(Preprint) AAS 25-468

ROBOCENTRIC VISUAL SLAM FOR PRECISE LUNAR LANDING WITHOUT PRIOR GENERATED TERRAIN MAPS

Timothy L. Goulet[*], Keith A. LeGrand[†], Kari C. Ward[‡], and Kyle W. Smith[§]

Future manned and robotic space missions require safe and precise landings to reach intriguing scientific targets. However, traditional navigation techniques are ill-equipped for precision landing on bodies without consistent access to Earth-based positioning networks or finely-detailed terrain maps, such as distant asteroids or the lunar poles. As more spacecraft incorporate hazard avoidance lidar scans for in-flight safe landing site selection, an opportunity arises for producing a terrain map on-the-fly for extended visual navigation. A key challenge in leveraging this opportunity is that the scan itself provides only an erroneous estimate of terrain relative to the spacecraft. To effectively incorporate this scan into a navigation system, a robocentric square-root EKF-SLAM algorithm is proposed that builds an initial terrain map from the scan and refines estimates of the spacecraft states and identified landmark positions from subsequent terrain camera images. A generalized process for deriving robocentric equations of motion is presented and applied to a dead-reckoning spacecraft viewing a rotating celestial body. The proposed approach generates consistent and observable relative state estimates in a simulated lunar south pole descent and landing scenario with only a handful of opportunistic landmarks discovered in-flight, providing sufficient navigation accuracy to support precision landing.

INTRODUCTION

Whether on Earth, the Moon, or a remote asteroid, descent and landing is one of the most dangerous phases of a space mission [1]. These risks are exacerbated on distant celestial bodies, which lack traditional information sources like the GPS constellation, surface beacons, and radar networks. However, these high-risk missions often contain some of the most intriguing scientific targets [2]. Therefore, advancements in landing capabilities are critical enablers to future space exploration [3].

[*]M.S. Student and Draper Scholar, School of Aeronautics and Astronautics, Purdue University, 701 W Stadium Ave, West Lafayette, IN 47907

[†]Assistant Professor, School of Aeronautics and Astronautics, Purdue University, 701 W Stadium Ave, West Lafayette, IN 47907

[‡]Senior Member of the Technical Staff, Draper, 17629 El Camino Real Suite 470, Houston, TX 77058

[§]Principal Member of the Technical Staff, Draper, 17629 El Camino Real Suite 470, Houston, TX 77058

In particular, navigation accuracy and robustness in the terminal landing stage is essential to safe and precise landing.

In recent years, a number of national and commercial missions have attempted landing on the lunar surface. Many of these missions, including JAXA SLIM [4], Intuitive Machines Odysseus [5], Astrobotic Peregrine [6], ISRO Chandrayaan-2 [7], and CNSA Chang'e 5 [8], have all relied (or intended to rely) in some way on optical navigation techniques. Specifically, SLIM, Odysseus, and Peregrine all hosted terrain relative navigation (TRN) systems onboard [4–6], which compare images taken by the spacecraft to a prior-generated map of identifiable features to determine the spacecraft's states. TRN provides rich navigation information and has successfully aided landing on celestial bodies [9–11], navigating the ocean floor, [12, 13], and flying a helicopter on Mars [14], for example. However, TRN's reliance on prior-generated feature databases imposes major hurdles to its application in unmapped environments. Even for well-mapped celestial bodies like the Moon, these maps lack the resolution needed to perform TRN below a certain operating altitude near the surface. Therefore, many of the missions listed previously used a vertical final descent profile to achieve precision landing [4, 7, 8], which can make navigation easier but at the cost of more fuel use and more thruster firings. Ideally, navigation should be able to support precision landing while on the glide slope. Some long duration missions have the potential to generate comprehensive models of a body's surface during a long approach phase [15], but these methods are not suitable for a short and fast landing. However, there are alternative methods for generating a map in-flight from the field of robotics. Specifically, there are many well-established simultaneous localization and mapping (SLAM), visual odometry (VO), and visual inertial odometry (VIO) techniques (VIO is VO with the addition of an inertial measurement unit (IMU)), which generally estimate a robot's pose with the aid of observed features (SLAM approaches also estimate a map of the robot's environment) [16]. Within this class of algorithms, computational concerns and predominant interest in knowledge of the current spacecraft state lend this problem to filtering methods [16]. Many have been applied to spacecraft descent and landing navigation in previous publications.

The square-root extended information filter (EIF) in [17] performs real-time VIO for a spacecraft traversing over and landing on the lunar surface. The accompanying simulations demonstrate the filter's efficiency in processing large numbers of mapped landmarks (ML), which are prior known, and opportunistic landmarks (OL), which are identified in-flight, in real-time. The so-called VISINAV algorithm proposed in [18] uses a tightly-coupled extended Kalman filter (EKF) to process IMU, ML, and OL measurements during descent and landing. By the efficient incorporation of OL as VO features, the VISINAV algorithm is suitable for real-time applications, as shown in a sounding rocket flight experiment. The use case for the direction-of-motion measurement model from [19] has many similarities to the problem at hand, working exclusively with OL to support a VO algorithm. There are noted sensitivities to landmark mismatch, described more in-depth in [5]; however, this approach does not require any ML or *a priori* knowledge of the surface. Outside of these space-focused publications, there are many generic SLAM and VIO techniques to consider. Although spacecraft onboard computational constraints must be kept in mind, some relevant approaches include the multi-state constraint Kalman filter [20], the observability-constrained (OC) and first-estimate Jacobian (FEJ) EKF-SLAM algorithms [21], and robocentric SLAM/VIO approaches [22–24].

This paper focuses on navigation in the terminal stage of landing, during which the primary objective is hazard avoidance and precision landing at a chosen landing site. It is assumed that prior-generated terrain maps have insufficient resolution for use at this altitude. Furthermore, as in

[25, 26], the spacecraft takes a single lidar scan of the proposed landing site during final approach to produce a digital elevation map (DEM) for hazard avoidance and safe landing site selection. This DEM is illustrated alongside the terrain camera field-of-view (FOV) in Figure 1.

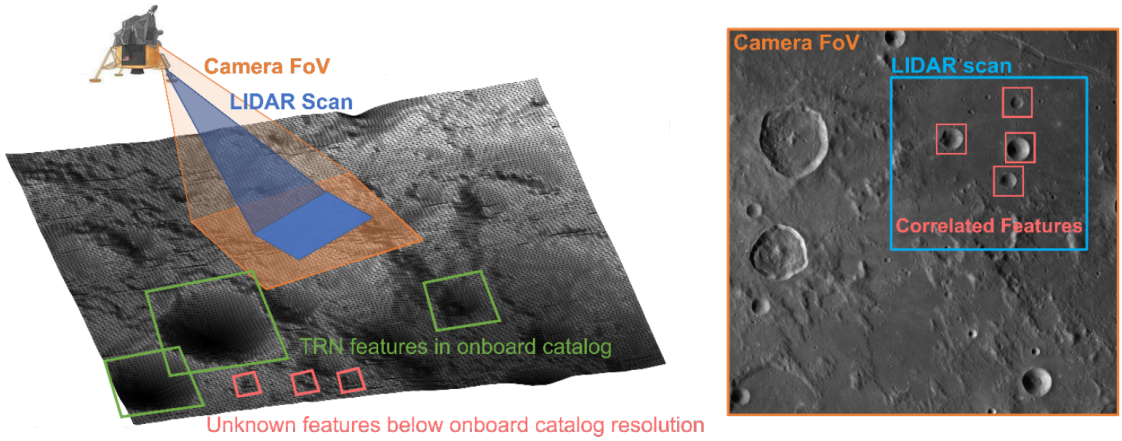


Figure 1: Lidar scan and camera field of view comparison with mission illustration.

Although intended only for hazard detection and avoidance, this finely detailed DEM provides additional navigation information that is underutilized in current operations. In contrast to TRN maps, the information provided by the DEM is only known relative to the spacecraft at the instant of formation, posing additional challenges. Furthermore, unavoidable errors in the DEM creation process and alignment of the DEM with a terrain camera means only estimated relative positions of landmarks are available. Therefore, landmark positions need to be estimated jointly with the spacecraft's state.

This paper proposes a robocentric square-root EKF-SLAM algorithm for short-term descent and landing navigation with a terrain camera and DEM. In the robocentric approach, global robot and map states are exchanged for relative states between the spacecraft and the landing site. This alternative state definition means that the robot is considered stationary by the filter and the environment moves around it. The robocentric approach is consistent and fully-observable, permitting more accurate linearizations than OC and FEJ EKF-SLAM [21]. The contributions of this paper are a generalized process for deriving robocentric kinematics in an environment with moving landmarks (such as landmarks rotating on a celestial body), a novel state and error kinematics model for the robocentric motion of a point on a rotating celestial body relative to a dead-reckoning spacecraft, a simple landmark map maintenance approach for a square-root Kalman filter, and use of a hazard avoidance lidar scan for navigation. The remainder of this paper will present a mathematical problem formulation, details of the employed methodology, and results from a simulated lunar descent and landing scenario.

PRELIMINARIES AND NOTATION

Throughout this paper, scalar quantities are denoted by unbolded letters, vectors are denoted by bold lower-case letters, and matrices are denoted by bold upper-case letters. Coordinate frames are denoted by calligraphic upper-case letters, e.g. $\mathcal{A} = (\mathcal{O}_A, \mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3)$. Here, $\mathcal{O}_A$ is the frame origin and $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$ are the frame's basis vectors which form a right-handed system. A left superscript

indicates the frame with respect to which a quantity was differentiated: e.g., ${}^{\mathcal{I}}\mathbf{v} = {}^{\mathcal{I}}d/dt\{\mathbf{r}\}$. The expression $[\mathbf{a}]_{\mathcal{M}}$ represents the array of pure vector $\mathbf{a}$'s coordinates with respect to the $\mathcal{M}$ frame. The subscript in the expression $\mathbf{a}_{\beta/\alpha,k}$ signifies a relative quantity of β with respect to α and given at time k.

Coordinate frame transformations are described via direction cosine matrix (DCM), $\mathbf{T}$, and scalar-first quaternions, $\bar{\mathbf{q}}$, e.g. $\mathbf{T}_{\mathcal{M}}^{\mathcal{L}}$ and $\bar{\mathbf{q}}_{\mathcal{M}}^{\mathcal{L}}$. Quaternion multiplication, denoted $\otimes$, is defined such that the order of quaternion multiplication and DCM multiplication is the same for any two transformations [27]. Coordinate frame transformations are passive and if the transformation is time-varying, an additional subscript is provided to indicate the appropriate time step.

Estimated quantities are marked with a circumflex, "$\hat{}$", whereas true quantities have no additional marking. For example, the true vector $\mathbf{a}$ has estimated value $\hat{\mathbf{a}}$. Time-derivatives are denoted with a dot, $\dot{}$. For a vector $[\mathbf{a}] = \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix}^T$, the cross-product matrix $[\mathbf{a}\times]$ is defined

$$[\mathbf{a}\times] = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \tag{1}$$

The generic functions $\mathbf{T}_p(\cdot)$, $\mathbf{T}_a(\cdot)$, $\bar{\mathbf{q}}(\cdot)$, and $\mathbf{e}(\cdot)$ denote the representation of their argument as a passive DCM, an active DCM, a quaternion set, and an Euler vector, respectively.

Coordinate Frames

The Moon-centered inertial (MCI) reference frame is defined $\mathcal{I} = (\mathcal{O}_m, \mathbf{i}_1, \mathbf{i}_2, \mathbf{i}_3)$, where $\mathcal{O}_m$ is the Moon's center of mass. The Moon-centered Moon-fixed (MCMF) reference frame is denoted $\mathcal{F} = (\mathcal{O}_m, \mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3)$, which is similar to the MCI frame but rotates with the Moon. DEM data is represented in its own frame, $\mathcal{D} = (\mathcal{O}_d, \mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3)$, where $\mathcal{O}_d$ is the frame origin at the center of the DEM scan and the basis vectors are defined relative to the surface. The IMU reference frame is denoted $\mathcal{B} = (\text{IMU}, \mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3)$ where IMU is the frame origin at the IMU's location onboard the spacecraft. Finally, $\mathcal{C} = (\text{TC}, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$ represents the terrain camera reference frame, where TC is the frame origin at the terrain camera's aperture, $\mathbf{c}_3$ is along the camera's boresight direction, and $\mathbf{c}_1$ and $\mathbf{c}_2$ reside within the image plane. The coordinate transformations $\mathcal{C} \leftrightarrow \mathcal{B}$, $\mathcal{I} \leftrightarrow \mathcal{F}$, and $\mathcal{F} \leftrightarrow \mathcal{D}$ are assumed to be known exactly.

