



All Faculty Publications

2016-6

Multi-Sensor Robust Relative Estimation Framework for GPS-Denied Multirotor Aircraft

Tim McLain

Brigham Young University - Provo, mclain@byu.edu

Daniel P. Koch

Department of Mechanical Engineering, Brigham Young University

See next page for additional authors

Follow this and additional works at: <https://scholarsarchive.byu.edu/facpub>

 Part of the [Mechanical Engineering Commons](#)

Original Publication Citation

Koch, D., McLain, T., and Brink, K. Multi-Sensor Robust Relative Estimation Framework for GPS-Denied Multirotor Aircraft, 2016 International Conference on Unmanned Aircraft Systems, pp. 589-597, June 2016, Arlington, Virginia.

BYU ScholarsArchive Citation

McLain, Tim; Koch, Daniel P.; and Brink, Kevin M., "Multi-Sensor Robust Relative Estimation Framework for GPS-Denied Multirotor Aircraft" (2016). *All Faculty Publications*. 1881.
<https://scholarsarchive.byu.edu/facpub/1881>

This Conference Paper is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in All Faculty Publications by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.

Authors

Tim McLain, Daniel P. Koch, and Kevin M. Brink

Multi-Sensor Robust Relative Estimation Framework for GPS-Denied Multirotor Aircraft

Daniel P. Koch¹, Timothy W. McLain², Kevin M. Brink³

Abstract—An estimation framework is presented that improves the robustness of GPS-denied state estimation to changing environmental conditions by fusing updates from multiple view-based odometry algorithms. This allows the vehicle to utilize a suite of complementary exteroceptive sensors or sensing modalities. By estimating the vehicle states relative to a local coordinate frame collocated with an odometry keyframe, observability of the relative state is maintained. A description of the general framework is given, as well as the specific equations for a multiplicative extended Kalman filter with a multirotor vehicle. Experimental results are presented that demonstrate the ability of the proposed algorithm to produce accurate and consistent estimates in challenging environments that cause a single-sensor solution to fail.

I. INTRODUCTION

GPS-denied navigation solutions for unmanned air vehicles fuse information from two categories of sensors to produce an estimate of the vehicle’s global position. *Proprioceptive* sensors, such as an inertial measurement unit (IMU), measure information that is intrinsic to the vehicle itself. *Exteroceptive* sensors such as cameras and laser scanners, on the other hand, obtain information about the vehicle’s state through observation of the external environment. The most common of these types of observations used in practice are measurements from algorithms such as visual odometry and laser scan matching. These types of measurements will be referred to collectively in this paper as *view-based odometry*, or for brevity simply as odometry.

A number of algorithms for GPS-denied or GPS-degraded navigation have been proposed in the literature. These approaches fuse inertial measurements with odometry computed using a variety of sensors including cameras, laser scanners, and RGB-D cameras. The vast majority of published solutions make use of only a single exteroceptive sensor or sensing modality. While these algorithms have been demonstrated to work well in certain environments, they have proven brittle to changes in environmental conditions. This is a result of limitations of both the sensors (e.g. RGB-D cameras cannot measure depth in direct sunlight) and the odometry algorithms (e.g. long hallways or the 2.5D-world assumption for planar laser scanners). In practice we have found that no one sensor or odometry algorithm is

able to provide sufficient performance in all environmental conditions that a multirotor may experience during a single real-world mission.

By carrying a suite of complementary sensors, a vehicle will be able to obtain good measurements from at least one sensor in a broader range of conditions than if it carried only a single sensor. A single sensor may also be used under a variety of sensing modalities, such as using the monocular image only, depth map only, or the combined modality for an RGB-D camera (e.g. [1]).

Some work has appeared in the recent literature on approaches for combining updates from multiple sensing modalities. Much of this work (e.g. [2]–[5]) fuses updates from view-based odometry algorithms with data from GPS and other sensors as each modality is available. Incorporating updates from multiple view-based odometry algorithms is less common. Separate odometry algorithms are run on the three sensing modalities of an RGB-D camera in [1], and the estimator heuristically switches between these algorithms based on metrics computed on the raw sensor data. In [6] vision and laser-scan measurements are combined in an EKF SLAM framework. The framework described in [5] is able to fuse measurements from multiple view-based odometry algorithms along with GPS measurements and has good robustness to changing conditions. However, the global nature of the estimation scheme has drawbacks in environments where GPS measurement updates are degraded, unavailable for prolonged periods, or absent entirely. This will be discussed in Section II.

The framework described in this paper is designed to fuse measurements from a suite of exteroceptive sensors and their associated odometry algorithms to improve the robustness of the state estimation in challenging and changing GPS-denied environments. It is an extension of the relative navigation paradigm presented in [7], and is designed to be general enough to incorporate updates from any sensor or algorithm that produces relative odometry-like updates. The proposed framework will still fail if all exteroceptive sensors drop out at the same time, but robustness is greatly improved by incorporating measurements from multiple sensors.

A brief overview of the relative navigation paradigm on which this framework is built is given in Section II. A general description of the proposed framework is then presented in Section III. The details of the specific implementation of this framework for a multirotor vehicle are given in Section IV, and experimental results are presented in Section V. Conclusions and future work are discussed in Section VI.

This research was supported in part by AFRL S&T sponsorship.

¹Daniel P. Koch is with the Department of Mechanical Engineering, Brigham Young University, Provo, UT 84602, USA daniel.koch@byu.edu

²Timothy W. McLain is with the Faculty of Mechanical Engineering, Brigham Young University, Provo, UT 84602, USA mclain@byu.edu

³Kevin M. Brink is with the Air Force Research Laboratory, Munitions Directorate, USA

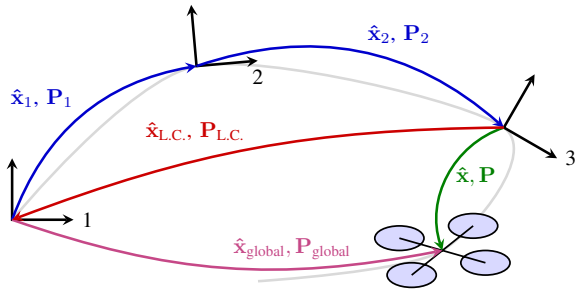


Fig. 1: Illustration of the relative navigation scheme. Axes represent node frames, the green line represents the current relative state estimate, and blue lines represent saved edges in the pose-graph map. The red line represents a loop closure constraint, and the magenta line represents the global state estimate obtained by compounding the relative edges.

II. RELATIVE NAVIGATION SCHEME

This section contains a brief overview of the relative navigation approach to state estimation as previously explained in [7]. The purpose of this section is to lay the groundwork for the extension to this approach that will be presented in the following sections. For more details on the relative navigation approach itself, the reader is referred to [7].

View-based odometry, by its nature, provides only a relative measurement update that gives the translation and rotation between the sensor’s current location and the location at which a *keyframe* image or scan was previously captured. This distinction has important implications when attempting to use these measurements to estimate the vehicle’s global state. No information is actually available from the algorithm about the keyframe’s location relative to the origin of some global coordinate system (the term global here may refer to a coordinate system defined either by an absolute measurement such as GPS, or by the starting location of the vehicle). As a result, the global state of the vehicle is unobservable [8].

