2017-08-18

Relative Multiplicative Extended Kalman Filter for Observable GPS-Denied Navigation

Daniel P. Koch
Department of Mechanical Engineering, Brigham Young University, dkoch89@gmail.com

David O. Wheeler
Department of Electrical and Computer Engineering, Brigham Young University, dowheeler@gmail.com

Randal Beard
beard@byu.edu

Tim McLain
Brigham Young University - Provo, mclain@byu.edu

Kevin M. Brink
Air Force Research Laboratory, Munitions Directorate

Relative Multiplicative Extended Kalman Filter for Observable GPS-Denied Navigation

Daniel P. Koch[1], David O. Wheeler[2], Randal W. Beard[2], Timothy W. McLain[1], and Kevin M. Brink[3]

Abstract
This work presents a multiplicative extended Kalman filter (MEKF) for estimating the relative state of a multirotor vehicle operating in a GPS-denied environment. The filter fuses data from an inertial measurement unit and altimeter with relative-pose updates from a keyframe-based visual odometry or laser scan-matching algorithm. Because the global position and heading states of the vehicle are unobservable in the absence of global measurements such as GPS, the filter in this paper estimates the state with respect to a local frame that is colocated with the odometry keyframe. As a result, the odometry update provides nearly-direct measurements of the relative vehicle pose, making those states observable. Recent publications have rigorously documented the theoretical advantages of such an observable parameterization, including improved consistency, accuracy, and system robustness, and have demonstrated the effectiveness of such an approach during prolonged multirotor flight tests. This paper complements this prior work by providing a complete, self-contained, tutorial derivation of the relative MEKF, which has been thoroughly motivated but only briefly described to date. This paper presents several improvements and extensions to the filter while clearly defining all quaternion conventions and properties used, including several new useful properties relating to error quaternions and their Euler-angle decomposition. Finally, this paper derives the filter both for traditional dynamics defined with respect to an inertial frame, and for robocentric dynamics defined with respect to the vehicle's body frame, and provides insights into the subtle differences that arise between the two formulations.

Keywords
Sensor fusion, vision-aided inertial navigation, multiplicative extended Kalman filter, aerial robotics

1 Introduction

GPS-denied navigation for small unmanned aircraft systems (UAS) is an active and rich field of research with significant practical applications such as infrastructure inspection and security. Most UAS fuse GPS with accelerometer and rate-gyro data to provide accurate global state estimates suitable for feedback control. When GPS is not available, however, additional sensors such as cameras or lidars are required. Because of the size, weight, and power constraints and fast vehicle dynamics associated with small UAS, many such systems incorporate these additional sensors using filter-based estimation techniques rather than traditional full simultaneous localization and mapping (SLAM) algorithms. Filter-based approaches are computationally efficient and ensure smooth, timely state estimates for control.

In the absence of GPS updates, many filtering methods utilize incremental odometry measurements from either visual odometry or laser scan matching. These odometry measurements can be computed frame-to-frame, or several measurements can be computed with respect to the same *keyframe* image or scan. The keyframe image or scan is updated when there is insufficient overlap with current images or scans to compute reliable odometry measurements. Keyframe-based approaches have the advantage of reducing temporal drift in the odometry measurements (Leutenegger et al., 2015).

Despite having only incremental measurements available, the majority of GPS-denied navigation approaches directly estimate the vehicle's global pose with respect to some fixed origin. Without global position measurements, however, the vehicle's global pose and heading are unobservable (Martinelli, 2012; Weiss et al., 2012; Jones et al., 2007). As a result, global filters can suffer from inconsistent and unbounded state uncertainties, erratic state jumps when applying relative measurements, and the inability to directly apply intermittent global information without causing

[1]Department of Mechanical Engineering,
Brigham Young University, Provo, UT 84602, USA
[2]Department of Electrical and Computer Engineering,
Brigham Young University, Provo, UT 84602, USA
[3]Air Force Research Laboratory, Eglin AFB, FL 32542, USA

Corresponding author:
Daniel Koch, Brigham Young University
Email: daniel.koch@byu.edu

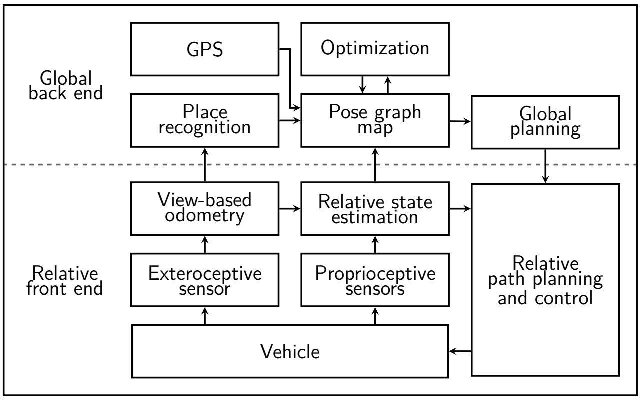


Fig. 1. Block diagram of the general relative navigation system architecture. Flight critical estimation and control is performed with respect to a local frame. The framework is described in more detail by Wheeler et al. (2017).

large state jumps (Julier and Uhlmann, 2001; Bailey and Durrant-Whyte, 2006; Kottas et al., 2013; Wheeler et al., 2018). In contrast, the relative navigation approach estimates only the relative state of the vehicle with respect to the location of the most recent odometry keyframe (Leishman et al., 2014b). As a result, the odometry provides direct measurements of the position and heading states, making them observable by construction and thereby ensure consistent and bounded state estimates. Each time a new keyframe is declared, the current state and covariance estimates are passed to a back-end map that concatenates them as edges in a pose graph to reconstruct the global path of the vehicle. The position and heading states in the filter are then reset to zero and estimation continues. The relative navigation architecture is illustrated graphically in Figure 1, and parallels ideas found in the SLAM literature (Chong and Kleeman, 1999; Kim et al., 2010).

Recent simulation results have shown that using the relative navigation framework to ensure observability provides significant advantages in terms of consistency of the estimated relative and global states, as well as some improvement in accuracy and system robustness when incorporating intermittent global information (Wheeler et al., 2018). Futhermore, recent multirotor hardware flight test results have demonstrated the effectiveness of relative navigation for prolonged GPS-degraded navigation of small UAS (Wheeler et al., 2017). This paper compliments (Wheeler et al., 2018) and (Wheeler et al., 2017) by providing a complete, self-contained, tutorial derivation of the relative state-estimation filter used in those papers to obtain hardware results. While those papers provide thorough theoretical and practical motivation for the relative navigation approach, the core relative estimator component itself has only been briefly described in the literature to date.

The purpose of this paper is to fill that gap. The flight tests presented by Wheeler et al. (2018) and Wheeler et al. (2017) successfully leveraged the relative multiplicative extended Kalman filter (RMEKF) presented in this paper. While that work demonstrated the effectiveness of the relative navigation framework and gave an overview of the various components, it did not describe the RMEKF in detail. The RMEKF builds upon the multiplicative Kalman filter (MEKF), which uses a quaternion to represent attitude and quaternion multiplication to define attitude error. The RMEKF extends the MEKF by defining the UAS state to be with respect to a local coordinate frame associated with the current keyframe image. To accommodate this relative state, the RMEKF introduces an additional *keyframe reset* step that is applied each time a new keyframe is declared.

This paper contributes to the literature in three ways. First, the paper provides a tutorial derivation of the MEKF for UAS state estimation given Hamilton quaternions. Second, the paper presents a complete derivation of the RMEKF for multirotor UAS, including several important extensions to the original presentation by Leishman and McLain (2014). Third, the paper provides a thorough derivation of the RMEKF for both inertial and body-fixed (robocentric) state representations, highlighting the subtle but important differences that exist between the two methods. The following paragraphs describe these contributions in further detail and relate how they compare to the existing literature.

MEKF Tutorial. Significant portions of the paper are tutorial in nature, clearly motivating why an indirect or error state formulation is necessary when quaternions are used to represent attitude, and providing complete explanations of each step in the derivation of the filter equations. The MEKF was first introduced by Lefferts et al. (1982), and several in-depth discussions and derivations of MEKF implementations have been published (Trawny and Roumeliotis, 2005; Markley, 2003; Sola, 2016). Some of these valuable publications are of similar scope to the current work, but this paper provides several meaningful extensions. First, this paper derives an estimator for the full state of a UAS (position, velocity, attitude, accelerometer biases, and gyroscope biases), while most previous MEKF papers of similar scope focus only on the attitude and bias estimation. Sola (2016) does derive a full-state MEKF, but this paper provides the derivation for a unique set of propagation and measurement models. Second, this paper derives the MEKF using the Hamilton quaternion convention as opposed to the JPL convention used in some other works. While the choice of quaternion convention does not fundamentally change the problem, Hamilton quaternions are commonly used in the robotics literature and subtle but important differences arise. This paper provides a contrasting perspective to help deepen understanding of quaternions, and aims to clarify some of the confusion that can arise due to the varying conventions used in different works that are not always explicitly defined. Third, the tutorial nature of this paper provides sufficient context for the derivation of several new properties relating to quaternions, their error representations, and their Euler-angle decomposition. These properties play a key role in the derivation of the RMEKF to allow partial attitude updates.

RMEKF Derivation. Another purpose of this paper is to provide a thorough derivation of the RMEKF estimator successfully used by Wheeler et al. (2017) for prolonged UAS navigation in GPS-degraded environments. The RMEKF presented in this paper extends the original RMEKF derivation by Leishman and McLain (2014) in several important ways that have proven necessary for prolonged flight. First, this paper presents a new visual odometry measurement model, laser scan-matching measurement model, and keyframe reset operation, that together ensure the state remains observable in GPS-denied environments. Second, several novel properties of error quaternions are derived that enable partial updates to quaternion states and their covariances. Third, new terms are added to the state vector in this paper to correctly account for uncertainty in the roll, pitch, and altitude of the vehicle at the time a keyframe is declared. Finally, smaller differences include reversing the direction of the odometry measurement model to avoid unnecessarily coupling heading uncertainty into the update, and estimating the global height of the vehicle above ground rather than treating altitude as a relative state.

Inertial and Body-Fixed Dynamics. Another unique contribution of this paper is the derivation of the RMEKF when the state is defined with respect to either an inertial frame or a body-fixed robocentric frame. Vehicle dynamics are traditionally expressed in an inertially-fixed, gravity-aligned frame (see Leishman et al. (2014b) and Leishman and McLain (2014)). This makes sense especially when the sensors are inertial sensors like GPS. However, expressing the dynamics in a robocentric frame is often more natural when using robocentric sensors like cameras and laser range finders, and can help address some of the inconsistency issues of EKF-SLAM (Castellanos et al. (2007); Bloesch et al. (2015)) when only relative states are required. A related contribution of this paper is a presentation of the subtle differences that arise between using an inertial and robocentric reference frame. These differences can cause confusion in the literature where a side-by-side comparison does not currently exist. For example, in addition to the change in dynamics, subtle changes appear in the quaternion integration, error state definition, measurement models, and keyframe reset operations. By presenting both formulations side-by-side, these differences are clearly explained.

The final contribution of this paper is a complete, self-contained derivation of the filter and all relevant quaternion properties. The definitions of quaternions and error states used across the current estimation literature differ in subtle ways. When these definitions are not thoroughly documented, it becomes difficult to correctly leverage properties from multiple sources. With its tutorial nature and step-by-step explanations, this paper is designed to present a complete, self-contained derivation with respect to a consistent, explicitly stated definition. This allows the reader to understand, implement, and potentially modify the RMEKF for new vehicles or applications. Note that while the keyframe reset step and several measurement models are specific to relative navigation, the propagation equations and general filter structure are equally relevant for other applications, such as GPS/INS navigation.

Section 2 summarizes the notation used throughout the paper. Section 3 provides an overview of the quaternion definitions used in the paper. Specifically, Section 3.8 derives several relevant, new properties of error quaternions. Section 4 outlines the structure of the MEKF. The keyframe reset step is described in Section 5, and an overview of the complete RMEKF algorithm is given. Section 6 derives the specific filter equations for inertial relative navigation (iRN), and Section 7 derives the equations for body-fixed robocentric relative navigation (bRN). Finally, Sections 8 and 9 respectively present results and conclusions.

2 Nomenclature

The following variables, operators, and notation are defined and motivated throughout the paper and are summarized here for convenience. Let $\mathcal{B}$ denote the vehicle's body frame and $\mathcal{I}$ denote an inertial frame.

State variables

$\mathbf{x}$	state
$\mathbf{x}_\mathbf{v}$	vector component of state
$\mathbf{x}_\mathbf{q}$	quaternion component of state
${}^c\mathbf{p}_a^b$	position of b with respect to a, expressed in c
$\mathbf{p}_a^b$	position of b with respect to a, expressed in a
$\mathbf{q}_a^b$	quaternion that rotates from a to b
$\mathbf{v}$	velocity of $\mathcal{B}$ with respect to $\mathcal{I}$, expressed in $\mathcal{B}$ ($\mathbf{v} \triangleq {}^{\mathcal{B}}\mathbf{v}_{\mathcal{I}}^{\mathcal{B}}$)
$\boldsymbol{\omega}$	angular velocity of $\mathcal{B}$ with respect to $\mathcal{I}$, expressed in $\mathcal{B}$ ($\boldsymbol{\omega} \triangleq {}^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{I}}^{\mathcal{B}}$)
$\mathbf{a}$	acceleration of $\mathcal{B}$ with respect to $\mathcal{I}$, expressed in $\mathcal{B}$ ($\mathbf{a} \triangleq {}^{\mathcal{B}}\mathbf{a}_{\mathcal{I}}^{\mathcal{B}}$)
$\boldsymbol{\beta}_{\boldsymbol{\omega}}$	rate gyro biases, expressed in $\mathcal{B}$
$\boldsymbol{\beta}_\mathbf{a}$	accelerometer biases, expressed in $\mathcal{B}$
μ	specific linear drag coefficient
$\boldsymbol{\eta}$	zero-mean Gaussian process noise
$\boldsymbol{\upsilon}$	zero-mean Gaussian input noise

Error state variables

$\delta\mathbf{x}$	error state
$\delta\mathbf{x}_\mathbf{v}$	vector component of error state
$\delta\mathbf{x}_{\boldsymbol{\theta}}$	attitude component of error state
$\delta\mathbf{p}$	position error state
$\delta\mathbf{q}$	quaternion error state
$\delta\boldsymbol{\theta}$	attitude error state (minimal representation)

Filter variables

$\mathbf{u}$	input
$\mathbf{z}$	measurement
$\mathbf{r}$	measurement residual
$\mathbf{P}$	state covariance
$\mathbf{Q}$	process noise covariance
$\mathbf{R}$	measurement noise covariance
$\mathbf{S}$	residual covariance
$\mathbf{K}$	Kalman gain
$\mathbf{F}, \mathbf{G}$	propagation Jacobians
$\mathbf{H}$	measurement Jacobian
$\mathbf{N}$	keyframe reset Jacobian
$\mathbf{N_p}$	position reset Jacobian
$\mathbf{N}_{\boldsymbol{\theta}}$	attitude reset Jacobian

Operators

$\otimes$	quaternion multiplication (Hamilton)
$\lfloor \cdot \rfloor$	skew-symmetric matrix
$(\cdot)^\wedge$	mapping from vector to quaternion
$(\cdot)^\vee$	mapping from quaternion to vector
$E[\cdot]$	expected value
$\mathbf{R}(\mathbf{q})$	rotation matrix associated with $\mathbf{q}$

Other

$\hat{\mathbf{y}}$	estimate (or expected value) of $\mathbf{y}$
$\tilde{\mathbf{y}}$	measurement of $\mathbf{y}$
$\dot{\mathbf{y}}$	time derivative of $\mathbf{y}$
$\mathbf{y}^+$	*a posteriori* value of $\mathbf{y}$
$\mathbf{k}$	unit vector $\begin{bmatrix}0 & 0 & 1\end{bmatrix}^\mathsf{T}$
$\mathbf{I}_{a \times b}$	identity matrix in $\mathbb{R}^{a \times b}$
$\mathbf{\Pi_k}$	Projection matrix $\mathbf{I}_{3\times 3} - \mathbf{k}\mathbf{k}^\mathsf{T}$
$\mathbf{g}$	gravity vector $g\mathbf{k}$

3 Quaternion Properties

Quaternions are a common method for representing attitude due to their improved computational efficiency and accuracy compared to alternative approaches (Casey et al., 2013). A variety of definitions exist for quaternions and their associated operations, leading to subtle discrepancies and potential confusion. The various approaches, described in more detail by Sola (2016), include left-handed vs. right-handed quaternion multiplication, active vs. passive representations, local-to-global vs. global-to-local attitude direction, and quaternion ordering. This section explicitly establishes the definitions and notation used throughout this paper and additionally derives several properties required for the filter's derivation. This section is not intended as a complete introduction to quaternions, but rather as a summary of relevant points.

3.1 Quaternion Conventions

A quaternion $\mathbf{q} \in \mathbb{H}$ is a hyper-complex number of rank four consisting of a scalar and vector portion as

$$\mathbf{q} = q_0 + q_x i + q_y j + q_z k \, .$$

We use the Hamilton definition of the quaternion, with

$$\begin{aligned} ij &= -ji = k \, , \\ jk &= -kj = i \, , \\ ki &= -ik = j \, , \\ i^2 = j^2 &= k^2 = ijk = -1 \, . \end{aligned} \tag{1}$$

For notational convenience we define the vector portion of the quaternion as

$$\bar{\mathbf{q}} = \begin{bmatrix} q_x & q_y & q_z \end{bmatrix}^\mathsf{T} \, ,$$

and write the quaternion as

$$\mathbf{q} = \begin{bmatrix} \bar{\mathbf{q}} \\ q_0 \end{bmatrix} \, . \tag{2}$$

Quaternion multiplication is denoted with the $\otimes$ operator, and is carried out according to the rules in (1) and standard algebraic multiplication. Using the notation in (2), quaternion multiplication can be written as a matrix multiplication according to

$$\begin{aligned} \mathbf{p} \otimes \mathbf{q} &= \begin{bmatrix} p_0 \mathbf{I} + \lfloor \bar{\mathbf{p}} \rfloor & \bar{\mathbf{p}} \\ -\bar{\mathbf{p}}^\mathsf{T} & p_0 \end{bmatrix} \begin{bmatrix} \bar{\mathbf{q}} \\ q_0 \end{bmatrix} \, , & \text{(3a)} \\ &= \begin{bmatrix} q_0 \mathbf{I} - \lfloor \bar{\mathbf{q}} \rfloor & \bar{\mathbf{q}} \\ -\bar{\mathbf{q}}^\mathsf{T} & q_0 \end{bmatrix} \begin{bmatrix} \bar{\mathbf{p}} \\ p_0 \end{bmatrix} \, , & \text{(3b)} \end{aligned}$$

where the operator $\lfloor \cdot \rfloor$ is the skew-symmetric operator

$$\lfloor \mathbf{a} \rfloor = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}$$

so that $\mathbf{a} \times \mathbf{b} = \lfloor \mathbf{a} \rfloor \mathbf{b}$. The skew-symmetric operator has the property that

$$\lfloor \mathbf{a} \rfloor \mathbf{b} = - \lfloor \mathbf{b} \rfloor \mathbf{a} \, . \tag{4}$$

The conjugate of a quaternion $\mathbf{q}$ is denoted by $\mathbf{q}^*$, and is equal to $\mathbf{q}$ but with the elements of the vector portion negated. The inverse of a quaternion is given by

$$\mathbf{q}^{-1} = \frac{\mathbf{q}^*}{\|\mathbf{q}\|} \, .$$

The quaternions used in this paper all represent rotations and so are *unit quaternions*, meaning that their norm is 1. Therefore, for unit quaternions we have

$$\mathbf{q}^{-1} = \mathbf{q}^* = \begin{bmatrix} -\bar{\mathbf{q}} \\ q_0 \end{bmatrix} \, . \tag{5}$$

Inverting the product of two quaternions results in the product of the inverse of each quaternion in the opposite order, as

$$(\mathbf{p} \otimes \mathbf{q})^{-1} = \mathbf{q}^{-1} \otimes \mathbf{p}^{-1} \, .$$

3.2 Vector Rotation

In this paper, quaternions are denoted passively, meaning that they represent the rotation necessary to express a vector in a different frame. Let quaternion $\mathbf{q}_a^b$ represent the rotation from frame a to frame b and let

${}^a\mathbf{y}$ represent a vector expressed in frame a. As described by Kuipers (1999), ${}^a\mathbf{y}$ can be expressed in frame b using the quaternion conjugation operation as

$$\begin{bmatrix} {}^b\mathbf{y} \\ 0 \end{bmatrix} = (\mathbf{q}_a^b)^{-1} \otimes \begin{bmatrix} {}^a\mathbf{y} \\ 0 \end{bmatrix} \otimes \mathbf{q}_a^b \, . \tag{6}$$

The term $\begin{bmatrix} {}^a\mathbf{y}^\mathsf{T} & 0 \end{bmatrix}^\mathsf{T}$ is referred to as the *pure quaternion* constructed from ${}^a\mathbf{y}$.

It is convenient to define an equivalent rotation matrix $\mathbf{R}(\mathbf{q})$ such that

$$\begin{bmatrix} \mathbf{R}(\mathbf{q})\mathbf{y} \\ 0 \end{bmatrix} = \mathbf{q}^{-1} \otimes \begin{bmatrix} \mathbf{y} \\ 0 \end{bmatrix} \otimes \mathbf{q} \, . \tag{7}$$

An expression for $\mathbf{R}(\mathbf{q})$ can be derived by expanding the right-hand side of (7) according to (5), (3a), and (3b) as

$$\begin{aligned} \begin{bmatrix} \mathbf{R}(\mathbf{q})\mathbf{y} \\ 0 \end{bmatrix} &= \begin{bmatrix} q_0\mathbf{I} - \lfloor\bar{\mathbf{q}}\rfloor & -\bar{\mathbf{q}} \\ \bar{\mathbf{q}}^\mathsf{T} & q_0 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ 0 \end{bmatrix} \otimes \mathbf{q} \\ &= \begin{bmatrix} q_0\mathbf{y} - \lfloor\bar{\mathbf{q}}\rfloor \mathbf{y} \\ \bar{\mathbf{q}}^\mathsf{T}\mathbf{y} \end{bmatrix} \otimes \mathbf{q} \\ &= \begin{bmatrix} q_0\mathbf{I} - \lfloor\bar{\mathbf{q}}\rfloor & \bar{\mathbf{q}} \\ -\bar{\mathbf{q}}^\mathsf{T} & q_0 \end{bmatrix} \begin{bmatrix} q_0\mathbf{y} - \lfloor\bar{\mathbf{q}}\rfloor \mathbf{y} \\ \bar{\mathbf{q}}^\mathsf{T}\mathbf{y} \end{bmatrix} \\ &= \begin{bmatrix} \left(q_0^2\mathbf{I} - 2q_0\lfloor\bar{\mathbf{q}}\rfloor + \bar{\mathbf{q}}\bar{\mathbf{q}}^\mathsf{T} + \lfloor\bar{\mathbf{q}}\rfloor^2\right)\mathbf{y} \\ 0 \end{bmatrix} , \end{aligned} \tag{8}$$

which implies that

$$\mathbf{R}(\mathbf{q}) = q_0^2\mathbf{I} - 2q_0\lfloor\bar{\mathbf{q}}\rfloor + \bar{\mathbf{q}}\bar{\mathbf{q}}^\mathsf{T} + \lfloor\bar{\mathbf{q}}\rfloor^2 \, .$$

It can be shown, however, that

$$\lfloor\bar{\mathbf{q}}\rfloor^2 = \bar{\mathbf{q}}\bar{\mathbf{q}}^\mathsf{T} - \left(1 - q_0^2\right)\mathbf{I} \, ,$$

so that

$$\mathbf{R}(\mathbf{q}) = \left(2q_0^2 - 1\right)\mathbf{I} - 2q_0\lfloor\bar{\mathbf{q}}\rfloor + 2\bar{\mathbf{q}}\bar{\mathbf{q}}^\mathsf{T} \, . \tag{9}$$

Rotation matrices exhibit the following properties:

$$\begin{aligned} \mathbf{R}^{-1}(\mathbf{q}_a^b) &= \mathbf{R}^\mathsf{T}(\mathbf{q}_a^b) = \mathbf{R}(\mathbf{q}_b^a) \\ \mathbf{R}(\mathbf{q}_a^c) &= \mathbf{R}(\mathbf{q}_b^c)\mathbf{R}(\mathbf{q}_a^b) \\ \det(\mathbf{R}(\mathbf{q})) &= 1 \, . \end{aligned} \tag{10}$$

The formula for vector rotation in equation (6) can be used to derive the manner in which two rotations are compounded together. If $\mathbf{q}_a^b$ defines the rotation from frame a to frame b, and $\mathbf{q}_b^c$ the rotation from frame b to frame c, then to take a vector expressed in frame a and express it in frame c we have

$$\begin{aligned} \begin{bmatrix} {}^c\mathbf{y} \\ 0 \end{bmatrix} &= (\mathbf{q}_b^c)^{-1} \otimes \begin{bmatrix} {}^b\mathbf{y} \\ 0 \end{bmatrix} \otimes \mathbf{q}_b^c \\ &= (\mathbf{q}_b^c)^{-1} \otimes \left((\mathbf{q}_a^b)^{-1} \otimes \begin{bmatrix} {}^a\mathbf{y} \\ 0 \end{bmatrix} \otimes \mathbf{q}_a^b \right) \otimes \mathbf{q}_b^c \\ &= \left(\mathbf{q}_a^b \otimes \mathbf{q}_b^c\right)^{-1} \otimes \begin{bmatrix} {}^a\mathbf{y} \\ 0 \end{bmatrix} \otimes \left(\mathbf{q}_a^b \otimes \mathbf{q}_b^c\right) \, . \end{aligned}$$

We therefore conclude that

$$\mathbf{q}_a^c = \mathbf{q}_a^b \otimes \mathbf{q}_b^c \, . \tag{11}$$

Comparing (10) and (11), we see that rotation matrices and quaternions compound in the opposite order:

$$\mathbf{R}(\mathbf{q}_a^b \otimes \mathbf{q}_b^c) = \mathbf{R}(\mathbf{q}_b^c)\mathbf{R}(\mathbf{q}_a^b) \, . \tag{12}$$

When the quaternion conjugation operation is applied to a quaternion that is not a pure quaternion, the analysis procedure in equation (8) can be repeated to show that

$$\mathbf{q}^{-1} \otimes \mathbf{p} \otimes \mathbf{q} = \begin{bmatrix} \mathbf{R}(\mathbf{q})\bar{\mathbf{p}} \\ p_0 \end{bmatrix} \tag{13}$$

for the same rotation matrix $\mathbf{R}(\mathbf{q})$ defined by (9). The result of this operation is that the basis of the vector portion of $\mathbf{p}$ is rotated by the rotation defined by $\mathbf{q}$. This property is useful when the quaternion is interpreted according to its axis-angle representation, and it is desirable to express the axis vector in a different coordinate frame. This arises in the derivation of some of the measurement models and Jacobians in this paper.

3.3 Unit Sphere Propagation

Attitude is represented using quaternions of unit length. Unit quaternions do not form a vector space, but rather form a group on the unit sphere $S^3 \subset \mathbb{H}$. The group operator is quaternion multiplication and the group of unit quaternions gives a double cover parameterization of the group of rotations $SO(3)$.

Multiplying a unit quaternion by a non-unit quaternion will cause the product to leave the unit sphere. Normalizing the resulting quaternion according to

$$\mathbf{q} \leftarrow \frac{\mathbf{q}}{\|\mathbf{q}\|}$$

returns the quaternion to the unit sphere, but linearization errors are introduced.

To properly rotate a quaternion along the manifold, it is necessary to represent the rotation in terms of a unit quaternion. A rotation can be represented using the unit quaternion as

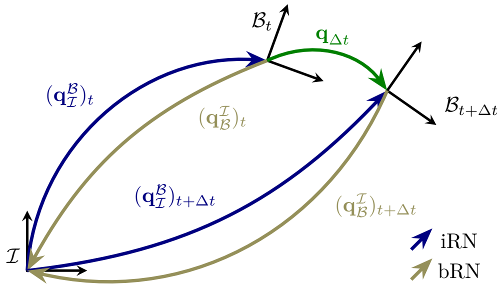
$$\mathbf{q} = \begin{bmatrix} \hat{\mathbf{e}} \sin\frac{\theta}{2} \\ \cos\frac{\theta}{2} \end{bmatrix} \, , \tag{14}$$

where $\hat{\mathbf{e}}$ is a unit vector defining the axis of rotation and θ is the angle of rotation about that axis. Let $\boldsymbol{\theta} \triangleq \theta\hat{\mathbf{e}} \in \mathbb{R}^3$ define the magnitude and direction of rotation. The mapping from this three-vector rotation parameterization to a quaternion is denoted by the operator ${}^\wedge : \mathbb{R}^3 \to \mathbb{H}$ and the inverse mapping by the operator ${}^\vee : \mathbb{H} \to \mathbb{R}^3$. The ${}^\wedge$ operator is defined using (14) as

$$\boldsymbol{\theta}^\wedge \triangleq \begin{bmatrix} \frac{\boldsymbol{\theta}}{\|\boldsymbol{\theta}\|} \sin\frac{\|\boldsymbol{\theta}\|}{2} \\ \cos\frac{\|\boldsymbol{\theta}\|}{2} \end{bmatrix} . \tag{15}$$

Rotating a quaternion $\mathbf{q}$ along the unit sphere by the rotation $\boldsymbol{\theta}$ is accomplished as $\mathbf{q} \otimes \boldsymbol{\theta}^\wedge$. This is analogous to the notation $\mathbf{q} \boxplus \boldsymbol{\theta}$ and $\mathbf{q} \otimes \exp(\frac{\boldsymbol{\theta}}{2})$ found in the estimation literature (Hertzberg et al., 2013). The ${}^\vee$ operator is defined as

$$\mathbf{q}^\vee \triangleq 2 \operatorname{atan2}\left(\|\bar{\mathbf{q}}\|, q_0\right) \frac{\bar{\mathbf{q}}}{\|\bar{\mathbf{q}}\|} \, . \tag{16}$$



$\mathcal{B}_t$
$\mathbf{q}_{\Delta t}$
$\mathcal{B}_{t+\Delta t}$
$(\mathbf{q}_{\mathcal{I}}^{\mathcal{B}})_t$
$(\mathbf{q}_{\mathcal{B}}^{\mathcal{I}})_t$
$(\mathbf{q}_{\mathcal{I}}^{\mathcal{B}})_{t+\Delta t}$
$(\mathbf{q}_{\mathcal{B}}^{\mathcal{I}})_{t+\Delta t}$
$\mathcal{I}$
iRN
bRN

Fig. 2. Quaternion definitions used in the derivation of the quaternion time derivative and integration. The same $\mathbf{q}_{\Delta t}$ is used for both iRN and bRN because it corresponds to the body-fixed angular velocity $\boldsymbol{\omega}$ measured by the rate gyros.

Extracting the underlying rotation between $\mathbf{q}_a$ and $\mathbf{q}_b$ is accomplished as $(\mathbf{q}_a \otimes (\mathbf{q}_b)^{-1})^{\vee}$. This is analogous to the notation $\log(\mathbf{q}_a \boxminus \mathbf{q}_b)$ found in the estimation literature (Hertzberg et al., 2013). As inverse mappings, it can be shown that $\boldsymbol{\theta} = \left(\boldsymbol{\theta}^{\wedge}\right)^{\vee}$ and $\mathbf{q} = (\mathbf{q}^{\vee})^{\wedge}$.

Eqn. (15) is undefined when $\|\boldsymbol{\theta}\|$ equals zero, and in practice becomes numerically unstable as $\|\boldsymbol{\theta}\|$ approaches zero. There are a number of common approximations of (15) for a small angle $\delta\boldsymbol{\theta}$, such as the second-order Gibbs vector parameterization (Markley, 2003)

$$\delta\boldsymbol{\theta}^{\wedge} \approx \frac{1}{\sqrt{4 + \delta\boldsymbol{\theta}^{\mathsf{T}}\delta\boldsymbol{\theta}}} \begin{bmatrix} \delta\boldsymbol{\theta} \\ 2 \end{bmatrix} . \tag{17}$$

The first-order approximation of both (15) and (17) is

$$\delta\boldsymbol{\theta}^{\wedge} \approx \begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\theta} \\ 1 \end{bmatrix} , \tag{18}$$

which is useful when deriving first-order Jacobians. Eqn. (16) can similarly be approximated for a small quaternion $\delta\mathbf{q}$ as

$$\delta\mathbf{q}^{\vee} \approx 2\,\mathrm{sign}\,(\delta q_0)\,\delta\bar{\mathbf{q}} . \tag{19}$$

3.4 Time Integration

Several methods can be used to numerically integrate a quaternion that represents the attitude of a rigid body. Numerical integration is carried out over a finite time step Δt, and is governed by the angular velocity of the body with respect to an inertial frame as expressed in the body frame, $\boldsymbol{\omega} \triangleq {}^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{I}}^{\mathcal{B}}$. Let $\mathbf{q}_{\Delta t}$ be an incremental quaternion and $\boldsymbol{\omega}_0$ be the nominal angular velocity. Zero-order quaternion integration assumes that the angular velocity is constant over the timestep, $\boldsymbol{\omega}_0 = \boldsymbol{\omega}_t$, while first-order integration uses linear interpolation, $\boldsymbol{\omega}_0 = \frac{1}{2}(\boldsymbol{\omega}_t + \boldsymbol{\omega}_{t-1})$. From (15), we write $\mathbf{q}_{\Delta t}$ as

$$\begin{aligned} \mathbf{q}_{\Delta t} &= (\boldsymbol{\omega}_0 \Delta t)^{\wedge} \\ &= \begin{bmatrix} \frac{\boldsymbol{\omega}_0}{\|\boldsymbol{\omega}_0\|} \sin\left(\frac{\|\boldsymbol{\omega}_0\|\Delta t}{2}\right) \\ \cos\left(\frac{\|\boldsymbol{\omega}_0\|\Delta t}{2}\right) \end{bmatrix} . \end{aligned} \tag{20}$$

The value of the quaternion $\mathbf{q}_{t+\Delta t}$ at time $t + \Delta t$ can be expressed as the combination of the quaternion $\mathbf{q}_t$ at time t and the incremental quaternion $\mathbf{q}_{\Delta t}$. The manner in which these quaternions are combined depends on whether the attitude quaternion represents the attitude of the body with respect to an inertial frame (iRN), or the attitude of an inertial frame with respect to the body (bRN). As illustrated in Figure 2, the attitude at time $t + \Delta t$ for these cases is

$$\begin{aligned} \text{iRN:} \quad \mathbf{q}_{t+\Delta t} &= \mathbf{q}_t \otimes \mathbf{q}_{\Delta t} \ , \\ \text{bRN:} \quad \mathbf{q}_{t+\Delta t} &= (\mathbf{q}_{\Delta t})^{-1} \otimes \mathbf{q}_t \ , \end{aligned} \tag{21}$$

where the order of compounding follows equation (11).

Substituting (20) into (21) gives

$$\text{iRN:} \quad \mathbf{q}_{t+\Delta t} = \mathbf{q}_t \otimes \begin{bmatrix} \frac{\boldsymbol{\omega}_0}{\|\boldsymbol{\omega}_0\|} \sin\left(\frac{\|\boldsymbol{\omega}_0\|\Delta t}{2}\right) \\ \cos\left(\frac{\|\boldsymbol{\omega}_0\|\Delta t}{2}\right) \end{bmatrix} , \tag{22a}$$

$$\text{bRN:} \quad \mathbf{q}_{t+\Delta t} = \begin{bmatrix} -\frac{\boldsymbol{\omega}_0}{\|\boldsymbol{\omega}_0\|} \sin\left(\frac{\|\boldsymbol{\omega}_0\|\Delta t}{2}\right) \\ \cos\left(\frac{\|\boldsymbol{\omega}_0\|\Delta t}{2}\right) \end{bmatrix} \otimes \mathbf{q}_t . \tag{22b}$$

Integrating according to (22) maintains unit norm, allowing the attitude to propagate on the unit sphere $S^3 \subset \mathbb{H}$. In practice, however, this definition becomes numerically unstable as $\|\boldsymbol{\omega}_0\|$ approaches zero. As described by Trawny and Roumeliotis (2005), applying L'Hospital's rule to (22a) for iRN shows that

$$\lim_{\|\boldsymbol{\omega}_0\| \to 0} \mathbf{q}_{t+\Delta t} = \mathbf{q}_t + \Delta t \left(\frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}_0)\mathbf{q}_t\right) , \tag{23}$$

where

$$\boldsymbol{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} -\lfloor\boldsymbol{\omega}\rfloor & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^{\mathsf{T}} & 0 \end{bmatrix} . \tag{24}$$

Comparing (24) to (3b) shows that (23) can be written as

$$\mathbf{q}_{t+\Delta t} = \mathbf{q}_t + \Delta t \left(\frac{1}{2}\mathbf{q}_t \otimes \begin{bmatrix} \boldsymbol{\omega}_0 \\ 0 \end{bmatrix}\right) .$$

For bRN, a similar analysis can be applied to (22b) to show

$$\mathbf{q}_{t+\Delta t} \approx \mathbf{q}_t + \Delta t \left(\frac{1}{2} \begin{bmatrix} -\boldsymbol{\omega}_0 \\ 0 \end{bmatrix} \otimes \mathbf{q}_t\right) .$$

Note that for bRN both the order of the quaternion multiplication and the sign of $\boldsymbol{\omega}_0$ have been reversed.

In summary, the attitude quaternion is integrated according to (22) when $\|\boldsymbol{\omega}_0\|$ is sufficiently large to avoid numerical issues. When $\|\boldsymbol{\omega}_0\|$ is small, the integration is approximated as

$$\text{iRN:} \quad \mathbf{q}_{t+\Delta t} = \mathbf{q}_t + \Delta t \left(\frac{1}{2}\mathbf{q}_t \otimes \begin{bmatrix} \boldsymbol{\omega}_0 \\ 0 \end{bmatrix}\right) , \tag{25a}$$

$$\text{bRN:} \quad \mathbf{q}_{t+\Delta t} = \mathbf{q}_t + \Delta t \left(\frac{1}{2} \begin{bmatrix} -\boldsymbol{\omega}_0 \\ 0 \end{bmatrix} \otimes \mathbf{q}_t\right) . \tag{25b}$$

Integrating according to (25) causes the quaternion to depart from the unit sphere S^3; when this method is used a normalization step therefore follows.

3.5 Attitude Kinematics

The attitude kinematics of a rigid body are described by the time derivative of the attitude quaternion. Some authors (Bloesch et al., 2016; Hertzberg et al.,

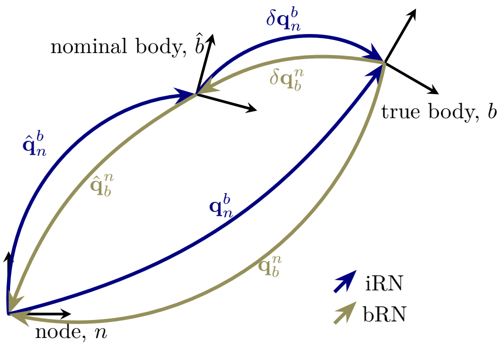


Fig. 3. Error state definitions for inertial and body-fixed relative navigation. To keep the same intermediate nominal frame, and as a result have similar error dynamics, the error state definitions are defined differently.

2013) emphasize the manifold structure of attitude dynamics by defining $\dot{\mathbf{q}} \in \mathbb{R}^3$ as a member of the associated Lie algebra, while other authors (Trawny and Roumeliotis, 2005; Markley, 2003) assume a first-order approximation such that $\dot{\mathbf{q}} \in \mathbb{R}^4$. In this paper, while we use (22) to propagate along the manifold S^3, we use the first-order approximation of the quaternion dynamics over a finite timestep Δt for computing the first-order Jacobian matrices required by the extended Kalman filter.

