Velocimeter LIDAR-Based Multiplicative Extended Kalman Filter for Terrain Relative Navigation

Davis Weaver Adams*, Caleb Peck † and Manoranjan Majji‡
Texas A&M University, College Station, TX, 77840

This paper presents a Multiplicative Extended Kalman Filter (MEKF) framework using a state-of-the-art velocimeter Light Detection and Ranging (LIDAR) sensor for Terrain Relative Navigation (TRN) applications. The newly developed velocimeter LIDAR is capable of providing simultaneous position, Doppler velocity, and reflectivity measurements for every point in the point cloud. This information, along with pseudo-measurements from point cloud registration techniques, a novel bulk velocity batch state estimation process and inertial measurement data, is fused within a traditional Kalman filter architecture. Results from extensive emulation robotics experiments performed at Texas A&M's Land, Air, and Space Robotics (LASR) laboratory and Monte Carlo simulations are presented to evaluate the efficacy of the proposed algorithms.

I. Introduction

The maturation of space related technology has enabled programs around the world to develop plans for increasingly challenging exploration missions. Missions involving landing on extra-terrestrial bodies such as planets, moons, or asteroids are inherently difficult due to large communication delays and the lack of an existing absolute positioning system (i.e. GPS), rendering human remote control impractical. During entry, descent, and landing (EDL), high accuracy position, attitude, and velocity estimates are extremely important for trajectory control to preclude landing in hazardous locations. Historically, EDL navigation systems relied on inertial navigation systems (INS) which integrate the sensed acceleration and angular velocity from an inertial measurement unit (IMU). Naturally, exclusively inertial based navigation systems are subject to unbounded error accumulation due to the integration of noises and biases. Combined with high initial state uncertainty upon re-entry, this error accumulation has historically led to large landing error ellipses on the order of 100-300 km as was the case for the Viking Landers, Mars Pathfinder, Mars Polar Lander, and the Mars Exploration Rovers [1].

To alleviate estimate drift, measurements from peripheral, often optical, sensors are fused within a navigation filter architecture. The abundance of known natural 'features', such as craters and large standing boulders, on extra-terrestial bodies of interest can serve as reference points for localization purposes. Upon initial implementation, it became evident that extracting useful information from a raw 2D image was impeded by challenges such as feature scale variance, dynamic lighting, and image blur. Compounding these challenges is the inherent low observability of a 3D scene from a 2D projection. To eliminate the effects of noisy measurements and inherent uncertainty, stochastic estimation techniques can be used. Trawny et. al formalized an Extended Kalman filter (EKF) framework to estimate vehicle pose and velocity during EDL by fusing IMU data and observations of *a priori* Mapped Landmarks (MLs) such as craters or other visual features [1]. Compared to a system relying exclusively on IMU integration, this stochastic approach reduced the size of the landing uncertainty ellipse by three orders of magnitude. In an alternate approach, Wong and Majji employed a stereo vision camera to perform simultaneous localization and mapping (SLAM) using a zero-jerk kinematic model and feature tracking [2]. It was shown that this use of basic relative motion models between a static scene and moving target can be recast as a recursive least squares problem. This simplification significantly reduces the computational burden that accompanies most computer vision techniques while still providing accurate estimates. While these works presented respectable results, they still do not address the large problem of scale variance when using features from 2D images which can lead to diverging 3D position and velocity estimates.

Light Detection and Ranging (LIDAR) systems are often used to accompany optical sensors for space exploration missions of high importance [3–6]. In contrast to 2D imaging, which project points in 3D onto a 2D image plane, LIDAR sensing systems eliminate any scale variance by using time-of-flight (ToF) measures and signal processing

[*] Graduate Research Assistant, Department of Aerospace Engineering, Texas A&M University, dwadams@tamu.edu

[†] Graduate Research Assistant, Department of Aerospace Engineering, Texas A&M University, cpeck7@tamu.edu

[‡] Assistant Professor, Director, LASR Laboratory, Department of Aerospace Engineering, AIAA Associate Fellow, mmajji@tamu.edu

to produce a 3D position estimate of points in its field-of-view (FOV), resulting in a point cloud. There has been significant work done to leverage the sensed point clouds for applications including object detection [7], 3D mapping [8], and odometry [9]. Toth et al. fused LIDAR and IMU sensing systems to recover the vehicle trajectory in GPS denied environments given a predefined reference surface map using classical point cloud registration techniques [10]. While clear drawbacks exist in this approach, including situations when IMU drift rates exceed desired thresholds or the registration technique fails to converge, it is highly advantageous for generating an *a priori* absolute position estimate without GPS support before terminal guidance algorithms are initiated. Recent notable work by Hewitt et al. proposed a LIDAR-based terrain relative navigation approach called Map Relative Localization (MRL) for autonomous landing on Europa: a mission which requires position errors of less than 100m [11]. Results showed that LIDAR-based MRL is capable of providing localization performance similar to passive optical MRL with a clear advantage of being light insensitive. One note declared in this referenced work was the extreme importance of accurate velocimetry during EDL operations. While traditional inertial-based velocimetry can provide reasonable estimates for applications with wide error margins, velocimetry is a research area that still needs significant attention.

The use of a Doppler LIDAR for precise altimetry and velocimetry during EDL operations was proposed in [12]. Informed by the Doppler shift between the emitted and received frequency modulated continuous wave (FMCW), a Doppler LIDAR is capable of accurately measuring line-of-sight (LOS) range and velocity. There are several nationwide projects, such as the NASA Safe and precise Landing - Integrated Capabilities Evolution (SPLICE) project [6], to mature GN&C technology for EDL operations. Researchers at NASA Langley Research Center sought to curb large uncertainty of vehicle velocity by developing the Navigation Doppler LIDAR (NDL). The NDL consists of three fiber-coupled transmit/receive telescopes driven by a seed laser and fiber optic amplifier to obtain LOS range and velocity. NDL measurements are utilized to minimize velocity and position navigation error, a significant source of localization error during EDL. Recent advances in this technology enable tighter control of vertical and lateral translational velocities during descent and landing with proposed velocity and range accuracies on the order of 2 cm/s and 2 m respectively.

This work presents a comprehensive Multiplicative Extended Kalman filter (MEKF) framework for state estimation during EDL operations using a newly developed, state-of-the-art velocimeter LIDAR capable of simultaneously measuring range, LOS velocity, and intensity return of every point in the point cloud. Most notably, novel measurement models for data obtained via the velocimeter LIDAR are derived in full. These new models include a relative norm scaled Doppler velocity measurement, frame-to-frame relative translation and orientation measurements obtained from point cloud registration, and instantaneous translational and angular velocity pseudo-measurements obtained from a novel batch state estimation methodology. The rest of the paper is organized as follows. Section II presents a mathematical description of using a velocimeter LIDAR for relative navigation. Next, Section III presents the MEKF in its entirety including pertinent background information regarding attitude kinematics and IMU motion models, and newly derived measurement models for the velocimeter LIDAR. Section IV delivers results from emulation robotics experiments performed at Texas A&M's Land, Air, and Space Robotics (LASR) laboratory. Finally, Section V provides an overview of the proposed techniques and conclusions on the work.

II. Velocimeter LIDAR Based Navigation

The ultimate goal of any TRN filter is to estimate the body's position (${}^{\mathcal{T}}\mathbf{p}_B$), attitude ($\mathbf{q}_{\mathcal{B}/\mathcal{T}}$), and velocity (${}^{\mathcal{T}}\mathbf{v}_B$) with respect to the terrain, as displayed in Fig. 1. As aforementioned, the velocimeter LIDAR directly measures 3D position and LOS velocity in the sensor's reference frame. In order to embed this information into a TRN filter, it must be related to the terrain of interest in a mathematically rigorous form.

A. Terrain Relative Position and Velocity

Consider a collection of known terrain features sensed by the LIDAR to be ${}^{\mathcal{L}}\mathbf{p}_{k_i}$, as displayed in Fig. 1, where the left superscript denotes the frame of reference, right subscript denotes the time step, and right sub-subscript denotes the i^{th} feature in the point cloud. The location of ${}^{\mathcal{L}}\mathbf{p}_{k_i}$ as a function of the states of interest is given by:

$$ {}^{\mathcal{L}}\mathbf{p}_{k_i} = C_{\mathcal{L}/\mathcal{B}} \left(C_{\mathcal{B}/\mathcal{T}} \left({}^{\mathcal{T}}\mathbf{p}_i - {}^{\mathcal{T}}\mathbf{p}_B \right) - {}^{\mathcal{B}}\boldsymbol{\rho}_L \right) \tag{1} $$

where $C_{\mathcal{L}/\mathcal{B}}$ is the direction cosine matrix mapping the body frame to the LIDAR frame, $C_{\mathcal{B}/\mathcal{T}}$ is the direction cosine matrix mapping the terrain frame to the body frame, and ${}^{\mathcal{B}}\boldsymbol{\rho}_L$ is the position of the LIDAR with respect to the origin of the body reference frame coordinatized in the body frame. Taking an inertial derivative of Eqn. 1 yields the relative

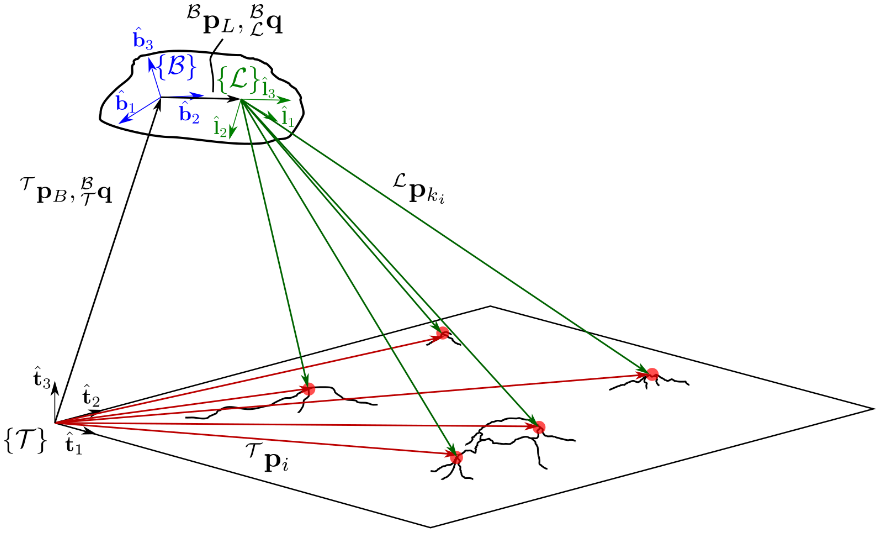

Fig. 1 Feature Detection

velocity of each observation which is given by:

$${}^{\mathcal{L}}\mathbf{v}_{k_i} = C_{\mathcal{L}/\mathcal{B}}\left[C_{\mathcal{B}/\mathcal{T}}\left({}^{\mathcal{T}}\mathbf{p}_i - {}^{\mathcal{T}}\mathbf{p}_B\right)^{\times}\right]{}^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{T}} - C_{\mathcal{L}/\mathcal{B}}C_{\mathcal{B}/\mathcal{T}}{}^{\mathcal{T}}\dot{\mathbf{p}}_B \tag{2}$$

where $[\cdot^{\times}]$ denotes the skew symmetric cross product matrix of a vector, ${}^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{T}}$ is the angular velocity of the body frame with respect to the terrain frame coordinatized in the body frame, and ${}^{\mathcal{T}}\dot{\mathbf{p}}_B$ is the translational velocity of the body coordinatized in the terrain frame. The velocimeter LIDAR measures LOS (or Doppler) velocity, which is defined as the projection of the relative velocity onto the relative position vector. Mathematically, the point-wise LOS velocity as sensed by the LIDAR is given by:

$$v_{k_i} = \frac{1}{\|{}^{\mathcal{L}}\mathbf{p}_{k_i}\|}\left({}^{\mathcal{L}}\mathbf{p}_{k_i}^{T}\,{}^{\mathcal{L}}\mathbf{v}_{k_i}\right) \tag{3}$$

where $\|\cdot\|$ denotes the magnitude of a vector. It is important to note that Eqn. 3 can also be written as

$$v_{k_i} = \frac{1}{\|{}^{\mathcal{T}}\mathbf{p}_{k_i}\|}\left({}^{\mathcal{T}}\mathbf{p}_{k_i}^{T}\,{}^{\mathcal{T}}\mathbf{v}_{k_i}\right) \tag{4}$$

where

$${}^{\mathcal{T}}\mathbf{p}_{k_i} = {}^{\mathcal{T}}\mathbf{p}_i - {}^{\mathcal{T}}\mathbf{p}_B - C_{\mathcal{B}/\mathcal{T}}^{T}\,{}^{\mathcal{B}}\boldsymbol{\rho}_L \tag{5}$$

$${}^{\mathcal{T}}\mathbf{v}_{k_i} = -{}^{\mathcal{T}}\dot{\mathbf{p}}_B + C_{\mathcal{B}/\mathcal{T}}^{T}\left[{}^{\mathcal{B}}\boldsymbol{\rho}_L^{\times}\right]{}^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{T}} \tag{6}$$

Traditional LIDAR-based Kalman filter approaches exclusively use the sensed 3D feature position (Eqn. 1) as the measurement model; however, one extremely common technique for leveraging successive point clouds to estimate the change in position and orientation is called point cloud registration. While relying on point cloud registration alone for trajectory recovery is ill-advised due to accumulated error in the registration process, the frame-to-frame relative pose change is valuable information that can be fused within a filter architecture. Further, when paired with Doppler velocity measurements, a sequential batch state least squares estimation process can be implemented to estimate the instantaneous translational and angular velocity of the body [13]. These estimates can also be embedded into a navigation filter as pseudo-measurements with associated measurement covariance from the batch state estimation approach. The point cloud registration and bulk velocity estimation processes are discussed in detail below.