PROBLEM FORMULATION

Kinematics

Let $[\mathbf{r}_{\text{IMU}/\mathcal{O}_m}]_{\mathcal{I}}$, ${}^{\mathcal{I}}[\mathbf{v}_{\text{IMU}/\mathcal{O}_m}]_{\mathcal{I}}$, and $\bar{\mathbf{q}}_{\mathcal{I}}^{\mathcal{B}}$ represent the IMU's MCI position vector, MCI velocity vector, and attitude quaternion, respectively. Then, the spacecraft motion is modeled by the continuous-time kinematics

$$[\dot{\mathbf{r}}_{\text{IMU}/\mathcal{O}_m}]_{\mathcal{I}} = [{}^{\mathcal{I}}\mathbf{v}_{\text{IMU}/\mathcal{O}_m}]_{\mathcal{I}} \tag{2}$$

$$[{}^{\mathcal{I}}\dot{\mathbf{v}}_{\text{IMU}/\mathcal{O}_m}]_{\mathcal{I}} = \mathbf{g}_{\mathcal{I}}([\mathbf{r}_{\text{IMU}/\mathcal{O}_m}]_{\mathcal{I}}) + \mathbf{T}_p(\bar{\mathbf{q}}_{\mathcal{I}}^{\mathcal{B}})^T\mathbf{a}_a \tag{3}$$

$$\dot{\bar{\mathbf{q}}}_{\mathcal{I}}^{\mathcal{B}} = \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}_g)\bar{\mathbf{q}}_{\mathcal{I}}^{\mathcal{B}} \tag{4}$$

where $\boldsymbol{\omega}_g$ is the true spacecraft angular velocity, $\mathbf{g}_{\mathcal{I}}(\mathbf{r})$ is the inertial gravitational acceleration at $\mathbf{r}$, $\mathbf{a}_a$ is the true non-gravitational acceleration as experienced by the IMU, and $\boldsymbol{\Omega}(\cdot)$ is a quaternion

product operator (see [27]). Other quantities of interest include the constant unknown camera bias $\mathbf{b}_{\text{TC}}$ and MCI landing site position vector $[\mathbf{r}_{\text{LS}/\mathcal{O}_m}]_{\mathcal{I}}$. Because the descent and landing scenario considered lasts only minutes, the effects of planetary precession and other rotational perturbations are assumed to be negligible, such that the MCI landing site location is governed by

$$[\dot{\mathbf{r}}_{\text{LS}/\mathcal{O}_m}]_{\mathcal{I}} = [\boldsymbol{\omega}_m\times][\mathbf{r}_{\text{LS}/\mathcal{O}_m}]_{\mathcal{I}} \tag{5}$$

where $\boldsymbol{\omega}_m$ is the Moon angular velocity vector in the MCI frame.

Measurements

The only navigation sensor available for state corrections is a terrain camera, which is modeled as a simple pinhole camera. The terrain camera provides noisy and biased pixel measurements of visible landmarks. Assuming that a feature extraction algorithm has already identified landmarks within an image, for a given landmark L_j, the corresponding measurement is given by the vector

$$\mathbf{z}_j = \begin{bmatrix} \text{f}\cdot(x_{L_j/\text{TC}}/z_{L_j/\text{TC}}) \\ \text{f}\cdot(y_{L_j/\text{TC}}/z_{L_j/\text{TC}}) \end{bmatrix} + \mathbf{b}_{\text{TC}} + \boldsymbol{v}_{\text{TC}} \tag{6}$$

where

$$[\mathbf{r}_{L_j/\text{TC}}]_{\mathcal{C}} = \begin{bmatrix} x_{L_j/\text{TC}} & y_{L_j/\text{TC}} & z_{L_j/\text{TC}} \end{bmatrix}^T, \tag{7}$$

$\mathbf{b}_{\text{TC}}$ is the unknown constant bias, f is the camera's known focal length in pixels, and the noise $\boldsymbol{v}_{\text{TC}}$ is zero-mean, Gaussian, and white. If the positions of landmarks with respect to the landing site in the DEM frame, $[\mathbf{r}_{L_j/\text{LS}}]_{\mathcal{D}}$, are known constants, the position vector $[\mathbf{r}_{L_j/\text{TC}}]_{\mathcal{C}}$ may be obtained via vector addition as

$$[\mathbf{r}_{L_j/\text{TC}}]_{\mathcal{C}} = [\mathbf{r}_{\text{IMU}/\text{TC}}]_{\mathcal{C}} + [\mathbf{r}_{\mathcal{O}_m/\text{IMU}}]_{\mathcal{C}} + [\mathbf{r}_{\text{LS}/\mathcal{O}_m}]_{\mathcal{C}} + [\mathbf{r}_{L_j/\text{LS}}]_{\mathcal{C}} \tag{8}$$

The vector from the terrain camera to the IMU, $[\mathbf{r}_{\text{IMU}/\text{TC}}]_{\mathcal{B}}$, is assumed to be known from spacecraft construction.

State Space Model

Because the IMU kinematics and camera measurement equation are nonlinear, this problem may be written as a generic nonlinear, continuous-discrete state space model

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) + \mathbf{G}(t)\mathbf{w}(t) \tag{9}$$

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \boldsymbol{v}_k \tag{10}$$

where $\mathbf{x}$ is the state vector containing the relevant states previously identified, $\mathbf{f}$ is a generalization of (2)-(5), $\mathbf{u}(t)$ contains the IMU measurements, $\mathbf{G}$ is a process noise mapping matrix, $\mathbf{w}$ is a zero-mean, Gaussian white-noise vector, $\mathbf{h}$ is a generalization of (8) and (6), and $\boldsymbol{v}_k$ is a generalization of $\boldsymbol{v}_{\text{TC}}$.

Regardless of the state vector form, the distribution of the process and measurement noise is assumed to be known, i.e. $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$ and $\boldsymbol{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$. The overarching goal is to estimate the distribution of $\mathbf{x}$ at each time-step using only IMU and camera measurements. These camera measurements are of landmarks identified within the DEM, although the only available knowledge of their position is an erroneous estimate. This means that the landmark positions are unknown in the global frame and imperfectly known relative to the spacecraft.

METHODOLOGY

The proposed approach is a robocentric square-root EKF-SLAM algorithm that builds an initial OL map using the results of the DEM. EKF-SLAM is chosen for its improved accuracy over EIF approaches, which are generally preferred for building large maps in real-time [16]. In this instance, spacecraft storage constraints and the computational burden of landmark identification, landmark extraction, and data association make tracking a large map infeasible. As for using a square-root formulation, this will yield better numerical performance and addresses issues seen during simulation of the filter. Finally, the choice of robocentric approach provides favorable observability and consistency characteristics compared to traditional EKF-SLAM. These concepts will be expanded upon in the following sections. Figure 2 illustrates how this process fits into a general landing scenario.

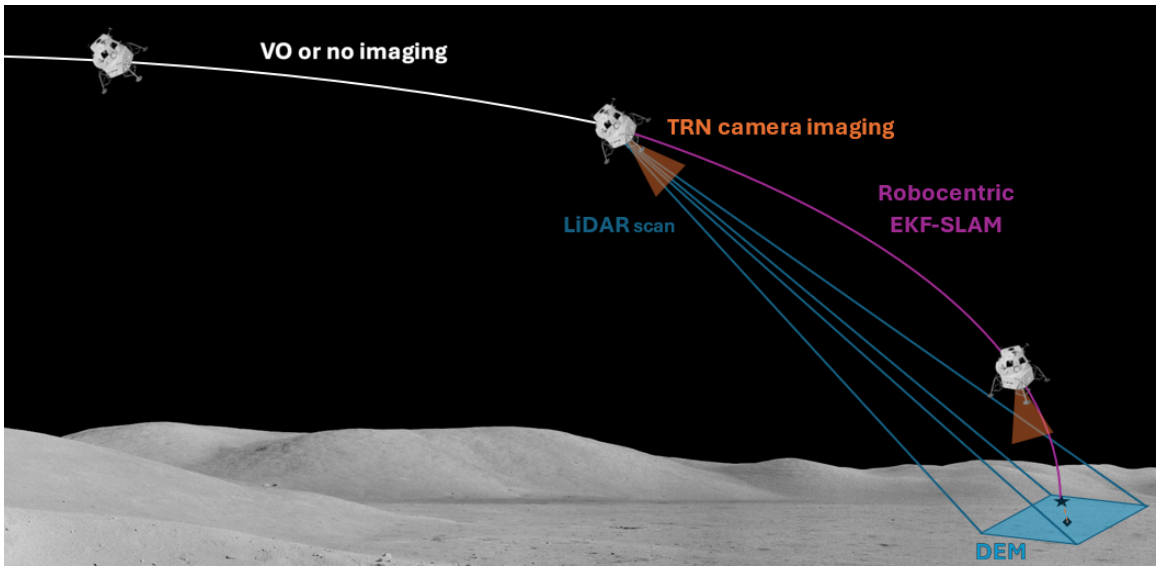


Figure 2: Landing illustration with methodology annotated (Moon and lander images credit of NASA [28, 29]).

Robocentric State Formulation

There are two options for a state vector in this problem. A *global* state vector would include variables such as $[\mathbf{r}_{\text{IMU}/\mathcal{O}_m}]_{\mathcal{I}}$, ${}^{\mathcal{I}}[\mathbf{v}_{\text{IMU}/\mathcal{O}_m}]_{\mathcal{I}}$, $[\mathbf{r}_{\text{LS}/\mathcal{O}_m}]_{\mathcal{I}}$, and $\bar{\mathbf{q}}^{\mathcal{B}}_{\mathcal{I}}$. Global representations are more common and have known kinematics; however, with only a camera, the system is unobservable and the EKF-SLAM estimator is inconsistent when linearizing about the *a priori* state estimate (see [21] and [30]). This is caused by the linearized system having less unobservable degrees of freedom than the nonlinear global SLAM problem. While modified approaches can address the consistency issue, notably OC and FEJ EKF-SLAM, these methods use previous state estimates for linearization, which can degrade performance and diverge without a sufficiently-accurate initial estimate [21].

The alternative approach is a *relative* state vector, consisting of the relative states $[\mathbf{r}_{\text{LS}/\text{IMU}}]_{\mathcal{B}}$, ${}^{\mathcal{I}}[\mathbf{v}_{\text{LS}/\text{IMU}}]_{\mathcal{B}}$, and $\bar{\mathbf{q}}^{\mathcal{B}}_{\mathcal{D}}$ between the IMU and the landing site. While the relative states can be described in any frame, the spacecraft body or *robocentric* frame $\mathcal{B}$ offers distinct advantages. The system is fully-observable and a traditional EKF-SLAM approach (like that in [16]) produces con-

sistent estimates while linearizing about the *a priori* estimate [22]. This makes robocentric SLAM less sensitive to large initial errors and ensures a more-accurate characterization of filter performance.

Robocentric approaches have received less attention compared to standard methods in terrestrial SLAM applications because, in those settings, the robocentric approach incurs higher computational cost and loop closure error [21]. However, the nature of descent and landing problems mitigate these disadvantages. The large computational cost of robocentric SLAM is incurred by propagating the motion of every landmark state instead of just a few robot states. By storing all landmark coordinates within the DEM, only one point within the DEM (in this case, the landing site) must be propagated at each time step. Even for descent and landing problems without a DEM, the quasi-planar distribution of landmarks on a body's surface permits a similar approach. Furthermore, the lander will not revisit locations in a descent and landing scenario. Combined with a relatively-stable attitude profile, this means that landmarks which fall outside of the terrain camera's FOV will generally not return to the image;if they do, These two characteristics make descent and landing SLAM problems unique from typical terrestrial robotic SLAM applications and further justify a robocentric approach.

Like in [23], the robocentric state vector developed here includes some global information. Here, global information appears in the form of the spacecraft pose in the MCI frame at the DEM scan time t_d (denoted with a subscript $_d$) and a relative position vector from the lidar scanner to the landing site in the IMU reference frame. These values permit recovery of the global states as required for propagation or other mission needs, and unlike [23] these values are of past states and so function more as parameters.