The first-order Taylor series approximation of (22), which assumes $\frac{1}{2}\|\boldsymbol{\omega}_0\|\Delta t$ is small*, yields the same result as (25). Using (25a), the quaternion time derivative for iRN is computed as

$$\begin{aligned}
\dot{\mathbf{q}} &= \lim_{\Delta t \to 0} \frac{1}{\Delta t}\left(\mathbf{q}_{t+\Delta t} - \mathbf{q}_t\right) \\
\dot{\mathbf{q}} &= \lim_{\Delta t \to 0} \frac{1}{\Delta t}\left(\mathbf{q}_t + \Delta t\left(\frac{1}{2}\mathbf{q}_t \otimes \begin{bmatrix}\boldsymbol{\omega}_0 \\ 0\end{bmatrix}\right) - \mathbf{q}_t\right) \\
&= \frac{1}{2}\mathbf{q}_t \otimes \begin{bmatrix}\boldsymbol{\omega} \\ 0\end{bmatrix} .
\end{aligned}$$

A similar analysis follows for bRN using (25b). In summary, attitude quaternion kinematics are represented by

$$\text{iRN:} \quad \dot{\mathbf{q}} = \frac{1}{2}\mathbf{q} \otimes \begin{bmatrix}\boldsymbol{\omega} \\ 0\end{bmatrix} , \tag{26}$$

$$\text{bRN:} \quad \dot{\mathbf{q}} = \frac{1}{2}\begin{bmatrix}-\boldsymbol{\omega} \\ 0\end{bmatrix} \otimes \mathbf{q} . \tag{27}$$

3.6 Error State

Because unit quaternions do not form a vector space, quaternion error cannot be computed using vector subtraction. Rather, a true quaternion state $\mathbf{q}$ is represented as the quaternion multiplication of an estimated quaternion $\hat{\mathbf{q}}$ and quaternion attitude error $\delta\mathbf{q}$. By varying the order and direction of the quaternion multiplication, there are four possible methods to define attitude error:

$$\begin{aligned}
&\text{Method 1:} \quad \mathbf{q} \triangleq \hat{\mathbf{q}} \otimes \delta\mathbf{q} , && (28a) \\
&\text{Method 2:} \quad \mathbf{q} \triangleq \delta\mathbf{q} \otimes \hat{\mathbf{q}} , && (28b) \\
&\text{Method 3:} \quad \mathbf{q} \triangleq \hat{\mathbf{q}} \otimes \delta\mathbf{q}^{-1} , && \\
&\text{Method 4:} \quad \mathbf{q} \triangleq \delta\mathbf{q}^{-1} \otimes \hat{\mathbf{q}} . &&
\end{aligned}$$

In this paper, we use (28a) for inertial relative navigation (iRN) described in Section 6 and use (28b) for body-fixed relative navigation (bRN) described in Section 7:

$$\text{iRN:} \quad \mathbf{q} \triangleq \hat{\mathbf{q}} \otimes \delta\mathbf{q} , \tag{29a}$$

$$\text{bRN:} \quad \mathbf{q} \triangleq \delta\mathbf{q} \otimes \hat{\mathbf{q}} . \tag{29b}$$

While using different definitions requires additional care when deriving the filter, ultimately this minimizes differences between the dynamics, measurement models, and keyframe reset steps of iRN and bRN. Figure 3 illustrates the choice of error state definitions and shows how this selection allows both approaches to keep the same intermediate nominal body frame $\hat{b}$. Rearranging (29), the quaternion error state is defined as

$$\text{iRN:} \quad \delta\mathbf{q} = \hat{\mathbf{q}}^{-1} \otimes \mathbf{q} , \tag{30a}$$

$$\text{bRN:} \quad \delta\mathbf{q} = \mathbf{q} \otimes \hat{\mathbf{q}}^{-1} . \tag{30b}$$

Using (12), we can express (29) as

$$\text{iRN:} \quad \mathbf{R}(\mathbf{q}) = \mathbf{R}(\delta\mathbf{q})\mathbf{R}(\hat{\mathbf{q}}) , \tag{31a}$$

$$\text{bRN:} \quad \mathbf{R}(\mathbf{q}) = \mathbf{R}(\hat{\mathbf{q}})\mathbf{R}(\delta\mathbf{q}) . \tag{31b}$$

When representing the attitude uncertainty associated with a quaternion error, a minimal representation is required. A quaternion is parameterized with four numbers, but only three are required to fully parameterize an orientation since orientations are associated with unit quaternions, elements of the three dimensional group S^3. Because the group is three dimensional, the tangent space at the identity element, or Lie algebra, will be isomorphic to $\mathbb{R}^3$, and error covariances can be defined in this 3-dimensional vector space. Accordingly, we represent the uncertainty in $\delta\mathbf{q}$ as the covariance of the vector $\delta\boldsymbol{\theta} \in \mathbb{R}^3$. The $^\wedge$ and $^\vee$ operators of (18) and (19) define the mapping between the error quaternion $\delta\mathbf{q}$ and its minimal representation $\delta\boldsymbol{\theta}$ as

$$\begin{aligned}
\delta\boldsymbol{\theta} &= \delta\mathbf{q}^\vee \approx 2\,\mathrm{sign}(q_0)\bar{\mathbf{q}} , \\
\delta\mathbf{q} &= \delta\boldsymbol{\theta}^\wedge \approx \begin{bmatrix}\frac{1}{2}\delta\boldsymbol{\theta} \\ 1\end{bmatrix} .
\end{aligned} \tag{32}$$

By substituting (32) into (9) and ignoring second-order terms, it follows that

$$\mathbf{R}(\delta\mathbf{q}) \approx \mathbf{I} - \lfloor\delta\boldsymbol{\theta}\rfloor \tag{33}$$

and

$$\mathbf{R}(\delta\mathbf{q}^{-1}) = \mathbf{R}^\top(\delta\mathbf{q}) \approx \mathbf{I} + \lfloor\delta\boldsymbol{\theta}\rfloor . \tag{34}$$

*Even for a large $\|\omega\| = 2\pi$ rad/s and moderate $\Delta t = 0.01$ s, the error introduced by linearizing the integration is only on the order of 10^{-6} rad.

3.7 Euler Decomposition

Aircraft attitude is commonly represented using three angles: roll ϕ, pitch θ, and yaw ψ. Yaw represents the rotation about the inertial z-axis (down). Pitch represents the rotation about the resulting y-axis. Roll represents the rotation about the x-axis formed after pitching and yawing. This sequence of rotations, known as 3-2-1 Euler angles, relates the vehicle's body frame to an inertial frame, and can be represented as the multiplication of three rotation matrices

$$\mathbf{R}(\mathbf{q}) = \mathbf{R}_\phi \mathbf{R}_\theta \mathbf{R}_\psi \ , \tag{35}$$

where

$$\begin{aligned}
\mathbf{R}_\phi &\triangleq \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} , \\
\mathbf{R}_\theta &\triangleq \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} , \\
\mathbf{R}_\psi &\triangleq \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} .
\end{aligned} \tag{36}$$

Because quaternions are generally less intuitive, they are often mapped into roll, pitch, and yaw angles for plotting, analysis, and control. Expanding (35) using (9) and comparing terms, we obtain

$$\begin{aligned}
\phi &= \operatorname{atan}\left(\frac{2q_0q_x + 2q_yq_z}{q_z^2 - q_x^2 - q_y^2 + q_0^2} \right) , \\
\theta &= \operatorname{asin}\left(2q_0q_y - 2q_xq_z\right) , \\
\psi &= \operatorname{atan}\left(\frac{2q_0q_z + 2q_xq_y}{q_x^2 - q_y^2 - q_z^2 + q_0^2} \right) .
\end{aligned} \tag{37}$$

To map Euler angles into a quaternion, we compose the attitude quaternion from the roll, pitch, and yaw components using the order derived in (11) such that

$$\mathbf{q} = \mathbf{q}_\psi \otimes \mathbf{q}_\theta \otimes \mathbf{q}_\phi \ . \tag{38}$$

Note the order of composition is opposite of (35) as described by (12). From (14), we get

$$\mathbf{q}_\phi \triangleq \begin{bmatrix} \sin\frac{\phi}{2} \\ 0 \\ 0 \\ \cos\frac{\phi}{2} \end{bmatrix} , \mathbf{q}_\theta \triangleq \begin{bmatrix} 0 \\ \sin\frac{\theta}{2} \\ 0 \\ \cos\frac{\theta}{2} \end{bmatrix} , \mathbf{q}_\psi \triangleq \begin{bmatrix} 0 \\ 0 \\ \sin\frac{\psi}{2} \\ \cos\frac{\psi}{2} \end{bmatrix} . \tag{39}$$

To show that (38) and (35) are consistent using the definitions in (39), we need only apply (9) and double-angle trigonometry identities to (39). For example, letting $\mathbf{k} \triangleq \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^\mathsf{T}$, we see that

$$\begin{aligned}
\mathbf{R}(\mathbf{q}_\psi) &= \left(2\cos^2(\frac{\psi}{2}) - 1\right)\mathbf{I} - 2\cos(\frac{\psi}{2})\sin(\frac{\psi}{2})\lfloor\mathbf{k}\rfloor \\
&\quad + 2\sin^2(\frac{\psi}{2})\mathbf{k}\mathbf{k}^\mathsf{T} \\
&= \cos\psi\mathbf{I} - \sin\psi\lfloor\mathbf{k}\rfloor + (1-\cos\psi)\,\mathbf{k}\mathbf{k}^\mathsf{T} \\
&= \mathbf{R}_\psi \ .
\end{aligned}$$

By substituting (39) into (38), a unit quaternion can be constructed from roll, pitch, and yaw angles as

$$\begin{aligned}
q_x &= \cos\frac{\psi}{2}\cos\frac{\theta}{2}\sin\frac{\phi}{2} - \sin\frac{\psi}{2}\sin\frac{\theta}{2}\cos\frac{\phi}{2} , \\
q_y &= \cos\frac{\psi}{2}\sin\frac{\theta}{2}\cos\frac{\phi}{2} + \sin\frac{\psi}{2}\cos\frac{\theta}{2}\sin\frac{\phi}{2} , \\
q_z &= \sin\frac{\psi}{2}\cos\frac{\theta}{2}\cos\frac{\phi}{2} - \cos\frac{\psi}{2}\sin\frac{\theta}{2}\sin\frac{\phi}{2} , \\
q_0 &= \cos\frac{\psi}{2}\cos\frac{\theta}{2}\cos\frac{\phi}{2} + \sin\frac{\psi}{2}\sin\frac{\theta}{2}\sin\frac{\phi}{2} .
\end{aligned}$$

Appendix A derives the Jacobian that relates errors in the Euler angle decomposition to the attitude error state. Defining

$$\boldsymbol{\Delta} \triangleq \begin{bmatrix} \phi - \hat{\phi} \\ \theta - \hat{\theta} \\ \psi - \hat{\psi} \end{bmatrix} ,$$

the covariance of $\boldsymbol{\Delta}$ is related to the covariance of $\delta\boldsymbol{\theta}$ as

$$E\left[\boldsymbol{\Delta}\boldsymbol{\Delta}^\mathsf{T}\right] = \mathbf{N}_{\boldsymbol{\Delta}} E\left[\delta\boldsymbol{\theta}\delta\boldsymbol{\theta}^\mathsf{T}\right]\mathbf{N}_{\boldsymbol{\Delta}}^\mathsf{T} ,$$

where $\mathbf{N}_{\boldsymbol{\Delta}} = \partial\boldsymbol{\Delta}/\partial\delta\boldsymbol{\theta}$ is given for the iRN case by (100) in Appendix A.

3.8 Error Quaternion Properties

This section presents several properties of error quaternions that are needed in the derivation of the Jacobians used in the RMEKF. The first of these properties relates to the product of two error quaternions. Using (32) and (3a), and by dropping second-order terms, we obtain

$$\begin{aligned}
\delta\mathbf{q}_c &= \delta\mathbf{q}_a \otimes \delta\mathbf{q}_b \\
\begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\theta}_c \\ 1 \end{bmatrix} &= \begin{bmatrix} \mathbf{I} + \frac{1}{2}\lfloor\delta\boldsymbol{\theta}_a\rfloor & \frac{1}{2}\delta\boldsymbol{\theta}_a \\ -\frac{1}{2}\delta\boldsymbol{\theta}_a^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\theta}_b \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\theta}_a + \frac{1}{4}\lfloor\delta\boldsymbol{\theta}_a\rfloor\delta\boldsymbol{\theta}_b + \frac{1}{2}\delta\boldsymbol{\theta}_b \\ -\frac{1}{4}\delta\boldsymbol{\theta}_a^\mathsf{T}\delta\boldsymbol{\theta}_b + 1 \end{bmatrix} \\
&\approx \begin{bmatrix} \frac{1}{2}\left(\delta\boldsymbol{\theta}_a + \delta\boldsymbol{\theta}_b\right) \\ 1 \end{bmatrix} ,
\end{aligned}$$

from which we can conclude that

$$\begin{aligned}
&\delta\mathbf{q}_c = \delta\mathbf{q}_a \otimes \delta\mathbf{q}_b \\
\implies &\delta\boldsymbol{\theta}_c \approx \delta\boldsymbol{\theta}_a + \delta\boldsymbol{\theta}_b \ .
\end{aligned} \tag{40}$$

It can be similarly shown that

$$\begin{aligned}
\delta\mathbf{q}_c = (\delta\mathbf{q}_a)^{-1} \otimes \delta\mathbf{q}_b &\implies \delta\boldsymbol{\theta}_c \approx -\delta\boldsymbol{\theta}_a + \delta\boldsymbol{\theta}_b , \\
\delta\mathbf{q}_c = \delta\mathbf{q}_a \otimes (\delta\mathbf{q}_b)^{-1} &\implies \delta\boldsymbol{\theta}_c \approx \delta\boldsymbol{\theta}_a - \delta\boldsymbol{\theta}_b .
\end{aligned}$$

For the next property, we recall from Section 3.2 that quaternion conjugation operating on an arbitrary quaternion rotates the basis of the vector portion of that quaternion. Using equation (13) with (32), we see that

$$\begin{aligned}
&\delta\mathbf{q}_b = \mathbf{q}^{-1} \otimes \delta\mathbf{q}_a \otimes \mathbf{q} \\
\implies &\delta\boldsymbol{\theta}_b \approx \mathbf{R}(\mathbf{q})\delta\boldsymbol{\theta}_a \ .
\end{aligned} \tag{41}$$

In addition, (40) and (41) can be combined to verify that, for example,

$$\begin{aligned}
&\delta\mathbf{q}_c = \delta\mathbf{q}_a \otimes \mathbf{q}^{-1} \otimes \delta\mathbf{q}_b \otimes \mathbf{q} \\
\implies &\delta\boldsymbol{\theta}_c \approx \delta\boldsymbol{\theta}_a + \mathbf{R}(\mathbf{q})\delta\boldsymbol{\theta}_b \ .
\end{aligned} \tag{42}$$

4 Multiplicative Extended Kalman Filter

With the quaternion notation and properties established, we are prepared to outline the proposed estimation framework: a continuous-discrete, indirect, multiplicative extended Kalman filter (MEKF). For generality, in this section we derive the MEKF without defining the specific state, input, and measurement variables. This derivation will be made more concrete in Sections 6 and 7, where the actual implementations with specific variable definitions are presented. This section is predominantly a review of techniques found in the literature, but introduces notation that helps to clarify the derivation.

A Kalman filter provides the optimal, maximum-likelihood state estimate for a linear system under Gaussian noise. It recursively estimates the evolution of the system state $\mathbf{x}$ as a function of the current state estimate $\hat{\mathbf{x}} = E[\mathbf{x}]$, input $\mathbf{u}$, and measurement $\mathbf{z}$. A Kalman filter also maintains an estimate of the state uncertainty, represented by the covariance matrix $\mathbf{P}$, typically defined as

$$\mathbf{P} = E\left[\left(\mathbf{x} - E\left[\mathbf{x}\right]\right)\left(\mathbf{x} - E\left[\mathbf{x}\right]\right)^\mathsf{T}\right] . \tag{43}$$

The extended Kalman filter (EKF) is an extension of the Kalman filter for systems with nonlinear dynamics and/or nonlinear measurement models. The EKF linearizes the system about the current maximum-likelihood state estimate. While optimality and convergence are no longer guaranteed as opposed to a linear Kalman filter, EKFs are widely used in practice for their robust performance and straightforward implementation. If the state $\mathbf{x}$ includes a quaternion, however, (43) is fundamentally flawed. First, quaternion subtraction is not well-defined as described in Section 3.6, and second, (43) is never full rank because quaternions are not a minimal representation. These issues are addressed using an error-state, or indirect, formulation of the Kalman filter.

The indirect Kalman filter tracks the error state $\delta\mathbf{x}$ and its uncertainty. Unlike the state $\mathbf{x}$, the error state is defined as an element of a vector space by using a minimal attitude representation. The error state is a measure of the discrepancy between the true state $\mathbf{x}$ and a nominal state $\mathbf{x}_\text{nom}$, where $\mathbf{x}_\text{nom}$ can be defined in a number of different ways, as described by Farrell (2008). When the system dynamics are especially well-modeled, such as for a spacecraft in orbit, $\mathbf{x}_\text{nom}$ may be a predetermined feedforward state estimate. More commonly, the nominal state is the maximum likelihood state estimate $\mathbf{x}_\text{nom} = E\left[\mathbf{x}\right]$. In this case, measurements provide feedback to update the nominal state, forcing the expected value of the error state to zero. Some indirect Kalman filter implementations differentiate between *fast* and *slow* measurements, and only update the nominal state for the slow measurements (Farrell, 2008; Maybeck, 1979). For such systems, $\mathbf{x}_\text{nom} \neq E\left[\mathbf{x}\right]$ at the fast rate, so that the expected error state is non-zero and must be propagated. For the derivation in this paper, the nominal state is updated equivalently for every measurement, ensuring that $\mathbf{x}_\text{nom} = \hat{\mathbf{x}} = E[\mathbf{x}]$ at any given time.

Let $\mathbf{x}_\mathbf{v}$ and $\mathbf{x}_\mathbf{q}$ represent vector and quaternion portions of the state $\mathbf{x}$, and $\delta\mathbf{x}_\mathbf{v}$ and $\delta\mathbf{x}_{\boldsymbol{\theta}}$ be the corresponding elements of the error-state $\delta\mathbf{x}$. We define these error-state elements as

$$\delta\mathbf{x}_\mathbf{v} \triangleq \mathbf{x}_\mathbf{v} - \hat{\mathbf{x}}_\mathbf{v} \ , \tag{44a}$$

$$\delta\mathbf{x}_{\boldsymbol{\theta}} \triangleq \delta\mathbf{q}^\vee \ , \tag{44b}$$

where $\delta\mathbf{q}$ is defined by (30) using $\mathbf{x}_\mathbf{q}$ and $\hat{\mathbf{x}}_\mathbf{q}$. This results in the property

$$E[\delta\mathbf{x}] = \mathbf{0} \ , \tag{45}$$

as derived in Appendix B.1. Because the error state is part of a vector space, and as a result of (45), the indirect Kalman filter represents state uncertainty with the well-defined covariance

$$\begin{aligned}\mathbf{P} &= E\left[\left(\delta\mathbf{x} - E\left[\delta\mathbf{x}\right]\right)\left(\delta\mathbf{x} - E\left[\delta\mathbf{x}\right]\right)^\mathsf{T}\right] \\ &= E\left[\delta\mathbf{x}\delta\mathbf{x}^\mathsf{T}\right] \ .\end{aligned} \tag{46}$$

As a note on interpretation, in this formulation the unknown true state is modeled as a random variable centered around the current state estimate. More specifically, the vector and quaternion portions of the true state are modeled as random variables using the inverse of (44) as

$$\begin{aligned} & \mathbf{x}_\mathbf{v} = \hat{\mathbf{x}}_\mathbf{v} + \delta\mathbf{x}_\mathbf{v} \ , \\ \text{iRN:} \quad & \mathbf{x}_\mathbf{q} = \hat{\mathbf{x}}_\mathbf{q} \otimes \left(\delta\mathbf{x}_{\boldsymbol{\theta}}\right)^\wedge \ , \\ \text{bRN:} \quad & \mathbf{x}_\mathbf{q} = \left(\delta\mathbf{x}_{\boldsymbol{\theta}}\right)^\wedge \otimes \hat{\mathbf{x}}_\mathbf{q} \ , \end{aligned}$$

where $\delta\mathbf{x}_\mathbf{v}$ and $\delta\mathbf{x}_{\boldsymbol{\theta}}$ are elements of the Gaussian random vector

$$\delta\mathbf{x} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{P}\right) \ .$$

Kalman filters are decomposed into two steps: the *propagation* step and the *update* step, which are described in Sections 4.1 and 4.2. Section 4.3 discusses a method to handle delayed or out-of-order measurements. In general, the MEKF can use either an inertial or body-fixed coordinate frame. Several nuanced differences exist, however, and are highlighted as iRN- or bRN-specific.

4.1 Propagation

Consider the continuous-time system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u} + \boldsymbol{\upsilon}) + \boldsymbol{\eta} \ , \tag{47}$$

where $\boldsymbol{\upsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_\mathbf{u})$ and $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_\mathbf{x})$ are zero-mean Gaussian random variables. More specifically, we assume that $\boldsymbol{\eta}$ and $\boldsymbol{\upsilon}$ are uncorrelated,

$$E\left[\boldsymbol{\eta}\boldsymbol{\upsilon}^\mathsf{T}\right] = \mathbf{0} \ , \tag{48}$$

and that the input and process noise are not correlated in time:

$$\begin{aligned} E\left[\boldsymbol{\eta}(t)\boldsymbol{\eta}(\tau)^\mathsf{T}\right] &= \mathbf{Q}_\mathbf{x}\delta(t-\tau) \ , \\ E\left[\boldsymbol{\upsilon}(t)\boldsymbol{\upsilon}(\tau)^\mathsf{T}\right] &= \mathbf{Q}_\mathbf{u}\delta(t-\tau) \ , \end{aligned} \tag{49}$$

where $\delta(t)$ is the Dirac delta function.

In the propagation step, the filter propagates the state estimates forward according to the nominal propagation dynamics as

$$\dot{\hat{\mathbf{x}}} = \mathbf{f}(\hat{\mathbf{x}}, \mathbf{u}) \ . \tag{50}$$

When $\mathbf{x}$ is part of a vector space, the Jacobians needed for propagating the covariance are obtained from the first-order Taylor series expansion of the error-state dynamics as

$$\begin{aligned} \delta\dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \mathbf{u} + \boldsymbol{\upsilon}) + \boldsymbol{\eta} - \mathbf{f}(\hat{\mathbf{x}}, \mathbf{u}) && \text{(51a)} \\ &\approx \mathbf{f}(\hat{\mathbf{x}}, \mathbf{u}) + \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}},\mathbf{u}} \delta\mathbf{x} + \left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_{\hat{\mathbf{x}},\mathbf{u}} \boldsymbol{\upsilon} + \boldsymbol{\eta} - \mathbf{f}(\hat{\mathbf{x}}, \mathbf{u}) \\ &= \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}},\mathbf{u}} \delta\mathbf{x} + \left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_{\hat{\mathbf{x}},\mathbf{u}} \boldsymbol{\upsilon} + \boldsymbol{\eta} \ . && \text{(51b)} \end{aligned}$$

When the state $\mathbf{x}$ includes quaternion terms, however, the error-state dynamics cannot be formed by simple subtraction as in (51a). In fact, as described in Section 3.6, the error state $\delta\mathbf{x}$ is of lower dimension than $\mathbf{x}$. Noting that $\mathbf{x}$ is a function of $\hat{\mathbf{x}}$ and $\delta\mathbf{x}$ according to (44), the dynamics of the error state can be expressed generally as a single function

$$\delta\dot{\mathbf{x}} = \bar{\mathbf{f}}(\delta\mathbf{x}, \boldsymbol{\upsilon}, \hat{\mathbf{x}}, \mathbf{u}) + \boldsymbol{\eta} \ , \tag{52}$$

where only $\delta\mathbf{x}$, $\boldsymbol{\upsilon}$, and $\boldsymbol{\eta}$ are stochastic variables[†]. The function $\bar{\mathbf{f}}$ then handles the quaternion portions of the state appropriately. We show in Appendix B.2 that $\bar{\mathbf{f}}(E[\delta\mathbf{x}], E[\boldsymbol{\upsilon}], \hat{\mathbf{x}}, \mathbf{u}) = \mathbf{0}$. The Jacobians for the covariance propagation are computed from the first-order Taylor series expansion of (52) about $(E[\delta\mathbf{x}], E[\boldsymbol{\upsilon}], \hat{\mathbf{x}}, \mathbf{u})$ as

$$\begin{aligned} \delta\dot{\mathbf{x}} &\approx \bar{\mathbf{f}}(E[\delta\mathbf{x}], E[\boldsymbol{\upsilon}], \hat{\mathbf{x}}, \mathbf{u}) + \boldsymbol{\eta} \\ &\quad + \mathbf{F}(\delta\mathbf{x} - E[\delta\mathbf{x}]) + \mathbf{G}(\boldsymbol{\upsilon} - E[\boldsymbol{\upsilon}]) \\ &= \mathbf{F}\delta\mathbf{x} + \mathbf{G}\boldsymbol{\upsilon} + \boldsymbol{\eta} \ , \end{aligned} \tag{53}$$

where

$$\mathbf{F} = \left.\frac{\partial \bar{\mathbf{f}}(\delta\mathbf{x}, \boldsymbol{\upsilon}, \hat{\mathbf{x}}, \mathbf{u})}{\partial \delta\mathbf{x}}\right|_{E[\delta\mathbf{x}], E[\boldsymbol{\upsilon}], \hat{\mathbf{x}}, \mathbf{u}}$$

and

$$\mathbf{G} = \left.\frac{\partial \bar{\mathbf{f}}(\delta\mathbf{x}, \boldsymbol{\upsilon}, \hat{\mathbf{x}}, \mathbf{u})}{\partial \boldsymbol{\upsilon}}\right|_{E[\delta\mathbf{x}], E[\boldsymbol{\upsilon}], \hat{\mathbf{x}}, \mathbf{u}} ,$$

with $E[\delta\mathbf{x}] = E[\boldsymbol{\upsilon}] = \mathbf{0}$. Note that (53) has the form of (51b), but the Jacobian terms differentiate the error-state dynamics with respect to the error state and input noise, rather than with respect to the state and input.

The error-state covariance propagation is given by differentiating (46) with respect to time and utilizing the linearized error dynamics from (53) as

$$\begin{aligned} \dot{\mathbf{P}} &= E\left[\delta\dot{\mathbf{x}}\delta\mathbf{x}^\mathsf{T} + \delta\mathbf{x}\delta\dot{\mathbf{x}}^\mathsf{T}\right] \\ &= E\left[\mathbf{F}\delta\mathbf{x}\delta\mathbf{x}^\mathsf{T} + \mathbf{G}\boldsymbol{\upsilon}\delta\mathbf{x}^\mathsf{T} + \boldsymbol{\eta}\delta\mathbf{x}^\mathsf{T}\right] \\ &\quad + E\left[\delta\mathbf{x}\delta\mathbf{x}^\mathsf{T}\mathbf{F}^\mathsf{T} + \delta\mathbf{x}\boldsymbol{\upsilon}^\mathsf{T}\mathbf{G}^\mathsf{T} + \delta\mathbf{x}\boldsymbol{\eta}^\mathsf{T}\right] \\ &= \mathbf{F}\mathbf{P} + \mathbf{P}\mathbf{F}^\mathsf{T} + E\left[\mathbf{G}\boldsymbol{\upsilon}\delta\mathbf{x}^\mathsf{T} + \delta\mathbf{x}\boldsymbol{\upsilon}^\mathsf{T}\mathbf{G}^\mathsf{T}\right] \\ &\quad + E\left[\boldsymbol{\eta}\delta\mathbf{x}^\mathsf{T} + \delta\mathbf{x}\boldsymbol{\eta}^\mathsf{T}\right] \ . \end{aligned} \tag{54}$$

To simplify terms, we solve the differential equation in (53) with initial conditions $\delta\mathbf{x}_0$ to obtain

$$\delta\mathbf{x}(t) = e^{\mathbf{F}t}\delta\mathbf{x}_0 + \int_0^t e^{\mathbf{F}(t-\tau)}\mathbf{G}\boldsymbol{\upsilon}(\tau)d\tau + \int_0^t e^{\mathbf{F}(t-\tau)}\boldsymbol{\eta}(\tau)d\tau \ . \tag{55}$$

Using (55) and the properties (48) and (49), we see that

$$\begin{aligned} E\left[\delta\mathbf{x}\boldsymbol{\upsilon}^\mathsf{T}\mathbf{G}^\mathsf{T}\right] &= E\left[e^{\mathbf{F}t}\delta\mathbf{x}_0\boldsymbol{\upsilon}^\mathsf{T}\mathbf{G}^\mathsf{T}\right] \\ &\quad + E\left[\int_0^t e^{\mathbf{F}(t-\tau)}\mathbf{G}\boldsymbol{\upsilon}(\tau)\boldsymbol{\upsilon}^\mathsf{T}(t)\mathbf{G}^\mathsf{T}d\tau\right] \\ &\quad + E\left[\int_0^t e^{\mathbf{F}(t-\tau)}\boldsymbol{\eta}(\tau)\boldsymbol{\upsilon}^\mathsf{T}(t)\mathbf{G}^\mathsf{T}d\tau\right] \\ &= E\left[\int_0^t e^{\mathbf{F}(t-\tau)}\mathbf{G}\mathbf{Q}_\mathbf{u}\delta(t-\tau)\mathbf{G}^\mathsf{T}d\tau\right] \\ &= \frac{1}{2}\mathbf{G}\mathbf{Q}_\mathbf{u}\mathbf{G}^\mathsf{T} \end{aligned} \tag{56}$$

where the $\frac{1}{2}$ is because the bounds of integration only use half of the area inside of the delta function. Similarly,

$$E\left[\delta\mathbf{x}\boldsymbol{\eta}^\mathsf{T}\right] = \frac{1}{2}\mathbf{Q}_\mathbf{x} \ . \tag{57}$$

Because $\mathbf{Q}_\mathbf{u}$ and $\mathbf{Q}_\mathbf{x}$ are symmetric, combining (54), (56), and (57) we have that $\mathbf{P}$ evolves between measurements as

$$\dot{\mathbf{P}} = \mathbf{F}\mathbf{P} + \mathbf{P}\mathbf{F}^\mathsf{T} + \mathbf{G}\mathbf{Q}_\mathbf{u}\mathbf{G}^\mathsf{T} + \mathbf{Q}_\mathbf{x} \ . \tag{58}$$

In summary, during the propagation step, $\hat{\mathbf{x}}$ is propagated forward using (50) and the error covariance is propagated forward using (58). Also, since the evolution of $\delta\mathbf{x}$ is given by (53) and $E[\delta\mathbf{x}](0) = \mathbf{0}$, we have that $E[\delta\mathbf{x}](t) = \mathbf{0}$ over the propagation window.

4.2 Measurement Update

For the *update* step, consider the measurement

$$\mathbf{z} = \mathbf{h}(\mathbf{x}, \boldsymbol{\eta}_\mathbf{z}) \ , \tag{59}$$

where $\boldsymbol{\eta}_\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ represents measurement noise. This measurement noise is usually additive if the measurement is a vector quantity, but if the measured quantity is a quaternion the noise is applied through quaternion multiplication.

The residual $\mathbf{r}$ is the discrepancy between the true measurement and the predicted measurement

$$\hat{\mathbf{z}} = \mathbf{h}(\hat{\mathbf{x}}, \mathbf{0}) \ .$$

Conventionally a vector-space measurement is assumed, such that (59) is simplified as

$$\mathbf{z} = \mathbf{h}(\mathbf{x}) + \boldsymbol{\eta}_\mathbf{z} \ .$$

In this case, the residual is found by subtraction,

$$\mathbf{r} = \mathbf{z} - \mathbf{h}(\hat{\mathbf{x}}) \ ,$$

[†] Note our slight abuse of notation in that $\boldsymbol{\eta}$ in (52) is of the same dimensionality as the error vector, while in (47) $\boldsymbol{\eta}$ is of the same dimensionality as the state vector.

and is modeled as

$$\mathbf{r} = \mathbf{h}(\mathbf{x}) + \boldsymbol{\eta}_\mathbf{z} - \mathbf{h}(\hat{\mathbf{x}}) \ .$$

Jacobians are then computed using the first-order Taylor-series expansion as

$$\begin{aligned} \mathbf{r} &\approx \mathbf{h}(\hat{\mathbf{x}}) + \left.\frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}} \delta\mathbf{x} + \boldsymbol{\eta}_\mathbf{z} - \mathbf{h}(\hat{\mathbf{x}}) \\ &= \left.\frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}} \delta\mathbf{x} + \boldsymbol{\eta}_\mathbf{z} \ . \end{aligned} \tag{60}$$

These measurement and residual models share similar shortcomings with the conventional error-state dynamics presented in (51), namely they do not hold for quaternion measurements and (60) assumes the state $\mathbf{x}$ is of the same dimensionality as $\delta\mathbf{x}$.

These issues can be addressed by expressing the residual model as a single function, paralleling the approach taken for the propagation step in Section 4.1. Again noting that $\mathbf{x}$ is a function of $\hat{\mathbf{x}}$ and $\delta\mathbf{x}$, the residual is modeled as a function

$$\mathbf{r} = \bar{\mathbf{h}}(\delta\mathbf{x}, \boldsymbol{\eta}_\mathbf{z}, \hat{\mathbf{x}}) \ ,$$

where only the first two variables are stochastic. For measurements $\mathbf{z}_\mathbf{v}$ of vector values, the residual value is computed by subtraction as

$$\mathbf{r} = \mathbf{z}_\mathbf{v} - \mathbf{h}(\hat{\mathbf{x}}, \mathbf{0}) \ , \tag{61}$$

and is modeled as

$$\bar{\mathbf{h}}(\delta\mathbf{x}, \boldsymbol{\eta}_\mathbf{z}, \hat{\mathbf{x}}) = \mathbf{h}(\mathbf{x}, \boldsymbol{\eta}_\mathbf{z}) - \mathbf{h}(\hat{\mathbf{x}}, \mathbf{0}) \ , \tag{62}$$

where $\mathbf{x}$ can be rewritten in terms of $\delta\mathbf{x}$ and $\hat{\mathbf{x}}$ according to (29) and by rearranging (44a). For measurements $\mathbf{z}_\mathbf{q}$ of quaternion values, the residual value is computed as the three-vector minimal representation of the error between the observed and expected quaternions according to (30) as

$$\begin{aligned} \text{iRN:} \quad \mathbf{r} &= \left(\mathbf{h}(\hat{\mathbf{x}}, \mathbf{0})^{-1} \otimes \mathbf{z}_\mathbf{q}\right)^\vee , \\ \text{bRN:} \quad \mathbf{r} &= \left(\mathbf{z}_\mathbf{q} \otimes \mathbf{h}(\hat{\mathbf{x}}, \mathbf{0})^{-1}\right)^\vee , \end{aligned} \tag{63}$$

and is modeled as

$$\begin{aligned} \text{iRN:} \quad \bar{\mathbf{h}}(\delta\mathbf{x}, \boldsymbol{\eta}_\mathbf{z}, \hat{\mathbf{x}}) &= \left(\mathbf{h}(\hat{\mathbf{x}}, \mathbf{0})^{-1} \otimes \mathbf{h}(\mathbf{x}, \boldsymbol{\eta}_\mathbf{z})\right)^\vee , \\ \text{bRN:} \quad \bar{\mathbf{h}}(\delta\mathbf{x}, \boldsymbol{\eta}_\mathbf{z}, \hat{\mathbf{x}}) &= \left(\mathbf{h}(\mathbf{x}, \boldsymbol{\eta}_\mathbf{z}) \otimes \mathbf{h}(\hat{\mathbf{x}}, \mathbf{0})^{-1}\right)^\vee , \end{aligned} \tag{64}$$

where again $\mathbf{x}$ can be rewritten in terms of $\delta\mathbf{x}$ and $\hat{\mathbf{x}}$ according to (29) and (44a).

It follows from (62) and (64) that the residual models have the property $\bar{\mathbf{h}}(E[\delta\mathbf{x}], E[\boldsymbol{\eta}_\mathbf{z}], \hat{\mathbf{x}}) = \mathbf{0}$. The measurement models in this paper are chosen so that the noise is additive in the residual space, implying that $\frac{\partial \mathbf{r}}{\partial \boldsymbol{\eta}_\mathbf{z}} = \mathbf{I}$. The measurement Jacobians are computed using the first-order Taylor-series expansion of $\bar{\mathbf{h}}$ about $(E[\delta\mathbf{x}], E[\boldsymbol{\eta}_\mathbf{z}], \hat{\mathbf{x}})$ as

$$\begin{aligned} \mathbf{r} &\approx \bar{\mathbf{h}}(E[\delta\mathbf{x}], E[\boldsymbol{\eta}_\mathbf{z}], \hat{\mathbf{x}}) + \mathbf{H}\left(\delta\mathbf{x} - E[\delta\mathbf{x}]\right) \\ &\quad + \mathbf{I}(\boldsymbol{\eta}_\mathbf{z} - E[\boldsymbol{\eta}_\mathbf{z}]) \\ &= \mathbf{H}\delta\mathbf{x} + \boldsymbol{\eta}_\mathbf{z} \ , \end{aligned}$$

where

$$\mathbf{H} = \left.\frac{\partial \bar{\mathbf{h}}(\delta\mathbf{x}, \boldsymbol{\eta}_\mathbf{z}, \hat{\mathbf{x}})}{\partial \delta\mathbf{x}}\right|_{E[\delta\mathbf{x}], E[\boldsymbol{\eta}_\mathbf{z}], \hat{\mathbf{x}}}$$

and $E[\delta\mathbf{x}] = E[\boldsymbol{\eta}_\mathbf{z}] = \mathbf{0}$.

Assuming the measurement noise, error state, and input noise are uncorrelated, the residual uncertainty is

$$\begin{aligned} \mathbf{S} &\triangleq E[\mathbf{r}\mathbf{r}^\mathsf{T}] \\ &= E[\mathbf{H}\delta\mathbf{x}\delta\mathbf{x}^\mathsf{T}\mathbf{H}^\mathsf{T} + \boldsymbol{\eta}_\mathbf{z}\boldsymbol{\eta}_\mathbf{z}^\mathsf{T}] \\ &= \mathbf{H}\mathbf{P}\mathbf{H}^\mathsf{T} + \mathbf{R} \ . \end{aligned} \tag{65}$$

The Kalman gain uses the residual and state uncertainty to find the extent to which the residual should be trusted and applied. Using the residual covariance, the Kalman gain is

$$\mathbf{K} = \mathbf{P}\mathbf{H}^\mathsf{T}\mathbf{S}^{-1} \ . \tag{66}$$

In fusing the information provided by the measurement, the *a posteriori* estimate of the error state, denoted with a $^+$, is

$$\delta\mathbf{x}^+ = \delta\mathbf{x} + \mathbf{K}\mathbf{r} \ ,$$

implying that

$$E[\delta\mathbf{x}^+] = \mathbf{K}\mathbf{r} \ .$$

With the additional information provided by the measurement update, the error state is no longer zero-mean, violating the property in (45). Let $\mathbf{x}_\mathbf{v}$ and $\mathbf{x}_\mathbf{q}$ again be vector and quaternion states within $\mathbf{x}$, and let $\Delta\mathbf{v}$ and $\Delta\boldsymbol{\theta}$ be the corresponding portions of the Kalman update $\mathbf{K}\mathbf{r}$. To ensure the error state remains zero mean, the Kalman update $\mathbf{K}\mathbf{r}$ is used to adjust $\hat{\mathbf{x}}$ as

$$\begin{aligned} \hat{\mathbf{x}}_\mathbf{v}^+ &= \hat{\mathbf{x}}_\mathbf{v} + \Delta\mathbf{v} \\ \text{iRN:} \quad \hat{\mathbf{x}}_\mathbf{q}^+ &= \hat{\mathbf{x}}_\mathbf{q} \otimes \Delta\boldsymbol{\theta}^\wedge && \text{(67a)} \\ \text{bRN:} \quad \hat{\mathbf{x}}_\mathbf{q}^+ &= \Delta\boldsymbol{\theta}^\wedge \otimes \hat{\mathbf{x}}_\mathbf{q} \ . && \text{(67b)} \end{aligned}$$

where (67a) and (67b) are specific to the quaternion error-state definition used.