B. Point Cloud Registration and Bulk Velocity Estimation

Point cloud registration algorithms have long been used in various fields and applications such has object detection, 3D localization and mapping, and trajectory recovery. The overarching concept of point cloud registration is to estimate the homogeneous transformation comprised of relative orientation, $C_{\mathcal{L}_k/\mathcal{L}_{k-1}}$, and translation, $\mathbf{t}_k$, which transforms the point cloud at step $k-1$ to the point cloud at step k by minimizing a pre-defined cost function. Mathematically, this transformation is given by:

$$\mathbf{p}_{k_i} = C_{\mathcal{L}_k/\mathcal{L}_{k-1}}\mathbf{p}_{k-1_i} + \mathbf{t}_k \tag{7}$$

where the right subscript k denotes quantities that change frame-to-frame, the right subscript i denotes the i^{th} point of the terrain where $i = 1, \ldots, N$, $C_{\mathcal{L}_k/\mathcal{L}_{k-1}}$ is the direction cosine matrix mapping vectors in the $k-1$ LIDAR frame to the k LIDAR frame, and $\mathbf{t}_k$ is the translation vector between the origins of the $k-1$ and k LIDAR reference frames, as displayed in Fig. 2. It is important to note that all vectors in the ensuing developments are coordinatized in the LIDAR reference frame so the left superscript will temporarily be removed for brevity.

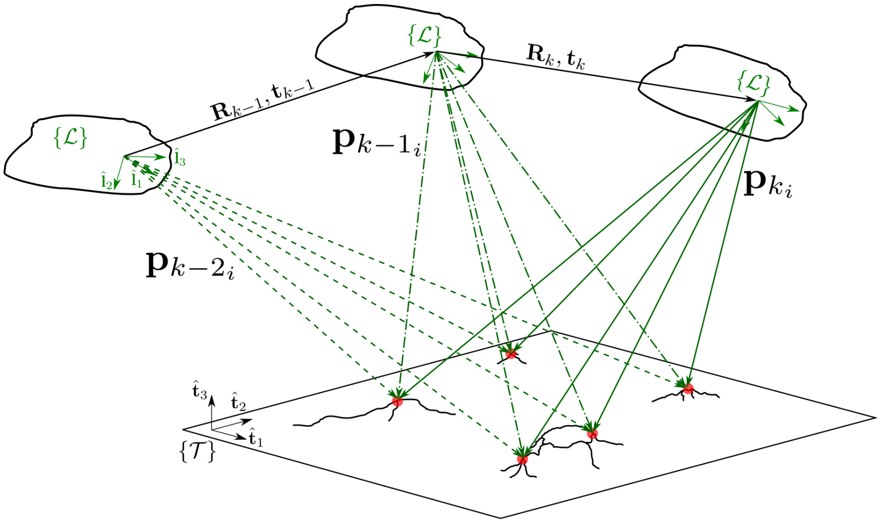

Fig. 2 Batch State Estimation

There are many approaches to point cloud registration; however, most employ either Singular Value Decomposition (SVD) [14] or Principal Component Analysis (PCA) [15] based registrations. Other methods use a more advanced iterative approach based on the Iterative Closest Point algorithm (ICP) such as the non-linear ICP [16], generalized ICP [17], and non-rigid ICP [18] variants. For most rigid-scene applications, as is the case for TRN, the standard ICP algorithm is suitable, and was used in this work to estimate the frame-to-frame homogeneous transformations.

As previously mentioned, assuming the point cloud registration of choice provides accurate estimates for $C_{\mathcal{L}_k/\mathcal{L}_{k-1}}$ and $\mathbf{t}_k$, the sensed position and LOS velocity field can be leveraged to estimate translational and angular velocity coordinatized in the LIDAR frame. Taking a time derivative of Eq. (7) leads to the relative velocity of the i^{th} point of the terrain resulting in

$$\mathbf{v}_{k_i} = -\left[\boldsymbol{\omega}_k{}^{\times}\right] C_{\mathcal{L}_k/\mathcal{L}_{k-1}}\mathbf{p}_{k-1_i} + \dot{\mathbf{t}}_k + C_{\mathcal{L}_k/\mathcal{L}_{k-1}}\dot{\mathbf{p}}_{k-1_i} \tag{8}$$

where $[\cdot^{\times}]$ denotes the skew symmetric cross product matrix of a vector and $\boldsymbol{\omega}_k$ is the angular velocity of the body coordinatized in the LIDAR reference frame. It is important to note that in this context, $\boldsymbol{\omega}_k = C_{\mathcal{L}/\mathcal{B}}{}^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{T}}$ and $\dot{\mathbf{t}}_k = -C_{\mathcal{L}/\mathcal{B}}C_{\mathcal{B}/\mathcal{T}}{}^{\mathcal{T}}\dot{\mathbf{p}}_B$ where ${}^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{T}}$ and ${}^{\mathcal{T}}\dot{\mathbf{p}}_B$ have been previously defined. Since the terrain is fixed, Eqn. (7) can be simplified by setting $\dot{\mathbf{p}}_{k-1_i} = 0$. Furthermore, Eqn. (8) can be rewritten to be linear in $\boldsymbol{\omega}_k$ by reversing the cross product resulting in

$$\mathbf{v}_{k_i} = \left[(C_{\mathcal{L}_k/\mathcal{L}_{k-1}}\mathbf{p}_{k-1_i})^{\times}\right]\boldsymbol{\omega}_k + \dot{\mathbf{t}}_k \tag{9}$$

Now, consider the relative Doppler velocity of the i^{th} point of the terrain from the LIDAR's perspective to be

$$\begin{aligned} \tilde{v}_{k_i} &= \frac{\mathbf{p}_{k_i}{}^T\mathbf{v}_{k_i}}{\|\mathbf{p}_{k_i}\|} \\ &= \frac{1}{\|\mathbf{p}_{k_i}\|}(\mathbf{p}_{k_i}{}^T\left[(C_{\mathcal{L}_k/\mathcal{L}_{k-1}}\mathbf{p}_{k-1_i})^{\times}\right]\boldsymbol{\omega}_k + \mathbf{p}_{k_i}{}^T\dot{\mathbf{t}}_k) \end{aligned} \tag{10}$$

It is important to note that Eqn. (11) is linear in both angular and translational velocity, which can directly be observed by rewriting Eqn. (9) in matrix form as

$$\tilde{v}_{k_i} = \begin{bmatrix} \frac{\mathbf{p}_{k_i}}{\|\mathbf{p}_{k_i}\|} & \frac{\mathbf{p}_{k_i}{}^T \left[(C_{\mathcal{L}_k/\mathcal{L}_{k-1}} \mathbf{p}_{k-1_i})^{\times} \right]}{\|\mathbf{p}_{k_i}\|} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{t}}_k \\ \boldsymbol{\omega}_k \end{bmatrix} \tag{11}$$

For each frame, there are N points with corresponding positions in the LIDAR frame and LOS velocities; therefore, an over-constrained system of linear equations of the form $\mathbf{y} = H\mathbf{z}$ where $\mathbf{y} \epsilon R^{N\times 1}$, $H \epsilon R^{N\times 6}$, and $\mathbf{z} \epsilon R^6 \times 1$ can be constructed. Linear system theory suggests the velocity state vector, $\left[\dot{\mathbf{t}}_k, \boldsymbol{\omega}_k\right]^T$, can be solved for by employing least squares techniques; however, further investigation into this system reveals extreme ill-conditioning of the H matrix leading to erroneous estimates for the state vector, $\mathbf{z}$. The condition number of the system, κ, defined by the ratio of the largest to the smallest singular value of $H^T H$ remains $\kappa > 1e6$.

To surmount this issue, regularization techniques can be implemented to aid in the conditioning of the system, as performed by Hu et al. [19]. However, selection of regularization values is an arbitrary process and this work found it to produce erroneous results due to a single viewpoint in contrast to Hu's multi-view setup. Another approach to aid in conditioning is to append other linear systems which are functions of the same state variables as described in detail by Adams and Majji [13]. Assuming over a small period of time, or batch of frames, the body is not accelerating in any degree-of-freedom, a set of linear equations can be constructed to solve for $\dot{\mathbf{t}}_k$ and $\boldsymbol{\omega}_k$ via a sequential batch least squares approach. This system is given by:

$$\begin{bmatrix} \tilde{v}_{f\,i} \\ \hdashline 0 \\ \hdashline \dot{\boldsymbol{\beta}}_f \end{bmatrix} = \begin{bmatrix} \frac{\mathbf{p}_{f_i}^T}{\|\mathbf{p}_{f_i}\|} & \frac{\mathbf{p}_{f_i}^T \left[(C_{f/k-F} \mathbf{p}_{k-F_i})^{\times} \right]}{\|\mathbf{p}_{f_i}\|} \\ \hdashline \nabla\tilde{v}_f^T & \nabla\tilde{v}_f^T \left[(C_{f/k-F} \mathbf{p}_{k-F_i})^{\times} \right] \\ \hdashline 0_{3\times 3} & \frac{1}{2}\left[I_{3\times 3} + \left[\boldsymbol{\beta}_f^{\times}\right] + \boldsymbol{\beta}_f \boldsymbol{\beta}_f^T \right] \end{bmatrix} \begin{bmatrix} \dot{\mathbf{t}}_k \\ \boldsymbol{\omega}_k \end{bmatrix} \tag{12}$$

where $\nabla\tilde{v}_f^T = \begin{bmatrix} \frac{\partial \tilde{v}_f}{\partial x} & \frac{\partial \tilde{v}_f}{\partial y} & \frac{\partial \tilde{v}_f}{\partial z} \end{bmatrix}$ is the velocity field gradient and $\boldsymbol{\beta}_f$ is the vector of Classical Rodrigues Parameters (CRPs) mapping the $(k-F)$ frame to the f frame for $f = k - F + 1, \ldots, k$ and $i = 1, \ldots, N_f$. Defining the total number of points acquired over F frames to be $N_T = \sum_{k-F+1}^{k} N_f$, the linear system presented by Eqn. (12) takes the form $\mathbf{y} = H\mathbf{z}$ where $\mathbf{y} \in \mathbb{R}^{(2N_T+3F)\times 1}$, $H \in \mathbb{R}^{(2N_T+3F)\times 6}$, and $\mathbf{z} \in \mathbb{R}^{6\times 1}$. The velocity state vector and associated error covariance are given by the standard least squares solution, $\hat{\mathbf{z}} = \left(H^T W H\right)^{-1} H^T W \tilde{\mathbf{y}}$ and $\left(H^T W H\right)^{-1}$, respectively, where W is a weighting matrix. Analysis of this new system indicated a significant improvement in the condition number such that, $\kappa \approx 1e2$, and is also much more capable of estimating bulk velocities with representative error covariance. It is important to note that point cloud registration will provide the direction cosine matrix, and thus CRP [20] estimates. Velocity field gradients ($\nabla\tilde{v}_f$) and CRP rates ($\dot{\boldsymbol{\beta}}_f$) can be numerically approximated allowing the only unknowns in the linear system to be $\dot{\mathbf{t}}_k$ and $\boldsymbol{\omega}_k$.

This batch state estimation methodology can be implemented in a 'sliding window' fashion. Upon the availability of each new frame of data, the measurement vector, $\tilde{\mathbf{y}}$ and linear system matrix, H are back-filled with the last $F - 1$ frames of information, and the new frame completes the 'batch'. As previously mentioned, the velocity vector estimates and error covariance can be used as pseudo-measurements with associated measurement noise within a Kalman filter framework, which is discussed in detail below.

III. Terrain Relative Navigation Multiplicative Extended Kalman Filter

The main difference between the MEKF and the standard EKF is the employment of the quaternion as the attitude parameterization of choice. The quaternion is a four-dimensional vector defined as $\mathbf{q} = \begin{bmatrix}\boldsymbol{\rho}^T & q_4\end{bmatrix}^T$ where $\boldsymbol{\rho} = \hat{\mathbf{e}}\sin(\theta/2)$ and $q_4 = \cos(\theta/2)$. The vector $\hat{\mathbf{e}}$ is Euler's principal axis and θ is the angle of rotation about that axis. Note that any direction cosine matrix (DCM) can be constructed from a quaternion via

$$C(\mathbf{q}) = \Xi^T(\mathbf{q})\,\Psi(\mathbf{q}) = \begin{bmatrix} q_4 I_{3\times 3} + [\boldsymbol{\rho}^{\times}] \\ -\boldsymbol{\rho}^T \end{bmatrix}^T \begin{bmatrix} q_4 I_{3\times 3} - [\boldsymbol{\rho}^{\times}] \\ -\boldsymbol{\rho}^T \end{bmatrix} \tag{13}$$

The quaternion attitude kinematics are given by [20]

$$\dot{\mathbf{q}}_{\mathcal{B}/\mathcal{T}} = \frac{1}{2}\Omega\left({}^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{T}}\right)\mathbf{q}_{\mathcal{B}/\mathcal{T}} \tag{14}$$

where

$$\Omega\left(\omega\right) = \begin{bmatrix} -\left[\omega^{\times}\right] & \omega \\ -\omega^{T} & 0 \end{bmatrix} \tag{15}$$

In contrast to all other attitude parameterizations, the quaternion must obey a unit norm constraint i.e $\mathbf{q}^T\mathbf{q} = 1$. Naturally, this suggests that a standard additive error model would lead to constraint violation. To ensure this condition is met at all times, a multiplicative error quaternion is defined as

$$\boldsymbol{\delta}\mathbf{q} = \mathbf{q} \otimes \hat{\mathbf{q}}^{-1} \tag{16}$$

where the ⊗ denotes quaternion multiplication and $\hat{\mathbf{q}}$ is the estimated quaternion. It is important to note that state error dynamics are needed to propagate the error covariance, which is discussed later. With that notion in mind, taking a time derivative of Eqn. 16 and employing Eqn. 15 yields the quaternion error dynamics given by [21]

$$\boldsymbol{\delta}\dot{\boldsymbol{\rho}} = -\left[\omega^{\times}\right]\boldsymbol{\delta\rho} + \frac{1}{2}\boldsymbol{\delta\omega} \tag{17}$$

$$\boldsymbol{\delta}\dot{\mathbf{q}}_4 = 0 \tag{18}$$

The foundation of any inertial-based EKF is the high-frequency input from an IMU. The large majority of IMUs consist of at least a gyroscope and accelerometer which are used to propagate the filter state dynamics in between measurement updates. The raw signals of the gyroscope and accelerometer are corrupted by Gaussian white noise and bias. The gyroscope and accelerometer measurement models are given by

$$^{\mathcal{B}}\tilde{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{T}} = {}^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{T}} + \boldsymbol{\beta}_g + \boldsymbol{\eta}_{g_v} \tag{19}$$

$$\dot{\boldsymbol{\beta}}_g = \boldsymbol{\eta}_{g_u} \tag{20}$$

$$^{\mathcal{B}}\tilde{\mathbf{a}}_B = {}^{\mathcal{B}}\mathbf{a}_B + \boldsymbol{\beta}_a + \boldsymbol{\eta}_{a_v} \tag{21}$$

$$\dot{\boldsymbol{\beta}} = \boldsymbol{\eta}_{a_u} \tag{22}$$

where $\boldsymbol{\beta}_g$ and $\boldsymbol{\beta}_a$ are the gyroscope and acclerometer biases, and $\boldsymbol{\eta}_{g_v}$, $\boldsymbol{\eta}_{g_u}$, $\boldsymbol{\eta}_{a_v}$ and $\boldsymbol{\eta}_{a_u}$ are zero-mean Gaussian white-noise processes. The estimated angular velocity and translational acceleration are given by

$$^{\mathcal{B}}\hat{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{T}} = {}^{\mathcal{B}}\tilde{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{T}} - \hat{\boldsymbol{\beta}}_g \tag{23}$$

$$^{\mathcal{B}}\hat{\mathbf{a}}_B = {}^{\mathcal{B}}\tilde{\mathbf{a}}_B - \hat{\boldsymbol{\beta}}_a \tag{24}$$