With this, the robocentric state vector is

$$\mathbf{x}_k = \begin{bmatrix} [\mathbf{r}_{\text{LS/IMU},k}]_{\mathcal{B}}^T & [{}^{\mathcal{I}}\mathbf{v}_{\text{LS/IMU},k}]_{\mathcal{B}}^T & \mathbf{b}_{\text{TC},k}^T & (\bar{\mathbf{q}}_{\mathcal{D},k}^{\mathcal{B}})^T & [\mathbf{r}_{\text{IMU}/\mathcal{O}_m,d}]_{\mathcal{I}}^T & (\bar{\mathbf{q}}_{\mathcal{I},d}^{\mathcal{B}})^T & [\mathbf{r}_{\text{LS/IMU},d}]_{\mathcal{B}}^T & \mathbf{m}_k{}^T \end{bmatrix}^T \tag{11}$$

where $\mathbf{m}$ contains landmark locations in $\mathcal{D}$. Once again, note that some states are exclusively values at the DEM scan time, denoted by the subscript d. These values will therefore not be subject to propagation or process noise; however, they may be updated if any onboard sensors provide global state information. For this particular state formulation, (8) is computed at time step k as

$$[\mathbf{r}_{L_j/\text{TC},k}]_{\mathcal{C}} = \mathbf{T}_{\mathcal{B}}^{\mathcal{C}}[\mathbf{r}_{\text{IMU/TC}}]_{\mathcal{B}} + \mathbf{T}_{\mathcal{B}}^{\mathcal{C}}[\mathbf{r}_{\text{LS/IMU},k}]_{\mathcal{B}} + \mathbf{T}_{\mathcal{B}}^{\mathcal{C}}\mathbf{T}_p(\bar{\mathbf{q}}_{\mathcal{D},k}^{\mathcal{B}})[\mathbf{r}_{L_j/\text{LS}}]_{\mathcal{D}} \tag{12}$$

The corresponding measurement Jacobian for this robocentric formulation, $\mathbf{H}$, is provided in appendix A.

Robocentric Kinematics

This section details how to derive robocentric propagation equations for the *relative* kinematic quantities (position, velocity, etc) between two *globally*-located objects (e.g. spacecraft and landing site), each with their own independent propagation equations (e.g. (2-5)). In practice, it may not be possible for the robocentric propagation equations to depend only on relative kinematic quantities. For example, gravitational acceleration acting on a spacecraft depends on the spacecraft's global position. Therefore, a global reference trajectory for one or both of the objects may be necessary. If the global propagation equation for either object only needs a global kinematic quantity, only one reference trajectory is needed because the other can be found by adding or subtracting the relative quantity. In this case, which object should be used for the reference trajectory depends

on the problem at hand, but some worthwhile considerations are how quickly global propagation equations can be evaluated and which reference trajectory would have less uncertainty throughout a scenario, because in a filter this reference trajectory will be based on an erroneous initial estimate. It is also important to note that this differs from the robocentric SLAM algorithms in [22–24], which consider stationary landmarks. The propagation equations in those papers can be thought of as a special case of the more generalized process herein, in which (globally) the landmarks are stationary and thus their robocentric motion is entirely dependent on the robot's global motion.

General Process Let $\mathbf{x}_k$ and $\mathbf{y}_k$ represent the global state vectors of two dynamical systems at the discrete time step k. The global state vectors are assumed to include all necessary states to describe the objects' motion, thereby yielding discrete-time propagation equations

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{x,k-1}, k-1, \Delta t) \tag{13}$$

$$\mathbf{y}_k = \mathbf{g}(\mathbf{y}_{k-1}, \mathbf{u}_{y,k-1}, k-1, \Delta t) \tag{14}$$

where $\mathbf{u}_{x,k-1}$ and $\mathbf{u}_{y,k-1}$ represent the control inputs for their respective systems at time step $k-1$ and Δt is the time elapsed between successive time steps. Note that $\mathbf{x}_k$ and $\mathbf{y}_k$ are assumed to take values at the same discrete time steps. Furthermore, assume the functions $\mathbf{f}$ and $\mathbf{g}$ are real analytic in the time step Δt.

The primary challenge of deriving robocentric equations of motion is to eliminate or minimize reference to global kinematic quantities (i.e. position, velocity). Without loss of generality, assume that $\mathbf{x}_k$ and $\mathbf{y}_k$ contain global position ($\mathbf{r}_{x,k}$, $\mathbf{r}_{y,k}$) and velocity ($\mathbf{v}_{x,k}$, $\mathbf{v}_{y,k}$) vectors in a common reference frame. These $\mathbf{r}$ and $\mathbf{v}$ vectors could alternatively represent other quantities (e.g. temperature, mass) and their time derivatives without modification to this derivation; however, for easier interpretation, $\mathbf{r}$ and $\mathbf{v}$ will be referred to as position and velocity, respectively. Let

$$\mathbf{r}_{x,k} = \mathbf{f}_r(\mathbf{x}_{k-1}, \mathbf{u}_{x,k-1}, k-1, \Delta t) \tag{15}$$

$$\mathbf{v}_{x,k} = \mathbf{f}_v(\mathbf{x}_{k-1}, \mathbf{u}_{x,k-1}, k-1, \Delta t) \tag{16}$$

$$\mathbf{r}_{y,k} = \mathbf{g}_r(\mathbf{y}_{k-1}, \mathbf{u}_{y,k-1}, k-1, \Delta t) \tag{17}$$

$$\mathbf{v}_{y,k} = \mathbf{g}_v(\mathbf{y}_{k-1}, \mathbf{u}_{y,k-1}, k-1, \Delta t) \tag{18}$$

Equations (15)-(18) merely label the vector equations from (13) and (14) that propagate the position (subscript r) and velocity (subscript v) vectors. Without loss of generality, let the relative kinematic vectors of interest be

$$\mathbf{r}_{rel,k} = \mathbf{r}_{y,k} - \mathbf{r}_{x,k} \tag{19}$$

$$\mathbf{v}_{rel,k} = \mathbf{v}_{y,k} - \mathbf{v}_{x,k} \tag{20}$$

Substitution of (15)-(18) into (19) and (20) produce the nonlinear robocentric propagation equations, which can be separated into functions that are closed recursive expressions of the relative quantites and functions that still rely on global quantities as

$$\begin{aligned}\mathbf{r}_{rel,k} =& \mathbf{b}_r(\mathbf{r}_{rel,k-1}, \mathbf{v}_{rel,k-1}, \mathbf{u}_{x,k-1}, \mathbf{u}_{y,k-1}, k-1, \Delta t) \\ &+ \mathbf{c}_r(\mathbf{r}_{rel,k-1}, \mathbf{v}_{rel,k-1}, \mathbf{x}_{k-1}, \mathbf{y}_{k-1}, \mathbf{u}_{x,k-1}, \mathbf{u}_{y,k-1}, k-1, \Delta t)\end{aligned} \tag{21}$$

$$\begin{aligned}\mathbf{v}_{rel,k} =& \mathbf{b}_v(\mathbf{r}_{rel,k-1}, \mathbf{v}_{rel,k-1}, \mathbf{u}_{x,k-1}, \mathbf{u}_{y,k-1}, k-1, \Delta t) \\ &+ \mathbf{c}_v(\mathbf{r}_{rel,k-1}, \mathbf{v}_{rel,k-1}, \mathbf{x}_{k-1}, \mathbf{y}_{k-1}, \mathbf{u}_{x,k-1}, \mathbf{u}_{y,k-1}, k-1, \Delta t)\end{aligned} \tag{22}$$

Here, the $\mathbf{b}$ functions reflect the propagation exclusively in terms of relative kinematic quantities; the $\mathbf{c}$ functions also incorporate the global states. Assuming the $\mathbf{c}$ functions are necessary (i.e. the propagation requires some knowledge of the global states), the goal is to maximize the contribution of the $\mathbf{b}$ functions and thereby minimize the contributions of the $\mathbf{c}$ functions. In other words, the goal is to minimize the dependence of the propagation on the global quantities that lead to observability issues in a visual SLAM system.

Beginning with the global propagation equations from (15)-(18) along with the relationships defined by (19) and (20),

$$\mathbf{r}_{rel,k} = \mathbf{g}_r(\mathbf{y}_{k-1}, \mathbf{u}_{y,k-1}, k-1, \Delta t) - \mathbf{f}_r(\mathbf{x}_{k-1}, \mathbf{u}_{x,k-1}, k-1, \Delta t) \tag{23}$$

$$\mathbf{v}_{rel,k} = \mathbf{g}_v(\mathbf{y}_{k-1}, \mathbf{u}_{y,k-1}, k-1, \Delta t) - \mathbf{f}_v(\mathbf{x}_{k-1}, \mathbf{u}_{x,k-1}, k-1, \Delta t) \tag{24}$$

Equations (23) and (24) trivially satisfy the form of (21) and (22), if $\mathbf{b}_r = \mathbf{b}_v = 0$. To produce a more useful expression, recall that $\mathbf{f}$ and $\mathbf{g}$ are assumed to be real-analytic with respect to the time step Δt, meaning $\mathbf{f}$ and $\mathbf{g}$ may be represented via a Maclaurin series about $\Delta t = 0$. This Maclaurin series for the functions $\mathbf{f}_r$, $\mathbf{g}_r$, $\mathbf{f}_v$, and $\mathbf{g}_v$ in particular will be a kinematic expansion of position, velocity, acceleration, etc at time step $k-1$. Therefore, the position and velocity terms at time step $k-1$ can be "pulled out" of (15)-(18).

$$\begin{aligned}\mathbf{r}_{rel,k} =& \mathbf{r}_{y,k-1} + \mathbf{v}_{y,k-1}\Delta t + \mathbf{g}_{r,l}(\mathbf{y}_{k-1}, \mathbf{u}_{y,k-1}, k-1, \Delta t) - \mathbf{r}_{x,k-1} - \mathbf{v}_{x,k-1}\Delta t \\ & - \mathbf{f}_{r,l}(\mathbf{x}_{k-1}, \mathbf{u}_{x,k-1}, k-1, \Delta t)\end{aligned} \tag{25}$$

$$\mathbf{v}_{rel,k} = \mathbf{v}_{y,k-1} + \mathbf{g}_{v,l}(\mathbf{y}_{k-1}, \mathbf{u}_{y,k-1}, k-1, \Delta t) - \mathbf{v}_{x,k-1} - \mathbf{f}_{v,l}(\mathbf{x}_{k-1}, \mathbf{u}_{x,k-1}, k-1, \Delta t) \tag{26}$$

The l subscript on the functions $\mathbf{f}_{r,l}$, $\mathbf{g}_{r,l}$, $\mathbf{f}_{v,l}$, and $\mathbf{g}_{v,l}$ denotes "leftovers"–the remainder of the propagation after pulling out the position and/or velocity vectors. At this point, combining like terms via the relationships (19) and (20) yields

$$\begin{aligned}\mathbf{r}_{rel,k} =& \mathbf{r}_{rel,k-1} + \mathbf{v}_{rel,k-1}\Delta t + \mathbf{g}_{r,l}(\mathbf{y}_{k-1}, \mathbf{u}_{y,k-1}, k-1, \Delta t) \\ & - \mathbf{f}_{r,l}(\mathbf{x}_{k-1}, \mathbf{u}_{x,k-1}, k-1, \Delta t)\end{aligned} \tag{27}$$

$$\mathbf{v}_{rel,k} = \mathbf{v}_{rel,k-1} + \mathbf{g}_{v,l}(\mathbf{y}_{k-1}, \mathbf{u}_{y,k-1}, k-1, \Delta t) - \mathbf{f}_{v,l}(\mathbf{x}_{k-1}, \mathbf{u}_{x,k-1}, k-1, \Delta t) \tag{28}$$

Now, non-trivial $\mathbf{b}$ functions are available to reduce the dependence on the global states in the $\mathbf{c}$ functions, which in this case consist of the difference of the leftover functions. Equations (27) and (28) are produced for the generic system described by (13) and (14); however, there are some special cases which deserve further attention. Often in SLAM applications, the landmarks will be stationary or on an uncontrolled ballistic path. This lack of control input can simplify the derivation process. Additionally, if only global kinematic quantities are necessary for one of the global reference trajectories (e.g. only the position vector from $\mathbf{x}_{k-1}$ is needed), only one reference trajectory is needed–the other can be found by adding the relative quantity. Both of these benefits will be utilized in the problem-specific derivation.

As a final note, this derivation produced propagation equations for relative kinematic quantities without consideration of reference frames. Further modification may be necessary to produce relative kinematic quantities exclusively in a body frame; however, this will depend on the specific problem and so is not covered here. However, an example of this is provided for deriving robocentric equations in the lunar landing SLAM problem.

