IS_j} = \hat{R}_{IS_j} \exp([\mathbf{v}_{R,i}]^{\wedge})$, $\mathbf{v}_{R,i} \sim \mathcal{N}(\mathbf{0}, \Sigma_R)$ from the spacecraft's inertial star tracker, we may rewrite equation (19) as

$$\begin{aligned} \exp([\boldsymbol{\omega}_{G_i \mathcal{I}}(t_j - t_i)]^{\wedge}) &= R_{G_i S_i} \exp([\mathbf{v}_{R,i}]^{\wedge}) \hat{R}_{IS_i} \\ &\quad \times \hat{R}_{IS_j} \exp([\mathbf{v}_{R,j}]^{\wedge}) R_{S_j G_j} \end{aligned}$$

Using Fact 2, and manipulating the right-hand side of the latter equation, we obtain

$$\begin{aligned} R_{G_j S_j} \hat{R}_{S_j \mathcal{I}} \hat{R}_{IS_i} R_{S_i G_i} \exp([\boldsymbol{\omega}_{G_i \mathcal{I}}(t_j - t_i)]^{\wedge}) \\ = \exp\left([R_{G_j S_j} \hat{R}_{S_j \mathcal{I}} \hat{R}_{IS_i} \mathbf{v}_{R,i}]^{\wedge}\right) \\ \times \exp([\mathbf{v}_{R,j}]^{\wedge}). \end{aligned}$$

Taking the logarithm map followed by the *vee* operator on both sides of the equation, and then applying the Baker-Campbell-Hausdorff formula ([Chirikjian, 2011](#)) while omitting any second-order noise term, we obtain the residual

$$\begin{aligned} \mathcal{E}_{ij}^{\text{RelKin}, R} &\triangleq \\ &\log\left(R_{G_j S_j} \hat{R}_{S_j \mathcal{I}} \hat{R}_{IS_i} R_{S_i G_i} \exp([\boldsymbol{\omega}_{G_i \mathcal{I}}(t_j - t_i)]^{\wedge})\right)^{\vee} \\ &= R_{G_j S_j} \hat{R}_{S_j \mathcal{I}} \hat{R}_{IS_i} \mathbf{v}_{R,i} - R_{G_j S_j} \mathbf{v}_{R,j}. \end{aligned} \quad (20)$$

Since (a) rotation matrices are full rank and norm preserving, (b) the star tracker measurement noise is typically isotropic, for example, $\Sigma_R = \sigma_R^2 \mathbf{I}_3$, and (c) that the star tracker measurement noises $\mathbf{v}_{R,i}$, $\mathbf{v}_{R,j}$ are independent for $i \neq j$, that is $\mathbb{E}[\mathbf{v}_{R,i} \mathbf{v}_{R,j}^{\top}] = \mathbf{0}_{3 \times 3}$, it follows that $\mathcal{E}_{ij}^{\text{RelKin}, R} \sim (\mathbf{0}, 2\Sigma_R)$.

We combine the two residuals to obtain the full relative kinematic factor

$$\begin{aligned} \epsilon_{ij}^{\text{RelKin}} &= \begin{bmatrix} \epsilon_{ij}^{\text{RelKin}, R} \\ \epsilon_{ij}^{\text{RelKin}, r} \end{bmatrix} \\ &\sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} 2\Sigma_R & 0 \\ 0 & B_i\Sigma_R B_i^\top + B_j\Sigma_R B_j^\top \end{bmatrix}\right). \end{aligned} \quad (21)$$

We incorporated this adapted factor into our factor graph formulation.

4.3. The RelDyn factor formulation

Similarly to Dong et al. (2016), which implements STEAM using a Gaussian Process with a linear time-varying stochastic differential equation, we derive a relative dynamics factor based on the moments of the distribution describing the dispersion of the solution realizations of a system of non-linear stochastic differential equations, evaluated at discrete times, for the purpose of improving the performance of baseline SLAM for small-body imagery.

Given a sequence of spacecraft states $\Xi_n = (x_k)_{k=0}^n$ at discrete times $(t_k)_{k=0}^n$, we want to devise factors which relate to the odometric probability distribution $p(x_k, x_{k+1})$. Note that, using Bayes' rule, the joint distribution $p(x_k, x_{k+1})$ can be rewritten as either $p(x_k, x_{k+1}) = p(x_{k+1}|x_k)p(x_k)$ or $p(x_k, x_{k+1}) = p(x_k|x_{k+1})p(x_{k+1})$, depending on which prior, $p(x_k)$ or $p(x_{k+1})$, is readily available at the time of computation. In our case, we typically know the state of the spacecraft and the distribution of the uncertainty of that state at the beginning of a segment, and therefore we pick $p(x_0) \sim \mathcal{N}(\hat{x}_0, \Sigma_0)$ as a known prior, where $\hat{x}_0$ is the known state mean and Σ_0 is the known state covariance. We associate the factor $\phi_0^{\text{prior}}(x_0)$ to the prior $p(x_0)$ accordingly. Now, the RelDyn factor ϕ_k^{RelDyn} can be formulated as

$$\phi_k^{\text{RelDyn}}(x_k, x_{k+1}) \propto p(x_{k+1}|x_k) = \mathcal{N}(\mathbf{0}, P_k), \quad (22)$$

where P_k is a covariance matrix derived from the propagation of the moments of the distribution describing the realizations of the stochastic differential equations.

Along with projection factors, denoted $\phi_i^{\text{proj}}, i = 1, \dots, r_k$, a factor graph of the SLAM problem with motion priors can be constructed, as conceptually illustrated in Figure 2. In light of this goal, a brief discussion of the RelDyn factor is provided throughout this section. To compute the value of the RelDyn, we need to model the equations of the relative motion induced by the relative navigation problem in the presence of gravitational attraction between the observer and target bodies.

4.4. Equations of relative motion

Recalling the vectors and frames in Figure 1, we assume that the frame $\mathcal{I}$ is inertially fixed, the frames $\mathcal{G}$ and $\mathcal{S}$ are rotating with associated inertial angular velocity vectors $\omega_{\mathcal{G}\mathcal{I}}$, and $\omega_{\mathcal{S}\mathcal{I}}$, and that frame $\mathcal{G}$ is a target small body body-fixed frame. The relative angular velocity between the spacecraft

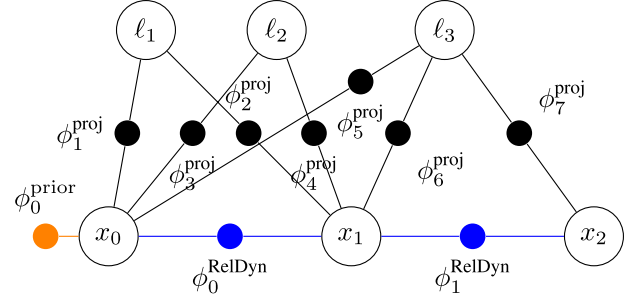


Figure 2. A factor graph encoding the SLAM problem and the RelDyn odometric factors conceptually.

frame $\mathcal{S}$ and frame $\mathcal{G}$ is $\omega_{\mathcal{S}\mathcal{G}} = \omega_{\mathcal{S}\mathcal{I}} - \omega_{\mathcal{G}\mathcal{I}}$. The orientation of the frame $\mathcal{S}$ relative to the frame $\mathcal{G}$ is encoded in the rotation matrix $R_{\mathcal{G}\mathcal{S}} \in \text{SO}(3)$, which satisfies the kinematic relationship

$$\dot{R}_{\mathcal{G}\mathcal{S}} = [\omega_{\mathcal{S}\mathcal{G}}^\mathcal{G}]^\wedge R_{\mathcal{G}\mathcal{S}}, \quad (23)$$

where $\omega_{\mathcal{S}\mathcal{G}}^\mathcal{G} \in \mathbb{R}^3$ is the relative angular velocity, expressed in the $\mathcal{G}$ frame. Rewriting (23) in terms of the frame $\mathcal{S}$ and frame $\mathcal{G}$ angular velocities, and using Fact 1, we obtain

$$\begin{aligned} \dot{R}_{\mathcal{G}\mathcal{S}} &= [R_{\mathcal{G}\mathcal{S}}\omega_{\mathcal{S}\mathcal{I}}^\mathcal{S} - \omega_{\mathcal{G}\mathcal{I}}^\mathcal{G}]^\wedge R_{\mathcal{G}\mathcal{S}} \\ &= R_{\mathcal{G}\mathcal{S}}[\omega_{\mathcal{S}\mathcal{I}}^\mathcal{S}]^\wedge - [\omega_{\mathcal{G}\mathcal{I}}^\mathcal{G}]^\wedge R_{\mathcal{G}\mathcal{S}}. \end{aligned} \quad (24)$$

We can derive the position vector of the spacecraft at point S relative to the arbitrary point G by writing

$$\mathbf{r}_{\mathcal{S}\mathcal{G}} = \mathbf{r}_{\mathcal{S}\mathcal{A}} + \mathbf{r}_{\mathcal{A}\mathcal{G}}.$$

Taking the time derivative in the $\mathcal{G}$ frame gives

$$\begin{aligned} \frac{d}{dt}(\mathbf{r}_{\mathcal{S}\mathcal{G}}) &= \frac{d}{dt}(\mathbf{r}_{\mathcal{S}\mathcal{A}}) + \frac{d}{dt}(\mathbf{r}_{\mathcal{A}\mathcal{G}}) \\ &= \frac{d}{dt}(\mathbf{r}_{\mathcal{S}\mathcal{A}}) - \omega_{\mathcal{G}\mathcal{I}} \times \mathbf{r}_{\mathcal{S}\mathcal{A}}. \end{aligned}$$

Expressing in the $\mathcal{G}$ coordinates, we get

$$\dot{\mathbf{r}}_{\mathcal{S}\mathcal{G}}^\mathcal{G} = R_{\mathcal{G}\mathcal{S}}R_{\mathcal{S}\mathcal{I}}^\top \mathbf{v}_{\mathcal{S}\mathcal{A}}^\mathcal{I} - [\omega_{\mathcal{G}\mathcal{I}}^\mathcal{G}]^\wedge R_{\mathcal{G}\mathcal{S}}R_{\mathcal{S}\mathcal{I}}^\top \mathbf{r}_{\mathcal{S}\mathcal{A}}^\mathcal{I}. \quad (25)$$

Finally, we derive the translational dynamics of the spacecraft relative to the small body center of mass. Our equations of motion for the relative position and relative velocity are then given as

$$\dot{\mathbf{r}}_{\mathcal{S}\mathcal{A}}^\mathcal{I} = \mathbf{v}_{\mathcal{S}\mathcal{A}}^\mathcal{I} \quad (26)$$

$$\dot{\mathbf{v}}_{\mathcal{S}\mathcal{A}}^\mathcal{I} = \mathbf{a}_{\mathcal{S}\mathcal{A}}^\mathcal{I}. \quad (27)$$

It thus suffices to find an expression for $\mathbf{a}_{\mathcal{S}\mathcal{A}}^\mathcal{I}$.

We assume herein that the spacecraft is subjected to the gravitation force of the small body, denoted by $\mathbf{F}_{sa}$, the gravitation of the Sun, denoted by $\mathbf{F}_{s\odot}$, the solar radiation pressure (SRP), denoted by $\mathbf{F}_{\text{SRP}}$, as well as the spacecraft actuation thrust force, denoted by $\mathbf{F}_s$. Assuming that the spacecraft's known mass, denoted by m_s , is fixed, the linear acceleration of the spacecraft with respect to the Sun's

origin O, is given by $\mathbf{a}_{SO} = 1/m_s(\mathbf{F}_{sa} + \mathbf{F}_{s\odot} + \mathbf{F}_{SRP} + \mathbf{F}_s)$. The small-body's gravitation force is obtained by computing

$$\mathbf{F}_{sa} = \frac{\partial U(\mathbf{r})}{\partial \mathbf{r}} \Big|_{\mathbf{r}=\mathbf{r}_{SA}} \quad (28)$$

for an appropriate gravity field potential function $U(\mathbf{r})$. Assume that $U(\mathbf{r})$ is parameterized using spherical harmonics. Then, when the probe is relatively distant from the small body, the spherical term of the potential dominates, in which case the attractive force is given by $\mathbf{F}_{sa} = -\mu_a m_s / \|\mathbf{r}_{SA}\|^3 \mathbf{r}_{SA}$. Consider the Sun as a point-mass central body. Then, $\mathbf{F}_{s\odot} = -\mu_\odot m_s / \|\mathbf{r}_{SO}\|^3 \mathbf{r}_{SO}$, where $\mathbf{r}_{SO}$ is the position vector of the spacecraft center of mass S with respect to the Sun origin O. Assume that the solar radiation pressure is a function of its position vector with respect to the Sun, $\mathbf{F}_{SRP} = \mathbf{F}_{SRP}(\mathbf{r}_{SO})$.

In turn, we assume that the mass of the small body, denoted m_a is fixed, and that the only force acting on the small body is the Sun's gravitational force. Then, the linear acceleration of the small body center of mass with respect to the Sun's origin O, is given by $\mathbf{a}_{AO} = 1/m_a \mathbf{F}_{a\odot}$, where $\mathbf{F}_{a\odot} = -\mu_\odot m_a / \|\mathbf{r}_{AO}\|^3 \mathbf{r}_{AO}$. The relative dynamics of the spacecraft-small-body system are obtained computing $\mathbf{a}_{SA} = \mathbf{a}_{SO} - \mathbf{a}_{AO}$, leading to the relationship

$$\begin{aligned} \mathbf{a}_{SA} = & -\frac{\mu_a}{\|\mathbf{r}_{SA}\|^3} \mathbf{r}_{SA} - \frac{\mu_\odot}{\|\mathbf{r}_{SO}\|^3} \mathbf{r}_{SO} + \frac{1}{m_s} \mathbf{F}_{SRP}(\mathbf{r}_{SO}) \\ & + \frac{1}{m_s} \mathbf{F}_s + \frac{\mu_\odot}{\|\mathbf{r}_{AO}\|^3} \mathbf{r}_{AO}. \end{aligned} \quad (29)$$

Assume that $\mathbf{r}_{SO} \approx \mathbf{r}_{AO}$, given the very large distance between the spacecraft-small-body system and the Sun, and rewrite equation (29) by making explicit in terms of the state variables and input variables of interest, yielding

$$\begin{aligned} \mathbf{a}_{SA}^T = & -\left(\frac{\mu_a}{\|\mathbf{r}_{SA}\|^3} + \frac{\mu_\odot}{\|\mathbf{r}_{AO}\|^3} \right) \mathbf{r}_{SA}^T \\ & + \frac{1}{m_s} \mathbf{F}_{SRP}^T(\mathbf{r}_{AO}^T) + \frac{1}{m_s} R_{TS}^T \mathbf{F}_s^S. \end{aligned} \quad (30)$$

Additionally, for the sake of readability, we hereafter define the shorthand notation $Q \triangleq R_{GS}$, $\mathbf{q} = \mathbf{r}_{SG}^G$, $\mathbf{w} \triangleq \boldsymbol{\omega}_{GT}^G$, $\mathbf{r} \triangleq \mathbf{r}_{SA}^T$, $\mathbf{v} \triangleq \mathbf{v}_{SA}^T$, $R \triangleq R_{TS}$, $\mathbf{s} \triangleq \boldsymbol{\omega}_{ST}^S$, $\mathbf{d} \triangleq \mathbf{r}_{AO}^T$, $\mathbf{c} \triangleq \mathbf{r}_{AG}^G$, $\mathbf{g}(\mathbf{d}) \triangleq 1/m_s \mathbf{F}_{SRP}^T(\mathbf{d})$, $\mathbf{f} \triangleq 1/m_s \mathbf{F}_s^S$, which allow us to restate the equations of motion as

$$\dot{Q} = Q[\mathbf{s}]^\wedge - [\mathbf{w}]^\wedge Q, \quad (31)$$

$$\dot{\mathbf{q}} = QR^T \mathbf{v} - [\mathbf{w}]^\wedge QR^T \mathbf{r} \quad (32)$$

$$\dot{\mathbf{v}} = -\left(\frac{\mu_a}{\|\mathbf{r}\|^3} + \frac{\mu_\odot}{\|\mathbf{d}\|^3} \right) \mathbf{r} + \mathbf{g}(\mathbf{d}) + R\mathbf{f}. \quad (33)$$

$$\dot{\mathbf{r}} = \mathbf{v}. \quad (34)$$

4.5. Modeling assumptions

We note that we may view R and $\mathbf{s}$, along with $\mathbf{f}$, as inputs to the dynamical system described in (31)-(34). Here we consider $R = \hat{R} \exp([\mathbf{v}_R]^\wedge)$ as the already-filtered estimate value $\hat{R}$ of the spacecraft's orientation provided by the *inertial star tracker instrument*, perturbed on the right by its associated uncertainty $\mathbf{v}_R$, and $\mathbf{s} = \hat{\mathbf{s}} + \mathbf{v}_s$ is the filtered and unbiased estimate value $\hat{\mathbf{s}}$ of its angular velocity, provided by the *rate gyro instrument*, perturbed by the associated uncertainty $\mathbf{v}_s$. This inclusion allows us to fuse the known filtered measurements of a star tracker system and a rate gyro into the overall navigation solution. Moreover, attitude and angular rate filtered measurements may be obtained at high rate and may therefore be associated in time so to closely coincide with the moment an image is taken and processed.

By modeling the sensor dynamics we avoid having to represent the Newton-Euler dynamics of the spacecraft. Indeed, attitude control inputs, such as the actual actuated torques applied on the spacecraft, are often not known with precision at the instant when they are applied on the spacecraft, but may be estimated a-posteriori. Yet, these would be required at the moment of integrating the vehicle's Newton-Euler equations of motion if implemented on-the-fly.

Furthermore, following our assumption that the small-body is in single-axis rotation, we set the spin state vector $\mathbf{w}_0 \triangleq \mathbf{w}(t)$. Along with the unknown gravity parameter μ_a , we may view $\mathbf{w}_0$ as a parameter of the dynamical system.

For the purposes of the RelDyn factor modeling, we may now define the state tuple $x \triangleq (Q, \mathbf{q}, \mathbf{v}, \mathbf{r}) \in \text{SO}(3) \times \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3 \triangleq X$, the input tuple $u \triangleq (R, \mathbf{s}, \mathbf{f}) \in \text{SO}(3) \times \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3 \triangleq U$, and the parameter tuple $p \triangleq (\mathbf{w}, \mu_a) \in \mathbb{R}^3 \times \mathbb{R}_{>0} \triangleq P$. To be consistent with earlier representations, we note that by virtue of Fact 4, given any $Q, \mathbf{q}$, we may construct the corresponding pose $T = T(Q, \mathbf{q})$.

Recall that the RelDyn factor function encodes the odometric constraint by means of a likelihood or prior distribution function predicated on the state and the known accumulated observations at discrete times. Therefore, we proceed as follows: (1) first, we identify and quantify the probability distribution which relates the dispersion of the state evolving over time. We wish to be able to evaluate this distribution at a sequence of a priori unknown discrete times along the trajectory. We conserve two moments of this distribution, as is classically done in Gaussian process procedures (Dong et al., 2016); (2) we propagate the first moment (the mean state) by using the non-linear equations of motion, through an on-manifold integration technique; This provides us a means to evaluate the residual of the factor by comparing the propagated state against the next state; (3) we propagate the process covariance by linearizing the model dynamics around specific points, using a piecewise constant assumption, and then integrate the linear

model in the Lie algebra of the manifold. This provides us with a discrete process covariance matrix, which we use as a weighting in the factor formulation.

4.6. Stochastic differential equations of motion

We assume that the dispersion of the state at any given time is due to the accumulated effect of exogenous perturbations on the trajectory across time. Specifically, we wish to exploit the stochastic form of the equations of motion to derive the distribution of trajectories as a function of time. Furthermore, to be able to evaluate the distribution at any desired time, we require the stochastic differential equations to be in continuous form. Finally, we desire to obtain the relevant stochastic differential equations directly from the equations of motion, as further detailed below, by admitting realistic and physical perturbations through the input channels.

To this end, let the 3-dimensional white-noise Gaussian process $\mathbf{v}_*(t)$ (where $*$ = $R, \mathbf{s}, \mathbf{f}$), such that $\mathbb{E}[\mathbf{v}_*(t)] = \mathbf{0}$ and $\mathbb{E}[\mathbf{v}_*(t)\mathbf{v}_*^\top(\tau)] = W_*\delta(t - \tau)$, where W_* is the spectral density matrix of the continuous-time noise $\mathbf{v}_*$. Additionally, we assume that $\mathbf{v}_*$ for all $*$ = $R, \mathbf{s}, \mathbf{f}$ are mutually uncorrelated.