Substituting Eqns. 19 and 23 into the angular velocity error, $\boldsymbol{\delta\omega} = \boldsymbol{\omega} - \hat{\boldsymbol{\omega}}$, yields

$$\boldsymbol{\delta\omega} = {}^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{T}} - {}^{\mathcal{B}}\hat{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{T}} = -\left(\Delta\boldsymbol{\beta}_g + \eta_{g_v}\right) \tag{25}$$

where $\Delta\boldsymbol{\beta}_g = \boldsymbol{\beta}_g - \hat{\boldsymbol{\beta}}_g$. Substituting this result into the error quaternion dynamics gives

$$\boldsymbol{\delta\rho} = -\left[{}^{\mathcal{B}}\hat{\boldsymbol{\omega}}^{\times}_{\mathcal{B}/\mathcal{T}}\right]\boldsymbol{\delta\rho} - \frac{1}{2}\left(\Delta\boldsymbol{\beta}_g + \boldsymbol{\eta}_{g_v}\right) \tag{26}$$

A common simplification of Eqn. 26 is given by the small angle approximation $\boldsymbol{\delta\rho} \approx \boldsymbol{\delta\alpha}/2$ where $\boldsymbol{\delta\alpha}$ is the vector of roll, pitch, and yaw error angles [21]. Employing this simplification yields

$$\boldsymbol{\delta}\dot{\boldsymbol{\alpha}} = -\left[{}^{\mathcal{B}}\hat{\boldsymbol{\omega}}^{\times}_{\mathcal{B}/\mathcal{T}}\right]\boldsymbol{\delta\alpha} - \left(\Delta\boldsymbol{\beta}_g + \boldsymbol{\eta}_{g_v}\right) \tag{27}$$

which completes the filter formulation for the attitude states. The translational dynamics are simply given by

$$^{\mathcal{T}}\dot{\mathbf{p}}_B = {}^{\mathcal{T}}\mathbf{v}_B \tag{28}$$

$$^{\mathcal{T}}\dot{\mathbf{v}}_B = C_{\mathcal{T}/\mathcal{B}}{}^{\mathcal{B}}\mathbf{a}_B + {}^{\mathcal{T}}\mathbf{g} \tag{29}$$

where $^{\mathcal{T}}\mathbf{g}$ is the gravity vector in the Terrain frame. The derivation for the translational error dynamics is forgone for brevity, but are given by

$$\Delta^{\mathcal{T}}\dot{\mathbf{p}}_B = \Delta^{\mathcal{T}}\mathbf{v}_B \tag{30}$$

$$\Delta^{\mathcal{T}}\dot{\mathbf{v}}_B = -\hat{C}^T_{\mathcal{B}/\mathcal{T}}\left(\Delta\boldsymbol{\beta}_a + \boldsymbol{\eta}_{a_v} + \left[\left({}^{\mathcal{B}}\tilde{\mathbf{a}} - \hat{\boldsymbol{\beta}}_a\right)^{\times}\right]\boldsymbol{\delta\alpha}\right) \tag{31}$$

Thus, if the MEKF filter error state vector is defined as

$$\boldsymbol{\Delta}\mathbf{x} = \begin{bmatrix}\boldsymbol{\delta\alpha} & \Delta^{\mathcal{T}}\mathbf{p}_B & \Delta^{\mathcal{T}}\mathbf{v}_B & \Delta\boldsymbol{\beta}_g & \Delta\boldsymbol{\beta}_a\end{bmatrix}^T \tag{32}$$

then the complete error dynamics can be represented by

$$\Delta\dot{\mathbf{x}} = F\Delta\mathbf{x} + G\mathbf{w} \tag{33}$$

where

$$F = \begin{bmatrix} -\left[\hat{\omega}^{\times}\right] & 0_{3\times3} & 0_{3\times3} & -\mathbb{I}_3 & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & \mathbb{I}_3 & 0_{3\times3} & 0_{3\times3} \\ -\hat{C}^T_{\mathcal{B}/\mathcal{T}}\left[\hat{\mathbf{a}}^{\times}\right] & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & -\hat{C}^T_{\mathcal{B}/\mathcal{T}} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \end{bmatrix} \;\&\; G = \begin{bmatrix} -\mathbb{I}_3 & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & -\hat{C}^T_{\mathcal{B}/\mathcal{T}} & 0_{3\times3} \\ 0_{3\times3} & \mathbb{I}_3 & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & \mathbb{I}_3 \end{bmatrix} \tag{34}$$

and $\mathbf{w} = \begin{bmatrix}\boldsymbol{\eta}_{g_v} & \boldsymbol{\eta}_{g_u} & \boldsymbol{\eta}_{a_v} & \boldsymbol{\eta}_{a_u}\end{bmatrix}^T$. The MEKF error covariance, defined as the expected value of the outer product of the state error vector, is given by $E\{\Delta\mathbf{x}\Delta\mathbf{x}^T\}$. The error covariance is propagated along with the core states each time a new IMU measurement is available via

$$\dot{P} = FP + PF^T + GQG^T \tag{35}$$

where $G = E\{\mathbf{w}\mathbf{w}^T\}$ is the process noise covariance matrix. Due to the drifting nature of the IMU bias, the bias error will grow unbounded unless the estimate is corrected through exterior measurements.

This work proposes the implementation of four different measurement models to correct the IMU bias drift: 1) 3D feature position, 2) norm scaled feature relative Doppler velocity, 3) relative orientation and translation provided by point cloud registration, and 4) bulk velocity estimates provided by the aforementioned batch state estimation process. Models (3) and (4) are useful for situations when features cannot be detected and/or corresponded. Adopting the standard Kalman filter measurement and update equation structure, all measurement equations are corrupted by zero-mean white Gaussian noise, $\boldsymbol{\nu}$, taking the form

$$\tilde{\mathbf{y}} = \mathbf{h}(\mathbf{x}) + \boldsymbol{\nu} \tag{36}$$

$$= \mathbf{h}(\hat{\mathbf{x}} + \boldsymbol{\Delta}\mathbf{x}) + \boldsymbol{\nu} \tag{37}$$

Upon receiving a measurement at time-step k, the states and error covariance are updated according to Eqns. 38 and 39 commensurate to the Kalman gain, K_k, which is given by Eqn. 40

$$\hat{\mathbf{x}}^+_k = \hat{\mathbf{x}}^-_k + K_k\mathbf{r}_k \tag{38}$$

$$P^+_k = \left(\mathbb{I} - K_kH_k\right)P^-_k \tag{39}$$

$$K_k = P^-_kH^T_k\left(H_kP^-_kH^T_k + R_k\right)^{-1} \tag{40}$$

where $\mathbf{r}_k$ is the measurement residual, H_k is the measurement sensitivity matrix and R_k is the measurement noise covariance. The classical EKF is an indirect filter which estimates the state errors online. The relation between the measurements and the sensitivity matrix can be seen by applying a first-order Taylor series expansion to the measurement residual about the nominal error state and noise trajectories as

$$\mathbf{r}_k = \tilde{\mathbf{y}}_k\left(\hat{\mathbf{x}} + \boldsymbol{\Delta}\mathbf{x}\right) - \hat{\mathbf{y}}_k\left(\hat{\mathbf{x}}\right) \tag{41}$$

$$\mathbf{r}_k \approx \hat{\mathbf{r}}_k + \frac{\partial\mathbf{r}_k}{\partial\boldsymbol{\Delta}\mathbf{x}_k}\left(\boldsymbol{\Delta}\mathbf{x}_k - E\{\boldsymbol{\Delta}\mathbf{x}_k\}\right) + \frac{\partial\mathbf{r}_k}{\partial\boldsymbol{\nu}_k}\left(\boldsymbol{\nu}_k - E\{\boldsymbol{\nu}_k\}\right) \tag{42}$$

which is further simplified by noting $\hat{\mathbf{r}}_k \approx 0$, $H_k \equiv \frac{\partial\mathbf{r}_k}{\partial\boldsymbol{\Delta}\mathbf{x}_k}$, $E\{\boldsymbol{\Delta}\mathbf{x}\} = 0$, $\frac{\partial\mathbf{r}_k}{\partial\boldsymbol{\nu}_k} = \mathbb{I}_3$, and $E\{\boldsymbol{\nu}_k\} = 0$ yielding

$$\mathbf{r}_k \approx H\boldsymbol{\Delta}\mathbf{x}_k + \boldsymbol{\nu}_k \tag{43}$$

Since the measurement sensitivity matrix linearly maps the error states to the measurement residual, differentiating the measurement residual equations with respect to the error states directly produces each respective sensitivity matrix.

The first measurement model to be considered is the direct 3D position measurement given by the LIDAR. Assuming the observed 3D features can be corresponded to known feature locations, the 3D position of each observed feature coordinatized in the LIDAR frame is given by Eqn. 1. The position measurement residual is given by

$$\Delta^{\mathcal{L}}\mathbf{p}_{k_i} = C_{\mathcal{L}/\mathcal{B}}\left[\left(\hat{C}_{\mathcal{B}/\mathcal{T}}\left({}^{\mathcal{T}}\mathbf{p}_i - {}^{\mathcal{T}}\hat{\mathbf{p}}_B\right)\right)^{\times}\right]\delta\alpha - C_{\mathcal{L}/\mathcal{B}}\hat{C}_{\mathcal{B}/\mathcal{T}}\Delta^{\mathcal{T}}\mathbf{p}_B \tag{44}$$

where $\hat{C}_{\mathcal{B}/\mathcal{T}}$ is the DCM formed by the estimated quaternion which maps the Terrain frame to the body frame. Differentiating Eqn. 44 with respect to the error states yields the position measurement sensitivity matrix given by

$$H_{\mathbf{p}} = \begin{bmatrix} C_{\mathcal{L}/\mathcal{B}}\left[\left(\hat{C}_{\mathcal{B}/\mathcal{T}}\left({}^{\mathcal{T}}\mathbf{p}_i - {}^{\mathcal{T}}\hat{\mathbf{p}}_B\right)\right)^{\times}\right] & -C_{\mathcal{L}/\mathcal{B}}\hat{C}_{\mathcal{B}/\mathcal{T}} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \end{bmatrix} \tag{45}$$

The second measurement model is the relative Doppler velocity measurement of each feature. To reduce analytical complexity during construction of the sensitivity matrix related to the Doppler velocity measurement, the sensed velocity can be scaled by the position magnitude. Thus, the norm scaled relative Doppler velocity measurement equation is written as

$$\tilde{y} = \|{}^{\mathcal{L}}\mathbf{p}_{k_i}\| v_{k_i} \tag{46}$$

Further, although v_{k_i} can be represented by Eqn. 3, using the form given by Eqn. 4 significantly simplifies the resulting measurement residual. The sensitivity matrix for the scaled Doppler velocity measurement is given by

$$H_v = \begin{bmatrix} H_{v_{\delta\boldsymbol{\alpha}}} & H_{v_{\Delta^{\mathcal{T}}\mathbf{p}_B}} & H_{v_{\Delta^{\mathcal{T}}\dot{\mathbf{p}}_B}} & H_{v_{\Delta\boldsymbol{\beta}_g}} & 0_{3\times3} \end{bmatrix} \tag{47}$$

where

$$\begin{aligned}
H_{v_{\delta\boldsymbol{\alpha}}} &= \left({}^{\mathcal{T}}\hat{\dot{\mathbf{p}}}_B^T - {}^{\mathcal{T}}\mathbf{p}_i^T\right)\left[\left(\hat{C}_{\mathcal{B}/\mathcal{T}}^T\left[{}^{\mathcal{B}}\boldsymbol{\rho}_L^{\times}\right]{}^{\mathcal{B}}\hat{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{T}}\right)^{\times}\right] + {}^{\mathcal{B}}\boldsymbol{\rho}_L^T\left[\left(\hat{C}_{\mathcal{B}/\mathcal{T}}{}^{\mathcal{T}}\hat{\dot{\mathbf{p}}}_B\right)^{\times}\right] && (48)\\
H_{v_{\Delta^{\mathcal{T}}\mathbf{p}_B}} &= {}^{\mathcal{T}}\hat{\dot{\mathbf{p}}}_B^T - {}^{\mathcal{B}}\hat{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{T}}^T\left[{}^{\mathcal{B}}\boldsymbol{\rho}_L^{\times}\right]^T\hat{C}_{\mathcal{B}/\mathcal{T}} && (49)\\
H_{v_{\Delta^{\mathcal{T}}\mathbf{v}_B}} &= -{}^{\mathcal{T}}\mathbf{p}_i^T + {}^{\mathcal{T}}\hat{\dot{\mathbf{p}}}_B^T + {}^{\mathcal{B}}\boldsymbol{\rho}_L^T\hat{C}_{\mathcal{B}/\mathcal{T}} && (50)\\
H_{v_{\Delta\boldsymbol{\beta}_g}} &= \left({}^{\mathcal{T}}\hat{\dot{\mathbf{p}}}_B^T - {}^{\mathcal{T}}\mathbf{p}_i^T\right)\hat{C}_{\mathcal{B}/\mathcal{T}}^T\left[{}^{\mathcal{B}}\boldsymbol{\rho}_L^{\times}\right] && (51)
\end{aligned}$$