Finally, the covariance is updated conventionally as

$$\mathbf{P}^+ = (\mathbf{I} - \mathbf{K}\mathbf{H})\,\mathbf{P} \ .$$

In practice we use the Joseph form Kalman update,

$$\mathbf{P}^+ = (\mathbf{I} - \mathbf{K}\mathbf{H})\,\mathbf{P}\,(\mathbf{I} - \mathbf{K}\mathbf{H})^\mathsf{T} + \mathbf{K}\mathbf{R}\mathbf{K}^\mathsf{T} \ , \tag{68}$$

because it improves numerical stability and ensures that the covariance matrix remains symmetric (Bar-Shalom et al., 2002).

In summary, during the update step the measurement residual (61) or (63) provides additional information causing a non-zero $E[\delta\mathbf{x}]$. Using the Kalman gain (66), $\hat{\mathbf{x}}$ is updated according to (67) to ensure $E[\delta\mathbf{x}]$ remains zero mean. The error-state covariance is updated according to (68).

4.3 Delayed Out-of-Order Measurements

The Kalman filter assumes the state evolves according to a Markov process. As such, state estimates for all previous time steps are marginalized. This makes it difficult to compute a residual for delayed or out-of-order measurements when they finally arrive. In practice, delayed measurements are common. For example, visual odometry algorithms may require hundreds of milliseconds to perform the computer vision operations necessary to compute a measurement. Ideally this measurement is used to constrain the state of the vehicle when the image was taken, not to constrain the state of the vehicle when the measurement arrives.

To address delayed measurements, we use priority queues to save the last T seconds of inputs, measurements, states, and error-state covariances ordered by time. If a delayed measurement arrives with a time stamp more than T seconds old, we discard the measurement. Otherwise, we discard all saved states and error-state covariances that have a time stamp later than the time stamp of the incoming delayed measurement. At this point, we are left with the state and covariance estimate at the instant the delayed measurement should have arrived. We apply the measurement normally, and then use the input and measurement queues to re-propagate the MEKF to the current time instance. This approach provides the same state estimate as if all measurements had arrived at the correct time. Handling delayed messages in this way may not be practical for all processors. Similar methods are described by Bopardikar et al. (2013) and Shen et al. (2014).

5 Relative Navigation

Section 4 provided a general overview of the indirect MEKF. This section describes how the MEKF is adapted to the relative navigation framework.

Conventional filtering approaches directly estimate the vehicle's global state with respect to some inertially-fixed origin, such as the GPS origin or the vehicle's starting location; however, when only relative position measurements such as those obtained from visual odometry or laser scan matching are available, the vehicle's global position and heading are unobservable (Martinelli, 2012; Weiss et al., 2012; Jones et al., 2007). Over time, directly estimating these unobservable states leads to inconsistent and unbounded state uncertainties, which can degrade accuracy and cause irregular state jumps in the filter (Julier and Uhlmann, 2001; Bailey and Durrant-Whyte, 2006; Kottas et al., 2013). Methods for mitigating these issues have been proposed (Bailey and Durrant-Whyte, 2006; Kottas et al., 2013; Castellanos et al., 2007), but the core underlying issue of unobservable states can be avoided entirely by reformulating the problem in terms of relative states (Wheeler et al., 2018).

The relative navigation approach maintains observability of the filter states by estimating the pose of the vehicle with respect to a local coordinate frame referred to as the *node frame*. This node frame is positioned at zero altitude directly below the most recent odometry keyframe, but is gravity-aligned (i.e. the heading is aligned with the vehicle's heading when the keyframe was declared, but there is no pitch or roll). As a result, the odometry provides nearly-direct measurements of the position and heading of the vehicle with respect to the current node frame, making those states observable by construction.

Because of the way the node frame is defined, the roll and pitch components of the vehicle's attitude (ϕ, θ), as well as the vehicle's altitude p_z, are estimated as if they were defined with respect to a global origin. These states are not affected when transitioning from one node frame to another and so are, in effect, independent of the current node frame. On the other hand, the horizontal position and heading states (p_x, p_y, and ψ) define how the vehicle has moved since the last node frame, and are termed *relative states.* Each time a new keyframe is declared, a new node frame is also declared and the relative states (p_x, p_y, and ψ) are reset to zero. The covariances associated with the relative states are also reset to zero, since the vehicle is, by definition, at the location of the node frame and so there is no uncertainty in these states. This is illustrated in Figure 4. The non-relative states (roll, pitch, altitude, body-fixed velocities, and body-fixed IMU biases) and their covariances are unchanged by the keyframe reset operation. Note that resetting the heading component of the state and covariance is non-trivial when attitude is parameterized with a quaternion. Prior to the reset, the vehicle's current relative pose estimate and covariance are passed to a back-end pose-graph map that concatenates the relative poses into an estimate of the vehicle's global path and current global pose (Wheeler et al., 2018).

We describe the keyframe reset operation mathematically as follows. Let $\mathbf{n}(\mathbf{x})$ define the keyframe reset operation. The estimated state is reset as

$$\hat{\mathbf{x}}^+ = \mathbf{n}(\hat{\mathbf{x}}) \,. \tag{69}$$

The error state after the reset, $\delta\mathbf{x}^+$, is the difference between $\mathbf{x}^+$ and $\hat{\mathbf{x}}^+$ as defined by (44). Again recalling that since $\mathbf{x}$ is a function of $\delta\mathbf{x}$ and $\hat{\mathbf{x}}$, we can express the error state after the reset as a single function

$$\delta\mathbf{x}^+ = \bar{\mathbf{n}}(\delta\mathbf{x}, \hat{\mathbf{x}}) \,.$$

This allows the covariance to be updated as

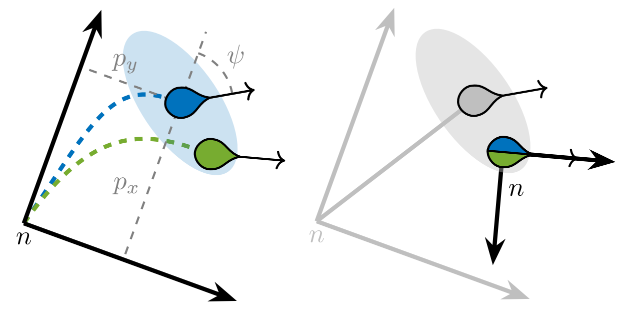
$$\mathbf{P}^+ = \mathbf{N}\mathbf{P}\mathbf{N}^\top \,, \tag{70}$$

where

$$\mathbf{N} = \left.\frac{\partial\bar{\mathbf{n}}(\delta\mathbf{x}, \hat{\mathbf{x}})}{\partial\delta\mathbf{x}}\right|_{E[\delta\mathbf{x}],\hat{\mathbf{x}}} \,.$$

The details of the reset operation $\mathbf{n}(\mathbf{x})$ and the Jacobian $\mathbf{N}$ are presented in Section 6.3 for iRN and in Section 7.3 for bRN.

This formulation provides several advantages in terms of estimator performance. One of these advantages is that when the covariance associated with the relative states is reset to zero, uncertainty is in essence removed



(a) Before reset **(b)** After reset

Fig. 4. Illustration of the keyframe reset operation as viewed from above. Before the reset (4a), there is some error between the estimated pose (blue) and the true pose (green), with the estimated uncertainty represented by the blue covariance ellipse. When the reset occurs, the vehicle is, by definition, at the location of the keyframe. As a result, after the reset (4b), both the relative states and the corresponding elements of the covariance matrix are set identically to zero and there is no error.

Algorithm 1 Relative multiplicative extended Kalman filter (RMEKF)

```
Initialize: $\hat{\mathbf{x}} = \mathbf{x}_0$
Initialize: $\mathbf{P} = \mathbf{P}_0$
for Each new available input $\mathbf{u}$ do
    Propagate nominal state $\hat{\mathbf{x}}$ using (50)
    Propagate error-state covariance $\mathbf{P}$ using (58)
    for $i$ in sensors do
        if Measurement is available from sensor $i$ then
            Compute residual $\mathbf{r}$ using (61) or (63)
            Compute residual uncertainty $\mathbf{S}$ using (65)
            Compute Kalman gain $\mathbf{K}$ using (66)
            Use $\mathbf{Kr}$ to update $\hat{\mathbf{x}}$ using (67)
            Update error-state covariance $\mathbf{P}$ using (68)
    if New keyframe is declared then
        Save $\hat{\mathbf{x}}$ and $\mathbf{P}$ as edge in pose-graph back end
        Reset state using (69)
        Reset uncertainty using (70)
```

from the filter and delegated to the back-end map, which helps to maintain filter consistency (Barfoot and Furgale, 2014). As a result, the covariance in the filter also remains bounded. In addition, since the distances between keyframes are relatively small, the state error remains small, avoiding significant linearization errors that can cause inconsistency in a global estimator (Castellanos et al., 2007).

While reconstructing the global pose estimate requires implementing a back-end pose-graph map, this architecture has also been shown to improve consistency and accuracy of the final global pose estimate over global filter approaches, even in the absence of additional pose-graph constraints such as loop closures (Wheeler et al., 2018). It also has the advantage of avoiding large, potentially destabilizing state jumps when new global information becomes available. The requirement to implement this pose-graph map is also not a particularly onerous one, seeing as some global filter approaches already use a back-end batch-processed map to provide updates to their global filter (Shen et al., 2014; Weiss and Siegwart, 2011).

The various steps for implementing an RMEKF are summarized in Algorithm 1, along with references to the key equations. The specific implementation equations are derived in Section 6 for inertial relative navigation and in Section 7 for body-fixed robocentric relative navigation.

6 Inertial Relative Navigation (iRN)

Inertial relative navigation (iRN) estimates the vehicle's position and attitude with respect to the current node frame. While the current node frame changes regularly, each is gravity-aligned and inertially defined. For this reason, typical UAS dynamics from the GPS/INS literature are applicable. Section 6.1 outlines the input, state, and dynamics for the system, including the error-state dynamics. Section 6.2 defines the measurement models and Section 6.3 outlines the keyframe reset step. In general, the derivations in this section improve upon the mathematical rigor of the derivations in prior work (Leishman and McLain, 2014) and provide some corrections. New contributions that extend that work are noted in their respective sections.

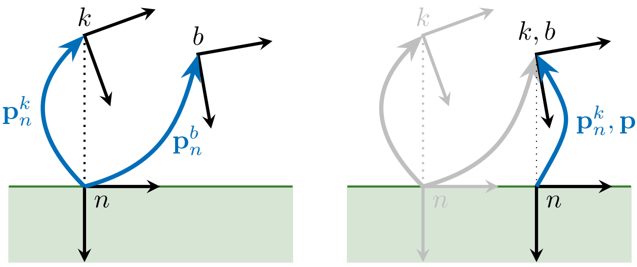
6.1 State Dynamics

This section derives the state propagation model used in the filter. Section 6.1.1 defines the state vector. Section 6.1.2 discusses how measurements from an inertial measurement unit (IMU) are incorporated into the propagation model, and Section 6.1.3 defines the state propagation model and derives the associated Jacobians.

6.1.1 State Vector. Vectors use a forward-right-down coordinate frame, with axes labeled x, y, and z. In this paper, the position vector ${}^{c}\mathbf{p}_{a}^{b}$ denotes the position of frame b with respect to frame a, expressed in frame c. Unless otherwise noted, position vectors are expressed in the originating frame, i.e. frames a and c are the same. When this is the case the prescript is usually omitted for brevity, so that $\mathbf{p}_{a}^{b} \triangleq {}^{a}\mathbf{p}_{a}^{b}$. However, the prescript is occasionally included for clarity.

Inertial relative navigation estimates the position and attitude of the vehicle's body frame b with respect to and expressed in the current node frame n, denoted by the pose $(\mathbf{p}_{n}^{b}, \mathbf{q}_{n}^{b})$. Let frame k represent the vehicle's body frame at the instant in time that the current keyframe image was taken. The estimator tracks the pose of k with respect to and expressed in n, denoted as $(\mathbf{p}_{n}^{k}, \mathbf{q}_{n}^{k})$, which is an extension to prior work (Leishman and McLain, 2014). As illustrated in Figure 5a, the keyframe states $(\mathbf{p}_{n}^{k}, \mathbf{q}_{n}^{k})$ include the altitude, roll, and pitch of the vehicle at the moment a keyframe is declared. Since the vehicle state is not a vector when it contains quaternion elements, we define the state as the tuple

$$\mathbf{x} \triangleq \left(\mathbf{p}_{n}^{b},\ \mathbf{q}_{n}^{b},\ \mathbf{v},\ \boldsymbol{\beta}_{\boldsymbol{\omega}},\ \boldsymbol{\beta}_{\mathbf{a}},\ \mathbf{p}_{n}^{k},\ \mathbf{q}_{n}^{k},\ \mu\right)$$



(a) Before reset **(b)** After reset

Fig. 5. Illustration of the keyframe reset operation as viewed from the side. (a) $\mathbf{p}_n^k$ and $\mathbf{q}_n^k$ encode the roll, pitch, and altitude of the vehicle when the keyframe was declared. (b) After a reset, the keyframe and body states are identical.

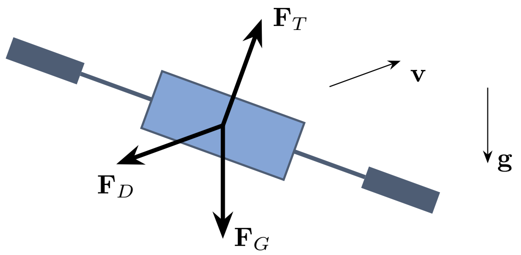


Fig. 6. Free body diagram describing the forces acting on a multirotor. The principal forces are gravity $\mathbf{F}_G$, thrust $\mathbf{F}_T$, and drag $\mathbf{F}_D$. The vector $\mathbf{v}$ represents the velocity of the vehicle, and $\mathbf{g}$ represents the gravity vector. The gravity force acts in the direction of gravity, the thrust force acts perpendicular to the plane of the rotors, and the simplified drag force acts opposite the velocity vector.

with $\mathbf{x} \in \mathbb{R}^3 \times \mathbb{H} \times \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{H} \times \mathbb{R}$, where $\mathbf{v} \in \mathbb{R}^3$ is the vehicle's inertial velocity expressed in the body frame (i.e. $\mathbf{v} \triangleq {}^b\mathbf{v}_{\mathcal{I}}^b$ where $\mathcal{I}$ is an inertial frame), $\boldsymbol{\beta}_{\boldsymbol{\omega}} \in \mathbb{R}^3$ and $\boldsymbol{\beta}_{\mathbf{a}} \in \mathbb{R}^3$ are unknown biases for the gyro and accelerometer, and $\mu \in \mathbb{R}$ is the lateral drag coefficient. The error-state vector $\delta\mathbf{x} \in \mathbb{R}^{22}$ is defined as

$$\delta\mathbf{x} = \begin{bmatrix} \delta\mathbf{p}_n^{b\,\top} & \delta\boldsymbol{\theta}_n^{b\,\top} & \delta\mathbf{v}^\top & \delta\boldsymbol{\beta}_{\boldsymbol{\omega}}^\top & \delta\boldsymbol{\beta}_{\mathbf{a}}^\top & \delta\mathbf{p}_n^{k\,\top} & \delta\boldsymbol{\theta}_n^{k\,\top} & \delta\mu \end{bmatrix}^\top .$$

6.1.2 Mechanization. Let $\boldsymbol{\omega} \triangleq {}^b\boldsymbol{\omega}_{\mathcal{I}}^b$ and $\mathbf{a} \triangleq {}^b\mathbf{a}_{\mathcal{I}}^b$ respectively define the true body-fixed angular rates and accelerations as measured by an ideal IMU. For brevity we omit specifying the frames and write $\boldsymbol{\omega}$ and $\mathbf{a}$ throughout the paper.

An IMU provides measurements $\tilde{\boldsymbol{\omega}}$ and $\tilde{\mathbf{a}}$ that are corrupted by unknown biases $\boldsymbol{\beta}_{\boldsymbol{\omega}}$ and $\boldsymbol{\beta}_{\mathbf{a}}$ and zero-mean Gaussian noise processes $\boldsymbol{\upsilon}_{\boldsymbol{\omega}}$ and $\boldsymbol{\upsilon}_{\mathbf{a}}$, such that the measured values could be modeled as

$$\begin{aligned} \tilde{\boldsymbol{\omega}} &= \boldsymbol{\omega} + \boldsymbol{\beta}_{\boldsymbol{\omega}} + \boldsymbol{\upsilon}_{\boldsymbol{\omega}} \\ \tilde{\mathbf{a}} &= \mathbf{a} + \boldsymbol{\beta}_{\mathbf{a}} + \boldsymbol{\upsilon}_{\mathbf{a}} \,. \end{aligned} \tag{71a}$$

Recognizing that the actual measured values $\tilde{\boldsymbol{\omega}}$ and $\tilde{\mathbf{a}}$ are fixed at a given timestep and that the noise terms are considered to be uncorrelated in time, the unknown true values are modeled as random variables as

$$\boldsymbol{\omega} = \tilde{\boldsymbol{\omega}} - \boldsymbol{\beta}_{\boldsymbol{\omega}} - \boldsymbol{\upsilon}_{\boldsymbol{\omega}} \tag{72a}$$

$$\mathbf{a} = \tilde{\mathbf{a}} - \boldsymbol{\beta}_{\mathbf{a}} - \boldsymbol{\upsilon}_{\mathbf{a}} \,, \tag{72b}$$

where $\boldsymbol{\omega}$, $\mathbf{a}$, $\boldsymbol{\beta}_{\boldsymbol{\omega}}$, $\boldsymbol{\beta}_{\mathbf{a}}$, $\boldsymbol{\upsilon}_{\boldsymbol{\omega}}$, and $\boldsymbol{\upsilon}_{\mathbf{a}}$ are random variables and $\tilde{\boldsymbol{\omega}}$ and $\tilde{\mathbf{a}}$ are considered as constants. The nominal value of the angular rate used for propagation is then the expected value of the true angular rate

$$\begin{aligned} \hat{\boldsymbol{\omega}} &\triangleq E[\boldsymbol{\omega}] \\ &= E[\tilde{\boldsymbol{\omega}} - \boldsymbol{\beta}_{\boldsymbol{\omega}} - \boldsymbol{\upsilon}_{\boldsymbol{\omega}}] \\ &= \tilde{\boldsymbol{\omega}} - \hat{\boldsymbol{\beta}}_{\boldsymbol{\omega}} \,, \end{aligned} \tag{73}$$

and the nominal value of the acceleration used for propagation is the expected value of the true acceleration

$$\begin{aligned} \hat{\mathbf{a}} &\triangleq E[\mathbf{a}] \\ &= E[\tilde{\mathbf{a}} - \boldsymbol{\beta}_{\mathbf{a}} - \boldsymbol{\upsilon}_{\mathbf{a}}] \\ &= \tilde{\mathbf{a}} - \hat{\boldsymbol{\beta}}_{\mathbf{a}} \,. \end{aligned} \tag{74}$$

IMU data can be treated either as a system input or as a measurement, in a trade-off based on IMU quality and model accuracy. When a reliable vehicle model is available, the controller's output (e.g. motor commands) can be used as an input to the observer to propagate the state forward. In this case, IMU data are incorporated as measurement updates providing feedback. This approach leverages the most information but requires careful characterization of the vehicle's dynamics. Another common approach, known as *mechanization*, treats the IMU measurements directly as inputs to the filter dynamics, which replaces the vehicle-specific dynamics with kinematic equations. This simplifies the propagation dynamics and eliminates sensitivity to modeling errors, but does not use any information about how the vehicle behaves.

For the filter design in this paper, the angular velocity measured by the rate gyros $\tilde{\boldsymbol{\omega}}$ is treated as an input to the propagation equations. Following Leishman et al. (2014a), the z component of the accelerometer measurement is also treated as an input, while the x and y components are used as measurement updates. The following paragraphs explain the derivation and justification for this approach.

Using Newton's second law, the velocity dynamics can be modeled as

$$\dot{\mathbf{v}} = \lfloor \mathbf{v} \rfloor \boldsymbol{\omega} + \frac{1}{m} \sum {}^{\mathcal{B}}\mathbf{F} \,, \tag{75}$$

where m is the mass of the vehicle and ${}^{\mathcal{B}}\mathbf{F}$ are the forces acting on the vehicle expressed in the body frame. As illustrated in Figure 6, the principal forces that act on a multirotor are gravity ${}^{\mathcal{B}}\mathbf{F}_G = m\mathbf{R}(\mathbf{q}_n^b)\mathbf{g}$, thrust ${}^{\mathcal{B}}\mathbf{F}_T = T\mathbf{k}$, and the simplified drag force ${}^{\mathcal{B}}\mathbf{F}_D = -\mu_0\mathbf{v}$, where $\mathbf{k} \triangleq \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^\top$, T is the total rotor thrust, μ_0 is the nominal drag coefficient, and $\mathbf{g} = g\mathbf{k}$ with g being the standard acceleration due to gravity. Substituting these forces into (75) yields

$$\begin{aligned} \dot{\mathbf{v}} &= \lfloor \mathbf{v} \rfloor \boldsymbol{\omega} + \frac{1}{m}\left({}^{\mathcal{B}}\mathbf{F}_G + {}^{\mathcal{B}}\mathbf{F}_T + {}^{\mathcal{B}}\mathbf{F}_D \right) \\ &= \lfloor \mathbf{v} \rfloor \boldsymbol{\omega} + \mathbf{R}(\mathbf{q}_n^b)\mathbf{g} + \frac{T}{m}\mathbf{k} - \frac{\mu_0}{m}\mathbf{v} \,. \end{aligned} \tag{76}$$

While technically correct, the dynamics in (76) are challenging to use in practice because the thrust T

is difficult to model and so is generally unknown. This difficulty is addressed through mechanization by utilizing the accelerometer measurements directly in the propagation model. As explained by Leishman et al. (2014a), accelerometers measure the specific force (not including gravity) expressed in the body frame, so that

$$\begin{aligned}\mathbf{a} &= \frac{1}{m}\sum {}^{\mathcal{B}}\mathbf{F} - {}^{\mathcal{B}}\mathbf{F}_G \\ &= \frac{1}{m}\left({}^{\mathcal{B}}\mathbf{F}_T + {}^{\mathcal{B}}\mathbf{F}_D\right) \\ &= \frac{T}{m}\mathbf{k} - \frac{\mu_0}{m}\mathbf{v} \; . \end{aligned} \tag{77}$$

Substituting (77) into (76) yields the common mechanization dynamics

$$\dot{\mathbf{v}} = \lfloor \mathbf{v} \rfloor \boldsymbol{\omega} + \mathbf{R}(\mathbf{q}_n^b)\mathbf{g} + \mathbf{a} \; . \tag{78}$$

Equation (78) eliminates the need for modeling complicated vehicle dynamics, but ignores information about the dynamics that might improve estimator performance by building up cross-correlation terms in the covariance matrix $\mathbf{P}$.

In practice, we have found the most success using a combination of (76) and (78). Noting that the unknown thrust term in (76) appears only in the z component, a hybrid propagation model is obtained by substituting only the z component of (77) into (76) to obtain

$$\dot{\mathbf{v}} = \lfloor \mathbf{v} \rfloor \boldsymbol{\omega} + \mathbf{R}(\mathbf{q}_n^b)\mathbf{g} + a_z\mathbf{k} - \mu\boldsymbol{\Pi}_{\mathbf{k}}\mathbf{v} \; , \tag{79}$$

where $a_z = \mathbf{k}^\mathsf{T}\mathbf{a}$, $\mu \triangleq \mu_0/m$ is the specific drag coefficient, and

$$\boldsymbol{\Pi}_{\mathbf{k}} \triangleq \mathbf{I} - \mathbf{k}\mathbf{k}^\mathsf{T}$$

is a projection matrix that projects onto the plane normal to the $\mathbf{k}$ axis.

As a result of this hybrid approach, we consider the gyroscope and z-axis accelerometer measurements as system inputs, while using the horizontal accelerometer measurements as feedback in the update step. The system input and input noise are therefore defined as

$$\mathbf{u} = \begin{bmatrix} \tilde{\boldsymbol{\omega}} \\ \tilde{a}_z \end{bmatrix} \; , \quad \boldsymbol{\upsilon} = \begin{bmatrix} \boldsymbol{\upsilon}_{\boldsymbol{\omega}} \\ \upsilon_{\mathbf{a}_z} \end{bmatrix} \; ,$$

where $\tilde{a}_z = \mathbf{k}^\mathsf{T}\tilde{\mathbf{a}}$ and $\upsilon_{\mathbf{a}_z} = \mathbf{k}^\mathsf{T}\boldsymbol{\upsilon}_{\mathbf{a}}$.

6.1.3 Propagation Model. The system dynamics are modeled as

$$\begin{aligned}\dot{\mathbf{p}}_n^b &= \mathbf{R}^\mathsf{T}(\mathbf{q}_n^b)\mathbf{v} && \text{(80a)} \\ \dot{\mathbf{q}}_n^b &= \frac{1}{2}\mathbf{q}_n^b \otimes \begin{bmatrix} \boldsymbol{\omega} \\ 0 \end{bmatrix} && \text{(80b)} \\ \dot{\mathbf{v}} &= \lfloor \mathbf{v} \rfloor \boldsymbol{\omega} + \mathbf{R}(\mathbf{q}_n^b)\mathbf{g} + a_z\mathbf{k} - \mu\boldsymbol{\Pi}_{\mathbf{k}}\mathbf{v} + \boldsymbol{\eta}_{\mathbf{v}} && \text{(80c)} \\ \dot{\boldsymbol{\beta}}_{\boldsymbol{\omega}} &= \boldsymbol{\eta}_{\boldsymbol{\beta}_{\boldsymbol{\omega}}} \\ \dot{\boldsymbol{\beta}}_{\mathbf{a}} &= \boldsymbol{\eta}_{\boldsymbol{\beta}_{\mathbf{a}}} \\ \dot{\mathbf{p}}_n^k &= \mathbf{0} \\ \dot{\mathbf{q}}_n^k &= \mathbf{0} \\ \dot{\mu} &= 0 \; , \end{aligned}$$

where $\boldsymbol{\omega}$ is given by (72a), a_z is given by the z component of (72b), and $\boldsymbol{\eta}_{\mathbf{v}}$, $\boldsymbol{\eta}_{\boldsymbol{\beta}_{\boldsymbol{\omega}}}$, and $\boldsymbol{\eta}_{\boldsymbol{\beta}_{\mathbf{a}}}$ are zero-mean Gaussian noise processes for the corresponding states. The state $\mathbf{p}_n^b$ is propagated according to a standard kinematic model, $\mathbf{q}_n^b$ is propagated according to (26) and the discussion in Section 3.4, and $\mathbf{v}$ is propagated according to (79). The dynamics for the bias states $\boldsymbol{\beta}_{\boldsymbol{\omega}}$ and $\boldsymbol{\beta}_{\mathbf{a}}$ are modeled as random walks, while the keyframe states $\mathbf{p}_n^k$ and $\mathbf{q}_n^k$ represent the relative pose of static coordinate frames and therefore do not change. The drag term μ is constant and so has zero dynamics, but uncertainty in this state is considered by assigning it a non-zero initial covariance.

The state estimate is propagated by substituting the expected values of the state, input, and process noise terms into (80) as

$$\begin{aligned}\dot{\hat{\mathbf{p}}}_n^b &= \mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_n^b)\hat{\mathbf{v}} \\ \dot{\hat{\mathbf{q}}}_n^b &= \frac{1}{2}\hat{\mathbf{q}}_n^b \otimes \begin{bmatrix} \hat{\boldsymbol{\omega}} \\ 0 \end{bmatrix} && \text{(81a)} \\ \dot{\hat{\mathbf{v}}} &= \lfloor \hat{\mathbf{v}} \rfloor \hat{\boldsymbol{\omega}} + \mathbf{R}(\hat{\mathbf{q}}_n^b)\mathbf{g} + \hat{a}_z\mathbf{k} - \hat{\mu}\boldsymbol{\Pi}_{\mathbf{k}}\hat{\mathbf{v}} \\ \dot{\hat{\boldsymbol{\beta}}}_{\boldsymbol{\omega}} &= \mathbf{0} \\ \dot{\hat{\boldsymbol{\beta}}}_{\mathbf{a}} &= \mathbf{0} \\ \dot{\hat{\mathbf{p}}}_n^k &= \mathbf{0} \\ \dot{\hat{\mathbf{q}}}_n^k &= \mathbf{0} \\ \dot{\hat{\mu}} &= 0 \; , \end{aligned}$$

where $\hat{\boldsymbol{\omega}}$ is given by (73) and $\hat{a}_z$ is given by the z component of (74).

The error-state dynamics are found by relating (80) and (81) using the error-state definition (44). The first-order approximation of the error-state dynamics are

$$\begin{aligned}\delta\dot{\mathbf{p}}_n^b &\approx -\mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_n^b)\lfloor \hat{\mathbf{v}} \rfloor \delta\boldsymbol{\theta}_n^b + \mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_n^b)\delta\mathbf{v} \\ \delta\dot{\boldsymbol{\theta}}_n^b &\approx -\left\lfloor \tilde{\boldsymbol{\omega}} - \hat{\boldsymbol{\beta}}_{\boldsymbol{\omega}} \right\rfloor \delta\boldsymbol{\theta}_n^b - \delta\boldsymbol{\beta}_{\boldsymbol{\omega}} - \boldsymbol{\upsilon}_{\boldsymbol{\omega}} \\ \delta\dot{\mathbf{v}} &\approx \left\lfloor \mathbf{R}(\hat{\mathbf{q}}_n^b)\mathbf{g} \right\rfloor \delta\boldsymbol{\theta}_n^b + \left(- \left\lfloor \tilde{\boldsymbol{\omega}} - \hat{\boldsymbol{\beta}}_{\boldsymbol{\omega}} \right\rfloor - \hat{\mu}\boldsymbol{\Pi}_{\mathbf{k}} \right)\delta\mathbf{v} \\ &\quad - \lfloor \hat{\mathbf{v}} \rfloor \delta\boldsymbol{\beta}_{\boldsymbol{\omega}} - \mathbf{k}\mathbf{k}^\mathsf{T}\delta\boldsymbol{\beta}_{\mathbf{a}} - \boldsymbol{\Pi}_{\mathbf{k}}\hat{\mathbf{v}}\delta\mu \\ &\quad - \lfloor \hat{\mathbf{v}} \rfloor \boldsymbol{\upsilon}_{\boldsymbol{\omega}} - \mathbf{k}\eta_{\mathbf{a}_z} + \boldsymbol{\eta}_{\mathbf{v}} \\ \delta\dot{\boldsymbol{\beta}}_{\boldsymbol{\omega}} &= \boldsymbol{\eta}_{\boldsymbol{\beta}_{\boldsymbol{\omega}}} \\ \delta\dot{\boldsymbol{\beta}}_{\mathbf{a}} &= \boldsymbol{\eta}_{\boldsymbol{\beta}_{\mathbf{a}}} \\ \delta\dot{\mathbf{p}}_n^k &= \mathbf{0} \\ \delta\dot{\boldsymbol{\theta}}_n^k &= \mathbf{0} \\ \delta\dot{\mu} &= 0 \end{aligned}$$

and are derived in Appendix C. Differentiating the error state dynamics with respect to the error state and input noise results in the following propagation Jacobians:

$$\mathbf{F} = \begin{bmatrix} \mathbf{0} & -\mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_n^b)\lfloor\hat{\mathbf{v}}\rfloor & \mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_n^b) & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\left\lfloor\tilde{\boldsymbol{\omega}}-\hat{\boldsymbol{\beta}}_{\boldsymbol{\omega}}\right\rfloor & \mathbf{0} & -\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \left\lfloor\mathbf{R}(\hat{\mathbf{q}}_n^b)\mathbf{g}\right\rfloor & -\left\lfloor\tilde{\boldsymbol{\omega}}-\hat{\boldsymbol{\beta}}_{\boldsymbol{\omega}}\right\rfloor-\hat{\mu}\boldsymbol{\Pi}_{\mathbf{k}} & -\lfloor\hat{\mathbf{v}}\rfloor & -\mathbf{k}\mathbf{k}^\mathsf{T} & \mathbf{0} & \mathbf{0} & -\boldsymbol{\Pi}_{\mathbf{k}}\hat{\mathbf{v}} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & 0 \end{bmatrix}$$

and

$$\mathbf{G} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ -\mathbf{I} & \mathbf{0} \\ -\lfloor \hat{\mathbf{v}} \rfloor & -\mathbf{k} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix} .$$

6.2 Measurement Models

The accelerometer, altimeter, and visual odometry or laser scan matching algorithm provide measurements to constrain state estimates. For each sensor, the measurement model, residual model, and residual Jacobians are defined.

6.2.1 Accelerometer. Because the z portion of the accelerometer measurement is used as an input to the propagation, the update step uses only the x and y components of $\tilde{\mathbf{a}}$, such that

$$\mathbf{z}_{\text{acc}} = \mathbf{I}_{2\times 3}\, \tilde{\mathbf{a}} \ .$$

The accelerometer measurement model can be derived from (71a) and (77) as

$$\begin{aligned} \mathbf{h}_{\text{acc}}(\mathbf{x}, \mathbf{u} + \boldsymbol{\upsilon}) &= \mathbf{I}_{2\times 3}\left(\mathbf{a} + \boldsymbol{\beta}_{\mathbf{a}} + \boldsymbol{\upsilon}_{\mathbf{a}}\right) \\ &= \mathbf{I}_{2\times 3}\left(\frac{T}{m}\mathbf{k} - \frac{\mu_0}{m}\mathbf{v} + \boldsymbol{\beta}_{\mathbf{a}} + \boldsymbol{\upsilon}_{\mathbf{a}}\right) \\ &= \mathbf{I}_{2\times 3}\left(-\mu\mathbf{v} + \boldsymbol{\beta}_{\mathbf{a}} + \boldsymbol{\upsilon}_{\mathbf{a}}\right) \ , \end{aligned} \tag{82}$$

where the thrust term $\frac{T}{m}\mathbf{k}$ drops out because $\mathbf{I}_{2\times 3}\frac{T}{m}\mathbf{k} = \mathbf{0}$. We can expand this model in terms of $\delta\mathbf{x}$ and $\hat{\mathbf{x}}$ according to (44) and drop second-order terms[‡] to obtain

$$\begin{aligned} \mathbf{h}_{\text{acc}}(\mathbf{x}, \mathbf{u}{+}\boldsymbol{\upsilon}) {=} \mathbf{I}_{2\times 3}\Big(&-(\hat{\mu}{+}\delta\mu)\left(\hat{\mathbf{v}}{+}\delta\mathbf{v}\right) \\ &+\hat{\boldsymbol{\beta}}_{\mathbf{a}}{+}\delta\boldsymbol{\beta}_{\mathbf{a}}{+}\boldsymbol{\upsilon}_{\mathbf{a}}\Big) \\ \approx \mathbf{I}_{2\times 3}\Big(&-\hat{\mu}\hat{\mathbf{v}}{-}\hat{\mu}\delta\mathbf{v}{-}\delta\mu\hat{\mathbf{v}} \\ &+\hat{\boldsymbol{\beta}}_{\mathbf{a}}{+}\delta\boldsymbol{\beta}_{\mathbf{a}}{+}\boldsymbol{\upsilon}_{\mathbf{a}}\Big). \end{aligned}$$

From (82), the estimated measurement is

$$\mathbf{h}_{\text{acc}}(\hat{\mathbf{x}}, \mathbf{u}) = \mathbf{I}_{2\times 3}\left(-\hat{\mu}\hat{\mathbf{v}} + \hat{\boldsymbol{\beta}}_{\mathbf{a}}\right) \ .$$

For a given acceleration measurement $\mathbf{z}_{\text{acc}}$, the residual is

$$\mathbf{r}_{\text{acc}} = \mathbf{z}_{\text{acc}} - \mathbf{h}_{\text{acc}}(\hat{\mathbf{x}}, \mathbf{u}) \ ,$$

and is modeled as

$$\begin{aligned} \mathbf{r}_{\text{acc}} &= \mathbf{h}_{\text{acc}}(\mathbf{x}, \mathbf{u} + \boldsymbol{\upsilon}) - \mathbf{h}_{\text{acc}}(\hat{\mathbf{x}}, \mathbf{u}) \\ &= \mathbf{I}_{2\times 3}\left(-\hat{\mu}\delta\mathbf{v} - \hat{\mathbf{v}}\delta\mu + \delta\boldsymbol{\beta}_{\mathbf{a}} + \boldsymbol{\upsilon}_{\mathbf{a}}\right) \ . \end{aligned}$$

The measurement Jacobian is therefore

$$\mathbf{H}_{\text{acc}} = \mathbf{I}_{2\times 3}\begin{bmatrix}\mathbf{0} & \mathbf{0} & -\hat{\mu}\mathbf{I} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & -\hat{\mathbf{v}}\end{bmatrix} \ .$$

The measurement noise is the x and y components of the accelerometer noise,

$$\mathbf{R}_{\text{acc}} = \mathbf{I}_{2\times 3} E\left[\boldsymbol{\upsilon}_{\mathbf{a}}\boldsymbol{\upsilon}_{\mathbf{a}}^{\mathsf{T}}\right]\mathbf{I}_{2\times 3}^{\mathsf{T}} \ .$$

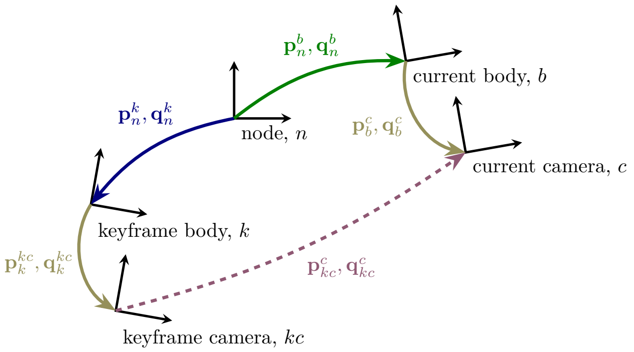

$\mathbf{p}_n^b, \mathbf{q}_n^b$
current body, b
$\mathbf{p}_n^k, \mathbf{q}_n^k$
node, n
$\mathbf{p}_b^c, \mathbf{q}_b^c$
current camera, c
keyframe body, k
$\mathbf{p}_k^{kc}, \mathbf{q}_k^{kc}$
$\mathbf{p}_{kc}^c, \mathbf{q}_{kc}^c$
keyframe camera, kc

Fig. 7. Transforms associated with the visual odometry measurement model for iRN. The transform $(\mathbf{p}_{kc}^c, \mathbf{q}_{kc}^c)$ is the output of the visual odometry algorithm. The transform $(\mathbf{p}_b^c, \mathbf{q}_b^c)$ defines the pose of the camera frame with respect to the vehicle's center of mass, and is assumed to be fixed and known. The frame k represents the pose of the body at the time of the keyframe, and the frame kc represents the pose of the camera at that time. As a result, $\mathbf{p}_k^{kc} = \mathbf{p}_b^c$ and $\mathbf{q}_k^{kc} = \mathbf{q}_b^c$.

