MONOCULAR VISION-AIDED INERTIAL NAVIGATION FOR UNMANNED AERIAL VEHICLES

A Thesis
Presented to
The Academic Faculty

by

Daniel Paul Magree

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Aerospace Engineering

Georgia Institute of Technology
August 2015

MONOCULAR VISION-AIDED INERTIAL NAVIGATION FOR UNMANNED AERIAL VEHICLES

Approved by:

Professor Eric N. Johnson, Advisor
School of Aerospace Engineering
Georgia Institute of Technology

Professor Mark Costello
School of Aerospace Engineering
Georgia Institute of Technology

Professor Frank Dellaert
School of Interactive Computing
Georgia Institute of Technology

Professor J.V.R. Prasad
School of Aerospace Engineering
Georgia Institute of Technology

Professor Panagiotis Tsiotras
School of Aerospace Engineering
Georgia Institute of Technology

Date Approved: 15 May 2015

In memory of my dad,

Paul Magree

ACKNOWLEDGEMENTS

I owe a great debt to the many people who supported me throughout my studies at Georgia Tech. First of all, I'd like to thank my advisor, Dr. Eric Johnson, for his guidance and support, and for the many hours of fruitful discussion. I'd like to thank my committee, Dr. Mark Costello, Dr. Frank Dellaert, Dr. J.V.R. Prasad, and Dr. Panagiotis Tsiotras, for their helpful comments and conversations.

I'd like to thank the many mambers of the UAV Research Facility, both past and present, who I've had the honor to work with over these years. A special thanks for the long hours spent preparing and testing vehicles in the Georgia Tech Aerial Robotics club, with Dmitry Bershadsky, Stephen Haviland, Takuma Nakamora, Eohan George, Tim Dyer, Hiroyuki Hashimoto, Ep Pravitra, and many others. I would not be here without those ahead of me showing me the ropes, giving me mock qualifying exams, proofreading papers – Nimrod Roos, Girish Chowdhary, Tansel Yucelen, Jack Mooney, Claus Christmann, and research engineers Jeong Hur and Henrik Christopherson. And to the many others who I've worked beside in our lab, Gerardo de la Torre, Lee Whitcher, Toshinobu Watanabe, Yongeun Yoon, Kyuman Lee, Fritz Langford, and Gerald van Dalen. You made it a joy to come to work each day. I had many great colleagues on funded projects, especially Kevin Betts and Jon Wetherbee at Leidos, and Andy Shein and Brad Powers at Cyphy Works.

There were many people who helped in non-academic ways, giving encouragement in one form or another. Several families opened their homes to me, including the Saccaggi's, the Berzovini's, the Gowasacks, the Casadei's and the Keirns, giving me a place to recover from the stress of studies. I was fortunate to have great roommates throughout my time in Atlanta, Pietro, Giovanni, Pierre, Alessandro, Emanuele,

Francesco, Samuele. You guys are the best. And I am very grateful for Emma Gowasack, who with patience, joy, and good humor bravely dated a guy in the final months of his doctorate.

Finally, I'd like to thank Mom, Michael, Theresa, Beth, Timothy and Mary, and my grandparents. Your love and support I carry with me every day.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS OR ABBREVIATIONS

BTEKF Bierman-Thornton Extended Kalman Filter.

EKF Extended Kalman Filter.

GCS Ground Control Station.

GPS Global Positioning System.

GUST Georgia Tech UAV Simulation Tool.

IMU Inertial Measurement Unit.

INS Inertial Navigation System.

SLAM Simultaneous Localization and Mapping.

UAV Unmanned Aerial Vehicle.

V-INS Vision-aided Inertial Navigation System.

SUMMARY

The reliance of unmanned aerial vehicles (UAVs) on GPS and other external navigation aids has become a limiting factor for many missions. UAVs are now physically able to fly in many enclosed or obstructed environments, due to the shrinking size and weight of electronics and other systems. These environments, such as urban canyons or enclosed areas, often degrade or deny external signals. Furthermore, many of the most valuable potential missions for UAVs are in hostile or disaster areas, where navigation infrastructure could be damaged, denied, or actively used against the vehicle. It is clear that developing alternative, independent, navigation techniques will increase the operating envelope of UAVs and make them more useful.