Application Applying this general process to the spacecraft descent and landing problem, consider the approximate discretized IMU and landing site kinematics derived from (2-5) [27]:

$$[\mathbf{r}_{\text{IMU}/\mathcal{O}_m,k}]_{\mathcal{I}} = [\mathbf{r}_{\text{IMU}/\mathcal{O}_m,k-1}]_{\mathcal{I}} + [{}^{\mathcal{I}}\mathbf{v}_{\text{IMU}/\mathcal{O}_m,k-1}]_{\mathcal{I}}\Delta t + (\mathbf{T}^{\mathcal{B}}_{\mathcal{I},k-1})^T\Delta\mathbf{r}_{ng,k} + \frac{1}{2}\mathbf{g}_{k-1}\Delta t^2 \tag{29}$$

$$[{}^{\mathcal{I}}\mathbf{v}_{\text{IMU}/\mathcal{O}_m,k}]_{\mathcal{I}} = [{}^{\mathcal{I}}\mathbf{v}_{\text{IMU}/\mathcal{O}_m,k-1}]_{\mathcal{I}} + (\mathbf{T}^{\mathcal{B}}_{\mathcal{I},k-1})^T\Delta\mathbf{v}_{ng,k} + \mathbf{g}_{k-1}\Delta t \tag{30}$$

$$\bar{\mathbf{q}}^{\mathcal{B}}_{\mathcal{I},k} = \bar{\mathbf{q}}(\Delta\boldsymbol{\theta}_{g,k}) \otimes \bar{\mathbf{q}}^{\mathcal{B}}_{\mathcal{I},k-1} \tag{31}$$

where

$$\Delta\mathbf{r}_{ng,k} = \frac{1}{2}\left(\mathbf{I}_{3\times3} + \frac{1}{3}[\Delta\boldsymbol{\theta}_{g,k}\times]\right)\Delta\mathbf{v}_{a,k}\Delta t, \tag{32}$$

$$\Delta\mathbf{v}_{ng,k} = \left(\mathbf{I}_{3\times3} + \frac{1}{2}[\Delta\boldsymbol{\theta}_{g,k}\times]\right)\Delta\mathbf{v}_{a,k}, \tag{33}$$

$\Delta\boldsymbol{\theta}_{g,k} = \boldsymbol{\omega}_{g,k}\Delta t$, $\boldsymbol{\omega}_{g,k}$ is the bias-corrected IMU angular velocity measurement at time step k, $\Delta\mathbf{v}_{a,k} = \mathbf{a}_{a,k}\Delta t$, $\mathbf{a}_{a,k}$ is the bias-corrected IMU non-gravitational acceleration measurement at time step k, and $\mathbf{g}_{k-1} = \mathbf{g}_{\mathcal{I}}([\mathbf{r}_{\text{IMU}/\mathcal{O}_m,k-1}]_{\mathcal{I}})$ for brevity. Any gravity model will work here; however, for an EKF application, the gravity Jacobian will be needed.

For the discretized landing site propagation, assuming a constant lunar angular velocity vector means the propagation can be expressed as the simple vector rotation

$$[\mathbf{r}_{\text{LS}/\mathcal{O}_m,k}]_{\mathcal{I}} = \mathbf{T}_a(\boldsymbol{\omega}_m\Delta t)[\mathbf{r}_{\text{LS}/\mathcal{O}_m,k-1}]_{\mathcal{I}} \tag{34}$$

Similarly, based on the relationship defined in (5),

$$[{}^{\mathcal{I}}\mathbf{v}_{\text{LS}/\mathcal{O}_m,k}]_{\mathcal{I}} = \mathbf{T}_a(\boldsymbol{\omega}_m\Delta t)[{}^{\mathcal{I}}\mathbf{v}_{\text{LS}/\mathcal{O}_m,k-1}]_{\mathcal{I}} \tag{35}$$

Equations (29)-(35) are clearly real-analytic. By vector addition,

$$[\mathbf{r}_{\text{LS}/\text{IMU},k}]_{\mathcal{I}} = [\mathbf{r}_{\text{LS}/\mathcal{O}_m,k}]_{\mathcal{I}} - [\mathbf{r}_{\text{IMU}/\mathcal{O}_m,k}]_{\mathcal{I}} \tag{36}$$

$$[{}^{\mathcal{I}}\mathbf{v}_{\text{LS}/\text{IMU},k}]_{\mathcal{I}} = [{}^{\mathcal{I}}\mathbf{v}_{\text{LS}/\mathcal{O}_m,k}]_{\mathcal{I}} - [{}^{\mathcal{I}}\mathbf{v}_{\text{IMU}/\mathcal{O}_m,k}]_{\mathcal{I}} \tag{37}$$

Therefore, relating back to the general process, (29) is equivalent to the $\mathbf{f}_r$ function (15), (30) to the $\mathbf{f}_v$ function (16), (34) to the $\mathbf{g}_r$ function (17), and (35) to the $\mathbf{g}_v$ function (18). Substituting (29) and (34) into (36), along with (30) and (35) into (37), yields

$$\begin{aligned}[\mathbf{r}_{\text{LS}/\text{IMU},k}]_{\mathcal{I}} =& \mathbf{T}_a(\boldsymbol{\omega}_m\Delta t)[\mathbf{r}_{\text{LS}/\mathcal{O}_m,k-1}]_{\mathcal{I}} - [\mathbf{r}_{\text{IMU}/\mathcal{O}_m,k-1}]_{\mathcal{I}} - [{}^{\mathcal{I}}\mathbf{v}_{\text{IMU}/\mathcal{O}_m,k-1}]_{\mathcal{I}}\Delta t \\ &- (\mathbf{T}^{\mathcal{B}}_{\mathcal{I},k-1})^T\Delta\mathbf{r}_{ng,k} - \frac{1}{2}\mathbf{g}_{k-1}\Delta t^2\end{aligned} \tag{38}$$

$$\begin{aligned}[{}^{\mathcal{I}}\mathbf{v}_{\text{LS}/\text{IMU},k}]_{\mathcal{I}} =& \mathbf{T}_a(\boldsymbol{\omega}_m\Delta t)[{}^{\mathcal{I}}\mathbf{v}_{\text{LS}/\mathcal{O}_m,k-1}]_{\mathcal{I}} - [{}^{\mathcal{I}}\mathbf{v}_{\text{IMU}/\mathcal{O}_m,k-1}]_{\mathcal{I}} - (\mathbf{T}^{\mathcal{B}}_{\mathcal{I},k-1})^T\Delta\mathbf{v}_{ng,k} \\ &- \mathbf{g}_{k-1}\Delta t\end{aligned} \tag{39}$$

Equations (38) and (39) are equivalent to (23) and (24). The IMU dead-reckoning equations already provide the desired kinematic expressions of position and velocity vectors. Thus, only the planetary rotation propagation needs to be modified. Rather than explicitly computing the Maclaurin series, it is simpler to recognize the identity [31]

$$\mathbf{T}_a(\mathbf{a}) = \exp([\mathbf{a}\times]) = \sum_{n=0}^{\infty}\frac{[\mathbf{a}\times]^n}{n!} \tag{40}$$

Therefore,

$$[\mathbf{r}_{\text{LS}/\mathcal{O}_m,k}]_{\mathcal{I}} = \sum_{n=0}^{\infty} \frac{[\boldsymbol{\omega_m}\times]^n}{n!}\Delta t^n [\mathbf{r}_{\text{LS}/\mathcal{O}_m,k-1}]_{\mathcal{I}} \tag{41}$$

$$= [\mathbf{r}_{\text{LS}/\mathcal{O}_m,k-1}]_{\mathcal{I}} + [\boldsymbol{\omega_m}\times][\mathbf{r}_{\text{LS}/\mathcal{O}_m,k-1}]_{\mathcal{I}}\Delta t + \sum_{n=2}^{\infty} \frac{[\boldsymbol{\omega_m}\times]^n}{n!}\Delta t^n [\mathbf{r}_{\text{LS}/\mathcal{O}_m,k-1}]_{\mathcal{I}} \tag{42}$$

$$= [\mathbf{r}_{\text{LS}/\mathcal{O}_m,k-1}]_{\mathcal{I}} + [{}^{\mathcal{I}}\mathbf{v}_{\text{LS}/\mathcal{O}_m,k-1}]_{\mathcal{I}}\Delta t \tag{43}$$

$$+ \Big(\mathbf{T}_a(\boldsymbol{\omega}_m\Delta t) - \mathbf{I}_{3\times 3} - [\boldsymbol{\omega}_m\times]\Delta t\Big)[\mathbf{r}_{\text{LS}/\mathcal{O}_m,k-1}]_{\mathcal{I}} \tag{44}$$

Substituting this result into (38), deriving and substituting a similar result for (39), and simplifying with the relationships in (36) and (37) yields the equivalent of (27) and (28):

$$[\mathbf{r}_{\text{LS}/\text{IMU},k}]_{\mathcal{I}} = [\mathbf{r}_{\text{LS}/\text{IMU},k-1}]_{\mathcal{I}} + [{}^{\mathcal{I}}\mathbf{v}_{\text{LS}/\text{IMU},k-1}]_{\mathcal{I}}\Delta t - (\mathbf{T}^{\mathcal{B}}_{\mathcal{I},k-1})^T\Delta\mathbf{r}_{ng,k} - \frac{1}{2}\mathbf{g}_{k-1}\Delta t^2 \tag{45}$$

$$+ \Big(\mathbf{T}_a(\boldsymbol{\omega}_m\Delta t) - \mathbf{I}_{3\times 3} - [\boldsymbol{\omega}_m\times]\Delta t\Big)[\mathbf{r}_{\text{LS}/\mathcal{O}_m,k-1}]_{\mathcal{I}}$$

$$[{}^{\mathcal{I}}\mathbf{v}_{\text{LS}/\text{IMU},k}]_{\mathcal{I}} = [{}^{\mathcal{I}}\mathbf{v}_{\text{LS}/\text{IMU},k-1}]_{\mathcal{I}} - (\mathbf{T}^{\mathcal{B}}_{\mathcal{I},k-1})^T\Delta\mathbf{v}_{ng,k} - \mathbf{g}_{k-1}\Delta t \tag{46}$$

$$+ \Big(\mathbf{T}_a(\boldsymbol{\omega}_m\Delta t) - \mathbf{I}_{3\times 3}\Big)[{}^{\mathcal{I}}\mathbf{v}_{\text{LS}/\mathcal{O}_m,k-1}]_{\mathcal{I}}$$

Thus, the only remaining step is to perform a transformation from the MCI frame to the robocentric frame, which is taken to be IMU frame in this application.

For this problem, only one reference trajectory is required since the landing site rotation is entirely determined by its MCI position vector and only global kinematic quantities, specifically the landing site and IMU MCI position vectors, are referenced. The landing site path is chosen as the reference trajectory due to its simplicity and exclusion of potentially noisy control inputs. This trajectory will be generated from the initial pose estimates at the DEM scan time (see (11)) via

$$[\bar{\mathbf{r}}_{\text{LS}/\mathcal{O}_m,k-1}]_{\mathcal{I}} = \mathbf{T}_a(\boldsymbol{\omega}_m\tau_d(k-1))\left([\mathbf{r}_{\text{IMU}/\mathcal{O}_m,d}]_{\mathcal{I}} + \mathbf{T}_p(\bar{\mathbf{q}}^{\mathcal{B}}_{\mathcal{I},d})^T[\mathbf{r}_{\text{LS}/\text{IMU},d}]_{\mathcal{B}}\right) \tag{47}$$

$$[\bar{\mathbf{r}}_{\text{IMU}/\mathcal{O}_m,k-1}]_{\mathcal{I}} = [\mathbf{r}_{\text{LS}/\mathcal{O}_m,k-1}]_{\mathcal{I}} - (\mathbf{T}^{\mathcal{B}}_{\mathcal{I},k-1})^T[\mathbf{r}_{\text{LS}/\text{IMU},k-1}]_{\mathcal{B}} \tag{48}$$

where $\tau_d(k-1)$ is the time elapsed from the DEM scan time t_d to time step $k-1$ and $\mathbf{T}^{\mathcal{B}}_{\mathcal{I},k-1} = \mathbf{T}_p(\bar{\mathbf{q}}^{\mathcal{B}}_{\mathcal{D},k-1}\otimes\bar{\mathbf{q}}^{\mathcal{D}}_{\mathcal{F}}\otimes\bar{\mathbf{q}}^{\mathcal{F}}_{\mathcal{I},k-1})$. The overbar notation here distinguishes that these values are produced by the reference trajectories. With these substitutions and appropriate coordinate transformations from $\mathcal{I}$ to $\mathcal{B}$, the final robocentric propagation equations for the position and velocity vectors are