6.2.2 Altimeter. The altimeter model is for an ultrasonic range finder, which reports the nearest return in its conical field of view. As a result, the sensor reports height above ground regardless of the current attitude of the vehicle, as long as roll and pitch angles are moderate. The measurement model and its estimate are

$$\begin{aligned} h_{\text{alt}}(\mathbf{x}) &= -\mathbf{k}^{\mathsf{T}}\mathbf{p}_n^b + \eta_{\text{alt}} \\ h_{\text{alt}}(\hat{\mathbf{x}}) &= -\mathbf{k}^{\mathsf{T}}\hat{\mathbf{p}}_n^b \ . \end{aligned} \tag{83}$$

For a given altimeter measurement z_{alt}, the residual is

$$r_{\text{alt}} = z_{\text{alt}} - h_{\text{alt}}(\hat{\mathbf{x}}) \ ,$$

which is modeled as

$$\begin{aligned} r_{\text{alt}} &= h_{\text{alt}}(\mathbf{x}) - h_{\text{alt}}(\hat{\mathbf{x}}) \\ &= -\mathbf{k}^{\mathsf{T}}\mathbf{p}_n^b + \eta_{\text{alt}} + \mathbf{k}^{\mathsf{T}}\hat{\mathbf{p}}_n^b \\ &= -\mathbf{k}^{\mathsf{T}}\delta\mathbf{p}_n^b + \eta_{\text{alt}} \end{aligned}$$

resulting in the measurement Jacobian

$$\mathbf{H}_{\text{alt}} = \begin{bmatrix}-\mathbf{k}^{\mathsf{T}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & 0\end{bmatrix} \ .$$

6.2.3 Visual Odometry Translation. Incorporating measurements from visual odometry algorithms is somewhat more involved than the previous measurement models. Figure 7 outlines the relationship between the visual odometry output and the state. The visual odometry output $(\mathbf{p}_{kc}^c, \mathbf{q}_{kc}^c)$ relates the current camera frame to the keyframe camera frame. The transform $(\mathbf{p}_b^c, \mathbf{q}_b^c)$ express the pose of the camera frame with respect to the vehicle's center of mass. Commonly the camera frame used in visual odometry algorithms is derived from the image plane such that the camera

[‡] Because the Jacobians are evaluated at $\delta\mathbf{x} = \boldsymbol{\upsilon} = 0$, any second-order terms in these variables will vanish. We take advantage of this fact to simplify the derivation by dropping these second-order terms earlier.

frame's z-axis references forward motion (depth). In this case, $\mathbf{q}_b^c$ encodes the mapping from the body frame's forward-right-down coordinate frame to the coordinate frame used by the camera. The transform $(\mathbf{p}_b^c, \mathbf{q}_b^c)$ can be found through offline calibration or online as described by Leishman and McLain (2014). In this paper we assume that the camera is rigidly mounted to the body and $(\mathbf{p}_b^c, \mathbf{q}_b^c)$ is static and known. In addition, since $(\mathbf{p}_k^{kc}, \mathbf{q}_k^{kc})$ also represents the transform between the body and camera, but at the time of the keyframe, $\mathbf{p}_k^{kc} = \mathbf{p}_b^c$ and $\mathbf{q}_k^{kc} = \mathbf{q}_b^c$.

Figure 7 can be used to informally understand how defining a relative state improves observability. The pose $(\mathbf{p}_n^k, \mathbf{q}_n^k)$ encodes the roll, pitch, and altitude of the vehicle when the keyframe is declared, all of which are observable using an altimeter and IMU. This fact, in connection with assuming $(\mathbf{p}_b^c, \mathbf{q}_b^c)$ is known, ensures that the measurements $(\mathbf{p}_{kc}^c, \mathbf{q}_{kc}^c)$ constrain the vehicle's current pose $(\mathbf{p}_n^b, \mathbf{q}_n^b)$.

Using Figure 7, the relative translation measurement is modeled as

$$\begin{aligned}
\mathbf{h}_{\text{vot}}(\mathbf{x}) &= {}^{kc}\mathbf{p}_{kc}^c + \boldsymbol{\eta}_{\text{vot}} \\
&= -\,{}^{kc}\mathbf{p}_k^{kc} - {}^{kc}\mathbf{p}_n^k + {}^{kc}\mathbf{p}_n^b + {}^{kc}\mathbf{p}_b^c + \boldsymbol{\eta}_{\text{vot}} \\
&= -\mathbf{R}(\mathbf{q}_k^{kc})\ {}^{k}\mathbf{p}_k^{kc} \\
&\quad - \mathbf{R}(\mathbf{q}_k^{kc})\mathbf{R}(\mathbf{q}_n^k)\ {}^{n}\mathbf{p}_n^k + \mathbf{R}(\mathbf{q}_k^{kc})\mathbf{R}(\mathbf{q}_n^k)\ {}^{n}\mathbf{p}_n^b \\
&\quad + \mathbf{R}(\mathbf{q}_k^{kc})\mathbf{R}(\mathbf{q}_n^k)\mathbf{R}^\top(\mathbf{q}_n^b)\ {}^{b}\mathbf{p}_b^c + \boldsymbol{\eta}_{\text{vot}}\ .
\end{aligned}$$

Dropping prescripts and recalling that $\mathbf{p}_k^{kc} = \mathbf{p}_b^c$ and $\mathbf{q}_k^{kc} = \mathbf{q}_b^c$, this becomes

$$\begin{aligned}
\mathbf{h}_{\text{vot}}(\mathbf{x}) &= -\mathbf{R}(\mathbf{q}_b^c)\mathbf{p}_b^c \\
&\quad + \mathbf{R}(\mathbf{q}_b^c)\mathbf{R}(\mathbf{q}_n^k)(\mathbf{p}_n^b - \mathbf{p}_n^k) \\
&\quad + \mathbf{R}(\mathbf{q}_b^c)\mathbf{R}(\mathbf{q}_n^k)\mathbf{R}^\top(\mathbf{q}_n^b)\mathbf{p}_b^c + \boldsymbol{\eta}_{\text{vot}}\ . \qquad (84)
\end{aligned}$$

We expand this model according to (44) and (31a), then use (33) and (34) to obtain

$$\begin{aligned}
\mathbf{h}_{\text{vot}}(\mathbf{x}) &= -\mathbf{R}(\mathbf{q}_b^c)\mathbf{p}_b^c \\
&\quad + \mathbf{R}(\mathbf{q}_b^c)\mathbf{R}(\delta\mathbf{q}_n^k)\mathbf{R}(\hat{\mathbf{q}}_n^k)(\hat{\mathbf{p}}_n^b + \delta\mathbf{p}_n^b - \hat{\mathbf{p}}_n^k - \delta\mathbf{p}_n^k) \\
&\quad + \mathbf{R}(\mathbf{q}_b^c)\mathbf{R}(\delta\mathbf{q}_n^k)\mathbf{R}(\hat{\mathbf{q}}_n^k)\mathbf{R}^\top(\hat{\mathbf{q}}_n^b)\mathbf{R}^\top(\delta\mathbf{q}_n^b)\mathbf{p}_b^c + \boldsymbol{\eta}_{\text{vot}} \\
&\approx -\mathbf{R}(\mathbf{q}_b^c)\mathbf{p}_b^c \\
&\quad + \mathbf{R}(\mathbf{q}_b^c)(\mathbf{I}-\lfloor\delta\boldsymbol{\theta}_n^k\rfloor)\mathbf{R}(\hat{\mathbf{q}}_n^k)(\hat{\mathbf{p}}_n^b+\delta\mathbf{p}_n^b-\hat{\mathbf{p}}_n^k-\delta\mathbf{p}_n^k) \\
&\quad + \mathbf{R}(\mathbf{q}_b^c)(\mathbf{I}-\lfloor\delta\boldsymbol{\theta}_n^k\rfloor)\mathbf{R}(\hat{\mathbf{q}}_n^k)\mathbf{R}^\top(\hat{\mathbf{q}}_n^b)(\mathbf{I}+\lfloor\delta\boldsymbol{\theta}_n^b\rfloor)\mathbf{p}_b^c \\
&\quad + \boldsymbol{\eta}_{\text{vot}}\ .
\end{aligned}$$

Expanding and removing second-order terms,

$$\begin{aligned}
\mathbf{h}_{\text{vot}}(\mathbf{x}) &\approx -\mathbf{R}(\mathbf{q}_b^c)\mathbf{p}_b^c \\
&\quad + \mathbf{R}(\mathbf{q}_b^c)\mathbf{R}(\hat{\mathbf{q}}_n^k)(\hat{\mathbf{p}}_n^b + \delta\mathbf{p}_n^b - \hat{\mathbf{p}}_n^k - \delta\mathbf{p}_n^k) \\
&\quad - \mathbf{R}(\mathbf{q}_b^c)\left\lfloor\delta\boldsymbol{\theta}_n^k\right\rfloor \mathbf{R}(\hat{\mathbf{q}}_n^k)(\hat{\mathbf{p}}_n^b - \hat{\mathbf{p}}_n^k) \\
&\quad + \mathbf{R}(\mathbf{q}_b^c)\mathbf{R}(\hat{\mathbf{q}}_n^k)\mathbf{R}^\top(\hat{\mathbf{q}}_n^b)\mathbf{p}_b^c \\
&\quad - \mathbf{R}(\mathbf{q}_b^c)\left\lfloor\delta\boldsymbol{\theta}_n^k\right\rfloor \mathbf{R}(\hat{\mathbf{q}}_n^k)\mathbf{R}^\top(\hat{\mathbf{q}}_n^b)\mathbf{p}_b^c \\
&\quad + \mathbf{R}(\mathbf{q}_b^c)\mathbf{R}(\hat{\mathbf{q}}_n^k)\mathbf{R}^\top(\hat{\mathbf{q}}_n^b)\left\lfloor\delta\boldsymbol{\theta}_n^b\right\rfloor \mathbf{p}_b^c + \boldsymbol{\eta}_{\text{vot}}\ .
\end{aligned}$$

The estimated measurement model is

$$\begin{aligned}
\mathbf{h}_{\text{vot}}(\hat{\mathbf{x}}) &= -\mathbf{R}(\mathbf{q}_b^c)\mathbf{p}_b^c + \mathbf{R}(\mathbf{q}_b^c)\mathbf{R}(\hat{\mathbf{q}}_n^k)(\hat{\mathbf{p}}_n^b - \hat{\mathbf{p}}_n^k) \\
&\quad + \mathbf{R}(\mathbf{q}_b^c)\mathbf{R}(\hat{\mathbf{q}}_n^k)\mathbf{R}^\top(\hat{\mathbf{q}}_n^b)\mathbf{p}_b^c\ .
\end{aligned}$$

For a given relative position measurement $\mathbf{z}_{\text{vot}}$, the residual is

$$\mathbf{r}_{\text{vot}} = \mathbf{z}_{\text{vot}} - \mathbf{h}_{\text{vot}}(\hat{\mathbf{x}})\ ,$$

which is modeled as

$$\begin{aligned}
\mathbf{r}_{\text{vot}} &= \mathbf{h}_{\text{vot}}(\mathbf{x}) - \mathbf{h}_{\text{vot}}(\hat{\mathbf{x}}) \\
&= \mathbf{R}(\mathbf{q}_b^c)\mathbf{R}(\hat{\mathbf{q}}_n^k)(\delta\mathbf{p}_n^b - \delta\mathbf{p}_n^k) \\
&\quad - \mathbf{R}(\mathbf{q}_b^c)\left\lfloor\delta\boldsymbol{\theta}_n^k\right\rfloor \mathbf{R}(\hat{\mathbf{q}}_n^k)(\hat{\mathbf{p}}_n^b - \hat{\mathbf{p}}_n^k) \\
&\quad - \mathbf{R}(\mathbf{q}_b^c)\left\lfloor\delta\boldsymbol{\theta}_n^k\right\rfloor \mathbf{R}(\hat{\mathbf{q}}_n^k)\mathbf{R}^\top(\hat{\mathbf{q}}_n^b)\mathbf{p}_b^c \\
&\quad + \mathbf{R}(\mathbf{q}_b^c)\mathbf{R}(\hat{\mathbf{q}}_n^k)\mathbf{R}^\top(\hat{\mathbf{q}}_n^b)\left\lfloor\delta\boldsymbol{\theta}_n^b\right\rfloor \mathbf{p}_b^c + \boldsymbol{\eta}_{\text{vot}} \\
&= \mathbf{R}(\mathbf{q}_b^c)\mathbf{R}(\hat{\mathbf{q}}_n^k)(\delta\mathbf{p}_n^b - \delta\mathbf{p}_n^k) \\
&\quad + \mathbf{R}(\mathbf{q}_b^c)\left\lfloor\mathbf{R}(\hat{\mathbf{q}}_n^k)(\hat{\mathbf{p}}_n^b - \hat{\mathbf{p}}_n^k)\right\rfloor \delta\boldsymbol{\theta}_n^k \\
&\quad + \mathbf{R}(\mathbf{q}_b^c)\left\lfloor\mathbf{R}(\hat{\mathbf{q}}_n^k)\mathbf{R}^\top(\hat{\mathbf{q}}_n^b)\mathbf{p}_b^c\right\rfloor \delta\boldsymbol{\theta}_n^k \\
&\quad - \mathbf{R}(\mathbf{q}_b^c)R(\hat{\mathbf{q}}_n^k)\mathbf{R}^\top(\hat{\mathbf{q}}_n^b)\lfloor\mathbf{p}_b^c\rfloor\,\delta\boldsymbol{\theta}_n^b + \boldsymbol{\eta}_{\text{vot}}\ .
\end{aligned}$$

Differentiating, we obtain the residual Jacobian

$$\mathbf{H}_{\text{vot}} = \begin{bmatrix}\mathbf{H}_1 & \mathbf{H}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{H}_3 & \mathbf{H}_4 & \mathbf{0}\end{bmatrix}\ ,$$

where

$$\begin{aligned}
\mathbf{H}_1 &= \mathbf{R}(\mathbf{q}_b^c)\mathbf{R}(\hat{\mathbf{q}}_n^k)\ , \\
\mathbf{H}_2 &= -\mathbf{R}(\mathbf{q}_b^c)\mathbf{R}(\hat{\mathbf{q}}_n^k)\mathbf{R}^\top(\hat{\mathbf{q}}_n^b)\lfloor\mathbf{p}_b^c\rfloor\ , \\
\mathbf{H}_3 &= -\mathbf{R}(\mathbf{q}_b^c)\mathbf{R}(\hat{\mathbf{q}}_n^k)\ , \\
\mathbf{H}_4 &= \mathbf{R}(\mathbf{q}_b^c)\left\lfloor\mathbf{R}(\hat{\mathbf{q}}_n^k)\left(\hat{\mathbf{p}}_n^b - \hat{\mathbf{p}}_n^k + \mathbf{R}^\top(\hat{\mathbf{q}}_n^b)\mathbf{p}_b^c\right)\right\rfloor\ .
\end{aligned}$$

6.2.4 Visual Odometry Rotation. The relative rotation measurement model also follows from Figure 7. Specifically,

$$\begin{aligned}
\mathbf{h}_{\text{vor}}(\mathbf{x}) &= \mathbf{q}_{kc}^c \\
&= (\mathbf{q}_b^c)^{-1} \otimes (\mathbf{q}_n^k)^{-1} \otimes \mathbf{q}_n^b \otimes \mathbf{q}_b^c \otimes (\boldsymbol{\eta}_{\text{vor}})^\wedge\ . \qquad (85)
\end{aligned}$$

We expand this according to (30a) as

$$\begin{aligned}
&\mathbf{h}_{\text{vor}}(\mathbf{x}) \\
&= (\mathbf{q}_b^c)^{-1} \otimes (\hat{\mathbf{q}}_n^k \otimes \delta\mathbf{q}_n^k)^{-1} \otimes \hat{\mathbf{q}}_n^b \otimes \delta\mathbf{q}_n^b \otimes \mathbf{q}_b^c \otimes (\boldsymbol{\eta}_{\text{vor}})^\wedge \\
&= (\mathbf{q}_b^c)^{-1} \otimes (\delta\mathbf{q}_n^k)^{-1} \otimes (\hat{\mathbf{q}}_n^k)^{-1} \otimes \hat{\mathbf{q}}_n^b \otimes \delta\mathbf{q}_n^b \otimes \mathbf{q}_b^c \otimes (\boldsymbol{\eta}_{\text{vor}})^\wedge\ .
\end{aligned}$$

The estimated measurement model is

$$\mathbf{h}_{\text{vor}}(\hat{\mathbf{x}}) = (\mathbf{q}_b^c)^{-1} \otimes (\hat{\mathbf{q}}_n^k)^{-1} \otimes \hat{\mathbf{q}}_n^b \otimes \mathbf{q}_b^c\ .$$

For a given relative attitude measurement $\mathbf{z}_{\text{vor}}$, from (64) the residual is modeled as

$$\mathbf{r}_{\text{vor}} = \left(\mathbf{h}_{\text{vor}}(\hat{\mathbf{x}})^{-1} \otimes \mathbf{z}_{\text{vor}}\right)^\vee\ ,$$

which is modeled and then simplified using (41) and (42) as

$$\begin{aligned}
\mathbf{r}_{\text{vor}} &= \left(\mathbf{h}_{\text{vor}}(\hat{\mathbf{x}})^{-1} \otimes \mathbf{h}_{\text{vor}}(\mathbf{x})\right)^\vee \\
&= \Big((\mathbf{q}_b^c)^{-1} \otimes (\hat{\mathbf{q}}_n^b)^{-1} \otimes \hat{\mathbf{q}}_n^k \otimes \mathbf{q}_b^c \otimes (\mathbf{q}_b^c)^{-1} \otimes (\delta\mathbf{q}_n^k)^{-1} \otimes \\
&\qquad \otimes (\hat{\mathbf{q}}_n^k)^{-1} \otimes \hat{\mathbf{q}}_n^b \otimes \delta\mathbf{q}_n^b \otimes \mathbf{q}_b^c \otimes (\boldsymbol{\eta}_{\text{vor}})^\wedge\Big)^\vee \\
&= \Big((\mathbf{q}_b^c)^{-1} \otimes \Big((\hat{\mathbf{q}}_n^b)^{-1} \otimes \Big(\hat{\mathbf{q}}_n^k \otimes (\delta\mathbf{q}_n^k)^{-1} \otimes (\hat{\mathbf{q}}_n^k)^{-1}\Big) \otimes \\
&\qquad \otimes \hat{\mathbf{q}}_n^b \otimes \delta\mathbf{q}_n^b\Big) \otimes \mathbf{q}_b^c\Big)^\vee + \boldsymbol{\eta}_{\text{vor}} \\
&= \mathbf{R}(\mathbf{q}_b^c)\left(-\mathbf{R}(\mathbf{q}_n^b)\mathbf{R}^\top(\mathbf{q}_n^k)\delta\boldsymbol{\theta}_n^k + \delta\boldsymbol{\theta}_n^b\right) + \boldsymbol{\eta}_{\text{vor}}\ .
\end{aligned}$$

The measurement Jacobian is

$$\mathbf{H}_{\text{vor}} = \left[\mathbf{0}, R(\mathbf{q}_b^c), \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, -R(\mathbf{q}_b^c)R(\mathbf{q}_n^b)R^\top(\mathbf{q}_n^k), \mathbf{0}\right] .$$

The derivation of this Jacobian is a new contribution compared to prior work (Leishman and McLain, 2014).

6.2.5 Laser Scan Matcher. The laser scan matcher measurement model is a novel contribution of this work. Laser scan matching algorithms, such as the iterative closest point (ICP) algorithm (Censi, 2008), compute the planar translation and heading change of a laser scanner between the keyframe and current scans. Because the out-of-plane motion of UAVs (roll and pitch) introduces distortions that are not modeled by the scan-matching algorithm, in practice laser scans are commonly tilt-compensated before they are passed to the scan matcher (Shen et al., 2011; Tomic et al., 2012). Tilt compensation orthorectifies the laser scan by projecting it onto a level horizontal plane, using the 2.5D-world assumption that all scanned surfaces are vertically uniform. Scan returns from the floor or ceiling are also often removed during this process. The 2.5D assumption is commonly violated in the real world, but in urban environments this approach has been demonstrated to provide acceptable performance (Shen et al., 2011).

To perform tilt-compensation on a laser scan, we first compute the quaternion $\mathbf{q}_c^o$ that rotates points expressed in the rolled and pitched laser frame c (including any roll or pitch in the body-to-sensor transform) to the orthorectified frame o as

$$\mathbf{q}_c^o = (\mathbf{q}_b^c)^{-1}(\mathbf{q}_{\theta_n^b} \otimes \mathbf{q}_{\phi_n^b})^{-1} ,$$

where $\mathbf{q}_{\theta_n^b}$ and $\mathbf{q}_{\phi_n^b}$ are computed from $\mathbf{q}_n^b$ using equations (37) and (39). Then for a laser scan $\mathcal{S} = \{(r, \theta)\}$ represented as a set of range/bearing tuples (r, θ), the tilt-compensated scan $\mathcal{S}'$ is computed as

$$\mathcal{S}' = \left\{ \mathbf{x} \in \mathbb{R}^2 \,\middle|\, \mathbf{x} = \mathbf{I}_{2\times3}\mathbf{R}(\mathbf{q}_c^o) \begin{bmatrix} r\cos\theta \\ r\sin\theta \\ 0 \end{bmatrix} \forall\, (r, \theta) \in \mathcal{S} \right\} .$$

For the orthorectification of the keyframe scan, we note that because $\mathbf{q}_n^k = \mathbf{q}_{\theta_n^b} \otimes \mathbf{q}_{\phi_n^b}$ (as shown in Section 6.3) and $\mathbf{q}_k^{kc} = \mathbf{q}_b^c$, the rotation $\mathbf{q}_c^o$ is equivalent to

$$\mathbf{q}_c^o = (\mathbf{q}_b^c)^{-1} \otimes (\mathbf{q}_n^k)^{-1} .$$

The associated rotation matrix is given by

$$\begin{aligned} \mathbf{R}(\mathbf{q}_c^o) &= \mathbf{R}((\mathbf{q}_b^c)^{-1} \otimes (\mathbf{q}_n^k)^{-1}) \\ &= \mathbf{R}^\top(\mathbf{q}_n^k)\mathbf{R}^\top(\mathbf{q}_b^c) . \end{aligned}$$

In the following discussion we derive the translation and heading portions of the measurement model independently, but they can be combined into a single update by stacking the two measurement model vectors and Jacobian matrices.

The translation measurement model is constructed by expressing the visual odometry measurement model of equation (84) in the orthorectified frame associated with the keyframe, and then extracting the x and y components:

$$\begin{aligned} \mathbf{h}_{\text{lt}}(\mathbf{x}) &= \mathbf{I}_{2\times3}\mathbf{R}(\mathbf{q}_c^o)\mathbf{h}_{\text{vot}}(\mathbf{x}) + \boldsymbol{\eta}_{\text{lt}} \\ &= \mathbf{I}_{2\times3}\mathbf{R}^\top(\mathbf{q}_n^k)\mathbf{R}^\top(\mathbf{q}_b^c)\mathbf{h}_{\text{vot}}(\mathbf{x}) + \boldsymbol{\eta}_{\text{lt}} \\ &= \mathbf{I}_{2\times3}\mathbf{R}^\top(\mathbf{q}_n^k)\mathbf{R}^\top(\mathbf{q}_b^c)\Big(-\mathbf{R}(\mathbf{q}_b^c)\mathbf{p}_b^c \\ &\qquad + \mathbf{R}(\mathbf{q}_b^c)\mathbf{R}(\mathbf{q}_n^k)(\mathbf{p}_n^b - \mathbf{p}_n^k) \\ &\qquad + \mathbf{R}(\mathbf{q}_b^c)\mathbf{R}(\mathbf{q}_n^k)\mathbf{R}^\top(\mathbf{q}_n^b)\mathbf{p}_b^c\Big) + \boldsymbol{\eta}_{\text{lt}} \\ &= \mathbf{I}_{2\times3}\Big(-\mathbf{R}^\top(\mathbf{q}_n^k)\mathbf{p}_b^c \\ &\qquad + \mathbf{p}_n^b - \mathbf{p}_n^k + \mathbf{R}^\top(\mathbf{q}_n^b)\mathbf{p}_b^c\Big) + \boldsymbol{\eta}_{\text{lt}} . \end{aligned}$$

We expand using (44), (31a), and (34) then drop second-order terms to obtain

$$\begin{aligned} \mathbf{h}_{\text{lt}}(\mathbf{x}) &= \mathbf{I}_{2\times3}\Big(-\mathbf{R}^\top(\hat{\mathbf{q}}_n^k)\mathbf{R}^\top(\delta\mathbf{q}_n^k)\mathbf{p}_b^c + \hat{\mathbf{p}}_n^b + \delta\mathbf{p}_n^b \\ &\qquad - \hat{\mathbf{p}}_n^k - \delta\mathbf{p}_n^b + \mathbf{R}^\top(\hat{\mathbf{q}}_n^b)\mathbf{R}^\top(\mathbf{q}_n^b)\mathbf{p}_b^c\Big) + \boldsymbol{\eta}_{\text{lt}} \\ &\approx \mathbf{I}_{2\times3}\Big(-\mathbf{R}^\top(\hat{\mathbf{q}}_n^k)\left(\mathbf{I} + \lfloor\delta\boldsymbol{\theta}_n^k\rfloor\right)\mathbf{p}_b^c + \hat{\mathbf{p}}_n^b + \delta\mathbf{p}_n^b \\ &\qquad - \hat{\mathbf{p}}_n^k - \delta\mathbf{p}_n^b + \mathbf{R}^\top(\hat{\mathbf{q}}_n^b)\left(I + \lfloor\delta\boldsymbol{\theta}_n^b\rfloor\right)\mathbf{p}_b^c\Big) + \boldsymbol{\eta}_{\text{lt}} \\ &= \mathbf{I}_{2\times3}\Big(-\mathbf{R}^\top(\hat{\mathbf{q}}_n^k)\mathbf{p}_b^c + \hat{\mathbf{p}}_n^b - \hat{\mathbf{p}}_n^k + \mathbf{R}^\top(\hat{\mathbf{q}}_n^b)\mathbf{p}_b^c\Big) \\ &\qquad + \mathbf{I}_{2\times3}\Big(\mathbf{R}^\top(\hat{\mathbf{q}}_n^k)\lfloor\mathbf{p}_b^c\rfloor\delta\boldsymbol{\theta}_n^k + \delta\mathbf{p}_n^b - \delta\mathbf{p}_n^k \\ &\qquad - \mathbf{R}^\top(\hat{\mathbf{q}}_n^b)\lfloor\mathbf{p}_b^c\rfloor\delta\boldsymbol{\theta}_n^b\Big) + \boldsymbol{\eta}_{\text{lt}} . \end{aligned}$$

The estimated measurement model is

$$\mathbf{h}_{\text{lt}}(\hat{\mathbf{x}}) = \mathbf{I}_{2\times3}\Big(-\mathbf{R}^\top(\hat{\mathbf{q}}_n^k)\mathbf{p}_b^c + \hat{\mathbf{p}}_n^b - \hat{\mathbf{p}}_n^k + \mathbf{R}^\top(\hat{\mathbf{q}}_n^b)\mathbf{p}_b^c\Big) ,$$

and the residual is

$$\mathbf{r}_{\text{lt}} = \mathbf{z}_{\text{lt}} - \mathbf{h}_{\text{lt}}(\hat{\mathbf{x}}) ,$$

which is modeled as

$$\begin{aligned} \mathbf{r}_{\text{lt}} &= \mathbf{h}_{\text{lt}}(\mathbf{x}) - \mathbf{h}_{\text{lt}}(\hat{\mathbf{x}}) \\ &= \mathbf{I}_{2\times3}\Big(\mathbf{R}^\top(\hat{\mathbf{q}}_n^k)\lfloor\mathbf{p}_b^c\rfloor\delta\boldsymbol{\theta}_n^k + \delta\mathbf{p}_n^b - \delta\mathbf{p}_n^k \\ &\qquad - \mathbf{R}^\top(\hat{\mathbf{q}}_n^b)\lfloor\mathbf{p}_b^c\rfloor\delta\boldsymbol{\theta}_n^b\Big) + \boldsymbol{\eta}_{\text{lt}} . \end{aligned}$$

The measurement Jacobian is

$$\mathbf{H}_{\text{lt}} = \mathbf{I}_{2\times3}\left[\mathbf{I}, -\mathbf{R}^\top(\hat{\mathbf{q}}_n^b)\lfloor\mathbf{p}_b^c\rfloor, \mathbf{0}, \mathbf{0}, \mathbf{0}, -\mathbf{I}, \mathbf{R}^\top(\hat{\mathbf{q}}_n^k)\lfloor\mathbf{p}_b^c\rfloor, \mathbf{0}\right] .$$

The derivation of the heading portion of the measurement model follows a similar strategy. Recalling from Section 3.2 that the quaternion conjugation operation can be used to express the vector portion of a quaternion in another coordinate frame, we construct the heading measurement model by first rotating the visual odometry rotation model from equation (85) into the orthorectified frame:

$$\begin{aligned} \mathbf{q}_{\text{lr}} &= (\mathbf{q}_c^o)^{-1} \otimes \mathbf{h}_{\text{vor}}(\mathbf{x}) \otimes \mathbf{q}_c^o \\ &= \mathbf{q}_n^k \otimes \mathbf{q}_b^c \otimes \mathbf{h}_{\text{vor}}(\mathbf{x}) \otimes (\mathbf{q}_b^c)^{-1} \otimes (\mathbf{q}_n^k)^{-1} \\ &= \mathbf{q}_n^k \otimes \mathbf{q}_b^c \otimes (\mathbf{q}_b^c)^{-1} \otimes (\mathbf{q}_n^k)^{-1} \\ &\qquad \otimes \mathbf{q}_n^b \otimes \mathbf{q}_b^c \otimes (\mathbf{q}_b^c)^{-1} \otimes (\mathbf{q}_n^k)^{-1} \\ &= \mathbf{q}_n^b \otimes (\mathbf{q}_n^k)^{-1} . \end{aligned}$$

The heading measurement model is then the yaw component of $\mathbf{q}_{\mathrm{lr}}$:

$$\begin{aligned} h_{\mathrm{lr}}(\mathbf{x}) &= \psi(\mathbf{q}_{\mathrm{lr}}) + \eta_{\mathrm{lr}} \\ &\triangleq \psi_{\mathrm{lr}} + \eta_{\mathrm{lr}} \ . \end{aligned}$$

The estimated measurement model is

$$\begin{aligned} h_{\mathrm{lr}}(\hat{\mathbf{x}}) &= \psi(\hat{\mathbf{q}}_{\mathrm{lr}}) \\ &\triangleq \hat{\psi}_{\mathrm{lr}} \ , \end{aligned}$$

where

$$\hat{\mathbf{q}}_{\mathrm{lr}} = \hat{\mathbf{q}}_n^b \otimes (\hat{\mathbf{q}}_n^k)^{-1} \ .$$

The residual is given by

$$r_{\mathrm{lr}} = z_{\mathrm{lr}} - h_{\mathrm{lr}}(\hat{\mathbf{x}}) \ ,$$

which is modeled as

$$\begin{aligned} r_{\mathrm{lr}} &= h_{\mathrm{lr}}(\mathbf{x}) - h_{\mathrm{lr}}(\hat{\mathbf{x}}) \\ &= \psi_{\mathrm{lr}} + \eta_{\mathrm{lr}} - \hat{\psi}_{\mathrm{lr}} \\ &= \mathbf{k}^{\mathsf{T}} \boldsymbol{\Delta}_{\mathrm{lr}} + \eta_{\mathrm{lr}} \ , \end{aligned}$$

where

$$\boldsymbol{\Delta}_{\mathrm{lr}} = \begin{bmatrix} \phi_{\mathrm{lr}} - \hat{\phi}_{\mathrm{lr}} \\ \theta_{\mathrm{lr}} - \hat{\theta}_{\mathrm{lr}} \\ \psi_{\mathrm{lr}} - \hat{\psi}_{\mathrm{lr}} \end{bmatrix}$$

as in Appendix A. We can then compute the Jacobian of the residual using the chain rule as

$$\begin{aligned} \mathbf{H}_{\mathrm{lr}} &= \frac{\partial r_{\mathrm{lr}}}{\partial \delta \mathbf{x}} \\ &= \frac{\partial r_{\mathrm{lr}}}{\partial \boldsymbol{\Delta}_{\mathrm{lr}}} \frac{\partial \boldsymbol{\Delta}_{\mathrm{lr}}}{\partial \delta \boldsymbol{\theta}_{\mathrm{lr}}} \frac{\partial \delta \boldsymbol{\theta}_{\mathrm{lr}}}{\partial \delta \mathbf{x}} \\ &= \mathbf{k}^{\mathsf{T}} \mathbf{N}_{\boldsymbol{\Delta}_{\mathrm{lr}}} \frac{\partial \delta \boldsymbol{\theta}_{\mathrm{lr}}}{\partial \delta \mathbf{x}} \ , \end{aligned} \tag{86}$$

where $\mathbf{N}_{\boldsymbol{\Delta}_{\mathrm{lr}}}$ is given by equation (100) in Appendix A. We compute $\delta\boldsymbol{\theta}_{\mathrm{lr}}$ using (30a), (29a), (40), and (41) as

$$\begin{aligned} \delta \mathbf{q}_{\mathrm{lr}} &= (\hat{\mathbf{q}}_{\mathrm{lr}})^{-1} \otimes \mathbf{q}_{\mathrm{lr}} \\ &= (\hat{\mathbf{q}}_n^b \otimes (\hat{\mathbf{q}}_n^k)^{-1})^{-1} \otimes \mathbf{q}_n^b \otimes (\mathbf{q}_n^k)^{-1} \\ &= \hat{\mathbf{q}}_n^k \otimes (\hat{\mathbf{q}}_n^b)^{-1} \\ &\qquad \otimes \hat{\mathbf{q}}_n^b \otimes \delta \mathbf{q}_n^b \otimes (\delta \mathbf{q}_n^k)^{-1} \otimes (\hat{\mathbf{q}}_n^k)^{-1} \\ &= \hat{\mathbf{q}}_n^k \otimes \delta \mathbf{q}_n^b \otimes (\delta \mathbf{q}_n^k)^{-1} \otimes (\hat{\mathbf{q}}_n^k)^{-1} \\ \implies \delta \boldsymbol{\theta}_{\mathrm{lr}} &= \mathbf{R}^{\mathsf{T}}(\hat{\mathbf{q}}_n^k) \left(\delta \boldsymbol{\theta}_n^b - \delta \boldsymbol{\theta}_n^k \right) \ , \end{aligned}$$

which we take the partial derivatives of and substitute into (86) to obtain

$$\mathbf{H}_{\mathrm{lr}} = \mathbf{k}^{\mathsf{T}} \mathbf{N}_{\boldsymbol{\Delta}_{\mathrm{lr}}} \left[\mathbf{0}, \mathbf{R}^{\mathsf{T}}(\hat{\mathbf{q}}_n^k), \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, -\mathbf{R}^{\mathsf{T}}(\hat{\mathbf{q}}_n^k), \mathbf{0} \right] \ .$$

6.3 Keyframe Reset

When a new keyframe is established, the relative part of the state is reset, as described in Section 5. As shown in Figure 5b, the new node frame is positioned at zero altitude directly below the vehicle's current true position such that the position portions of the state are reset as

$$\mathbf{p}_n^{b\,+} = \mathbf{p}_n^{k\,+} = \begin{bmatrix} 0 \\ 0 \\ p_{n\,z}^b \end{bmatrix}$$

and estimated as

$$\hat{\mathbf{p}}_n^{b\,+} = \hat{\mathbf{p}}_n^{k\,+} = \begin{bmatrix} 0 \\ 0 \\ \hat{p}_{n\,z}^b \end{bmatrix} \ . \tag{87}$$

As such, the error state reset is

$$\delta \mathbf{p}_n^{b\,+} = \delta \mathbf{p}_n^{k\,+} = \begin{bmatrix} 0 \\ 0 \\ \delta p_{n\,z}^b \end{bmatrix}$$

and

$$\frac{\partial \delta \mathbf{p}_n^{b\,+}}{\partial \delta \mathbf{p}_n^b} = \frac{\partial \delta \mathbf{p}_n^{k\,+}}{\partial \delta \mathbf{p}_n^b} = \mathbf{k}\mathbf{k}^{\mathsf{T}} \ .$$

The other vector portions of the state, including $\mathbf{v}$, $\boldsymbol{\beta}_{\boldsymbol{\omega}}$, $\boldsymbol{\beta}_{\mathbf{a}}$, and μ, do not change.

Resetting the yaw portion of the attitude states is slightly more complicated. Each new node frame is established such that the vehicle's yaw is identically zero. Setting $\psi = 0$ in (39), we see that

$$\hat{\mathbf{q}}_n^{b\,+} = \hat{\mathbf{q}}_n^{k\,+} = \mathbf{q}_{\hat{\theta}} \otimes \mathbf{q}_{\hat{\phi}} \tag{88}$$

$$= \begin{bmatrix} \cos\frac{\hat{\theta}}{2} \sin\frac{\hat{\phi}}{2} \\ \sin\frac{\hat{\theta}}{2} \cos\frac{\hat{\phi}}{2} \\ -\sin\frac{\hat{\theta}}{2} \sin\frac{\hat{\phi}}{2} \\ \cos\frac{\hat{\theta}}{2} \cos\frac{\hat{\phi}}{2} \end{bmatrix} \ , \tag{89}$$

where $\hat{\phi}$ and $\hat{\theta}$ are computed from $\hat{\mathbf{q}}_n^b$ using (37). The covariance update for the attitude error states is governed by

$$\begin{aligned} \mathbf{N}_{\boldsymbol{\theta}} &= \frac{\partial \delta \boldsymbol{\theta}_n^{b\,+}}{\partial \delta \boldsymbol{\theta}_n^b} = \frac{\partial \delta \boldsymbol{\theta}_n^{k\,+}}{\partial \delta \boldsymbol{\theta}_n^b} \\ &= \begin{bmatrix} 1 & \sin\hat{\phi} \tan\hat{\theta} & \cos\hat{\phi} \tan\hat{\theta} \\ 0 & \cos^2\hat{\phi} & -\cos\hat{\phi}\sin\hat{\phi} \\ 0 & -\cos\hat{\phi}\sin\hat{\phi} & \sin^2\hat{\phi} \end{bmatrix} \ , \end{aligned}$$

which is derived in Appendix E. When roll and pitch are approximately zero, we note that

$$\mathbf{N}_{\boldsymbol{\theta}} \approx \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \ .$$

In summary, the keyframe reset requires updating the state estimate according to (87) and (89) and updating the covariance according to (70) where

$$\mathbf{N} = \begin{bmatrix} \mathbf{k}\mathbf{k}^{\mathsf{T}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{N}_{\boldsymbol{\theta}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{k}\mathbf{k}^{\mathsf{T}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{N}_{\boldsymbol{\theta}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & 1 \end{bmatrix} \ .$$

7 Body-fixed Relative Navigation (bRN)

While defining the vehicle's pose with respect to an inertially defined origin uses more conventional dynamic equations and results in more intuitive state estimates, an increasing number of estimators use body-fixed or robocentric state definitions. In the body-fixed case, the position and attitude of an inertially-fixed origin is estimated with respect to the current vehicle's pose. Body-fixed state definitions have been shown to improve filter consistency (Castellanos et al., 2007) and facilitate local guidance and control algorithms (Yu and Beard, 2013; Owen and Montano, 2006). For example, the obstacle avoidance and visual-servoing problems requires a vehicle to make navigation decisions after estimating the pose of objects with respect to its current pose. When an inertial representation of the state is desired, the body-fixed state and its covariance can be readily transformed into an inertial frame as described in Appendix G.

The relative estimator in the relative navigation architecture can be formulated using a body-fixed state definition. In this case, body-fixed relative navigation (bRN) estimates the state of the node frame with respect to the current body. As before, when a new keyframe is declared, the horizontal position and heading states are reset. The principal difference is that body-fixed dynamics are used and attitude error is defined differently.

Unless explicitly specified, the equations and definitions in Section 6 are also assumed for bRN. This section follows the same outline as Section 6, first describing the input, state, and dynamics for the system in Section 7.1, and then defining the measurement models and keyframe reset in Sections 7.2 and 7.3 respectively. The derivations in this section represent a novel contribution not contained in prior work.