This thesis presents work in the development of reliable monocular vision-aided inertial navigation for UAVs. The work focuses on developing a stable and accurate navigation solution in a variety of realistic conditions. First, a vision-aided inertial navigation algorithm is developed which assumes uncorrelated feature and vehicle states. Flight test results on a 80 kg UAV are presented, which demonstrate that it is possible to bound the horizontal drift with vision aiding. Additionally, a novel implementation method is developed for integration with a variety of navigation systems. Finally, a vision-aided navigation algorithm is derived within a Bierman-Thornton factored extended Kalman Filter (BTEKF) framework, using fully correlated vehicle and feature states. This algorithm shows improved consistency and accuracy by 2 to 3 orders of magnitude over the previous implementation, both in simulation and flight testing. Flight test results of the BTEKF on large (80 kg) and small (600 g) vehicles show accurate navigation over numerous tests.

CHAPTER I

INTRODUCTION

This thesis presents several monocular vision-aided navigation solutions for use on unmanned aerial vehicles (UAVs), and evaluates their performance with simulation and flight testing.

The vision-aided inertial navigation systems (V-INS) presented here operates by estimating the state of the vehicle and the location of features in the world, a paradigm known as simultaneous localization and mapping (SLAM). A stream of images from a monocular camera is used to gain information on feature locations and update the state. The state updates are combined with data from an inertial sensor and any additional sensors to provide a full navigation solution for the UAV. The navigation architecture has the benefit of not requiring external infrastructure such as predefined features or GPS satellites to bound the drift of the inertial navigation solution.

The navigation systems are intended for use on UAVs, which often have fast, nonlinear dynamics and use software controllers to maintain stability and follow a trajectory. It cannot be taken for granted that a navigation algorithm designed independently from the vehicle dynamics controller will perform adequately when inserted into a control loop with a nonlinear plant and/or nonlinear controller, due to the lack of applicability of the separation principle to nonlinear systems. Therefore, the algorithms presented here are simulated and flight tested on representative vehicles with controllers in the loop to verify robustness to nonlinearities.

In contrast with other vision-aided inertial navigation approaches for UAVs, the algorithms developed in this research do not require generation of an initial map and extension of the map based on spatially separated keyframes. Map and state updates

happen on every frame, allowing what information is available to be incorporated immediately and at high frame rates. Also, the Bierman-Thornton EKF implementation is believed to be one of the first EKF SLAM implementations that explicitly takes numerical stability considerations into account.

Specific contributions made in this thesis are listed below:

- Development of a practical EKF-based V-INS which ignores feature-vehicle correlations.
- Development of a theorem allowing the integration of Kalman filter updates with limited shared information.
- Application of the theorem to a novel integration method of the uncorrelated vision-aiding module and a dynamically configurable INS solution.
- Development of a V-INS with fully correlated features for UAVs.
- Development of a novel implementation of a fully correlated V-INS in a factored EKF formulation, with efficient feature marginalization and initialization.
- Monte Carlo analysis of the numerical stability of the proposed algorithm, showing 2 to 3 orders of magnitude improvement over standard EKF and Joseph-form EKF.
- Demonstration of this numerical stability improvement on flight code, along with a comparison of computation time with the standard EKF.
- Simulation results for all systems, illustrating the performance in realistic controller-in-the-loop scenarios.
- Flight test results showing the performance of the navigation algorithms with controller in the loop on multiple vehicles, indoors and outdoors.

These contributions have advanced the understanding of the performance and capabilities of visual EKF-SLAM as applied to UAV navigation.

1.1 Motivation

Unmanned aerial vehicles have existed since the beginning of heavier-than-air flight. Their use has increased greatly in recent years due to the miniaturization of avionics equipment. The availability of low cost sensors, computers and communication devices has enabled the expanded operation of UAVs by allowing greater operational control of their behavior and improved data acquisition. As the size and weight of aerial vehicles and their payloads continue to shrink, vehicles are physically able to operate in new and challenging environments. The guidance, navigation and control challenges posed by these new environments, and the opportunities offered by the development of technology, make this an important area of research.