Consider the equations of motion given in (31)–(34) and substitute in u , where for $\hat{u} \triangleq (\hat{R}, \hat{\mathbf{s}}, \hat{\mathbf{f}}) \in U$, we have

$$u = (\hat{R}\exp([\mathbf{v}_R]^\wedge), \hat{\mathbf{s}} + \mathbf{v}_s, \hat{\mathbf{f}} + \mathbf{v}_f).$$

To establish the governing stochastic differential equations of the system (Sage and Melsa, 1971), we first define the 3-dimensional Wiener process $\mathbf{\epsilon}_*(t)$, ($*$ = $R, \mathbf{s}, \mathbf{f}$), such that

$$\mathbf{\epsilon}_*(t) = \int_0^t \mathbf{v}_*(\tau) d\tau, \quad \mathbf{\epsilon}_*(0) = \mathbf{0}, \quad (35)$$

with the increment $d\mathbf{\epsilon}_*(t) = \mathbf{\epsilon}_*(t + dt) - \mathbf{\epsilon}_*(t) = \mathbf{v}_*(t)dt$, satisfying $\mathbb{E}[\mathbf{\epsilon}_*(t)] = \mathbf{0}$, $\mathbb{E}[(\mathbf{\epsilon}_*(t) - \mathbf{\epsilon}_*(\tau))(\mathbf{\epsilon}_*(t) - \mathbf{\epsilon}_*(\tau))^\top] = W_*|t - \tau|$, which when $\tau \rightarrow t$ yields the relationship $\mathbb{E}[d\mathbf{\epsilon}_*d\mathbf{\epsilon}_*^\top] = W_*dt$. Note also that $d\mathbf{\epsilon}_*dt = 0$.

Given any two states $x_2 = (Q_2, \mathbf{q}_2, \mathbf{v}_2, \mathbf{r}_2) \in X$ and $x_1 = (Q_1, \mathbf{q}_1, \mathbf{v}_1, \mathbf{r}_1) \in X$, we define the error $\Delta_X(x_1, x_2) \in T_{x_1}X$ between x_1 and x_2 centered at x_1 (Speyer and Chung, 2008) such that

$$\Delta_X(x_1, x_2) \triangleq \begin{bmatrix} \log(Q_1^\top Q_2)^\vee \\ \mathbf{q}_2 - \mathbf{q}_1 \\ \mathbf{v}_2 - \mathbf{v}_1 \\ \mathbf{r}_2 - \mathbf{r}_1 \end{bmatrix}. \quad (36)$$

To obtain the stochastic differential equations resulting from this noise injection in the input, we evaluate the stochastic increment $d\mathbf{x} = \Delta_X(x(t), x(t + dt))$, we use Facts 2, 3, and 4 and we separate the equations, while dropping the time dependence for readability, yielding

$$d\mathbf{\kappa} = (Q^\top dQ)^\vee = (\hat{\mathbf{s}} - Q^\top \mathbf{w})dt + d\mathbf{\epsilon}_s, \quad (37)$$

$$d\mathbf{q} = \left(Q\hat{R}^\top \mathbf{v} - [\mathbf{w}]^\wedge Q\hat{R}^\top \mathbf{r} \right) dt + \left(Q[\hat{R}^\top \mathbf{v}]^\wedge - [\mathbf{w}]^\wedge Q[\hat{R}^\top \mathbf{r}]^\wedge \right) d\mathbf{\epsilon}_R, \quad (38)$$

$$d\mathbf{v} = \left(-\left(\frac{\mu_a}{\|\mathbf{r}\|^3} + \frac{\mu_\odot}{\|\mathbf{d}\|^3} \right) \mathbf{r} + \mathbf{g}(\mathbf{d}) + \hat{R}\hat{\mathbf{f}} \right) dt - \hat{R}[\hat{\mathbf{f}}]^\wedge d\mathbf{\epsilon}_R + \hat{R}d\mathbf{\epsilon}_f, \quad (39)$$

$$d\mathbf{r} = \mathbf{v}dt. \quad (40)$$

By decomposition of the covariances $W_* = L_*L_*^{-1}$, ($*$ = $R, \mathbf{s}, \mathbf{f}$), and by defining the standard Wiener process $\mathbf{\epsilon}$, where $\mathbb{E}[d\mathbf{\epsilon}(t)] = \mathbf{0}$ and $\mathbb{E}[d\mathbf{\epsilon}(t)d\mathbf{\epsilon}(t)^\top] = I_9dt$, obtain the system of stochastic differential equations (41).

$$\begin{aligned} d\mathbf{\kappa} &= f_Q(x, \hat{u}, p)dt + [0_{3 \times 3} \quad L_{Qs} \quad 0_{3 \times 3}]d\mathbf{\epsilon}, \\ d\mathbf{q} &= f_q(x, \hat{u}, p)dt + [L_{qR} \quad 0_{3 \times 3} \quad 0_{3 \times 3}]d\mathbf{\epsilon}, \\ d\mathbf{v} &= f_v(x, \hat{u}, p)dt + [L_{vR} \quad 0_{3 \times 3} \quad L_{vf}]d\mathbf{\epsilon}, \\ d\mathbf{r} &= f_r(x)dt, \end{aligned} \quad (41)$$

where $x = (Q, \mathbf{q}, \mathbf{v}, \mathbf{r}) \in X$, $\hat{u} = (\hat{R}, \hat{\mathbf{s}}, \hat{\mathbf{f}}) \in U$, and $p = (\mathbf{w}_0, \mu_a) \in P$, and where

$$\begin{aligned} L_{Qs} &= I_3, \quad L_{vR} = -\hat{R}[\hat{\mathbf{f}}]^\wedge, \quad L_{vf} = \hat{R}, \\ L_{qR} &= Q[\hat{R}^\top \mathbf{v}]^\wedge - [\mathbf{w}]^\wedge Q[\hat{R}^\top \mathbf{r}]^\wedge. \end{aligned}$$

Collecting into $d\mathbf{x} = [d\mathbf{\kappa}^\top d\mathbf{q}^\top d\mathbf{v}^\top d\mathbf{r}^\top]^\top$, and into $f(x, \hat{u}, p) = [f_Q^\top(x, \hat{u}, p) f_q^\top(x, \hat{u}, p) f_v^\top(x, \hat{u}, p) f_r^\top(x)]^\top$, we can rewrite (41) in succinct form as

$$d\mathbf{x} = f(x, \hat{u}, p)dt + L(x, \hat{u}, p)d\mathbf{\epsilon}. \quad (42)$$

where

$$L = \begin{bmatrix} 0_{3 \times 3} & L_s & 0_{3 \times 3} \\ L_{qR} & 0_{3 \times 3} & 0_{3 \times 3} \\ L_{vR} & 0_{3 \times 3} & L_{vf} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix}. \quad (43)$$

Due to the non-linearity of the considered dynamical system in equation (42), the distribution of the Itô process $x(t)$ may require more than two moments to fully describe at any time. However, as a stepping stone to obtain a practical representation of the distribution $p(x(t)|x(t_0))$, we are interested in computing the first two moments $\hat{x}(t) = (\hat{Q}(t), \hat{\mathbf{q}}(t), \hat{\mathbf{v}}(t), \hat{\mathbf{r}}(t)) \triangleq \mathbb{E}[x(t)]$ and $\Sigma(t) \triangleq \mathbb{E}[\Delta_X(\hat{x}(t), x) \Delta_X^\top(\hat{x}(t), x)]$ at each time t .

Naturally, we want the relative dynamics factor to be proportional to the distribution $p(x_{k+1}|x_k)$, as shown in the beginning of the Section 4.3. Noting that the time step between t_k and t_{k+1} is not known a-priori during the navigation segment, it is crucial to compute a process noise

covariance P_k , which appropriately scales with the time step length, using a discretization scheme. Such a treatment is similar to the prediction step of an Extended Kalman Filter with first-order approximation assumptions, and can be referenced in [Sage and Melsa \(1971\)](#).

First, for the continuous case, we define the RelDyn factor residual ϵ^{RelDyn} such that

$$\epsilon^{\text{RelDyn}}(x(t), x(t+dt)) \triangleq \mathbf{dx}(t) - f(x(t), \hat{u}(t), p)dt.$$

Then, the factor ϕ^{RelDyn} , as a function proportional to $\mathcal{N}(\mathbf{0}, dP(t))$, where

$$dP(t) = \mathbb{E}[(\mathbf{dx}(t) - f(x(t), \hat{u}(t), p))(\cdot)^\top],$$

may be simply defined as

$$\begin{aligned} \phi^{\text{RelDyn}}(x(t), x(t+dt)) \\ \triangleq \exp\left((\epsilon^{\text{RelDyn}}(x(t), x(t+dt)))^\top dP^{-1}(t) \epsilon^{\text{RelDyn}}(*)\right). \end{aligned}$$

Let the partition $\{t_k\}_{k=0}^N$ of the time interval $[t_0, t_f]$ be such that $t_0 < t_1 < \dots < t_k < \dots < t_N = t_f$. Given some $\hat{u}(t), t \in [t_0, t_f]$, we generate the sequence of predictions $\{\hat{x}_k\}_{k=0}^N$ such that

$$\begin{aligned} \Delta_X(\hat{x}_{k+1}, \hat{x}_k) \\ = \int_{t_k}^{t_{k+1}} \hat{f}(\hat{x}(\tau), \hat{u}(\tau), p) d\tau, \quad \hat{x}(t_k) = \hat{x}_k, \end{aligned} \quad (44)$$

is a valid discretization of the first-order approximation ([Gelb et al., 1974](#)) $\hat{\dot{x}}(t) \approx \hat{f}(\hat{x}(t), \hat{u}(t), p)$ for all $k = 1, \dots, N-1$, and such that

$$\begin{aligned} P_k = \int_{t_k}^{t_{k+1}} L(\hat{x}(\tau), \hat{u}(\tau), p) L^\top(\hat{x}(\tau), \hat{u}(\tau), p) d\tau, \\ x(t_k) = x_k, \end{aligned} \quad (45)$$

characterizes the discrete process noise over the time-span $[t_k, t_{k+1}]$. Then, using the definition in equation (36) and the equation (44), we define the RelDyn residual $\epsilon_k^{\text{RelDyn}}$ for the discrete case, as

$$\begin{aligned} \epsilon_k^{\text{RelDyn}}(x_k, x_{k+1}) \\ \triangleq \begin{bmatrix} \log\left((Q_k^\top Q_{k+1})^\top \Omega_k^{\text{prop}}\right)^\vee \\ \mathbf{q}_{k+1} - \mathbf{q}_k - \int_{t_k}^{t_{k+1}} f_q(x(\tau), \hat{u}(\tau), p) d\tau \\ \mathbf{v}_{k+1} - \mathbf{v}_k - \int_{t_k}^{t_{k+1}} f_v(x(\tau), \hat{u}(\tau), p) d\tau \\ \mathbf{r}_{k+1} - \mathbf{r}_k - \int_{t_k}^{t_{k+1}} f_r(x(\tau), p) d\tau \end{bmatrix}, \end{aligned} \quad (46)$$

where $x(t_k) = (Q_k, \mathbf{q}_k, \mathbf{v}_k, \mathbf{r}_k)$ and where Ω_k^{prop} is defined such that, for a partition $t_k = s_0 < s_1 < \dots < s_n = t_{k+1}$,

$$\begin{aligned} \Omega_k^{\text{prop}} \triangleq \lim_{n \rightarrow \infty} \prod_{i=0}^{n-1} \exp\left(\int_{s_i}^{s_{i+1}} [f_Q(x(\tau), \hat{u}(\tau), p)]^\wedge d\tau\right), \\ x(t_k) = x_k, \end{aligned}$$

and is dubbed the McKean-Gangolli injection method ([Chirikjian, 2011](#)), allowing for the stochastic process on the Lie group $\text{SO}(3)$ to be written as a product integral. With appropriate correction factors, the limit can be truncated to numerically perform on-manifold integration with minimal error accumulation stemming from the approximation ([Andrieu and Crassidis, 2013](#)), as discussed in [Section 5.6](#) below.

The factor ϕ_k^{RelDyn} for the discrete case may now be written as

$$\phi_k^{\text{RelDyn}}(x_k, x_{k+1}) \triangleq \exp\left(\left\|\epsilon_k^{\text{RelDyn}}(x_k, x_{k+1})\right\|_{P_k^{-1}}^2\right).$$

4.7. Smoothability of chosen state variables

It has been shown in [Sage and Melsa \(1971\)](#) that batch-style maximum a-posteriori solution to the factorization of a joint distribution over the variables to be estimated, which was discussed in [Section 4](#), is equivalent to the solution obtained from optimal fixed-interval smoothing. Typically, the optimal smoothed state is defined as a linear combination of the state of the *forward filter* and the state of the *backward filter* at each time, with optimally chosen weights. A state is said to be *smoothable* if an optimal smoother provides a state estimate superior to that obtained when the final optimal filter estimate is extrapolated backwards in time ([Gelb et al., 1974](#)). Furthermore, [Fraser \(1967\)](#) has shown that, given linear forward and backward filters, only those states which are controllable by the noise driving the system state vector are smoothable.

In this context, we evaluate the relevance of our choice of state variables and the noise input and derived stochastic equations of motion in [Section 4.6](#) in terms of smoothability. However, since the concept of smoothability pertains to linear smoothing, direct application of the criterion to our problem is not possible. Instead, by demonstrating small-time local controllability through the noise by analyzing a local linearization of the equations (31) to (33) at some $(\hat{x}_k, \hat{u}_k)$, we can assess the smoothability of our chosen state variables at that point.

Specifically, we can compute the local linearization matrices

$$F_k \triangleq \frac{\partial f}{\partial x} \Big|_{\substack{x = \hat{x}_k \\ u = \hat{u}_k \\ p = \hat{p}}} = \begin{bmatrix} \frac{\partial f_Q}{\partial \mathbf{q}} & 0 & 0 & 0 \\ \frac{\partial f_q}{\partial \mathbf{q}} & 0 & \frac{\partial f_q}{\partial \mathbf{v}} & \frac{\partial f_q}{\partial \mathbf{r}} \\ 0 & 0 & 0 & \frac{\partial f_v}{\partial \mathbf{r}} \\ 0 & 0 & \frac{\partial f_r}{\partial \mathbf{v}} & 0 \end{bmatrix} \begin{matrix} x = \hat{x}_k \\ u = \hat{u}_k \\ p = \hat{p} \end{matrix}, \quad (47)$$

where

$$\begin{aligned}\frac{\partial f_Q}{\partial \kappa} &= [Q^\top \mathbf{w}]^\wedge, [\kappa]^\wedge \in \mathcal{T}_{Q\text{SO}(3)}, \\ \frac{\partial f_q}{\partial \kappa} &= -Q[R^\top \mathbf{v}]^\wedge + [\mathbf{w}]^\wedge Q[R^\top \mathbf{r}]^\wedge \\ \frac{\partial f_q}{\partial \mathbf{v}} &= QR^\top, \frac{\partial f_q}{\partial \mathbf{r}} = -[\mathbf{w}]^\wedge QR^\top, \\ \frac{\partial f_r}{\partial \mathbf{v}} &= \mathbf{I}_3, \frac{\partial f_v}{\partial \mathbf{r}} = \frac{\mu_a}{\|\mathbf{r}\|^5} (3\mathbf{r}\mathbf{r}^\top - \|\mathbf{r}\|^2 \mathbf{I}_3) - \frac{\mu_\odot}{\|\mathbf{d}\|^3} \mathbf{I}_3\end{aligned}$$

and $L_k = L(\hat{\mathbf{x}}_k, \hat{\mathbf{u}}_k, p)$, where L is as defined in (43).

We now apply the PBH controllability test (Fraser, 1967; Williams and Lawrence, 2007) to show that our system does have small-time local smoothability at all points of interest in the state space X .

For simplifying purposes, since both $\hat{Q}$ and $\hat{R}$ are full rank, next we assume $\hat{Q} = \hat{R} = \mathbf{I}_3$ without loss of generality. Specifically, we see that the matrix $\frac{\partial f_Q}{\partial \kappa} = [\mathbf{w}]^\wedge$ is skew-symmetric and is therefore of rank 2. It follows that $\lambda = 0$ is an eigenvalue of F_k . We compute the matrix $[F_k - \lambda \mathbf{I}_3, L_k]$, which for $\lambda = 0$ takes the form

$$[F_k, L_k] = \begin{bmatrix} [\hat{\mathbf{w}}]^\wedge & 0 & 0 & 0 & 0 & \mathbf{I}_3 & 0 \\ [\hat{\mathbf{v}}]^\wedge + [\hat{\mathbf{w}}]^\wedge [\hat{\mathbf{r}}]^\wedge & 0 & \mathbf{I}_3 & [\hat{\mathbf{w}}]^\wedge & [\hat{\mathbf{v}}]^\wedge - [\hat{\mathbf{w}}]^\wedge [\hat{\mathbf{r}}]^\wedge & 0 & 0 \\ 0 & 0 & 0 & \mathbf{I}_3 & -[\hat{\mathbf{f}}]^\wedge & 0 & \mathbf{I}_3 \\ 0 & 0 & \mathbf{I}_3 & 0 & 0 & 0 & 0 \end{bmatrix}$$

We observe by inspection that, as long as $\hat{\mathbf{v}}$ and $\hat{\mathbf{w}}$ are not null at the same time, that is, that the spacecraft has some non-null relative velocity in the small-body fixed frame, then the PBH controllability matrix has rank 12, and thus the system is small-time locally controllable through the noise.

Therefore, it is crucial to include appropriate and physically justified noise terms in the considered dynamics to support smoothability. Furthermore, the inclusion of such noise avoids the degeneracy of the discrete process noise Gaussian distribution, with covariance P_k , as computed in Section 4.3. This argument supports the injection of noise through the input channel in the derivation of the stochastic equations of motion in Section 4.3.

5. Algorithm and implementation details

We are now ready to finally state the full solution to the problem and provide the key implementation details. Equations (44) and (45) constitute constraints on the evolution of the mean system state and its associated system noise covariance between time instances t_k and t_{k+1} . We wish to enforce these constraints. Practically, at each time instance t_k , $k = 1, \dots, N_s$ we let $\tilde{\mathbf{x}}_k$ be the current best guess

of $\hat{\mathbf{x}}_k$ and $\tilde{P}_k$ be the current best guess of P_k . Given $\tilde{\mathbf{x}}_k, \tilde{\mathbf{x}}_{k+1}$ and $\hat{\mathbf{u}}(t), t \in [t_k, t_{k+1}]$, we compute the residual due to equation (44), then the guess matrix $\tilde{P}_k$, and finally the value of the factor $\phi_k^{\text{RelDyn}}(\tilde{\mathbf{x}}_k, \tilde{\mathbf{x}}_{k+1})$. Note that the integrals in equations (46) and (45) are computed numerically using an on-manifold integration scheme, detailed later in this section. We note that the inclusion of the RelDyn factor introduces the spin state $\mathbf{w}_0$ and the gravity parameter μ_a as variables into the factor graph.

As both the prior and projection factors are developed in detail in other works (see Dellaert and Kaess, (2017), for example), we simply restate them here using our notation, such that

$$\begin{aligned}\phi^{\text{prior}}(x_0) &\triangleq \exp\left(\left\|\Delta_X(\hat{\mathbf{x}}_0, x_0)\right\|_{\Sigma_0^{-1}}^2\right), \\ \phi_*^{\text{proj}}(x_k, \ell_i) &\triangleq \exp\left(\left\|\mathbf{y}_{ik}^{\text{m}} - \mathbf{y}_{ik}\right\|_{(\Sigma_Y^{\text{m}})^{-1}}^2\right),\end{aligned}$$

where each $\mathbf{y}_{ik}^{\text{m}} \in \mathcal{Y}_k$ is obtained as per the relationship in equation (6). We incorporate these prior factors on the initial state tuple. We finally incorporate the relative kinematics factors introducing the center-of-mass position parameter vector $\mathbf{c}$.

The structure of the problem may now be organically schematized and analyzed as a factor graph. We have illustrated two initial steps of the AstroSLAM factor graph in Figure 3, where the RelKin factors and their related dependencies are highlighted in red, and the RelDyn factors and their related dependencies are highlighted in blue.

Next, we provide the implementation details of the algorithm, while underlining the specificities of the asteroid navigation problem. We first discuss the issue of complexity growth in the graph induced by the inclusion of global parameters. Then we discuss the state and map initialization steps of the algorithm, in which we insert prior factors for initial poses and kinematic variables of both the spacecraft and small-body, as well as generate an initial estimate for the map Ψ_1 .

Finally, we discuss the overall architecture of AstroSLAM. Our processing pipeline consists of a “front-end” system and a “back-end” system predicated on the iSAM2 engine and GTSAM library, along with an initialization step and a loop closure detection step. We further illustrate the overall architecture of the front-end, back-end, and loop closure subsystems in Figure 6 for the convenience of the reader.