The majority of GPS-denied estimation schemes directly estimate the global state of the vehicle. The most common approaches for incorporating these relative measurements into a global state estimate, such as pseudo-global updates [9] or stochastic cloning [5], [10], facilitate the use of these measurements but fail to address the underlying lack of observability and resulting filter inconsistency or unbounded uncertainty.

In the relative navigation approach, the vehicle state is estimated with respect to a local coordinate frame called the *node frame*. This node frame is defined as the local level frame (aligned with the vehicle’s heading, but with the down axis parallel to the gravity vector) collocated with the odometry algorithm’s keyframe. The horizontal translation and the heading of the vehicle are referred to as the relative portion of the state, and are measured relative to this coordinate frame. As a result, the odometry algorithm provides direct measurements of the vehicle state, making the state fully observable.

Figure 1 illustrates how the relative navigation approach

constructs an estimate of the vehicle’s global state. Each time the odometry algorithm declares a new keyframe, the estimator passes its current state and covariance estimates to a back-end map, then declares a new node frame by resetting the relative portions of its state and the associated covariance to zero. The resetting is what causes the node frame to be identically aligned with the odometry keyframe, but with pitch and roll removed. The back-end map chains these relative estimates together as edges in a pose-graph map, from which the global state can be reconstructed at any time. Information such as loop closure constraints or GPS measurements can be incorporated as additional edges in the pose graph map, and an optimization routine can use these constraints to remove global drift from the map. This formulation, as well as the decoupling between the front-end relative state estimates and the back-end global map, provides a number of additional benefits related to estimation and control performance that are discussed in [7].

III. MULTI-SENSOR RELATIVE FRAMEWORK

The relative navigation scheme described in Section II defines the node frame to be identically aligned with the odometry keyframe. As a result, without modification this framework is only able to accept relative updates from a single sensor and odometry algorithm. In this section we present the modification that allows the framework to accept updates from any number of sensors and/or odometry algorithms while still maintaining observability of the relative state and filter consistency.

The proposed approach is similar to the stochastic cloning approaches in [5], [10] in that it augments the state vector with additional information required to apply the measurement updates received from the odometry algorithms. The difference is that these additional states are still relative to the current node frame instead of to a global frame as in the other examples.

The following sections present the general formulation of the multi-sensor relative estimation framework in the context of a continuous-propagation, discrete-update extended Kalman filter (EKF). It is assumed that the vehicle state \mathbf{x} propagates according to the dynamics

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) + \boldsymbol{\eta}, \quad (1)$$

with process noise $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$. The measurement noise on updates from the i th view-based odometry algorithm is assumed Gaussian with covariance matrix \mathbf{R}_i .

The relevant equations are in general nonlinear, but simplified linear equations are also given to aid understanding and intuition. The specific details for a multirotor vehicle are given in Section IV.

A. Augmenting the State Vector

The various coordinate frames, states, and measurements associated with the multi-sensor framework are illustrated in Figure 2. The axes represent the keyframes declared by each of the odometry algorithms at different times as the vehicle traverses the environment. The node frame is again

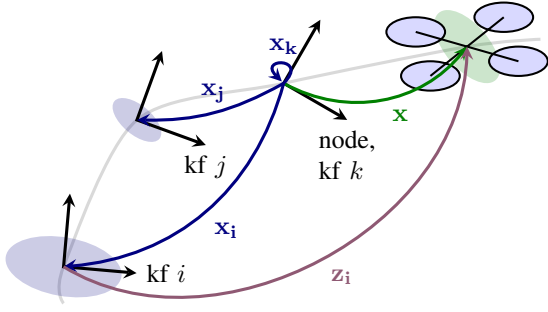


Fig. 2: Depiction of the various states and measurement updates in the multi-sensor estimation framework. Axes represent either the node frame or keyframes as labeled, the green line represents the current relative state, blue lines represent the keyframe states, and the magenta line represents an odometry measurement.

defined as a local level frame, but is now collocated with the most recent keyframe received from *any* of the odometry algorithms (algorithm k for the example in the figure). Measurements received from this odometry can be applied as direct measurements of the relative state, maintaining the observability of that state.

To apply an update from one of the other odometry algorithms, we need to know where that odometry's keyframe is relative to the current node frame. We therefore augment the state vector with the *keyframe states* $\mathbf{x}_{\mathcal{I}} = [\mathbf{x}_1^T \cdots \mathbf{x}_N^T]^T$, one for each odometry algorithm, as

$$\check{\mathbf{x}} = [\mathbf{x}^T \quad \mathbf{x}_1^T \quad \cdots \quad \mathbf{x}_N^T]^T, \quad (2)$$

where \mathbf{x} is the current relative state as described in Section II, and $\check{\mathbf{x}}$ is the augmented state vector. These keyframe states encode the position and orientation of the keyframe to and expressed in the node frame coordinate system. The keyframe coordinate frame is aligned with the body frame of the vehicle at the instant the keyframe image or scan was captured. The covariance is also augmented accordingly as

$$\check{\mathbf{P}} = \begin{bmatrix} \mathbf{P}^{\mathbf{x}\mathbf{x}} & \mathbf{P}^{\mathbf{x}\mathbf{x}_1} & \cdots & \mathbf{P}^{\mathbf{x}\mathbf{x}_N} \\ \mathbf{P}^{\mathbf{x}_1\mathbf{x}} & \mathbf{P}^{\mathbf{x}_1\mathbf{x}_1} & \cdots & \mathbf{P}^{\mathbf{x}_1\mathbf{x}_N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}^{\mathbf{x}_N\mathbf{x}} & \mathbf{P}^{\mathbf{x}_N\mathbf{x}_1} & \cdots & \mathbf{P}^{\mathbf{x}_N\mathbf{x}_N} \end{bmatrix}, \quad (3)$$

where $\mathbf{P}^{\mathbf{x}\mathbf{x}_i}$ is the cross-correlation between the original state \mathbf{x} and the keyframe state \mathbf{x}_i , and so forth.

The keyframe state for the odometry which also defines the current node frame consists only of the altitude, roll, and pitch of the vehicle when the keyframe was declared (the non-relative portion of the state) with the relative portion of the state (horizontal translation and yaw) identically equal to zero.

B. Prediction Step

The augmented state estimates propagate according to the continuous-time dynamics

$$\dot{\check{\mathbf{x}}} = \check{\mathbf{f}}(\check{\mathbf{x}}, \mathbf{u}) = \begin{bmatrix} \mathbf{f}(\hat{\mathbf{x}}, \mathbf{u}) \\ \mathbf{0} \end{bmatrix}, \quad (4)$$

where the time derivatives of the keyframe states are zero since they are not expected to change.