The specific challenges posed by UAVs to navigation comes in several forms. The fact that the vehicle is unmanned allows a vehicle to operate in more dangerous and less developed areas than a manned vehicle. Navigational infrastruture such as VOR systems or GPS satellites might not be available for use in hostile or disaster areas. With the reduction of vehicle size, operation in indoor and other enclosed environments has become possible, and in such areas the use of GPS is limited or non-existent. The reliance on an external infrastructure for vehicle navigation is a limiting factor.

On the other hand, the abundance of sensor data and our ability to process it has increase exponentially. The availability of low-cost ranging devices, cameras, pressure sensors and inertial senors can provide a wealth of data for generating a navigation solution. These devices do not rely on external infrastructure, and in the case of inertial sensors are entirely interoceptive.

These considerations have led to research in navigation systems that take advantage of available sensor data and reduce reliance on GPS. In particular, navigation based on visual sensing is of interest due to low sensor cost, and the fact that many UAV applications require a camera payload already. Furthermore, cameras provide a wealth of data about the environment. Use of a monocular camera, rather than stereo or another configuration, is attractive since it is a more common sensor configuration than stereo on UAVs. The development of monocular vision-aided inertial navigation for UAVs will expand the environment in which UAVs can operate, and increase their usefulness.

1.2 Related Work

The literature pertaining to monocular V-INS for UAVs is quite large. Monocular visual SLAM has been an active area of research in the computer science and robotics communities for many years, and is only recently beginning to be applied to UAVs. It developed with different constraints and priorities than those specific to the aerospace community. UAV navigation has developed somewhat separately, and is now beginning to take advantage of lightweight sensors and computing power. The relevant literature from both areas is considered.

1.2.1 Visual Simultaneous Localization and Mapping

Simultaneous localization and mapping has been a very active research area over the past 20 years. A large amount of valuable work has been done, using a variety of sensors and estimations methods. Specifically in the area of monocular visual SLAM, much progress has been made since an effective camera-only system was first presented by Davison et al. in [9]. This method as well as subsequent extensions [10, 6, 7] is based on an extended Kalman filter for estimating the camera pose as well as point features. A one-point RANSAC algorithm is used to improve feature correspondence, and inverse depth parameterization for feature points allows fast depth estimation.

Alternatively, Eade and Drummond [12] developed a particle filter based approach to the problem of monocular SLAM. One of the most successful paradigms has been developed by Klein and Murray [27], called parallel tracking and mapping (PTAM). This method separates the tracking and mapping functions of the SLAM algorithm into separate threads, so that fast feature tracking and camera localization can be done at frame rate, and map building can happen when necessary at a slower rate. This division allows a full bundle adjustment solution of the mapping problem, usually too computationally intensive a task for real-time applications. Recently, attempts have been made to characterize the relative merits of bundle adjustment and filter-based methods [44].

The combination of inertial measurements and vision information has been studied also. Corke et al [8] identify the complimentarity of the sensing modalities and highlight the fact that the combination requires no external infrastructure. Many approaches for fusing this information have been researched, including non-SLAM approaches such as optical flow, [22], losely coupled approaches[51, 41], and tightly coupled approaches [34, 24]. The fusion of inertial and visual data is especially applicable to UAVs, due to need to estimate full 6 DOF motion of the vehicle, and has been successfully applied to commerical products such as the Parrot AR.Drone[1] This literature review will focus on V-INS solutions that perform mapping and localization, with an emphasis on systems applicable to UAVs.

1.2.2 Estimator Consistency

The concept of estimator consistency can be defined as the correctness of the uncertainty given by the estimator. An estimate is consistent if it accurately accounts for the uncertainty in the estimate. Much recent work has gone into characterizing and

[1] `http://ardrone2.parrot.com/`

improving the consistency of navigation algorithms. In Castellanos [2], the importance of feature correlations to the vision-aided navigation consistency was noted. It was found through experimental results that while ignoring feature correlations did not dramatically affect the resulting estimate, the covariance of the estimate quickly became overconfident due to ignoring feature-vehicle correlation. Martinelli analyzed the observability of the V-INS system in [35] and characterized the observable subspace for a variety of configurations. It was shown that in the presence of gravity there is always an unobservable subspace about the gravitational axis. In Huang et al. [23] and Hesch et al. [20], it was noted that a significant source of estimator inconsistency is linearization in the estimator algorithm, which causes spurious information gain. It was shown that calculating and maintaining the unobservable subspace through optimally adjusting the measurement Jacobian and state transition matrix improved consistency of the estimator. In [20] and [21] this was analyzed specifically in the context of visual SLAM and visual odometry, and experiments were conducted on pre-recorded datasets.