7.1 State Dynamics

Body-fixed relative navigation estimates the pose of the node frame n with respect to and expressed in the current body frame b, denoted as $({}^{b}\mathbf{p}_b^n, \mathbf{q}_b^n)$. The states ${}^{b}\mathbf{p}_b^n$ and ${}^{n}\mathbf{p}_n^b$ represent the same displacement, but are pointed in opposite directions and are expressed in different frames. Specifically,

$$ {}^{b}\mathbf{p}_b^n = -\mathbf{R}(\mathbf{q}_n^b)\, {}^{n}\mathbf{p}_n^b \; . \tag{90} $$

The states $\mathbf{q}_b^n$ and $\mathbf{q}_n^b$ are inverses of each other:

$$ \mathbf{q}_b^n = \left(\mathbf{q}_n^b\right)^{-1} \; . \tag{91} $$

The estimator also tracks the pose of frame n with respect to and expressed in frame k, denoted as $({}^{k}\mathbf{p}_k^n, \mathbf{q}_k^n)$ and shown in Figure 8. For bRN, the vehicle's state is the tuple

$$ \mathbf{x} \triangleq \left(\mathbf{p}_b^n,\ \mathbf{q}_b^n,\ \mathbf{v},\ \boldsymbol{\beta}_{\boldsymbol{\omega}},\ \boldsymbol{\beta}_{\mathbf{a}},\ \mathbf{p}_k^n,\ \mathbf{q}_k^n,\ \mu\right) \; , $$

while the input remains unchanged from Section 6.1. The state dynamics are

$$ \begin{aligned} \dot{\mathbf{p}}_b^n &= -\left\lfloor \boldsymbol{\omega} \right\rfloor \mathbf{p}_b^n - \mathbf{v} \\ \dot{\mathbf{q}}_b^n &= -\frac{1}{2}\begin{bmatrix} \boldsymbol{\omega} \\ 0 \end{bmatrix} \otimes \mathbf{q}_b^n && \text{(92a)} \\ \dot{\mathbf{v}} &= \left\lfloor \mathbf{v} \right\rfloor \boldsymbol{\omega} + \mathbf{R}^{\top}(\mathbf{q}_b^n)\mathbf{g} + a_z\mathbf{k} - \mu\boldsymbol{\Pi}_{\mathbf{k}}\mathbf{v} + \boldsymbol{\eta}_{\mathbf{v}} && \text{(92b)} \\ \dot{\boldsymbol{\beta}}_{\boldsymbol{\omega}} &= \boldsymbol{\eta}_{\boldsymbol{\beta}_{\boldsymbol{\omega}}} \\ \dot{\boldsymbol{\beta}}_{\mathbf{a}} &= \boldsymbol{\eta}_{\boldsymbol{\beta}_{\mathbf{a}}} \\ \dot{\mathbf{p}}_k^n &= \mathbf{0} \\ \dot{\mathbf{q}}_k^n &= \mathbf{0} \\ \dot{\mu} &= 0 \; , \end{aligned} $$

where derivations for the position and attitude dynamics are found in Appendix D, and where $\boldsymbol{\omega}$ is given by (72a) and a_z is given by the z component of (72b). Note that (92b) only differs from (80c) by a single transpose.

For bRN, the error-state vector $\delta\mathbf{x} \in \mathbb{R}^{22}$ is defined as

$$ \delta\mathbf{x} = \left[\, \delta\mathbf{p}_b^{n\top}\ \delta\boldsymbol{\theta}_b^{n\top}\ \delta\mathbf{v}^{\top}\ \delta\boldsymbol{\beta}_{\boldsymbol{\omega}}{}^{\top}\ \delta\boldsymbol{\beta}_{\mathbf{a}}{}^{\top}\ \delta\mathbf{p}_k^{n\top}\ \delta\boldsymbol{\theta}_k^{n\top}\ \delta\mu \,\right]^{\top} \; . $$

The first-order approximation of the error-state dynamics, also derived in Appendix D, is

$$ \begin{aligned} \delta\dot{\mathbf{p}}_b^n &\approx -\left\lfloor \tilde{\boldsymbol{\omega}} - \hat{\boldsymbol{\beta}}_{\boldsymbol{\omega}} \right\rfloor \delta\mathbf{p}_b^n - \delta\mathbf{v} - \left\lfloor \hat{\mathbf{p}}_b^n \right\rfloor \delta\boldsymbol{\beta}_{\boldsymbol{\omega}} - \left\lfloor \hat{\mathbf{p}}_b^n \right\rfloor \boldsymbol{\upsilon}_{\boldsymbol{\omega}} \\ \delta\dot{\boldsymbol{\theta}}_b^n &\approx -\left\lfloor \tilde{\boldsymbol{\omega}} - \hat{\boldsymbol{\beta}}_{\boldsymbol{\omega}} \right\rfloor \delta\boldsymbol{\theta}_b^n + \delta\boldsymbol{\beta}_{\boldsymbol{\omega}} + \boldsymbol{\upsilon}_{\boldsymbol{\omega}} \\ \delta\dot{\mathbf{v}} &\approx -\left\lfloor \mathbf{R}^{\top}(\hat{\mathbf{q}}_b^n)\mathbf{g} \right\rfloor \delta\boldsymbol{\theta}_b^n + \left(-\left\lfloor \tilde{\boldsymbol{\omega}} - \hat{\boldsymbol{\beta}}_{\boldsymbol{\omega}} \right\rfloor - \hat{\mu}\boldsymbol{\Pi}_{\mathbf{k}}\right)\delta\mathbf{v} \\ &\quad - \left\lfloor \hat{\mathbf{v}} \right\rfloor \delta\boldsymbol{\beta}_{\boldsymbol{\omega}} - \mathbf{k}\mathbf{k}^{\top}\delta\boldsymbol{\beta}_{\mathbf{a}} - \boldsymbol{\Pi}_{\mathbf{k}}\hat{\mathbf{v}}\delta\mu - \left\lfloor \hat{\mathbf{v}} \right\rfloor \boldsymbol{\upsilon}_{\boldsymbol{\omega}} \\ &\quad - \mathbf{k}\eta_{\mathbf{a}_z} + \boldsymbol{\eta}_{\mathbf{v}} \\ \delta\dot{\boldsymbol{\beta}}_{\boldsymbol{\omega}} &= \boldsymbol{\eta}_{\boldsymbol{\beta}_{\boldsymbol{\omega}}} \\ \delta\dot{\boldsymbol{\beta}}_{\mathbf{a}} &= \boldsymbol{\eta}_{\boldsymbol{\beta}_{\mathbf{a}}} \\ \delta\dot{\mathbf{p}}_k^n &= \mathbf{0} \\ \delta\dot{\boldsymbol{\theta}}_k^n &= \mathbf{0} \\ \delta\dot{\mu} &= 0 \; . \end{aligned} $$

Differentiating the error-state dynamics with respect to the error state and input noise results in the following propagation Jacobians:

$$ \mathbf{F} = \begin{bmatrix} -\lfloor\hat{\boldsymbol{\omega}}\rfloor & \mathbf{0} & -\mathbf{I} & -\lfloor\hat{\mathbf{p}}_b^n\rfloor & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\lfloor\hat{\boldsymbol{\omega}}\rfloor & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\left\lfloor \mathbf{R}^{\top}(\hat{\mathbf{q}}_b^n)\mathbf{g} \right\rfloor & -\lfloor\hat{\boldsymbol{\omega}}\rfloor - \hat{\mu}\boldsymbol{\Pi}_{\mathbf{k}} & -\lfloor\hat{\mathbf{v}}\rfloor & -\mathbf{k}\mathbf{k}^{\top} & \mathbf{0} & \mathbf{0} & -\boldsymbol{\Pi}_{\mathbf{k}}\hat{\mathbf{v}} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & 0 \end{bmatrix} $$

and

$$ \mathbf{G} = \begin{bmatrix} -\lfloor\hat{\mathbf{p}}_b^n\rfloor & \mathbf{0} \\ \mathbf{I} & \mathbf{0} \\ -\lfloor\hat{\mathbf{v}}\rfloor & -\mathbf{k} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix} \; , $$

where $\hat{\boldsymbol{\omega}}$ is defined in (73).

7.2 Measurement Models

The measurement models differ when using a body-fixed parameterization as outlined below.

7.2.1 Accelerometer. Because the accelerometer model is independent of attitude and position, the model remains unchanged for bRN. See Section 6.2.1.

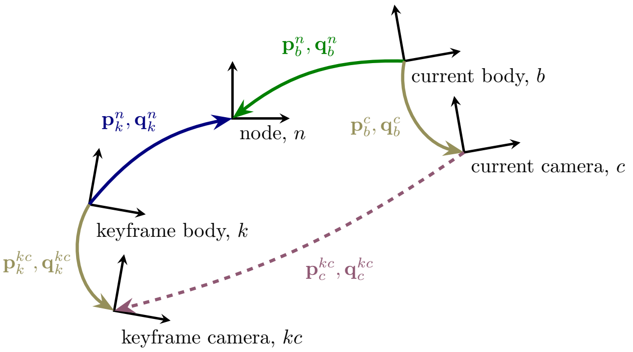

$\mathbf{p}_b^n, \mathbf{q}_b^n$
current body, b
$\mathbf{p}_k^n, \mathbf{q}_k^n$
node, n
$\mathbf{p}_b^c, \mathbf{q}_b^c$
current camera, c
keyframe body, k
$\mathbf{p}_k^{kc}, \mathbf{q}_k^{kc}$
$\mathbf{p}_c^{kc}, \mathbf{q}_c^{kc}$
keyframe camera, kc

Fig. 8. Transforms associated with the visual odometry measurement model for bRN. Note that $(\mathbf{p}_b^n, \mathbf{q}_b^n)$, $(\mathbf{p}_k^n, \mathbf{q}_k^n)$, and $(\mathbf{p}_c^{kc}, \mathbf{q}_c^{kc})$ are reversed from Figure 7. Again, $\mathbf{p}_k^{kc} = \mathbf{p}_b^c$ and $\mathbf{q}_k^{kc} = \mathbf{q}_b^c$.

7.2.2 Altimeter. As described in Section 6.2.2, the sonar altimeter measures height-above-ground. By relating (90) and (83), the vehicle's altitude is

$$h_{\text{alt}}(\mathbf{x}) = \mathbf{k}^\mathsf{T}\mathbf{R}(\mathbf{q}_b^n)\mathbf{p}_b^n + \eta_{\text{alt}} \ ,$$

where the rotation expresses the height in an inertial frame. The estimated measurement model is

$$h_{\text{alt}}(\hat{\mathbf{x}}) = \mathbf{k}^\mathsf{T}\mathbf{R}(\hat{\mathbf{q}}_b^n)\hat{\mathbf{p}}_b^n \ .$$

For a given altimeter measurement $\mathbf{z}_{\text{alt}}$, the residual is

$$r_{\text{alt}} = z_{\text{alt}} - h_{\text{alt}}(\hat{\mathbf{x}})$$

which is approximated using (31b), (44a), (33), and (4) as

$$\begin{aligned}
r_{\text{alt}} &= h_{\text{alt}}(\mathbf{x}) - h_{\text{alt}}(\hat{\mathbf{x}}) \\
&= \mathbf{k}^\mathsf{T}\mathbf{R}(\mathbf{q}_b^n)\mathbf{p}_b^n + \eta_{\text{alt}} - \mathbf{k}^\mathsf{T}\mathbf{R}(\hat{\mathbf{q}}_b^n)\hat{\mathbf{p}}_b^n \\
&= \mathbf{k}^\mathsf{T}\mathbf{R}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\delta\mathbf{q}_b^n)(\hat{\mathbf{p}}_b^n + \delta\mathbf{p}_b^n) + \eta_{\text{alt}} - \mathbf{k}^\mathsf{T}\mathbf{R}(\hat{\mathbf{q}}_b^n)\hat{\mathbf{p}}_b^n \\
&\approx \mathbf{k}^\mathsf{T}\mathbf{R}(\hat{\mathbf{q}}_b^n)\left(\mathbf{I} - \lfloor\delta\boldsymbol{\theta}_b^n\rfloor\right)(\hat{\mathbf{p}}_b^n + \delta\mathbf{p}_b^n) + \eta_{\text{alt}} - \mathbf{k}^\mathsf{T}\mathbf{R}(\hat{\mathbf{q}}_b^n)\hat{\mathbf{p}}_b^n \\
&\approx \mathbf{k}^\mathsf{T}\mathbf{R}(\hat{\mathbf{q}}_b^n)\delta\mathbf{p}_b^n - \mathbf{k}^\mathsf{T}\mathbf{R}(\hat{\mathbf{q}}_b^n)\lfloor\delta\boldsymbol{\theta}_b^n\rfloor\hat{\mathbf{p}}_b^n + \eta_{\text{alt}} \\
&= \mathbf{k}^\mathsf{T}\mathbf{R}(\hat{\mathbf{q}}_b^n)\delta\mathbf{p}_b^n + \mathbf{k}^\mathsf{T}\mathbf{R}(\hat{\mathbf{q}}_b^n)\lfloor\hat{\mathbf{p}}_b^n\rfloor\delta\boldsymbol{\theta}_b^n + \eta_{\text{alt}} \ ,
\end{aligned}$$

resulting in the measurement Jacobian

$$\mathbf{H}_{\text{alt}} = \left[\mathbf{k}^\mathsf{T}\mathbf{R}(\hat{\mathbf{q}}_b^n), \mathbf{k}^\mathsf{T}\mathbf{R}(\hat{\mathbf{q}}_b^n)\lfloor\hat{\mathbf{p}}_b^n\rfloor, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, 0\right] \ .$$

7.2.3 Visual Odometry Translation. Figure 8 presents the relative position and attitude measurement model for bRN. Note that the direction of $(\mathbf{p}_b^n, \mathbf{q}_b^n)$, $(\mathbf{p}_k^n, \mathbf{q}_k^n)$, and $(\mathbf{p}_c^{kc}, \mathbf{q}_c^{kc})$ are reversed when compared to Figure 7. A relative measurement can generally be measured either direction. Defining the measurement to point in the same direction as the state estimate avoids unnecessary coupling of the heading estimate into the translation measurement model, which in practice has shown to improve performance.

From Figure 8, the relative position update is

$$\begin{aligned}
\mathbf{h}_{\text{vot}}(\mathbf{x}) &= {}^c\mathbf{p}_c^{kc} + \boldsymbol{\eta}_{\text{vot}} \\
&= -\,{}^c\mathbf{p}_b^c + {}^c\mathbf{p}_b^n - {}^c\mathbf{p}_k^n + {}^c\mathbf{p}_k^{kc} + \boldsymbol{\eta}_{\text{vot}} \\
&= -\mathbf{R}(\mathbf{q}_b^c)\ {}^b\mathbf{p}_b^c + \mathbf{R}(\mathbf{q}_b^c)\ {}^b\mathbf{p}_b^n \\
&\quad - \mathbf{R}(\mathbf{q}_b^c)\mathbf{R}^\mathsf{T}(\mathbf{q}_b^n)\mathbf{R}(\mathbf{q}_k^n)\ {}^k\mathbf{p}_k^n \\
&\quad + \mathbf{R}(\mathbf{q}_b^c)\mathbf{R}^\mathsf{T}(\mathbf{q}_b^n)\mathbf{R}(\mathbf{q}_k^n)\ {}^k\mathbf{p}_k^{kc} + \boldsymbol{\eta}_{\text{vot}} \ .
\end{aligned}$$

Dropping prescripts and recalling that $\mathbf{p}_k^{kc} = \mathbf{p}_b^c$ and $\mathbf{q}_k^{kc} = \mathbf{q}_b^c$, this becomes

$$\begin{aligned}
\mathbf{h}_{\text{vot}}(\mathbf{x}) = \mathbf{R}(\mathbf{q}_b^c)\left(\mathbf{p}_b^n - \mathbf{p}_b^c\right)& \\
+ \mathbf{R}(\mathbf{q}_b^c)\mathbf{R}^\mathsf{T}(\mathbf{q}_b^n)\mathbf{R}(\mathbf{q}_k^n)\left(\mathbf{p}_b^c - \mathbf{p}_k^n\right) + \boldsymbol{\eta}_{\text{vot}}& \ .
\end{aligned}$$

Expanding according to (44), (31b), (33), and (34) gives

$$\begin{aligned}
&\mathbf{h}_{\text{vot}}(\mathbf{x}) \\
&{=}\mathbf{R}(\mathbf{q}_b^c)\left(\hat{\mathbf{p}}_b^n{+}\delta\mathbf{p}_b^n{-}\mathbf{p}_b^c\right) \\
&\quad{+}\mathbf{R}(\mathbf{q}_b^c)\mathbf{R}^\mathsf{T}(\delta\mathbf{q}_b^n)\mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n)\mathbf{R}(\delta\mathbf{q}_k^n)\left(\mathbf{p}_b^c{-}\hat{\mathbf{p}}_k^n{-}\delta\mathbf{p}_k^n\right) \\
&\quad{+}\boldsymbol{\eta}_{\text{vot}} \\
&{=}\mathbf{R}(\mathbf{q}_b^c)\left(\hat{\mathbf{p}}_b^n{+}\delta\mathbf{p}_b^n{-}\mathbf{p}_b^c\right) \\
&\quad{+}\mathbf{R}(\mathbf{q}_b^c)\left(\mathbf{I}{+}\lfloor\delta\boldsymbol{\theta}_b^n\rfloor\right)\mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n)\left(\mathbf{I}{-}\lfloor\delta\boldsymbol{\theta}_k^n\rfloor\right)\left(\mathbf{p}_b^c{-}\hat{\mathbf{p}}_k^n{-}\delta\mathbf{p}_k^n\right) \\
&\quad{+}\boldsymbol{\eta}_{\text{vot}} .
\end{aligned}$$

Expanding and removing higher-order terms,

$$\begin{aligned}
\mathbf{h}_{\text{vot}}(\mathbf{x}) \approx \mathbf{R}(\mathbf{q}_b^c)&\left(\hat{\mathbf{p}}_b^n + \delta\mathbf{p}_b^n - \mathbf{p}_b^c\right) \\
&+ \mathbf{R}(\mathbf{q}_b^c)\mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n)\left(\mathbf{p}_b^c - \hat{\mathbf{p}}_k^n - \delta\mathbf{p}_k^n\right) \\
&+ \mathbf{R}(\mathbf{q}_b^c)\lfloor\delta\boldsymbol{\theta}_b^n\rfloor\mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n)\left(\mathbf{p}_b^c - \hat{\mathbf{p}}_k^n\right) \\
&- \mathbf{R}(\mathbf{q}_b^c)\mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n)\lfloor\delta\boldsymbol{\theta}_k^n\rfloor\left(\mathbf{p}_b^c - \hat{\mathbf{p}}_k^n\right) + \boldsymbol{\eta}_{\text{vot}} \ .
\end{aligned}$$

The estimated measurement model is

$$\begin{aligned}
\mathbf{h}_{\text{vot}}(\hat{\mathbf{x}}) = \mathbf{R}(\mathbf{q}_b^c)\left(\hat{\mathbf{p}}_b^n - \mathbf{p}_b^c\right)& \\
+ \mathbf{R}(\mathbf{q}_b^c)\mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n)\left(\mathbf{p}_b^c - \hat{\mathbf{p}}_k^n\right)& \ .
\end{aligned}$$

We then model the residual according to (62) as

$$\begin{aligned}
\mathbf{r}_{\text{vot}} &= \mathbf{h}_{\text{vot}}(\mathbf{x}) - \mathbf{h}_{\text{vot}}(\hat{\mathbf{x}}) \\
&= \mathbf{R}(\mathbf{q}_b^c)\delta\mathbf{p}_b^n - \mathbf{R}(\mathbf{q}_b^c)\mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n)\delta\mathbf{p}_k^n \\
&\quad + \mathbf{R}(\mathbf{q}_b^c)\lfloor\delta\boldsymbol{\theta}_b^n\rfloor\mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n)\left(\mathbf{p}_b^c - \hat{\mathbf{p}}_k^n\right) \\
&\quad - \mathbf{R}(\mathbf{q}_b^c)\mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n)\lfloor\delta\boldsymbol{\theta}_k^n\rfloor\left(\mathbf{p}_b^c - \hat{\mathbf{p}}_k^n\right) + \boldsymbol{\eta}_{\text{vot}} \\
&= \mathbf{R}(\mathbf{q}_b^c)\delta\mathbf{p}_b^n - \mathbf{R}(\mathbf{q}_b^c)\left\lfloor\mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n)\left(\mathbf{p}_b^c - \hat{\mathbf{p}}_k^n\right)\right\rfloor\delta\boldsymbol{\theta}_b^n \\
&\quad - \mathbf{R}(\mathbf{q}_b^c)\mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n)\delta\mathbf{p}_k^n \\
&\quad + \mathbf{R}(\mathbf{q}_b^c)\mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n)\lfloor\mathbf{p}_b^c - \hat{\mathbf{p}}_k^n\rfloor\delta\boldsymbol{\theta}_k^n + \boldsymbol{\eta}_{\text{vot}} \ ,
\end{aligned}$$

resulting in the residual Jacobian

$$\mathbf{H}_{\text{vot}} = \begin{bmatrix}\mathbf{H}_1 & \mathbf{H}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{H}_3 & \mathbf{H}_4 & \mathbf{0}\end{bmatrix}$$

where

$$\begin{aligned}
\mathbf{H}_1 &= \mathbf{R}(\mathbf{q}_b^c) \ , \\
\mathbf{H}_2 &= -\mathbf{R}(\mathbf{q}_b^c)\left\lfloor\mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n)\left(\mathbf{p}_b^c - \hat{\mathbf{p}}_k^n\right)\right\rfloor \ , \\
\mathbf{H}_3 &= -\mathbf{R}(\mathbf{q}_b^c)\mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n) \ , \\
\mathbf{H}_4 &= \mathbf{R}(\mathbf{q}_b^c)\mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n)\lfloor\mathbf{p}_b^c - \hat{\mathbf{p}}_k^n\rfloor \ .
\end{aligned}$$

7.2.4 Visual Odometry Rotation. The relative rotation measurement model also follows from Figure 8. Specifically,

$$\begin{aligned}
\mathbf{h}_{\text{vor}}(\mathbf{x}) = \mathbf{q}_c^{kc}& \\
= (\boldsymbol{\eta}_{\text{vor}})^\vee \otimes (\mathbf{q}_b^c)^{-1} \otimes \mathbf{q}_b^n \otimes (\mathbf{q}_k^n)^{-1} \otimes \mathbf{q}_b^c& \\
= (\boldsymbol{\eta}_{\text{vor}})^\vee \otimes (\mathbf{q}_b^c)^{-1} \otimes \delta\mathbf{q}_b^n \otimes \hat{\mathbf{q}}_b^n \otimes& \\
\otimes (\hat{\mathbf{q}}_k^n)^{-1} \otimes (\delta\mathbf{q}_k^n)^{-1} \otimes \mathbf{q}_b^c& \ .
\end{aligned}$$

The estimated measurement model is

$$\mathbf{h}_{\text{vor}}(\hat{\mathbf{x}}) = (\mathbf{q}_b^c)^{-1} \otimes \hat{\mathbf{q}}_b^n \otimes (\hat{\mathbf{q}}_k^n)^{-1} \otimes \mathbf{q}_b^c \ .$$

For a given relative attitude measurement $\mathbf{z}_{\text{vor}}$, the residual is

$$\mathbf{r}_{\text{vor}} = \left(\mathbf{z}_{\text{vor}} \otimes \mathbf{h}_{\text{vor}}(\hat{\mathbf{x}})^{-1}\right)^{\vee} \ ,$$

which is modeled and then simplified using (41) and (42) as

$$\begin{aligned}
\mathbf{r}_{\text{vor}} &= \left(\mathbf{h}_{\text{vor}}(\mathbf{x}) \otimes \mathbf{h}_{\text{vor}}(\hat{\mathbf{x}})^{-1}\right)^{\vee} \\
&= \Big((\boldsymbol{\eta}_{\text{vor}})^{\vee} \otimes (\mathbf{q}_b^c)^{-1} \otimes \delta\mathbf{q}_b^n \otimes \hat{\mathbf{q}}_b^n \otimes (\hat{\mathbf{q}}_k^n)^{-1} \otimes \\
&\quad \otimes (\delta\mathbf{q}_k^n)^{-1} \otimes \mathbf{q}_b^c \otimes (\mathbf{q}_b^c)^{-1} \otimes \hat{\mathbf{q}}_k^n \otimes (\hat{\mathbf{q}}_b^n)^{-1} \otimes \mathbf{q}_b^c\Big)^{\vee} \\
&= \Big((\boldsymbol{\eta}_{\text{vor}})^{\vee} \otimes (\mathbf{q}_b^c)^{-1} \otimes \delta\mathbf{q}_b^n \otimes \big(\hat{\mathbf{q}}_b^n \otimes \\
&\quad \otimes \big((\hat{\mathbf{q}}_k^n)^{-1} \otimes (\delta\mathbf{q}_k^n)^{-1} \otimes \hat{\mathbf{q}}_k^n\big) \otimes (\hat{\mathbf{q}}_b^n)^{-1}\big) \otimes \mathbf{q}_b^c\Big)^{\vee} \\
&= \mathbf{R}(\mathbf{q}_b^c)\left(\delta\boldsymbol{\theta}_b^n - \mathbf{R}^{\mathsf{T}}(\mathbf{q}_b^n)\mathbf{R}(\mathbf{q}_k^n)\delta\boldsymbol{\theta}_k^n\right) + \boldsymbol{\eta}_{\text{vor}} \\
&= \mathbf{R}(\mathbf{q}_b^c)\delta\boldsymbol{\theta}_b^n - \mathbf{R}(\mathbf{q}_b^c)\mathbf{R}^{\mathsf{T}}(\mathbf{q}_b^n)\mathbf{R}(\mathbf{q}_k^n)\delta\boldsymbol{\theta}_k^n + \boldsymbol{\eta}_{\text{vor}} \ ,
\end{aligned}$$

resulting in the measurement Jacobian

$$\mathbf{H}_{\text{vor}} = \left[\mathbf{0}, \mathbf{R}(\mathbf{q}_b^c), \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, -\mathbf{R}(\mathbf{q}_b^c)\mathbf{R}^{\mathsf{T}}(\mathbf{q}_b^n)\mathbf{R}(\mathbf{q}_k^n), 0\right] \ .$$

7.2.5 Laser Scan Matcher. As with the iRN case, laser scans are orthorectified before being passed to the scan matcher. The quaternion $\mathbf{q}_c^o$ that describes the rotation from the laser frame c to the orthorectified frame o is given by

$$\begin{aligned}
\mathbf{q}_c^o &= (\mathbf{q}_b^c)^{-1} \otimes (\mathbf{q}_{\theta_n^b} \otimes \mathbf{q}_{\phi_n^b})^{-1} \\
&= (\mathbf{q}_b^c)^{-1} \otimes \mathbf{q}_b^o \ ,
\end{aligned}$$

where $\mathbf{q}_n^b = (\mathbf{q}_b^n)^{-1}$, $\mathbf{q}_{\theta_n^b}$ and $\mathbf{q}_{\phi_n^b}$ are computed from $\mathbf{q}_n^b$ using equations (37), and $\mathbf{q}_b^o$ is defined as

$$\mathbf{q}_b^o \triangleq (\mathbf{q}_{\theta_n^b} \otimes \mathbf{q}_{\phi_n^b})^{-1} \ .$$

The orthorectified laser scan $\mathcal{S}'$ is then computed from the original laser scan $\mathcal{S} = \{(r,\theta)\}$ as

$$\mathcal{S}' = \left\{\mathbf{x} \in \mathbb{R}^2 \,\middle|\, \mathbf{x} = \mathbf{I}_{2\times3}\mathbf{R}(\mathbf{q}_c^o)\begin{bmatrix} r\cos\theta \\ r\sin\theta \\ 0 \end{bmatrix} \forall\,(r,\theta) \in \mathcal{S}\right\} \ .$$

We note that the rotation matrix associated with $\mathbf{q}_c^o$ can be written as

$$\begin{aligned}
\mathbf{R}(\mathbf{q}_c^o) &= \mathbf{R}((\mathbf{q}_b^c)^{-1} \otimes \mathbf{q}_b^o) \\
&= \mathbf{R}(\mathbf{q}_b^o)\mathbf{R}^{\mathsf{T}}(\mathbf{q}_b^c) \ .
\end{aligned}$$

The translation portion of the measurement is modeled as the x and y component of visual odometry translation model in Section 7.2.3, after it has been rotated into the orthorectified frame:

$$\begin{aligned}
\mathbf{h}_{\text{lt}}(\mathbf{x}) &= \mathbf{I}_{2\times3}\mathbf{R}(\mathbf{q}_c^o)\mathbf{h}_{\text{vot}}(\mathbf{x}) + \boldsymbol{\eta}_{\text{lt}} \\
&= \mathbf{I}_{2\times3}\mathbf{R}(\mathbf{q}_b^o)\mathbf{R}^{\mathsf{T}}(\mathbf{q}_b^c)\Big(\mathbf{R}(\mathbf{q}_b^c)\left(\mathbf{p}_b^n - \mathbf{p}_b^c\right) \\
&\quad + \mathbf{R}(\mathbf{q}_b^c)\mathbf{R}^{\mathsf{T}}(\mathbf{q}_b^n)\mathbf{R}(\mathbf{q}_k^n)\left(\mathbf{p}_b^c - \mathbf{p}_k^n\right)\Big) + \boldsymbol{\eta}_{\text{lt}} \\
&= \mathbf{I}_{2\times3}\mathbf{R}(\mathbf{q}_b^o)\Big(\mathbf{p}_b^n - \mathbf{p}_b^c \\
&\quad + \mathbf{R}^{\mathsf{T}}(\mathbf{q}_b^n)\mathbf{R}(\mathbf{q}_k^n)\left(\mathbf{p}_b^c - \mathbf{p}_k^n\right)\Big) + \boldsymbol{\eta}_{\text{lt}} \ . \qquad (93)
\end{aligned}$$

The estimated measurement model is

$$\begin{aligned}
\mathbf{h}_{\text{lt}}(\hat{\mathbf{x}}) &= \mathbf{I}_{2\times3}\mathbf{R}(\hat{\mathbf{q}}_b^o)\Big(\hat{\mathbf{p}}_b^n - \mathbf{p}_b^c \\
&\quad + \mathbf{R}^{\mathsf{T}}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n)\left(\mathbf{p}_b^c - \hat{\mathbf{p}}_k^n\right)\Big) \ .
\end{aligned}$$

The residual Jacobian is derived in Appendix F, and is given by

$$\mathbf{H}_{\text{lt}} = \begin{bmatrix}\mathbf{H}_1 & \mathbf{H}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{H}_3 & \mathbf{H}_4 & \mathbf{0}\end{bmatrix} \ ,$$

where

$$\begin{aligned}
\mathbf{H}_1 &= \mathbf{I}_{2\times3}\mathbf{R}(\hat{\mathbf{q}}_b^o), \\
\mathbf{H}_2 &= -\mathbf{I}_{2\times3}\mathbf{R}(\hat{\mathbf{q}}_b^o)\left\lfloor\mathbf{R}^{\mathsf{T}}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n)\left(\mathbf{p}_b^c - \hat{\mathbf{p}}_k^n\right)\right\rfloor \\
&\quad + \mathbf{I}_{2\times3}\mathbf{R}(\hat{\mathbf{q}}_b^o)\left\lfloor\hat{\mathbf{p}}_b^n - \mathbf{p}_b^c + \mathbf{R}^{\mathsf{T}}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n)\left(\mathbf{p}_b^c - \hat{\mathbf{p}}_k^n\right)\right\rfloor \\
&\quad \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\hat{\phi}_n^b & 0 \\ 0 & -\sin\hat{\phi}_n^b & 0 \end{bmatrix}\mathbf{N}_{\boldsymbol{\Delta}_n^b}, \\
\mathbf{H}_3 &= -\mathbf{I}_{2\times3}\mathbf{R}(\hat{\mathbf{q}}_b^o)\mathbf{R}^{\mathsf{T}}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n), \\
\mathbf{H}_4 &= \mathbf{I}_{2\times3}\mathbf{R}(\hat{\mathbf{q}}_b^o)\mathbf{R}^{\mathsf{T}}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n)\left\lfloor\mathbf{p}_b^c - \hat{\mathbf{p}}_k^n\right\rfloor .
\end{aligned}$$

For the rotation portion of the measurement model, we first express the visual odometry rotation model in the orthorectified frame:

$$\begin{aligned}
\mathbf{q}_{\text{lr}} &= (\mathbf{q}_c^o)^{-1} \otimes \mathbf{h}_{\text{vor}}(\mathbf{x}) \otimes \mathbf{q}_c^o \\
&= ((\mathbf{q}_b^c)^{-1} \otimes \mathbf{q}_b^o)^{-1} \\
&\quad \otimes (\mathbf{q}_b^c)^{-1} \otimes \mathbf{q}_b^n \otimes (\mathbf{q}_k^n)^{-1} \otimes \mathbf{q}_b^c \otimes \left((\mathbf{q}_b^c)^{-1} \otimes \mathbf{q}_b^o\right) \\
&= (\mathbf{q}_b^o)^{-1} \otimes \mathbf{q}_b^n \otimes (\mathbf{q}_k^n)^{-1} \otimes \mathbf{q}_b^o \ ,
\end{aligned}$$

which is estimated as

$$\hat{\mathbf{q}}_{\text{lr}} = (\hat{\mathbf{q}}_b^o)^{-1} \otimes \hat{\mathbf{q}}_b^n \otimes (\hat{\mathbf{q}}_k^n)^{-1} \otimes \hat{\mathbf{q}}_b^o \ .$$

The measurement model is then the yaw portion of $\mathbf{q}_{\text{lr}}$,

$$\mathbf{h}_{\text{lr}}(\mathbf{x}) = \psi(\mathbf{q}_{\text{lr}}) + \eta_{\text{lr}} \ ,$$

which is estimated as

$$\mathbf{h}_{\text{lr}}(\hat{\mathbf{x}}) = \psi(\hat{\mathbf{q}}_{\text{lr}}) \ .$$

The residual Jacobian is derived in Appendix F, and is given by

$$\mathbf{H}_{\text{lr}} = \begin{bmatrix}\mathbf{0} & \mathbf{H}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{H}_2 & \mathbf{0}\end{bmatrix} \ ,$$

where

$$\begin{aligned}
\mathbf{H}_1 &= \mathbf{k}^{\mathsf{T}}\mathbf{N}_{\boldsymbol{\Delta}_{\text{lr}}}\mathbf{R}(\hat{\mathbf{q}}_b^o) \\
&\quad \times \left(\mathbf{I} + \left(\mathbf{R}^{\mathsf{T}}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n) - \mathbf{I}\right)\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\hat{\phi}_n^b & 0 \\ 0 & \sin\hat{\phi}_n^b & 0 \end{bmatrix}\mathbf{N}_{\boldsymbol{\Delta}_n^b}\right), \\
\mathbf{H}_2 &= -\mathbf{k}^{\mathsf{T}}\mathbf{N}_{\boldsymbol{\Delta}_{\text{lr}}}\mathbf{R}(\hat{\mathbf{q}}_b^o)\mathbf{R}^{\mathsf{T}}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n).
\end{aligned}$$

7.3 Keyframe Reset

The keyframe reset step for bRN is somewhat less intuitive due to the body-centric representation of position and orientation states, but can be derived by following the iRN reset step. Following (91) and (88),

$$\begin{aligned}\hat{\mathbf{q}}_b^{n+} &= (\hat{\mathbf{q}}_n^{b+})^{-1} \\ &= \left(\hat{\mathbf{q}}_{\hat{\theta}} \otimes \hat{\mathbf{q}}_{\hat{\phi}}\right)^{-1} \\ &= \begin{bmatrix} -\cos\frac{\hat{\theta}}{2}\sin\frac{\hat{\phi}}{2} \\ -\sin\frac{\hat{\theta}}{2}\cos\frac{\hat{\phi}}{2} \\ \sin\frac{\hat{\theta}}{2}\sin\frac{\hat{\phi}}{2} \\ \cos\frac{\hat{\theta}}{2}\cos\frac{\hat{\phi}}{2} \end{bmatrix} ,\end{aligned}$$

where ϕ and θ are computed from $\hat{\mathbf{q}}_n^b = (\hat{\mathbf{q}}_b^n)^{-1}$ using (37). Note that even though bRN expresses attitude in the body frame, the angles ϕ, θ, and ψ continue to represent conventional roll, pitch, and yaw Euler angles which express the orientation of a body with respect to an inertial frame. The Jacobian for the attitude reset is derived in Appendix E, and happens to be identical to that for the iRN attitude reset:

$$\frac{\partial \delta \boldsymbol{\theta}_b^{n+}}{\partial \delta \boldsymbol{\theta}_b^n} = \frac{\partial \delta {\boldsymbol{\theta}_n^b}^+}{\partial \delta \boldsymbol{\theta}_n^b} = \mathbf{N}_{\boldsymbol{\theta}} .$$

The derivation for the position reset is more involved, and is given in Appendix E. The resulting reset operation is

$$\hat{\mathbf{p}}_b^{n+} = \mathbf{N}_{\mathbf{p}} \hat{\mathbf{p}}_b^n ,$$

where

$$\mathbf{N}_{\mathbf{p}} = \mathbf{R}^{\mathsf{T}}(\hat{\mathbf{q}}_b^{n+})\mathbf{k}\mathbf{k}^{\mathsf{T}}\mathbf{R}(\hat{\mathbf{q}}_b^{n+}) .$$

Because frames b and k are at the same location when the reset occurs, we also have

$$\begin{aligned}\hat{\mathbf{p}}_k^{n+} &= \hat{\mathbf{p}}_b^{n+} , \\ \hat{\mathbf{q}}_k^{n+} &= \hat{\mathbf{q}}_b^{n+} .\end{aligned}$$

The total keyframe reset Jacobian, also derived in Appendix E, is given by

$$\mathbf{N} = \begin{bmatrix} \mathbf{N}_{\mathbf{p}} & \left(-\lfloor \mathbf{N}_{\mathbf{p}}\mathbf{p}_b^n \rfloor + \mathbf{N}_{\mathbf{p}} \lfloor \mathbf{p}_b^n \rfloor\right)\mathbf{N}_{\boldsymbol{\theta}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{N}_{\boldsymbol{\theta}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{N}_{\mathbf{p}} & \left(-\lfloor \mathbf{N}_{\mathbf{p}}\mathbf{p}_b^n \rfloor + \mathbf{N}_{\mathbf{p}} \lfloor \mathbf{p}_b^n \rfloor\right)\mathbf{N}_{\boldsymbol{\theta}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{N}_{\boldsymbol{\theta}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & 1 \end{bmatrix} .$$

8 Results

The RMEKF was implemented in C++ using the ROS (Quigley et al., 2009) framework. The following sections present simulation and hardware results for the performance of the RMEKF on a multirotor vehicle. The simulation results in Section 8.1 illustrate the performance of the estimator under ideal conditions, where all noise is Gaussian with known covariance and all biases are known. The hardware results in Section 8.2 demonstrate the performance of the estimator in real-world conditions, where the Gaussian noise assumption is not necessarily met and where covariance and biases are unknown.

The results in this section illustrate the typical accuracy of the relative state estimates produced by the RMEKF. Since estimating relative states is a unique approach, direct comparison to related global estimation techniques in the literature cannot readily be accomplished unless the estimator in this paper is combined with a simple global back end. For those comparisons, the reader is referred to the companion work by Wheeler et al. (2018). The more extensive flight-test results of Wheeler et al. (2017) additionally demonstrate the practical performance of the RMEKF for several extended flights of a multirotor platform, including closing control loops around the estimates.