Linearizing these dynamics about the current estimate $\check{\hat{\mathbf{x}}}$ gives the Jacobian

$$\check{\mathbf{F}} = \frac{\partial \check{\mathbf{f}}}{\partial \check{\mathbf{x}}} = \begin{bmatrix} \mathbf{F} \\ \mathbf{0} \end{bmatrix}, \quad (5)$$

where $\mathbf{F} = \frac{\partial \mathbf{f}}{\partial \hat{\mathbf{x}}}$. The augmented process noise matrix can be written as

$$\check{\mathbf{Q}} = \begin{bmatrix} \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad (6)$$

and the augmented covariance propagates according to

$$\dot{\check{\mathbf{P}}} = \check{\mathbf{F}}\check{\mathbf{P}} + \check{\mathbf{P}}\check{\mathbf{F}}^T + \check{\mathbf{Q}} \quad (7)$$

$$= \begin{bmatrix} \mathbf{F}\mathbf{P}^{\mathbf{x}\mathbf{x}} + \mathbf{P}^{\mathbf{x}\mathbf{x}}\mathbf{F}^T + \mathbf{Q} & \mathbf{F}\mathbf{P}^{\mathbf{x}\mathbf{x}_\mathcal{I}} \\ \mathbf{P}^{\mathbf{x}_\mathcal{I}\mathbf{x}}\mathbf{F}^T & \mathbf{0} \end{bmatrix}. \quad (8)$$

C. Measurement Updates

Measurements are processed as they become available using the discrete-time update equations. This allows the filter to accept measurements from odometry algorithms with independent update rates. The full augmented state, including both the normal relative state and the keyframe state, is allowed to change during measurement updates. This allows the estimates of the keyframe locations to be refined when sufficient information is available. Because the covariance associated with relative portions of the keyframe state that is currently coincident with the node frame is zero, this state will not change during updates. This is desirable since those portions of the state are zero by definition to maintain the observability of the system.

Measurement models for sensors besides the view-based odometry algorithms, such as IMUs and altimeters, depend only on the current relative state and not on the keyframe states. As such they can be applied as normal, except that the Jacobian of the measurement model with respect to the state must be augmented on the right by zeros to account for the keyframe states as $\check{\mathbf{H}} = [\mathbf{H} \quad \mathbf{0}]$. The full augmented state and covariance can then be updated using the normal Kalman filter update equations with this augmented Jacobian.

The measurement model for odometry updates depends on both the current relative state and the keyframe states. An odometry measurement gives the current position of the vehicle relative to and expressed in the keyframe coordinate frame, as illustrated in Figure 2. A simple linear measurement model for an update from odometry algorithm i could be given by

$$\hat{\mathbf{z}}_i = \mathbf{\Gamma}(\hat{\mathbf{x}}_i)(\hat{\mathbf{x}} - \hat{\mathbf{x}}_i), \quad (9)$$

where $\mathbf{\Gamma}(\hat{\mathbf{x}}_i)$ is a matrix that rotates the compounded relative state and keyframe state, $\hat{\mathbf{x}} - \hat{\mathbf{x}}_i$, into the coordinate frame of

keyframe i . In practice this compounding and rotation of the current and keyframe states may not be a linear operation (as in Section IV), in which case we write $\hat{\mathbf{z}}_i = \mathbf{h}_{\text{odom}}(\hat{\mathbf{x}}, \hat{\mathbf{x}}_i)$. The Jacobian of this model with respect to the augmented state is computed as

$$\check{\mathbf{H}}_{\text{odom},i} = \begin{bmatrix} \frac{\partial \mathbf{h}_{\text{odom}}}{\partial \hat{\mathbf{x}}} & \mathbf{0} & \cdots & \frac{\partial \mathbf{h}_{\text{odom}}}{\partial \hat{\mathbf{x}}_i} & \cdots & \mathbf{0} \end{bmatrix} \quad (10)$$

and the full augmented state and covariance are then again updated using the normal Kalman filter update equations.

D. Resetting the Relative State

Each time an odometry algorithm declares a new keyframe, the estimator declares a new node frame. There are three actions that must be completed each time this occurs: 1) reset the relative portion of the current relative state estimate $\hat{\mathbf{x}}$ to zero, 2) update each keyframe state estimate $\hat{\mathbf{x}}_i$ to be relative to and expressed in the new node frame coordinate system, and 3) update the augmented covariance matrix $\check{\mathbf{P}}$ accordingly.

The current relative estimate is reset by zeroing the relative portions of the state while not changing the other portions. For the linear case, this can be accomplished as

$$\hat{\mathbf{x}}^+ = \Xi \hat{\mathbf{x}} \quad (11)$$

where Ξ is a matrix of zeros with ones on the diagonal for each of the non-relative states. For the nonlinear case (as in Section IV), we can write $\hat{\mathbf{x}}^+ = \xi(\hat{\mathbf{x}})$. We can then compute the Jacobian $\frac{\partial \xi(\hat{\mathbf{x}})}{\partial \hat{\mathbf{x}}}$ that is needed to update the covariance.

Suppose that odometry algorithm j in Figure 2 declares a new keyframe. The new keyframe for algorithm j is defined to be identically aligned with the body frame at that instant, while the node frame is defined as the current local level frame (no pitch or roll). Therefore any discrepancy between the node frame and keyframe is in the non-relative portions of the state, and is completely defined by the current relative state estimate $\hat{\mathbf{x}}$. We can therefore update the keyframe state for odometry j as

$$\hat{\mathbf{x}}_j^+ = \Xi' \hat{\mathbf{x}}, \quad (12)$$

where the matrix Ξ' is the concatenation of the rows of Ξ corresponding to elements of the keyframe state (again, this operation may be nonlinear in practice, in which case we write $\hat{\mathbf{x}}_j^+ = \xi'(\hat{\mathbf{x}})$). We again compute the Jacobian $\frac{\partial \xi'(\hat{\mathbf{x}})}{\partial \hat{\mathbf{x}}}$ for use in updating the covariance.

The other keyframe states must now be updated to be relative to and expressed in the new node frame. This can be done for the simplified linear case as

$$\hat{\mathbf{x}}_i^+ = \Lambda(\hat{\mathbf{x}}_i)(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}) \quad (13)$$

where $\Lambda(\hat{\mathbf{x}}_i)$ rotates the relative portions of $\hat{\mathbf{x}}_i - \hat{\mathbf{x}}$ into the coordinate system of the new node frame. For the nonlinear case this becomes $\hat{\mathbf{x}}_i^+ = \lambda(\hat{\mathbf{x}}, \hat{\mathbf{x}}_i)$ and the appropriate Jacobians can be computed.

From equations (11), (12), and (13) or their nonlinear equivalents, we can construct the matrix

$$\mathbf{T} = \begin{bmatrix} \frac{\partial \hat{\mathbf{x}}^+}{\partial \hat{\mathbf{x}}} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \frac{\partial \hat{\mathbf{x}}_1^+}{\partial \hat{\mathbf{x}}} & \frac{\partial \hat{\mathbf{x}}_1^+}{\partial \hat{\mathbf{x}}_1} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \frac{\partial \hat{\mathbf{x}}_2^+}{\partial \hat{\mathbf{x}}} & \mathbf{0} & \frac{\partial \hat{\mathbf{x}}_2^+}{\partial \hat{\mathbf{x}}_2} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{\partial \hat{\mathbf{x}}_j^+}{\partial \hat{\mathbf{x}}} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{\partial \hat{\mathbf{x}}_N^+}{\partial \hat{\mathbf{x}}} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \frac{\partial \hat{\mathbf{x}}_N^+}{\partial \hat{\mathbf{x}}_N} \end{bmatrix}, \quad (14)$$

where odometry algorithm j is the one that has declared a new keyframe. The augmented covariance matrix is then updated as

$$\check{\mathbf{P}}^+ = \mathbf{T} \check{\mathbf{P}} \mathbf{T}^T. \quad (15)$$

This process correctly accounts for all of the uncertainty in the system. Because the relative portions of the keyframe state for the odometry that just declared a new keyframe are set to zero, the covariance in these states remains small as each odometry algorithm continues to declare new keyframes.