1.2.3 Numerical Stability

In the context of V-INS, numerical conditioning can become a significant factor because the problem is inherently not completely observable [35], and the covariance will increase monotonically in certain subspaces. This can lead to a poorly conditioned covariance matrix over time. Variants of the EKF have been proposed to improve the numerical conditioning over the standard EKF. The UD filter used in this paper was first proposed by Bierman and Thornton [45] [17]. In a detailed case study of a portion of an interplanetary space mission, they compare the UD filter to standard EKF, the Joseph-stabilized EKF, and the square-root filter [1]. They demonstrate that the UD filter performs accurately using single precision arithmatic where both the standard EKF and Joseph-stabilized filters fail. Additionally, the

UD filter demonstrated computational load which was approximately 1.5 times the standard and Joseph-stabilized, and much less than the square-root filter. The formulation of the UD filter presented here extends the original algorithm to allow feature states to be efficiently added and removed from the filter.

The problem of numerical stability only rarely been considered in EKF-SLAM formulations, but was recognized early in large-scale and full solution SLAM formulations. For example, factor graph approaches uses a square root formulation and provably numerically stable algorithms due to the potential for numerical difficulties in the large optimization problem[11]. The formulation of the UD filter presented here, first presented in Magree et al. [34], uses an extension of the original algorithm to allow feature states to be efficiently added and removed from the filter. In parallel with that work, Schmid et al. [39] applied the same technique to adding and removing reference states, allowing a change in the reference frame of the SLAM problem. Another approach is to track the uncertainty in absolute position externally to the filter. In [30], a relative navigation approach is taken, where estimates are tracked with respect to a local reference frame. The optimization then remains local and the resulting collection of reference frames may then be put through a global optimizer. Uncertainty in position is transfered to the local reference, and the estimator need only capture the relative uncertainty with respect to the keyframe.

1.2.4 Vision-aided Navigation for Aerial Vehicles

Application of visual navigation techniques to UAVs is relatively new compared to ground vehicles. Until recently there has been a lack of published work characterizing the performance of V-INS systems on a UAV with a controller in the loop, with the notable exceptions listed in this literature review. The author speculates that this is due to the difficulty of performing flight tests, and perhaps an misplaced reliance on the separation principle, which states that for linear systems a state estimator and

controller may be designed independently. However, there is no corresponding principle for nonlinear systems, and the author argues that any comprehensive evaluation of a V-INS system for such a configuration must be tested. The flight test results presented here aim to build on previous discussions of this topic such as [33].

The following section emphasizes implementations of vision-aided navigation systems on UAVs, which have been flight tested with controller in the loop.

1.2.4.1 Monocular Vision

Work on the application of V-INS to aerospace systems is becoming more common. This work is driven in part by the need for independence or reduced reliance on GPS for unmanned aerial vehicles. Vision aiding for obstacle avoidance [50], and for vehicle localization [55] were early efforts in this area. Langelaan et al. used an unscented Kalman filter for simultaneously estimating vehicle states as well as feature locations in inertial space [28]. A Mahalonobis norm was used as a statistical correspondence for data association from frame-to-frame for each estimated feature. New feature locations were initialized using a projection onto the ground plane. Simulation results for a UAV navigating through a 2D environment were presented, however no flight test results were reported. Similarly, Veth [49] developed an EKF-based visual SLAM system, and demonstrated it on recorded data and ground vehicles.