5.1. Considerations regarding complexity growth

Since we want to estimate the navigational states and simultaneously update the center of mass position $\mathbf{c}$, the spin vector $\mathbf{w}_0$, and the gravity parameter μ_a estimates on-the-fly, we have introduced these as variables in the factor graph, by means of the relative kinematic factor and RelDyn factor

formulations. It is clear, however, that the inserted edges render the graph fully connected, since each state tuple x_i , $i = 1, \dots, n$ is linked to the single parameter μ_a by means of the RelDyn factor $\phi_{i-1}^{\text{RelDyn}}$ and the factor $\phi_{i+1}^{\text{RelDyn}}$. Due to

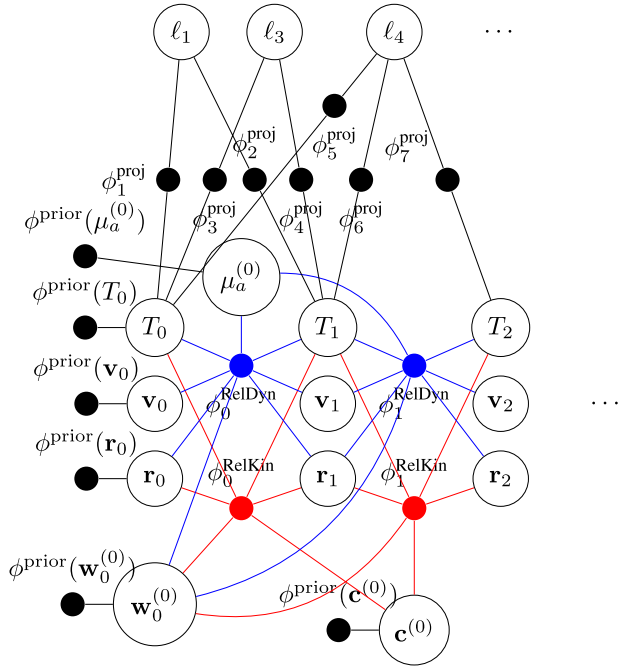


Figure 3. Full AstroSLAM factor graph. Red edges correspond to dependencies related to the relative kinematic factor, denoted as RelKin, and blue edges correspond to the dependencies related to the RelDyn factor.

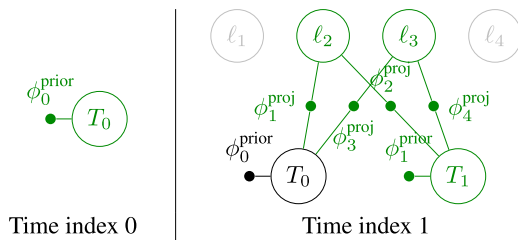


Figure 4. Initialization steps of the SLAM problem.

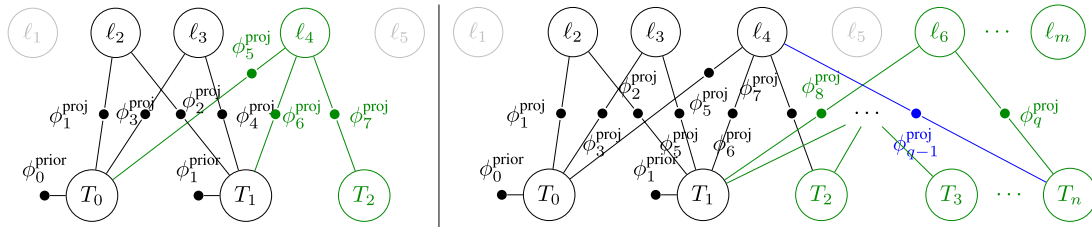


Figure 5. Front-end pipeline factor-graph following the initialization of SLAM. On the left is factor graph at time step index 2, with the subgraph related projection factors added, and on the right for time steps 3 and beyond. The edges, factors and variables colored in green relate to the generic projection factors added for newly triangulated landmarks, while those in blue relate to projection factors added for previously known landmarks. In gray are the potential landmarks that have had sightings in the past, but which have not been initialized yet.

this, the clique sizes in the Bayes tree determined at each solution step of iSAM2 would get larger as the graph grows incrementally, and the updates in the Bayes tree would involve cliques that are ever more deep in the tree. If not addressed, this issue essentially forces iSAM2 to perform expensive batch optimization at every time step, with the size of the batch update growing at every time step (Setterfield et al., 2018a; Terán Espinoza, 2021). One intuitive workaround, as implemented in Setterfield et al. (2018a), is to solve for the variables related to the inertial factors in an incremental fashion at every step, but only sporadically optimize the kinematic rotation factors using a batch solve step, so to avoid the repeated and expensive batch optimization. However, we cannot implement such a method for our procedure, as our relative odometry RelDyn factor also depends on the global parameter μ_a .

In our approach, we somewhat emulate the procedure of a fixed-lag smoother, by defining epochs $i = 1, \dots, m$ to which we associate specific global variables $\mathbf{c}^{(i)}$, $\mathbf{w}_0^{(i)}$ and $\mu_a^{(i)}$. Over the epoch, the iSAM2 optimizer performs a batch update of the variable only related to that epoch at each time step, by virtue of the fully connected property of the graph induced by the relative kinematic factor and the RelDyn factors. At every new epoch i , we marginalize the global parameter variables $\mathbf{c}^{(i-1)}$, $\mathbf{w}_0^{(i-1)}$, and $\mu_a^{(i-1)}$ from the previous epoch, and insert a new prior factor for each of the current global parameter variables $\mathbf{c}^{(i)}$, $\mathbf{w}_0^{(i)}$, and $\mu_a^{(i)}$. When inserting relative kinematic factors or RelDyn factors during epoch i , we associate these factors to their corresponding epoch's global parameter variables $\mathbf{c}_i$ and $\mu_{a,i}$. A new epoch is induced at a fixed time step interval (e.g., every 10 steps).

5.2. State initialization

We leverage estimates from pre-encounter Earth-based measurements and approach phase sensor measurements to perform the initialization of both the spacecraft and small-body states. Inertial position measurements $\mathbf{r}_{S_kO}^{\mathcal{I},m}$ of the spacecraft, modeled as $\mathbf{r}_{S_kO}^{\mathcal{I},m} \triangleq \mathbf{r}_{S_kO}^{\mathcal{I}} + \mathbf{v}_r$, $\mathbf{v}_r \sim \mathcal{N}(\mathbf{0}_{3 \times 1}, \Sigma_r^m)$, are based on Earth-relative radiometric ranging and bearing measurements, a method of localization widely

practiced in deep space mission spacecraft tracking using communication station networks, such as the DSN. Accurate ground-based navigation estimates using Earth-relative range and bearing measurements, such as uplink-downlink pulse ranging and Delta-Differenced One-Way Ranging (DDOR), a type of Very Long Baseline Interferometry (VLBI) (Miller, 2019; Miller and Rourke, 1977), are available to be used for initialization. It is important to note that our algorithm requires DSN-type measurements only during the initialization phase, to anchor the initial pose as described below.

Inertial position estimates of the small-body, denoted by $\hat{\mathbf{r}}_{G_kO}^{\mathcal{I}}$, are predicated on orbit determination (OD) performed using Earth-based telescopic measurements. An initial relative position vector $\hat{\mathbf{r}}_{S_0G}^{\mathcal{I}}$ estimate may be obtained by differencing spacecraft and small-body inertial position estimates. Alternatively, relative optical navigation (OpNAV) performed using pre-encounter sensor measurements, accurate in the order of several hundreds of meters for large small-bodies and tens of meters for small small-bodies (Bhaskaran et al., 2011), may be used to provide such an initial prior on the relative position.

To further simplify the initialization step, we assume that the small body is a stable single-axis rotator. Thus, we align the arbitrary frame $\mathcal{G}$ such that $\boldsymbol{\omega}_{\mathcal{G}O} = w_0 \bar{\mathbf{g}}_3$, where w_0 is the magnitude of the angular velocity vector of the small-body. The unit vector $\bar{\mathbf{g}}_3$ is classically parameterized by the spin pole tilt angles relative to the J2000 ecliptic plane. If sufficient pre-encounter observations are available, a prior for these angles can be estimated and their associated uncertainty computed (Thomas et al., 1997).

Typically, a prominent and salient feature on the surface is hand-picked as the prime meridian direction, thus fixing the $\bar{\mathbf{g}}_1$ axis. In our case, given the initial prior $\mathbf{r}_{S_0G_0}^{\mathcal{I}}$ discussed earlier, the prime meridian $\bar{\mathbf{g}}_1$ may be initialized by first computing $\bar{\mathbf{g}}_2$ following $\bar{\mathbf{g}}_2^{\mathcal{I}} = [\bar{\mathbf{g}}_3^{\mathcal{I}}]^\wedge \mathbf{r}_{S_0G_0}^{\mathcal{I}} / \|\mathbf{r}_{S_0G_0}^{\mathcal{I}}\|$, and then computing $\bar{\mathbf{g}}_1^{\mathcal{I}} = [\bar{\mathbf{g}}_2^{\mathcal{I}}]^\wedge \bar{\mathbf{g}}_3^{\mathcal{I}}$. Consequently, $R_{\mathcal{I}G_0} = [\bar{\mathbf{g}}_1^{\mathcal{I}} \ \bar{\mathbf{g}}_2^{\mathcal{I}} \ \bar{\mathbf{g}}_3^{\mathcal{I}}]$ is determined. Note that, for this method to work, the spacecraft relative position at initialization time cannot be coincident with the small-body spin axis. In practice, an on-board star tracker system is leveraged to obtain an orientation measurement $R_{\mathcal{I}S_k}^m$, modeled as $R_{\mathcal{I}S_k}^m \triangleq R_{\mathcal{I}S_k} \exp([\mathbf{v}_R]^\wedge)$, $\mathbf{v}_R \sim \mathcal{N}(\mathbf{0}_{3 \times 1}, \Sigma_R^m)$. Orientation measurements are usually known with very good accuracy and little uncertainty.

We may now establish the prior factor for relative pose $T_{G_0S_0}$ by combining the information we have available at initialization. We first define the relative measurement at time $k = 0$, using (4), given by

$$\begin{aligned} T_{G_0S_0}^m &= \begin{bmatrix} \hat{R}_{\mathcal{I}G_0}^\top R_{\mathcal{I}S_0}^m & \hat{R}_{\mathcal{I}G_0}^\top \mathbf{r}_{S_0G_0}^{\mathcal{I},m} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \\ &= \begin{bmatrix} \hat{R}_{\mathcal{I}G_0}^\top R_{\mathcal{I}S_0} \exp([\mathbf{v}_R]^\wedge) & \mathbf{r}_{S_0G_0}^{G_0} + \hat{R}_{\mathcal{I}G_0}^\top \mathbf{v}_R \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}, \\ &= \begin{bmatrix} \hat{R}_{\mathcal{I}G_0}^\top R_{\mathcal{I}S_0} & \mathbf{r}_{S_0G_0}^{G_0} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \begin{bmatrix} \exp([\mathbf{v}_R]^\wedge) & R_{\mathcal{I}S_0}^\top \mathbf{v}_R \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \end{aligned}$$

where $\mathbf{r}_{S_0G}^{\mathcal{I},m}$ is the relative position vector at time $k = 0$ obtained by differencing inertial position estimates of the spacecraft and the small-body, or by means of an OpNav solution. It then follows that

$$T_{G_0S_0}^{-1} T_{G_0S_0}^m = \begin{bmatrix} \exp([\boldsymbol{\varepsilon}_1]^\wedge) & \boldsymbol{\varepsilon}_2 \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}, \quad (48)$$

where $\boldsymbol{\varepsilon}_1 = \mathbf{v}_R$ and $\boldsymbol{\varepsilon}_2 = R_{\mathcal{I}S_k}^\top \mathbf{v}_R$. Using the logarithm map of the SE(3) group of homogeneous transformations at the identity and the *vee* operator, as detailed in Section 2.1, we can write

$$\left[\log \left(T_{G_0S_0}^{-1} T_{G_0S_0}^m \right) \right]^\vee = \begin{bmatrix} \boldsymbol{\varepsilon}_1 \\ \boldsymbol{\varepsilon}_2 \end{bmatrix} \sim (\mathbf{0}_{6 \times 1}, \Sigma_{T,0}^m), \quad (49)$$

where, by first-order linear approximation, we have $\Sigma_{T,0}^m = J_R \Sigma_R^m J_R^\top + J_r (\hat{R}_{\mathcal{I}G_0}^\top R_{\mathcal{I}S_0}^m) \Sigma_r^\top (\hat{R}_{\mathcal{I}G_0}^\top R_{\mathcal{I}S_0}^m)$, computed using Fact 4 where $R \in \text{SO}(3)$. A prior factor $\phi_0^{\text{prior}}(T_{G_0S_0})$ is emplaced in the graph. This factor encodes the residual between the pose $T_{G_0S_0} \in \text{SE}(3)$ and the measurement $T_{\mathcal{I}S_0}^m$, with covariance $\Sigma_{T,0}^m$.

5.3. Map initialization

Initialization of the map is delayed until time index $k = 1$, at which point at least two images of the target with sufficient parallax are captured. Local image features Υ_0 and Υ_1 are extracted and undergo data association, with outlier rejection, producing a set of 2D-2D correspondences. A strict outlier rejection criterion is used to obtain a subset of reliable correspondences. It is now possible to apply a typical 8-point algorithm (Hartley and Zisserman, 2004) using the inlier 2D-2D correspondences to find a guess pose transformation $\hat{T}^\circ \in \text{SE}(3)$ such that

$$\hat{T}^\circ \triangleq \begin{bmatrix} \hat{R}^\circ & \hat{\mathbf{t}}^\circ \\ \mathbf{0} & 1 \end{bmatrix},$$

where $\hat{R}^\circ \in \text{SO}(3)$ and $\hat{\mathbf{t}}^\circ \in \mathbb{R}^3$, describe the estimated change in relative orientation and relative position between the poses of the camera frame at time indices $k = 0$ and $k = 1$. From the frame kinematics and composition rules, and knowing that $\hat{R}^\circ = \hat{R}_{C_0C_1}$, it follows that $\hat{R}_{\mathcal{G}_1S_1} = \hat{R}_{\mathcal{G}_1G_0} \hat{R}_{G_0S_0} R_{SC} \hat{R}^\circ R_{CS}$. We note that $\hat{R}_{G_0G_1} \approx \exp([\bar{\boldsymbol{\omega}}_{G_0\mathcal{I}}^\top \Delta t_0]^\wedge)$. It follows that $\hat{R}_{\mathcal{G}_1S_1} \approx \exp([\bar{\boldsymbol{\omega}}_{G_0\mathcal{I}}^\top \Delta t_0]^\wedge) \hat{R}_{G_0\mathcal{I}} \hat{R}_{\mathcal{I}S_0}^m R_{SC} \hat{R}^\circ R_{CS}$. Additionally, we know that

$$\begin{aligned}
\mathbf{t}^o &= \lambda \mathbf{r}_{C_1 C_0}^{C_0} \\
&= \lambda R_{CS}(\mathbf{r}_{CS}^S + R_{S_0 G_0} R_{G_0 G_1} \mathbf{r}_{S_1 G}^{G_1} - (\mathbf{r}_{CS}^S + R_{S_0 G_0} \mathbf{r}_{S_0 G}^{G_0})) \\
&= \lambda R_{CS} \hat{R}_{S_0 G_0} \left(\exp \left(\left[\hat{\boldsymbol{\omega}}_{G_0 L}^{G_0} \Delta t_0 \right]^\wedge \right) \mathbf{r}_{S_1 G}^{G_1} - \mathbf{r}_{S_0 G}^{G_0} \right),
\end{aligned} \tag{50}$$

with $\lambda > 0$, an unknown scaling factor. We see that the ambiguity in the scale parameter λ still remains. Typically, scale can be established using off-nadir altimeter measurements over a range of orbital configurations in a multi-arc solution, as performed by the OSIRIS-REx mission around asteroid Bennu (Goossens et al., 2021). However, the latter solution, in its current format, is not amenable to on-the-fly autonomy, as the batch estimation process requires a compilation of data from multiple mission arcs. Furthermore, altimeter measurements have to be used at close range to the target small-body, again restricting autonomy since judiciously pre-designed maneuvers have to be executed to first obtain the necessary altimeter measurements at close range. If a good initial estimate of the angular velocity $\mathbf{w}_0$ of the small-body is available, the map scale can be determined by combining DSN-like inertial position measurements with the orientation matrix and position vectors extracted from the relative poses T_0 and T_1 , as shown in equation (50). The availability of such prior knowledge currently depends heavily on Earth-based and pre-encounter measurements of the target small-body. Regarding this matter, as detailed in Goossens et al. (2021), the ambiguity in scale due to the parameter λ is determining in the estimation of an appropriate spherical gravity parameter term μ_a . An error in μ_a will then affect the propagation step of the smoothing process. For the purpose of this work, we have assumed that the scale parameter λ is well-known. Future work will develop novel autonomous methods to estimate the scale.

Having an estimate of poses $\hat{T}_0$ and $\hat{T}_1$, we compute the triangulation of the landmarks Ψ_1 using measurements $\mathbf{Y}_0$ and $\mathbf{Y}_1$. We then generate guess values for the estimated landmark positions $\{\mathbf{r}_{LG}^G\}_{L \in \Psi_1}$. At this step, all appropriate factors are inserted based on the 2D-to-3D correspondences, resulting in the factor graphs illustrated in Figure 4, where the shorthands $\ell_i \triangleq \mathbf{r}_{LG}^G$, $L_i \in \Psi_k$ and $T_k \triangleq T_{G_k S_k}$, $k = 0, \dots, n$ are used for brevity.

5.4. Front-end

The front-end system includes feature detection and matching, and encodes the structure of a typical monocular bundle adjustment pipeline in a factor graph based on index mappings defined in Section 2.3.

At every new image acquisition, first, we store the corresponding wall clock time, as the difference between two consecutive image acquisition times is necessary for

propagating the RelDyn factors inserted later in the factor graph. We then detect ORB features (Rublee et al., 2011) in the new image. ORB features perform well in practice and they are fast to compute. These are also good placeholders for more robust automatic features to be implemented in the future. We use brute force nearest neighbor search based on the Hamming distance for binary feature descriptors to perform feature matching against previous image frames (e.g., $n - 1$, $n - 2$) and populate an initial 2D-to-2D correspondence map. We also enforce an essential matrix constraint as a geometric check for matched features per a RANSAC procedure, thus eliminating outlier associations in the 2D-to-2D correspondence map. We append the corresponding matches to a list of feature tracks maintained since the beginning of the algorithm procedure.

All tracked feature points in the current frame n may be separated into three subsets: a set of sightings for new landmark points to be triangulated, a set of sightings of known landmark points to be tracked, and a set of outlier points to be rejected. The outlier set of feature points is discarded.

The first subset of the tracked features corresponds to measurements of surface point landmarks that have not been previously sighted and which need to be triangulated using all accumulated sightings in the feature track. Leveraging the latest guess pose $\tilde{T}_{G_n C_n}$, a guess value for the 3D position of the landmarks is generated using triangulation. We delay triangulation and insertion of new landmarks into the graph based on the number of sightings of the associated landmark. If that number is above a predetermined threshold, say 3, then we attempt triangulation of the point using all of the measurements in the sighting of that surface point. If the landmark point has been successfully triangulated using the feature point sightings that were used for the triangulation, appropriate 2D-to-3D correspondences capturing the new data association are accumulated. Projection factors between the new landmark variable and all camera poses where the landmark was sighted are added to the factor graph.

The second subset of tracked features corresponds to new sightings of surface point landmarks previously triangulated. This subset is populated by verifying if the currently tracked feature has an associated landmark already in the 2D-to-3D correspondences maps. The feature-landmark matched pairs undergo a reprojection error test, after which the surviving pairs establish new GTSAM projection factors inserted into the graph between known landmarks and relating to the most recent frame n . If there are enough tracked features from time index $k = n - 1$, then visual tracking is successful and the well-known PnP algorithm (Lepetit et al., 2009) is used to guess the camera pose $\tilde{T}_{G_n C_n}$ value from matched correspondences. Alternatively, when RelDyn factors are included and if the PnP solution is of poor quality, we use the prediction from the motion model at time t_n to guess a new camera pose $\tilde{T}_{G_n C_n}$.

5.5. Back-end

For every new frame inserted, the graph is incrementally augmented with the new variables and factors by the front-end, as is illustrated in Figure 5.

The underlying structure of the navigation problem is thus captured by encoding visual SLAM measurement constraints, relative kinematic constraints, Earth-relative inertial position measurement constraints as factors, and RelDyn factors in a single factor graph, as we explained in Section 4.1.

In practice, the optimization associated with the small-body relative navigation problem is performed incrementally in the factor graph framework using the iSAM2 (Kaess et al., 2012) algorithm. The back-end implements iSAM2 algorithm's procedure, which was discussed earlier in Section 4.1. iSAM2 evaluates the losses at each factor marked for update, computes the associated Jacobians at the guess values, and performs the minimization for inference (Dellaert and Kaess, 2017), yielding a new estimate solution. Note that since the process noise covariance related to the RelDyn factor, as detailed in Section 4.6, is not fixed, but is rather the solution to a dynamical equation, the out-of-the-box GTSAM factor template had to be modified to allow for a variable process noise covariance to be incorporated accordingly. This procedure deviates from the pre-integrated IMU accelerometer factor (Forster et al., 2016), since our case, the integrated quantities are non-linear in terms of the state. Indeed, at each evaluation of the factor error, the discrete process noise covariance, also a non-linear function of the state, is propagated accordingly. This covariance is associated with a Gaussian process distribution, and once computed, we replace GTSAM's internal representation of the covariance of the factor error with new process noise covariance.