Another important result of this approach is that since the new keyframe state for an odometry algorithm that has just declared a new keyframe depends only on the current relative state, updates from an odometry algorithm that has been in a failure state and then reinitializes with a new keyframe can be seamlessly integrated.

IV. MULTIROTOR IMPLEMENTATION DETAILS

This section presents the implementation details of the multi-sensor estimation framework for a multirotor vehicle. The attitude of the vehicle is represented as a quaternion, and the estimator is an indirect multiplicative extended Kalman filter following [11]. The term ‘‘multiplicative’’ refers to the definition of the attitude error as the quaternion multiplication of the true and estimated attitude states:

$$\delta \mathbf{q} = \mathbf{q} \otimes \hat{\mathbf{q}}^{-1}. \quad (16)$$

All other non-quaternion error states are defined in the usual way as

$$\delta \mathbf{a} = \mathbf{a} - \hat{\mathbf{a}}. \quad (17)$$

The current relative state of the vehicle is

$$\mathbf{x} = \begin{bmatrix} \mathbf{p}_n^{\text{bT}} & \mathbf{q}_n^{\text{bT}} & \mathbf{v}^{\text{bT}} & \boldsymbol{\beta}^{\text{T}} & \boldsymbol{\alpha}^{\text{T}} \end{bmatrix}^{\text{T}} \quad (18)$$

where \mathbf{p}_n^{b} is the position relative to the node frame, \mathbf{q}_n^{b} is the attitude relative to the node frame, \mathbf{v}^{b} is the body-frame velocity, $\boldsymbol{\beta}$ is the rate gyro biases, and $\boldsymbol{\alpha}$ is the x and y accelerometer biases.

The filter is an indirect filter, and so the covariance of the error state is estimated rather than that of the full state. Because attitude has three degrees of freedom, the 4-element unit quaternion is an over-constrained representation. Therefore rather than estimating the covariance of the quaternion directly, the covariance of the attitude error vector $\delta \boldsymbol{\theta}_n^{\text{b}}$ is

estimated instead, which for small attitude errors is defined as

$$\delta \mathbf{q} \approx \begin{bmatrix} \frac{1}{2} \delta \boldsymbol{\theta} \\ 1 \end{bmatrix}. \quad (19)$$

The current relative error state is then given by

$$\Delta \mathbf{x} = \begin{bmatrix} \delta \mathbf{p}_n^b \top & \delta \boldsymbol{\theta}_n^b \top & \delta \mathbf{v}^b \top & \delta \boldsymbol{\beta} \top & \delta \boldsymbol{\alpha} \top \end{bmatrix}^\top. \quad (20)$$

Each keyframe state is defined as

$$\mathbf{x}_i = \begin{bmatrix} \mathbf{p}_n^i \top & \mathbf{q}_n^i \top \end{bmatrix}^\top, \quad (21)$$

where \mathbf{p}_n^i is the position of keyframe i relative to the node frame, and \mathbf{q}_n^i is the attitude of the keyframe relative to the node frame. The covariance is again estimated for the attitude error vector $\delta \boldsymbol{\theta}_n^i$ and the keyframe error state becomes

$$\Delta \mathbf{x}_i = \begin{bmatrix} \delta \mathbf{p}_n^i \top & \delta \boldsymbol{\theta}_n^i \top \end{bmatrix}^\top. \quad (22)$$

The following sections describe the equations associated with the Kalman filter prediction and measurement update steps as well as those associated with resetting the relative state when a new keyframe is received.

A. Prediction

The state dynamics $\dot{\hat{\mathbf{x}}} = \mathbf{f}(\hat{\mathbf{x}}, \mathbf{u})$ are given by

$$\dot{\hat{\mathbf{p}}}_n^b = \mathbf{R}^\top(\hat{\mathbf{q}}_n^b) \hat{\mathbf{v}}^b \quad (23)$$

$$\dot{\hat{\mathbf{q}}}_n^b = \frac{1}{2} \Omega(\boldsymbol{\omega}^m - \hat{\boldsymbol{\beta}}) \hat{\mathbf{q}}_n^b \quad (24)$$

$$\dot{\hat{\mathbf{v}}}^b = \hat{\mathbf{v}}^b \times (\boldsymbol{\omega}^m - \hat{\boldsymbol{\beta}}) + \mathbf{R}(\hat{\mathbf{q}}_n^b) \mathbf{g} - \mu \mathbf{M} \hat{\mathbf{v}}^b + a_z^m \mathbf{e}_3 \quad (25)$$

$$\dot{\hat{\boldsymbol{\beta}}} = 0 \quad (26)$$

$$\dot{\hat{\boldsymbol{\alpha}}} = 0, \quad (27)$$

where $\boldsymbol{\omega}^m$ is the angular rates measured by the rate gyros, a_z^m is the measured z acceleration, and \mathbf{g} is the gravity vector. The matrix \mathbf{M} is given by $\mathbf{M} = \text{diag}(1/m, 1/m, 0)$, and μ is the rotor drag coefficient as described in [12]. \mathbf{e}_3 is the unit vector $[0 \ 0 \ 1]^\top$, and $\mathbf{R}(\mathbf{q})$ is a rotation matrix that is equivalent to the quaternion rotational operation $\mathbf{q}_x^y \otimes \mathbf{v} \otimes \mathbf{q}_x^y^{-1}$ that rotates the vector \mathbf{v} expressed in frame x to frame y [13]. For a quaternion of the order $\mathbf{q} = [q_x \ q_y \ q_z \ q_w]^\top$, the matrix $\Omega(\boldsymbol{\omega})$ is given in [14] as

$$\Omega(\boldsymbol{\omega}) = \begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix}. \quad (28)$$

From these dynamics the linearized error state dynamics

$$\Delta \dot{\mathbf{x}} = \mathbf{F} \Delta \mathbf{x} + \mathbf{G} \mathbf{u} \quad (29)$$

can be derived, where the Jacobians with respect to the error state \mathbf{F} and input \mathbf{G} are used to propagate the covariance. For details refer to [11].