In addition to EKF-SLAM formulations, numerous parallel localization and mapping implementations have been developed for UAVs. Weiss et al.[51] describe results from the development of a monocular visual navigation system over several years. The system is based on a modified version of the parallel tracking and mapping (PTAM) algorithm of Klein and Murray [27]. The integration of PTAM on an unmanned vehicle system required improvement of overall system robustness, including robustness to repeated patterns in images, and limiting keyframes for constant time processing. The 6 degree of freedom (DOF) state estimate is incorporated as a black box pose

measurement in an EKF. Forster et al.[15] also use a parallel localizing and mapping framework similar to PTAM, but uses image intensity patches rather than features and descriptors in the tracking thread. In Faessler et al.[14], their method was combined with dense mapping system which computed dense surface reconstruction at a ground station in real time. The system was validated with over 100 flights. Mostegel et al.[37] describe a system for evaluating the quality of the map and current view for localization purposes. They use the results to plan a trajectory which finds the best next pose in a 4-dimensional search space (3 position, 1 attitude). While these approaches show great potential, there can be large delays during the addition of new maps. Localization happens at high frame rate, but mapping is slow, and can be a limitation on speed of exploration. The EKF-based systems, on the other hand, calculate new state estimates for map and vehicle at frame rate, at the cost of severely limiting the number of features.

A similar approach to the keyframe-based methods are those that compute the full SLAM solution. In Indelman et al. [24] the full SLAM solution is computed by taking advantage of the graph structure of the problem through the use of factor graphs. The method is validated on prerecorded datasets. While not a constant time algorithm, the method uses the graph structure to efficiently compute solutions in real time.

1.2.4.2 Stereo and Hybrid Approaches

Two notable stereo approaches are presented in [40] and [19]. Schmid et al.[40] describe a complete UAV system based on a quadrotor platform. The vehicle sensors are an inertial measurement unit (IMU) and a stereo camera module, in combination these allow a full 6 DOF navigation solution. The navigation algorithm is EKF based, and extends the state vector with delayed states to account for measurement latency from the stereo camera module. The stereo camera module provides odometry

measurements to the EKF and depth images to update the map of voxels in open loop. The use of stereo cameras for processing vision data removes the ambiguity in depth of monocular cameras, but at the expense of a much higher processing burden and measurement latency. Heng et al. takes a similar approach [19], aligning new stereo frames to previous keyframes, though all processing was done on the onboard CPU to reduce latency. The navigation systems presented here processes at a much higher frame rate, but requires a depth prior, typically acheived by using a downward facing camera and an altitude measurement.

Some approaches try to balance the benefits of visual odometry (low computation) with full slam (accuracy). Shen et al.[41] develop a system for high speed flight of a quadrotor which makes several simplifications to improve frame rate on an embedded computer. The navigation system is fundamentally monocular, but uses low frequency stereo measurements for mapping and estimating scale. Mapping and localization are separated, and mapping assumes known camera location, therefore the full slam problem is not attempted. However, the navigation algorithm is still robust enough to allow aggressive accelerations with forward-facing cameras and IMU as the only sensors.

Another vision-aided system that avoids the full SLAM problem is given in [38] and [3]. Stereo frames are used to estimate average velocity between pairs. Rather than update velocity directly, the avearage velocity is used to calculate a relative motion measurement (change in position and attitude). The expected relative motion is calculated using delayed pose and pose covariance stored during last image frame.

1.2.4.3 Absolute Localization

While the goal of SLAM formulations is to provide highly accurate navigation solutions, at the end of the day it is a relative estimation paradigm, giving no absolute

position information. Incorporating additional *a priori* information into the navigation solution can resolve the ambiguity. In Chiu et al. [4], georeferenced features are matched to features extracted from the current frame, and used to update a smoothing information filter. In van Dalen et al. [48], a system for matching aerial images to satellite maps is presented, which uses normalized cross-correlation within a particle filter of horizontal vehicle position samples. A resulting location estimate and covariance is fed into the EKF estimator performing visual SLAM. For such systems, it is important that the underlying SLAM system maintain a consistent horizontal position covariance.

1.3 Context of This Work

This thesis builds on previous research developing vision-aided inertial navigation systems. Wu et al. developed a method for fusing feature information from a monocular vision sensor in an extended Kalman filter framework [53, 52, 54]. The approach in those papers relied on tracking features whose locations were estimated when GPS was active. When GPS was inactive, they demonstrated that this method ensures bounded hover of a rotorcraft UAV through flight test results. This work was continued in Chowdhary et al. in which a fully independent vision-aided inertial navigation system was presented and flight tested.[5]. The systems presented here improve upon Chowdhary et al. in numerous ways, in particular by using improved feature correspondence in all systems, and in the factored EKF by accounting for correlations between the vehicle and feature states.