5.6. Computation of the RelDyn factor residual

Inspired by the procedure in Forster et al. (2016), we wish to compute numerically integrated quantities between successive states, so to evaluate the RelDyn factor error established in Section 4.3. To this end, we perform on-manifold discrete Crouch-Grossman geometric integration as detailed by Andrieu and Crassidis (2013) to obtain accurate predictions which also respect the constraints of the relative rotation matrix $Q(t) \in \text{SO}(3)$. Given $\{t_k\}_{k=0}^N$, for every $k=0, \dots, N-1$, let $\Delta t_k \triangleq t_{k+1} - t_k$, we use an N_s -stage Crouch-Grossman geometric integration scheme and compute the propagated quantities

$$\tilde{\Omega}_k^{\text{prop}} \triangleq \prod_{i=1}^{N_s} \exp\left(\left[b_i \Delta t_k \tilde{f}_Q^{(i)}\right]^\wedge\right), \quad (51)$$

$$\begin{bmatrix} \Delta \tilde{\mathbf{q}}_k^{\text{prop}} \\ \Delta \tilde{\mathbf{v}}_k^{\text{prop}} \\ \Delta \tilde{\mathbf{r}}_k^{\text{prop}} \end{bmatrix} \triangleq \sum_{i=1}^{N_s} b_i \Delta t_k \begin{bmatrix} \tilde{f}_{\mathbf{q},k}^{(i)} \\ \tilde{f}_{\mathbf{v},k}^{(i)} \\ \tilde{f}_{\mathbf{r},k}^{(i)} \end{bmatrix}, \quad (52)$$

whereby the shorthands

$$\tilde{f}_{*,k}^{(i)} \triangleq f_{*}^{(i)}(\tilde{x}_k^{(i)}, \tilde{u}_k^{(i)}, p), \quad * = Q, \mathbf{q}, \mathbf{v}, \mathbf{r} \quad (53)$$

$$\tilde{x}_k^{(i)} \triangleq \left(\tilde{Q}_k^{(i)}, \tilde{\mathbf{q}}_k^{(i)}, \tilde{\mathbf{v}}_k^{(i)}, \tilde{\mathbf{r}}_k^{(i)} \right), \quad (54)$$

$$\tilde{u}_k^{(i)} \triangleq \tilde{u}(t_k + c_i \Delta t_k), \quad (55)$$

$$\tilde{Q}_k^{(i)} \triangleq \tilde{Q}_k \prod_{j=1}^{i-1} \exp\left(\left[a_{ij} \Delta t_k \tilde{f}_{Q,k}^{(j)}\right]^\wedge\right), \quad (56)$$

$$\begin{bmatrix} \tilde{\mathbf{q}}_k^{(i)} \\ \tilde{\mathbf{v}}_k^{(i)} \\ \tilde{\mathbf{r}}_k^{(i)} \end{bmatrix} \triangleq \begin{bmatrix} \tilde{\mathbf{q}}_k \\ \tilde{\mathbf{v}}_k \\ \tilde{\mathbf{r}}_k \end{bmatrix} + \sum_{j=1}^{i-1} a_{ij} \Delta t_k \begin{bmatrix} \tilde{f}_{\mathbf{q},k}^{(j)} \\ \tilde{f}_{\mathbf{v},k}^{(j)} \\ \tilde{f}_{\mathbf{r},k}^{(j)} \end{bmatrix} \quad (57)$$

are defined and computed for every $i = 1, \dots, N_s$. The coefficients a_{ij} , b_i , c_i , $j < i = 1, \dots, N_s$ are obtained from an appropriate Butcher table (Andrieu and Crassidis, 2013), with the coefficients for the 3rd order 3-stage Crouch-Grossman method provided in the Table 1 below. The number of rows in the table represents the number of time stages, with the $\{c_i\}$ column representing the coefficients of steps in time $t_i = t_k + c_i \Delta t_k$ where are functions are to be evaluated, the number of columns is representing the number of half-steps, with $\{c_i\}$ representing the weight of each half-step, and the coefficients $\{a_{ij}\}$ representing the weight of the intermediary stage i when computing the half-step j .

Table 1. Crouch-Grossman 3rd Order 3-Stage Butcher Table.

c_1	a_{11}	a_{12}	$\dots$	a_{1,N_s}
c_2	a_{21}	a_{22}	$\dots$	a_{2,N_s}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
c_{N_s}	$a_{N_s,1}$	$a_{N_s,2}$	$\dots$	a_{N_s,N_s}
	b_1	b_2	$\dots$	b_{N_s}
		$\uparrow$		
0				
3/4		3/4		
17/24	119/216	17/108		
	13/51	-2/3	24/17	

The RelDyn residual $\epsilon_k^{\text{RelDyn}}(\tilde{x}_k, \tilde{x}_{k+1})$ is now computed such that

$$\epsilon_k^{\text{RelDyn}}(\tilde{x}_k, \tilde{x}_{k+1}) \triangleq \begin{bmatrix} \log \left(\left(\tilde{Q}_k^\top \tilde{Q}_{k+1} \right)^\top \tilde{\Omega}_k^{\text{prop}} \right)^\vee \\ \tilde{\mathbf{q}}_{k+1} - \tilde{\mathbf{q}}_k - \Delta \tilde{\mathbf{q}}_k^{\text{prop}} \\ \tilde{\mathbf{v}}_{k+1} - \tilde{\mathbf{v}}_k - \Delta \tilde{\mathbf{v}}_k^{\text{prop}} \\ \tilde{\mathbf{r}}_{k+1} - \tilde{\mathbf{r}}_k - \Delta \tilde{\mathbf{r}}_k^{\text{prop}} \end{bmatrix}. \quad (58)$$

5.7. Process noise propagation

To compute P_k we introduce the matrix Λ_k , such that, given any pair $(\tilde{x}_k, \tilde{u}_k)$,

$$\begin{aligned} \Delta_X(x(t_{k+1}), \tilde{x}_k) &= \Phi(t_{k+1}, t_k) \Delta_X(x(t_k), \tilde{x}_k) \\ &+ \int_{t_k}^{t_{k+1}} \Phi(t_{k+1}, \tau) L(\tilde{x}(\tau), p) d\mathbf{\epsilon}(\tau), \\ &= \Phi(t_{k+1}, t_k) \Delta_X(x(t_k), \tilde{x}_k) + \Lambda_k \mathbf{\epsilon}_k, \end{aligned}$$

with initial conditions $\hat{x}(t_k) = \tilde{x}_k$, and $\hat{u}(t_k) = \tilde{u}_k$, and where $\mathbf{\epsilon}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_9)$ and $\dot{\Phi}(t, t_k) = F(\hat{x}(t), \hat{u}(t), p) \Phi(t, t_k)$, $\Phi(t_k, t_k) = \mathbf{I}_9$. We see that $P_k = \Lambda_k \Lambda_k^\top$. It follows that

$$\begin{aligned} P_k &= \int_{t_k}^{t_{k+1}} \Phi(t_{k+1}, \tau) L(\tilde{x}(\tau), p) L^\top(\tilde{x}(\tau), p) \\ &\times \Phi^\top(t_{k+1}, \tau) d\tau, \quad \hat{x}(t_k) = \tilde{x}_k, \quad \hat{u}(t_k) = \tilde{u}_k. \end{aligned}$$

We further approximate the linearized system by assuming that the matrices F , L shown in Section 4.7 are piecewise constant over the interval $[t_k, t_{k+1}]$, that is, $F_k \triangleq F(\hat{x}(t_k), \hat{u}(t_k), p)$, $L_k \triangleq L(\hat{x}(t_k), \hat{u}(t_k), p)$. Then, we construct the following matrix (Van Loan, 1978)

$$\mathcal{C} = \begin{bmatrix} F_k & L_k L_k^\top \\ 0_{12 \times 12} & F_k^\top \end{bmatrix},$$

take the matrix exponential, yielding

$$e^{\mathcal{C} \Delta t_k} = \begin{bmatrix} \Phi_k & \Gamma_k \\ 0_{12 \times 12} & \Phi_k^\top \end{bmatrix},$$

and we extract from the right-hand side the desired submatrices, that is, the discrete state transition matrix $\Phi_k = \Phi(t_{k+1}, t_k)$ of the piecewise-constant linearization, as well as the covariance of the discretized process noise $\Lambda_k \mathbf{\epsilon}_k$, given as $P_k = \Gamma_k \Phi_k^\top$.

5.8. Loop closure

As we further illustrate with an example in Section 6.1, opportunities for revisiting and remeasuring surface feature points of the small-body exist owing to the fact that the target small-body is typically rotating, while the spacecraft maintains a hovering positing. As is well-known in the

computer vision community, loop-closure constraints allow for eliminating drift in navigation and mapping solutions caused by the accumulation of 2D-3D measurement errors in successive robot poses, especially when these are a result of incremental smoothing SLAM solutions. We therefore expect the inclusion of loop-closure constraints to improve the overall solution of both the pure visual odometry (VO) solution, and to a lesser extent, the VO + RelDyn solution. Indeed we expect that the inclusion of the odometric factor would be to have a corrective effect on the accumulating VO projection errors and therefore for the drift in the solution to be less prevalent.

To perform loop closure, we leverage the bag-of-words (BoW) representation developed by Galvez-López and Tardos (2012). Specifically, we convert each image $i = 1, \dots, N$ to a bag-of-words vector $\mathbf{v}_i$ and compute the similarity metric

$$s(\mathbf{v}_i, \mathbf{v}_j) \triangleq 1 - \frac{1}{2} \left| \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|} - \frac{\mathbf{v}_j}{\|\mathbf{v}_j\|} \right|. \quad (59)$$

We compare all prior images that are at least 10 frames away from the current frame. When the similarity score for two images is greater than a threshold η , we perform an additional geometric check and then add a factor—in the form of a GTSAM BetweenFactor—between the poses corresponding to the detected loop. The BetweenFactor requires an error model, which we pick as a Gaussian distribution with associated covariance Σ_{LC} given as

$$\Sigma_{\text{LC}} = J_R \Sigma_{R_{\text{LC}}} J_R^\top + J_r (R_{\text{LC}}) \Sigma_{r_{\text{LC}}} J_r^\top (R_{\text{LC}}), \quad (60)$$

where $\Sigma_{R_{\text{LC}}} = \sigma_{R_{\text{LC}}}^2 \mathbf{I}_3$ and $\Sigma_{r_{\text{LC}}} = \sigma_{r_{\text{LC}}}^2 \mathbf{I}_3$.

5.9. Overall architecture

We now present the overall architecture of AstroSLAM algorithm in Figure 6, demonstrating the flow of information from input images and time stamps, all the way to the output estimates of state. We distinguish the three main components of the algorithm, as discussed above in this section, that is the Front-end subsystem, the Back-end subsystem and the Loop closure subsystem. In addition, we've explicitly illustrated the locations in the pipeline where the RelDyn formulation—and its corresponding odometric factor—intervene. Notably, the guess values for the current camera pose may either be obtained from the EPnP algorithm or the RelDyn propagator by integrating the state since the last time step. Nominally, the EPnP-derived pose is selected as candidate guess for the current camera pose, from which preliminary triangulation and known landmark association are performed, as explained earlier in Section 5.4. The selection is predicated on the number of successfully tracked features previously associated to known

landmarks. If the EPnP algorithm fails, such as in the case where there are too few 2D-2D matches, we select the RelDyn-derived instead for this process.

We've modeled the Front-end subsystem, the Back-end subsystem and the Loop closure subsystem as separate subsystems and capable of working in parallel, provided multiple worker threads, shared data and mutual execution exclusions. Note that, although possible, no attempt at multi-threading has been undertaken for improving the performance of the algorithm in AstroSLAM. Future work incorporating further optimizations to the code may also focus on this aspect.

6. Validation on legacy mission imagery

In this section, we discuss the design of the validation process for the proposed algorithm. We provide the details of a validation test case using imagery and data pertaining to a previously flown mission. Since access to real ground-truth data for a flown mission is impossible, in the next session we also test our algorithm using data generated in a controlled lab environment, allowing us to compare the results of the estimation problem against actual ground-truth.

6.1. PDS small body imagery dataset

We use real imagery (Nathues et al., 2011) of Asteroid (4) Vesta acquired during the Rotation Characterization 3 (RC3) observation phase of the Dawn mission (Russell and Raymond, 2012), and archived in the Small Bodies Node of the NASA Planetary Data System (PDS), to validate the algorithm. In the chosen sequence, the

1024 × 1024 pixel size images were captured, while the spacecraft performed one apparent revolution around Vesta in the asteroid body-fixed frame, with a mean orbital radius of 5,470 km. The images thus provide a spatial resolution of 0.5 km/pixel of the surface (see Figure 7 for sample images). We display the optical flow of the detected features, indicating that an apparent revolution around the small body over the course of the image sequence would allow revisiting regions of the surface previously seen. This sequence therefore enables possible loop-closures to be tested as well. We discuss the effects of loop closure in Section 6.6.

Three tests were conducted for the PDS Dawn RC3 dataset, each with the same prior uncertainty on poses $T_{G_0S_0}$ and $T_{G_1S_1}$, that is, the covariance $\Sigma_{T,0}$ of the initial pose measurements. The first test uses our pure visual odometry (VO) algorithm, where we excluded the RelDyn factors. The second uses the VO algorithm, with included RelDyn and RelKin factors. The third tests uses the approach developed in Setterfield et al. (2018a), using pre-integrated IMU measurements and the kinematic rotational factors.

For the third case, IMU measurements were simulated by sampling a zero-mean Gaussian distribution, mimicking the scenario described in Section 3. The values are chosen as $\Sigma_R^m = \sigma_R^2 I_3$ and $\Sigma_r^m = \sigma_r^2 I_3$, with $\sigma_R = 1 \times 10^{-5}$ rad and $\sigma_r = 0.05$ km. The intrinsic parameters of the Dawn Framing Camera (FC) were taken to be those computed during calibration (Russell and Raymond, 2012). In the conducted tests, the algorithm was set to extract and describe 2,500 ORB features where eight scale pyramid levels are explored for detection.

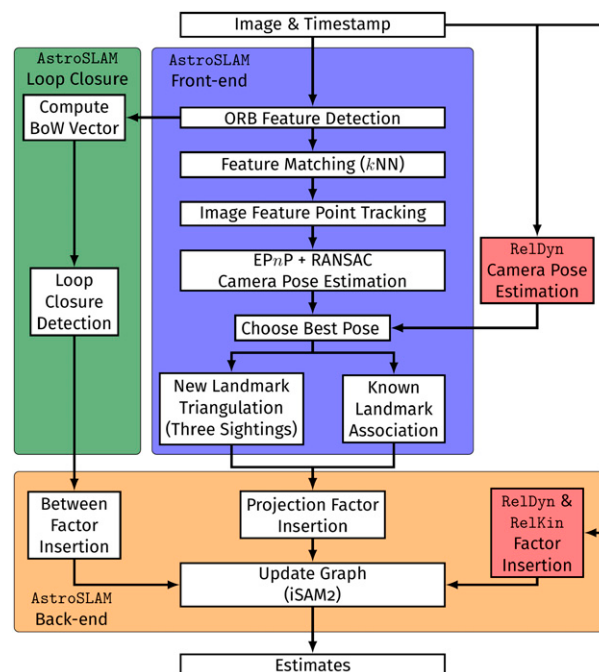


Figure 6. Architecture of the AstroSLAM algorithm, with the interactions of the components detailed in Section 5 made explicit.

6.2. Determining the ground truth data for the NASA PDS dataset

To validate the estimated spacecraft relative trajectory, we use the archival SPICE kernel datasets maintained by NASA's Navigation and Ancillary Information Facility (NAIF). For the missions archived, SPICE kernels may be queried to provide **S**pacecraft ephemeris and **P**lanetary ephemeris as a function of time, as well as **I**nstrument descriptive data, **C**amera orientation matrix data and **E**vents information, such as mission phases.

To validate the reconstructed map, we use an archival shape model of Vesta (Preusker et al., 2016) courtesy of the PDS Small Bodies Node. This shape model was derived using stereo photogrammetry (SPG) from a subset of DAWN mission Framing Camera 2 (FC2) images captured during the High-Altitude Mapping Orbit (HAMO) mission segment. The model comprises approximately 100k vertices and 197k triangular faces and is shown in Figure 7.

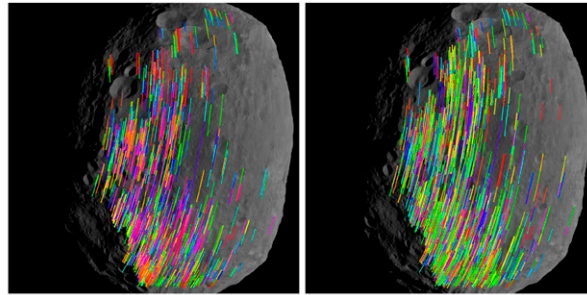
6.3. Quantitative evaluation of trajectory estimation and reconstructed map

We analyze the errors in the spacecraft's local-horizontal-local-vertical (LHLV) frame $\mathcal{L}$ to better reveal the

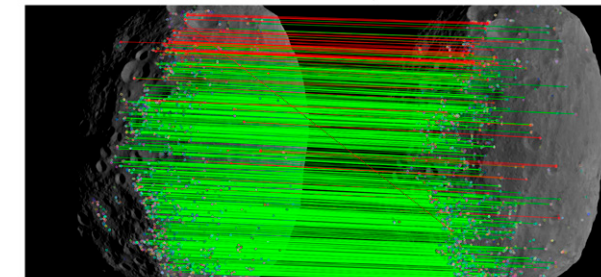
performance of the algorithm. For this purpose, we define the frame $\mathcal{A} = (\mathbf{A}, \{\vec{\mathbf{a}}_i\}_{i=0}^3)$ assumed to be the ground truth frame centered at the true center-of-mass of the small-body, and oriented such that $\{\vec{\mathbf{a}}_i\}_{i=0}^3$ correspond to the small-body's principal axes. This ground truth frame is provided by a combination of the NAIF SPICE kernels and the PDS shape model of the small-body. By virtue of the assumed true (e.g., NAIF SPICE kernel) position vector $\mathbf{r}_{\text{SA}}$ and true velocity $\mathbf{v}_{\text{SA}}$ vector, further define the LVLH frame $\mathcal{L} \triangleq (\mathbf{S}; \{\ell_i\}_{i=1}^3)$ such that $\vec{\ell}_3 \triangleq \mathbf{r}_{\text{SA}} / \|\mathbf{r}_{\text{SA}}\|$ corresponds to the radial direction, $\vec{\ell}_2 \triangleq (\mathbf{r}_{\text{SA}} \times \mathbf{v}_{\text{SA}}) / \|\mathbf{r}_{\text{SA}} \times \mathbf{v}_{\text{SA}}\|$ corresponds to the cross-track direction, and $\vec{\ell}_1 = \vec{\ell}_2 \times \vec{\ell}_3$ corresponds to the along-track direction. Further, using the NAIF SPICE data, we know the small-body's principal axes $\{\vec{\mathbf{a}}_i\}_{i=0}^3$, leading to the definition of the rotation $(R_{\mathcal{AL}})_{ij} = \vec{\mathbf{a}}_i^T \cdot \vec{\ell}_j$. For each time index $k = 1, \dots, n$ we compute the position error

$$\delta \mathbf{r}_k \triangleq R_{\mathcal{AL}_k}^T \left(\hat{\mathbf{r}}_{\text{SA}_k}^{A_k} - \mathbf{r}_{\text{SA}_k}^{A_k} \right), \quad (61)$$

where $\hat{\mathbf{r}}_{\text{SA}_k}^{A_k} = \hat{R}_{\mathcal{AG}}(\hat{\mathbf{r}}_{\text{SG}_k}^{\mathcal{G}} - \hat{\mathbf{r}}_{\text{AG}}^{\mathcal{G}})$, the constant parameter $\hat{\mathbf{r}}_{\text{AG}}^{\mathcal{G}}$ is estimated, and where the constant rotation $R_{\mathcal{AG}}$ is assumed to be known from ground truth (NAIF SPICE) for the purpose of this error analysis. The error in relative orientation is then better described by making use of the SO(3) logarithm map followed by the *vee* operator, as



(a) Image ORB feature point tracking over 2 consecutive images. Lines of different colors indicate different feature point tracks.



(b) Image ORB feature point matching between two consecutive image frames. Green lines show inlier matches (kept) and red lines show outlier matches (rejected) by virtue of RANSAC geometric check.

Figure 7. Visualizing the image feature tracking performance of the front-end subsystem on the NASA DAWN RC3 legacy mission image sequence. (a) Image ORB feature point tracking over two consecutive images. Lines of different colors indicate different feature point tracks. (b) Image ORB feature point matching between two consecutive image frames. Green lines show inlier matches (kept) and red lines show outlier matches (rejected) by virtue of RANSAC geometric check.

detailed in Section 2.1, to produce the error between the ground-truth NAIF SPICE rotation $R_{A_k S_k}$ and the estimated rotation $\hat{R}_{A_k S_k}$, such that

$$\delta \mathbf{\hat{K}} = \log \left(\hat{R}_{A_k S_k}^\top R_{A_k S_k} \right)^\vee. \quad (62)$$

The results of this evaluation are presented in Section 6.4 for the DAWN RC3 sequence and in Section 7.4 for the in-lab experiment sequence.