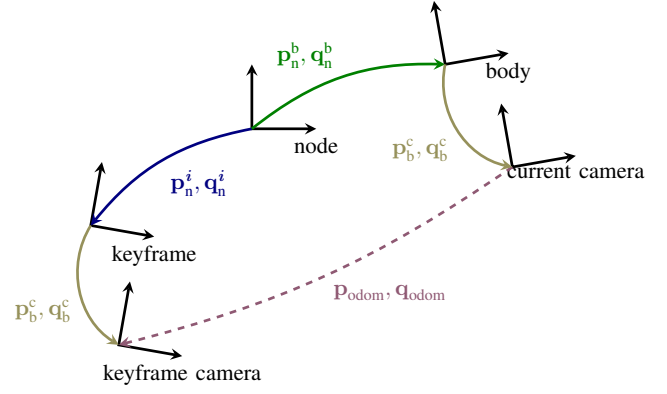


Fig. 3: Transforms associated with an odometry update. Solid lines represent transforms we already know, and dashed lines represent transforms we wish to construct.

B. Measurement Updates

The following sections give the measurement models for the accelerometer, altimeter, and odometry updates. The true and estimated measurement models (\mathbf{h} and $\hat{\mathbf{h}}$ respectively) for each update are given, and the analytic residual $\Delta \mathbf{h} = \mathbf{h} - \hat{\mathbf{h}}$ is derived. The non-zero portions of the Jacobian of the residual with respect to the error state $\check{\mathbf{H}} = \partial \Delta \mathbf{h} / \partial \Delta \mathbf{x}$, needed for the Kalman update step, are then given.

1) *Accelerometer Update:* Only the x and y accelerometer measurements are applied as updates since the z measurement is used in the propagation. The measurement models are given by

$$\mathbf{h}_{\text{acc}} = -\frac{\mu}{m} \mathbf{v}^b_{(1:2)} + \boldsymbol{\alpha} + \boldsymbol{\eta}_{\text{acc}} \quad (30)$$

$$\hat{\mathbf{h}}_{\text{acc}} = -\frac{\mu}{m} \hat{\mathbf{v}}^b_{(1:2)} + \hat{\boldsymbol{\alpha}} \quad (31)$$

with analytic residual

$$\Delta \mathbf{h}_{\text{acc}} = -\frac{\mu}{m} \delta \hat{\mathbf{v}}^b_{(1:2)} + \delta \hat{\boldsymbol{\alpha}} + \boldsymbol{\eta}_{\text{acc}}. \quad (32)$$

The non-zero portions of the Jacobian with respect to the error state are then

$$\frac{\partial \Delta \mathbf{h}_{\text{acc}}}{\partial \delta \mathbf{v}^b} = -\frac{\mu}{m} \mathbf{I}_{2 \times 3} \quad (33)$$

$$\frac{\partial \Delta \mathbf{h}_{\text{acc}}}{\partial \delta \boldsymbol{\alpha}} = \mathbf{I}_{2 \times 2}. \quad (34)$$

2) *Altimeter Update:* The altimeter measurement models are given by

$$h_{\text{alt}} = -p_d + \eta_{\text{alt}} \quad (35)$$

$$\hat{h}_{\text{alt}} = -\hat{p}_d \quad (36)$$

with analytic residual

$$\Delta h_{\text{alt}} = -\delta p_d + \eta_{\text{alt}} \quad (37)$$

and non-zero Jacobian element

$$\frac{\partial \Delta h_{\text{alt}}}{\partial \delta \mathbf{p}_n^b} = [0 \ 0 \ -1]. \quad (38)$$

3) *Odometry Update*: Currently only the translation portions of the odometry update are applied. The transforms associated with a measurement update from a view-based odometry are shown in Figure 3. In our formulation an odometry update provides the transform from the current sensor location to the keyframe sensor location, expressed in the current sensor frame. We also take into account translation and rotational offsets ($\mathbf{p}_b^c, \mathbf{q}_b^c$) of the sensor with respect to the body frame. The true measurement model, equivalent to \mathbf{p}_{odom} in Figure 3, is given by

$$\mathbf{h}_{\text{odom}} = -\mathbf{R}(\mathbf{q}_b^c) \mathbf{p}_b^c - \mathbf{R}(\mathbf{q}_b^c) \mathbf{R}(\mathbf{q}_n^b) (\mathbf{p}_n^b - \mathbf{p}_n^i) + \mathbf{R}(\mathbf{q}_b^c) \mathbf{R}(\mathbf{q}_n^b) \mathbf{R}(\mathbf{q}_n^i) \mathbf{p}_b^c + \boldsymbol{\eta}_{\text{odom}} \quad (39)$$

and the estimated measurement model by

$$\hat{\mathbf{h}}_{\text{odom}} = -\mathbf{R}(\hat{\mathbf{q}}_b^c) \mathbf{p}_b^c - \mathbf{R}(\hat{\mathbf{q}}_b^c) \mathbf{R}(\hat{\mathbf{q}}_n^b) (\hat{\mathbf{p}}_n^b - \hat{\mathbf{p}}_n^i) + \mathbf{R}(\hat{\mathbf{q}}_b^c) \mathbf{R}(\hat{\mathbf{q}}_n^b) \mathbf{R}(\hat{\mathbf{q}}_n^i) \mathbf{p}_b^c. \quad (40)$$

The position and orientation of the sensor in the body frame, \mathbf{p}_b^c and \mathbf{q}_b^c , are assumed to be known from calibration. Using the facts $\mathbf{p} = \hat{\mathbf{p}} + \delta\mathbf{p}$ and $\mathbf{q} = \delta\mathbf{q} \otimes \hat{\mathbf{q}} \implies \mathbf{R}(\mathbf{q}) = \mathbf{R}(\hat{\mathbf{q}}) \mathbf{R}(\delta\mathbf{q})$, identities [14] $\mathbf{R}(\delta\mathbf{q}) \approx \mathbf{I} - [\delta\boldsymbol{\theta}]_{\times}$ and $[\mathbf{a}]_{\times} \mathbf{b} = -\mathbf{a} [\mathbf{b}]_{\times}$ where $[\cdot]_{\times}$ is the skew symmetric matrix operator on a vector, and dropping second-order terms, we can calculate the analytic residual $\Delta\mathbf{h}_{\text{odom}} = \mathbf{h}_{\text{odom}} - \hat{\mathbf{h}}_{\text{odom}}$ as

$$\Delta\mathbf{h}_{\text{odom}} = -\mathbf{R}(\mathbf{q}_b^c) \mathbf{R}(\hat{\mathbf{q}}_n^b) (\delta\mathbf{p}_n^b - \delta\mathbf{p}_n^i) - \mathbf{R}(\mathbf{q}_b^c) \mathbf{R}(\hat{\mathbf{q}}_n^b) [\hat{\mathbf{p}}_n^b - \hat{\mathbf{p}}_n^i - \mathbf{R}(\hat{\mathbf{q}}_n^i) \mathbf{p}_b^c]_{\times} \delta\boldsymbol{\theta}_n^b + \mathbf{R}(\mathbf{q}_b^c) \mathbf{R}(\hat{\mathbf{q}}_n^b) \mathbf{R}(\hat{\mathbf{q}}_n^i) [\mathbf{p}_b^c]_{\times} \delta\boldsymbol{\theta}_n^i + \boldsymbol{\eta}_{\text{odom}}. \quad (41)$$