1.4 A Guide to This Document

The rest of this document is ordered as follows: In Chapter 2, background material for the understanding the key algorithms of the thesis is presented. The vehicle and measurement models used throughout this document are described, and an brief

review of the standard and Joseph-form EKF is given. Next, in Chapter 3 a vision-aided inertial navigation system is presented which is based on a sequential Kalman filter formulation and ignores feature-vehicle correlations. Simulation and flight test results are shown to evaluate system performance. Chapter 4 presents a theorem which defines requirements for a modular filter structure, and then applies this theorem to a dynamically configurable filter with a vision-aiding module. Simulation results characterize the behavior of the filter. Chapter 5 describes a new formulation of the EKF-SLAM problem in terms of a Bierman-Thornton factored Kalman filter (BTEKF). A Monte Carlo simulation demonstrates the benefits of the implementation, and simulation and flight test results are presented on two vehicles. Finally, Chapter 6 concludes the thesis and identifies future work.

CHAPTER II

PRELIMINARIES

The following section lays the groundwork for the sections that follow. In it are presented an overview of the extended Kalman filter (EKF). The EKF forms the basis of all the estimator techniques that are used in this thesis. Next, the reference frames used throughout the work are defined. Finally, the vehicle state vector, non-linear propagation equations and the camera model are given. Finally two alternative feature parameterizations are described.

2.1 The Extended Kalman Filter

The following gives a brief summary of the standard and Joseph-form EKF methods. For a more detailed treatment, see Gelb [16] or Grewal and Andrews [17].

Let the model be defined as a hybrid continuous plant with discrete updates:

$$\dot{x}(t) = f(x(t)) + B(u(t) + \eta(t)), \quad x(0) = x_0 \tag{1}$$

$$z_k = h(x(t_k)) + \nu_k \tag{2}$$

The plant model f and measurement model h are assumed to be infinitely differentiable. The stochastic plant noise $\eta(t)$ is drawn from a Gaussian distribution such the expectation $\mathrm{E}(\eta(t)) = 0$ and $\mathrm{E}(\eta(t)\eta(s)^{\mathrm{T}}) = \delta(t-s)Q$. Similarly, the discrete stochastic variable ν_k is drawn from a Gaussian distribution such that $\mathrm{E}(\nu_k) = 0$ and $\mathrm{E}(\nu_k\nu_j^{\mathrm{T}}) = \Delta(k-j)R$. The functions $\delta(t)$ and $\Delta(k)$ correspond to the Dirac and Kroniker delta functions, respectively, and B is the input matrix.

The standard EKF makes the assumptions that in the vicinity of the current estimate and measurement, plant and measurement functions behave linearly. The function can be linearized about the current estimate and the Kalman update and

propagation equations can be applied. Furthermore, to facilitate implementation on a digital computer, the continuous plant derivative is evaluated at discrete time intervals and numerically integrated. The propagation of the state estimate at sample time t_k is given by

$$\dot{\hat{x}}(t_k) = f(\hat{x}(t_k)) + Bu(t_k) \tag{3}$$

$$\hat{x}(t_{k+1}) = \text{Integration}(\dot{\hat{x}}(t_k), \Delta t_k) \tag{4}$$

A variety of methods are available to perform the integration. The covariance of the estimate, $P = \mathrm{E}(x - \hat{x})(x - \hat{x})^{\mathrm{T}}$ must also be propagated. Let $P_k = P(t_k)$. Then P_{k+1} is given by the following discrete propagation equation.