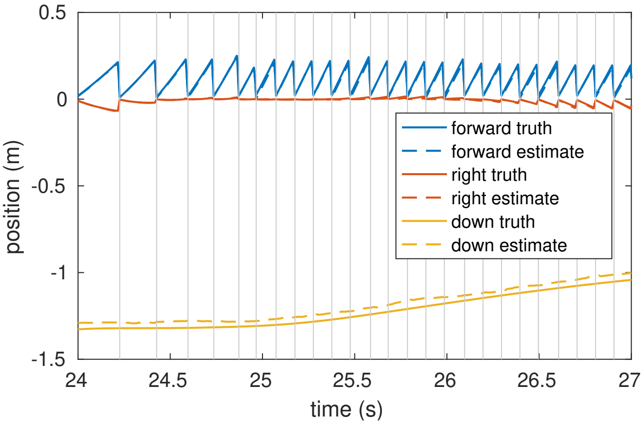
8.1 Simulation Results

The simulation provided accelerometer, gyro, altimeter, and visual odometry measurements corrupted by normally-distributed noise. Slowly-drifting biases were also added to the simulated IMU data. Sensor noise ($\boldsymbol{\upsilon}_{\boldsymbol{\omega}}$, $\boldsymbol{\upsilon}_{\mathbf{a}}$, $\boldsymbol{\eta}_{\text{vot}}$, $\boldsymbol{\eta}_{\text{vor}}$, and η_{alt}) was sampled from normal distributions with the following standard deviations that are typical of low-cost hardware sensors: $\sigma_{\boldsymbol{\omega}} = 0.13\,\text{rad/s}$, $\sigma_{\mathbf{a}} = 1.15\,\text{m/s}^2$, $\sigma_{\text{vot}} = 0.02\,\text{m}$, $\sigma_{\text{vor}} = 0.01\,\text{rad}$, and $\sigma_{\text{alt}} = 0.01\,\text{m}$. New keyframes were established when the vehicle moved more than 0.2 m or yawed more than 20 degrees. The estimator was evaluated during various maneuvers ranging from mild to aggressive, where during the aggressive maneuvers the vehicle's speed exceeded 25 m/s and the bank angle exceeded 45 degrees.

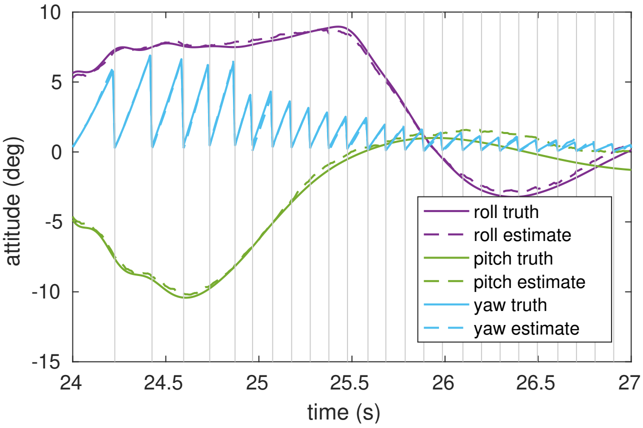
Figures 9 and 10 show a three-second snapshot of the performance of the iRN and bRN estimator at tracking the vehicle's pose, where the small time window was selected to make the relative state reset visible. During these three seconds, the vehicle was moving forward at nearly-constant velocity while maintaining a nominal height above ground of 1.25 m, and while gradually slowing its clockwise yaw motion. The vertical gray lines indicate the time when a new keyframe is declared. While the state is defined using quaternions, Figure 9b uses (37) to plot roll, pitch, and yaw angles.

Figure 9 highlights several interesting practicalities of relative navigation. As discussed in Section 5, Figure 9 illustrates how the forward, right, and yaw states are reset to zero with each newly-declared keyframe, while the altitude, roll, and pitch states remain continuous. It should be noted that while roll and pitch are continuous, discontinuities appear in each of the four quaternion states. Because the discontinuities in the relative states occur at known times, they are easily accounted for and so in practice do not cause problems with control stability. Figure 9 also illustrates that keyframes do not reset at fixed intervals, but rather reset based on how far the vehicle has traveled since the previous keyframe. Certain sensors, such as a laser scanner with a long range and wide field of view, facilitate longer distances between keyframes.

The RMEKF performed very similarly when using the body-fixed dynamics presented in Section 7. Figure 10 shows the performance of the bRN estimator for position states over the same window of time as Figure 9. The



(a) Position



(b) Attitude

Fig. 9. Simulation RMEKF estimation performance for inertial relative navigation (iRN). The vertical gray lines indicate when a new keyframe is declared.

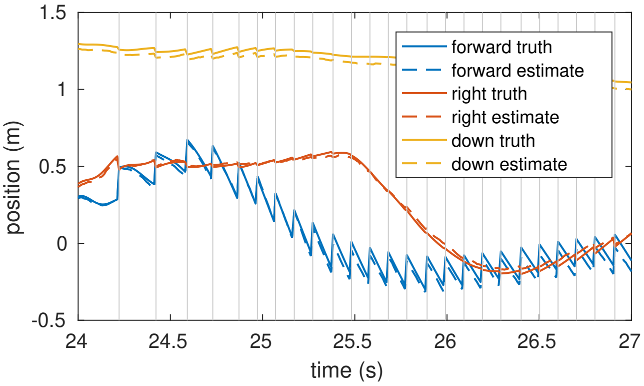


Fig. 10. Simulation RMEKF position estimation performance for body-fixed relative navigation (bRN). The bRN attitude performance is nearly identical to the iRN attitude performance but with the signs negated.

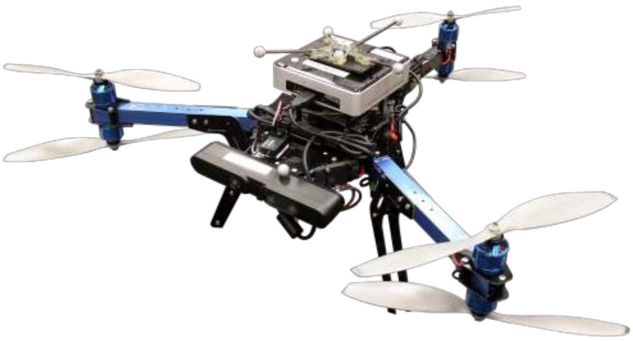


Fig. 11. The vehicle used for the hardware result flight tests

attitude performance of bRN (not shown) is nearly identical to the attitude performance of iRN with the exception of a change in sign for the angles. Note in Figure 10 that each of the position states experiences slight discontinuities at the keyframe resets and that the position estimates do not reset to zero. This is because the keyframe reset step removes all horizontal translation from the state but continues to track the vehicle's height above ground. Since the bRN position state is expressed with respect to the rolled and pitched body frame, some of the height above ground is mapped into the forward and right components. When the bRN state estimates are expressed with respect to the current node frame using (90) and (91), they are nearly identical to the estimates shown in Figure 9.

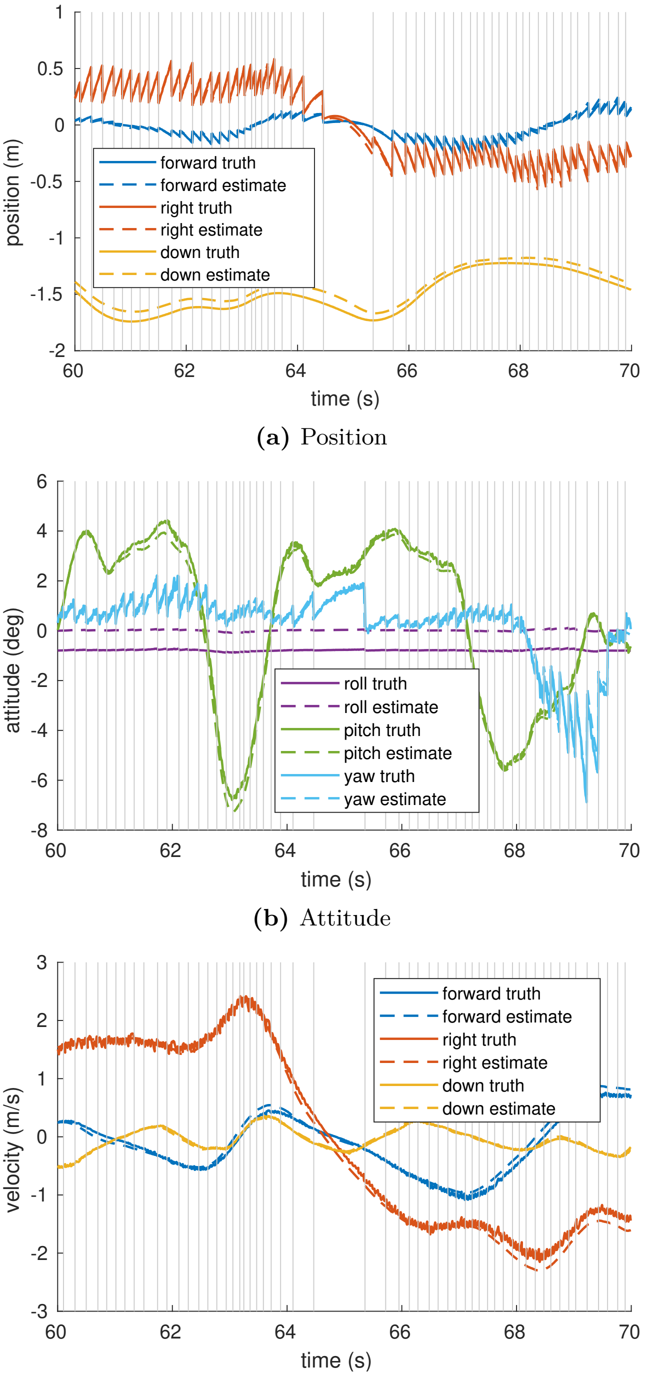
8.2 Hardware Results

The vehicle used for hardware results, a hexacopter in a Y6 configuration, is shown in Figure 11. Attitude control is performed by a Naze32 autopilot running the ROSflight[§] firmware. IMU data is streamed from the low-cost MEMS IMU on the autopilot (an InvenSense MPU-6050) at 500 Hz. Altimeter data is obtained from a MaxBotix LV-MaxSonar-EZ3 ultrasonic range finder.

All estimation is performed in real-time on the onboard computer with an Intel i7 processor. Visual odometry is also computed on the onboard computer at 15 Hz from a forward-facing Asus Xtion Pro Live RGB-D camera, using the depth-enhanced monocular odometry algorithm (Zhang et al., 2014) modified to output relative measurements. New keyframes were established when the vehicle moved more than 0.5 m or yawed more than 20 degrees, although more sophisticated keyframe-selection algorithms could be implemented. A constant, typical value was chosen for the visual odometry measurement covariance, but future implementations could take advantage of algorithms that estimate the covariance from frame to frame (e.g. Anderson et al. (2019)).

For the comparisons in this paper, truth reference was provided by a motion capture system. The positioning accuracy of the motion capture system is on the order of 0.5 cm or better, and the attitude accuracy is on the order of 0.1 degrees. A calibration routine was used to register the quadrotor body frame to the motion-capture frame. To compare the relative state estimates with the global motion-capture data, the reference frames were synchronized by applying the same position and orientation reset steps described in Sections 6.3 and 7.3 to the motion-capture data each time a visual-odometry keyframe reset occured. While there are small timing delays and other error sources inherent in this

[§] `http://rosflight.org`



(a) Position

(b) Attitude

(c) Velocity

Fig. 12. Hardware RMEKF estimation performance for inertial relative navigation (iRN). The vertical gray lines indicate when a new keyframe is declared.

Table 1. RMS error for the hardware results

State	Axis	RMS error	
		iRN	bRN
Position	forward	0.0306 m	0.0318 m
	right	0.0439 m	0.0436 m
	down	0.0648 m	0.0414 m
Attitude	roll	0.7983 deg	1.6259 deg
	pitch	0.3785 deg	0.5714 deg
	yaw	0.2933 deg	0.3422 deg
Velocity	forward	0.1313 m/s	0.1481 m/s
	right	0.1483 m/s	0.1784 m/s
	down	0.0702 m/s	0.0806 m/s

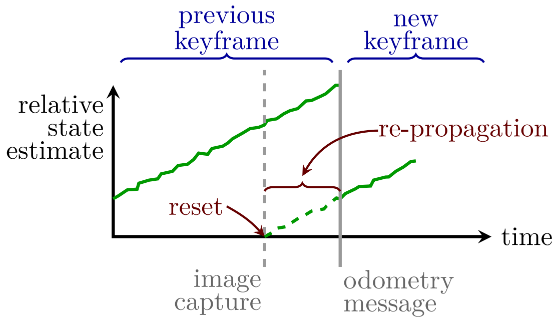


Fig. 13. Keyframe reset with delayed visual odometry measurements. The relative state may appear to not reset all the way to zero as a result of the handling of the delayed measurements as described in Section 4.3. Only the solid portions of the estimate line are published.

approach, we believe this “truth” reference to be highly representative of the true motion and an appropriate reference source for system evaluation.

Figure 12 shows the estimator performance for the RMEKF using inertial dynamics (iRN). A typical 10-second window of the flight is shown to make the relative state reset visible. The estimates track the truth well. Again, note that the forward and right position estimates in Figure 12a reset with each new keyframe, while the down position estimate is continuous. Similarly, the yaw estimate in Figure 12b resets with each keyframe, but roll and pitch are continuous. The velocity estimates in Figure 12c are continuous. The RMS errors in each of the state estimates over the duration of the 90-second flight are summarized in Table 1.

The performance of the estimator using body-fixed dynamics (bRN) was very similar to the inertial dynamics. Plots are not shown for the bRN results, but the RMS errors for a comparable flight test are summarized in Table 1.

Looking more closely at Figures 12a and 12b, it appears as if the relative states do not reset all the way to zero. The reason for this is the delay in the visual odometry processing, and is illustrated in Figure 13. On average, it takes approximately 115 ms from the time an image is captured (the dashed vertical line in Figure 13) to the time the visual odometry message is published (the solid vertical line). However, the estimator uses the timestamp for when the image was captured when applying the odometry measurement, as described in Section 4.3. When a new keyframe image is captured, during the time window between the image capture and odometry message the estimator continues operating with respect to the previous keyframe. Once the new keyframe odometry message arrives, the estimator rewinds to the time of the image capture, resets the state to zero, then re-propagates the IMU and altimeter messages that were queued up to the visual odometry message time. As a result, the first message that the estimator publishes with respect to the new keyframe includes the 115 ms of IMU re-propagation and is non-zero. However, the relative state was actually reset to zero at the time of the image capture.

9 Conclusions

While the global state of a UAS is not observable when navigating through GPS-denied environments, it is possible to maintain observability by subtly restructuring the problem using the relative navigation framework. Previous simulation studies show strong theoretical motivations for the relative navigation framework, including improved consistency, bounded covariance, and improved accuracy (Wheeler et al., 2018). In addition, significant hardware results have demonstrated the efficacy and practicality of relative navigation for small UAS when using a relative MEKF for state estimation (Wheeler et al., 2017). This paper rigorously derives the RMEKF used in (Wheeler et al., 2018) and (Wheeler et al., 2017), presenting the mathematics necessary to apply relative navigation to the UAS state estimation problem. Its tutorial nature and step-by-step derivations make this paper a self-contained resource for extending the approach to other applications.

By defining the state with respect to a local coordinate frame, this paper demonstrates how to leverage relative measurements from a visual odometry algorithm to ensure an observable state. New visual odometry and laser scan-matching measurement models are proposed, and a unique keyframe reset step is presented to ensure filter states are fully observable even when global information is not available.

The RMEKF is demonstrated in simulation and hardware to work effectively for both inertially-defined and body-fixed vehicle dynamics to produce accurate state estimates with bounded uncertainty. While the inertial and body-fixed definitions yield similar results, this paper explicitly outlines the differences that arise in the state estimator, including differences in the error state definition, measurement models, quaternion integration procedure, and keyframe reset step. Additional contributions of this paper include a tutorial introduction to indirect multiplicative extended Kalman filtering, an exposition of Hamilton quaternions, and the derivation of several novel properties of error quaternions necessary for partially updating a quaternion and its covariance.

Acknowledgements

The authors would like to thank Prashant Ganesh for his assistance with obtaining flight-test results and hardware validation of the estimation approach in this paper. Prashant is a research engineer in the Autonomous Vehicle Lab at the University of Florida's Research & Engineering Education Facility.

Funding

This work has been funded by the Center for Unmanned Aircraft Systems (C-UAS), a National Science Foundation Industry/University Cooperative Research Center (I/UCRC) under NSF award Numbers IIP-1161036 and CNS-1650547 along with significant contributions from C-UAS industry members. This work was also supported in part by Air Force Research Laboratory Science and Technology (AFRL S&T) sponsorship. This research was conducted with Government support under and awarded by DoD, Air Force Office of Scientific Research, National Defense Science and Engineering Graduate (NDSEG) Fellowship, 32 CFR 168a.

Appendix A Euler Decomposition Jacobians

In this section we explore the relationship between errors in the Euler-angle decomposition of a quaternion ($\psi - \hat{\psi}$, $\theta - \hat{\theta}$, and $\phi - \hat{\phi}$) and the minimal representation for the error in that quaternion, $\boldsymbol{\delta\theta}$. Specifically, defining

$$\boldsymbol{\Delta} \triangleq \begin{bmatrix} \phi - \hat{\phi} \\ \theta - \hat{\theta} \\ \psi - \hat{\psi} \end{bmatrix} ,$$

we wish to derive an expression for the Jacobian

$$\mathbf{N}_{\boldsymbol{\Delta}} \triangleq \frac{\partial \boldsymbol{\Delta}}{\partial \boldsymbol{\delta\theta}} .$$

The derivation and expressions that follow show some similarity to those presented by Barfoot et al. (2011), but are derived in the context of an error-state formulation.

We begin by deriving expressions for the error quaternions for the yaw, pitch, and roll quaternions of equations (38) and (39). For example, with the iRN definition of the error quaternion, we can compute $\boldsymbol{\delta}\mathbf{q}_\psi$ from (30a) and (39) as

$$\begin{aligned} \boldsymbol{\delta}\mathbf{q}_\psi &= \hat{\mathbf{q}}_\psi{}^{-1} \otimes \mathbf{q}_\psi \\ &= \begin{bmatrix} 0 \\ 0 \\ -\sin(\frac{1}{2}\hat{\psi}) \\ \cos(\frac{1}{2}\hat{\psi}) \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 0 \\ \sin(\frac{1}{2}\psi) \\ \cos(\frac{1}{2}\psi) \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ 0 \\ \sin(\frac{1}{2}\psi)\cos(\frac{1}{2}\hat{\psi}) - \cos(\frac{1}{2}\psi)\sin(\frac{1}{2}\hat{\psi}) \\ \cos(\frac{1}{2}\psi)\cos(\frac{1}{2}\hat{\psi}) + \sin(\frac{1}{2}\psi)\sin(\frac{1}{2}\hat{\psi}) \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ 0 \\ \sin(\frac{1}{2}\left(\psi - \hat{\psi}\right)) \\ \cos(\frac{1}{2}\left(\psi - \hat{\psi}\right)) \end{bmatrix} , \end{aligned} \qquad (94)$$

where the final step uses the standard angle-difference trigonometric identities. Following a similar analysis, we see that (94) also holds for bRN using (30b). We likewise see that

$$\delta\mathbf{q}_\phi = \begin{bmatrix} \sin(\frac{1}{2}(\phi-\hat{\phi})) \\ 0 \\ 0 \\ \cos(\frac{1}{2}(\phi-\hat{\phi})) \end{bmatrix}, \quad \delta\mathbf{q}_\theta = \begin{bmatrix} 0 \\ \sin(\frac{1}{2}(\theta-\hat{\theta})) \\ 0 \\ \cos(\frac{1}{2}(\theta-\hat{\theta})) \end{bmatrix} \tag{95}$$

for both the iRN and bRN attitude error definitions. Because these errors are all expressed in different intermediate frames of the Euler rotation sequence, $\delta\mathbf{q} \neq \delta\mathbf{q}_\psi \otimes \delta\mathbf{q}_\theta \otimes \delta\mathbf{q}_\phi$ in general.

Assuming small attitude errors, we approximate (94) and (95) with the first-order Taylor series as

$$\delta\mathbf{q}_\phi \approx \begin{bmatrix} \frac{\phi-\hat{\phi}}{2} \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad \delta\mathbf{q}_\theta \approx \begin{bmatrix} 0 \\ \frac{\theta-\hat{\theta}}{2} \\ 0 \\ 1 \end{bmatrix}, \quad \delta\mathbf{q}_\psi \approx \begin{bmatrix} 0 \\ 0 \\ \frac{\psi-\hat{\psi}}{2} \\ 1 \end{bmatrix}.$$

These error states are represented minimally according to (19) as

$$\delta\boldsymbol{\theta}_\phi = \begin{bmatrix} \phi-\hat{\phi} \\ 0 \\ 0 \end{bmatrix}, \quad \delta\boldsymbol{\theta}_\theta = \begin{bmatrix} 0 \\ \theta-\hat{\theta} \\ 0 \end{bmatrix}, \quad \delta\boldsymbol{\theta}_\psi = \begin{bmatrix} 0 \\ 0 \\ \psi-\hat{\psi} \end{bmatrix}. \tag{96}$$

We then define

$$\begin{aligned} \boldsymbol{\Delta} &\triangleq \delta\boldsymbol{\theta}_\phi + \delta\boldsymbol{\theta}_\theta + \delta\boldsymbol{\theta}_\psi \\ &= \begin{bmatrix} \phi-\hat{\phi} \\ \theta-\hat{\theta} \\ \psi-\hat{\psi} \end{bmatrix}. \end{aligned}$$

Again, it should be noted that $\delta\boldsymbol{\theta} \neq \boldsymbol{\Delta}$ in general.

We now wish to solve for $\delta\boldsymbol{\theta}$ in terms $\delta\boldsymbol{\theta}_\psi$, $\delta\boldsymbol{\theta}_\theta$, and $\delta\boldsymbol{\theta}_\phi$. For the iRN case, starting with (30a) and using (38), we see that

$$\begin{aligned} \delta\mathbf{q} &= \hat{\mathbf{q}}^{-1} \otimes \mathbf{q} \\ &= \hat{\mathbf{q}}_\phi^{-1} \otimes \hat{\mathbf{q}}_\theta^{-1} \otimes \hat{\mathbf{q}}_\psi^{-1} \otimes \\ &\qquad \otimes \hat{\mathbf{q}}_\psi \otimes \delta\mathbf{q}_\psi \otimes \hat{\mathbf{q}}_\theta \otimes \delta\mathbf{q}_\theta \otimes \hat{\mathbf{q}}_\phi \otimes \delta\mathbf{q}_\phi \\ &= \left(\hat{\mathbf{q}}_\phi^{-1} \otimes \left(\hat{\mathbf{q}}_\theta^{-1} \otimes \delta\mathbf{q}_\psi \otimes \hat{\mathbf{q}}_\theta\right) \otimes \delta\mathbf{q}_\theta \otimes \hat{\mathbf{q}}_\phi\right) \otimes \delta\mathbf{q}_\phi, \end{aligned}$$

which from (42) implies

$$\delta\boldsymbol{\theta} = \mathbf{R}_{\hat{\phi}}\mathbf{R}_{\hat{\theta}}\delta\boldsymbol{\theta}_\psi + \mathbf{R}_{\hat{\phi}}\delta\boldsymbol{\theta}_\theta + \delta\boldsymbol{\theta}_\phi. \tag{97}$$

We can expand (97) using (36) and (96) to express $\delta\boldsymbol{\theta}$ in terms of $\boldsymbol{\Delta}$ as

$$\begin{aligned} \delta\boldsymbol{\theta} &= \begin{bmatrix} (\phi-\hat{\phi}) - (\psi-\hat{\psi})\sin\hat{\theta} \\ (\theta-\hat{\theta})\cos\hat{\phi} + (\psi-\hat{\psi})\sin\hat{\phi}\cos\hat{\theta} \\ -(\theta-\hat{\theta})\sin\hat{\phi} + (\psi-\hat{\psi})\cos\hat{\phi}\cos\hat{\theta} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & -\sin\hat{\theta} \\ 0 & \cos\hat{\phi} & \sin\hat{\phi}\cos\hat{\theta} \\ 0 & -\sin\hat{\phi} & \cos\hat{\phi}\cos\hat{\theta} \end{bmatrix} \boldsymbol{\Delta}. \end{aligned} \tag{98}$$

The determinant of the matrix in (98) equals $\cos\hat{\theta}$. Therefore, assuming that $\hat{\theta} \neq \pm\frac{\pi}{2}$, we solve (98) for $\boldsymbol{\Delta}$ as

$$\boldsymbol{\Delta} = \begin{bmatrix} 1 & \sin\hat{\phi}\tan\hat{\theta} & \cos\hat{\phi}\tan\hat{\theta} \\ 0 & \cos\hat{\phi} & -\sin\hat{\phi} \\ 0 & \sin\hat{\phi}\sec\hat{\theta} & \cos\hat{\phi}\sec\hat{\theta} \end{bmatrix} \delta\boldsymbol{\theta}, \tag{99}$$

from which we obtain

$$\mathbf{N}_{\boldsymbol{\Delta}} \triangleq \frac{\partial\boldsymbol{\Delta}}{\partial\delta\boldsymbol{\theta}} = \begin{bmatrix} 1 & \sin\hat{\phi}\tan\hat{\theta} & \cos\hat{\phi}\tan\hat{\theta} \\ 0 & \cos\hat{\phi} & -\sin\hat{\phi} \\ 0 & \sin\hat{\phi}\sec\hat{\theta} & \cos\hat{\phi}\sec\hat{\theta} \end{bmatrix}. \tag{100}$$

A similar procedure could be used to derive the Jacobian $\mathbf{N}_{\boldsymbol{\Delta}}$ for the bRN case. However, the definition and interpretation of roll, pitch, and yaw are less obvious for the bRN case, so we will omit the derivation of $\mathbf{N}_{\boldsymbol{\Delta}}$ here and instead derive the relevant expressions in the text as they arise.

Appendix B Zero-Mean Error State

In this appendix we show that the expected error state remains zero mean. Specifically, we show that the expected value of the error state is zero, as indicated in (45), and that the error-state dynamics are trivial when the estimation error and input noise are both zero.

B.1 Expected Value of Error State

Property (45) stems from the linearity of the expectation and quaternion multiplication, inverse, and $^\vee$ operations. For the following derivations, recall that $\mathbf{x}$ is modeled as a random variable, while $\hat{\mathbf{x}}$ is a constant value for a given timestep. Given (44a) and $\hat{\mathbf{x}} = E[\mathbf{x}]$, for vector portions of the state we have

$$\begin{aligned} E[\delta\mathbf{x}_\mathbf{v}] &= E[\mathbf{x}_\mathbf{v} - \hat{\mathbf{x}}_\mathbf{v}] \\ &= E[\mathbf{x}_\mathbf{v}] - \hat{\mathbf{x}}_\mathbf{v} \\ &= \mathbf{0} \end{aligned}$$

Using (30a) as the attitude error definition for (44b), for quaternion portions of the state we have

$$\begin{aligned} E[\delta\mathbf{x}_{\boldsymbol{\theta}}] &= E\left[\left(\hat{\mathbf{x}}_\mathbf{q} \otimes \mathbf{x}_\mathbf{q}^{-1}\right)^\vee\right] \\ &= \left(\hat{\mathbf{x}}_\mathbf{q} \otimes E[\mathbf{x}_\mathbf{q}]^{-1}\right)^\vee \\ &= \left(\hat{\mathbf{x}}_\mathbf{q} \otimes \hat{\mathbf{x}}_\mathbf{q}^{-1}\right)^\vee \\ &= \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix}^\vee \\ &= \mathbf{0}. \end{aligned}$$

A similar derivation is also possible for the attitude error definition in (30b).

B.2 Error-State Dynamics

For vector portions of the state, from (44a), (47), and (50) we have

$$\begin{aligned} \delta\dot{\mathbf{x}}_\mathbf{v} &= \dot{\mathbf{x}}_\mathbf{v} - \dot{\hat{\mathbf{x}}}_\mathbf{v} \\ &= \mathbf{f}_\mathbf{v}(\mathbf{x}, \mathbf{u}+\boldsymbol{\upsilon}) + \boldsymbol{\eta}_\mathbf{v} - \mathbf{f}_\mathbf{v}(\hat{\mathbf{x}}, \mathbf{u}) \\ &\triangleq \bar{\mathbf{f}}_\mathbf{v}(\delta\mathbf{x}, \boldsymbol{\upsilon}, \hat{\mathbf{x}}, \mathbf{u}) + \boldsymbol{\eta}_\mathbf{v}. \end{aligned}$$

We note that $\mathbf{x}$ can be rewritten in terms of $\delta\mathbf{x}$ and $\hat{\mathbf{x}}$ using (29) and by rearranging (44a), and that $\mathbf{x} = \hat{\mathbf{x}}$ when $\delta\mathbf{x} = \mathbf{0}$. Evaluating $\bar{\mathbf{f}}_{\mathbf{v}}$ at $(E[\delta\mathbf{x}], E[\boldsymbol{\upsilon}], \hat{\mathbf{x}}, \mathbf{u})$, where $E[\delta\mathbf{x}] = E[\boldsymbol{\upsilon}] = \mathbf{0}$, then gives

$$\begin{aligned}\bar{\mathbf{f}}_{\mathbf{v}}(E[\delta\mathbf{x}], E[\boldsymbol{\upsilon}], \hat{\mathbf{x}}, \mathbf{u}) &= \mathbf{f}_{\mathbf{v}}(\hat{\mathbf{x}}, \mathbf{u}) - \mathbf{f}_{\mathbf{v}}(\hat{\mathbf{x}}, \mathbf{u}) \\ &= \mathbf{0}\ .\end{aligned}$$

For quaternion portions of the state, for iRN we start with (29a):

$$\begin{aligned}\mathbf{x}_{\mathbf{q}} &= \hat{\mathbf{x}}_{\mathbf{q}} \otimes (\delta\mathbf{x}_{\boldsymbol{\theta}})^{\wedge} \\ \implies \dot{\mathbf{x}}_{\mathbf{q}} &= \dot{\hat{\mathbf{x}}}_{\mathbf{q}} \otimes (\delta\mathbf{x}_{\boldsymbol{\theta}})^{\wedge} + \hat{\mathbf{x}}_{\mathbf{q}} \otimes (\delta\dot{\mathbf{x}}_{\boldsymbol{\theta}})^{\wedge}\ .\end{aligned}$$

Multiplying on the left by $(\hat{\mathbf{x}}_{\mathbf{q}})^{-1}$ gives

$$\begin{aligned}(\hat{\mathbf{x}}_{\mathbf{q}})^{-1} \otimes \dot{\mathbf{x}}_{\mathbf{q}} &= (\hat{\mathbf{x}}_{\mathbf{q}})^{-1} \otimes \dot{\hat{\mathbf{x}}}_{\mathbf{q}} \otimes (\delta\mathbf{x}_{\boldsymbol{\theta}})^{\wedge} + (\delta\dot{\mathbf{x}}_{\boldsymbol{\theta}})^{\wedge} \\ \implies \delta\dot{\mathbf{x}}_{\boldsymbol{\theta}} &= \left((\hat{\mathbf{x}}_{\mathbf{q}})^{-1} \otimes \dot{\mathbf{x}}_{\mathbf{q}} - (\hat{\mathbf{x}}_{\mathbf{q}})^{-1} \otimes \dot{\hat{\mathbf{x}}}_{\mathbf{q}} \otimes (\delta\mathbf{x}_{\boldsymbol{\theta}})^{\wedge}\right)^{\vee} \\ &= \big((\hat{\mathbf{x}}_{\mathbf{q}})^{-1} \otimes \mathbf{f}_{\mathbf{q}}(\mathbf{x}, \mathbf{u} + \boldsymbol{\upsilon}) \\ &\quad - (\hat{\mathbf{x}}_{\mathbf{q}})^{-1} \otimes \mathbf{f}_{\mathbf{q}}(\hat{\mathbf{x}}, \mathbf{u}) \otimes (\delta\mathbf{x}_{\boldsymbol{\theta}})^{\wedge}\big)^{\vee} \\ &\triangleq \bar{\mathbf{f}}_{\mathbf{q}}(\delta\mathbf{x}, \boldsymbol{\upsilon}, \hat{\mathbf{x}}, \mathbf{u})\ ,\end{aligned}$$

where again $\mathbf{x}$ can be rewritten in terms of $\delta\mathbf{x}$ and $\hat{\mathbf{x}}$ using (29a) and (44a). Note that we have omitted the process noise $\boldsymbol{\eta}$ since the quaternion propagation dynamics in this paper are exact kinematic expressions. Evaluating $\bar{\mathbf{f}}_{\mathbf{q}}$ at $(E[\delta\mathbf{x}], E[\boldsymbol{\upsilon}], \hat{\mathbf{x}}, \mathbf{u})$ gives

$$\begin{aligned}\bar{\mathbf{f}}_{\mathbf{q}}(E[\delta\mathbf{x}], E[\boldsymbol{\upsilon}], \hat{\mathbf{x}}, \mathbf{u}) &= \big((\hat{\mathbf{x}}_{\mathbf{q}})^{-1} \otimes \mathbf{f}_{\mathbf{q}}(\hat{\mathbf{x}}, \mathbf{u}) \\ &\quad - (\hat{\mathbf{x}}_{\mathbf{q}})^{-1} \otimes \mathbf{f}_{\mathbf{q}}(\hat{\mathbf{x}}, \mathbf{u}) \otimes \mathbf{0}^{\wedge}\big)^{\vee} \\ &= \mathbf{0}\ .\end{aligned}$$

A similar analysis can be carried out with the bRN attitude error definition in (29b).

Appendix C Inertial Dynamics

This appendix derives the error-state dynamics for an inertially-defined state.

C.1 Position

We begin from (44a):

$$\begin{aligned}\delta\dot{\mathbf{p}}_n^b &= \dot{\mathbf{p}}_n^b - \dot{\hat{\mathbf{p}}}_n^b \\ &= \mathbf{R}^{\mathsf{T}}(\mathbf{q}_n^b)\mathbf{v} - \mathbf{R}^{\mathsf{T}}(\hat{\mathbf{q}}_n^b)\hat{\mathbf{v}}\ .\end{aligned}$$

We then use (31a), (34), (44a), and (4) to obtain

$$\begin{aligned}\delta\dot{\mathbf{p}}_n^b &= \mathbf{R}^{\mathsf{T}}(\hat{\mathbf{q}}_n^b)\mathbf{R}^{\mathsf{T}}(\delta\mathbf{q}_n^b)\left(\hat{\mathbf{v}} + \delta\mathbf{v}\right) - \mathbf{R}^{\mathsf{T}}(\hat{\mathbf{q}}_n^b)\hat{\mathbf{v}} \\ &\approx \mathbf{R}^{\mathsf{T}}(\hat{\mathbf{q}}_n^b)\left(\mathbf{I} + \left\lfloor\delta\boldsymbol{\theta}_n^b\right\rfloor\right)\left(\hat{\mathbf{v}} + \delta\mathbf{v}\right) - \mathbf{R}^{\mathsf{T}}(\hat{\mathbf{q}}_n^b)\hat{\mathbf{v}} \\ &= \mathbf{R}^{\mathsf{T}}(\hat{\mathbf{q}}_n^b)\hat{\mathbf{v}} + \mathbf{R}^{\mathsf{T}}(\hat{\mathbf{q}}_n^b)\left\lfloor\delta\boldsymbol{\theta}_n^b\right\rfloor\hat{\mathbf{v}} + \mathbf{R}^{\mathsf{T}}(\hat{\mathbf{q}}_n^b)\delta\mathbf{v} \\ &\qquad + \mathbf{R}^{\mathsf{T}}(\hat{\mathbf{q}}_n^b)\left\lfloor\delta\boldsymbol{\theta}_n^b\right\rfloor\delta\mathbf{v} - \mathbf{R}^{\mathsf{T}}(\hat{\mathbf{q}}_n^b)\hat{\mathbf{v}} \\ &= \mathbf{R}^{\mathsf{T}}(\hat{\mathbf{q}}_n^b)\left\lfloor\delta\boldsymbol{\theta}_n^b\right\rfloor\hat{\mathbf{v}} + \mathbf{R}^{\mathsf{T}}(\hat{\mathbf{q}}_n^b)\delta\mathbf{v} + \mathbf{R}^{\mathsf{T}}(\hat{\mathbf{q}}_n^b)\left\lfloor\delta\boldsymbol{\theta}_n^b\right\rfloor\delta\mathbf{v} \\ &= -\mathbf{R}^{\mathsf{T}}(\hat{\mathbf{q}}_n^b)\left\lfloor\hat{\mathbf{v}}\right\rfloor\delta\boldsymbol{\theta}_n^b + \mathbf{R}^{\mathsf{T}}(\hat{\mathbf{q}}_n^b)\delta\mathbf{v} + \mathbf{R}^{\mathsf{T}}(\hat{\mathbf{q}}_n^b)\left\lfloor\delta\boldsymbol{\theta}_n^b\right\rfloor\delta\mathbf{v} \\ &\approx -\mathbf{R}^{\mathsf{T}}(\hat{\mathbf{q}}_n^b)\left\lfloor\hat{\mathbf{v}}\right\rfloor\delta\boldsymbol{\theta}_n^b + \mathbf{R}^{\mathsf{T}}(\hat{\mathbf{q}}_n^b)\delta\mathbf{v},\end{aligned}$$

where the second-order terms have been dropped.