The non-zero elements of the Jacobian of the residual with respect to the current relative error state are then

$$\frac{\partial \Delta\mathbf{h}_{\text{odom}}}{\partial \delta\mathbf{p}_n^b} = -\mathbf{R}(\hat{\mathbf{q}}_b^c) \mathbf{R}(\hat{\mathbf{q}}_n^b) \quad (42)$$

$$\frac{\partial \Delta\mathbf{h}_{\text{odom}}}{\partial \delta\boldsymbol{\theta}_n^b} = -\mathbf{R}(\hat{\mathbf{q}}_b^c) \mathbf{R}(\hat{\mathbf{q}}_n^b) [\hat{\mathbf{p}}_n^b - \hat{\mathbf{p}}_n^i - \mathbf{R}(\hat{\mathbf{q}}_n^i) \mathbf{p}_b^c]_{\times}, \quad (43)$$

and with respect to the keyframe error state are

$$\frac{\partial \Delta\mathbf{h}_{\text{odom}}}{\partial \delta\mathbf{p}_n^i} = \mathbf{R}(\hat{\mathbf{q}}_b^c) \mathbf{R}(\hat{\mathbf{q}}_n^b) \quad (44)$$

$$\frac{\partial \Delta\mathbf{h}_{\text{odom}}}{\partial \delta\boldsymbol{\theta}_n^i} = \mathbf{R}(\hat{\mathbf{q}}_b^c) \mathbf{R}(\hat{\mathbf{q}}_n^b) \mathbf{R}(\hat{\mathbf{q}}_n^i) [\mathbf{p}_b^c]_{\times}. \quad (45)$$

C. Resetting the Relative State

The transforms associated with receiving a new keyframe, declaring a new node frame, and resetting the relative state are shown in Figure 4. In this example odometry j has declared a new keyframe, while odometry i has not.

We first define $(\mathbf{p}_{\text{edge}}, \mathbf{q}_{\text{edge}})$ as the transform between the current node frame and the new node frame:

$$\mathbf{p}_{\text{edge}} \triangleq \begin{bmatrix} (\mathbf{p}_n^b)_x \\ (\mathbf{p}_n^b)_y \\ 0 \end{bmatrix} \quad (46)$$

$$\mathbf{q}_{\text{edge}} \triangleq \text{Yaw}(\mathbf{q}_n^b) \quad (47)$$

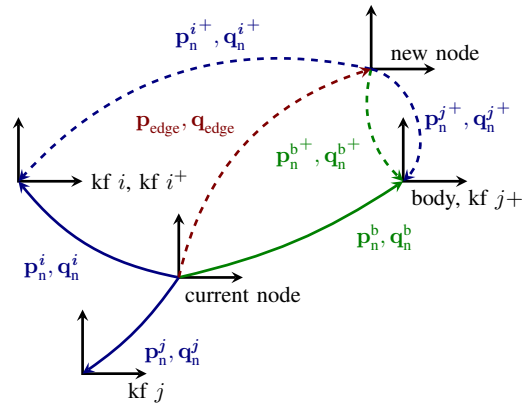


Fig. 4: Transforms associated with resetting the relative state when a new keyframe is received from odometry j . Solid lines represent transforms we already know, and dashed lines represent transforms we wish to construct.

where the operator $\text{Yaw}(\mathbf{q})$ returns a quaternion that is equivalent to only the yaw portion of the rotation defined by \mathbf{q} . The new filter states are then given by

$$\mathbf{p}_n^{b+} \triangleq \begin{bmatrix} 0 \\ 0 \\ (\mathbf{p}_n^b)_z \end{bmatrix} \quad (48)$$

$$\mathbf{q}_n^{b+} \triangleq \text{RollPitch}(\mathbf{q}_n^b) \quad (49)$$

$$\mathbf{p}_n^{j+} \triangleq \mathbf{p}_n^{b+} \quad (50)$$

$$\mathbf{q}_n^{j+} \triangleq \mathbf{q}_n^{b+} \quad (51)$$

$$\mathbf{p}_n^{i+} = \mathbf{R}^T(\mathbf{q}_{\text{edge}}) (\mathbf{p}_n^i - \mathbf{p}_{\text{edge}}) \quad (52)$$

$$\mathbf{q}_n^{i+} = \mathbf{q}_{\text{edge}}^{-1} \otimes \mathbf{q}_n^i. \quad (53)$$

The operator $\text{RollPitch}(\mathbf{q})$ returns a quaternion that is equivalent to only the roll and pitch portions of the rotation defined by \mathbf{q} .

We now wish to compute the Jacobians associated with each of these transformations for use in the matrix \mathbf{T} of equations (14) and (15). The Jacobian for \mathbf{p}_n^{b+} is trivial. Because we represent the covariance of the attitude as the attitude error vector $\delta\boldsymbol{\theta}$, and since we simply zero out the yaw but keep the same roll and pitch, the new attitude error vector is $\delta\boldsymbol{\theta}_n^{b+} = [(\delta\boldsymbol{\theta}_n^b)_x \quad (\delta\boldsymbol{\theta}_n^b)_y \quad 0]^T$. This Jacobian then also becomes trivial. The non-zero elements of the Jacobian for $\delta\mathbf{p}_n^{b+}$ and $\delta\boldsymbol{\theta}_n^{b+}$ are then

$$\frac{\partial (\delta\mathbf{p}_n^{b+})}{\partial (\delta\mathbf{p}_n^b)} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (54)$$

$$\frac{\partial (\delta\boldsymbol{\theta}_n^{b+})}{\partial (\delta\boldsymbol{\theta}_n^b)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (55)$$

$$\frac{\partial (\delta\mathbf{v}^{b+})}{\partial (\delta\mathbf{v}^b)} = \frac{\partial (\delta\boldsymbol{\beta}^+)}{\partial (\delta\boldsymbol{\beta})} = \frac{\partial (\delta\boldsymbol{\alpha}^+)}{\partial (\delta\boldsymbol{\alpha})} = \mathbf{I}. \quad (56)$$

The non-zero elements of the Jacobian for $\delta\mathbf{p}_n^{j+}, \delta\boldsymbol{\theta}_n^{j+}$ are similarly trivial and are given by

$$\frac{\partial(\delta\mathbf{p}_n^{j+})}{\partial(\delta\mathbf{p}_n^b)} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (57)$$

$$\frac{\partial(\delta\boldsymbol{\theta}_n^{j+})}{\partial(\delta\boldsymbol{\theta}_n^b)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (58)$$

The Jacobians for the error state $\delta\mathbf{p}_n^{i+}$ are computed using an argument similar to that used for the odometry update equations as

$$\frac{\partial(\delta\mathbf{p}_n^{i+})}{\partial(\delta\mathbf{p}_n^b)} = -\mathbf{R}(\hat{\mathbf{q}}_{\text{edge}}) \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (59)$$

$$\frac{\partial(\delta\mathbf{p}_n^{i+})}{\partial(\delta\boldsymbol{\theta}_n^b)} = \mathbf{R}(\hat{\mathbf{q}}_{\text{edge}}) \left[\hat{\mathbf{p}}_n^i - \hat{\mathbf{p}}_{\text{edge}}^i \right] \times \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (60)$$

$$\frac{\partial(\delta\mathbf{p}_n^{i+})}{\partial(\delta\mathbf{p}_n^i)} = \mathbf{R}(\hat{\mathbf{q}}_{\text{edge}}). \quad (61)$$