Next, assuming the point cloud registration technique of choice converges at time-step k, the frame-to-frame relative translation and orientation, ${}^{\mathcal{L}}\mathbf{t}_k$ and $\mathbf{q}_{\mathcal{L}_k/\mathcal{L}_{k-1}}$ are obtained which can be used as pseudo-measurements with associated covariance. Construction of the relative translation sensitivity matrix is trivial, and is given by

$$H_{\mathbf{t}} = \begin{bmatrix} C_{\mathcal{L}/\mathcal{B}}\left[\left(\hat{C}_{\mathcal{B}/\mathcal{T}}\left({}^{\mathcal{T}}\hat{\mathbf{p}}_{B_k} - {}^{\mathcal{T}}\hat{\mathbf{p}}_{B_{k-1}}\right)\right)^{\times}\right] & C_{\mathcal{L}/\mathcal{B}}\hat{C}_{\mathcal{B}/\mathcal{T}} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \end{bmatrix} \tag{52}$$

Formulating the sensitivity matrix associated with the frame-to-frame change in orientation is slightly more challenging due to the fact that the residual, $\tilde{\mathbf{q}}$, is multiplicative in nature. Mathematically, it is written as

$$\tilde{\mathbf{q}} = \mathbf{q}_{\mathcal{L}_k/\mathcal{L}_{k-1}} \otimes \hat{\mathbf{q}}_{\mathcal{L}_k/\mathcal{L}_{k-1}}^{-1} \tag{53}$$

This residual can be expanded and subsequently simplified as

$$\begin{aligned}
\tilde{\mathbf{q}} &= \mathbf{q}_{\mathcal{L}/\mathcal{B}} \otimes \begin{bmatrix}\delta\boldsymbol{\rho}\\ \delta\mathbf{q}_4\end{bmatrix} \otimes \hat{\mathbf{q}}_{\mathcal{B}/\mathcal{T}_k} \otimes \mathbf{q}_{\mathcal{B}/\mathcal{T}_{k-1}}^{-1} \otimes \mathbf{q}_{\mathcal{L}/\mathcal{B}}^{-1} \otimes \left(\mathbf{q}_{\mathcal{L}/\mathcal{B}} \otimes \hat{\mathbf{q}}_{\mathcal{B}/\mathcal{T}_k} \otimes \mathbf{q}_{\mathcal{B}/\mathcal{T}_{k-1}}^{-1} \otimes \mathbf{q}_{\mathcal{L}/\mathcal{B}}^{-1}\right)^{-1}\\
&= \mathbf{q}_{\mathcal{L}/\mathcal{B}} \otimes \begin{bmatrix}\delta\boldsymbol{\rho}\\ \delta\mathbf{q}_4\end{bmatrix} \otimes \mathbf{q}_{\mathcal{L}/\mathcal{B}}^{-1} && (54)\\
&= \mathbf{q}_{\mathcal{L}/\mathcal{B}} \otimes \mathbf{q}_{\mathcal{L}/\mathcal{B}}^{-1} \circledast \begin{bmatrix}\delta\boldsymbol{\rho}\\ \delta\mathbf{q}_4\end{bmatrix}
\end{aligned}$$

where

$$[\mathbf{q}\otimes] \equiv \begin{bmatrix} -\left[\boldsymbol{\rho}^{\times}\right] + q_4\mathbb{I}_3 & \boldsymbol{\rho} \\ -\boldsymbol{\rho}^T & q_4 \end{bmatrix} \;\&\; [\mathbf{q}\circledast] \equiv \begin{bmatrix} \left[\boldsymbol{\rho}^{\times}\right] + q_4\mathbb{I}_3 & \boldsymbol{\rho} \\ -\boldsymbol{\rho}^T & q_4 \end{bmatrix} \tag{55}$$

thus

$$\tilde{\mathbf{q}} = \frac{1}{2}\left[\mathbf{q}_{\mathcal{L}/\mathcal{B}}\otimes\right]\left[\mathbf{q}^{-1}_{\mathcal{L}/\mathcal{B}}\circledast\right]\begin{bmatrix} \boldsymbol{\delta\alpha} \\ 2\boldsymbol{\delta}\mathbf{q}_4 \end{bmatrix} \tag{56}$$

leading to the measurement sensitivity matrix for the change in attitude given by point cloud registration which itself is given by

$$H_{\mathbf{q}} = \begin{bmatrix} \frac{1}{2}\left(\mathbf{q}_{\mathcal{L}/\mathcal{B}}\otimes\right)\left(\mathbf{q}^{-1}_{\mathcal{L}/\mathcal{B}}\circledast\right)_{(1:3,1:3)} & 0_{3\times12} \end{bmatrix} \tag{57}$$

The last models are the direct 'measurements' of the angular and translational velocity coordinatized in the LIDAR frame given by the batch state estimation process discussed in Section II. The least squares estimates can directly be used as psuedo-measurements in the MEKF and the least squares covariance given by ${}^{\mathcal{L}}R_k = \left(H^T W H\right)^{-1}$ is used as the measurement noise covariance. Taking ${}^{\mathcal{L}}\tilde{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{T}}$ and ${}^{\mathcal{L}}\mathbf{v}_{\mathcal{B}}$ to be the direct 'measurements' provided by the batch least squares process, the respective sensitivity matrices are given by

$$H_{\omega} = \begin{bmatrix} 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & -C_{\mathcal{L}/\mathcal{B}} & 0_{3\times3} \end{bmatrix} \tag{58}$$

$$H_{\mathbf{v}} = \begin{bmatrix} C_{\mathcal{L}/\mathcal{B}}\left[\left(\hat{C}_{\mathcal{B}/\mathcal{T}}{}^{\mathcal{T}}\hat{\mathbf{v}}_B\right)^{\times}\right] & 0_{3\times3} & C_{\mathcal{L}/\mathcal{B}}C_{\mathcal{B}/\mathcal{T}} & 0_{3\times3} & 0_{3\times3} \end{bmatrix} \tag{59}$$

It is important to note that the least squares covariance is coordinatized in the LIDAR frame, so it must be transformed to the terrain frame to be applicable in the MEKF context according to

$${}^{\mathcal{T}}R_k = \hat{C}_{\mathcal{T}/\mathcal{B}}C_{\mathcal{B}/\mathcal{L}}{}^{\mathcal{L}}R_k C^T_{\mathcal{B}/\mathcal{L}}\hat{C}^T_{\mathcal{T}/\mathcal{B}} \tag{60}$$

The velocimeter LIDAR based TRN MEKF high-level algorithm is as follows. The filter states are propagated every time new IMU data is received using the state dynamics given by Eqns. 15, 28, and 29, respectively. The gyroscope and acclerometer bias values are not expected to change over time; therefore, they are not propagated along with the filter states. Next, each time a new point cloud is acquired, 3D features are extracted and corresponded to known locations (if available). The current point cloud is registered to the previous point cloud via the ICP algorithm and the batch state least squares process is performed to estimate the instantaneous translational and angular velocity. Then, the measurement sensitivity and covariance matrices are formed for each individual measurement model. They are all subsequently concatenated together to form the full measurement sensitivity and covariance matrices. Finally, the standard Kalman update process occurs and the procedure repeats.

IV. Experimental Methodology and Results

To evaluate the performance of the proposed velocimeter LIDAR based MEKF methodology, extensive experimental testing was performed at Texas A&M's Land, Air, and Space (LASR) laboratory. To emulate an EDL scenario, the velocimeter LIDAR was mounted on one of LASR lab's novel robotic platforms called Holonomic Omni-directional Motion Emulation Robot (HOMER) which was commanded to perform an approach trajectory towards a mock terrain as displayed in Fig. 3. HOMER is capable of simultaneous unrestricted motion in all planar directions. When paired with a Stewart platform, HOMER enables the emulation of almost any conceivable aerospace proximity operation including rendezvous, proximity operations, and docking (RPOD) and TRN applications.

For this experiment, HOMER was commanded to perform a near constant velocity translational trajectory with small angular perturbations. The true position and translational velocity data produced by Vicon Motion Capture system is displayed in Fig. 4. Note that these quantities are coordinatized in the terrain frame. The mock terrain is high-density foam which was precisely machined to mimic a terrain with various peaks, valleys, and ridges. In a real world scenario, the LIDAR would detect, classify, and correspond 3D terrain features itself. However, this work is focused on the development of new stochastic estimation techniques, so the feature generation and identification processes were forgone by using retro-reflective markers which could easily be mapped and detected.

Point cloud and Doppler velocity data is acquired by the velocimeter LIDAR at ~10 Hz and VectorNav VN-200 IMU data is received at ~50Hz. Although the 3D position, Doppler velocity, and point cloud registration updates can

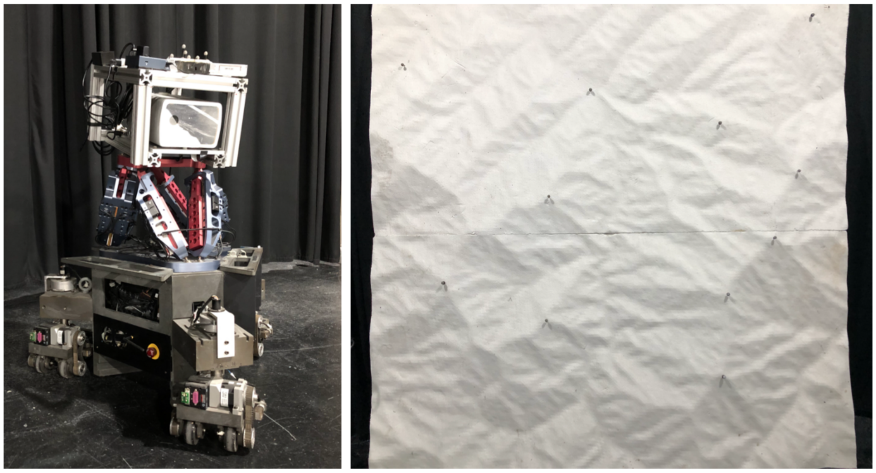

Fig. 3 (Left) Velocimeter LIDAR mounted on Stewart platform and HOMER (right) Machined high-density foam mock terrain

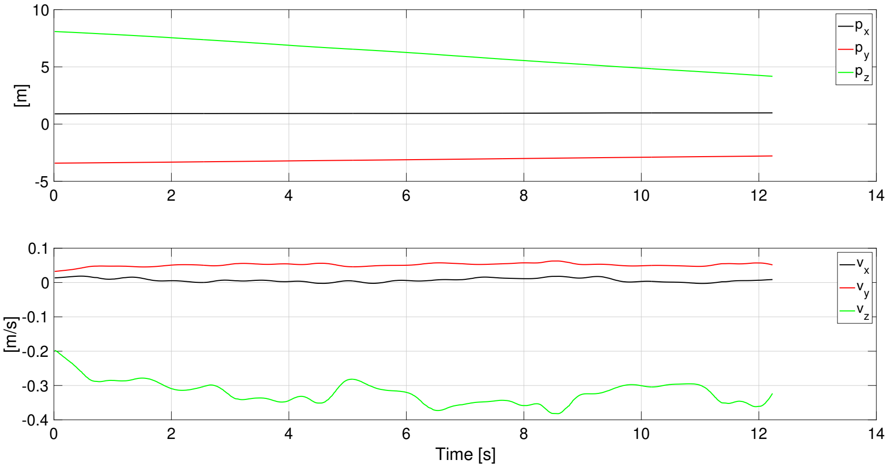

Fig. 4 True (top) position and (bottom) velocity trajectories produced by Vicon Motion Capture system

proceed upon initial LIDAR data return, the bulk velocity pseudo-measurements can only be used after the first F frames are processed. Experimentation with the batch state estimation algorithm revealed that choosing 3 frames-per-batch yielded the best results. Choosing more frames-per-batch results in estimates with higher confidence (lower covariance), however, the computational effort is significantly increased due to the back-filling process of the H matrix precluding real-time operation. For the batch state estimation, the weighting matrix, W, was simply chosen to be an appropriately sized identity matrix. The initial covariance matrix was set to a diagonal matrix with values according to Table 1, and the initial state estimates were set to be the true states plus a random number chosen from a zero-mean multi-dimensional normal distribution with variance equal to the initial covariance matrix. The noises for all measurement models are given in Table 2. Note that the measurement noises for ${}^{\mathcal{L}}\boldsymbol{t}_k$ and ${}^{\mathcal{L}}\boldsymbol{\omega}_k$ are denoted to be rough quantities in Table 2 because they are directly obtained from the normal equations during the batch state estimation process, but on average result in the denoted quantities. The continuous time process noise matrix was set to $Q = diag\left[\sigma_{gv},\ \sigma_{gu},\ \sigma_{av},\ \sigma_{au}\right]$ where the σ's are power spectral densities obtained via the VectorNav VN-200 data sheet.

Table 1 Initial 1σ state uncertainties for TRN MEKF

State	$q_{\mathcal{B}/\mathcal{T}}$	${}^{\mathcal{T}}\mathbf{p}_B$	${}^{\mathcal{T}}\mathbf{v}_B$	$\boldsymbol{\beta}_g$	$\boldsymbol{\beta}_a$
σ_0	25°	1 m	0.75 m/s	0.25 rad/s	0.5 m/s^2

Table 2 Measurement noises for TRN MEKF

Measurement	${}^{\mathcal{L}}\mathbf{p}_{k_i}$	$\Vert{}^{\mathcal{L}}\mathbf{p}_{k_i}\Vert v_{k_i}$	${}^{\mathcal{L}}\mathbf{t}_k$	$q_{\mathcal{L}_k/\mathcal{L}_{k-1}}$	${}^{\mathcal{L}}\mathbf{v}_k$	${}^{\mathcal{L}}\boldsymbol{\omega}_k$
σ_k	0.1 m	0.5 m$^2/s$	0.5 m	50°	0.5 m/s	0.5 m

Early results indicated that the measurements obtained directly from the point cloud registration process (${}^{\mathcal{L}}\mathbf{t}_k$ and $\mathbf{q}_{\mathcal{L}_k/\mathcal{L}_{k-1}}$) degrade estimates when using any of the other four measurement updates. This is most likely due to the high associated uncertainty or the poor quality of the registration results. However, during times when features cannot be identified or corresponded to *a priori* locations, using point cloud registration results as direct measurements can maintain attitude, position, and velocity estimates whereas an exclusively feature-based filter quickly diverges when features are lost. Further, it was found that during times when no features are available, solely using bulk velocity pseudo-measurements obtained from the batch least squares process performs better without fusing measurements from point cloud registration. Therefore, it is recommended that if Doppler velocity measurements are available, measurements from point cloud registration should not be used. On the other hand, using the results from point cloud registration can enhance a standard feature based MEKF (i.e. optical camera based or LIDAR with no Doppler velocity measurements) by maintaining estimates during times when features are lost.