$$[\mathbf{r}_{\text{LS}/\text{IMU},k}]_{\mathcal{B}} = \mathbf{T}_p(\Delta\boldsymbol{\theta}_{g,k})[\mathbf{r}_{\text{LS}/\text{IMU},k-1}]_{\mathcal{B}} + \mathbf{T}_p(\Delta\boldsymbol{\theta}_{g,k})[{}^{\mathcal{I}}\mathbf{v}_{\text{LS}/\text{IMU},k-1}]_{\mathcal{B}}\Delta t + \mathbf{T}^{\mathcal{B}}_{\mathcal{I},k}\mathbf{E}_{r,k} \tag{49}$$

$$- \mathbf{T}_p(\Delta\boldsymbol{\theta}_{g,k})\Delta\mathbf{r}_{ng,k} - \frac{1}{2}\mathbf{T}^{\mathcal{B}}_{\mathcal{I},k}\bar{\mathbf{g}}_{k-1}\Delta t^2$$

$$[{}^{\mathcal{I}}\mathbf{v}_{\text{LS}/\text{IMU},k}]_{\mathcal{B}} = \mathbf{T}_p(\Delta\boldsymbol{\theta}_{g,k})[{}^{\mathcal{I}}\mathbf{v}_{\text{LS}/\text{IMU},k-1}]_{\mathcal{B}} + \mathbf{T}^{\mathcal{B}}_{\mathcal{I},k}\mathbf{E}_{v,k} - \mathbf{T}_p(\Delta\boldsymbol{\theta}_{g,k})\Delta\mathbf{v}_{ng,k} \tag{50}$$

$$- \mathbf{T}^{\mathcal{B}}_{\mathcal{I},k}\bar{\mathbf{g}}_{k-1}\Delta t$$

where $\mathbf{T}^{\mathcal{B}}_{\mathcal{I},k} = \mathbf{T}_p(\Delta\boldsymbol{\theta}_{g,k})\mathbf{T}^{\mathcal{B}}_{\mathcal{I},k-1}$, $\mathbf{E}_{r,k} = \mathbf{M}_r[\bar{\mathbf{r}}_{\text{LS}/\mathcal{O}_m,k-1}]_{\mathcal{I}}$, $\mathbf{M}_r = \mathbf{T}_p(\boldsymbol{\omega}_m\Delta t) - \mathbf{I}_{3\times 3} - [\boldsymbol{\omega}_m\times]\Delta t$, $\mathbf{E}_{v,k} = \mathbf{M}_v[\boldsymbol{\omega}_m\times][\bar{\mathbf{r}}_{\text{LS}/\mathcal{O}_m,k-1}]_{\mathcal{I}}$, and $\mathbf{M}_v = \mathbf{T}_p(\boldsymbol{\omega}_m\Delta t) - \mathbf{I}_{3\times 3}$. An overbar is now put on the gravitational acceleration to indicate it is computed from $[\bar{\mathbf{r}}_{\text{IMU}/\mathcal{O}_m,k-1}]_{\mathcal{I}}$.

Deriving new attitude propagation equations is a much simpler process. Recalling (31), it is simple to perform a change of reference frames and recognize

$$\bar{\mathbf{q}}^{\mathcal{B}}_{\mathcal{D},k} = \bar{\mathbf{q}}(\Delta\boldsymbol{\theta}_{g,k}) \otimes \bar{\mathbf{q}}^{\mathcal{B}}_{\mathcal{D},k-1} \otimes \bar{\mathbf{q}}^{\mathcal{D}}_{\mathcal{F}} \otimes \bar{\mathbf{q}}^{\mathcal{F}}_{\mathcal{I},k-1} \otimes (\bar{\mathbf{q}}^{\mathcal{F}}_{\mathcal{I},k})^{-1} \otimes (\bar{\mathbf{q}}^{\mathcal{D}}_{\mathcal{F}})^{-1} \tag{51}$$

The camera bias nominally remains the same between time steps, although it is subject to process noise. The global DEM time pose estimates remain the same, as well as the map states. Therefore, (49)-(51) constitute the robocentric propagation equations.

Error propagation is performed with the Jacobian of these robocentric propagation equations as

$$\mathbf{e}_k = \mathbf{F}\mathbf{e}_{k-1} = \begin{bmatrix} \mathbf{F}_s & \mathbf{0}_{20\times n_m} \\ \mathbf{0}_{n_m\times 20} & \mathbf{I}_{n_m\times n_m} \end{bmatrix} \mathbf{e}_{k-1} \tag{52}$$

where n_m is the number of map states and $\mathbf{F}_s$ is defined in appendix A. It is important to note that while most state errors are additive, all attitude errors are multiplicative. Specifically, the attitude error is expressed as a 3-dimensional Euler vector, meaning the covariance matrix has a smaller dimension that the state vector. For more information on this topic, see [31].

Square-Root EKF-SLAM

Using a square-root formulation will improve the numerical stability of the filter by abating floating point arithmetic errors. The square-root formulation operates on a Cholesky factor of the full covariance matrix, referred to as a square-root factor. While the results are theoretically equivalent between a full-covariance filter and a square-root filter, this is not the case in practice. Previous simulations for this paper showed an indefinite covariance matrix forming in the full covariance approach that eventually caused the state estimates to diverge. Alternatively, a UD-factorized approach could also be useful here. For more information on implementation of these methods, see [32, 33]. Implementing the square-root EKF-SLAM algorithm is very similar to a normal square-root EKF (a thorough derivation of this fact for the full-covariance approach is given in [16]). Because individual map states are not propagated and have no process noise, they are similar to parameters in an EKF-based fixed point smoother algorithm.

Map State Construction and Maintenance

Landmarks are selected at random to populate the map state as they are detected by the camera. Once the map state is full, additional landmark measurements are discarded–this is necessary to reduce computational costs. Replacing a landmark means replacing its values in both the state vector and square-root covariance factor. Replacement in the state vector just requires switching out the DEM coordinates of one landmark for another. Replacement in the square-root covariance factor is less straightforward, as the landmark must be marginalized out.

In the full-covariance matrix approach, replacement can be accomplished by zeroing out the rows/columns corresponding to the old landmark and replacing the main diagonal entries with uncorrelated initial variances of the new landmark. To replicate this process for a square-root covariance factor, without loss of generality assume a lower-triangular square root factor is being used. First, zero out only the rows of the square-root factor corresponding to the old landmark. Then, perform a rank-1 Cholesky update of the square-root factor with a column vector containing the initial standard deviations of the new landmark in the appropriate indices. Analytically, this produces the exact same result as the full-covariance approach; however, small numerical errors may arise in performing the Cholesky update. Though, with a small landmark map and short scenario, this error will be trivial.

RESULTS

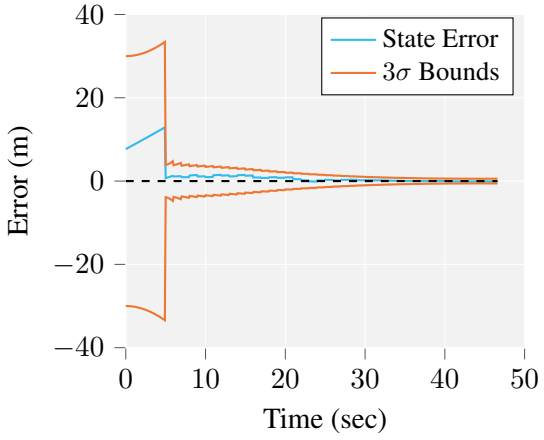
The proposed algorithm is tested on the final approach of a lunar south pole descent and landing scenario. In this simulation, 100 landmarks are randomly distributed according to a uniform distribution in a 100 m by 100 m square centered at the landing site on the Moon's surface, representing the DEM. Though a spherical Moon is assumed, landmark altitude may vary ± 50 m from the spherical lunar surface. The map state has five landmarks, and camera measurements arrive at a 1 Hz frequency beginning 5 seconds after the DEM is produced. Landmark positions within the DEM are assumed to be known to 1σ within 30 cm along-track/cross-track and 1 m elevation. Additionally, perfect data association and feature extraction is assumed. Alternative methods exist to improve the robustness of TRN, and thereby this approach, to the nonstandard errors from optical measurements (including flawed data association) [34]; however, at the moment such techniques will not be implemented. The camera has an angular noise standard deviation of 0.058594° (equivalent to 1 pixel) and is co-boresighted with the lidar scanner. IMU measurements are corrupted according to the specifications of the LN-200S [35], and truth values propagate with the LP165P gravity model with degree and order 165 [36]. The associated process noise covariance matrix for the IMU corruption is derived in appendix B. The filter will assume two-body gravity to investigate the effect of model mismatch.

Initial values for states and their associated variances are presented in Table 1. If a filter begins with global states but later on switches to relative states, appropriate covariance matrix modification should occur (example provided in [26]).

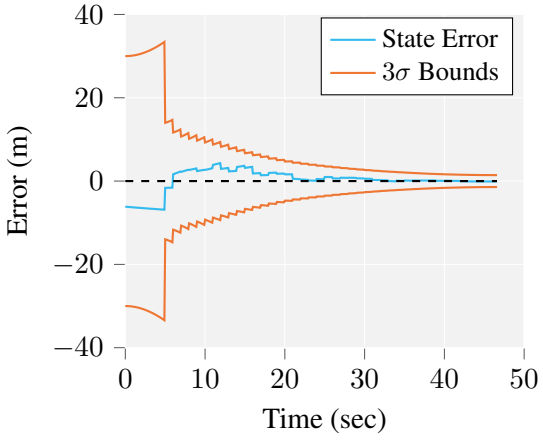
Table 1: Initial state and covariance values.

	$[\mathbf{r}_{\text{LS/IMU},d}]_{\mathcal{B}}$	$[{}^{\mathcal{I}}\mathbf{v}_{\text{LS/IMU},d}]_{\mathcal{B}}$	$\mathbf{b}_{\text{TC}}$	$\bar{\mathbf{q}}^{\mathcal{B}}_{\mathcal{D},d}$	$[\mathbf{r}_{\text{IMU}/\mathcal{O}_m,d}]_{\mathcal{I}}$	$\bar{\mathbf{q}}^{\mathcal{B}}_{\mathcal{I},d}$
Initial Value	$\begin{bmatrix}116.36\\600.87\\388.87\end{bmatrix}$ (m)	$\begin{bmatrix}-1.0530\\-16.633\\-13.258\end{bmatrix}$ (m/sec)	$\begin{bmatrix}-1\\0.5\end{bmatrix}$ (pixels)	$\begin{bmatrix}0.47196\\0.55575\\-0.48772\\-0.48013\end{bmatrix}$ (unitless)	$\begin{bmatrix}42073\\676006\\-1600589\end{bmatrix}$ (m)	$\begin{bmatrix}0.51603\\0.85628\\-0.02206\\0.00182\end{bmatrix}$ (unitless)
Initial Variance	100 $(\text{m})^2$	1 $(\text{m/sec})^2$	1 $(\text{pixels})^2$	$1.3539 \cdot 10^{-4}$ $(\text{rad})^2$	100 $(\text{m})^2$	$1.3539 \cdot 10^{-4}$ $(\text{rad})^2$

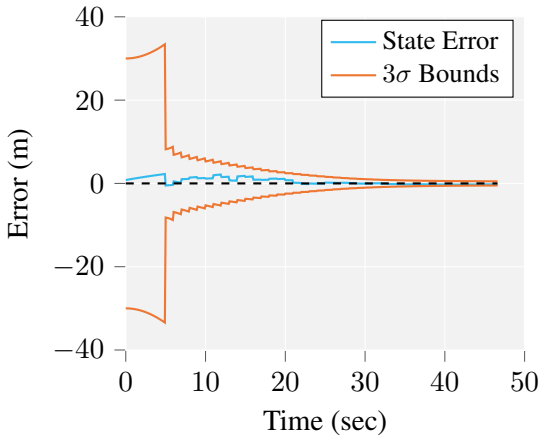
The initial estimated state vector estimate is sampled from a normal distribution with mean $\mathbf{x}_0$ and covariance $\mathbf{P}_0$, using the values in Table 1. Initial results from this simulation are presented as estimate error and 3σ covariance bound plots in Figures 3 and 4.