For the rotational portion of the keyframe state $\delta\boldsymbol{\theta}_n^{i+}$, we begin with $\delta\mathbf{q}_n^{i+} = \delta\mathbf{q}_{\text{edge}}^{-1} \otimes \delta\mathbf{q}_n^i$, and expand it as [14]

$$\begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\theta}_n^{i+} \\ 1 \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}\delta\boldsymbol{\theta}_{\text{edge}} \\ 1 \end{bmatrix} \otimes \begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\theta}_n^i \\ 1 \end{bmatrix} \quad (62)$$

$$= \begin{bmatrix} \mathbf{I} - \frac{1}{2} \begin{bmatrix} \delta\boldsymbol{\theta}_{\text{edge}} \\ \delta\boldsymbol{\theta}_{\text{edge}}^\top \end{bmatrix} \times & -\frac{1}{2}\delta\boldsymbol{\theta}_{\text{edge}} \\ & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\theta}_n^i \\ 1 \end{bmatrix}, \quad (63)$$

which after multiplying out and dropping second-order terms produces

$$\delta\boldsymbol{\theta}_n^{i+} = \delta\boldsymbol{\theta}_n^i - \delta\boldsymbol{\theta}_{\text{edge}}. \quad (64)$$

The non-zero elements of the Jacobian can then be computed as

$$\frac{\partial(\delta\boldsymbol{\theta}_n^{i+})}{\partial(\delta\boldsymbol{\theta}_n^b)} = - \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (65)$$

$$\frac{\partial(\delta\boldsymbol{\theta}_n^{i+})}{\partial(\delta\boldsymbol{\theta}_n^i)} = \mathbf{I}. \quad (66)$$

This completes the derivation of the equations and Jacobians used in the multirotor MEKF implementation.

V. EXPERIMENTAL SETUP AND RESULTS

Experimental results were obtained by flying a multirotor vehicle in both a motion capture environment and through a challenging indoor environment. Visual odometry updates were obtained at 15 Hz using the algorithm from [15] with an ASUS Xtion Pro Live RGB-D camera. Laser scan matching updates were obtained at 20 to 40 Hz using the algorithm from [16] with a Hokuyo UTM-30LX planar laser scanner. The vehicle also carried a MicroStrain 3DM-GX3-15 IMU and a MaxBotix MB1242 ultrasonic altimeter.

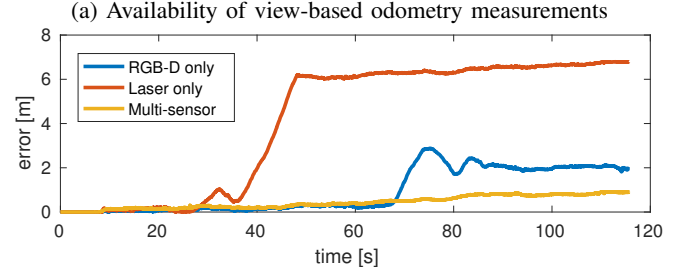
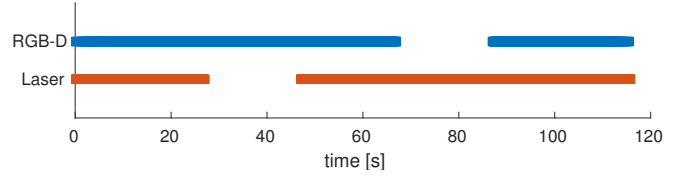


Fig. 5: Flight test results for the motion capture environment

TABLE I: RMS global position estimation error in motion capture environment. Error is presented as both an absolute measurement and normalized as a percentage of the total path length.

Estimator	RMS Error (m)	RMS Error (%)
RGB-D only	1.34	5.12%
Laser only	5.02	25.9%
Multi-sensor	0.517	1.97%

Three estimators were run on the data from each test flight for comparison. Two of the estimators were the single-sensor, relative MEKF of [11] using either the RGB-D odometry or the laser scan matching odometry. The third was the multi-sensor extension of [11] presented in this paper, using both sources of odometry. The visual odometry and scan matching algorithms ran real-time on the onboard computer, a quad-core Intel i7 processor at 2.1GHz. The estimators were run offline on recorded data for ease of development, but would easily also run in real-time on the onboard computer.

A. Motion Capture Environment

As the vehicle flew in the motion capture volume, each of the odometry measurements was turned off at different times to simulate failure. The availability of these measurements for one test flight is shown in Figure 5a, and the error in the estimated global position for the single-sensor and multi-sensor approaches is shown in Figure 5b. Table I summarizes the root mean square (RMS) error over the entire flight, presented both as an absolute value and normalized as a percentage of the total path length.

The estimation error was much lower for the multi-sensor approach, as the accuracy of the single-sensor approaches degraded significantly during the periods when measurements were unavailable. The magnitude of the error for the single-sensor approaches was highly dependent on the specific time that the measurements were turned off. While the error using only RGB-D was relatively low for the example summarized in Table I, for the same path errors above 25% were obtained

by turning the off the measurements for the same duration but at a different time. The performance of the multi-sensor approach, on the other hand, remained consistent as long as both odometry sources were not turned off simultaneously.

B. Challenging Indoor Environment

The indoor test environment was chosen because it is a good example of flight conditions that are of practical interest, but in which single-sensor approaches fail. The layout of the environment, and the vehicle’s path through it, are shown with the estimation results in Figure 6.

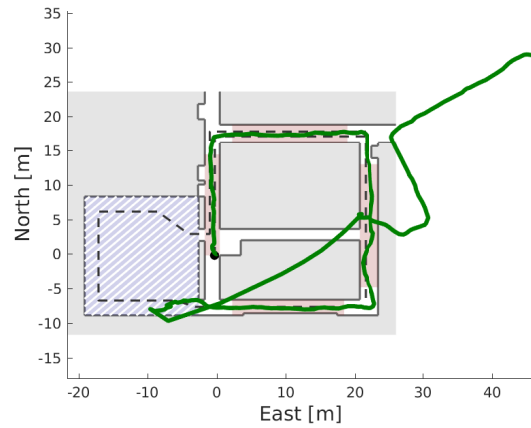
Two aspects of the environment were challenging for the single-sensor approaches, and are highlighted in Figure 6. The first of these was a dark room, shaded in blue, in the southwest corner of environment. Visual features could not be extracted from images due to the lack of illumination, which caused the visual odometry to fail in this region. The other challenging feature was the long hallways, shaded in red. The laser scan matching produced inaccurate measurements in these environments because there was not enough information to determine how far down the hallway the vehicle had moved.

For the multi-sensor approach, the visual odometry and laser scan matching were manually gated in the indicated failure regions. Eliminating the need for this manual gating is a key objective for future research, as further discussed in Section VI. The measurements were not gated for the single-sensor approaches because gating resulted in poorer estimation results than those presented here.