Figs. 5-7 display the attitude, position, and velocity errors for three different variations of the velocimeter LIDAR0-based TRN MEKF. The first variation only uses 3D feature position (${}^{\mathcal{L}}\mathbf{p}_B$) measurements. The second variation uses both 3D feature position and position norm scaled feature Doppler velocity ($\Vert{}^{\mathcal{L}}\mathbf{p}_{k_i}\Vert v_{k_i}$) measurements. The third variation uses 3D feature position and position norm scaled feature Doppler velocity measurements, and bulk velocity pseudo-measurements (${}^{\mathcal{L}}\mathbf{v}_k$ and ${}^{\mathcal{L}}\boldsymbol{\omega}_k$) obtained from the batch state estimation process. All filter variations produce estimates with errors that are encapsulated by their respective 3σ bounds for the entirety of the time span, indicating well-tuned filters. All variations perform similarly with respect to attitude estimation. With respect to position and velocity, in general having Doppler velocity measurements for each feature enhances the confidence and precision of the filter when compared to having only position measurements. However, the lack of lateral acceleration causes biased estimates in the first and second variations exposing the need for more information during periods of constant velocity. The third variation fulfills this need and exhibits superior performance with respect to position and velocity by having the lowest state error and covariance in all directions.

Fig. 8 displays the average Mahalanobis distance, defined as $d_k = \sqrt{\Delta\mathbf{x}_k^T P_k^{-1} \Delta\mathbf{x}_k}$, for attitude, position, and velocity for all three filter variations over 1000 Monte Carlo runs. For each run, the initial conditions were varied to aid in the determination of each variation's consistency. All filter variations maintain reasonable values throughout the entire time span, indicating overall consistency. Filter variations 2 and 3 converge to approximately the same value, whereas

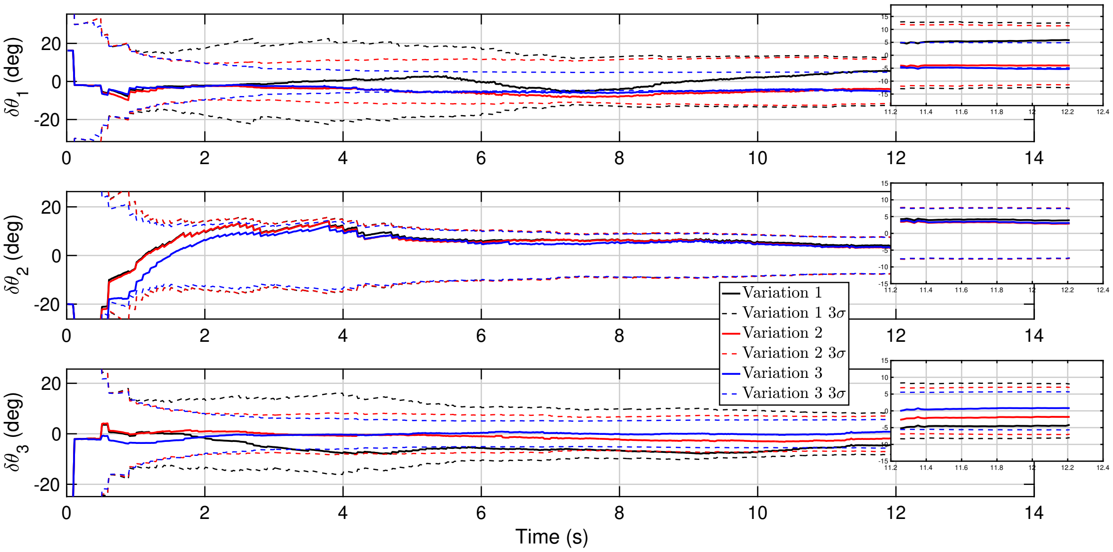

Fig. 5 Attitude estimation errors and 3 σ covariance for 1D linear translate

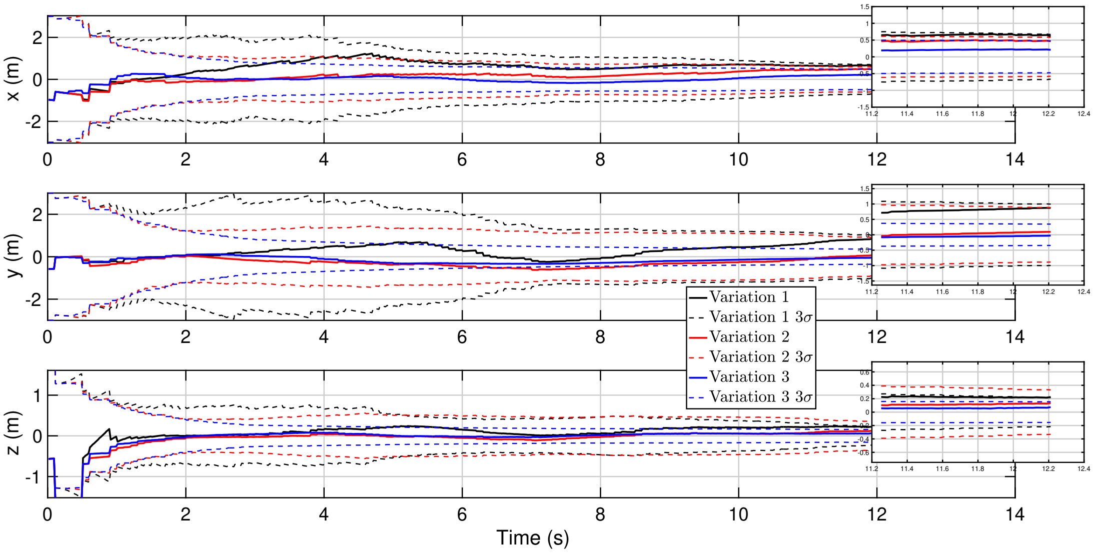

Fig. 6 Position estimation errors and 3 σ covariance for 1D linear translate

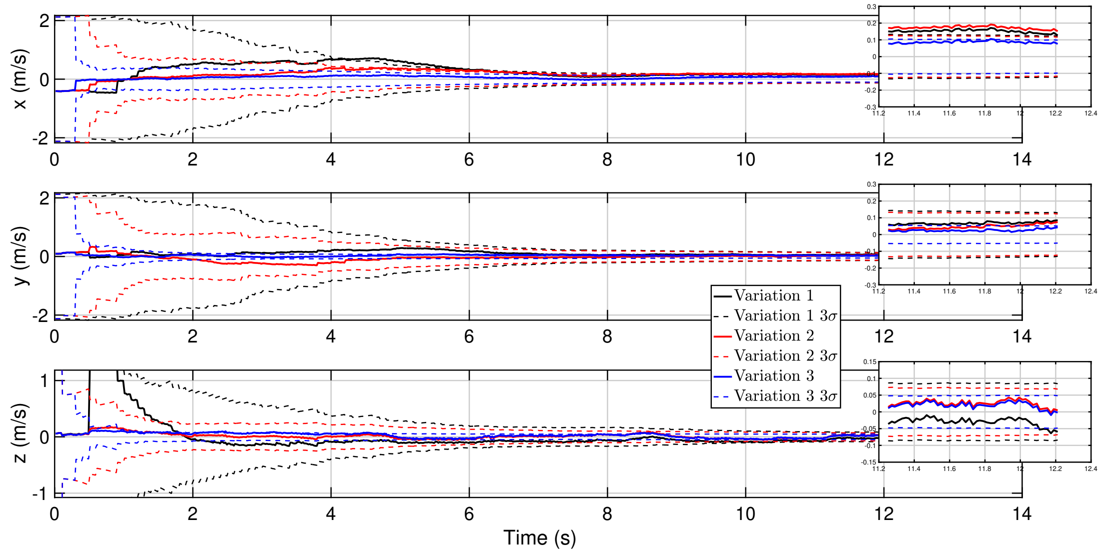

Fig. 7 Velocity estimation errors and 3 σ covariance for 1D linear translate

exclusively using feature position measurements yields a Mahalanobis distance of 1.5× that of the other filter variations.

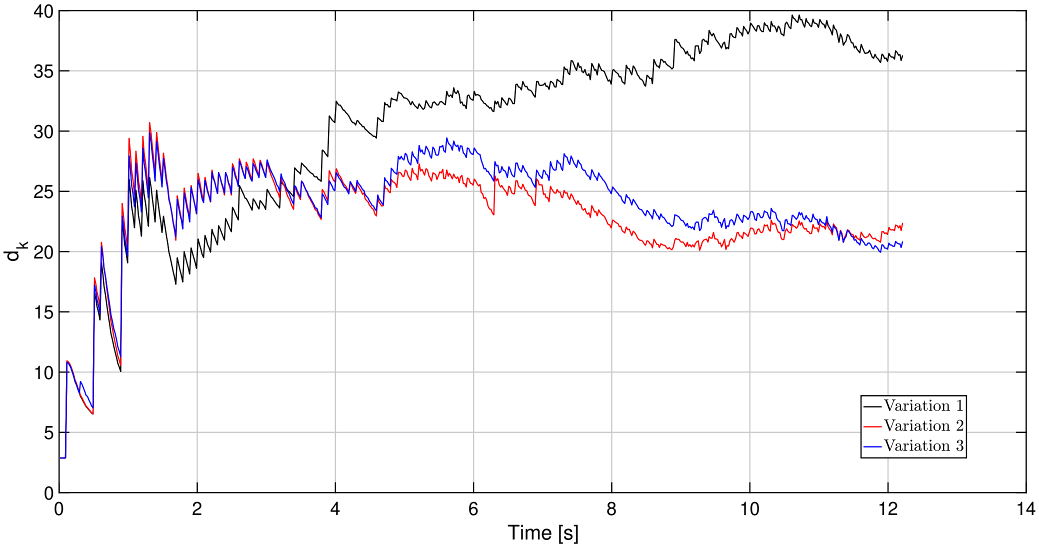

Fig. 8 Average Mahalanobis distance for attitude, position, and velocity over 1000 Monte Carlo Runs

V. Conclusion

A velocimeter LIDAR based MEKF for TRN applications was presented in this work. Four different measurement models were developed including a standard 3D corresponded position measurement, and 3 novel models corresponding to Doppler velocity measurements, point cloud registration based measurements, and bulk velocity pseudo-measurements obtained via a new batch state estimation approach. Hardware in the loop testing was performed to evaluate the proposed methodologies. Results indicated that simply adding the Doppler velocity measurement model to a standard 3D feature location based filter improves the accuracy and quality of the position and velocity estimates. Further, extremely accurate and confident velocity and position estimates can be achieved when using 3D feature position, feature Doppler velocity, and bulk velocity measurement models together. In situations where features cannot be detected or corresponded with *a priori* mapped landmarks, estimates of the relative change in position and orientation produced by point cloud registration can be used as direct measurements to maintain estimates for attitude, position and velocity. However, it was found that using point cloud registration results as measurements while also using 3D position measurements consistently resulted in degraded estimates. Moreover, if Doppler velocity measurements are available but features are lost, exclusively using the bulk velocity estimates yields better performance than when accompanying the bulk velocity measurements with point cloud registration measurements. Thus, it is recommended to only use the point cloud registration results as direct measurements when features are lost and there is no Doppler velocity measurement data available, i.e. when an optical camera or standard LIDAR is being used to generate feature measurements. Overall, the velocimeter LIDAR informed MEKF enables more accurate and confident state estimates when compared to a standard 3D feature based framework.

Acknowledgments

The authors acknowledge John Carson, Ron Sostaric, Ronney Lovelace, Teming Tse, and the EG Division at NASA Johnson Space Center (JSC) for partially supporting this work through a cooperative agreement between NASA JSC and Texas A&M. This research is partially supported by the Office of Naval Research (Grant No. N00014-19-1-2435). The authors acknowledge Drs. Brian Holm-Hansen and David Gonzales for their encouragement. The authors would also like to acknowledge Dr. John Junkins for providing the theoretical foundation and unparalleled insight to the proposed formulations.

References

[1] Trawny, N., Mourikis, A. I., Roumeliotis, S. I., Johnson, A. E., and Montgomery, J. F., "Vision-aided inertial navigation for pin-point landing using observations of mapped landmarks," *Journal of Field Robotics*, Vol. 24, No. 5, 2007, pp. 357–378. https://doi.org/10.1002/rob.20189.

[2] Wong, X. I., and Majji, M., "Extended Kalman Filter for Stereo Vision-Based Localization and Mapping Applications," *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, Vol. 140, No. 3, 2018, pp. 1–16. https://doi.org/10.1115/1.4037784.

[3] Epp, C. D., Robertson, E. A., and Brady, T., "Autonomous landing and hazard avoidance technology (ALHAT)," *IEEE Aerospace Conference Proceedings*, 2008. https://doi.org/10.1109/AERO.2008.4526297.

[4] Amzajerdian, F., Petway, L. B., Hines, G. D., Roback, V. E., Reisse, R. A., and Pierrottet, f. . U. i. . . j. . A. p. . . t. . L. y. . ., Diego F. doi = 10.2514/6.2013-5312, ????

[5] Johnson, A., Aaron, S., Chang, J., Cheng, Y., Montgomery, J., Mohan, S., Schroeder, S., Tweddle, B., Trawny, N., and Zheng, J., "the Lander Vision System for Mars 2020 Entry Descent and Landing," *Guidance, Navigation and Control 2017: Proceedings of the 40th Annual AAS Rocky Mountain Section Guidance and Control Conference*, 2017, pp. 143–158.

[6] Carson, J. M., Munk, M. M., Sostaric, R. R., Estes, J. N., Amzajerdian, F., Bryan Blair, J., Rutishauser, D. K., Restrepo, C. I., Cianciolo, A. D., Chen, G. T., and Tse, T., "The splice project: Continuing nasa development of gnc technologies for safe and precise landing," *AIAA Scitech 2019 Forum*, , No. January, 2019, pp. 1–9. https://doi.org/10.2514/6.2019-0660.