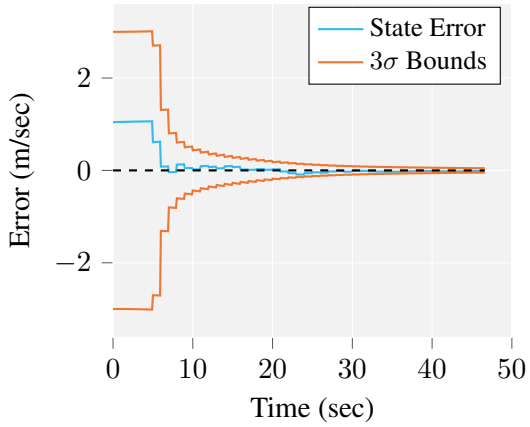
(a) Relative position vector x-component.



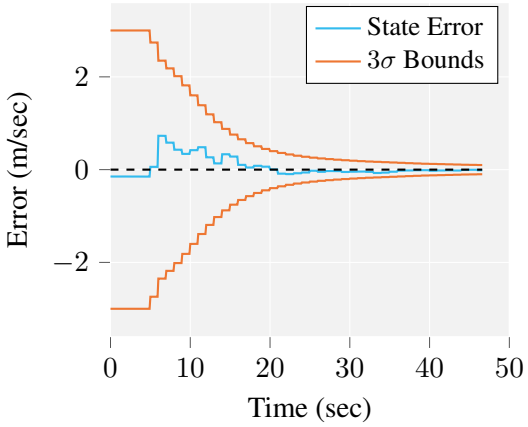
(b) Relative position vector y-component.



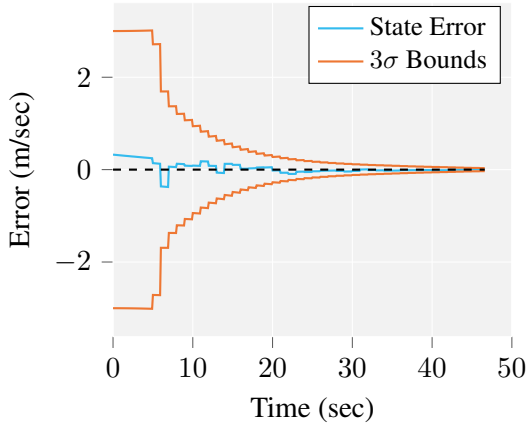
(c) Relative position vector z-component.



(d) Relative velocity vector x-component.

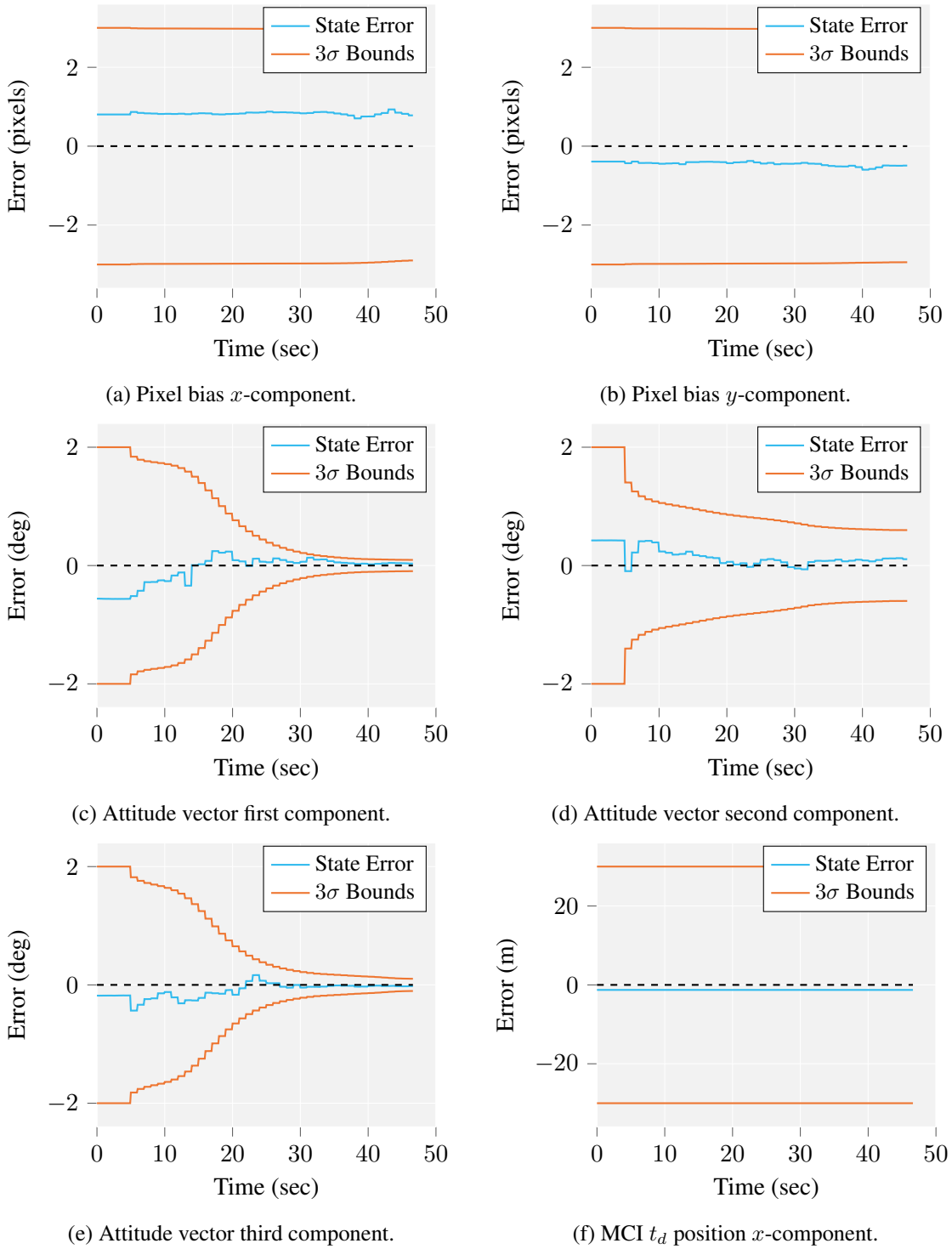


(e) Relative velocity vector y-component.



(f) Relative velocity vector z-component.

Figure 3: Relative position and velocity error plots with 3σ bounds.



(a) Pixel bias x-component.

(b) Pixel bias y-component.

(c) Attitude vector first component.

(d) Attitude vector second component.

(e) Attitude vector third component.

(f) MCI t_d position x-component.

Figure 4: Bias, attitude, and sample DEM scan time state error plots with 3σ bounds.

All state estimation errors besides pixel bias and the MCI position vector x-component of the IMU at the DEM scan time show significant convergence and reduction of error. The position

error plots are especially encouraging for aiding safe and precise landings, achieving final error of roughly 0.5 m 3σ for the x- and z-components and 1.5 m 3σ for the y-component. Throughout the plots, it seems that there is slower convergence for y-components of position and velocity and the 2nd attitude error. The y-direction in this simulation roughly corresponds to the IMU MCI radial direction; that is, it points roughly straight down at the lunar surface. While the DEM provides scale to the image, permitting estimation of altitude, the y-component is less observable than the x- and z-directions because small changes in the altitude produce negligible changes in the image. That is not to say it is unobservable, but rather that the convergence is expected to be slower. The pixel biases show minimal convergence, which is to be expected for a short scenario duration with minimal attitude motion. Pixel bias error is difficult to differentiate from small attitude or position errors, so without a more accurate initial map that can aid in isolating pixel biases from these other effects, minimal information will be gained. However, this does not significantly affect overall performance.

The MCI scan position x-component shown in Figure 4f, along with all of the other global pose estimates at the DEM scan time, do not undergo any updates because as previously mentioned, there is no information about these terms from the terrain camera. Because there is no information provided by the update, as well as no process noise, they are static values. Their inclusion in this manner is almost similar to that of a consider Kalman filter. However, if additional sensors are incorporated, it is possible that these state estimates could improve.

CONCLUSION

This paper proposes a robocentric visual square-root EKF-SLAM algorithm for navigating through a lunar descent and landing scenario without prior-mapped landmarks. The generic process for deriving robocentric equations of motion permits extension of robocentric SLAM to problems with moving landmarks, yielding significant benefits to both observability and consistency. Simulation results demonstrate these benefits with significant reduction in navigation uncertainty while only tracking five landmarks, maintaining low computational complexity. While this algorithm is designed for landing on the lunar surface, it is applicable to a wide variety of problems beyond unmapped celestial bodies, ranging from navigating the ocean floor, to spacecraft rendezvous and landing unmanned aircraft. Future work on this topic may include enhanced in-flight landmark selection and replacement algorithms and validation through real-world experiments.

APPENDIX A: MEASUREMENT AND ERROR PROPAGATION JACOBIANS

The Jacobian equations in this appendix will utilize estimated quantities, as these Jacobians are often computed for EKF implementation. The terrain camera measurement Jacobian for the robocentric state vector and a specified landmark L_j is

$$\mathbf{H}_{L_j,k} = \begin{bmatrix} \mathbf{\Lambda}(\mathbf{T}^{\mathcal{C}}_{\mathcal{B}}, [\mathbf{r}_{L_j/\text{TC},k}]_{\mathcal{C}})^T \\ \mathbf{0}^T_{2\times 3} \\ \mathbf{I}^T_{2\times 2} \\ \mathbf{\Lambda}(\mathbf{T}^{\mathcal{C}}_{\mathcal{B}}[\mathbf{T}_p(\hat{\bar{\mathbf{q}}}^{\mathcal{B}}_{\mathcal{D},k})[\mathbf{r}_{L_j/\text{LS}}]_{\mathcal{D}}\times], [\mathbf{r}_{L_j/\text{TC},k}]_{\mathcal{C}})^T \\ \mathbf{0}^T_{2\times\alpha} \\ \mathbf{\Lambda}(\mathbf{T}^{\mathcal{C}}_{\mathcal{B}}\mathbf{T}_p(\hat{\bar{\mathbf{q}}}^{\mathcal{B}}_{\mathcal{D},k}), [\mathbf{r}_{L_j/\text{TC},k}]_{\mathcal{C}})^T \\ \mathbf{0}^T_{2\times\beta} \end{bmatrix}^T \tag{53}$$

where

$$\mathbf{\Lambda}(\mathbf{M}, [\mathbf{r}_{L_j/\text{TC}}]_{\mathcal{C}}) = \begin{bmatrix} \dfrac{\text{f} \cdot (\mathbf{M}_{(\text{row } 1)}\hat{z}_{L_j/\text{TC}} - \mathbf{M}_{(\text{row } 3)}\hat{x}_{L_j/\text{TC}})}{(\hat{z}_{L_j/\text{TC}})^2} \\ \dfrac{\text{f} \cdot (\mathbf{M}_{(\text{row } 2)}\hat{z}_{L_j/\text{TC}} - \mathbf{M}_{(\text{row } 3)}\hat{y}_{L_j/\text{TC}})}{(\hat{z}_{L_j/\text{TC}})^2} \end{bmatrix} \tag{54}$$

and α, β are appropriate positive integers so that the final nonzero term in $\mathbf{H}_{L_j,k}$ corresponds to L_j's map state. Here, $\mathbf{\Lambda}$ operates on the rows of a matrix $\mathbf{M}$ and other values are consistent with the definitions in (6) and (7).