Given the map Ψ_N at the final time $k = N$ and the set of ground truth 3D shape model vertices $\mathcal{V}$, we evaluate the quality of the estimated landmarks by computing the distances $\{d(L, \mathcal{V})\}_{L \in \Psi_N}$, where

$$d(L, \mathcal{V}) \triangleq \min_{V \in \mathcal{V}} \|\mathbf{r}_{LA}^A - \mathbf{r}_{VA}^A\|_2, \quad (63)$$

which, in our case, minimizes the 2-norm. The results of this evaluation are presented in Figure 13, where the estimated

landmarks are colored as a function of their distance to the closest point in the ground-truth set of vertices. Note that this distance metric is one-sided. Thus, choosing to instead search over the set Ψ_N would yield different distance values $\{d(V, \Psi_N)\}_{V \in \mathcal{V}}$. Nevertheless, given the much higher vertex density of the ground truth shape model as compared to the estimated landmarks, we deem the described point-to-set distance $d(L, \mathcal{V})$, $L \in \Psi_N$ to be an appropriate measure of the deviation of our solution landmarks from the 3D shape.

6.4. Trajectory estimation results and discussion

In this section, we illustrate and discuss the results of the real-mission dataset. We visualize in Figure 8 the estimated 3D trajectory $\{\hat{\mathbf{r}}_{S_k A_k}^A\}_{k=0}^N$. We compare the visual odometry (VO), the VO + RelDyn and the (Setterfield et al., 2018a) solutions at the final incremental smoothing iteration with the trajectory extracted from the DAWN mission SPICE kernel, assumed to be the ground truth relative position

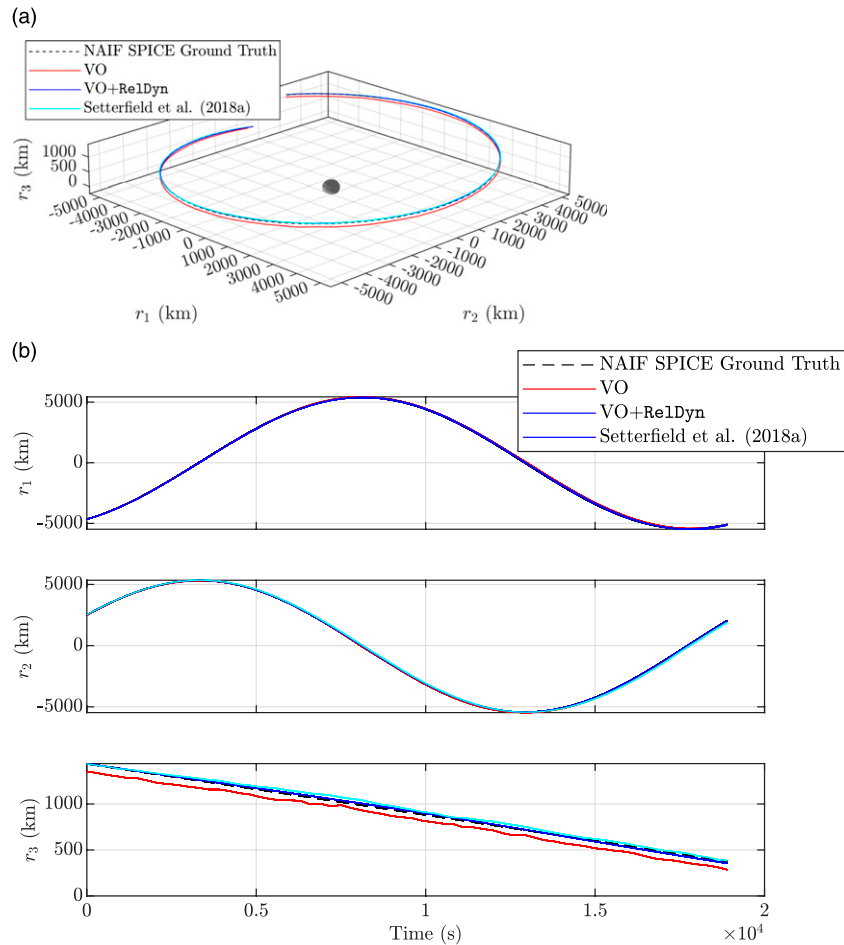


Figure 8. The relative position trajectory estimates obtained following the final incremental smoothing solution, as expressed in the small-body's ground truth principal axes frame. In this figure, we compare the obtained relative position vectors $\mathbf{r}_{SA}^A = [r_1, r_2, r_3]^\top$ from visual odometry alone (VO) in red, from VO + RelDyn in blue and from the approach of Setterfield et al. (2018a) in cyan against the ground truth trajectory taken as the NAIF SPICE relative positions. Note that these solutions are without loop closure. We see from the time history plots that the estimated components of the trajectories aligned with the body-fixed orbital plane (r_1, r_2) are virtually indistinguishable when the relative position are expressed in the small body principal axes frame. This is why we opt to present the errors in the LHLV frame in Figure 9. (a) 3D visualization of the relative position vector expressed in the Vesta's principal axes frame. (b) Time history of the relative position vector.

vectors $\{\mathbf{r}_{S_k A_k}^{A_k}\}_{k=0}^N$. At the scale necessary to show the entirety of the circumnavigating trajectory in three dimensions, the difference in the estimated solutions appears too small to adequately establish which algorithm is best performing by mere visual inspection. Instead, we opt to analyze the relative position navigation error expressed in the LHLV frame coordinates, as presented in Figure 9. Note that the cases shown exclude any loop closure constraints, as we desire to assess the corrective nature of the odometric factors alone. We separately provide further analysis using loop-closures in Section 6.6. In addition, note that the estimate of the small body center of mass location $\hat{\mathbf{r}}_{AG}^G$ used in the trajectory analysis is the one we obtain at the last incremental smoothing step, as is further discussed in Section 6.5.

When comparing VO, VO + RelDyn, and the (Setterfield et al., 2018a) cases in Figure 9, we observe a clear improvement in terms of radial navigation error $\delta\hat{r}_{RAD}$ and cross-track error $\delta\hat{r}_{XT}$ for the VO + RelDyn approach. Specifically, considering that the orbit radius is roughly 5,470 km throughout the sequence, we calculate from the errors in the radial direction ($\delta\hat{r}_{RAD}$) shown in Figure 9, an average error of 1.29%, with a worst case of 1.5% for the pure VO case, an average of 0.17%, with a worst case of 0.33% for the VO + RelDyn case, and an average of 0.48%, with a worst case of 1.01% for the (Setterfield et al., 2018a) case.

The improvement in the cross-track error is even starker, with the VO case showing an average of 75 km error, with a worst case of 218 km error, while the VO + RelDyn shows an average of 12.3 km error, with a worst case of 23 km

error, and an average of 68 km for the (Setterfield et al., 2018a) case, with a worst case of 140 km deviation.

The improvements can be summarily explained by considering the corrective nature of the encoded orbital mechanics in the odometric factor on components of the relative position estimate. Specifically, (a) in the radial direction—which typically also corresponds to the camera boresite direction—the error is improved owing to the better description of the velocity vector, since we account for gravitational acceleration, (b) in the cross-track direction, the error is improved since the orbital plane of the motion is restricted by the orbital angular momentum being constrained. Furthermore, the large cross-track error ($\delta\hat{r}_{XT} \sim 50 - 100\text{km}$) of the pure VO solution can be explained by the fact that the sequence images provide little change in perspective in the actual cross-track direction, leading to poor estimate correction in that direction. In fact, as can also be seen in Figure 9, we observe that this component is the major contributor to the overall navigation error norm in the VO case. We note that the Setterfield solution provides no improvement in the cross-track direction, yet the VO + RelDyn solution provides the best improvement, reducing cross-track error by a full order of magnitude. Therefore, by constraining the solution using RelDyn factors, which encourages a solution where the modulus of the radius changes little due to the large distance to the asteroid during the approach phase, and where the motion of the spacecraft is quasi-planar due to a quasi-Keplerian configuration of the orbit, we improve the error considerably in both of these directions.

In the VO + RelDyn case, we additionally obtain estimates of the relative linear velocity $\hat{\mathbf{v}}_{SA}^I$, which we show in

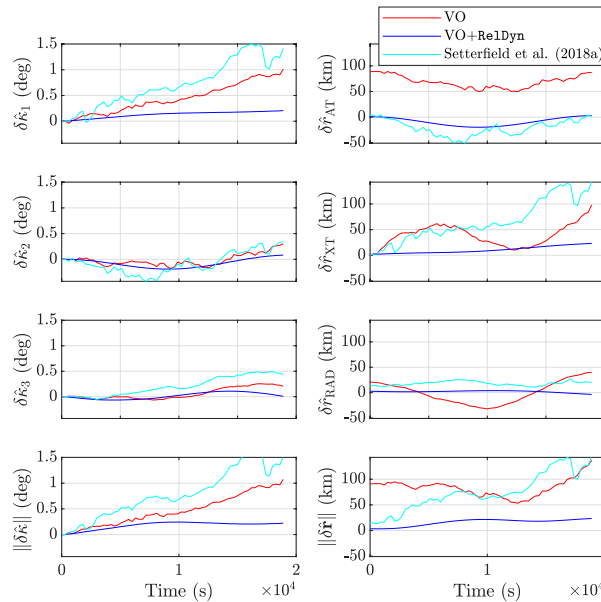


Figure 9. Time history of navigation errors following the final incremental smoothing step—without loop closure—expressed in the LHLV coordinate frame. Here we compare the navigation errors of visual odometry alone (VO) in red, visual odometry with RelDyn in blue, and the (Setterfield et al., 2018a) approach in cyan against the NAIF SPICE relative position and attitude kernels, chosen as the ground truth. We compare the attitude error local coordinate vector $\delta\kappa = [\delta\kappa_1, \delta\kappa_2, \delta\kappa_3]^T$ and the position error vector $\delta\mathbf{r} = [\delta\hat{r}_{AT}, \delta\hat{r}_{XT}, \delta\hat{r}_{RAD}]^T$, as well as their respective 2-norm, across the three procedures considered.

overlay to ground truth NAIF SPICE velocity estimates in Figure 10. We note that in the Setterfield et al. (2018a) case, the inertial linear velocity $\mathbf{v}_{\text{SW}}^T$ of the observing spacecraft is estimated, but not the inertial velocity of the target and, therefore, there is no direct way of computing the relative velocity for comparison to the velocities in Figure 10. Instead, we obtain a comparable relative velocity by performing a numerical derivative by finite differencing of the estimated relative position vectors $\hat{\mathbf{r}}_{\text{SA}k}^T = R_{\text{IS}_k} \hat{R}_{\text{S}_k \text{G}_k} (\hat{\mathbf{r}}_{\text{S}_k \text{G}_k}^{\text{G}_k} - \hat{\mathbf{r}}_{\text{AG}}^{\text{G}_k})$. As is obvious in Figure 10, (a) there is non-negligible velocity variation over the length of the sequence owing to the gravitational interaction and weak external forces (SRP); (b) the VO + RelDyn algorithm adequately captures this interaction while the Setterfield et al. (2018a) method struggles to do so, at times performing no better than the pure VO case. This is an expected result of smoothing predicated on the pre-integrated IMU factors in Setterfield et al. (2018a), since these are insensitive to the gravitational attraction.

6.5. Dynamical parameter estimation results and discussion

In this section, we present and discuss the on-the-fly estimation results of key dynamical parameters obtained from the NASA DAWN RC3 real sequence. Recall that the full factor graph presented in Section 5 allows us to estimate several parameters inherent to the dynamical system on-the-fly. Specifically, we want the two first components c_1, c_2 of the location of the center-of-mass (these are components perpendicular to the spin axis), the target small-body's standard gravity parameter μ_a and the spin rate w_0 of the single-axis rotator configuration of the target object. The time history of the values estimated for these parameters are reported in Figure 11.

When comparing the VO + RelDyn procedure to the approach of Setterfield et al. (2018a) in Figure 11, we notice a quicker and more accurate convergence of the center-of-mass parameters c_1, c_2 using our method. Moreover, we

obtain an estimate of the gravity parameter with the RelDyn factor, in contrast to the Setterfield et al. (2018a) method, which does not inherently provide such an estimate on-the-fly. The initial conditions considered for the parameters were $\mu_a^{(0)} = 0.1 \text{ km}^3 \text{ s}^{-2}$ and $[c_1^{(0)}, c_2^{(0)}] = [0 \text{ km}, 0 \text{ km}]$. We note that these values are not close guesses of the true values $\mu_a = 17.29 \text{ km}^3 \text{ s}^{-2}$ and $[c_1, c_2] = [200 \text{ km}, 125 \text{ km}]$. Yet, our procedure converges to within 1% error of the actual values c_1, c_2 within 20 steps and to within 5% error of the actual value μ_a within 50 steps.

The faster settling time and better precision of these parameter estimates is explained by the fact that our dynamical model is in agreement with the true physics of the problem owing to the modeling of the gravitational interactions, while the procedure proposed in Setterfield et al. (2018a) is insensitive to the gravity vector, as discussed in Section 3.

In fact, it is crucial to note at this point that to merely obtain a workable run of our implementation of the Setterfield et al. (2018a) approach, we had to relax the process noise covariances of the pre-integrated IMU factors and the fixed covariance of the kinematic rotation factors proposed in Setterfield et al. (2018a). Restricting the covariances in order to obtain a higher “smoothing” effect resulted in convergence issues, with each estimation step taking up to a minute to solve, as well as several unsuccessful runs, during which the iSAM2 solver would be unable to find a plausible gradient descent direction for the non-linear optimization step. Consequently, we have omitted comparing the timing results of our algorithm—which we show in Section 6.8—against our implementation of the Setterfield et al. (2018a) approach.

As a side note, we should remark that we do not estimate $\hat{\mathbf{r}}_{\text{AG}}^{\text{G}_k}$ based on a physical model in the pure VO case. However, solely for the purpose of providing a comparison point for the VO case, a cursory, but fair value for the center-of-mass position may be instead predicated on the mapping solution, such as the uniformly weighted average

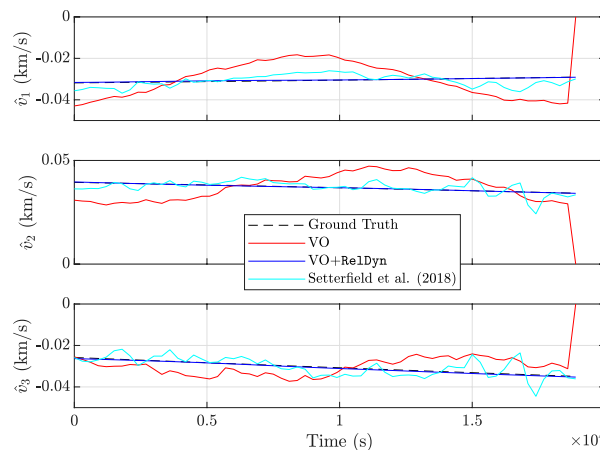


Figure 10. Relative linear velocity estimates versus NAIF SPICE kernel ground truth for DAWN RC3 sequence. Here, we compare the relative linear velocity estimate $\hat{\mathbf{v}}_{\text{SA}}^T = [v_1, v_2, v_3]^T$ of the VO, VO + RelDyn and Setterfield et al. (2018a) solutions, noting that the values for VO and Setterfield et al. (2018a) are obtained by finite-differencing.

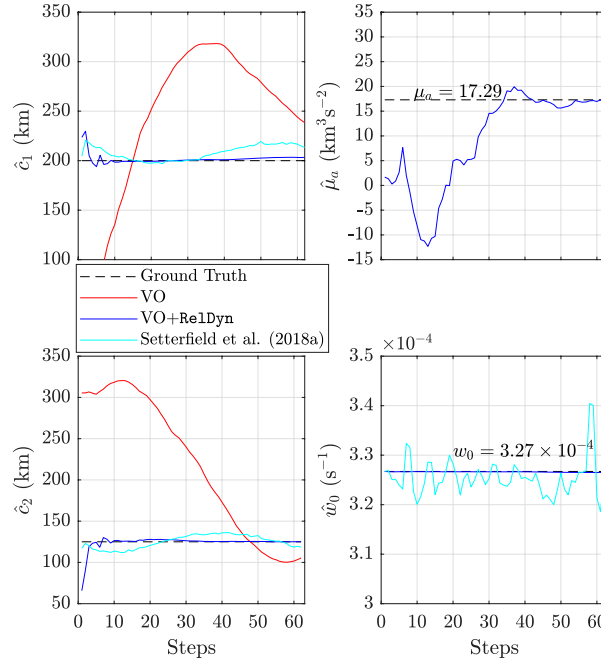


Figure 11. Estimation of dynamical parameters at every algorithm step. The estimated parameters are the two first components $\hat{c}_1, \hat{c}_2$ of the center-of-mass position vector $\hat{\mathbf{r}}_{AG}^G = [\hat{c}_1, \hat{c}_2, \hat{c}_3]^T$, the standard gravity parameter $\hat{\mu}_a$ of Asteroid (4) Vesta, and the small-body angular velocity magnitude $\hat{w}_0$, such that $\hat{w}_0 \hat{\mathbf{g}}_3 = \hat{\omega}_{GT}$. For the VO case, we show in red the position of the center-of-mass location obtained from averaging all of the successfully initialized landmarks up to the current algorithm time step.

of the estimated landmark positions, the combined solution to image centroiding or the center of a bounding box of the map landmarks. We choose here to use the averaging of the latest smoothing solution of the mapped landmark positions at every time step, as shown in red in Figure 11.

6.6. Loop closure results and discussion

In this section, we discuss and illustrate the effect of incorporating loop closure constraints for the NASA DAWN RC3 dataset. Specifically, we compare the improvement in navigation error for the visual odometry (VO) case and the VO + RelDyn case, when a loop closure factor—in the form of a GTSAM BetweenFactor—using loop detection activated at the last frame of the sequence, since features similar to the ones detected in the first image of the sequence are detected in the last frame. As observed in Figure 12, the drift due to the accumulation of errors inherent to incremental SLAM algorithms manifests itself in a cross-track error ($\delta\hat{r}_{XT}$) and, particularly, in the norm of the relative position error ($\|\delta\hat{\mathbf{r}}\|$). As expected, under the effect of loop closure constraints, this drift is eliminated in the VO case, reducing cross-track error by an order of magnitude, with little change in the along-track and radial direction errors ($\delta\hat{r}_{AT}, \delta\hat{r}_{RAD}$). We further note the reduction by an order of magnitude in the drift of the attitude error around the spacecraft body axis $\hat{\mathbf{s}}_1$, captured in the error term $\delta\kappa_1$.

Similarly to the VO case, the VO + RelDyn case solution obtained after the last incremental smoothing step shows

little to no change as a consequence of the incorporation of loop closure constraints, except for the error in cross track direction position. In fact, as can be seen in Figure 12, the norm of the relative position error vector is barely affected by loop closure, with negligible correction in the radial direction as well. This is an expected behavior since the strong odometric constraints encoded in the RelDyn and the RelKin factors already compensate for the drift in scale and the subsequent compounding mapping errors which typically beset the pure VO SLAM solution.

6.7. Mapping results and discussion

For the purpose of evaluating the landmark mapping quality, we compare the estimated landmark positions to the ground truth shape model obtained by stereophotoclinometry (SPC) of Gaskell et al. (2008). We perform this comparison for the VO, VO + RelDyn and Setterfield et al. (2018a) solutions. We note that the shape model is centered at the center-of-mass point A and the coordinates of its vertices are expressed in the asteroid principal axes frame $\mathcal{A}$. On the other hand, the positions of the mapped landmarks from our solution are with respect to the target geometric frame origin G and are expressed in the $\mathcal{G}$ -frame coordinates. To eliminate the effect of the center-of-mass location estimation on the mapped coordinates of the landmarks and to provide a common baseline for the three algorithms compared, we subtract the ground truth value of center-of-mass lever arm vector $\mathbf{r}_{AG}^G$ and rotate the points into the ground truth frame $\mathcal{A}$ coordinates, such that, for all landmarks $\{\mathbf{L}\}_{L \in \Psi_N}$, we have $\hat{\mathbf{r}}_{LA}^A = R_{AG}(\hat{\mathbf{r}}_{LG}^G - \mathbf{r}_{AG}^G)$.