[7] Niemeyer, J., Rottensteiner, F., and Soergel, U., "Contextual classification of lidar data and building object detection in urban areas," *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 87, 2014, pp. 152–165. https://doi.org/10.1016/j.isprsjprs.2013.11.001, URL http://dx.doi.org/10.1016/j.isprsjprs.2013.11.001.

[8] Schwarz, B., "Lidar: Mapping the world in 3D," *Nature Photonics*, Vol. 4, No. 7, 2010, pp. 429–430. https://doi.org/10.1038/nphoton.2010.148.

[9] Zhang, J., and Singh, S., "LOAM: Lidar Odometry and Mapping in Real-time," , No. August, 2015. https://doi.org/10.15607/rss.2014.x.007.

[10] Toth, C., Grejner-Brzezinska, D. A., and Lee, Y. J., "Terrain-based navigation: Trajectory recovery from LiDAR Data," *Record - IEEE PLANS, Position Location and Navigation Symposium*, 2008, pp. 760–765. https://doi.org/10.1109/PLANS.2008.4570067.

[11] Hewitt, R. A., Setterfield, T. P., and Trawny, N., "LiDAR-Based Map Relative Localization Performance Analysis for Landing on Europa," *IEEE Aerospace Conference Proceedings*, Vol. 2021-March, 2021. https://doi.org/10.1109/AERO50100.2021.9438478.

[12] Amzajerdian, F., Pierrottet, D., Petway, L., Hines, G., and Roback, V., "Lidar systems for precision navigation and safe landing on planetary bodies," *International Symposium on Photoelectronic Detection and Imaging 2011: Laser Sensing and Imaging; and Biological and Medical Applications of Photonics Sensing and Imaging*, Vol. 8192, No. August 2011, 2011, p. 819202. https://doi.org/10.1117/12.904062.

[13] Adams, D., Majji, M., Stevens, S. U., Kulkarni, T., Katake, A., San Martin, A., and Skulsky, E., "Velocimeter LIDAR Based Bulk Velocity Estimation for Terrain Relative Navigation Applications," *AIAA Scitech 2022 Forum*, 2022.

[14] Marden, S., and Guivant, J., "Improving the Performance of ICP for Real-Time Applications using an Approximate Nearest Neighbour Search," *Proceedings of Australasian Conference on Robotics and Automation*, 2012.

[15] Besl, P., and McKay, N. D., "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 2, 1992, pp. 239–256. https://doi.org/10.1109/34.121791.

[16] Fantoni, S., Castellani, U., and Fusiello, A., "Accurate and Automatic Alignment of Range Surfaces," *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization Transmission*, 2012, pp. 73–80. https://doi.org/10.1109/3DIMPVT.2012.63.

[17] Segal, A., Haehnel, D., and Thrun, S., "Generalized-ICP," *Proceedings of Robotics: Science and Systems*, Seattle, USA, 2009. https://doi.org/10.15607/RSS.2009.V.021.

[18] Rueckert, D., Sonoda, L., Hayes, C., Hill, D., Leach, M., and Hawkes, D., "Nonrigid registration using free-form deformations: application to breast MR images," *IEEE Transactions on Medical Imaging*, Vol. 18, No. 8, 1999, pp. 712–721. https://doi.org/10.1109/42.796284.

[19] Hu, Y., Miyashita, L., Watanabe, Y., and Ishikawa, M., "Robust 6-DOF motion sensing for an arbitrary rigid body by multi-view laser Doppler measurements," *Optics Express*, Vol. 25, No. 24, 2017. https://doi.org/10.1364/oe.25.030371.

[20] Schaub, H., and Junkins, J. L., *Analytical Mechanics of Space Systems*, 3rd ed., AIAA Education Series, Reston, VA, 2014. https://doi.org/10.2514/4.102400.

[21] Crassidis, J. L., and Junkins, J. L., *Optimal estimation of dynamic systems*, CRC press, 2011.

Here's a library of all the typeset math. My intention right now is to have it all here and then we can copy/paste into where it needs to go. I'll be trying to organize sets of equations into appropriate groups.

VI. 5.1: Motion Models

A. Rotational Motion

Attitude Kinematics:

$$\dot{\mathbf{q}}_{\mathcal{B}/\mathcal{T}} = \frac{1}{2}\Omega\left({}^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{T}}\right)\mathbf{q}_{\mathcal{B}/\mathcal{T}}; \qquad \Omega\left(\omega\right) = \begin{bmatrix} -\left[\omega^{\times}\right] & \omega \\ -\omega^{T} & 0 \end{bmatrix} \tag{61}$$

$$= \frac{1}{2}{}^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{T}} \otimes \mathbf{q}_{\mathcal{B}/\mathcal{T}} \tag{62}$$

$$= \frac{1}{2}\Theta\left(\mathbf{q}\right)\omega; \qquad \Theta\left(\mathbf{q}\right) = \begin{bmatrix} q_4\mathbb{I}_3 + \left[\mathbf{q}_v^{\times}\right] \\ -\mathbf{q}_v^{T} \end{bmatrix} \tag{63}$$

$$note: \ \left[\mathbf{q}\otimes\right] = \begin{bmatrix} -\left[\mathbf{q}_v^{\times}\right] + q_0\mathbb{I}_3 & \mathbf{q}_v \\ -\mathbf{q}_v^{T} & q_0 \end{bmatrix} \ if \ \mathbf{q} = \begin{bmatrix} \mathbf{q_v} \\ q_0 \end{bmatrix} \tag{64}$$

Error Quaternion:

$$\boldsymbol{\delta}\mathbf{q} = \mathbf{q} \otimes \hat{\mathbf{q}}^{-1} \tag{65}$$

$$\boldsymbol{\delta}\dot{\mathbf{q}}_v = -\hat{\omega} \times \boldsymbol{\delta}\mathbf{q}_v + \frac{1}{2}\boldsymbol{\delta}\omega \tag{66}$$

$$\boldsymbol{\delta}\dot{\mathbf{q}}_0 = 0 \tag{67}$$

Euler's Equations:

$$
\begin{aligned}
{}^{\mathcal{B}}\dot{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{T}} &= -J^{-1}\left[{}^{\mathcal{B}}\boldsymbol{\omega}^{\times}_{\mathcal{B}/\mathcal{T}}\right] J\,{}^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{T}} + N\left(t\right) && (68)\\
\tilde{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{T}} &= \boldsymbol{\omega}_{\mathcal{B}/\mathcal{T}} + \boldsymbol{\beta}_g + \boldsymbol{\eta}_{g_v}; \qquad \dot{\boldsymbol{\beta}}_g = \boldsymbol{\eta}_{g_u} && (69)\\
\boldsymbol{\omega}_{\mathcal{B}/\mathcal{T}} &= \tilde{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{T}} - \boldsymbol{\beta}_g - \boldsymbol{\eta}_{g_v}; \qquad \hat{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{T}} = \tilde{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{T}} - \hat{\boldsymbol{\beta}}_g && (70)\\
\boldsymbol{\delta\omega} &= \boldsymbol{\omega}_{\mathcal{B}/\mathcal{T}} - \hat{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{T}} = -\left(\Delta\boldsymbol{\beta}_g + \eta_{g_v}\right) && (71)\\
\boldsymbol{\delta}\mathbf{q}_v &= \hat{\omega}\times\boldsymbol{\delta}\mathbf{q}_v - \frac{1}{2}\left(\Delta\boldsymbol{\beta}_g + \boldsymbol{\eta}_{g_v}\right) && (72)\\
\boldsymbol{\delta}\dot{\boldsymbol{\alpha}} &= -\hat{\omega}\times\boldsymbol{\delta\alpha} - \Delta\boldsymbol{\beta}_g - \boldsymbol{\eta}_{g_v} && (73)\\
&= -\left[\left(\tilde{\omega} - \hat{\beta}_g\right)^{\times}\right]\boldsymbol{\delta\alpha} - \Delta\boldsymbol{\beta}_g - \boldsymbol{\eta}_{g_v} && (74)
\end{aligned}
$$

B. Translational Motion

Kinematics:

$$
\begin{aligned}
{}^{\mathcal{T}}\dot{\mathbf{p}}_B &= {}^{\mathcal{T}}\mathbf{v}_B && (75)\\
{}^{\mathcal{T}}\dot{\mathbf{v}}_B &= C_{\mathcal{T}/\mathcal{B}}\,{}^{\mathcal{B}}\mathbf{a}_{\mathcal{B}/\mathcal{T}} + {}^{\mathcal{T}}\mathbf{g} && (76)\\
{}^{\mathcal{B}}\mathbf{a} &= C_{\mathcal{B}/\mathcal{T}}\left[{}^{\mathcal{T}}\mathbf{a}_B - {}^{\mathcal{T}}\mathbf{g}\right] && (77)\\
{}^{\mathcal{T}}\hat{\mathbf{a}}_B &= \hat{C}^T_{\mathcal{B}/\mathcal{T}}\left[{}^{\mathcal{B}}\tilde{\mathbf{a}}_B - \hat{\boldsymbol{\beta}}_a\right] - {}^{\mathcal{T}}\mathbf{g} && (78)
\end{aligned}
$$

Error Models:

$$
\begin{aligned}
\Delta{}^{\mathcal{T}}\dot{\mathbf{p}}_B &= \Delta{}^{\mathcal{T}}\mathbf{v}_B && (79)\\
{}^{\mathcal{B}}\tilde{\mathbf{a}} &= C_{\mathcal{B}/\mathcal{T}}\left[{}^{\mathcal{T}}\mathbf{a}_B - {}^{\mathcal{T}}\mathbf{g}\right] + \boldsymbol{\beta}_a + \boldsymbol{\eta}_{a_v} && (80)\\
{}^{\mathcal{B}}\tilde{\mathbf{a}} - \boldsymbol{\beta}_a - \boldsymbol{\eta}_{a_v} &= C_{\mathcal{B}/\mathcal{T}}{}^{\mathcal{T}}\mathbf{a}_B - C_{\mathcal{B}/\mathcal{T}}{}^{\mathcal{T}}\mathbf{g} && (81)\\
{}^{\mathcal{T}}\mathbf{a}_B &= C^T_{\mathcal{B}/\mathcal{T}}\left[{}^{\mathcal{B}}\tilde{\mathbf{a}} - \boldsymbol{\beta}_a - \boldsymbol{\eta}_{a_v} + C_{\mathcal{B}/\mathcal{T}}{}^{\mathcal{T}}\mathbf{g}\right] && (82)\\
&= C^T_{\mathcal{B}/\mathcal{T}}{}^{\mathcal{B}}\tilde{\mathbf{a}} - C^T_{\mathcal{B}/\mathcal{T}}\boldsymbol{\beta}_a - C^T_{\mathcal{B}/\mathcal{T}}\boldsymbol{\eta}_{a_v} + {}^{\mathcal{T}}\mathbf{g} && (83)\\
&= C^T_{\mathcal{B}/\mathcal{T}}\left[{}^{\mathcal{B}}\tilde{\mathbf{a}} - \boldsymbol{\beta}_a\right] + {}^{\mathcal{T}}\mathbf{g} - C^T_{\mathcal{B}/\mathcal{T}}\boldsymbol{\eta}_{a_v} && (84)\\
&= \hat{C}^T_{\mathcal{B}/\mathcal{T}}\left[\mathbb{I} + \left[\boldsymbol{\delta\alpha}^{\times}\right]\right]\left[{}^{\mathcal{B}}\tilde{\mathbf{a}} - \Delta\boldsymbol{\beta}_a - \hat{\boldsymbol{\beta}}_a\right] + {}^{\mathcal{T}}\mathbf{g} - \left[\mathbb{I} + \left[\boldsymbol{\delta\alpha}^{\times}\right]\right]\boldsymbol{\eta}_{a_v} && (85)\\
&= \left[\hat{C}^T_{\mathcal{B}/\mathcal{T}} + \hat{C}^T_{\mathcal{B}/\mathcal{T}}\left[\boldsymbol{\delta\alpha}^{\times}\right]\right]\left[{}^{\mathcal{B}}\tilde{\mathbf{a}} - \Delta\boldsymbol{\beta}_a - \hat{\boldsymbol{\beta}}_a\right] + {}^{\mathcal{T}}\mathbf{g} - \left[\hat{C}_{\mathcal{B}/\mathcal{T}} + \hat{C}_{\mathcal{B}/\mathcal{T}}\left[\boldsymbol{\delta\alpha}^{\times}\right]\right]\boldsymbol{\eta}_{a_v} && (86)\\
\therefore\ \Delta{}^{\mathcal{T}}\dot{\mathbf{v}}_B &= -\hat{C}^T_{\mathcal{B}/\mathcal{T}}\left[\Delta\boldsymbol{\beta}_a + \boldsymbol{\eta}_{a_v} + \left[\left({}^{\mathcal{B}}\tilde{\mathbf{a}} - \hat{\boldsymbol{\beta}}_a\right)^{\times}\right]\boldsymbol{\delta\alpha}\right] && (87)\\
& && (88)
\end{aligned}
$$

Complete Dynamics:

$$
\dot{\mathbf{x}} = \begin{bmatrix} \frac{1}{2}\Omega\left({}^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{T}}\right)\mathbf{q}_{\mathcal{B}/\mathcal{T}} \\ {}^{\mathcal{T}}\mathbf{v}_B \\ C_{\mathcal{T}/\mathcal{B}}\,{}^{\mathcal{B}}\mathbf{a}_{\mathcal{B}/\mathcal{T}} + {}^{\mathcal{T}}\mathbf{g} \\ \boldsymbol{\eta}_{g_u} \\ \boldsymbol{\eta}_{a_u} \end{bmatrix} \tag{89}
$$

$$\tag{90}$$

$$
\begin{aligned}
\Delta\dot{\mathbf{x}} &= \begin{bmatrix} -\left[\hat{\omega}^{\times}\right]\boldsymbol{\delta\alpha} - \Delta\boldsymbol{\beta}_g - \boldsymbol{\eta}_{g_v} \\ \Delta^{\mathcal{T}}\mathbf{v}_B \\ -\hat{C}_{\mathcal{B}/\mathcal{T}}\left[\hat{\mathbf{a}}^{\times}\right]\boldsymbol{\delta\alpha} - \hat{C}^T_{\mathcal{B}/\mathcal{T}}\Delta\boldsymbol{\beta}_a - \hat{C}^T_{\mathcal{B}/\mathcal{T}}\boldsymbol{\eta}_{a_v} \\ \boldsymbol{\eta}_{g_u} \\ \boldsymbol{\eta}_{a_u} \end{bmatrix} && (91) \\
&= F\Delta\mathbf{x} + G\mathbf{w} && (92)
\end{aligned}
$$

where:

$$
\begin{aligned}
F &= \begin{bmatrix} -\left[\hat{\omega}^{\times}\right] & 0_{3\times3} & 0_{3\times3} & -\mathbb{I}_3 & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & \mathbb{I}_3 & 0_{3\times3} & 0_{3\times3} \\ -\hat{C}^T_{\mathcal{B}/\mathcal{T}}\left[\hat{\mathbf{a}}^{\times}\right] & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} - \hat{C}^T_{\mathcal{B}/\mathcal{T}} & \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \end{bmatrix} && (93) \\
G &= \begin{bmatrix} -\mathbb{I}_3 & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & -\hat{C}^T_{\mathcal{B}/\mathcal{T}} & 0_{3\times3} \\ 0_{3\times3} & \mathbb{I}_3 & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & \mathbb{I}_3 \end{bmatrix} && (94) \\
\mathbf{w} &= \begin{bmatrix} \boldsymbol{\eta}_{g_v} \\ \boldsymbol{\eta}_{g_u} \\ \boldsymbol{\eta}_{a_v} \\ \boldsymbol{\eta}_{a_u} \end{bmatrix} && (95)
\end{aligned}
$$