C.2 Attitude

We begin with (29a) and differentiate with respect to time:

$$\begin{aligned}\mathbf{q}_n^b &= \hat{\mathbf{q}}_n^b \otimes \delta\mathbf{q}_n^b \\ \implies \dot{\mathbf{q}}_n^b &= \dot{\hat{\mathbf{q}}}_n^b \otimes \delta\mathbf{q}_n^b + \hat{\mathbf{q}}_n^b \otimes \delta\dot{\mathbf{q}}_n^b\ .\end{aligned}$$

Multiplying on the left by $(\hat{\mathbf{q}}_n^b)^{-1}$ gives

$$\begin{aligned}(\hat{\mathbf{q}}_n^b)^{-1} \otimes \dot{\mathbf{q}}_n^b &= (\hat{\mathbf{q}}_n^b)^{-1} \otimes \dot{\hat{\mathbf{q}}}_n^b \otimes \delta\mathbf{q}_n^b + \delta\dot{\mathbf{q}}_n^b \\ \implies \delta\dot{\mathbf{q}}_n^b &= (\hat{\mathbf{q}}_n^b)^{-1} \otimes \dot{\mathbf{q}}_n^b - (\hat{\mathbf{q}}_n^b)^{-1} \otimes \dot{\hat{\mathbf{q}}}_n^b \otimes \delta\mathbf{q}_n^b\ .\end{aligned}$$

Using (80b), (81a), (30a) and simplifying yields

$$\begin{aligned}\delta\dot{\mathbf{q}}_n^b &= \frac{1}{2}(\hat{\mathbf{q}}_n^b)^{-1} \otimes \mathbf{q}_n^b \otimes \begin{bmatrix}\boldsymbol{\omega} \\ 0\end{bmatrix} - \frac{1}{2}(\hat{\mathbf{q}}_n^b)^{-1} \otimes \hat{\mathbf{q}}_n^b \otimes \begin{bmatrix}\hat{\boldsymbol{\omega}} \\ 0\end{bmatrix} \otimes \delta\mathbf{q}_n^b \\ &= \frac{1}{2}\delta\mathbf{q}_n^b \otimes \begin{bmatrix}\boldsymbol{\omega} \\ 0\end{bmatrix} - \frac{1}{2}\begin{bmatrix}\hat{\boldsymbol{\omega}} \\ 0\end{bmatrix} \otimes \delta\mathbf{q}_n^b. \qquad (101)\end{aligned}$$

Using (72a) and (73), let

$$\begin{aligned}\delta\boldsymbol{\omega} &\triangleq \boldsymbol{\omega} - \hat{\boldsymbol{\omega}} \qquad (102a) \\ &= (\tilde{\boldsymbol{\omega}} - \boldsymbol{\beta}_{\boldsymbol{\omega}} - \boldsymbol{\upsilon}_{\boldsymbol{\omega}}) - \left(\tilde{\boldsymbol{\omega}} - \hat{\boldsymbol{\beta}}_{\boldsymbol{\omega}}\right) \\ &= -\left(\hat{\boldsymbol{\beta}}_{\boldsymbol{\omega}} + \delta\boldsymbol{\beta}_{\boldsymbol{\omega}}\right) - \boldsymbol{\upsilon}_{\boldsymbol{\omega}} + \hat{\boldsymbol{\beta}}_{\boldsymbol{\omega}} \\ &= -\delta\boldsymbol{\beta}_{\boldsymbol{\omega}} - \boldsymbol{\upsilon}_{\boldsymbol{\omega}}\ . \qquad (102b)\end{aligned}$$

Applying (3) and (102a) to (101) yields

$$\begin{aligned}\delta\dot{\mathbf{q}}_n^b &= \frac{1}{2}\begin{bmatrix}-\left\lfloor\boldsymbol{\omega}\right\rfloor & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^{\mathsf{T}} & 0\end{bmatrix}\delta\mathbf{q}_n^b - \frac{1}{2}\begin{bmatrix}\left\lfloor\hat{\boldsymbol{\omega}}\right\rfloor & \hat{\boldsymbol{\omega}} \\ -\hat{\boldsymbol{\omega}}^{\mathsf{T}} & 0\end{bmatrix}\delta\mathbf{q}_n^b \\ &= \frac{1}{2}\begin{bmatrix}\left\lfloor -2\hat{\boldsymbol{\omega}} + \delta\boldsymbol{\omega}\right\rfloor & \delta\boldsymbol{\omega} \\ -\delta\boldsymbol{\omega}^{\mathsf{T}} & 0\end{bmatrix}\delta\mathbf{q}_n^b\ ,\end{aligned}$$

which implies that

$$\begin{bmatrix}\frac{1}{2}\delta\dot{\boldsymbol{\theta}}_n^b \\ 1\end{bmatrix} = \frac{1}{2}\begin{bmatrix}\left\lfloor -2\hat{\boldsymbol{\omega}} + \delta\boldsymbol{\omega}\right\rfloor & \delta\boldsymbol{\omega} \\ -\delta\boldsymbol{\omega}^{\mathsf{T}} & 0\end{bmatrix}\begin{bmatrix}\frac{1}{2}\delta\boldsymbol{\theta}_n^b \\ 1\end{bmatrix}\ .$$

Dropping the scalar equation and second-order terms yields

$$\begin{aligned}\delta\dot{\boldsymbol{\theta}}_n^b &= \frac{1}{2}\left\lfloor -2\hat{\boldsymbol{\omega}} + \delta\boldsymbol{\omega}\right\rfloor\delta\boldsymbol{\theta}_n^b + \delta\boldsymbol{\omega} \\ &\approx -\left\lfloor\hat{\boldsymbol{\omega}}\right\rfloor\delta\boldsymbol{\theta}_n^b + \delta\boldsymbol{\omega} \\ &\approx -\left\lfloor\tilde{\boldsymbol{\omega}} - \hat{\boldsymbol{\beta}}_{\boldsymbol{\omega}}\right\rfloor\delta\boldsymbol{\theta}_n^b - \delta\boldsymbol{\beta}_{\boldsymbol{\omega}} - \boldsymbol{\upsilon}_{\boldsymbol{\omega}}\ .\end{aligned}$$

C.3 Velocity

We begin by applying (44a) to the dynamics:

$$\begin{aligned}\delta\dot{\mathbf{v}} &= \left\lfloor\mathbf{v}\right\rfloor\boldsymbol{\omega} + \mathbf{R}(\mathbf{q}_n^b)\mathbf{g} + a_z\mathbf{k} - \mu\boldsymbol{\Pi}_{\mathbf{k}}\mathbf{v} + \boldsymbol{\eta}_{\mathbf{v}} \\ &\quad - \left(\left\lfloor\hat{\mathbf{v}}\right\rfloor\hat{\boldsymbol{\omega}} + \mathbf{R}(\hat{\mathbf{q}}_n^b)\mathbf{g} + \hat{a}_z\mathbf{k} - \hat{\mu}\boldsymbol{\Pi}_{\mathbf{k}}\hat{\mathbf{v}}\right) \\ &= \left\lfloor\mathbf{v}\right\rfloor\boldsymbol{\omega} - \left\lfloor\hat{\mathbf{v}}\right\rfloor\hat{\boldsymbol{\omega}} \qquad (103a) \\ &\quad + \mathbf{R}(\mathbf{q}_n^b)\mathbf{g} - \mathbf{R}(\hat{\mathbf{q}}_n^b)\mathbf{g} \qquad (103b) \\ &\quad - \mu\boldsymbol{\Pi}_{\mathbf{k}}\mathbf{v} + \hat{\mu}\boldsymbol{\Pi}_{\mathbf{k}}\hat{\mathbf{v}} \qquad (103c) \\ &\quad + a_z\mathbf{k} - \hat{a}_z\mathbf{k} \qquad (103d) \\ &\quad + \boldsymbol{\eta}_{\mathbf{v}}\ .\end{aligned}$$

We will simplify (103) one term at a time. We expand term (103a) using (44a) and (102a), drop second-order terms, then use (4), (73), and (102b) to obtain

$$\begin{aligned}\lfloor \mathbf{v} \rfloor \boldsymbol{\omega} - \lfloor \hat{\mathbf{v}} \rfloor \hat{\boldsymbol{\omega}} &\approx \lfloor \hat{\mathbf{v}} + \delta\mathbf{v} \rfloor (\hat{\boldsymbol{\omega}} + \delta\boldsymbol{\omega}) - \lfloor \hat{\mathbf{v}} \rfloor (\hat{\boldsymbol{\omega}}) \\ &= \lfloor \delta\mathbf{v} \rfloor \hat{\boldsymbol{\omega}} + \lfloor \hat{\mathbf{v}} \rfloor \delta\boldsymbol{\omega} \\ &= -\lfloor \hat{\boldsymbol{\omega}} \rfloor \delta\mathbf{v} + \lfloor \hat{\mathbf{v}} \rfloor \delta\boldsymbol{\omega} \\ &= -\left\lfloor \tilde{\boldsymbol{\omega}} - \hat{\boldsymbol{\beta}}_{\boldsymbol{\omega}} \right\rfloor \delta\mathbf{v} + \lfloor \hat{\mathbf{v}} \rfloor (-\delta\boldsymbol{\beta}_{\boldsymbol{\omega}} - \boldsymbol{\upsilon}_{\boldsymbol{\omega}}) \\ &= -\left\lfloor \tilde{\boldsymbol{\omega}} - \hat{\boldsymbol{\beta}}_{\boldsymbol{\omega}} \right\rfloor \delta\mathbf{v} - \lfloor \hat{\mathbf{v}} \rfloor \delta\boldsymbol{\beta}_{\boldsymbol{\omega}} - \lfloor \hat{\mathbf{v}} \rfloor \boldsymbol{\upsilon}_{\boldsymbol{\omega}} \ .\end{aligned}$$

For term (103b) we use (31a) and (33) to obtain

$$\begin{aligned}\mathbf{R}(\mathbf{q}^b_n)\mathbf{g} - \mathbf{R}(\hat{\mathbf{q}}^b_n)\mathbf{g} &= \mathbf{R}(\delta\mathbf{q}^b_n)\mathbf{R}(\hat{\mathbf{q}}^b_n)\mathbf{g} - \mathbf{R}(\hat{\mathbf{q}}^b_n)\mathbf{g} \\ &\approx \left(\mathbf{I} - \left\lfloor \delta\boldsymbol{\theta}^b_n \right\rfloor\right) \mathbf{R}(\hat{\mathbf{q}}^b_n)\mathbf{g} - \mathbf{R}(\hat{\mathbf{q}}^b_n)\mathbf{g} \\ &= -\left\lfloor \delta\boldsymbol{\theta}^b_n \right\rfloor \mathbf{R}(\hat{\mathbf{q}}^b_n)\mathbf{g} \\ &= \left\lfloor \mathbf{R}(\hat{\mathbf{q}}^b_n)\mathbf{g} \right\rfloor \delta\boldsymbol{\theta}^b_n \ .\end{aligned}$$

We use (44a) and drop second-order terms to simplify term (103c) as

$$\begin{aligned}-\mu\boldsymbol{\Pi}_{\mathbf{k}}\mathbf{v} + \hat{\mu}\boldsymbol{\Pi}_{\mathbf{k}}\hat{\mathbf{v}} &= -(\hat{\mu} + \delta\mu)\,\boldsymbol{\Pi}_{\mathbf{k}}(\hat{\mathbf{v}} + \delta\mathbf{v}) + \hat{\mu}\boldsymbol{\Pi}_{\mathbf{k}}\hat{\mathbf{v}} \\ &= -\hat{\mu}\boldsymbol{\Pi}_{\mathbf{k}}\delta\mathbf{v} - \delta\mu\boldsymbol{\Pi}_{\mathbf{k}}\hat{\mathbf{v}} - \delta\mu\boldsymbol{\Pi}_{\mathbf{k}}\delta\mathbf{v} \\ &= -\hat{\mu}\boldsymbol{\Pi}_{\mathbf{k}}\delta\mathbf{v} - \delta\mu\boldsymbol{\Pi}_{\mathbf{k}}\hat{\mathbf{v}} \ .\end{aligned}$$

Finally term (103d) is simplified using (72b), (74), and (44a) as

$$\begin{aligned}a_z\mathbf{k} - \hat{a}_z\mathbf{k} &= (\tilde{a}_z - \beta_{\mathbf{a}_z} - \eta_{\mathbf{a}_z})\,\mathbf{k} - \left(\tilde{a}_z - \hat{\beta}_{\mathbf{a}_z}\right)\mathbf{k} \\ &= (-\delta\beta_{\mathbf{a}_z} - \eta_{\mathbf{a}_z})\,\mathbf{k} \\ &= -\mathbf{k}\mathbf{k}^{\mathsf{T}}\delta\boldsymbol{\beta}_{\mathbf{a}} - \mathbf{k}\eta_{\mathbf{a}_z} \ .\end{aligned}$$

Substituting all of these results into (103) and gathering terms gives

$$\begin{aligned}\delta\dot{\mathbf{v}} = {} & \left\lfloor \mathbf{R}(\hat{\mathbf{q}}^b_n)\mathbf{g} \right\rfloor \delta\boldsymbol{\theta}^b_n + \left(-\left\lfloor \tilde{\boldsymbol{\omega}} - \hat{\boldsymbol{\beta}}_{\boldsymbol{\omega}} \right\rfloor - \hat{\mu}\boldsymbol{\Pi}_{\mathbf{k}}\right)\delta\mathbf{v} - \mathbf{k}\mathbf{k}^{\mathsf{T}}\delta\boldsymbol{\beta}_{\mathbf{a}} \\ & - \lfloor \hat{\mathbf{v}} \rfloor \delta\boldsymbol{\beta}_{\boldsymbol{\omega}} - \boldsymbol{\Pi}_{\mathbf{k}}\hat{\mathbf{v}}\delta\mu - \lfloor \hat{\mathbf{v}} \rfloor \boldsymbol{\upsilon}_{\boldsymbol{\omega}} - \mathbf{k}\eta_{\mathbf{a}_z} + \boldsymbol{\eta}_{\mathbf{v}} \ .\end{aligned}$$

Appendix D Body-Fixed Dynamics

This appendix derives the body-fixed state dynamics from their inertial counterparts, and then derives the body-fixed error-state dynamics.

D.1 Position

We represent (90) as

$$\begin{bmatrix}\mathbf{p}^n_b \\ 0\end{bmatrix} = -(\mathbf{q}^b_n)^{-1} \otimes \begin{bmatrix}\mathbf{p}^b_n \\ 0\end{bmatrix} \otimes \mathbf{q}^b_n \ .$$

Left-multiplying by $\mathbf{q}^b_n$, taking the time derivative, and left-multiplying by $(\mathbf{q}^b_n)^{-1}$ results in the following sequence of equations:

$$\begin{aligned}\mathbf{q}^b_n \otimes \begin{bmatrix}\mathbf{p}^n_b \\ 0\end{bmatrix} &= -\begin{bmatrix}\mathbf{p}^b_n \\ 0\end{bmatrix} \otimes \mathbf{q}^b_n \\ \dot{\mathbf{q}}^b_n \otimes \begin{bmatrix}\mathbf{p}^n_b \\ 0\end{bmatrix} + \mathbf{q}^b_n \otimes \begin{bmatrix}\dot{\mathbf{p}}^n_b \\ 0\end{bmatrix} &= -\begin{bmatrix}\dot{\mathbf{p}}^b_n \\ 0\end{bmatrix} \otimes \mathbf{q}^b_n - \begin{bmatrix}\mathbf{p}^b_n \\ 0\end{bmatrix} \otimes \dot{\mathbf{q}}^b_n \\ \frac{1}{2}\mathbf{q}^b_n \otimes \begin{bmatrix}\boldsymbol{\omega} \\ 0\end{bmatrix} \otimes \begin{bmatrix}\mathbf{p}^n_b \\ 0\end{bmatrix} + \mathbf{q}^b_n \otimes \begin{bmatrix}\dot{\mathbf{p}}^n_b \\ 0\end{bmatrix} &= -\begin{bmatrix}\dot{\mathbf{p}}^b_n \\ 0\end{bmatrix} \otimes \mathbf{q}^b_n - \frac{1}{2}\begin{bmatrix}\mathbf{p}^b_n \\ 0\end{bmatrix} \otimes \mathbf{q}^b_n \otimes \begin{bmatrix}\boldsymbol{\omega} \\ 0\end{bmatrix} \\ \frac{1}{2}\begin{bmatrix}\boldsymbol{\omega} \\ 0\end{bmatrix} \otimes \begin{bmatrix}\mathbf{p}^n_b \\ 0\end{bmatrix} + \begin{bmatrix}\dot{\mathbf{p}}^n_b \\ 0\end{bmatrix} &= -(\mathbf{q}^b_n)^{-1} \otimes \begin{bmatrix}\dot{\mathbf{p}}^b_n \\ 0\end{bmatrix} \otimes \mathbf{q}^b_n \\ & \quad - \frac{1}{2}(\mathbf{q}^b_n)^{-1}\begin{bmatrix}\mathbf{p}^b_n \\ 0\end{bmatrix} \otimes \mathbf{q}^b_n \otimes \begin{bmatrix}\boldsymbol{\omega} \\ 0\end{bmatrix} .\end{aligned}$$

Rearranging and using (90), (7), (80a), (3), and (4) gives

$$\begin{aligned}\begin{bmatrix}\dot{\mathbf{p}}^n_b \\ 0\end{bmatrix} &= -(\mathbf{q}^b_n)^{-1} \otimes \begin{bmatrix}\dot{\mathbf{p}}^b_n \\ 0\end{bmatrix} \otimes \mathbf{q}^b_n - \frac{1}{2}(\mathbf{q}^b_n)^{-1}\begin{bmatrix}\mathbf{p}^b_n \\ 0\end{bmatrix} \otimes \mathbf{q}^b_n \otimes \begin{bmatrix}\boldsymbol{\omega} \\ 0\end{bmatrix} \\ & \quad - \frac{1}{2}\begin{bmatrix}\boldsymbol{\omega} \\ 0\end{bmatrix} \otimes \begin{bmatrix}\mathbf{p}^n_b \\ 0\end{bmatrix} \\ &= -\begin{bmatrix}R(\mathbf{q}^b_n)\dot{\mathbf{p}}^b_n \\ 0\end{bmatrix} + \frac{1}{2}\begin{bmatrix}\mathbf{p}^n_b \\ 0\end{bmatrix} \otimes \begin{bmatrix}\boldsymbol{\omega} \\ 0\end{bmatrix} - \frac{1}{2}\begin{bmatrix}\boldsymbol{\omega} \\ 0\end{bmatrix} \otimes \begin{bmatrix}\mathbf{p}^n_b \\ 0\end{bmatrix} \\ &= -\begin{bmatrix}\mathbf{v} \\ 0\end{bmatrix} + \frac{1}{2}\begin{bmatrix}\lfloor \mathbf{p}^n_b \rfloor \boldsymbol{\omega} \\ -\mathbf{p}^{n\mathsf{T}}_b\boldsymbol{\omega}\end{bmatrix} - \frac{1}{2}\begin{bmatrix}\lfloor \boldsymbol{\omega} \rfloor \mathbf{p}^n_b \\ -\boldsymbol{\omega}^{\mathsf{T}}\mathbf{p}^n_b\end{bmatrix} \\ &= -\begin{bmatrix}\mathbf{v} \\ 0\end{bmatrix} - \frac{1}{2}\begin{bmatrix}\lfloor \boldsymbol{\omega} \rfloor \mathbf{p}^n_b \\ \mathbf{p}^{n\mathsf{T}}_b\boldsymbol{\omega}\end{bmatrix} - \frac{1}{2}\begin{bmatrix}\lfloor \boldsymbol{\omega} \rfloor \mathbf{p}^n_b \\ -\boldsymbol{\omega}^{\mathsf{T}}\mathbf{p}^n_b\end{bmatrix} .\end{aligned}$$

Dropping the scalar equation we see that

$$\dot{\mathbf{p}}^n_b = -\mathbf{v} - \lfloor \boldsymbol{\omega} \rfloor \mathbf{p}^n_b \ .$$

Letting $\delta\boldsymbol{\omega} = \boldsymbol{\omega} - \hat{\boldsymbol{\omega}}$, the error-state position dynamics when using the body frame are

$$\begin{aligned}\delta\dot{\mathbf{p}}^n_b &= \dot{\mathbf{p}}^n_b - \dot{\hat{\mathbf{p}}}^n_b \\ &= (-\mathbf{v} - \lfloor \boldsymbol{\omega} \rfloor \mathbf{p}^n_b) - (-\hat{\mathbf{v}} - \lfloor \hat{\boldsymbol{\omega}} \rfloor \hat{\mathbf{p}}^n_b) \\ &= -\hat{\mathbf{v}} - \delta\mathbf{v} - \lfloor \hat{\boldsymbol{\omega}} + \delta\boldsymbol{\omega} \rfloor \hat{\mathbf{p}}^n_b - \lfloor \hat{\boldsymbol{\omega}} + \delta\boldsymbol{\omega} \rfloor \delta\mathbf{p}^n_b \\ & \quad + \hat{\mathbf{v}} + \lfloor \hat{\boldsymbol{\omega}} \rfloor \hat{\mathbf{p}}^n_b \\ . &= -\delta\mathbf{v} - \lfloor \delta\boldsymbol{\omega} \rfloor \hat{\mathbf{p}}^n_b - \lfloor \hat{\boldsymbol{\omega}} \rfloor \delta\mathbf{p}^n_b \\ &= -\delta\mathbf{v} + \lfloor \hat{\mathbf{p}}^n_b \rfloor \delta\boldsymbol{\omega} - \lfloor \hat{\boldsymbol{\omega}} \rfloor \delta\mathbf{p}^n_b \\ &= -\delta\mathbf{v} + \lfloor \hat{\mathbf{p}}^n_b \rfloor (-\delta\boldsymbol{\beta}_{\boldsymbol{\omega}} - \boldsymbol{\upsilon}_{\boldsymbol{\omega}}) - \left\lfloor \tilde{\boldsymbol{\omega}} - \hat{\boldsymbol{\beta}}_{\boldsymbol{\omega}} \right\rfloor \delta\mathbf{p}^n_b \\ &= -\left\lfloor \tilde{\boldsymbol{\omega}} - \hat{\boldsymbol{\beta}}_{\boldsymbol{\omega}} \right\rfloor \delta\mathbf{p}^n_b - \delta\mathbf{v} - \lfloor \hat{\mathbf{p}}^n_b \rfloor \delta\boldsymbol{\beta}_{\boldsymbol{\omega}} - \lfloor \hat{\mathbf{p}}^n_b \rfloor \boldsymbol{\upsilon}_{\boldsymbol{\omega}} \ .\end{aligned}$$

D.2 Attitude

The body-fixed attitude dynamics are defined by (27). They can alternately be derived from (91) and (26) as

$$\begin{aligned}\dot{\mathbf{q}}^n_b &= (\dot{\mathbf{q}}^b_n)^{-1} \\ &= \frac{1}{2}\left(\mathbf{q}^b_n \otimes \begin{bmatrix}\boldsymbol{\omega} \\ 0\end{bmatrix}\right)^{-1} \\ &= \frac{1}{2}\begin{bmatrix}-\boldsymbol{\omega} \\ 0\end{bmatrix} \otimes (\mathbf{q}^b_n)^{-1} \\ &= -\frac{1}{2}\begin{bmatrix}\boldsymbol{\omega} \\ 0\end{bmatrix} \otimes \mathbf{q}^n_b \ .\end{aligned}$$

Using (29b), the body-fixed attitude is decomposed into an estimate and error state as

$$\mathbf{q}^n_b = \delta\mathbf{q}^n_b \otimes \hat{\mathbf{q}}^n_b \ .$$

Differentiating with respect to time gives

$$\dot{\mathbf{q}}_b^n = \delta\dot{\mathbf{q}}_b^n \otimes \hat{\mathbf{q}}_b^n + \delta\mathbf{q}_b^n \otimes \dot{\hat{\mathbf{q}}}_b^n \; .$$

Solving for the attitude error dynamics and using (92a) (29b), and (3) results in

$$\begin{aligned}
\delta\dot{\mathbf{q}}_b^n &= \left(\dot{\mathbf{q}}_b^n - \delta\mathbf{q}_b^n \otimes \dot{\hat{\mathbf{q}}}_b^n\right) \otimes (\hat{\mathbf{q}}_b^n)^{-1} \\
&= -\frac{1}{2}\left(\begin{bmatrix}\boldsymbol{\omega} \\ 0\end{bmatrix} \otimes \mathbf{q}_b^n - \delta\mathbf{q}_b^n \otimes \begin{bmatrix}\hat{\boldsymbol{\omega}} \\ 0\end{bmatrix} \otimes \hat{\mathbf{q}}_b^n\right) \otimes (\hat{\mathbf{q}}_b^n)^{-1} \\
&= -\frac{1}{2}\left(\begin{bmatrix}\boldsymbol{\omega} \\ 0\end{bmatrix} \otimes \delta\mathbf{q}_b^n - \delta\mathbf{q}_b^n \otimes \begin{bmatrix}\hat{\boldsymbol{\omega}} \\ 0\end{bmatrix}\right) \\
&= -\frac{1}{2}\left(\begin{bmatrix}\lfloor\boldsymbol{\omega}\rfloor & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^\mathsf{T} & 0\end{bmatrix} \delta\mathbf{q}_b^n - \begin{bmatrix}-\lfloor\hat{\boldsymbol{\omega}}\rfloor & \hat{\boldsymbol{\omega}} \\ -\hat{\boldsymbol{\omega}}^\mathsf{T} & 0\end{bmatrix} \delta\mathbf{q}_b^n\right) \\
&= -\frac{1}{2}\begin{bmatrix}\lfloor 2\hat{\boldsymbol{\omega}} + \delta\boldsymbol{\omega}\rfloor & \delta\boldsymbol{\omega} \\ -\delta\boldsymbol{\omega}^\mathsf{T} & 0\end{bmatrix} \delta\mathbf{q}_b^n \; ,
\end{aligned}$$

which given (32) implies that

$$\begin{bmatrix}\frac{1}{2}\delta\dot{\boldsymbol{\theta}}_b^n \\ 1\end{bmatrix} = -\frac{1}{2}\begin{bmatrix}\lfloor 2\hat{\boldsymbol{\omega}} + \delta\boldsymbol{\omega}\rfloor & \delta\boldsymbol{\omega} \\ -\delta\boldsymbol{\omega}^\mathsf{T} & 0\end{bmatrix}\begin{bmatrix}\frac{1}{2}\delta\boldsymbol{\theta}_b^n \\ 1\end{bmatrix} \; .$$

Multiplying both sides by two, dropping the scalar term, and ignoring higher order terms yields

$$\begin{aligned}
\delta\dot{\boldsymbol{\theta}}_b^n &= -\begin{bmatrix}\lfloor 2\hat{\boldsymbol{\omega}} + \delta\boldsymbol{\omega}\rfloor & \delta\boldsymbol{\omega}\end{bmatrix}\begin{bmatrix}\frac{1}{2}\delta\boldsymbol{\theta}_b^n \\ 1\end{bmatrix} \\
&= -\frac{1}{2}\lfloor 2\hat{\boldsymbol{\omega}} + \delta\boldsymbol{\omega}\rfloor \delta\boldsymbol{\theta}_b^n - \delta\boldsymbol{\omega} \\
&\approx -\lfloor\hat{\boldsymbol{\omega}}\rfloor \delta\boldsymbol{\theta}_b^n - \delta\boldsymbol{\omega} \\
&= -\left\lfloor \tilde{\boldsymbol{\omega}} - \hat{\boldsymbol{\beta}}_{\boldsymbol{\omega}} \right\rfloor \delta\boldsymbol{\theta}_b^n + \delta\boldsymbol{\beta}_{\boldsymbol{\omega}} + \boldsymbol{\upsilon}_{\boldsymbol{\omega}} \; .
\end{aligned}$$

D.3 Velocity

The only change in the velocity error dynamics is the gravity term, so that

$$\begin{aligned}
\delta\dot{\mathbf{v}} &= \mathbf{R}^\mathsf{T}(\mathbf{q}_b^n)\mathbf{g} - \mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^n)\mathbf{g} \\
&= \mathbf{R}^\mathsf{T}(\delta\mathbf{q}_b^n)\mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^n)\mathbf{g} - \mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^n)\mathbf{g} \\
&\approx (\mathbf{I} + \lfloor\delta\boldsymbol{\theta}_b^n\rfloor)\, \mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^n)\mathbf{g} - \mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^n)\mathbf{g} \\
&= \lfloor\delta\boldsymbol{\theta}_b^n\rfloor\, \mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^n)\mathbf{g} \\
&= -\left\lfloor \mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^n)\mathbf{g} \right\rfloor \delta\boldsymbol{\theta}_b^n \; .
\end{aligned}$$

Appendix E Keyframe-Reset Derivation

During the keyframe reset step, introduced in Section 5 and detailed in Sections 6.3 and 7.3, the relative states and their associated covariance are reset to zero. For the position states, only the altitude of the vehicle is kept, while for the attitude states the uncertainty associated with yaw is removed from the filter while the uncertainty associated with roll and pitch is maintained. Sections E.1 and E.2 show that

$$\mathbf{N}_{\boldsymbol{\theta}} \triangleq \frac{\partial \delta\boldsymbol{\theta}^+}{\partial \delta\boldsymbol{\theta}} = \begin{bmatrix}1 & \sin\hat{\phi}\tan\hat{\theta} & \cos\hat{\phi}\tan\hat{\theta} \\ 0 & \cos^2\hat{\phi} & -\cos\hat{\phi}\sin\hat{\phi} \\ 0 & -\cos\hat{\phi}\sin\hat{\phi} & \sin^2\hat{\phi}\end{bmatrix} \tag{104}$$

for both iRN and bRN attitude definitions. Section E.3 derives the position reset and its Jacobian for bRN.

E.1 iRN Attitude Reset

Recall from (99) in Appendix A that the Euler angle errors $\boldsymbol{\Delta}$ are related to the attitude error state $\delta\boldsymbol{\theta}$ as

$$\boldsymbol{\Delta}_n^b = \begin{bmatrix}1 & \sin\hat{\phi}\tan\hat{\theta} & \cos\hat{\phi}\tan\hat{\theta} \\ 0 & \cos\hat{\phi} & -\sin\hat{\phi} \\ 0 & \sin\hat{\phi}\sec\hat{\theta} & \cos\hat{\phi}\sec\hat{\theta}\end{bmatrix} \delta\boldsymbol{\theta}_n^b \; . \tag{105}$$

The true and estimated attitude states after the keyframe reset step consist of the roll and pitch components of the attitude prior to the reset,

$$\begin{aligned}
{\mathbf{q}_n^b}^+ &= \mathbf{q}_\theta \otimes \mathbf{q}_\phi \; , \\
{\hat{\mathbf{q}}_n^b}{}^+ &= \hat{\mathbf{q}}_\theta \otimes \hat{\mathbf{q}}_\phi \; .
\end{aligned}$$

The resulting attitude error is obtained from (30a) and (29a) as

$$\begin{aligned}
{\delta\mathbf{q}_n^b}^+ &= ({\hat{\mathbf{q}}_n^b}{}^+)^{-1} \otimes {\mathbf{q}_n^b}^+ \\
&= \hat{\mathbf{q}}_\phi^{-1} \otimes \hat{\mathbf{q}}_\theta^{-1} \otimes \mathbf{q}_\theta \otimes \mathbf{q}_\phi \\
&= \hat{\mathbf{q}}_\phi^{-1} \otimes \hat{\mathbf{q}}_\theta^{-1} \otimes \hat{\mathbf{q}}_\theta \otimes \delta\mathbf{q}_\theta \otimes \hat{\mathbf{q}}_\phi \otimes \delta\mathbf{q}_\phi \\
&= \left(\hat{\mathbf{q}}_\phi^{-1} \otimes \delta\mathbf{q}_\theta \otimes \hat{\mathbf{q}}_\phi\right) \otimes \delta\mathbf{q}_\phi \; ,
\end{aligned}$$

which, similar to (42), implies that

$${\delta\boldsymbol{\theta}_n^b}^+ = \mathbf{R}_\phi \delta\boldsymbol{\theta}_\theta + \delta\boldsymbol{\theta}_\phi \; .$$

Expanding and factoring out the Euler angle errors, this becomes

$${\delta\boldsymbol{\theta}_n^b}^+ = \begin{bmatrix}1 & 0 & 0 \\ 0 & \cos\hat{\phi} & 0 \\ 0 & -\sin\hat{\phi} & 0\end{bmatrix} \boldsymbol{\Delta}_n^b \; , \tag{106}$$

which differs from (98) by removing the yaw and its uncertainty. Substituting (105) into (106) we see that

$${\delta\boldsymbol{\theta}_n^b}^+ = \begin{bmatrix}1 & \sin\hat{\phi}\tan\hat{\theta} & \cos\hat{\phi}\tan\hat{\theta} \\ 0 & \cos^2\hat{\phi} & -\cos\hat{\phi}\sin\hat{\phi} \\ 0 & -\cos\hat{\phi}\sin\hat{\phi} & \sin^2\hat{\phi}\end{bmatrix} \delta\boldsymbol{\theta}_n^b \; , \tag{107}$$

verifying (104) for iRN.

E.2 bRN Attitude Reset

Continuing the notation presented in Section 7, we describe bRN attitude error with $\delta\mathbf{q}_b^n$ and $\delta\boldsymbol{\theta}_b^n$. The typical 3-2-1 Euler angles assume an inertial attitude. To express a body-fixed attitude while still maintaining the intuitive roll, pitch, and yaw rotations, the order and sign of the rotation sequence must be flipped as

$$\begin{aligned}
\mathbf{q}_b^n &= \left(\mathbf{q}_n^b\right)^{-1} \\
&= \left(\mathbf{q}_\psi \otimes \mathbf{q}_\theta \otimes \mathbf{q}_\phi\right)^{-1} \\
&= {\mathbf{q}_\phi}^{-1} \otimes {\mathbf{q}_\theta}^{-1} \otimes {\mathbf{q}_\psi}^{-1} \; .
\end{aligned}$$

Expanding using the error state definition (30b),

$$\mathbf{q}_b^n = {\hat{\mathbf{q}}_\phi}^{-1} \otimes {\delta\mathbf{q}_\phi}^{-1} \otimes {\hat{\mathbf{q}}_\theta}^{-1} \otimes {\delta\mathbf{q}_\theta}^{-1} \otimes {\hat{\mathbf{q}}_\psi}^{-1} \otimes {\delta\mathbf{q}_\psi}^{-1} \; , \tag{108}$$

which is approximated as

$$\hat{\mathbf{q}}_b^n = \hat{\mathbf{q}}_\phi{}^{-1} \otimes \hat{\mathbf{q}}_\theta{}^{-1} \otimes \hat{\mathbf{q}}_\psi{}^{-1} \ . \tag{109}$$

Combining (108) and (109) using (30b) gives

$$\begin{aligned}
\delta\mathbf{q}_b^n &= \mathbf{q}_b^n \otimes (\hat{\mathbf{q}}_b^n)^{-1} \\
&= \hat{\mathbf{q}}_\phi{}^{-1} \otimes \delta\mathbf{q}_\phi{}^{-1} \otimes \hat{\mathbf{q}}_\theta{}^{-1} \otimes \delta\mathbf{q}_\theta{}^{-1} \otimes \\
&\qquad \otimes \hat{\mathbf{q}}_\psi{}^{-1} \otimes \delta\mathbf{q}_\psi{}^{-1} \otimes \hat{\mathbf{q}}_\psi \otimes \hat{\mathbf{q}}_\theta \otimes \hat{\mathbf{q}}_\phi \\
&= \hat{\mathbf{q}}_\phi{}^{-1} \otimes \Big(\delta\mathbf{q}_\phi{}^{-1} \otimes \hat{\mathbf{q}}_\theta{}^{-1} \otimes \big(\delta\mathbf{q}_\theta{}^{-1} \otimes \\
&\qquad \otimes \hat{\mathbf{q}}_\psi{}^{-1} \otimes \delta\mathbf{q}_\psi{}^{-1} \otimes \hat{\mathbf{q}}_\psi\big) \otimes \hat{\mathbf{q}}_\theta\Big) \otimes \hat{\mathbf{q}}_\phi \ ,
\end{aligned}$$

which implies that

$$\delta\boldsymbol{\theta}_b^n = -\mathbf{R}_\phi \delta\boldsymbol{\theta}_\phi - \mathbf{R}_\phi \mathbf{R}_\theta \delta\boldsymbol{\theta}_\theta - \mathbf{R}_\phi \mathbf{R}_\theta \mathbf{R}_\psi \delta\boldsymbol{\theta}_\psi \ . \tag{110}$$

Note that

$$\begin{aligned}
\mathbf{R}_\phi \delta\boldsymbol{\theta}_\phi &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \phi - \hat{\phi} \\ 0 \\ 0 \end{bmatrix} \\
&= \delta\boldsymbol{\theta}_\phi \ .
\end{aligned}$$

Similarly, $\mathbf{R}_\theta \delta\boldsymbol{\theta}_\theta = \delta\boldsymbol{\theta}_\theta$, and $\mathbf{R}_\psi \delta\boldsymbol{\theta}_\psi = \delta\boldsymbol{\theta}_\psi$. Using these results to simplify (110) and comparing the resulting equation to (97) for iRN, we obtain

$$\begin{aligned}
\delta\boldsymbol{\theta}_b^n &= -\delta\boldsymbol{\theta}_\phi - \mathbf{R}_\phi \delta\boldsymbol{\theta}_\theta - \mathbf{R}_\phi \mathbf{R}_\theta \delta\boldsymbol{\theta}_\psi \\
&= -\delta\boldsymbol{\theta}_n^b \ .
\end{aligned} \tag{111}$$

Again, the reset step keeps only roll and pitch, giving

$$\begin{aligned}
\delta\boldsymbol{\theta}_b^{n+} &= -\delta\boldsymbol{\theta}_\phi - \mathbf{R}_\phi \delta\boldsymbol{\theta}_\theta \\
&= -\delta\boldsymbol{\theta}_n^{b\,+} \ .
\end{aligned} \tag{112}$$

Substituting (112) and (111) into (107), we see that

$$\begin{aligned}
&\left(-\delta\boldsymbol{\theta}_b^{n+}\right) = \begin{bmatrix} 1 & \sin\hat{\phi}\tan\hat{\theta} & \cos\hat{\phi}\tan\hat{\theta} \\ 0 & \cos^2\hat{\phi} & -\cos\hat{\phi}\sin\hat{\phi} \\ 0 & -\cos\hat{\phi}\sin\hat{\phi} & \sin^2\hat{\phi} \end{bmatrix} (-\delta\boldsymbol{\theta}_b^n) \\
&\implies \delta\boldsymbol{\theta}_b^{n+} = \mathbf{N}_{\boldsymbol{\theta}} \delta\boldsymbol{\theta}_b^n \ ,
\end{aligned} \tag{113}$$

which confirms (104) for bRN. Note that the values of $\hat{\phi}$ and $\hat{\theta}$ used to evaluate $\mathbf{N}_{\boldsymbol{\theta}}$ are obtained from the Euler decomposition of $(\hat{\mathbf{q}}_b^n)^{-1}$.