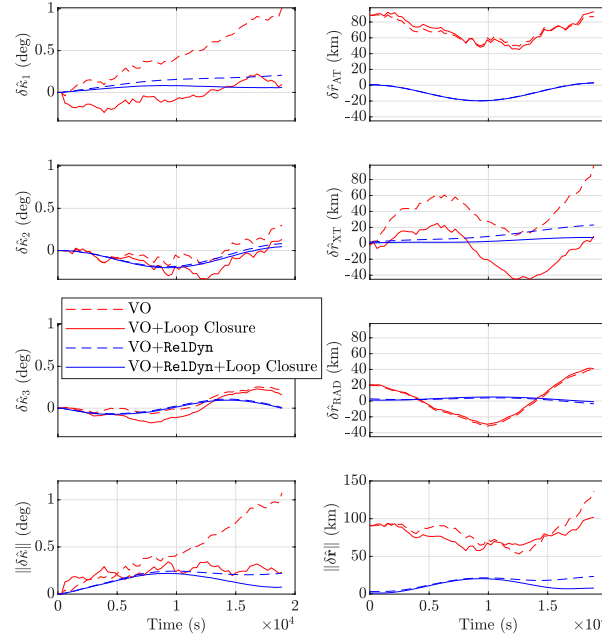


Figure 12. Effect of loop closure on navigation errors from smoothing solution, expressed in the LHLV coordinate frame. Here, we show the effect of incorporating a loop closure factor between the last frame and the first frame of the sequences by virtue of place recognition (loop detection) using the BoW vectors described in Section 5.8. We compare the attitude error local coordinate vector $\delta\hat{\kappa} = [\delta\hat{\kappa}_1, \delta\hat{\kappa}_2, \delta\hat{\kappa}_3]^\top$ and the position error vector $\delta\hat{r} = [\delta\hat{r}_{AT}, \delta\hat{r}_{XT}, \delta\hat{r}_{RAD}]^\top$, as well as their respective 2-norm, with the inclusion (solid line) and without the inclusion (dashed line) of loop closure constraints.

We observe considerable improvement in terms of the mapping results between the pure VO case baseline and the two odometric factor aided methods VO + RelDyn and Setterfield et al. (2018a). The landmarks shown in Figure 13 and color-coded in terms of their error with respect to the surface of the ground truth shape model demonstrate that, due to the tightly coupled relationship between camera poses and estimated landmark depth in the bundle adjustment setup, the inclusion of an odometric factor which improves the pose estimates will in turn cause an improved triangulation of new landmarks and improved known landmark measurement constraints. Consequently, the estimated location of landmarks is also improved after optimization. Specifically, we observe that the distribution of landmark position errors for the VO + RelDyn case is more tightly spread around its mean value of 2.34 km and a standard deviation of 1.36 km when compared to the Setterfield et al. (2018a) method, where the error is more spread out around its higher mean value of 4.36 km with a standard deviation of 2.61 km or when compared to the pure VO case where the error is spread out around its mean of 3.26 km with a standard deviation of 1.87 km. This is due to the RelDyn odometric factor providing a more realistic motion constraint as compared to the factor proposed in Setterfield et al. (2018a).

As seen in the color-coded histogram of errors in Figure 13, both the VO and Setterfield et al. (2018a) methods present a secondary bulge in the errors not concentrated around 2 km error value. From the color coding, we observe that these bulges correspond to the same area mapped in the +y axis direction for both VO and for Setterfield et al. (2018a). This seems to indicate that drift

and compounding mapping errors have accumulated in this area, but are not corrected by the odometric constraints of the method in Setterfield et al. (2018a) due to the relaxed process covariances that had to be considered for a successful run, as mentioned in Section 6.5. In turn, these higher errors in mapping adversely affect the trajectory errors as shown in Figure 9. In contrast, the VO + RelDyn method errors only present a singular bulge centered around 2 km, and we see from the color coding that the landmark position errors are not higher in any preferential body-centered direction. We note, overall, that the VO + RelDyn also presented 23,739 successfully triangulated and tracked landmarks, the most among the three when compared to 10,460 landmarks for the (Setterfield et al., 2018a) method, and 23,191 landmarks for the pure VO case. Again, the relaxations required to have a successful trial in the (Setterfield et al., 2018a) method severely hinder the successful triangulation of landmarks.

6.8. Timing results and discussion

In this section, we provide a brief description of the timing performances of the algorithm using the DAWN RC3 dataset. The algorithm is run on a desktop computer, with an AMD CPU core x58003D, with 64 GB of RAM. We note that the algorithm's front-end does not make use of any GPU-offloaded computations for an increase in speed, nor does it make use of multi-threading for the main pipeline. The breakdown of the timing for the 63-frame NASA DAWN RC3 sequence, obtained by profiling the C++ code using simple time counters, is as provided in Table 2. We

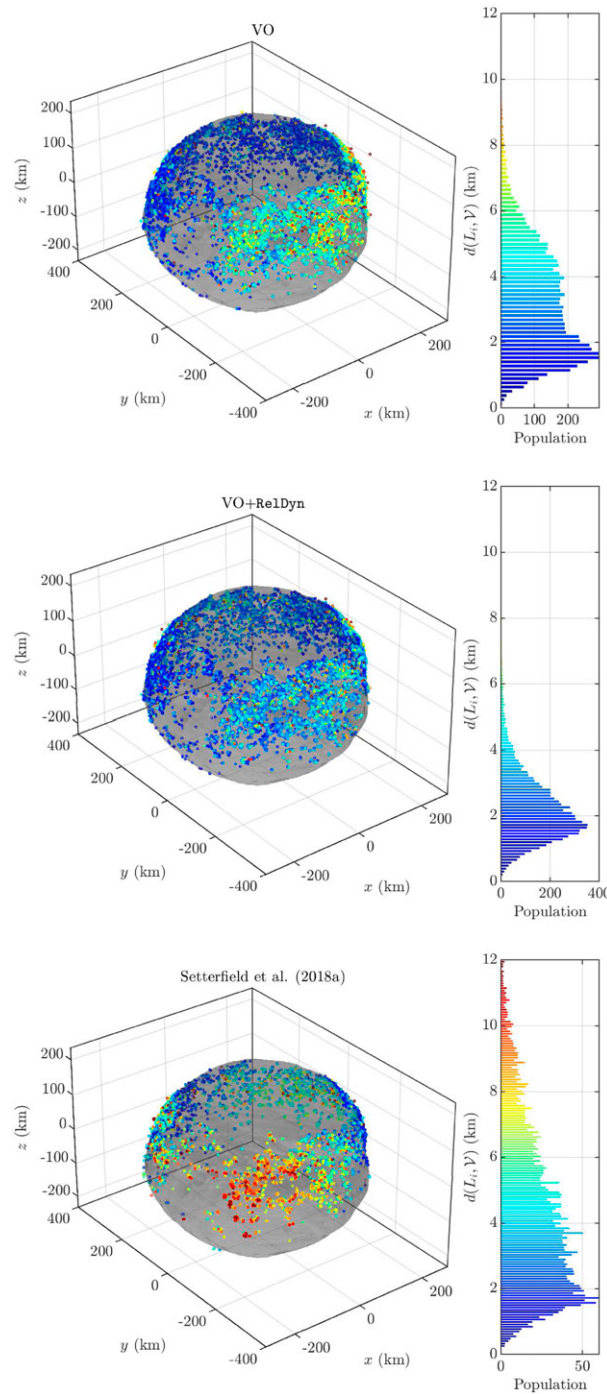


Figure 13. Estimated landmark positions and errors with respect to SPC-derived PDS shape model. The estimated map of landmarks corresponding to the VO, VO + RelDyn and [Setterfield et al. \(2018a\)](#) methods are plotted against the stereophotoclinometry (SPC)-derived shape model. The color coding corresponds to the magnitude of the position error of the landmark with respect to its closest point on the surface of the shape model, as given by equation (63), with hotter colors representing larger deviations from the shape model surface. The histogram to the right illustrates the distribution of the errors.

note that a large portion (76%) of the total algorithm time is spent in the triangulation of new landmarks. This is due to the strict thresholding of the RANSAC-based outlier rejection we use in the PnP DAWN RC3 case to guess the current frame pose, so that this better pose may be used to triangulate reliable landmarks.

The timing results presented in [Table 2](#) strongly support the feasibility of an incremental and online implementation. Specifically, in this sequence, there is a constant 300 s image time stamp separation, which means that, with the hardware considered—and without any specific attempt at optimizing the code runtime—the algorithm may be executed

Table 2. Timing Results of the VO+RelDyn+Loop Closure Procedure in AstroSLAM for Obtaining the Incremental Smoothing Solution to the Totality of the 63-Frame NASA DAWN RC3 Image-Dataset Sequence.

Task	Total task time (s)	Percentage of total time (%)
Detect and describe features, detect loop-closures	2.404	11.3
Match features	1.222	5.7
Triangulate	16.347	76.7
Build Graph	0.052	0.2
Solve (iSAM2)	1.289	6.0
Total	21.314	100

up to 14 times on the on-board computer before the next image is input. We did not perform a benchmarking of our algorithm against the cited [Setterfield et al. \(2018a\)](#) approach, since we did not have access to the original implementation of that algorithm.

Nevertheless, we note that our implementation of the approach in ([Setterfield et al., 2018a](#)) using the same VO front-end and GTSAM and iSAM2 solvers, is very slow, with upwards of 30 s required to process a new solution since the moment a new frame is inserted. We believe that the sluggish behavior of the [Setterfield et al. \(2018a\)](#) approach is owed to the inability of the IMU preintegration factors to capture the change in velocity under the influence of gravitational attraction. This means that the optimizer is unable to perform efficient optimization since the error does not diminish when taking a step at the linearization point using the prescribed IMU factor Jacobian.

7. Experimental validation using in-lab generated imagery

To perform additional validation of the proposed AstroSLAM algorithm, we use image and ground-truth data generated at the Autonomous Spacecraft Robotic Operations in Space (ASTROS) experimental facility ([Cho et al., 2009](#)), located at the Dynamics and Control Systems Laboratory of the Georgia Institute of Technology.

7.1. Experimental setup

The ASTROS facility houses an eponymous 5 degree-of-freedom spacecraft simulator test-bed, a 7 degree-of-freedom robotic manipulator system (RMS) consisting of a SchenckTM linear stage and a Universal RobotsTM UR10e robotic arm, a 12-camera VICONTM motion capture system, as well as a dedicated control room. Mechanically, the ASTROS platform is composed of two structures, called the upper and lower stages, allowing for motion in a 5 degree-of-freedom motion ([Cho et al., 2009](#)). For the purposes of this experiment, however, the hemispherical joint between the two stages is maintained fixed at a preset attitude, and hence the test-bed is in 2 + 1 configuration, that is, providing two degrees of planar translation plus one degree of rotation around the vertical axis. A linear air-bearing system between the lower stage and the

floor levitates its lower stage off the near-perfectly flat floor for this purpose, as shown in [Figure 14](#).

The platform is fitted with 12 cold-air gas thrusters which, when firing, generate forces and torques allowing it to actively maneuver in the test arena. The ASTROS test-bed also possesses an inertial measurement unit and a rate gyro, which when paired with an extended Kalman filter, allow it to estimate the position, attitude, linear velocity and angular velocity of the upper stage. The actuation of the thrusters is performed by dedicated power electronics in response to control computed on an embedded SpeedGoatTM computer. The computer compiles and executes a program derived from a prototyped SimulinkTM model incorporating sensor measurement acquisition, control computation, actuator allocation and input-output communications with devices on the platform in real-time.

As target small-body, we use a mock asteroid affixed to the RMS end-effector. Its motion is scheduled in open-loop control mode. Using the linear stage and robotic arm joint encoder values only, we achieve sub-millimeter end-effector positioning error in the test arena with respect to an arbitrarily pre-defined inertial frame.

Throughout the experiment, the motion capture system's UDP data stream was used to save the position and attitude of the ASTROS upper stage at a frequency of 100 Hz. The frame number index of the data stream was subsequently used to synchronize and time stamp all signal histories across the multiple devices, thus providing a single clock baseline for all the acquired data.

7.2. Establishing a ground truth by simulating and tracking an idealized orbital trajectory using the ASTROS test-bed

To emulate a true unforced orbital motion as ground truth, an idealized trajectories was generated. The re-scaled version of this trajectory, performed via non-dimensionalization and re-dimensionalization to fit the physical limits of the arena and the safe allowable velocities, was tracked in the ASTROS facility test arena. The goal is to achieve a reasonably long segment duration and speed-to-downrange-distance ratio which emulate the real in-space mission scenario. We use the vectors and dimensions provided in [Table 3](#) to perform the non-dimensionalization of the ideal orbital trajectory, and the re-dimensionalization for the lab. limits. The data

chosen for the orbital case corresponds to the initial data point of the DAWN RC3 sequence. By tracking the desired trajectory with small enough errors while accumulating high-rate and precise measurements, we assume that the actual trajectory accomplished closely mimics the ideal trajectory. In theory, if this tracking is successful, the difference between the actual and desired trajectories yields a white-noise process, which we may characterize a-posteriori. Consequently, the ideal trajectory of the maneuver may now be viewed as a reasonable ground truth, with an associated “process noise.”

We used the dataset of the ASTROS on-board state estimates, along with the sequence of images acquired over this lab experiment segment to evaluate and validate the performance of AstroSLAM. The details are given next.

Let the desired camera frame, denoted by $\mathcal{E}$, be associated to the real camera frame $\mathcal{C}$, and the desired ASTROS upper stage frame, denoted by $\mathcal{D}$, be associated to the real ASTROS upper stage frame $\mathcal{S}^\circ$. The camera is fixed on the ASTROS platform’s upper stage, yielding a constant rotation matrix $R_{\mathcal{S}^\circ \mathcal{C}}$ ($R_{\mathcal{D} \mathcal{E}}$) and a constant translation vector $\mathbf{r}_{\mathcal{C} \mathcal{S}^\circ}^{\mathcal{S}^\circ}$ ($\mathbf{r}_{\mathcal{E} \mathcal{D}}^{\mathcal{D}}$), which are both estimated along with the camera intrinsic parameters by non-linear calibration. To respect the proportions of the emulated scenario, we assume that the modeled ideal spacecraft frame $\mathcal{S}$, as defined in Section 2.2, is coincident and aligned with the experimental camera frame $\mathcal{C}$. We want the desired camera frame $\mathcal{E} \triangleq \{\mathcal{E}; \{\vec{\mathbf{e}}_i\}_{i=1}^3\}$ to trace a trajectory described by the tuple $(R_{\mathcal{I} \mathcal{E}}(t), \mathbf{r}_{\mathcal{E} \mathcal{A}}^{\mathcal{I}}(t), \omega_{\mathcal{E} \mathcal{I}}^{\mathcal{I}}(t), \mathbf{v}_{\mathcal{E} \mathcal{A}}^{\mathcal{I}}(t)), t \in [t_0, t_f]$, such that

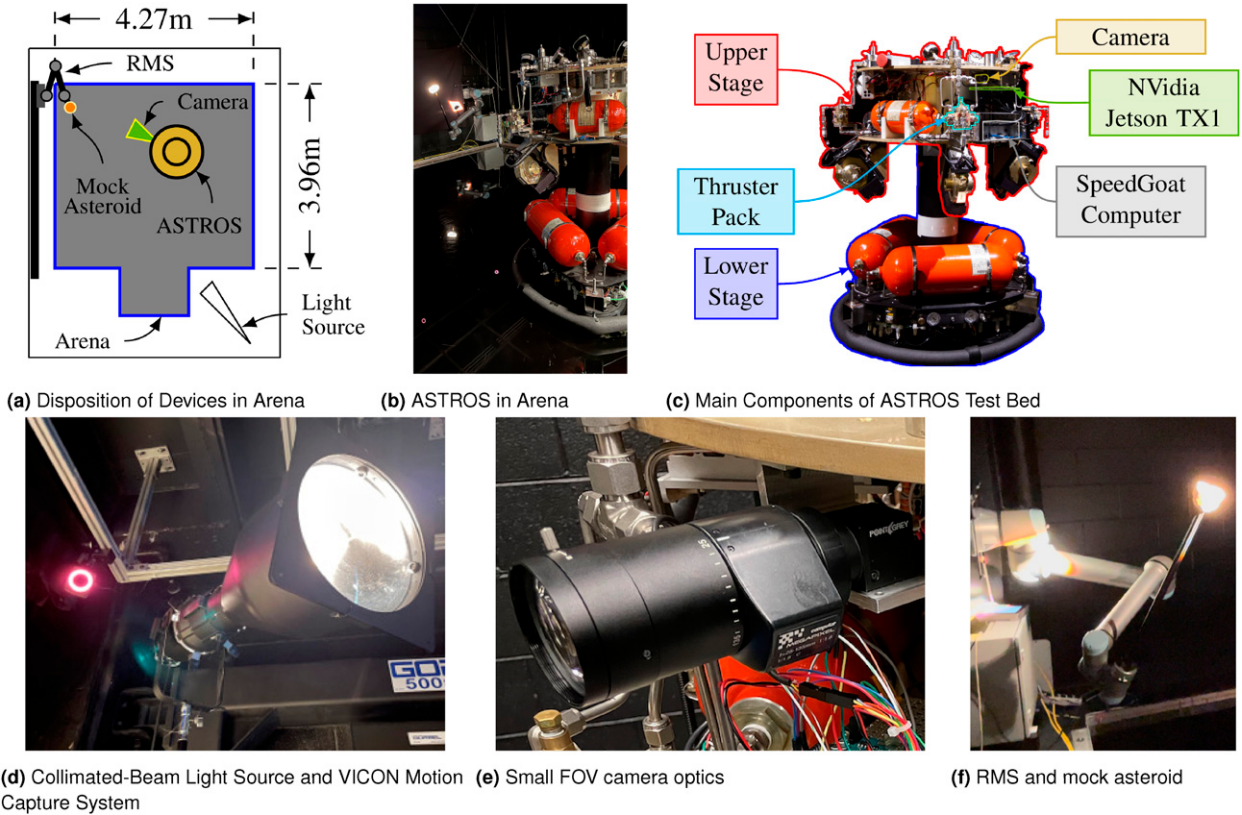


Figure 14. AstroSLAM experiment setup. (a) Disposition of Devices in Arena. (b) ASTROS in Arena. (c) Main Components of ASTROS Test-Bed. (d) Collimated-Beam Light Source and VICON Motion Capture System. (e) Small FOV camera optics. (f) RMS and mock asteroid.

Table 3. Parameters Used for Non-dimensionalization of the Idealized Orbital Trajectory.

Mass of small body m_a	$2.5907 \times 10^{20} \text{ kg}$
Gravity parameter μ_a	$17.29 \text{ km}^3 \text{ s}^{-2}$
Dist. to the Sun $\ \mathbf{r}_{\text{AO}}\ $	$3.3368 \times 10^8 \text{ km}$
Unit of distance in orbit LU_{orb}	$\ \mathbf{r}_{\text{SA}}(0)\ = 5479.4 \text{ km}$
Unit of velocity in orbit VU_{orb}	$\ \mathbf{v}_{\text{SA}}(0)\ = 5.687 \times 10^{-2} \text{ kms}^{-1}$
Unit of time in orbit $\text{TU}_{\text{orb}} = \text{VU}_{\text{orb}}/\text{LU}_{\text{orb}}$	$9.635 \times 10^4 \text{ s (26.76 h)}$

$$\begin{aligned}\dot{\mathbf{r}}_{\text{EA}}^{\mathcal{I}}(t) &= \mathbf{v}_{\text{EA}}^{\mathcal{I}}(t), \mathbf{r}_{\text{EA}}^{\mathcal{I}}(0) = \mathbf{r}_{\text{EA},0}^{\mathcal{I}}, \\ \dot{\mathbf{v}}_{\text{EA}}^{\mathcal{I}}(t) &= -\left(\frac{\mu_a}{\|\mathbf{r}_{\text{EA}}(t)\|^3} + \frac{\mu_{\odot}}{\|\mathbf{r}_{\text{AO}}(t)\|^3}\right)\mathbf{r}_{\text{EA}}^{\mathcal{I}}(t) \\ &\quad + \frac{1}{m_s}\mathbf{F}_{\text{SRP}}^{\mathcal{I}}(\mathbf{r}_{\text{EO}}^{\mathcal{I}}(t)) + \frac{1}{m_s}\mathbf{F}_s^{\mathcal{I}}(t), \mathbf{v}_{\text{EA}}^{\mathcal{I}}(0) = \mathbf{v}_{\text{EA},0}^{\mathcal{I}},\end{aligned}$$

is satisfied, while maintaining line-of-sight with the mock asteroid throughout the segment. Assuming $\mathbf{F}_s^{\mathcal{I}}(t) \equiv \mathbf{0}$, we can find a solution $\mathbf{r}_{\text{EA}}^{\mathcal{I}}(t)$ and $\mathbf{v}_{\text{EA}}^{\mathcal{I}}(t)$ readily, without requiring $R_{\mathcal{IE}}(t)$. Subsequently, we pick a center pointing strategy, such that, at any time $t \in [t_0, t_f]$,

$$\begin{aligned}\bar{\mathbf{e}}_3^{\mathcal{I}}(t) &= \mathbf{r}_{\text{AE}}^{\mathcal{I}}(t) / \|\mathbf{r}_{\text{AE}}^{\mathcal{I}}(t)\|, \\ \bar{\mathbf{e}}_2^{\mathcal{I}}(t) &= \left[\bar{\mathbf{e}}_3^{\mathcal{I}}(t)\right]^{\wedge} \mathbf{v}_{\text{AE}}^{\mathcal{I}}(t) / \left\| \left[\bar{\mathbf{e}}_3^{\mathcal{I}}(t)\right]^{\wedge} \mathbf{v}_{\text{AE}}^{\mathcal{I}}(t) \right\|, \\ \bar{\mathbf{e}}_1^{\mathcal{I}}(t) &= \left[\bar{\mathbf{e}}_2^{\mathcal{I}}(t)\right]^{\wedge} \bar{\mathbf{e}}_3^{\mathcal{I}}(t),\end{aligned}$$

and we construct $R_{\mathcal{IE}}(t) = [\bar{\mathbf{e}}_1^{\mathcal{I}}(t) \ \bar{\mathbf{e}}_2^{\mathcal{I}}(t) \ \bar{\mathbf{e}}_3^{\mathcal{I}}(t)]$, and $\boldsymbol{\omega}_{\mathcal{EI}}^{\mathcal{I}}(t) = (\dot{R}_{\mathcal{IE}}(t)R_{\mathcal{IE}}^{\mathcal{I}}(t))^{\vee}$. Finally, we obtain $R_{\mathcal{ID}}(t) = R_{\mathcal{IE}}(t)R_{\mathcal{ED}}(t)$, $\mathbf{r}_{\text{DO}}^{\mathcal{I}}(t) = \mathbf{r}_{\text{EA}}^{\mathcal{I}}(t) + \mathbf{r}_{\text{AO}}^{\mathcal{I}}(t) - R_{\mathcal{ID}}(t)\mathbf{r}_{\text{ED}}^{\mathcal{D}}(t)$, $\boldsymbol{\omega}_{\mathcal{DI}}^{\mathcal{I}}(t) = \boldsymbol{\omega}_{\mathcal{EI}}^{\mathcal{I}}(t)$, and $\mathbf{v}_{\text{DO}}^{\mathcal{I}} = \mathbf{v}_{\text{EA}}^{\mathcal{I}}(t) - [\boldsymbol{\omega}_{\mathcal{DI}}^{\mathcal{I}}(t)]^{\wedge} R_{\mathcal{ID}}(t)\mathbf{r}_{\text{ED}}^{\mathcal{D}}(t)$.