VII. Measurement Models

A. Feature Measurement

$$
\begin{aligned}
{}^{\mathcal{L}}\mathbf{p}_{k_i} &= C_{\mathcal{L}/\mathcal{B}}\left(C_{\mathcal{B}/\mathcal{T}}\left({}^{\mathcal{T}}\mathbf{p}_i - {}^{\mathcal{T}}\mathbf{p}_B\right) - {}^{\mathcal{B}}\boldsymbol{\rho}_L\right) && (96) \\
&= C_{\mathcal{L}/\mathcal{B}}\left(\left[\mathbb{I} - \left[\boldsymbol{\delta\alpha}^{\times}\right]\right]\hat{C}_{\mathcal{B}/\mathcal{T}}\left({}^{\mathcal{T}}\mathbf{p}_i - {}^{\mathcal{T}}\mathbf{p}_B\right) - {}^{\mathcal{B}}\boldsymbol{\rho}_L\right) && (97) \\
&= C_{\mathcal{L}/\mathcal{B}}\left(\left[\mathbb{I} - \left[\boldsymbol{\delta\alpha}^{\times}\right]\right]\hat{C}_{\mathcal{B}/\mathcal{T}}\left({}^{\mathcal{T}}\mathbf{p}_i - {}^{\mathcal{T}}\hat{\mathbf{p}}_B - \Delta\tilde{\mathbf{p}}_B\right) - {}^{\mathcal{B}}\boldsymbol{\rho}_L\right) && (98) \\
&= C_{\mathcal{L}/\mathcal{B}}\left(\hat{C}_{\mathcal{B}/\mathcal{T}}\left({}^{\mathcal{T}}\mathbf{p}_i - {}^{\mathcal{T}}\hat{\mathbf{p}}_B - \Delta\tilde{\mathbf{p}}_B\right) - \left[\boldsymbol{\delta\alpha}^{\times}\right]\hat{C}_{\mathcal{B}/\mathcal{T}}\left({}^{\mathcal{T}}\mathbf{p}_i - {}^{\mathcal{T}}\hat{\mathbf{p}}_B\right) - {}^{\mathcal{B}}\boldsymbol{\rho}_L\right) && (99) \\
\Delta{}^{\mathcal{L}}\mathbf{p}_{k_i} &= {}^{\mathcal{L}}\mathbf{p}_{k_i} - {}^{\mathcal{L}}\hat{\mathbf{p}}_{k_i} && (100) \\
&= C_{\mathcal{L}/\mathcal{B}}\left(\hat{C}_{\mathcal{B}/\mathcal{T}}\left({}^{\mathcal{T}}\mathbf{p}_i - {}^{\mathcal{T}}\hat{\mathbf{p}}_B - \Delta\tilde{\mathbf{p}}_B\right) - \left[\boldsymbol{\delta\alpha}^{\times}\right]\hat{C}_{\mathcal{B}/\mathcal{T}}\left({}^{\mathcal{T}}\mathbf{p}_i - {}^{\mathcal{T}}\hat{\mathbf{p}}_B\right) - {}^{\mathcal{B}}\boldsymbol{\rho}_L\right) && (101) \\
\Delta{}^{\mathcal{L}}\mathbf{p}_{k_i} &= {}^{\mathcal{L}}\mathbf{p}_{k_i} - {}^{\mathcal{L}}\hat{\mathbf{p}}_{k_i} - C_{\mathcal{L}/\mathcal{B}}\left(\hat{C}_{\mathcal{B}/\mathcal{T}}\left({}^{\mathcal{T}}\mathbf{p}_i - {}^{\mathcal{T}}\hat{\mathbf{p}}_B\right) - {}^{\mathcal{B}}\boldsymbol{\rho}_L\right) && (102) \\
&= -C_{\mathcal{L}/\mathcal{B}}\hat{C}_{\mathcal{B}/\mathcal{T}}\Delta^{\mathcal{T}}\mathbf{p}_B + C_{\mathcal{L}/\mathcal{B}}\left[\left(\hat{C}_{\mathcal{B}/\mathcal{T}}\left({}^{\mathcal{T}}\mathbf{p}_i - {}^{\mathcal{T}}\hat{\mathbf{p}}_B\right)\right)^{\times}\right]\boldsymbol{\delta\alpha} && (103)
\end{aligned}
$$

B. Feature Velocity Measurement (5.2.2 after OR)

$$v_{k_i} = \frac{1}{\|{}^{\mathcal{L}}\mathbf{p}_{k_i}\|}\left({}^{\mathcal{T}}\mathbf{p}_{k_i}^{T}\,{}^{\mathcal{T}}\mathbf{v}_{k_i}\right) \tag{104}$$

$${}^{\mathcal{T}}\mathbf{p}_{k_i} = {}^{\mathcal{T}}\mathbf{p}_i - {}^{\mathcal{T}}\mathbf{p}_B - C_{\mathcal{T}/\mathcal{B}}\,{}^{\mathcal{B}}\boldsymbol{\rho}_L \tag{105}$$

$$= {}^{\mathcal{T}}\mathbf{p}_i - {}^{\mathcal{T}}\mathbf{p}_B - C_{\mathcal{B}/\mathcal{T}}^{T}\,{}^{\mathcal{B}}\boldsymbol{\rho}_L \tag{106}$$

$${}^{\mathcal{T}}\mathbf{v}_{k_i} = {}^{\mathcal{T}}\dot{\mathbf{p}}_i - {}^{\mathcal{T}}\dot{\mathbf{p}}_B - \left(C_{\mathcal{B}/\mathcal{T}}^{T}\left[{}^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{T}}^{\times}\right]{}^{\mathcal{B}}\boldsymbol{\rho}_L\right) \tag{107}$$

$$= -{}^{\mathcal{T}}\dot{\mathbf{p}}_B + C_{\mathcal{B}/\mathcal{T}}^{T}\left[{}^{\mathcal{B}}\boldsymbol{\rho}_L^{\times}\right]{}^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{T}} \tag{108}$$

$$\tilde{y} = \|{}^{\mathcal{L}}\mathbf{p}_{k_i}\|v_{k_i} = {}^{\mathcal{T}}\mathbf{p}_{k_i}^{T}\,{}^{\mathcal{T}}\mathbf{v}_{k_i} = \left({}^{\mathcal{T}}\mathbf{p}_i^{T} - {}^{\mathcal{T}}\mathbf{p}_B^{T} - {}^{\mathcal{B}}\mathbf{p}_L^{T}TB\right)\left(-{}^{\mathcal{T}}\dot{\mathbf{p}}_B + C_{\mathcal{B}/\mathcal{T}}^{T}\left[{}^{\mathcal{B}}\boldsymbol{\rho}_L^{\times}\right]{}^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{T}}\right) \tag{109}$$

$$= -{}^{\mathcal{T}}\mathbf{p}_i^{T}\,{}^{\mathcal{T}}\dot{\mathbf{p}}_B + {}^{\mathcal{T}}\mathbf{p}_i^{T}\,C_{\mathcal{B}/\mathcal{T}}^{T}\left[{}^{\mathcal{B}}\boldsymbol{\rho}_L^{\times}\right]{}^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{T}} + {}^{\mathcal{T}}\mathbf{p}_B^{T}\,{}^{\mathcal{T}}\dot{\mathbf{p}}_B - {}^{\mathcal{T}}\mathbf{p}_B^{T}C_{\mathcal{B}/\mathcal{T}}^{T}\left[{}^{\mathcal{B}}\boldsymbol{\rho}_L^{\times}\right]{}^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{T}} + {}^{\mathcal{B}}\mathbf{p}_L^{T}TB\,{}^{\mathcal{T}}\dot{\mathbf{p}}_B - {}^{\mathcal{B}}\mathbf{p}_L^{T}TBC_{\mathcal{B}/\mathcal{T}}^{T}\,[{}^{\mathcal{B}}\boldsymbol{\rho} \tag{110}$$

$$= -{}^{\mathcal{T}}\mathbf{p}_i^{T}\,{}^{\mathcal{T}}\dot{\mathbf{p}}_B + {}^{\mathcal{T}}\mathbf{p}_i^{T}\,C_{\mathcal{B}/\mathcal{T}}^{T}\left[{}^{\mathcal{B}}\boldsymbol{\rho}_L^{\times}\right]{}^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{T}} + {}^{\mathcal{T}}\mathbf{p}_B^{T}\,{}^{\mathcal{T}}\dot{\mathbf{p}}_B - {}^{\mathcal{T}}\mathbf{p}_B^{T}C_{\mathcal{B}/\mathcal{T}}^{T}\left[{}^{\mathcal{B}}\boldsymbol{\rho}_L^{\times}\right]{}^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{T}} + {}^{\mathcal{B}}\mathbf{p}_L^{T}TB\,{}^{\mathcal{T}}\dot{\mathbf{p}}_B \tag{111}$$

$$y\left(\mathbf{x}\right) = y\left(\hat{\mathbf{x}} + \Delta\mathbf{x}\right) = -{}^{\mathcal{T}}\mathbf{p}_i^{T}\left({}^{\mathcal{T}}\hat{\dot{\mathbf{p}}}_B + \Delta{}^{\mathcal{T}}\dot{\mathbf{p}}_B\right) + {}^{\mathcal{T}}\mathbf{p}_i^{T}\left(\mathbb{I} + \left[\boldsymbol{\delta\alpha}^{\times}\right]\right)\hat{C}_{\mathcal{B}/\mathcal{T}}^{T}\left[{}^{\mathcal{B}}\boldsymbol{\rho}_L^{\times}\right]\left({}^{\mathcal{B}}\hat{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{T}} + \delta{}^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{T}}\right) + \left({}^{\mathcal{T}}\hat{\mathbf{p}}_B^{T} + \Delta{}^{\mathcal{T}}\mathbf{p}_B^{T}\right)\left({}^{\mathcal{T}}\hat{\dot{\mathbf{p}}}_B\Delta{}^{\mathcal{T}}\dot{\mathbf{p}}_B\right) - \Big({}^{\mathcal{T}}\hat{\mathbf{p}} \tag{112}$$

$$= -{}^{\mathcal{T}}\mathbf{p}_i^{T}\,{}^{\mathcal{T}}\hat{\dot{\mathbf{p}}}_B - {}^{\mathcal{T}}\mathbf{p}_i^{T}\,\Delta{}^{\mathcal{T}}\dot{\mathbf{p}}_B + {}^{\mathcal{T}}\mathbf{p}_i^{T}\,\hat{C}_{\mathcal{B}/\mathcal{T}}^{T}\left[{}^{\mathcal{B}}\boldsymbol{\rho}_L^{\times}\right]{}^{\mathcal{B}}\hat{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{T}} + {}^{\mathcal{T}}\mathbf{p}_i^{T}\,\hat{C}_{\mathcal{B}/\mathcal{T}}^{T}\left[{}^{\mathcal{B}}\boldsymbol{\rho}_L^{\times}\right]\delta{}^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{T}} + {}^{\mathcal{T}}\mathbf{p}_i^{T}\left[\boldsymbol{\delta\alpha}^{\times}\right]\hat{C}_{\mathcal{B}/\mathcal{T}}^{T}\left[{}^{\mathcal{B}}\boldsymbol{\rho}_L^{\times}\right]{}^{\mathcal{B}}\hat{\boldsymbol{\omega}}_{\mathcal{B}} \tag{113}$$

$$= -{}^{\mathcal{T}}\mathbf{p}_i^{T}\,\Delta{}^{\mathcal{T}}\dot{\mathbf{p}}_B + {}^{\mathcal{T}}\mathbf{p}_i^{T}\,\hat{C}_{\mathcal{B}/\mathcal{T}}^{T}\left[{}^{\mathcal{B}}\boldsymbol{\rho}_L^{\times}\right]\delta{}^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{T}} - {}^{\mathcal{T}}\mathbf{p}_i^{T}\left[\left(\hat{C}_{\mathcal{B}/\mathcal{T}}^{T}\left[{}^{\mathcal{B}}\boldsymbol{\rho}_L^{\times}\right]{}^{\mathcal{B}}\hat{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{T}}\right)^{\times}\right]\boldsymbol{\delta\alpha} + {}^{\mathcal{T}}\hat{\mathbf{p}}_B^{T}\,{}^{\mathcal{T}}\hat{\dot{\mathbf{p}}}_B + {}^{\mathcal{T}}\hat{\mathbf{p}}_B^{T}\Delta{}^{\mathcal{T}}\dot{\mathbf{p}}_B - {}^{\mathcal{T}}\hat{\mathbf{p}}_B^{T}\hat{C}^{T}_{\mathcal{B}} \tag{114}$$

$$\therefore H_{v_{\delta\alpha}} = \left({}^{\mathcal{T}}\hat{\mathbf{p}}_B^{T} - {}^{\mathcal{T}}\mathbf{p}_i^{T}\right)\left[\left(\hat{C}_{\mathcal{B}/\mathcal{T}}^{T}\left[{}^{\mathcal{B}}\boldsymbol{\rho}_L^{\times}\right]{}^{\mathcal{B}}\hat{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{T}}\right)^{\times}\right] + {}^{\mathcal{B}}\mathbf{p}_L^{T}\left[\left(\hat{C}_{\mathcal{B}/\mathcal{T}}\,{}^{\mathcal{T}}\hat{\dot{\mathbf{p}}}_B\right)^{\times}\right] \tag{115}$$

$$H_{v_{\Delta{}^{\mathcal{T}}\mathbf{p}_B}} = {}^{\mathcal{T}}\hat{\dot{\mathbf{p}}}_B^{T} - {}^{\mathcal{B}}\hat{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{T}}^{T}\left[{}^{\mathcal{B}}\boldsymbol{\rho}_L^{\times}\right]^{T}\hat{C}_{\mathcal{B}/\mathcal{T}} \tag{116}$$

$$H_{v_{\Delta{}^{\mathcal{T}}\dot{\mathbf{p}}_B}} = -{}^{\mathcal{T}}\mathbf{p}_i^{T} + {}^{\mathcal{T}}\hat{\mathbf{p}}_B^{T} + {}^{\mathcal{B}}\mathbf{p}_L^{T}\hat{C}_{\mathcal{B}/\mathcal{T}} \tag{117}$$

$$H_{v_{\Delta\boldsymbol{\beta}_g}} = \left({}^{\mathcal{T}}\hat{\mathbf{p}}_B^{T} - {}^{\mathcal{T}}\mathbf{p}_i^{T}\right)\hat{C}_{\mathcal{B}/\mathcal{T}}^{T}\left[{}^{\mathcal{B}}\boldsymbol{\rho}_L^{\times}\right]; \qquad \boldsymbol{\delta\omega} = -\left(\Delta\boldsymbol{\beta}_g + \boldsymbol{\eta}_{g_v}\right) \tag{118}$$