The spacecraft states error propagation matrix $\mathbf{F}_s$ is defined

$$\mathbf{F}_s = \begin{bmatrix} \mathbf{F}_{r,r} & \mathbf{T}_p(\Delta\hat{\boldsymbol{\theta}}_{g,k})\Delta t & \mathbf{0}_{3\times2} & \mathbf{F}_{r,q} & \mathbf{F}_{r,r_{imu,d}} & \mathbf{F}_{r,q_d} & \mathbf{F}_{r,r_{rel,d}} \\ \mathbf{F}_{v,r} & \mathbf{T}_p(\Delta\hat{\boldsymbol{\theta}}_{g,k}) & \mathbf{0}_{3\times2} & \mathbf{F}_{v,q} & \mathbf{F}_{v,r_{imu,d}} & \mathbf{F}_{v,q_d} & \mathbf{F}_{v,r_{rel,d}} \\ \mathbf{0}_{2\times3} & \mathbf{0}_{2\times3} & \mathbf{I}_{2\times2} & \mathbf{0}_{2\times3} & \mathbf{0}_{2\times3} & \mathbf{0}_{2\times3} & \mathbf{0}_{2\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times2} & \mathbf{T}_p(\Delta\hat{\boldsymbol{\theta}}_{g,k}) & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times2} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times2} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times2} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} \end{bmatrix} \tag{55}$$

where

$$\mathbf{F}_{r,r} = \mathbf{T}_p(\Delta\hat{\boldsymbol{\theta}}_{g,k}) + \frac{1}{2}\hat{\mathbf{T}}^{\mathcal{B}}_{\mathcal{I},k}\hat{\mathbf{G}}_{k-1}(\hat{\mathbf{T}}^{\mathcal{B}}_{\mathcal{I},k-1})^T\Delta t^2 \tag{56}$$

$$\begin{aligned}\mathbf{F}_{r,q} =& \mathbf{T}_p(\Delta\hat{\boldsymbol{\theta}}_{g,k})[\hat{\mathbf{T}}^{\mathcal{B}}_{\mathcal{I},k-1}\hat{\mathbf{E}}_{r,k}(n)\times] - \frac{1}{2}\hat{\mathbf{T}}^{\mathcal{B}}_{\mathcal{I},k}\hat{\mathbf{G}}_{k-1}(\hat{\mathbf{T}}^{\mathcal{B}}_{\mathcal{I},k-1})^T[[\hat{\mathbf{r}}_{\text{LS/IMU},k-1}]_{\mathcal{B}}\times]\Delta t^2 \\ & - \frac{1}{2}\mathbf{T}_p(\Delta\hat{\boldsymbol{\theta}}_{g,k})[\hat{\mathbf{T}}^{\mathcal{B}}_{\mathcal{I},k-1}\hat{\bar{\mathbf{g}}}_{k-1}\times]\Delta t^2\end{aligned} \tag{57}$$

$$\mathbf{F}_{r,r_{imu,d}} = \hat{\mathbf{T}}^{\mathcal{B}}_{\mathcal{I},k}\mathbf{M}_r\mathbf{T}_a(\boldsymbol{\omega}_m\tau_d(k-1)) - \frac{1}{2}\hat{\mathbf{T}}^{\mathcal{B}}_{\mathcal{I},k}\hat{\mathbf{G}}_{k-1}\mathbf{T}_a(\boldsymbol{\omega}_m\tau_d(k-1))\Delta t^2 \tag{58}$$

$$\begin{aligned}\mathbf{F}_{r,q_d} =& -\hat{\mathbf{T}}^{\mathcal{B}}_{\mathcal{I},k}\mathbf{M}_r\mathbf{T}_a(\boldsymbol{\omega}_m\tau_d(k-1))\mathbf{T}_p(\hat{\bar{\mathbf{q}}}^{\mathcal{B}}_{\mathcal{I},d})^T[[\hat{\mathbf{r}}_{\text{LS/IMU},d}]_{\mathcal{B}}\times] \\ & + \frac{1}{2}\hat{\mathbf{T}}^{\mathcal{B}}_{\mathcal{I},k}\hat{\mathbf{G}}_{k-1}\mathbf{T}_a(\boldsymbol{\omega}_m\tau_d(k-1))\mathbf{T}_p(\hat{\bar{\mathbf{q}}}^{\mathcal{B}}_{\mathcal{I},d})^T[[\hat{\mathbf{r}}_{\text{LS/IMU},d}]_{\mathcal{B}}\times]\Delta t^2\end{aligned} \tag{59}$$

$$\begin{aligned}\mathbf{F}_{r,r_{rel,d}} =& \hat{\mathbf{T}}^{\mathcal{B}}_{\mathcal{I},k}\mathbf{M}_r\mathbf{T}_a(\boldsymbol{\omega}_m\tau_d(k-1))\mathbf{T}_p(\hat{\bar{\mathbf{q}}}^{\mathcal{B}}_{\mathcal{I},d})^T \\ & - \frac{1}{2}\hat{\mathbf{T}}^{\mathcal{B}}_{\mathcal{I},k}\hat{\mathbf{G}}_{k-1}\mathbf{T}_a(\boldsymbol{\omega}_m\tau_d(k-1))\mathbf{T}_p(\hat{\bar{\mathbf{q}}}^{\mathcal{B}}_{\mathcal{I},d})^T\Delta t^2\end{aligned} \tag{60}$$

$$\mathbf{F}_{v,r} = \hat{\mathbf{T}}^{\mathcal{B}}_{\mathcal{I},k}\hat{\mathbf{G}}_{k-1}(\hat{\mathbf{T}}^{\mathcal{B}}_{\mathcal{I},k-1})^T\Delta t \tag{61}$$

$$\begin{aligned}\mathbf{F}_{v,q} =& \mathbf{T}_p(\Delta\hat{\boldsymbol{\theta}}_{g,k})[\hat{\mathbf{T}}^{\mathcal{B}}_{\mathcal{I},k-1}\hat{\mathbf{E}}_{v,k}(n)\times] - \hat{\mathbf{T}}^{\mathcal{B}}_{\mathcal{I},k}\hat{\mathbf{G}}_{k-1}(\hat{\mathbf{T}}^{\mathcal{B}}_{\mathcal{I},k-1})^T[[\hat{\mathbf{r}}_{\text{LS/IMU},k-1}]_{\mathcal{B}}\times]\Delta t \\ & - \mathbf{T}_p(\Delta\hat{\boldsymbol{\theta}}_{g,k})[\hat{\mathbf{T}}^{\mathcal{B}}_{\mathcal{I},k-1}\hat{\bar{\mathbf{g}}}_{k-1}\times]\Delta t\end{aligned} \tag{62}$$

$$\mathbf{F}_{v,r_{imu,d}} = \hat{\mathbf{T}}^{\mathcal{B}}_{\mathcal{I},k}\mathbf{M}_v\mathbf{T}_a(\boldsymbol{\omega}_m\tau_d(k-1))[\boldsymbol{\omega}_m\times] - \hat{\mathbf{T}}^{\mathcal{B}}_{\mathcal{I},k}\hat{\mathbf{G}}_{k-1}\mathbf{T}_a(\boldsymbol{\omega}_m\tau_d(k-1))\Delta t \tag{63}$$

$$\begin{aligned}\mathbf{F}_{r,q_d} =& -\hat{\mathbf{T}}^{\mathcal{B}}_{\mathcal{I},k}\mathbf{M}_v\mathbf{T}_a(\boldsymbol{\omega}_m\tau_d(k-1))[\boldsymbol{\omega}_m\times]\mathbf{T}_p(\hat{\bar{\mathbf{q}}}^{\mathcal{B}}_{\mathcal{I},d})^T[[\hat{\mathbf{r}}_{\text{LS/IMU},d}]_{\mathcal{B}}\times] \\ & + \hat{\mathbf{T}}^{\mathcal{B}}_{\mathcal{I},k}\hat{\mathbf{G}}_{k-1}\mathbf{T}_a(\boldsymbol{\omega}_m\tau_d(k-1))\mathbf{T}_p(\hat{\bar{\mathbf{q}}}^{\mathcal{B}}_{\mathcal{I},d})^T[[\hat{\mathbf{r}}_{\text{LS/IMU},d}]_{\mathcal{B}}\times]\Delta t\end{aligned} \tag{64}$$

$$\begin{aligned}\mathbf{F}_{v,r_{rel,d}} =& \hat{\mathbf{T}}^{\mathcal{B}}_{\mathcal{I},k}\mathbf{M}_v\mathbf{T}_a(\boldsymbol{\omega}_m\tau_d(k-1))[\boldsymbol{\omega}_m\times]\mathbf{T}_p(\hat{\bar{\mathbf{q}}}^{\mathcal{B}}_{\mathcal{I},d})^T \\ & - \hat{\mathbf{T}}^{\mathcal{B}}_{\mathcal{I},k}\hat{\mathbf{G}}_{k-1}\mathbf{T}_a(\boldsymbol{\omega}_m\tau_d(k-1))\mathbf{T}_p(\hat{\bar{\mathbf{q}}}^{\mathcal{B}}_{\mathcal{I},d})^T\Delta t\end{aligned} \tag{65}$$

Here, $\hat{\mathbf{G}}_{k-1}$ is the Jacobian matrix of the selected gravity model evaluated at the estimated MCI IMU position vector. All of these values share the same definitions as those for the state propagation equations.

APPENDIX B: PROCESS NOISE COVARIANCE

IMU noise is typically expressed as an angular random walk for the gyroscope and power spectral density for the accelerometer. There are two steps to derive the process noise covariance: compute the discrete-time IMU measurement noise covariance, and map to states. Computing the discrete-time IMU measurement noise covariance requires unit conversion. For the LN-200S specifications with 0.01 sec between measurements (100 Hz sampling rate) [35],

$$\text{Gyroscope: } \left(0.07\frac{\text{deg}}{\sqrt{\text{hr}}}\cdot\frac{\pi \text{ rad}}{180 \text{ deg}}\cdot\frac{\sqrt{\text{hr}}}{60\sqrt{\text{sec}}}\cdot\right)^2\cdot 0.01 \text{ sec} = 4.1462\cdot 10^{-12} \text{ (rad)}^2 \tag{66}$$

$$\text{Accelerometer: } \left(35\cdot 10^{-6}\frac{\text{g}}{\sqrt{\text{Hz}}}\cdot\frac{9.80 \text{ m/sec}^2}{\text{g}}\right)^2\cdot 0.01 \text{ sec} = 1.1765\cdot 10^{-9} \text{ (m/sec)}^2 \tag{67}$$

The gyroscope and accelerometer measurements are 3-dimensional vectors, therefore these variances apply to each component of those vectors. Assuming uncorrelated measurement noise, this means that the bias-corrected IMU measurement vector

$$\hat{\mathbf{z}}_{\text{IMU}} = \begin{bmatrix}\Delta\hat{\boldsymbol{\theta}}_{g,k} & \Delta\hat{\mathbf{v}}_{a,k}\end{bmatrix}^T \tag{68}$$

has covariance matrix

$$\mathbf{R}_{\text{IMU}} = \begin{bmatrix}(4.1462\cdot 10^{-12})\mathbf{I}_{3\times 3} & \mathbf{0}_{3\times 3}\\ \mathbf{0}_{3\times 3} & (1.1765\cdot 10^{-9})\mathbf{I}_{3\times 3}\end{bmatrix} \tag{69}$$

The IMU measurement noise to process noise mapping matrix is computed via linearization of the robocentric propagation equations. The mapping matrix is

$$\boldsymbol{\Pi} = \begin{bmatrix}\boldsymbol{\Pi}_{r,g} & \boldsymbol{\Pi}_{r,a}\\ \boldsymbol{\Pi}_{v,g} & \boldsymbol{\Pi}_{v,a}\\ \mathbf{0}_{2\times 3} & \mathbf{0}_{2\times 3}\\ \mathbf{I}_{3\times 3} & \mathbf{0}_{3\times 3}\\ \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3}\\ \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3}\\ \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3}\end{bmatrix} \tag{70}$$

where

$$\begin{aligned}\boldsymbol{\Pi}_{r,g} =&[\mathbf{T}_p(\Delta\hat{\boldsymbol{\theta}}_{g,k})[\hat{\mathbf{r}}_{\text{LS/IMU},k-1}]_{\mathcal{B}}\times] + [\mathbf{T}_p(\Delta\hat{\boldsymbol{\theta}}_{g,k})\hat{\mathbf{v}}^{\mathcal{B}}_{\text{rel},k-1}\times]\Delta t\\ &+ [\mathbf{T}_p(\Delta\hat{\boldsymbol{\theta}}_{g,k})\hat{\mathbf{T}}^{\mathcal{B}}_{\mathcal{I},k-1}\hat{\mathbf{E}}_{r,k}(n)\times] + \frac{1}{6}\mathbf{T}_p(\Delta\hat{\boldsymbol{\theta}}_{g,k})[\Delta\hat{\mathbf{v}}_{a,k}\times]\Delta t\\ &- [\mathbf{T}_p(\Delta\hat{\boldsymbol{\theta}}_{g,k})\Delta\hat{\mathbf{r}}_{ng,k}\times] - \frac{1}{2}[\mathbf{T}_p(\Delta\hat{\boldsymbol{\theta}}_{g,k})\hat{\mathbf{T}}^{\mathcal{B}}_{\mathcal{I},k-1}\hat{\mathbf{g}}_{k-1}\times]\Delta t^2\end{aligned} \tag{71}$$

$$\boldsymbol{\Pi}_{r,a} = -\frac{1}{2}\mathbf{T}_p(\Delta\hat{\boldsymbol{\theta}}_{g,k})\Delta t - \frac{1}{6}\mathbf{T}_p(\Delta\hat{\boldsymbol{\theta}}_{g,k})[\Delta\hat{\boldsymbol{\theta}}_{g,k}\times]\Delta t \tag{72}$$

$$\boldsymbol{\Pi}_{v,g} = [\mathbf{T}_p(\Delta\hat{\boldsymbol{\theta}}_{g,k})\hat{\mathbf{v}}^{\mathcal{B}}_{\text{rel},k-1}\times] + [\mathbf{T}_p(\Delta\hat{\boldsymbol{\theta}}_{g,k})\hat{\mathbf{T}}^{\mathcal{B}}_{\mathcal{I},k-1}\hat{\mathbf{E}}_{v,k}(n)\times] + \frac{1}{2}\mathbf{T}_p(\Delta\hat{\boldsymbol{\theta}}_{g,k})[\Delta\hat{\mathbf{v}}_{a,k}\times] \quad (73)$$
$$- [\mathbf{T}_p(\Delta\hat{\boldsymbol{\theta}}_{g,k})\Delta\hat{\mathbf{v}}_{ng,k}\times] - [\mathbf{T}_p(\Delta\hat{\boldsymbol{\theta}}_{g,k})\hat{\mathbf{T}}^{\mathcal{B}}_{\mathcal{I},k-1}\hat{\mathbf{g}}_{k-1}\times]\Delta t$$
$$\boldsymbol{\Pi}_{v,a} = -\mathbf{T}_p(\Delta\hat{\boldsymbol{\theta}}_{g,k}) - \frac{1}{2}\mathbf{T}_p(\Delta\hat{\boldsymbol{\theta}}_{g,k})[\Delta\hat{\boldsymbol{\theta}}_{g,k}\times] \quad (74)$$