The tuple $(R_{\mathcal{ID}}(t), \mathbf{r}_{\text{DO}}^{\mathcal{I}}(t), \boldsymbol{\omega}_{\mathcal{DI}}^{\mathcal{I}}(t), \mathbf{v}_{\text{DO}}^{\mathcal{I}}(t))$, $t \in [t_0, t_f]$ constitutes the reference trajectory, which is tracked by the

tuple $(R_{\mathcal{IS}^{\circ}}, \mathbf{r}_{\mathcal{S}^{\circ}\text{O}}^{\mathcal{I}}, \boldsymbol{\omega}_{\mathcal{S}^{\circ}\mathcal{I}}^{\mathcal{I}}, \mathbf{v}_{\mathcal{S}^{\circ}\text{O}}^{\mathcal{I}})$ of frame $\mathcal{S}^{\circ}$ by means of a feedback controller. The resulting *actual* trajectory of the camera, given as the tuple $(\hat{R}_{\mathcal{IC}}, \hat{\mathbf{r}}_{\text{CO}}^{\mathcal{I}}, \hat{\boldsymbol{\omega}}_{\mathcal{CI}}^{\mathcal{I}}, \hat{\mathbf{v}}_{\text{CO}}^{\mathcal{I}})$ of frame $\mathcal{C}$, if executed with small enough error, provides a ground truth dataset for the purpose of validating AstroSLAM's VO + RelDyn procedure. We provide the idealized trajectory tracking results, including the error in position and in attitude between the ideal and actual trajectories, as they pertain to the ASTROS-generated sequence, in Figure 15. We note in this figure that the worst case positional error in the radial direction is less than 0.2% for the segment of time considered in the sequence, between $t = 50$ s and $t = 178$ s. This error is less than the average error of 0.48% obtained in the DAWN RC3 sequence navigation solution results reported in Section 6.4.

For simplicity, we fix the position of the asteroid in the test arena, thus $\mathbf{r}_{\text{AO}}^{\mathcal{I}}(t) = \mathbf{r}_{\text{AO},0}^{\mathcal{I}}$, and we rotate it at a constant angular velocity around a single body axis, hence $\boldsymbol{\omega}_{\mathcal{AI}}^{\mathcal{A}}(t) = \boldsymbol{\omega}_{\mathcal{AI},0}^{\mathcal{A}} = [0 \ 0 \ \omega_a]$, starting from some initial orientation $R_{\mathcal{IA},0}$. To further simplify the planned maneuver in the ASTROS arena, we devise a planar orbital trajectory. We restrict the motion of the test-bed to the 2 + 1 planar case, by fixing the rotation of the upper stage and freeing the lower stage to move

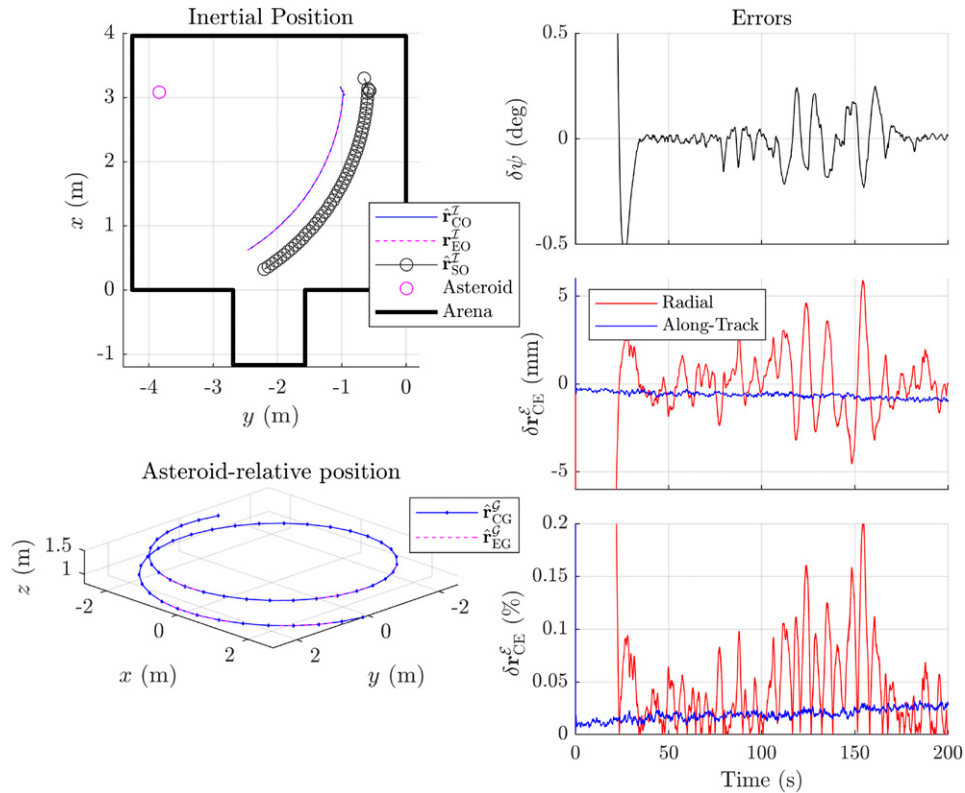


Figure 15. ASTROS lab experiment idealized orbital trajectory and tracking errors for the purpose of establishing a ground truth. Here we show the error state $\delta \mathbf{r}_{\text{CE}}^{\mathcal{E}} = [\delta r_{\text{AT}}^{\mathcal{E}}, \delta r_{\text{XT}}^{\mathcal{E}}, \delta r_{\text{RAD}}^{\mathcal{E}}]$ between the desired idealized orbital trajectory $\mathbf{r}_{\text{EO}}^{\mathcal{I}}$ and the actual trajectory $\hat{\mathbf{r}}_{\text{CO}}^{\mathcal{I}}$ obtained in the lab experiment from the on-board ASTROS EKF, which uses VICON and IMU measurements to establish a state estimate. We additionally keep track of the error between the maintained attitude and the schedule attitude by analyzing the error $\delta \psi$ of the yaw heading angle.

along the inertial x - y directions, and rotate around the inertial z direction. We impose that the vertical component of the asteroid's inertial position corresponds to the vertical component of the inertial position of the camera frame, expressed in inertial frame coordinates, or simply put, $\mathbf{r}_{CA}(t) \cdot \hat{\mathbf{n}}_3 \equiv 0$. This guarantees that, once the upper stage attitude is locked, the camera frame can only move in an x - y aligned plane and that this plane always intersects the center of the asteroid, itself manipulated by the robotic arm. It can now be assumed that the desired camera frame mimics the motion of the spacecraft frame in an emulated orbital motion. We mention the parameters of the tracked trajectory in Table 4, and we present, in Figure 15, sub-millimeter tracking error in the along-track direction and sub-5-millimeter tracking error in the radial direction.

Table 4. Parameters of the Tracked Ideal Trajectory.

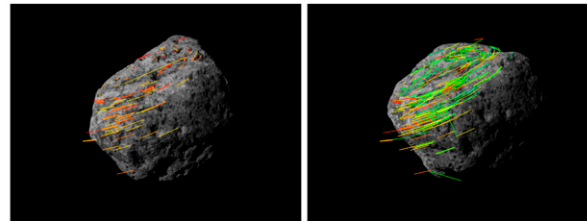
Parameter	Values
$\mathbf{r}_{EO,0}^T$	$[3.1150 \quad -0.9719 \quad -1.2228]^T$ m
$\mathbf{v}_{EO,0}^T$	$[-0.0165 \quad -0.0010 \quad 0.0000]^T$ m/s
$\mathbf{r}_{AO,0}^T$	$[3.083 \quad -3.839 \quad -1.223]^T$ m
$\boldsymbol{\omega}_{AZ,0}^A$	$[0 \quad 0 \quad 4/180\pi]^T$ rad
$R_{IA,0}$	$\begin{bmatrix} -0.8761 & 0.0084 & -0.4820 \\ -0.1041 & 0.9730 & 0.2061 \\ 0.4707 & 0.2308 & -0.8516 \end{bmatrix}$
$\mathbf{r}_{ED}^D$	$[0.3744 \quad -0.1278 \quad -0.1898]^T$
$R_{DE,0}$	$\begin{bmatrix} 0.1088 & -0.0016 & 0.9940 \\ 0.9940 & 0.0038 & -0.1088 \\ -0.0036 & 1.0000 & 0.0020 \end{bmatrix}$

To achieve the level of precision required to simulate orbital motion, all of the parameters relevant to the experiment were estimated using an accurate calibration scheme. Specifically, we carried out the accurate determination of the asteroid mounting boom length, the estimation of the RMS home position and attitude, the simultaneous calibration of the camera position and attitude relative to the upper stage and of the camera intrinsic parameters, using 2D-to-3D correspondences induced by taking images of a known 3D calibration target.

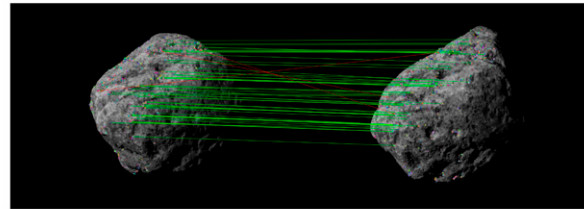
7.3. Lab experiment image-dataset

To capture the images, we used a Teledyne FLIRTM Flea3 visible-spectrum global-shutter camera along with a MegaPixel 25-135 mm tele-objective lens. When set at around 100 mm focal length, the lens produces a field-of-view angle of about 5° mimicking the navigation camera of a typical asteroid surveying mission. The camera imager captures images of array size 1600×1200 pixels. The size of the mock asteroid and working distance were chosen accordingly to produce an apparent size of the mock asteroid in the image corresponding to 700-800 pixels in the horizontal direction. Given the 10 cm diameter of the mock asteroid and the radial distance to the target of 2.84 m, this choice results in a ground resolution of $1, 25 \times 10^{-5}$ m/pixel. An on-board NVidia TX1 computer acquired images of the mock asteroid as the ASTROS platform maneuvered in the arena.

To emulate space-like lighting, typically characterized by collimated light rays arriving from a source infinitely far away, a tight-beam stage lighting source was used. The light source, a Source 4TM Ellipsoidal with a 5° beam angle



(a) Image ORB feature point tracking over 2 consecutive images on imagery of the in-lab mock asteroid. Lines of different colors indicate different feature point tracks.



(b) Image ORB feature point matching between two consecutive image frames. Green lines show inlier matches (kept) and red lines show outlier matches (rejected) by virtue of RANSAC geometric check.

Figure 16. Visualizing the image feature tracking performance of the front-end subsystem on the ASTROS in-lab generated sequence. (a) Image ORB feature point tracking over two consecutive images on imagery of the in-lab mock asteroid. Lines of different colors indicate different feature point tracks. (b) Image ORB feature point matching between two consecutive image frames. Green lines show inlier matches (kept) and red lines show outlier matches (rejected) by virtue of RANSAC geometric check.

constrained by dedicated optics, illuminated the target mock asteroid throughout the experiment. Note that the light source is fixed inertially. This mimics the scenario in space where, for the short duration of the navigation segment, there is negligible angular change in sunlight direction, when viewed inertially. Although the challenge of light back-scattering still persists due to the presence of atmosphere in the facility, the tight-beam light source produces very crisp and harsh shadowing in the captured images. A sample of the images captured in the ASTROS experiment and being processed by the front-end can be viewed in Figure 16.

7.4. Quantitative evaluation results

Figure 17 shows the trajectory of the AstroSLAM-estimated 3D position $\hat{\mathbf{r}}_{SA,k}^A, k = 1, \dots, N$, as derived from the solution $\{\hat{T}_{A_k S_k}\}_{k=0}^N$ and relating to the idealized simulated spacecraft frame $S_k, k = 1, \dots, N$, overlayed on top of the trajectory of the EKF-estimated 3D position $\hat{\mathbf{r}}_{CA,k}^A$, relating to the ASTROS camera frame. We thus assume that the EKF-estimated pose $\{\hat{T}_{A_k C_k}\}_{k=0}^N$ of the experimental camera

corresponds to the ground truth simulated spacecraft pose $\{T_{A_k S_k}\}_{k=0}^N$, allowing us to compare the AstroSLAM solution to the millimeter precision ground truth estimate. The procedure used here to obtain the quantitative evaluation of the trajectory error is similar to the one detailed in Section 6.3.

We note that, unlike the results of the DAWN RC3 sequence, when the RelDyn factors are included along with the VO solution, the navigation errors are not improved in terms of position and attitude errors, as may be observed in Figure 18. Specifically, the norms of both the position and attitude errors are similar in order of magnitude to those of the VO-only solution. The relative scale of the errors with respect to the orbital radius is telling, however. With a mean radius of $\sim 2.8\text{m}$ over the segment arc, the radial error represents a 0.98% percentage error on average, with a worst case of 2.1% for the VO + RelDyn solution, whereas, for the pure VO case, we have an average of 0.86% and a worse case of 1.6% percentage error. Nevertheless, we observe that the relative velocity, as illustrated in Figure 19, is affected by the odometric factor since its variation is prescribed in large part by the gravitational interaction. At the same time, due to the fact that these variations in velocity

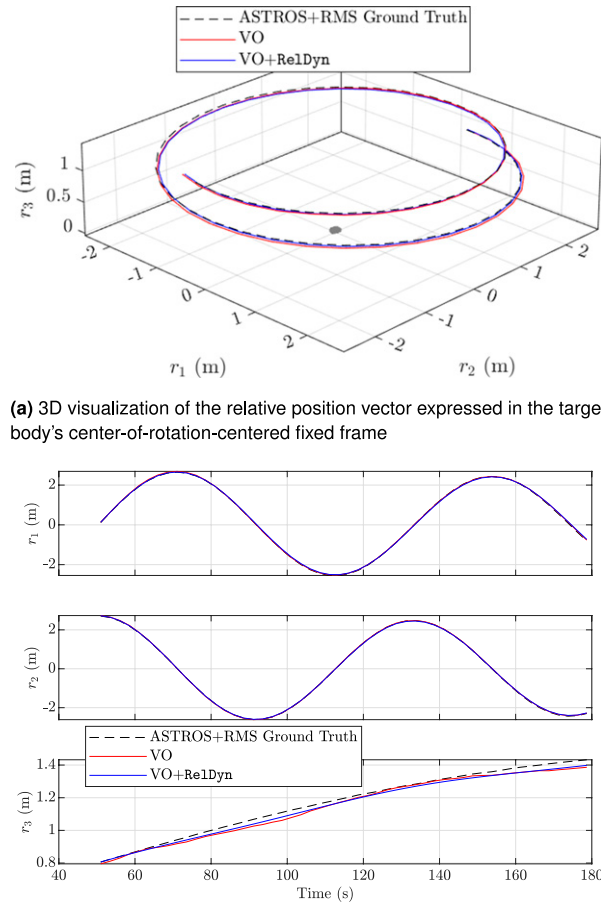


Figure 17. Relative position trajectory estimate versus ASTROS sequence ground truth obtained following the final incremental smoothing step. Here we compare the obtained relative position vectors $\mathbf{r}_{SA}^A = [r_1, r_2, r_3]^T$ from visual odometry alone (VO) in red and from VO + RelDyn in blue against the ground truth trajectory taken as the ASTROS + RMS-End-Effector relative position and attitude. (a) 3D visualization of the relative position vector expressed in the target body's center-of-rotation-centered fixed frame

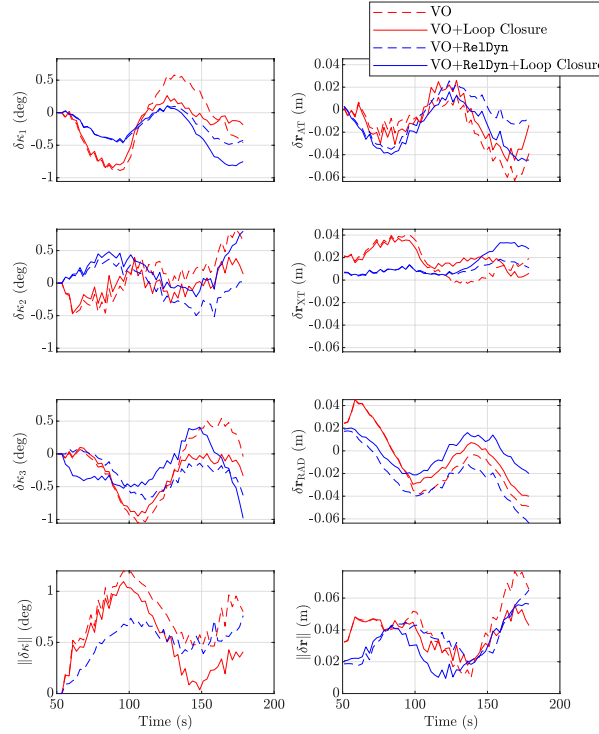


Figure 18. Time history of navigational errors with respect to the ground truth for the ASTROS lab sequence after final incremental smoothing step. In this figure, we compare the change in navigational errors for the VO and the VO + Reldyn cases when loop closure is included using the ASTROS dataset.

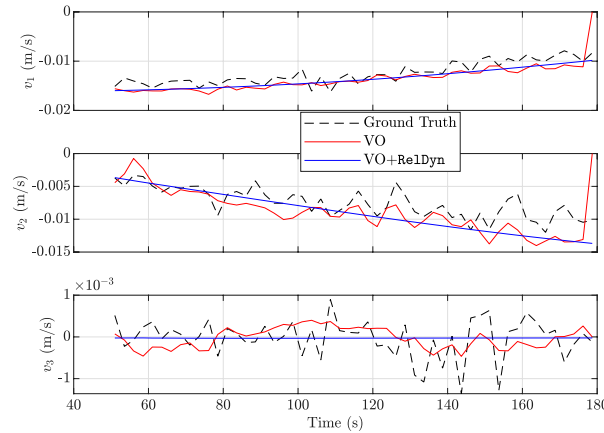


Figure 19. Relative velocity estimates versus estimated ground truth for ASTROS sequence. We note that the velocity estimates for pure VO, as drawn in red, are produce by means of taking a numerical derivative by finite differencing.

are captured by the odometric factor, we observe in Figure 20, that we have simultaneous convergence of both the standard gravity parameter $\mu_a \rightarrow 7.5 \times 10^{-2} \text{km}^3 \text{s}^{-2}$ and the planar location of the center-of-mass $[c_1, c_2] \rightarrow [0.0246 \text{ m}, 0.0076 \text{ m}]$. We notice, however, that the c_1 parameter has not quite settled around the expected value. This may be due to a combination of insufficient segment length observation and un-modelled deviations from the expected kinematic model during the experiment; indeed, there may be additional misalignments in the lab experimental setup which were not considered. These may be overcome by further perfecting the precision of the in-lab experiment.

Figure 21 illustrates the obtained map of the landmarks for the ASTROS sequence. We note here that a ground truth shape model is not available for this mock asteroid, and unlike the case of target asteroid Vesta in the last section, we forgo any analysis of the deviations of the estimated landmark positions.