VIII. 5.3 Point Cloud Registration

A. 5.3.1 Change in Position

$${}^{\mathcal{L}}\mathbf{t}_k = {}^{\mathcal{L}}\mathbf{p}_{B_{k+1}} - {}^{\mathcal{L}}\mathbf{p}_{B_k} \tag{119}$$

$$= C_{\mathcal{L}/\mathcal{B}}C_{\mathcal{B}/\mathcal{T}}\left({}^{\mathcal{T}}\mathbf{p}_{B_{k+1}} - {}^{\mathcal{T}}\mathbf{p}_{B_k}\right) \tag{120}$$

$$\therefore \Delta{}^{\mathcal{L}}\mathbf{t}_k = {}^{\mathcal{L}}\mathbf{t}_k - {}^{\mathcal{L}}\hat{\mathbf{t}}_k \tag{121}$$

$$= C_{\mathcal{L}/\mathcal{B}}\left(\mathbb{I} - \left[\boldsymbol{\delta\alpha}^{\times}\right]\right)\hat{C}_{\mathcal{B}/\mathcal{T}}\left({}^{\mathcal{T}}\hat{\mathbf{p}}_{B_{k+1}} + \Delta{}^{\mathcal{T}}\mathbf{p}_{B_{k+1}} - {}^{\mathcal{T}}\mathbf{p}_{B_k}\right) - C_{\mathcal{L}/\mathcal{B}}\hat{C}_{\mathcal{B}/\mathcal{T}}\left({}^{\mathcal{T}}\hat{\mathbf{p}}_{B_{k+1}} - {}^{\mathcal{T}}\hat{\mathbf{p}}_{B_k}\right) \tag{122}$$

$$= C_{\mathcal{L}/\mathcal{B}}\hat{C}_{\mathcal{B}/\mathcal{T}}\Delta{}^{\mathcal{T}}\mathbf{p}_{B_{k+1}} + \left[\left(\hat{C}_{\mathcal{B}/\mathcal{T}}\left({}^{\mathcal{T}}\hat{\mathbf{p}}_{B_{k+1}} - {}^{\mathcal{T}}\hat{\mathbf{p}}_{B_k}\right)\right)^{\times}\right]\boldsymbol{\delta\alpha} \tag{123}$$

B. 5.3.2 Change in Orientation

$$
\begin{aligned}
\mathbf{q}_{\mathcal{L}_k/\mathcal{L}_{k-1}} &= \mathbf{q}_{\mathcal{L}/\mathcal{B}} \otimes \mathbf{q}_{\mathcal{B}_k/\mathcal{B}_{k-1}} \otimes \mathbf{q}_{\mathcal{L}/\mathcal{B}}^{-1} && (124)\\
&= \mathbf{q}_{\mathcal{L}/\mathcal{B}} \otimes \mathbf{q}_{\mathcal{B}_{\|+\infty}/\mathcal{T}} \otimes \mathbf{q}_{\mathcal{B}_{\|}/\mathcal{T}}^{-1} \otimes \mathbf{q}_{\mathcal{L}/\mathcal{B}}^{-1} && (125)\\
&= \mathbf{q}_{\mathcal{L}/\mathcal{B}} \otimes \begin{bmatrix} \boldsymbol{\delta\rho} \\ \boldsymbol{\delta}\mathbf{q}_4 \end{bmatrix} \otimes \hat{\mathbf{q}}_{\mathcal{B}_{\|+\infty}/\mathcal{T}} \otimes \mathbf{q}_{\mathcal{B}_{\|}/\mathcal{T}}^{-1} \otimes \mathbf{q}_{\mathcal{L}/\mathcal{B}}^{-1} && (126)\\
so\ \boldsymbol{\delta}\mathbf{r} &= \mathbf{q}_{\mathcal{L}_{\|+\infty}/\mathcal{L}_{\|}} \otimes \hat{\mathbf{q}}_{\mathcal{L}_{\|+\infty}/\mathcal{L}_{\|}}^{-1} && (127)\\
&= \mathbf{q}_{\mathcal{L}/\mathcal{B}} \otimes \begin{bmatrix} \boldsymbol{\delta\rho} \\ \boldsymbol{\delta}\mathbf{q}_4 \end{bmatrix} \otimes \hat{\mathbf{q}}_{\mathcal{B}_{\|+\infty}/\mathcal{T}} \otimes \mathbf{q}_{\mathcal{B}_{\|}/\mathcal{T}}^{-1} \otimes \mathbf{q}_{\mathcal{L}/\mathcal{B}}^{-1} \otimes \left(\mathbf{q}_{\mathcal{L}/\mathcal{B}} \otimes \hat{\mathbf{q}}_{\mathcal{B}_{\|+\infty}/\mathcal{T}} \otimes \mathbf{q}_{\mathcal{B}_{\|}/\mathcal{T}}^{-1} \otimes \mathbf{q}_{\mathcal{L}/\mathcal{B}}^{-1}\right)^{-1} && (128)\\
&= \mathbf{q}_{\mathcal{L}/\mathcal{B}} \otimes \begin{bmatrix} \boldsymbol{\delta\rho} \\ \boldsymbol{\delta}\mathbf{q}_4 \end{bmatrix} \otimes \mathbf{q}_{\mathcal{L}/\mathcal{B}}^{-1} && (129)\\
&= \mathbf{q}_{\mathcal{L}/\mathcal{B}} \otimes \mathbf{q}_{\mathcal{L}/\mathcal{B}}^{-1} \circledast \begin{bmatrix} \boldsymbol{\delta\rho} \\ \boldsymbol{\delta}\mathbf{q}_4 \end{bmatrix} && (130)\\
where\ \ [\mathbf{a}\otimes] \equiv \begin{bmatrix} -\left[\mathbf{a}_v^{\times}\right] + a_0\mathbb{I} & \mathbf{a}_v \\ -\mathbf{a}_v^T & a_0 \end{bmatrix};\qquad [\mathbf{a}\circledast] &\equiv \begin{bmatrix} \left[\mathbf{a}_v^{\times}\right] + a_0\mathbb{I} & \mathbf{a}_v \\ -\mathbf{a}_v^T & a_0 \end{bmatrix};\qquad \mathbf{a} \equiv \begin{bmatrix} \mathbf{a}_v \\ a_0 \end{bmatrix} && (131)\\
thus\ \mathbf{q}_{\mathcal{L}_{\|+\infty}/\mathcal{L}_{\|}} &= \mathbf{q}_{\mathcal{L}/\mathcal{B}} \otimes \begin{bmatrix} -\left[\boldsymbol{\delta\rho}^{\times}\right] + \delta q_4\mathbb{I} & \boldsymbol{\delta\rho} \\ -\boldsymbol{\delta\rho}_v^T & \delta q_4 \end{bmatrix} \begin{bmatrix} -\mathbf{q}_{\mathcal{L}/\mathcal{B}_{\sqsubseteq}} \\ \mathbf{q}_{\mathcal{L}/\mathcal{B}_{\triangle}} \end{bmatrix} && (132)\\
&= \mathbf{q}_{\mathcal{L}/\mathcal{B}} \otimes \begin{bmatrix} -\left[\mathbf{q}_{\mathcal{L}/\mathcal{B}_{\sqsubseteq}}^{\times}\right] + \mathbf{q}_{\mathcal{L}/\mathcal{B}_{\triangle}}\mathbb{I} & -\mathbf{q}_{\mathcal{L}/\mathcal{B}_{\sqsubseteq}} \\ \mathbf{q}_{\mathcal{L}/\mathcal{B}_{\sqsubseteq}}^T & \mathbf{q}_{\mathcal{L}/\mathcal{B}_{\triangle}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta\rho} \\ \delta q_4 \end{bmatrix} && (133)\\
&= \frac{1}{2}\left[\mathbf{q}_{\mathcal{L}/\mathcal{B}}\otimes\right]\left[\mathbf{q}_{\mathcal{L}/\mathcal{B}}^{-1}\otimes\right] \begin{bmatrix} \boldsymbol{\delta\alpha} \\ 2\boldsymbol{\delta}\mathbf{q}_4 \end{bmatrix} && (134)\\
\therefore H_{reg-att} &= \begin{bmatrix} \tfrac{1}{2}\left(\mathbf{q}_{\mathcal{L}/\mathcal{B}}\otimes\right)\left(\mathbf{q}_{\mathcal{L}/\mathcal{B}}^{-1}\circledast\right) & 0_{3\times12} \end{bmatrix} && (135)
\end{aligned}
$$

IX. 5.4 Bulk Velocity Estimates

A. 5.4.1 Angular Velocity

$$
\begin{aligned}
\tilde{\mathbf{y}} &= {}^{\mathcal{L}}\tilde{\omega} + \boldsymbol{\eta}_{\omega} = C_{\mathcal{L}/\mathcal{B}}{}^{\mathcal{B}}\boldsymbol{\omega} + \boldsymbol{\eta}_{\omega} && (136)\\
\hat{\mathbf{y}} &= C_{\mathcal{L}/\mathcal{B}}{}^{\mathcal{B}}\hat{\boldsymbol{\omega}} && (137)\\
\mathbf{y} &= {}^{\mathcal{L}}\boldsymbol{\omega} && (138)\\
\omega &= \hat{\omega} + \boldsymbol{\delta\omega} && (139)\\
\therefore \mathbf{r} = \mathbf{y} - \hat{\mathbf{y}} &= C_{\mathcal{L}/\mathcal{B}}\boldsymbol{\delta\omega} = -C_{\mathcal{L}/\mathcal{B}}\left(\Delta\boldsymbol{\beta}_g + \boldsymbol{\eta}_{g_v}\right) && (140)\\
\therefore H_{\omega} &= \begin{bmatrix} 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & -C_{\mathcal{L}/\mathcal{B}} * 0_{3\times3} \end{bmatrix} && (141)
\end{aligned}
$$

B. 5.4.2 Translational Velocity

$$
\begin{aligned}
\tilde{\mathbf{y}} &= {}^{\mathcal{L}}\mathbf{v} + \boldsymbol{\eta}_v \qquad (142)\\
\hat{\mathbf{y}} = {}^{\mathcal{L}}\mathbf{v} &= C_{\mathcal{L}/\mathcal{B}}C_{\mathcal{B}/\mathcal{T}}{}^{\mathcal{T}}\mathbf{v} + \boldsymbol{\eta}_v \qquad (143)\\
\mathbf{y} &= C_{\mathcal{L}/\mathcal{B}}C_{\mathcal{B}/\mathcal{T}}{}^{\mathcal{T}}\mathbf{v} \qquad (144)\\
{}^{\mathcal{T}}\mathbf{v} &= {}^{\mathcal{T}}\hat{\mathbf{v}} + \Delta{}^{\mathcal{T}}\mathbf{v} \qquad (145)\\
\therefore\ \mathbf{r} = \mathbf{y} - \hat{\mathbf{y}} &= C_{\mathcal{L}/\mathcal{B}}\left(\mathbb{I} - \left[\boldsymbol{\delta\alpha}^{\times}\right]\right)\hat{C}_{\mathcal{B}/\mathcal{T}}\left({}^{\mathcal{T}}\mathbf{v} + \Delta{}^{\mathcal{T}}\mathbf{v}\right) - C_{\mathcal{L}/\mathcal{B}}\hat{C}_{\mathcal{B}/\mathcal{T}}{}^{\mathcal{T}}\mathbf{v} \qquad (146)\\
&= C_{\mathcal{L}/\mathcal{B}}\left[\left(\hat{C}_{\mathcal{B}/\mathcal{T}}{}^{\mathcal{T}}\hat{\mathbf{v}}\right)^{\times}\right]\boldsymbol{\delta\alpha} + C_{\mathcal{L}/\mathcal{B}}C_{\mathcal{B}/\mathcal{T}}\Delta{}^{\mathcal{T}}\mathbf{v} \qquad (147)\\
\therefore\ H_v &= \begin{bmatrix} C_{\mathcal{L}/\mathcal{B}}\left[\left(\hat{C}_{\mathcal{B}/\mathcal{T}}{}^{\mathcal{T}}\hat{\mathbf{v}}\right)^{\times}\right] & 0_{3\times3} & C_{\mathcal{L}/\mathcal{B}}C_{\mathcal{B}/\mathcal{T}} & 0_{3\times3} & 0_{3\times3} \end{bmatrix} \qquad (148)\\
{}^{\mathcal{T}}R_k &= \hat{C}_{\mathcal{T}/\mathcal{B}}C_{\mathcal{B}/\mathcal{L}}{}^{\mathcal{L}}R_k C_{\mathcal{B}/\mathcal{L}}^{T}C_{\mathcal{T}/\mathcal{B}}^{T} \qquad (149)
\end{aligned}
$$