Thus, the process noise changes with the state and IMU measurement values–a consequence of the robocentric formulation. Now, the process noise covariance matrix can be computed as $\mathbf{Q} = \boldsymbol{\Pi}\mathbf{R}_{\text{IMU}}\boldsymbol{\Pi}^T$. For the square-root formulation, given a lower-triangular Cholesky factor of $\mathbf{R}_{\text{IMU}}$ denoted $\mathbf{S}_{\text{IMU}}$, the square-root factor of $\mathbf{Q}$ is $\boldsymbol{\Pi}\mathbf{S}_{\text{IMU}}$

REFERENCES

[1] J. Schmitz, M. Komorowski, T. Russomano, O. Ullrich, and J. Hinkelbein, "Sixty Years of Manned Spaceflight—Incidents and Accidents Involving Astronauts between Launch and Landing," *Aerospace*, Vol. 9, Nov. 2022, p. 675, 10.3390/aerospace9110675.

[2] "Safe and Precise Landing Integrated Capabilities Evolution (SPLICE)," `https://ntrs.nasa.gov/citations/20230015493`, Dec. 2023.

[3] R. R. Sostaric, S. Pedrotty, J. M. Carson, J. N. Estes, F. Amzajerdian, A. M. Dwyer-Cianciolo, and J. B. Blair, "The SPLICE Project: Safe and Precise Landing Technology Development and Testing," *AIAA Scitech 2021 Forum*, VIRTUAL EVENT, American Institute of Aeronautics and Astronautics, Jan. 2021, 10.2514/6.2021-0256.

[4] "SLIM Project Press Kit," `https://global.jaxa.jp/countdown/slim/SLIM-mediakit-EN_2310.pdf`.

[5] G. Molina, M. Hansen, J. Getchius, R. Christensen, J. A. Christian, S. Stewart, and T. Crain, "VISUAL ODOMETRY FOR PRECISION LUNAR LANDING," *Proceedings of the 44th Annual American Astronautical Society Guidance, Navigation, and Control Conference, 2022* (M. Sandnas and D. B. Spencer, eds.), Vol. 179, pp. 1021–1042, Cham: Springer International Publishing, 2024, 10.1007/978-3-031-51928-4_58.

[6] C. Owens, K. Macdonald, J. Hardy, R. Lindsay, M. Redfield, M. Bloom, E. Bailey, Y. Cheng, D. Clouse, C. Y. Villalpando, A. Hambardzumyan, A. E. Johnson, and A. D. Horchler, "Development of a Signature-based Terrain Relative Navigation System for Precision Landing," *AIAA Scitech 2021 Forum*, VIRTUAL EVENT, American Institute of Aeronautics and Astronautics, Jan. 2021, 10.2514/6.2021-0376.

[7] M. S, K. Negi, and G. Vaibhav, "ISRO's Unprecedented Journey to the Moon," *Acta Astronautica*, Vol. 177, Dec. 2020, pp. 286–298, 10.1016/j.actaastro.2020.07.046.

[8] H. Zhang, J. Li, Z. Wang, and Y. Guan, "Guidance Navigation and Control for Chang'E-5 Powered Descent," *Space: Science & Technology*, Vol. 2021, Jan. 2021, p. 2021/9823609, 10.34133/2021/9823609.

[9] A. E. Johnson, Y. Cheng, N. Trawny, J. F. Montgomery, S. Schroeder, J. Chang, D. Clouse, S. Aaron, and S. Mohan, "Implementation of a Map Relative Localization System for Planetary Landing," *Journal of Guidance, Control, and Dynamics*, Vol. 46, Apr. 2023, pp. 618–637, 10.2514/1.G006780.

[10] L. Matthies, S. Daftry, B. Rothrock, A. Davis, R. Hewitt, E. Sklyanskiy, J. Delaune, A. Schutte, M. Quadrelli, M. Malaska, and J. Yurtsever, "Terrain Relative Navigation for Guided Descent on Titan," *2020 IEEE Aerospace Conference*, Big Sky, MT, USA, IEEE, Mar. 2020, pp. 1–16, 10.1109/AERO47225.2020.9172286.

[11] D. A. Lorenz, R. Olds, A. May, C. Mario, M. E. Perry, E. E. Palmer, and M. Daly, "Lessons Learned from OSIRIS-REx Autonomous Navigation Using Natural Feature Tracking," *2017 IEEE Aerospace Conference*, Big Sky, MT, USA, IEEE, Mar. 2017, pp. 1–12, 10.1109/AERO.2017.7943684.

[12] D. K. Meduna, S. M. Rock, and R. McEwen, "Low-Cost Terrain Relative Navigation for Long-Range AUVs," *OCEANS 2008*, Quebec City, QC, Canada, IEEE, 2008, pp. 1–7, 10.1109/OCEANS.2008.5152043.

[13] D. Di Massa and W. Stewart, "Terrain-Relative Navigation for Autonomous Underwater Vehicles," *Oceans '97. MTS/IEEE Conference Proceedings*, Vol. 1, Halifax, NS, Canada, IEEE, 1997, pp. 541–546, 10.1109/OCEANS.1997.634423.

[14] D. S. Bayard, D. T. Conway, R. Brockers, J. H. Delaune, L. H. Matthies, H. F. Grip, G. B. Merewether, T. L. Brown, and A. M. San Martin, "Vision-Based Navigation for the NASA Mars Helicopter," *AIAA Scitech 2019 Forum*, San Diego, California, American Institute of Aeronautics and Astronautics, Jan. 2019, 10.2514/6.2019-1411.

[15] C. D. Adam, L. K. McCarthy, J. M. Leonard, R. Gaskell, P. G. Antreasian, A. J. Liounis, K. Getzandanner, M. C. Moreau, E. E. Palmer, J. Weirich, O. S. Barnouin, J. L. Geeraert, E. M. Sahr, B. Ashman, D. S. Nelson, J. Y. Pelgrift, E. J. Lessac-Chenen, D. Wibben, B. G. Williams, M. G. Daly, and D. S. Lauretta, "Stereophotoclinometry for OSIRIS-REx Spacecraft Navigation," *The Planetary Science Journal*, Vol. 4, Sept. 2023, p. 167, 10.3847/PSJ/ace31d.

[16] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, Massachusetts: The MIT Press, 2005.

[17] M. W. Givens and J. W. McMahon, "Square-Root Extended Information Filter for Visual-Inertial Odometry for Planetary Landing," *Journal of Guidance, Control, and Dynamics*, Vol. 46, Feb. 2023, pp. 231–245, 10.2514/1.G006849.

[18] A. Mourikis, N. Trawny, S. Roumeliotis, A. Johnson, A. Ansar, and L. Matthies, "Vision-Aided Inertial Navigation for Spacecraft Entry, Descent, and Landing," *IEEE Transactions on Robotics*, Vol. 25, Apr. 2009, pp. 264–280, 10.1109/TRO.2009.2012342.

[19] J. A. Christian, L. Hong, P. McKee, R. Christensen, and T. P. Crain, "Image-Based Lunar Terrain Relative Navigation Without a Map: Measurements," *Journal of Spacecraft and Rockets*, Vol. 58, Jan. 2021, pp. 164–181, 10.2514/1.A34875.

[20] A. I. Mourikis and S. I. Roumeliotis, "A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation," *Proceedings 2007 IEEE International Conference on Robotics and Automation*, Rome, Italy, IEEE, Apr. 2007, pp. 3565–3572, 10.1109/ROBOT.2007.364024.

[21] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "Observability-Based Rules for Designing Consistent EKF SLAM Estimators," *The International Journal of Robotics Research*, Vol. 29, Apr. 2010, pp. 502–528, 10.1177/0278364909353640.

[22] J. Castellanos, R. Martinez-Cantin, J. Tardós, and J. Neira, "Robocentric Map Joining: Improving the Consistency of EKF-SLAM," *Robotics and Autonomous Systems*, Vol. 55, Jan. 2007, pp. 21–29, 10.1016/j.robot.2006.06.005.

[23] S. P. A. Ram and R. Zanetti, "Robocentric Simultaneous Localization and Mapping," *Journal of Guidance, Control, and Dynamics*, Vol. 47, Apr. 2024, pp. 742–752, 10.2514/1.G007587.

[24] Z. Huai and G. Huang, "Robocentric Visual–Inertial Odometry," *The International Journal of Robotics Research*, Vol. 41, June 2022, pp. 667–689, 10.1177/0278364919853361.

[25] C. I. Restrepo, P.-T. Chen, R. R. Sostaric, and J. M. Carson, "Next-Generation NASA Hazard Detection System Development," *AIAA Scitech 2020 Forum*, Orlando, FL, American Institute of Aeronautics and Astronautics, Jan. 2020, 10.2514/6.2020-0368.

[26] K. C. Ward, K. W. Smith, I. S. Rowe, J. M. Harper, D. W. Adams, S. M. Pedrotty, and G. F. Mendeck, "Lidar-Based Safe Site Relative Navigation," *AIAA SCITECH 2024 Forum*, Orlando, FL, American Institute of Aeronautics and Astronautics, Jan. 2024, 10.2514/6.2024-1583.

[27] R. Zanetti, *Advanced Navigation Algorithms for Precision Landing*. PhD thesis, University of Texas at Austin, Austin, Dec. 2007.

[28] "Apollo 17 Image Library," `https://www.nasa.gov/history/alsj/a17/images17.html#Pans`.

[29] "NASA Image and Video Library," `https://images.nasa.gov/details/LRC-1964-B701_P-11454`.

[30] K. Lee, W. Wijesoma, and J. Guzman, "On the Observability and Observability Analysis of SLAM," *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, IEEE, Oct. 2006, pp. 3569–3574, 10.1109/IROS.2006.281646.

[31] F. L. Markley and J. L. Crassidis, *Fundamentals of Spacecraft Attitude Determination and Control*. New York, NY: Springer New York, 2014, 10.1007/978-1-4939-0802-8.

[32] D. Simon, *Optimal State Estimation: Kalman, H [Infinity], and Nonlinear Approaches*. Hoboken, NJ: Wiley-Interscience, 2006.

[33] K. P. B. Chandra and D.-W. Gu, *Nonlinear Filtering: Methods and Applications*. Cham: Springer International Publishing, 2019, 10.1007/978-3-030-01797-2.

[34] J. S. McCabe and K. J. DeMars, "Anonymous Feature-Based Terrain Relative Navigation," *Journal of Guidance, Control, and Dynamics*, Vol. 43, Mar. 2020, pp. 410–421, 10.2514/1.G004423.

[35] "LN-200S & LN-200HPS Inertial Measurement Unit (IMU)," `https://cdn.northropgrumman.com/-/media/Project/Northrop-Grumman/ngc/what-we-do/space/ln-200s-ln-200hps-imu/LN-200S-Inertial-Measurement-Unit-IMU-datasheet.pdf`.

[36] A. Konopliv, "Recent Gravity Models as a Result of the Lunar Prospector Mission," *Icarus*, Vol. 150, Mar. 2001, pp. 1–18, 10.1006/icar.2000.6573.