E.3 bRN Position Reset

Following (90) we can write

$$\begin{aligned}
\mathbf{p}_b^{n+} &= -\mathbf{R}(\mathbf{q}_n^{b\,+})\mathbf{p}_n^{b\,+} \\
&= -\mathbf{R}^\mathsf{T}(\mathbf{q}_b^{n+})\mathbf{p}_n^{b\,+} \ ,
\end{aligned}$$

where

$$\begin{aligned}
\mathbf{p}_n^{b\,+} &= \mathbf{k}\mathbf{k}^\mathsf{T}\mathbf{p}_n^b \\
&= -\mathbf{k}\mathbf{k}^\mathsf{T}\mathbf{R}^\mathsf{T}(\mathbf{q}_n^b)\mathbf{p}_b^n \\
&= -\mathbf{k}\mathbf{k}^\mathsf{T}\mathbf{R}(\mathbf{q}_b^n)\mathbf{p}_b^n
\end{aligned} \tag{114}$$

by rearranging (90). We note from (91) and (38) that

$$\begin{aligned}
&\mathbf{q}_b^n = \mathbf{q}_\phi{}^{-1} \otimes \mathbf{q}_\theta{}^{-1} \otimes \mathbf{q}_\psi{}^{-1} \\
\implies &\mathbf{R}(\mathbf{q}_b^n) = \mathbf{R}^\mathsf{T}(\mathbf{q}_\psi)\mathbf{R}^\mathsf{T}(\mathbf{q}_\theta)\mathbf{R}^\mathsf{T}(\mathbf{q}_\phi) \ ,
\end{aligned}$$

so (114) becomes

$$\mathbf{p}_n^{b\,+} = -\mathbf{k}\mathbf{k}^\mathsf{T}\mathbf{R}^\mathsf{T}(\mathbf{q}_\psi)\mathbf{R}^\mathsf{T}(\mathbf{q}_\theta)\mathbf{R}^\mathsf{T}(\mathbf{q}_\phi)\mathbf{p}_b^n \ .$$

We next observe that

$$\begin{aligned}
\mathbf{k}\mathbf{k}^\mathsf{T}\mathbf{R}^\mathsf{T}(\mathbf{q}_\psi) &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
&= \mathbf{k}\mathbf{k}^\mathsf{T} \ ,
\end{aligned}$$

so that

$$\mathbf{p}_n^{b\,+} = -\mathbf{k}\mathbf{k}^\mathsf{T}\mathbf{R}^\mathsf{T}(\mathbf{q}_\theta)\mathbf{R}^\mathsf{T}(\mathbf{q}_\phi)\mathbf{p}_b^n \ .$$

Noting that

$$\begin{aligned}
&\mathbf{q}_b^{n+} = \mathbf{q}_\phi{}^{-1} \otimes \mathbf{q}_\theta{}^{-1} \\
\implies &\mathbf{R}(\mathbf{q}_b^{n+}) = \mathbf{R}^\mathsf{T}(\mathbf{q}_\theta)\mathbf{R}^\mathsf{T}(\mathbf{q}_\phi) \ ,
\end{aligned}$$

we have

$$\mathbf{p}_n^{b\,+} = -\mathbf{k}\mathbf{k}^\mathsf{T}\mathbf{R}(\mathbf{q}_b^{n+})\mathbf{p}_b^n$$

and so

$$\mathbf{p}_b^{n+} = \mathbf{R}^\mathsf{T}(\mathbf{q}_b^{n+})\mathbf{k}\mathbf{k}^\mathsf{T}\mathbf{R}(\mathbf{q}_b^{n+})\mathbf{p}_b^n \ . \tag{115}$$

We then expand and simplify according to (31b), (33), and (34) while dropping second-order terms as

$$\begin{aligned}
\mathbf{p}_b^{n+} &= \mathbf{R}^\mathsf{T}(\delta\mathbf{q}_b^{n+})\mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^{n+})\mathbf{k}\mathbf{k}^\mathsf{T}\mathbf{R}(\hat{\mathbf{q}}_b^{n+})\mathbf{R}(\delta\mathbf{q}_b^{n+})\mathbf{p}_b^n \\
&\approx \left(\mathbf{I} + \left\lfloor \delta\boldsymbol{\theta}_b^{n+} \right\rfloor\right) \mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^{n+})\mathbf{k}\mathbf{k}^\mathsf{T}\mathbf{R}(\hat{\mathbf{q}}_b^{n+}) \left(\mathbf{I} - \left\lfloor \delta\boldsymbol{\theta}_b^{n+} \right\rfloor\right) \left(\hat{\mathbf{p}}_b^n + \delta\mathbf{p}_n^b\right) \\
&\approx \mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^{n+})\mathbf{k}\mathbf{k}^\mathsf{T}\mathbf{R}(\hat{\mathbf{q}}_b^{n+})\hat{\mathbf{p}}_b^n \\
&\qquad + \mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^{n+})\mathbf{k}\mathbf{k}^\mathsf{T}\mathbf{R}(\hat{\mathbf{q}}_b^{n+})\delta\mathbf{p}_b^n \\
&\qquad + \left\lfloor \delta\boldsymbol{\theta}_b^{n+} \right\rfloor \mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^{n+})\mathbf{k}\mathbf{k}^\mathsf{T}\mathbf{R}(\hat{\mathbf{q}}_b^{n+})\hat{\mathbf{p}}_b^n \\
&\qquad - \mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^{n+})\mathbf{k}\mathbf{k}^\mathsf{T}\mathbf{R}(\hat{\mathbf{q}}_b^{n+}) \left\lfloor \delta\boldsymbol{\theta}_b^{n+} \right\rfloor \hat{\mathbf{p}}_b^n .
\end{aligned}$$

From (115) the estimated reset is

$$\hat{\mathbf{p}}_b^{n+} = \mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^{n+})\mathbf{k}\mathbf{k}^\mathsf{T}\mathbf{R}(\hat{\mathbf{q}}_b^{n+})\hat{\mathbf{p}}_b^n \ .$$

Using (4) and (113), the error state reset is then

$$\begin{aligned}
\delta\mathbf{p}_b^{n+} &= \mathbf{p}_b^{n+} - \hat{\mathbf{p}}_b^{n+} \\
&= \mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^{n+})\mathbf{k}\mathbf{k}^\mathsf{T}\mathbf{R}(\hat{\mathbf{q}}_b^{n+})\delta\mathbf{p}_b^n \\
&\qquad + \left\lfloor \delta\boldsymbol{\theta}_b^{n+} \right\rfloor \mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^{n+})\mathbf{k}\mathbf{k}^\mathsf{T}\mathbf{R}(\hat{\mathbf{q}}_b^{n+})\hat{\mathbf{p}}_b^n \\
&\qquad - \mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^{n+})\mathbf{k}\mathbf{k}^\mathsf{T}\mathbf{R}(\hat{\mathbf{q}}_b^{n+}) \left\lfloor \delta\boldsymbol{\theta}_b^{n+} \right\rfloor \hat{\mathbf{p}}_b^n \\
&= \mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^{n+})\mathbf{k}\mathbf{k}^\mathsf{T}\mathbf{R}(\hat{\mathbf{q}}_b^{n+})\delta\mathbf{p}_b^n \\
&\qquad - \left\lfloor \mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^{n+})\mathbf{k}\mathbf{k}^\mathsf{T}\mathbf{R}(\hat{\mathbf{q}}_b^{n+})\hat{\mathbf{p}}_b^n \right\rfloor \delta\boldsymbol{\theta}_b^{n+} \\
&\qquad + \mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^{n+})\mathbf{k}\mathbf{k}^\mathsf{T}\mathbf{R}(\hat{\mathbf{q}}_b^{n+}) \lfloor \hat{\mathbf{p}}_b^n \rfloor \delta\boldsymbol{\theta}_b^{n+} \\
&= \mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^{n+})\mathbf{k}\mathbf{k}^\mathsf{T}\mathbf{R}(\hat{\mathbf{q}}_b^{n+})\delta\mathbf{p}_b^n \\
&\qquad - \left\lfloor \mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^{n+})\mathbf{k}\mathbf{k}^\mathsf{T}\mathbf{R}(\hat{\mathbf{q}}_b^{n+})\hat{\mathbf{p}}_b^n \right\rfloor \mathbf{N}_{\boldsymbol{\theta}} \delta\boldsymbol{\theta}_b^n \\
&\qquad + \mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^{n+})\mathbf{k}\mathbf{k}^\mathsf{T}\mathbf{R}(\hat{\mathbf{q}}_b^{n+}) \lfloor \hat{\mathbf{p}}_b^n \rfloor \mathbf{N}_{\boldsymbol{\theta}} \delta\boldsymbol{\theta}_b^n \ ,
\end{aligned}$$

and the non-zero Jacobian terms are

$$\begin{aligned}\frac{\partial \delta \mathbf{p}_b^{n+}}{\partial \delta \mathbf{p}_b^n} &= \mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^{n+})\mathbf{k}\mathbf{k}^\mathsf{T}\mathbf{R}(\hat{\mathbf{q}}_b^{n+}) \\ \frac{\partial \delta \mathbf{p}_b^{n+}}{\partial \delta \boldsymbol{\theta}_b^n} &= \Big(- \Big\lfloor \mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^{n+})\mathbf{k}\mathbf{k}^\mathsf{T}\mathbf{R}(\hat{\mathbf{q}}_b^{n+})\hat{\mathbf{p}}_b^n \Big\rfloor \\ &\qquad + \mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^{n+})\mathbf{k}\mathbf{k}^\mathsf{T}\mathbf{R}(\hat{\mathbf{q}}_b^{n+}) \lfloor \hat{\mathbf{p}}_b^n \rfloor \Big)\mathbf{N}_{\boldsymbol{\theta}} \ .\end{aligned}$$

Appendix F Body-Fixed Laser Scan Measurement Model

In this section we derive the residual Jacobians for the body-fixed laser scan measurement model presented in Section 7.2.5. For the translation portion of the measurement model, we begin by expanding equation (93) according to (31b), (33), and (34) then drop second-order terms and use (4) to obtain

$$\begin{aligned}\mathbf{h}_{\text{lt}}(\mathbf{x}) =& \mathbf{I}_{2\times 3}\mathbf{R}(\mathbf{q}_b^o)\Big(\mathbf{p}_b^n - \mathbf{p}_b^c + \mathbf{R}^\mathsf{T}(\mathbf{q}_b^n)\mathbf{R}(\mathbf{q}_k^n)\,(\mathbf{p}_b^c - \mathbf{p}_k^n)\,\Big) + \boldsymbol{\eta}_{\text{lt}} \\ =& \mathbf{I}_{2\times 3}\mathbf{R}(\hat{\mathbf{q}}_b^o)\mathbf{R}(\delta\mathbf{q}_b^o)\Big(\hat{\mathbf{p}}_b^n + \delta\mathbf{p}_b^n - \mathbf{p}_b^c \\ &+ \mathbf{R}^\mathsf{T}(\delta\mathbf{q}_b^n)\mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n)\mathbf{R}(\delta\mathbf{q}_k^n) \\ &\times(\mathbf{p}_b^c - \hat{\mathbf{p}}_k^n - \delta\mathbf{p}_k^n)\,\Big) + \boldsymbol{\eta}_{\text{lt}} \\ =& \mathbf{I}_{2\times 3}\mathbf{R}(\hat{\mathbf{q}}_b^o)\,(\mathbf{I} - \lfloor\delta\boldsymbol{\theta}_b^o\rfloor)\,\Big(\hat{\mathbf{p}}_b^n + \delta\mathbf{p}_b^n - \mathbf{p}_b^c \\ &+ (\mathbf{I} + \lfloor\delta\boldsymbol{\theta}_b^n\rfloor)\,\mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n)\,(\mathbf{I} - \lfloor\delta\boldsymbol{\theta}_k^n\rfloor) \\ &\times(\mathbf{p}_b^c - \hat{\mathbf{p}}_k^n - \delta\mathbf{p}_k^n)\,\Big) + \boldsymbol{\eta}_{\text{lt}} \\ =& \mathbf{I}_{2\times 3}\mathbf{R}(\hat{\mathbf{q}}_b^o)\Big(\hat{\mathbf{p}}_b^n - \mathbf{p}_b^c + \mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n)\,(\mathbf{p}_b^c - \hat{\mathbf{p}}_k^n)\,\Big) \\ &+ \mathbf{I}_{2\times 3}\mathbf{R}(\hat{\mathbf{q}}_b^o)\delta\mathbf{p}_b^n - \mathbf{I}_{2\times 3}\mathbf{R}(\hat{\mathbf{q}}_b^o)\mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n)\delta\mathbf{p}_k^n \\ &- \mathbf{I}_{2\times 3}\mathbf{R}(\hat{\mathbf{q}}_b^o)\,\Big\lfloor\mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n)\,(\mathbf{p}_b^c - \hat{\mathbf{p}}_k^n)\Big\rfloor\,\delta\boldsymbol{\theta}_b^n \\ &+ \mathbf{I}_{2\times 3}\mathbf{R}(\hat{\mathbf{q}}_b^o)\mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n)\,\lfloor\mathbf{p}_b^c - \hat{\mathbf{p}}_k^n\rfloor\,\delta\boldsymbol{\theta}_k^n \\ &+ \mathbf{I}_{2\times 3}\mathbf{R}(\hat{\mathbf{q}}_b^o)\,\Big\lfloor\hat{\mathbf{p}}_b^n - \mathbf{p}_b^c + \mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n)\,(\mathbf{p}_b^c - \hat{\mathbf{p}}_k^n)\Big\rfloor\,\delta\boldsymbol{\theta}_b^o.\end{aligned}$$

Recalling that

$$\mathbf{h}_{\text{lt}}(\hat{\mathbf{x}}) = \mathbf{I}_{2\times 3}\mathbf{R}(\hat{\mathbf{q}}_b^o)\Big(\hat{\mathbf{p}}_b^n - \mathbf{p}_b^c + \mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n)\,(\mathbf{p}_b^c - \hat{\mathbf{p}}_k^n)\,\Big),$$

the residual is given by

$$\mathbf{h}_{\text{lt}} = \mathbf{z}_{\text{lt}} - \mathbf{h}_{\text{lt}}(\hat{\mathbf{x}}) \ ,$$

which is modeled as

$$\begin{aligned}\mathbf{r}_{\text{lt}} =& \mathbf{h}_{\text{lt}}(\mathbf{x}) - \mathbf{h}_{\text{lt}}(\hat{\mathbf{x}}) \\ =& \mathbf{I}_{2\times 3}\mathbf{R}(\hat{\mathbf{q}}_b^o)\delta\mathbf{p}_b^n - \mathbf{I}_{2\times 3}\mathbf{R}(\hat{\mathbf{q}}_b^o)\mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n)\delta\mathbf{p}_k^n \\ &- \mathbf{I}_{2\times 3}\mathbf{R}(\hat{\mathbf{q}}_b^o)\,\Big\lfloor\mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n)\,(\mathbf{p}_b^c - \hat{\mathbf{p}}_k^n)\Big\rfloor\,\delta\boldsymbol{\theta}_b^n \\ &+ \mathbf{I}_{2\times 3}\mathbf{R}(\hat{\mathbf{q}}_b^o)\mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n)\,\lfloor\mathbf{p}_b^c - \hat{\mathbf{p}}_k^n\rfloor\,\delta\boldsymbol{\theta}_k^n \\ &+ \mathbf{I}_{2\times 3}\mathbf{R}(\hat{\mathbf{q}}_b^o)\,\Big\lfloor\hat{\mathbf{p}}_b^n - \mathbf{p}_b^c + \mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n)\,(\mathbf{p}_b^c - \hat{\mathbf{p}}_k^n)\Big\rfloor\,\delta\boldsymbol{\theta}_b^o.\end{aligned}$$

Noting that $\mathbf{q}_b^o$ is a function of $\mathbf{q}_b^n$ only, we take partial derivatives and apply the chain rule to obtain the following non-zero Jacobian terms:

$$\begin{aligned}\frac{\partial \mathbf{r}_{\text{lt}}}{\partial \delta \mathbf{p}_b^n} =& \mathbf{I}_{2\times 3}\mathbf{R}(\hat{\mathbf{q}}_b^o), \\ \frac{\partial \mathbf{r}_{\text{lt}}}{\partial \delta \boldsymbol{\theta}_b^n} =& -\mathbf{I}_{2\times 3}\mathbf{R}(\hat{\mathbf{q}}_b^o)\,\Big\lfloor\mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n)\,(\mathbf{p}_b^c - \hat{\mathbf{p}}_k^n)\Big\rfloor \\ &+ \mathbf{I}_{2\times 3}\mathbf{R}(\hat{\mathbf{q}}_b^o)\,\Big\lfloor\hat{\mathbf{p}}_b^n - \mathbf{p}_b^c + \mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n)\,(\mathbf{p}_b^c - \hat{\mathbf{p}}_k^n)\Big\rfloor\,\frac{\partial \delta \boldsymbol{\theta}_b^o}{\partial \delta \boldsymbol{\theta}_b^n}, \\ \frac{\partial \mathbf{r}_{\text{lt}}}{\partial \delta \mathbf{p}_k^n} =& -\mathbf{I}_{2\times 3}\mathbf{R}(\hat{\mathbf{q}}_b^o)\mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n), \\ \frac{\partial \mathbf{r}_{\text{lt}}}{\partial \delta \boldsymbol{\theta}_k^n} =& \mathbf{I}_{2\times 3}\mathbf{R}(\hat{\mathbf{q}}_b^o)\mathbf{R}^\mathsf{T}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n)\,\lfloor\mathbf{p}_b^c - \hat{\mathbf{p}}_k^n\rfloor\,.\end{aligned}$$

To derive an expression for $\partial \delta\boldsymbol{\theta}_b^o / \partial \delta\boldsymbol{\theta}_b^n$, we follow a process similar to that used for the attitude reset derivation in Sections E.1 and E.2. We expand the definition of $\mathbf{q}_b^o$ according to (29b) as

$$\begin{aligned}\mathbf{q}_b^o &= (\mathbf{q}_{\theta_n^b} \otimes \mathbf{q}_{\phi_n^b})^{-1} \\ &= (\hat{\mathbf{q}}_{\phi_n^b})^{-1} \otimes (\delta\mathbf{q}_{\phi_n^b})^{-1} \otimes (\hat{\mathbf{q}}_{\theta_n^b})^{-1} \otimes (\delta\mathbf{q}_{\theta_n^b})^{-1} \ .\end{aligned}$$

Noting that $\hat{\mathbf{q}}_b^o = (\hat{\mathbf{q}}_{\theta_n^b} \otimes \hat{\mathbf{q}}_{\phi_n^b})^{-1}$, the error quaternion is computed according to (30b) as

$$\begin{aligned}\delta\mathbf{q}_b^o &= \mathbf{q}_b^o \otimes (\hat{\mathbf{q}}_b^o)^{-1} \\ &= (\hat{\mathbf{q}}_{\phi_n^b})^{-1} \otimes (\delta\mathbf{q}_{\phi_n^b})^{-1} \\ &\qquad \otimes (\hat{\mathbf{q}}_{\theta_n^b})^{-1} \otimes (\delta\mathbf{q}_{\theta_n^b})^{-1} \otimes \hat{\mathbf{q}}_{\theta_n^b} \otimes \hat{\mathbf{q}}_{\phi_n^b} \ ,\end{aligned}$$

from which we use (42) to obtain

$$\delta\boldsymbol{\theta}_b^o = \mathbf{R}(\hat{\mathbf{q}}_{\phi_n^b})\,\Big(-\delta\boldsymbol{\theta}_{\phi_n^b} - \mathbf{R}(\hat{\mathbf{q}}_{\theta_n^b}\,\delta\boldsymbol{\theta}_{\theta_n^b}\Big) \ .$$

Noting, as shown in Section E.2, that $\mathbf{R}(\hat{\mathbf{q}}_{\phi_n^b})\delta\boldsymbol{\theta}_{\phi_n^b} = \delta\boldsymbol{\theta}_{\phi_n^b}$ and $\mathbf{R}(\hat{\mathbf{q}}_{\phi_n^b})\delta\boldsymbol{\theta}_{\theta_n^b} = \delta\boldsymbol{\theta}_{\theta_n^b}$, this becomes

$$\delta\boldsymbol{\theta}_b^o = -\delta\boldsymbol{\theta}_{\phi_n^b} - \mathbf{R}(\hat{\mathbf{q}}_{\phi_n^b})\delta\boldsymbol{\theta}_{\theta_n^b} \ .$$

Expanding $\mathbf{R}(\hat{\mathbf{q}}_{\phi_n^b})$ according to (36) and combining terms, this can be written as

$$\delta\boldsymbol{\theta}_b^o = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -\cos\hat{\phi}_n^b & 0 \\ 0 & \sin\hat{\phi}_n^b & 0 \end{bmatrix} \boldsymbol{\Delta}_n^b \ ,$$

where

$$\boldsymbol{\Delta} = \begin{bmatrix} \phi_n^b - \hat{\phi}_n^b \\ \theta_n^b - \hat{\theta}_n^b \\ \psi_n^b - \hat{\psi}_n^b \end{bmatrix}$$

as in Appendix A. We then use the chain rule to obtain

$$\begin{aligned}\frac{\partial \delta\boldsymbol{\theta}_b^o}{\partial \delta\boldsymbol{\theta}_b^n} &= \frac{\partial \delta\boldsymbol{\theta}_b^o}{\partial \boldsymbol{\Delta}_b^n}\frac{\partial \boldsymbol{\Delta}_b^n}{\partial \delta\boldsymbol{\theta}_n^b}\frac{\partial \delta\boldsymbol{\theta}_n^b}{\partial \delta\boldsymbol{\theta}_b^n} \\ &= \begin{bmatrix} -1 & 0 & 0 \\ 0 & -\cos\hat{\phi}_n^b & 0 \\ 0 & \sin\hat{\phi}_n^b & 0 \end{bmatrix} \mathbf{N}_{\boldsymbol{\Delta}_n^b}\,(-\mathbf{I}) \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\hat{\phi}_n^b & 0 \\ 0 & -\sin\hat{\phi}_n^b & 0 \end{bmatrix} \mathbf{N}_{\boldsymbol{\Delta}_n^b} \ ,\end{aligned}$$

where $\mathbf{N}_{\boldsymbol{\Delta}_n^b}$ is given by equation (100) in Appendix A. We then substitute this result above to obtain the complete expression for $\partial \mathbf{r}_{\text{lt}} / \partial \delta\boldsymbol{\theta}_b^n$.

For the rotation portion of the measurement model, we begin by expanding the expression for $\mathbf{q}_{\text{lr}}$ according to (29b) as

$$\begin{aligned}\mathbf{q}_{\text{lr}} &= (\mathbf{q}_b^o)^{-1} \otimes \mathbf{q}_b^n \otimes (\mathbf{q}_k^n)^{-1} \otimes \mathbf{q}_b^o \\ &= (\hat{\mathbf{q}}_b^o)^{-1} \otimes (\delta\mathbf{q}_b^o)^{-1} \otimes \delta\mathbf{q}_b^n \\ &\quad \otimes \hat{\mathbf{q}}_b^n \otimes (\hat{\mathbf{q}}_k^n)^{-1} \otimes (\delta\mathbf{q}_k^n)^{-1} \otimes \delta\mathbf{q}_b^o \otimes \hat{\mathbf{q}}_b^o \ .\end{aligned}$$

Recalling that $\hat{\mathbf{q}}_{\text{lr}} = (\hat{\mathbf{q}}_b^o)^{-1} \otimes \hat{\mathbf{q}}_b^n \otimes (\hat{\mathbf{q}}_k^n)^{-1} \otimes \hat{\mathbf{q}}_b^o$, the error quaternion is computed according to (30b) as

$$\begin{aligned}\delta\mathbf{q}_{\text{lr}} &= \mathbf{q}_{\text{lr}} \otimes (\hat{\mathbf{q}}_{\text{lr}})^{-1} \\ &= (\hat{\mathbf{q}}_b^o)^{-1} \otimes (\delta\mathbf{q}_b^o)^{-1} \otimes \delta\mathbf{q}_b^n \otimes \hat{\mathbf{q}}_b^n \otimes (\hat{\mathbf{q}}_k^n)^{-1} \\ &\quad \otimes (\delta\mathbf{q}_k^n)^{-1} \otimes \delta\mathbf{q}_b^o \otimes \hat{\mathbf{q}}_b^n \otimes (\hat{\mathbf{q}}_k^n)^{-1} \otimes \hat{\mathbf{q}}_b^o \ ,\end{aligned}$$

from which we use (42) to obtain

$$\begin{aligned}\delta\boldsymbol{\theta}_{\text{lr}} &= \mathbf{R}(\hat{\mathbf{q}}_b^o)\Big(- \delta\boldsymbol{\theta}_b^o + \delta\boldsymbol{\theta}_b^n + \\ &\qquad \mathbf{R}^{\mathsf{T}}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n)\left(-\delta\boldsymbol{\theta}_k^n + \delta\boldsymbol{\theta}_b^o\right)\Big) \\ &= \mathbf{R}(\hat{\mathbf{q}}_b^o)\delta\boldsymbol{\theta}_b^n - \mathbf{R}(\hat{\mathbf{q}}_b^o)\mathbf{R}^{\mathsf{T}}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n)\delta\boldsymbol{\theta}_k^n \\ &\quad + \mathbf{R}(\hat{\mathbf{q}}_b^o)\left(\mathbf{R}^{\mathsf{T}}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n) - \mathbf{I}\right)\delta\boldsymbol{\theta}_b^o \ .\end{aligned}$$

The measurement model is the yaw portion of $\mathbf{q}_{\text{lr}}$,

$$\begin{aligned}h_{\text{lr}}(\mathbf{x}) &= \psi(\mathbf{q}_{\text{lr}}) + \eta_{\text{lr}} \\ &\triangleq \psi_{\text{lr}} + \eta_{\text{lr}} \ ,\end{aligned}$$

and the estimated measurement is

$$\begin{aligned}h_{\text{lr}}(\hat{\mathbf{x}}) &= \psi(\hat{\mathbf{q}}_{\text{lr}}) \\ &\triangleq \hat{\psi}_{\text{lr}} \ .\end{aligned}$$

The residual is given by

$$r_{\text{lr}} = z_{\text{lr}} - h_{\text{lr}}(\hat{\mathbf{x}}) \ ,$$

which is modeled as

$$\begin{aligned}r_{\text{lr}} &= h_{\text{lr}}(\mathbf{x}) - h_{\text{lr}}(\hat{\mathbf{x}}) \\ &= \psi_{\text{lr}} - \hat{\psi}_{\text{lr}} \\ &= \mathbf{k}^{\mathsf{T}}\boldsymbol{\Delta}_{\text{lr}} \ .\end{aligned}$$

We then use the chain rule to compute the Jacobian of r_{lr} as

$$\begin{aligned}\frac{\partial r_{\text{lr}}}{\partial\delta\mathbf{x}} &= \frac{\partial r_{\text{lr}}}{\partial\boldsymbol{\Delta}_{\text{lr}}}\frac{\partial\boldsymbol{\Delta}_{\text{lr}}}{\partial\delta\boldsymbol{\theta}_{\text{lr}}}\frac{\partial\delta\boldsymbol{\theta}_{\text{lr}}}{\partial\delta\mathbf{x}} \\ &= \left(\mathbf{k}^{\mathsf{T}}\right)\left(\mathbf{N}_{\boldsymbol{\Delta}_{\text{lr}}}\right)\frac{\partial\delta\boldsymbol{\theta}_{\text{lr}}}{\partial\delta\mathbf{x}} \ ,\end{aligned}$$

which has non-zero terms

$$\begin{aligned}\frac{\partial r_{\text{lr}}}{\partial\delta\boldsymbol{\theta}_b^n} &= \mathbf{k}^{\mathsf{T}}\mathbf{N}_{\boldsymbol{\Delta}_{\text{lr}}}\mathbf{R}(\hat{\mathbf{q}}_b^o) \\ &\quad \times \left(\mathbf{I} + \left(\mathbf{R}^{\mathsf{T}}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n) - \mathbf{I}\right)\frac{\partial\delta\boldsymbol{\theta}_b^o}{\partial\delta\boldsymbol{\theta}_b^n}\right) \ , \\ \frac{\partial r_{\text{lr}}}{\partial\delta\boldsymbol{\theta}_k^n} &= -\mathbf{k}^{\mathsf{T}}\mathbf{N}_{\boldsymbol{\Delta}_{\text{lr}}}\mathbf{R}(\hat{\mathbf{q}}_b^o)\mathbf{R}^{\mathsf{T}}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\hat{\mathbf{q}}_k^n) \ ,\end{aligned}$$

into which we substitute the expression for $\partial\delta\boldsymbol{\theta}_b^o/\partial\delta\boldsymbol{\theta}_b^n$ derived above.

Appendix G Converting from bRN to iRN

When estimating in the body frame (bRN), it is often useful to transform the estimated state into an inertial frame (iRN) for visualization or path planning and control. This section describes how to accomplish this transformation, and derives the Jacobians for transforming the bRN covariance into the iRN frame.

The velocity and bias states ($\mathbf{v}$, $\boldsymbol{\beta}_{\boldsymbol{\omega}}$, and $\boldsymbol{\beta}_{\mathbf{a}}$) are expressed in the body frame for both iRN and bRN, and so require no transformation. The drag coefficient term μ likewise requires no transformation since it is a scalar term. The states that require transformation are the relative pose of the vehicle $(\mathbf{p}_b^n, \mathbf{q}_b^n)$ and the keyframe pose $(\mathbf{p}_k^n, \mathbf{q}_k^n)$.

The vehicle position $\mathbf{p}_b^n$ is transformed using (90) as

$$\mathbf{p}_n^b = -\mathbf{R}(\mathbf{q}_b^n)\mathbf{p}_b^n \ .$$

Expanding according to (31b), (44a), and (33), and then using (4), this becomes

$$\begin{aligned}\mathbf{p}_n^b &= \mathbf{R}(\hat{\mathbf{q}}_b^n)\mathbf{R}(\delta\mathbf{q}_b^n)\left(\hat{\mathbf{p}}_b^n + \delta\mathbf{p}_b^n\right) \\ &\approx \mathbf{R}(\hat{\mathbf{q}}_b^n)\left(\mathbf{I} - \lfloor\delta\boldsymbol{\theta}_b^n\rfloor\right)\left(\hat{\mathbf{p}}_b^n + \delta\mathbf{p}_b^n\right) \\ &\approx \mathbf{R}(\hat{\mathbf{q}}_b^n)\hat{\mathbf{p}}_b^n + \mathbf{R}(\hat{\mathbf{q}}_b^n)\delta\mathbf{p}_b^n - \mathbf{R}(\hat{\mathbf{q}}_b^n)\lfloor\delta\boldsymbol{\theta}_b^n\rfloor\hat{\mathbf{p}}_b^n \\ &= \mathbf{R}(\hat{\mathbf{q}}_b^n)\hat{\mathbf{p}}_b^n + \mathbf{R}(\hat{\mathbf{q}}_b^n)\delta\mathbf{p}_b^n + \mathbf{R}(\hat{\mathbf{q}}_b^n)\lfloor\hat{\mathbf{p}}_b^n\rfloor\delta\boldsymbol{\theta}_b^n \ .\end{aligned}$$

The estimated state is transformed as

$$\hat{\mathbf{p}}_n^b = -\mathbf{R}(\hat{\mathbf{q}}_b^n)\hat{\mathbf{p}}_b^n \ ,$$

and the resulting error state is

$$\begin{aligned}\delta\mathbf{p}_n^b &= \mathbf{p}_n^b - \hat{\mathbf{p}}_n^b \\ &= \mathbf{R}(\hat{\mathbf{q}}_b^n)\delta\mathbf{p}_b^n + \mathbf{R}(\hat{\mathbf{q}}_b^n)\lfloor\hat{\mathbf{p}}_b^n\rfloor\delta\boldsymbol{\theta}_b^n \ .\end{aligned}$$

The non-zero Jacobian terms are then

$$\begin{aligned}\frac{\partial\delta\mathbf{p}_n^b}{\partial\delta\mathbf{p}_b^n} &= \mathbf{R}(\hat{\mathbf{q}}_b^n) \ , \\ \frac{\partial\delta\mathbf{p}_n^b}{\partial\delta\boldsymbol{\theta}_b^n} &= \mathbf{R}(\hat{\mathbf{q}}_b^n)\lfloor\hat{\mathbf{p}}_b^n\rfloor \ .\end{aligned}$$

The vehicle attitude $\mathbf{q}_b^n$ is transformed as

$$\mathbf{q}_n^b = (\mathbf{q}_b^n)^{-1} \ ,$$

which is expanded according to (29b) as

$$\begin{aligned}\mathbf{q}_n^b &= (\delta\mathbf{q}_b^n \otimes \hat{\mathbf{q}}_b^n)^{-1} \\ &= (\hat{\mathbf{q}}_b^n)^{-1} \otimes (\delta\mathbf{q}_b^n)^{-1} \ .\end{aligned}$$

The estimated attitude is transformed as

$$\hat{\mathbf{q}}_n^b = (\hat{\mathbf{q}}_b^n)^{-1} \ ,$$

and from (30a) the resulting error state is

$$\begin{aligned}\delta\mathbf{q}_n^b &= (\hat{\mathbf{q}}_n^b)^{-1} \otimes \mathbf{q}_n^b \\ &= ((\hat{\mathbf{q}}_b^n)^{-1})^{-1} \otimes (\hat{\mathbf{q}}_b^n)^{-1} \otimes (\delta\mathbf{q}_b^n)^{-1} \\ &= (\delta\mathbf{q}_b^n)^{-1} \ ,\end{aligned}$$

which implies that

$$\delta\boldsymbol{\theta}_n^b = -\delta\boldsymbol{\theta}_b^n \ .$$

The non-zero Jacobian term is

$$\frac{\partial\delta\boldsymbol{\theta}_n^b}{\partial\delta\boldsymbol{\theta}_b^n} = -\mathbf{I} \ .$$

The keyframe pose $(\mathbf{p}_k^n, \mathbf{q}_k^n)$ is transformed similarly, yielding

$$\begin{aligned} \hat{\mathbf{p}}_n^k &= -\mathbf{R}(\hat{\mathbf{q}}_k^n)\hat{\mathbf{p}}_k^n \ , \\ \hat{\mathbf{q}}_n^k &= (\hat{\mathbf{q}}_k^n)^{-1} \ , \end{aligned}$$

and

$$\begin{aligned} \frac{\partial\delta\mathbf{p}_n^k}{\partial\delta\mathbf{p}_k^n} &= \mathbf{R}(\hat{\mathbf{q}}_k^n) \ , \\ \frac{\partial\delta\mathbf{p}_n^k}{\partial\delta\boldsymbol{\theta}_k^n} &= \mathbf{R}(\hat{\mathbf{q}}_k^n) \left\lfloor \hat{\mathbf{p}}_k^n \right\rfloor \ , \\ \frac{\partial\delta\boldsymbol{\theta}_n^k}{\partial\delta\boldsymbol{\theta}_k^n} &= -\mathbf{I} \ . \end{aligned}$$

The covariance is transformed as

$$\mathbf{P}_{\mathrm{iRN}} = \mathbf{J}\mathbf{P}_{\mathrm{bRN}}\mathbf{J}^{\mathsf{T}} \ ,$$

where

$$\begin{aligned} \mathbf{J} &= \frac{\partial\delta\mathbf{x}_{\mathrm{iRN}}}{\partial\delta\mathbf{x}_{\mathrm{bRN}}} \\ &= \begin{bmatrix} \mathbf{R}(\hat{\mathbf{q}}_b^n) & \mathbf{R}(\hat{\mathbf{q}}_b^n)\left\lfloor \hat{\mathbf{p}}_b^n \right\rfloor & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & 0 \\ \mathbf{0} & -\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{R}(\hat{\mathbf{q}}_k^n) & \mathbf{R}(\hat{\mathbf{q}}_k^n)\left\lfloor \hat{\mathbf{p}}_k^n \right\rfloor & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & 1 \end{bmatrix} . \end{aligned}$$

References

Anderson ML, Brink KM and Willis AR (2019) Real-time visual odometry covariance estimation for unmanned air vehicle navigation. *Journal of Guidance, Control, and Dynamics* 42(6): 1272–1288.

Bailey T and Durrant-Whyte H (2006) Simultaneous localization and mapping (SLAM): Part II. *IEEE Robotics and Automation Magazine* 13(3): 108–117.

Bar-Shalom Y, Kirubarajan T and Li XR (2002) *Estimation with Applications to Tracking and Navigation*. New York: John Wiley & Sons, Inc.

Barfoot T, Forbes JR and Furgale PT (2011) Pose estimation using linearized rotations and quaternion algebra. *Acta Astronautica* 68(1–2): 101–112.

Barfoot TD and Furgale PT (2014) Associating uncertainty with three-dimensional poses for use in estimation problems. *IEEE Transactions on Robotics* 30(3): 679–693.

Bloesch M, Omari S, Hutter M and Siegwart R (2015) Robust visual inertial odometry using a direct EKF-based approach. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 298–304.

Bloesch M, Sommer H, Laidlow T, Burri M, Nuetzi G, Fankhauser P, Bellicoso D, Gehring C, Leutenegger S, Hutter M and Siegwart R (2016) A primer on the differential calculus of 3D orientations. *ArXiv e-prints* Available at `https://arxiv.org/abs/1606.05285`.

Bopardikar SD, Zhang S and Speranzon A (2013) A robust and adaptive framework for localization under varying sensor modalities. In: *AIAA Guidance, Navigation, and Control Conference*.

Casey RT, Karpenko M, Curry R and Elkaim G (2013) Attitude representation and kinematic propagation for low-cost UAVs. In: *AIAA Guidance, Navigation, and Control Conference*. pp. 1–15.

Castellanos JA, Martinez-Cantin R, Tardós JD and Neira J (2007) Robocentric map joining: Improving the consistency of EKF-SLAM. *Robotics and Autonomous Systems* 55(1): 21–29.

Censi A (2008) An ICP variant using a point-to-line metric. In: *IEEE International Conference on Robotics and Automation*. pp. 19–25.

Chong KS and Kleeman L (1999) Feature-based mapping in real, large scale environments using an ultrasonic array. *International Journal of Robotics Research* 18(1): 3–19.

Farrell J (2008) *Aided Navigation: GPS with High Rate Sensors*. 1st edition. New York, NY, USA: McGraw-Hill, Inc.

Hertzberg C, Wagner R, Frese U and Schröder L (2013) Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds. *Information Fusion* 14(1): 57–77.

Jones E, Vedaldi A and Soatto S (2007) Inertial structure from motion with autocalibration. In: *ICCV Workshop on Dynamical Vision*.

Julier SJ and Uhlmann JK (2001) A counter example to the theory of simultaneous localization and map building. In: *IEEE International Conference on Robotics and Automation*, volume 4. pp. 4238–4243.

Kim B, Kaess M, Fletcher L, Leonard J, Bachrach A, Roy N and Teller S (2010) Multiple relative pose graphs for robust cooperative mapping. In: *IEEE International Conference on Robotics and Automation*. pp. 3185–3192.

Kottas DG, Hesch JA, Bowman SL and Roumeliotis SI (2013) On the consistency of vision-aided inertial navigation. *Experimental Robotics* : 303–317.

Kuipers JB (1999) *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace, and Virtual Reality*. Princeton: Princeton University Press.

Lefferts E, Markley F and Shuster M (1982) Kalman filtering for spacecraft attitude estimation. *Journal of Guidance, Control, and Dynamics* 5(5): 417–429.

Leishman RC, Macdonald JC, Beard RW and McLain TW (2014a) Quadrotors and accelerometers: State estimation with an improved dynamic model. *IEEE Control Systems Magazine* 34(1): 28–41.

Leishman RC and McLain TW (2014) Multiplicative extended Kalman filter for relative rotorcraft navigation. *Journal of Aerospace Information Systems* 12(12): 728–744.

Leishman RC, McLain TW and Beard RW (2014b) Relative navigation approach for vision-based aerial GPS-denied navigation. *Journal of Intelligent and Robotic Systems* 74(1): 97–111.

Leutenegger S, Lynen S, Bosse M, Siegwart R and Furgale P (2015) Keyframe-based visual-inertial odometry using nonlinear optimization. *International Journal of Robotics Research* 34(3): 314–334.

Markley FL (2003) Attitude error representations for Kalman filtering. *Journal of Guidance, Control, and Dynamics* 26(2): 311–317.

Martinelli A (2012) Vision and IMU data fusion : Closed-form solutions for attitude, speed, absolute scale, and bias determination. *IEEE Transactions on Robotics* 28(1): 44–60.

Maybeck PS (1979) *Stochastic models, estimation, and control*, *Mathematics in Science and Engineering*, volume 141.

Owen E and Montano L (2006) A robocentric motion planner for dynamic environments using the velocity space. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 4368–4374.

Quigley M, Conley K, Gerkey B, Faust J, Foote T, Leibs J, Wheeler R and Ng AY (2009) ROS: An open-source robot operating system. In: *ICRA Workshop on Open Source Software*.

Shen S, Michael N and Kumar V (2011) Autonomous multi-floor indoor navigation with a computationally constrained MAV. In: *IEEE International Conference on Robotics and Automation*. pp. 20–25.

Shen S, Mulgaonkar Y, Michael N and Kumar V (2014) Multi-sensor fusion for robust autonomous flight in indoor and outdoor environments with a rotorcraft MAV. In: *IEEE International Conference on Robotics and Automation*. pp. 4974–4981.

Sola J (2016) Quaternion kinematics for the error-state Kalman filter. Technical Report IRI-TR-16-02, Institut de Robòtica i Informàtica Industrial, CSIC-UPC.

Tomic T, Schmid K, Lutz P, Domel A, Kassecker M, Mair E, Grixa I, Ruess F, Suppa M and Burschka D (2012) Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue. *IEEE Robotics and Automation Magazine* 19(3): 46–56.

Trawny N and Roumeliotis SI (2005) Indirect Kalman filter for 3D attitude estimation: A tutorial for quaternion algebra. Technical Report 2005-002, Rev. 57, University of Minnesota, Department of Computer Science and Engineering, Minneapolis.

Weiss S, Achtelik MW, Lynen S, Chli M and Siegwart R (2012) Real-time onboard visual-inertial state estimation and self-calibration of MAVs in unknown environments. In: *IEEE International Conference on Robotics and Automation*. pp. 957–964.

Weiss S and Siegwart R (2011) Real-time metric state estimation for modular vision-inertial systems. In: *IEEE International Conference on Robotics and Automation*. pp. 4531–4537.

Wheeler DO, Koch DP, Jackson JS, Ellingson GJ, Nyholm PW, McLain TW and Beard RW (2017) Relative navigation of autonomous GPS-degraded micro air vehicles Available at `http://scholarsarchive.byu.edu/facpub/1962/`.

Wheeler DO, Koch DP, Jackson JS, Mclain TW and Beard RW (2018) Relative navigation: A keyframe-based approach for observable GPS-degraded navigation. *IEEE Control Systems Magazine* 38(4): 30–48.

Yu H and Beard RW (2013) Vision-based local-level frame mapping and planning in spherical coordinates for miniature air vehicles. *IEEE Transactions on Control Systems Technology* 21(3): 695–703.

Zhang J, Kaess M and Singh S (2014) Real-time depth enhanced monocular odometry. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 4973–4980.