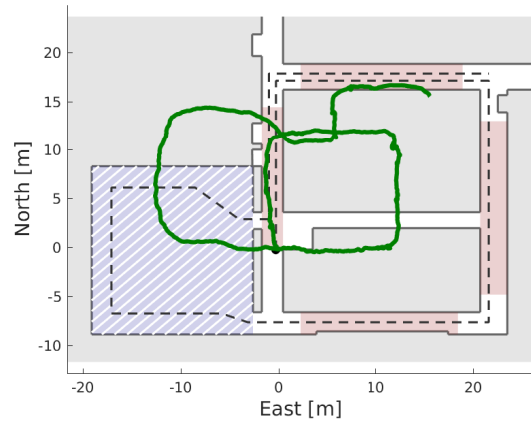
Figures 6a and 6b show that the single-sensor approaches accumulated large estimation errors in the regions in which their respective odometry measurements degraded. This resulted in significant error in the estimated global path. The results in Figure 6c for the multi-sensor approach, on the other hand, show that the estimated path is very close to the true path of the vehicle. The error for this approach is consistent with the global drift expected for an estimation scheme using only relative measurement updates, and can be reduced further using the map optimization techniques discussed in Section II. These results demonstrate that the proposed approach is able to produce accurate and useful estimates in environments that cause single-sensor approaches to fail.

VI. DISCUSSION AND CONCLUSIONS

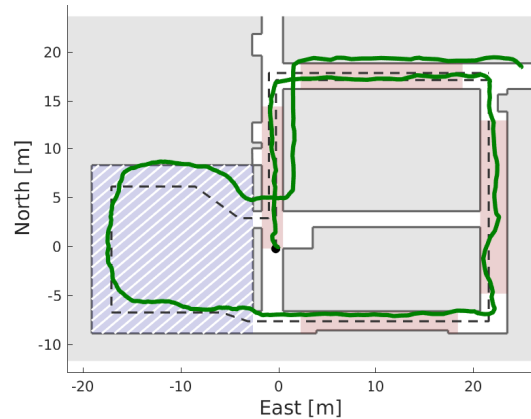
This paper presented a relative estimation framework that incorporates measurement updates from any number of view-based odometry algorithms running on a collection of exteroceptive sensors or sensing modalities. The framework is general and can work with any algorithm that provides odometry-like relative updates, and is not tied to specific sensors or odometry algorithms. The odometry algorithms can also operate independently of each other without the need for synchronizing keyframes or update rates. Incorporating measurements from multiple sensors in this manner allows the estimator to continue producing accurate estimates as the vehicle traverses challenging environments that cause



(a) Single-sensor approach with RGB-D visual odometry



(b) Single-sensor approach with laser scan matching



(c) Multi-sensor approach with RGB-D visual odometry and laser scan matching

Fig. 6: Estimation results for an indoor flight with an RGB-D camera and laser scanner. The dashed line shows the approximate true path of the vehicle overlaid on the building floor plan. The thick green line is the estimated path. The black circle represents the starting position. The shaded red areas indicate long hallways where the laser scan matching performed poorly and was gated for the multi-sensor approach. The blue hatched area indicates a dark room where the RGB-D odometry failed and was gated.

intermittent failure in the various sensing modalities. Experimental results demonstrate the validity and value of this approach.

The framework outlined in this paper currently produces good results when the odometry algorithms either provide valid measurement updates, or provide no updates at all when they are in a failure state. For the results in this paper, this was achieved with manual gating of the updates. Future work will investigate methods for automatically detecting and rejecting erroneous measurement updates. Additional future work will investigate incorporating global measurements such as GPS into this framework when they are available.

REFERENCES

- [1] R. C. Leishman, D. P. Koch, T. W. McLain, and R. W. Beard, "Robust motion estimation using an RGB-D camera," in *AIAA Infotech@Aerospace Conference*, Boston, 2013.
- [2] O. Aboutalib, B. Awalt, A. Fung, B. Thai, J. Leibs, T. J. Klausutis, R. Wehling, and M. Iames, "All source adaptive fusion for aided navigation in non-GPS environment," *Proceedings of SPIE*, vol. 6575, no. Visual Information Processing XVI, p. 657 509, Apr. 2007.
- [3] S. D. Bopardikar, S. Zhang, and A. Speranzon, "A robust and adaptive framework for localization under varying sensor modalities," in *AIAA Guidance, Navigation, and Control (GNC) Conference*, Boston, Aug. 2013.
- [4] A. Chambers, S. Scherer, L. Yoder, S. Jain, S. Nuske, and S. Singh, "Robust multi-sensor fusion for micro aerial vehicle navigation in GPS-degraded/denied environments," in *American Control Conference (ACC)*, 2014, pp. 1892–1899.
- [5] S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, "Multi-sensor fusion for robust autonomous flight in indoor and outdoor environments with a rotorcraft MAV," in *2014 IEEE International Conference on Robotics & Automation (ICRA)*, 2014, pp. 4974–4981.
- [6] D. Magree and E. N. Johnson, "Combined laser and vision-aided inertial navigation for an indoor unmanned aerial vehicle," in *American Control Conference (ACC)*, 2014, pp. 1900–1905.
- [7] R. C. Leishman, T. W. McLain, and R. W. Beard, "Relative navigation approach for vision-based aerial GPS-denied navigation," *Journal of Intelligent and Robotic Systems*, vol. 74, no. 1, pp. 79–111, 2014.
- [8] D. G. Kottas, J. A. Hesch, S. L. Bowman, and S. I. Roumeliotis, "On the consistency of vision-aided inertial navigation," *Experimental Robotics*, pp. 1–15, 2013.
- [9] A. G. Bachrach, "Autonomous flight in unstructured and unknown indoor environments," Master of Science, Massachusetts Institute of Technology, 2009.
- [10] S. I. Roumeliotis and J. Burdick, "Stochastic cloning: A generalized framework for processing relative state measurements," in *Proceedings 2002 IEEE International Conference on Robotics and Automation*, vol. 2, May 2002, pp. 1788–1795.
- [11] R. C. Leishman and T. W. McLain, "Multiplicative extended Kalman filter for relative rotorcraft navigation," *Journal of Aerospace Information Systems*, vol. 12, no. 12, pp. 728–744, 2014.
- [12] R. C. Leishman, J. C. Macdonald, R. W. Beard, and T. W. McLain, "Quadrotors and accelerometers: State estimation with an improved dynamic model," *Control Systems, IEEE*, vol. 34, no. 1, pp. 28–41, 2014.
- [13] J. B. Kuipers, *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace, and Virtual Reality*. Princeton: Princeton University Press, 1999.
- [14] N. Trawny and S. I. Roumeliotis, "Indirect Kalman filter for 3D attitude estimation: A tutorial for quaternion algebra," University of Minnesota, Department of Computer Science and Engineering, Minneapolis, Tech. Rep. 2005-002, Rev. 57, 2005. [Online]. Available: http://www-users.cs.umn.edu/~trawny/Publications/Quaternions_3D.pdf.
- [15] J. Zhang, M. Kaess, and S. Singh, "Real-time depth enhanced monocular odometry," in *IEEE International Conference on Intelligent Robots and Systems*, IEEE, Sep. 2014, pp. 4973–4980.
- [16] A. Censi, "An ICP variant using a point-to-line metric," in *IEEE International Conference on Robotics and Automation*, IEEE, May 2008, pp. 19–25.