7.5. Timing results for the ASTROS in-lab generated image-data sequence

In this section, we analyze the timing performance of the AstroSLAM algorithm with the ASTROS-generated image-

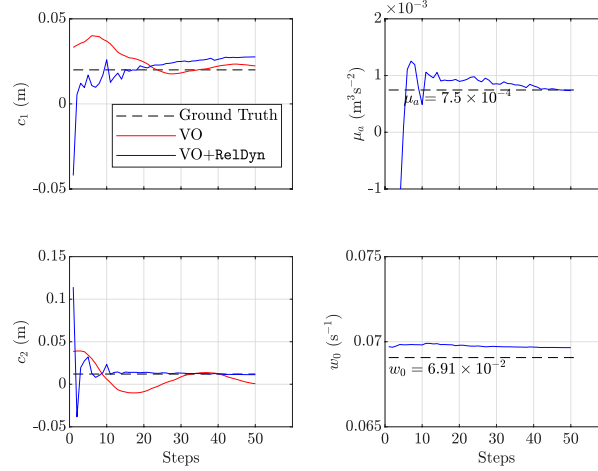


Figure 20. Time step history of the estimation of dynamical parameters. The estimated parameters are the two first components c_1, c_2 of the center-of-mass position vector $\hat{\mathbf{r}}_{AG}^G = [c_1, c_2, c_3]$, the standard gravity parameter μ_a of Asteroid (4) Vesta, and the small-body angular velocity magnitude $\hat{w}_0$, such that $\hat{w}_0 \hat{\mathbf{g}}_3 = \hat{\boldsymbol{\omega}}_{GL}$. For the VO case, we show in red the position of the center-of-mass location obtained from averaging all of the successfully initialized landmarks up to the most recent algorithm time step.

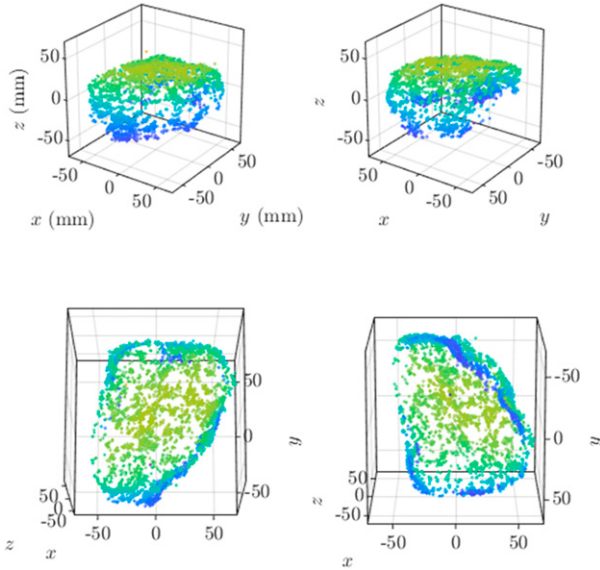


Figure 21. Estimated Landmark Positions for ASTROS Experiment. The color coding in these plots are not indicative of deviation or error, but are instead for providing a sense of depth to visualize the estimated landmark map, here viewed as a colored point cloud.

data sequence, as presented in Table 5. We immediately notice that the time spent in triangulation is a much smaller portion (18.4%) of the total time of the algorithm when compared to the DAWN RC3 scenario. This is due to the more lax threshold in the PnP RANSAC-based outlier rejection. We observe, however, that the solver time is similar to the DAWN RC3 scenario. Moreover, the overall run time on the hardware mentioned in Section 6.8, is sufficiently small to support

Table 5. Timing Results of the VO+RelDyn+Loop Closure Procedure in AstroSLAM for Obtaining the Incremental Smoothing Solution to the Totality of the 52-Frame ASTROS Image-Dataset Sequence.

Task	Total task time (s)	Percentage of total time (%)
Detect and describe features	2.080	29.3
Match features	1.407	19.8
Triangulate	1.308	18.4
Build Graph	0.047	0.7
Solve (iSAM2)	2.264	31.9
Total	7.106	100

online implementation. Indeed, in the considered idealized orbital scenario, the image arrival time difference is 134 s, which leaves ample time to perform at least one iteration of the algorithm.

8. Conclusions

A comprehensive vision-based relative navigation solution, called AstroSLAM, is proposed for the motion of a spacecraft in the vicinity of a celestial small-body. AstroSLAM solves for the navigation solution of a spacecraft under motion in the vicinity of a small body by exploiting monocular SLAM, sensor fusion, and relative motion priors. The developed motion priors are based on the dynamics of the spacecraft-small-body-Sun system, incorporating realistic perturbing effects, which affect the motion of the spacecraft in a non-negligible manner. We show how RelDyn, a vehicle dynamics model-based

factor, out-performs the state-of-the-art pre-integrated IMU factors commonly used in visual-inertial SLAM solutions. We further show that the appropriate inclusion of noise terms in the stochastic modeling can impact the smoothability of the state. The algorithm utilizes the factor graph formalism to cast the vision-based navigation problem as a SLAM smoothing problem that is solved efficiently using the iSAM2 solver (Kaess et al., 2012) and the GTSAM library (Dellaert and Kaess, 2017). The factor graph approach allows the incorporation of asynchronous measurements of diverse modalities, as well as the inclusion of kinematic and dynamic constraints, thus explicitly specifying the structure of the likelihood function. The algorithm was tested and its performance was validated using both real imagery and trajectory data sequences pertaining to the DAWN mission and in a controlled lab. environment. For the DAWN mission, the results demonstrate good baseline performance of AstroSLAM in a typical real-world mission scenario and also shows an improvement in terms of navigation error, landmark map reconstruction error, and global parameter estimation errors when compared to the state-of-the-art procedure of Setterfield et al. (2018a). The AstroSLAM algorithm was also tested against imagery and data produced in the ASTROS spacecraft simulation facility at Georgia Tech by emulating realistic lighting and motion conditions. The results of the in-lab validation further support the claim that appropriate modeling of the forces affecting the spacecraft is key to exploiting the odometric factor for corrective and smoothing effects once these are incorporated into the estimation problem.

Acknowledgments

The authors would like to thank the following individuals: Katherine Skinner for contributions in the implementation of the algorithm, Frank Dellaert for early discussions, as well as Andrew Liounis and Joshua Lyzhof of the NASA Goddard Space Flight Center for many invaluable discussion, comments, and suggestions.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the Early Stage Innovations (ESI) grant award 80NSSC18K0251 sponsored by the U.S. National Aeronautics and Space Administration (NASA).

ORCID iD

Mehregan Dor  <https://orcid.org/0000-0002-1670-0103>

Supplemental Material

Supplemental material for this article is available online.

References

- Anderson S and Barfoot TD (2015) Full steam ahead: exactly sparse Gaussian process regression for batch continuous-time trajectory estimation on SE(3). In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 157–164.
- Anderson S, Barfoot TD, Tong CH, et al. (2015) Batch nonlinear continuous-time trajectory estimation as exactly sparse Gaussian process regression. *Autonomous Robots* 39(3): 221–238.
- Andrieu MS and Crassidis JL (2013) Geometric integration of quaternions. *Journal of Guidance, Control, and Dynamics* 36(6): 1762–1767.
- Antreasian P, Adam C, Leonard J, et al. (2022) OSIRIS-REx proximity operations and navigation performance at Bennu. In The 32nd AIAA/AAS Space Flight Mechanics Meeting. San Diego, CA: AIAA.
- Baldini F, Harvard A, Chung S, et al. (2018) Autonomous small body mapping and spacecraft navigation via real-time SPC-SLAM. In: 69th IAF International Astronautical Congress (IAC), Bremen, Germany, October 2018. IAF.
- Barnouin O, Daly M, Palmer E, et al. (2020) Digital terrain mapping by the OSIRIS-REx mission. *Planetary and Space Science* 180: 104764.
- Bay H, Tuytelaars T and Van Gool L (2006) SURF: Speeded up robust features. In: European Conference on Computer Vision. Graz, Austria: Springer, 404–417.
- Bercovici B and McMahon JW (2019) Inertia parameter statistics of an uncertain small body shape. *Icarus* 328: 32–44.
- Bercovici B and McMahon JW (2019) Robust autonomous small-body shape reconstruction and relative navigation using range images. *Journal of Guidance, Control, and Dynamics* 42(7): 1473–1488.
- Berry K, Getzandanner K, Moreau M, et al. (2022) Contact with Bennu! Flight performance versus prediction of OSIRIS-REx TAG sample collection. In: *The 32nd AIAA/AAS Space Flight Mechanics Meeting*. San Diego, CA: AIAA.
- Bhaskaran S and Kennedy B (2014) Closed loop terminal guidance navigation for a kinetic impactor spacecraft. *Acta Astronautica* 103: 322–332.
- Bhaskaran S, Nandi S, Broschart S, et al. (2011) Small body landings using autonomous onboard optical navigation. *Journal of the Astronautical Sciences* 58(3): 409–427.
- Chirikjian G (2011) *Stochastic Models, Information Theory, and Lie Groups, Volume 2: Analytic Methods and Modern Applications, volume 2*. New York, NY: Springer Science & Business Media.
- Cho DM, Jung D and Tsiotras P (2009) A 5-DOF experimental platform for spacecraft rendezvous and docking. In: AIAA Infotech@ Aerospace Conference and AIAA Unmanned...Unlimited Conference, Seattle, WA: AIAA, p. 1869.
- Christian J and Lightsey G (2012) Onboard image-processing algorithm for a spacecraft optical navigation sensor system. *Journal of Spacecraft and Rockets* 49(2): 337–352.

- Church E, Bourbeau T, Curriden J, et al. (2020) Flash LIDAR on-orbit performance at asteroid Bennu. In: 43rd Annual AAS Guidance and Control Conference. Breckenridge, CO: AIAA. AAS, 20–148.
- Cocaud C and Kubota T (2010) SURF-based SLAM scheme using octree occupancy grid for autonomous landing on asteroids. In: Proceedings of the 10th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS), Sapporo, Japan: ESA, Vol. volume 29.
- Cocaud C and Kubota T (2012) Autonomous navigation near asteroids based on visual SLAM. In: Proceedings of the 23rd International Symposium on Space Flight Dynamics, Pasadena, CA, October 2012.
- Dellaert F (2021) Factor graphs: exploiting structure in robotics. *Annual Review of Control, Robotics, and Autonomous Systems* 4: 141–166.
- Dellaert F and Kaess M (2017) Factor graphs for robot perception. *Foundations and Trends in Robotics* 6(1-2): 1–139.
- Delpech M, Bissonnette V and Rastel L (2015) Vision-based navigation for proximity operations around asteroid 99942 Apophis. In: Proceedings of 25th International Symposium on Space Flight Dynamics. Munich, Germany.
- Dong J, Mukadam M, Dellaert F, et al. (2016) Motion planning as probabilistic inference using Gaussian processes and factor graphs. In: Robotics: Science and Systems (RSS). IEEE. Ann Arbor, MI, volume 12.
- Dor M, Skinner KA, Driver T, et al. (2021) Visual SLAM for asteroid relative navigation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Virtual, June 2021. IEEE, 2066–2075.
- Driver T, Dor M, Skinner K, et al. (2020) Space carving in space: a visual-SLAM approach to 3D shape reconstruction of a small celestial body. In: AAS/AIAA Astrodynamics Specialist Conference. Virtual, August 2020.
- Forster C, Carlone L, Dellaert F, et al. (2016) On-manifold pre-integration for real-time visual-inertial odometry. *IEEE Transactions on Robotics* 33(1): 1–21.
- Fourie D, Espinoza AT, Kaess M, et al. (2021) Characterizing marginalization and incremental operations on the bayes tree. In: Algorithmic Foundations of Robotics XIV: Proceedings of the Fourteenth Workshop on the Algorithmic Foundations of Robotics 14. Oulu, Finland, June 2021. Springer, 227–242.
- Fraser DC (1967) *A new technique for the optimal smoothing of data*. PhD Thesis. Massachusetts Institute of Technology.
- Galvez-López D and Tardos JD (2012) Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics* 28(5): 1188–1197.
- Gaskell R, Barnouin-Jha O, Scheeres D, et al. (2008) Characterizing and navigating small bodies with imaging data. *Meteoritics & Planetary Science* 43(6): 1049–1061.
- Gelb A, Kasper J, Nash R, et al. (1974) *Applied Optimal Estimation*. Cambridge, MA: MIT Press.
- Getzandanner KM, Berry KE, Antreasian PG, et al. (2022) Small-body proximity operations & TAG: navigation experiences & lessons learned from the OSIRIS-REx mission. *The 32nd AIAA/AAS Space Flight Mechanics Meeting*. San Diego, CA: AIAA.
- Goel A, Ul Islam A, Ansari A, et al. (2021) An introduction to inertial navigation from the perspective of state estimation [focus on education]. *IEEE Control Systems* 41(5): 104–128.
- Goossens S, Rowlands DD, Mazarico E, et al. (2021) Mass and shape determination of (101955) Bennu using differenced data from multiple OSIRIS-REx mission phases. *The Planetary Science Journal* 2(6): 219.
- Hartley R and Zisserman A (2004) *Multiple View Geometry in Computer Vision*. 2nd edition. Cambridge, UK: Cambridge University Press.
- Kaess M, Johannsson H, Roberts R, et al. (2012) iSAM2: incremental smoothing and mapping using the Bayes tree. *The International Journal of Robotics Research* 31(2): 216–235.
- Konopliv A, Asmar S, Park R, et al. (2014) The Vesta gravity field, spin pole and rotation period, landmark positions, and ephemeris from the dawn tracking and optical data. *Icarus* 240: 103–117.
- Lauretta D (2021) The OSIRIS-REx touch-and-go sample acquisition event and implications for the nature of the returned sample. In: 52nd Lunar and Planetary Science Conference. Virtual, March 2021.
- Leonard JM, Geeraert JL, Page BR, et al. (2020) OSIRIS-REx shape model performance during the navigation campaign. In: 43rd Annual AAS Guidance and Control Conference. Breckenridge, CO, February 2020.
- Leonard J, Moreau M, Antreasian P, et al. (2022) Cross-calibration of GNC and OLA LIDAR systems onboard OSIRIS-REx. In: 44th Annual AAS Guidance and Control Conference. Breckenridge, CO.
- Leonard JM, Geeraert JL, Pelgrift JY, et al. (2022) Navigation prediction performance during OSIRIS-REx proximity operations at (101955) Bennu. In: *The 32nd AIAA/AAS Space Flight Mechanics Meeting*. San Diego, CA: AIAA.
- Lepetit V, Moreno-Noguer F and Fua P (2009) Epnnp: an accurate O(n) solution to the pnp problem. *International Journal of Computer Vision* 81(2): 155–166.
- Lowe D (2004) Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60(2): 91–110.
- Matsuzaki T, Kameda H, Tsujimichi S, et al. (2000) Maneuvering target tracking using constant velocity and constant angular velocity model. In: IEEE International Conference on Systems, Man and Cybernetics, volume 5. Nashville, TN, pp. 3230–3234.
- Miller JK (2002) Determination of shape, gravity and rotation state of asteroid 433 Eros. *Icarus* 155(1): 3–17.
- Miller J (2019) Measurements and calibrations. In: *Planetary Spacecraft Navigation*. Cham: Springer International Publishing, 255–292.
- Miller J and Rourke K (1977) The application of differential VLBI to planetary approach orbit determination. *The Deep Space Network Progress Report* 42: 40.
- Nakath D, Clemens J and Schill K (2018) Multi-sensor fusion and active perception for autonomous deep space navigation. In: 21st International Conference on Information Fusion (FUSION). Cambridge, UK: IEEE, 2596–2605.
- Nakath D, Clemens J and Rachuy C (2020) Active asteroid-slam. *Journal of Intelligent and Robotic Systems* 99(2): 303–333.

- Nathues A, Sierks H, Gutierrez-Marques P, et al. (2011) *DAWN FC2 calibrated Vesta images v1.0*: NASA Planetary Data System.
- Nesnas IAD, Hockman BJ, Bandopadhyay S, et al. (2021) Autonomous exploration of small bodies toward greater autonomy for deep space missions. *Frontiers in Robotics and AI* 8: 1–26.
- Oestreich C, Espinoza AT, Todd J, et al. (2021) On-orbit inspection of an unknown, tumbling target using nasa's astrobe robotic free-flyers. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2039–2047.
- Preusker F, Scholten F, Matz K, et al. (2016) *DAWN FC2 Derived Vesta DTM SPG V1. 0*: NASA Planetary Data System.
- Prockter L, Murchie S, Cheng A, et al. (2002) The near shoemaker mission to asteroid 433 Eros. *Acta Astronautica* 51(1): 491–500.
- Rathinam A and Dempster A (2017) Monocular vision based simultaneous localization and mapping for close proximity navigation near an asteroid. In: *68th IAF International Astronautical Congress*. Adelaide, Australia: IAF.
- Rublee E, Rabaud V, Konolige K, et al. (2011) ORB: an efficient alternative to SIFT and SURF. In: *IEEE International Conference on Computer Vision*, Barcelona, Spain, 2564–2571.
- Russell C and Raymond C (2012) *The Dawn Mission to Minor Planets 4 Vesta and 1 Ceres*. New York, NY: Springer Science & Business Media.
- Sage AP and Melsa JL (1971) *Estimation theory with applications to communications and control*. McGraw-Hill Series in Systems Science. New York, NY: McGraw-Hill.
- Schaub H and Junkins JL (2003) *Analytical Mechanics of Space Systems*. Aiaa.
- Scheeres D (2016) *Orbital Motion in Strongly Perturbed Environments: Applications to Asteroid, Comet and Planetary Satellite Orbiters*. Springer Berlin Heidelberg: Springer Praxis Books. URL <https://books.google.com/books?id=kAuPDAAAQBAJ>
- Setterfield TP (2017) *On-orbit inspection of a rotating object using a moving observer*. PhD Thesis. Cambridge, MA: Massachusetts Institute of Technology.
- Setterfield TP, Miller D, Leonard JJ, et al. (2017) Smoothing-based estimation of an inspector satellite trajectory relative to a passive object. In: *2017 IEEE Aerospace Conference*, March 2017. pp. 1–11.
- Setterfield TP, Miller DW, Leonard JJ, et al. (2018) Mapping and determining the center of mass of a rotating object using a moving observer. *The International Journal of Robotics Research* 37(1): 83–103.
- Setterfield TP, Miller DW, Saenz-Otero A, et al. (2018) Inertial properties estimation of a passive on-orbit object using polhode analysis. *Journal of Guidance, Control, and Dynamics* 41(10): 2214–2231.
- Speyer JL and Chung WH (2008) *Stochastic Processes, Estimation, and Control*. Philadelphia, PA: SIAM.
- Starek JA, Açıkmeşe B, Nesnas IA, et al. (2016) *Spacecraft Autonomy Challenges for Next-Generation Space Missions*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1–48.
- Takeishi N, Yairi T, Tsuda Y, et al. (2015) Simultaneous estimation of shape and motion of an asteroid for automatic navigation. In: *IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, WA, MAY 2015. 2861–2866.
- Terán Espinoza A (2021) *Versatile inference algorithms using the bayes tree for robot navigation*. PhD Thesis. Cambridge, MA: Massachusetts Institute of Technology.
- Terui F, Ogawa N, Ono G, et al. (2020) Guidance, navigation, and control of Hayabusa 2 touchdown operations. *Astrodynamics* 4(4): 393–409.
- Thomas PC, Binzel RP, Gaffey MJ, et al. (1997) Vesta: spin pole, size, and shape from HST images. *Icarus* 128(1): 88–94.
- Tweddle BE (2013) *Computer vision based navigation for spacecraft proximity operations*. PhD Thesis. Cambridge, MA: Massachusetts Institute of Technology.
- Van Loan C (1978) Computing integrals involving the matrix exponential. *IEEE Transactions on Automatic Control* 23(3): 395–404.
- Villa J, McMahon J, Hockman B, et al. (2022) Autonomous navigation and dense shape reconstruction using stereo-photogrammetry at small celestial bodies. In: *44th Annual AAS Guidance, Navigation, and Control Conference*. Breckenridge, Colorado, February 2022: AIAA.
- Wang Y and Chirikjian GS (2008) Nonparametric second-order theory of error propagation on motion groups. *The International Journal of Robotics Research* 27(11-12): 1258–1273.
- Williams B (2002) Technical challenges and results for navigation of NEAR Shoemaker. *Johns Hopkins APL Technical Digest* 23(1): 34–45.
- Williams RL and Lawrence DA (2007) *Linear State-Space Control Systems*. Hoboken, NJ: John Wiley & Sons.
- Xie M, Escontrela A and Dellaert F (2020) *A Factor-Graph Approach for Optimization Problems with Dynamics Constraints*. arXiv preprint arXiv:2011.06194.
- Yan X, Indelman V and Boots B (2017) Incremental sparse GP regression for continuous-time trajectory estimation and mapping. *Robotics and Autonomous Systems* 87: 120–132.
- Yoshikawa M, Kawaguchi J, Fujiwara A, et al. (2015) HAY-ABUSA sample return mission. *Asteroids IV* 1: 397–418.