$$P_{k+1} = \Phi P_k \Phi^{\mathrm{T}} + Q\Delta t_k \tag{5}$$

The state propagation equation can be generated by from the linearized plant model

$$A = \frac{\partial f(x)}{\partial x}_{x=\hat{x}(t_k)} \tag{6}$$

$$\Phi = I + A\Delta t_k \tag{7}$$

Matrix exponential may also be used for computing the state transition matrix, but Euler integration is often used for computational efficiency and ease of implementation. The discrete measurement updates are applied by generating the Kalman gain from the linearize measurement equation;

$$C = \frac{\partial h(x)}{\partial x}_{x=\hat{x}(t_k)} \tag{8}$$

The discrete Kalman update is generated in the following way. Letting $\hat{x}_k = \hat{x}(t_k)$

$$K = P_k C^{\mathrm{T}} \left(C P_k(-) C^{\mathrm{T}} + R \right)^{-1} \tag{9}$$

$$\hat{x}_k(+) = \hat{x}_k(-) + K(z - h(\hat{x}_k(-))) \tag{10}$$

$$P_k(+) = P_k(-) - KCP_k(-) \tag{11}$$

It was known very early that a naive implementation of the Kalman filter equations was numerically unstable. An early attempt to improve numerical behavior was the so-called Joseph-form of the covariance update equation. This implementation addressed the problem of off-optimal gain values causing assymetries in P[16]. Joseph-form covariance update is given by

$$P_k(+) = (I - KC)P_k(-)(I - KC)^{\mathrm{T}} + KRK^{\mathrm{T}} \tag{12}$$

Both the standard and Joseph-form EKF implementations are commonly found in the filtering literature and in visual EKF-SLAM in particular.

2.2 Reference Frames

A local inertial reference frame is defined, with axes in the North-East-Down configuration. In addition to the local inertial frame, the vehicle and feature states are described in two other reference frames, the vehicle body frame and the camera frame. The body frame is attached to the vehicle, with its origin at the vehicle center of gravity. The orientation of the body frame is chosen to have the x axis point out the front of the vehicle and the z axis point down. The camera frame is attached to the camera and located at the optical center. The x axis lies along the optical axis, and the z axes points down along the image plane. For clarity of presentation, it is assumed that the camera frame and body frame have the same origin and differ only in orientation. In practice, a static body frame offset vector is applied to account for the camera not being at the origin. The local inertial frame is denoted with a i, body frame with b, and camera frame with c. Figure 1 illustrates the coordinate frames.

2.3 Vehicle Model

The vehicle model is based on the specific force and angular velocity input from an IMU. The non-linear dynamics of the vehicle are driven by raw IMU input, which is

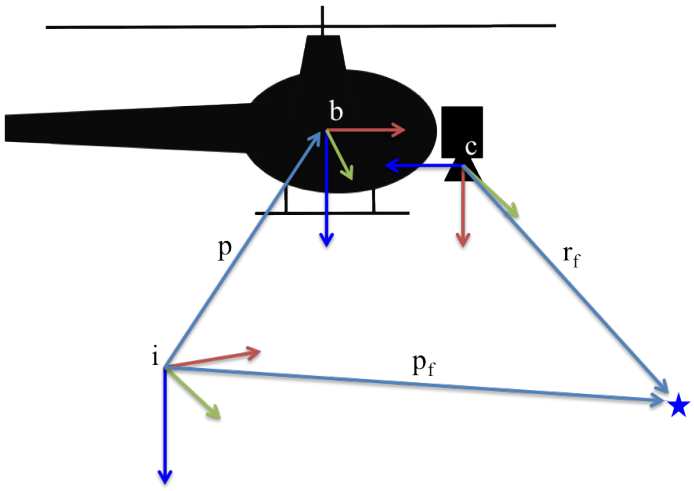


Figure 1: A schematic of the key reference frames used in this presentation: the inerial frame i, the vehicle body frame b, and the camera frame c. In general, the camera and body frame origins can be in different locations, but for simplicity of presentation it is assumed they are co-located.

assumed to have a static or slowly evolving bias and corrupted by white Gaussian noise.

The vehicle state is given by the following vector:

$$\hat{x}_a = \begin{bmatrix} \hat{p}^i & \hat{v}^i & \hat{q}^i & \hat{s}_b & \hat{\omega}_b \end{bmatrix}^{\mathrm{T}} \tag{13}$$

where p, v, q, is the vehicle position, velocity and attitude quaternion, respectively, s_b is the acceleration bias and ω_b is the gyro bias. Superscript i denotes the inertial frame and hatted variables indicate estimated quantities. The rotation matrix from body to inertial is denoted $L_{ib} = L_{bi}^{\mathrm{T}}$. The vehicle state is propagated by integrating data from the IMU. IMU sensor measurements are corrupted by noise and bias as follows:

$$s_{raw} = a + s_b + L_{bi}g + \eta_a, \tag{14}$$

$$\omega_{raw} = \omega_t + \omega_b + \eta_\omega. \tag{15}$$

where a, and ω_t are are the true acceleration and angular velocity, respectively, and g is the acceleration due to gravity in the inertial frame. It is assumed that the

noise is zero mean and white Gaussian, i.e. $\eta_a \sim \mathcal{N}(0, Q_a)$ and $\eta_\omega \sim \mathcal{N}(0, Q_\omega)$. The estimated bias is subtracted from the IMU data before propagation in the model

$$s = s_{raw} - \hat{s}_b, \tag{16}$$

$$\omega = \omega_{raw} - \hat{\omega}_b. \tag{17}$$

The vehicle dynamics are given by the following:

$$\dot{\hat{p}}^i = v^i \tag{18}$$

$$\dot{\hat{v}}^i = L_{ib}s - g = L_{ib}(s_{raw} - \hat{s}_b) - g \tag{19}$$

$$\dot{\hat{q}}^i = \frac{1}{2}\mathcal{Q}(\omega)\hat{q}^i = \frac{1}{2}\mathcal{Q}(\omega_{raw} - \hat{\omega}_b)\hat{q}^i \tag{20}$$

$$\dot{\hat{s}}_b = 0 \tag{21}$$

$$\dot{\hat{\omega}}_b = 0 \tag{22}$$

where s is bias-corrected specific force, and angular velocity ω is the bias-corrected angular velocity. The function $\mathcal{Q} : \mathbb{R}^3 \to \mathbb{R}^{4\times4}$ maps angular velocity to the quaternion derivative matrix coefficient and, in the first-element-scalar convention used here, is given by [43]

$$\mathcal{Q}([a_1 \quad a_2 \quad a_3]^{\mathrm{T}}) = \begin{bmatrix} 0 & -a_1 & -a_2 & -a_3 \\ a_1 & 0 & a_3 & -a_2 \\ a_2 & -a_3 & 0 & a_1 \\ a_3 & a_2 & -a_1 & 0 \end{bmatrix} \tag{23}$$

Using the quaternion representation in the estimation algorithm causes the covariance matrix to become singular and requires careful accounting of the quaternion constraints. To avoid these difficulties, a minimal representation of the vehicle's attitude is used, which defines the vehicle's current attitude with respect to an arbitrary reference frame, in this case the attitude in the previous time step. Since the time step is short, we can assume this attitude change is small and it can be defined as an

infinitesimal error quaternion.

$$\delta q = \begin{bmatrix} 1 & \hat{R} \end{bmatrix}^{\mathrm{T}} \tag{24}$$

such that

$$\delta q = \hat{q}_{ref}^{-1} \otimes \hat{q}. \tag{25}$$

Additional details on this formulation can be found in [52, 29]. The resulting minimal vehicle state vector is

$$\hat{x}_v = \begin{bmatrix} \hat{p}^i & \hat{v}^i & \hat{R} & \hat{s}_b & \hat{\omega}_b \end{bmatrix}^{\mathrm{T}} \tag{26}$$

2.4 Camera Model

The camera model describes the relationship between objects in the world and their image as recorded by the camera. A standard distortion-compensated pinhole camera model is used to describe this relationship, as described in [32]. A camera calibration is performed to determine intrinsic camera parameters of principle point, focal length and distortion. For simplicity, it will be assumed that the principle point is in the center of the image plane, and distortion has been compensated for, thus reducing the model to an ideal pinhole camera model. Additionally, the the image plane will be treated as if it is located in front of the optical center. Given the feature point location in the camera frame, $r_f^c = [X, Y, Z]^{\mathrm{T}}$, the image plane location (u, v) is given by

$$z = \begin{bmatrix} u \\ v \end{bmatrix} = h(x) + \nu = \begin{bmatrix} f_u \frac{Y}{X} + \nu_u \\ f_v \frac{Z}{X} + \nu_v \end{bmatrix} \tag{27}$$

where f_u and f_v are the horizontal and vertical focal lengths, respectively, and $\nu_u \sim \mathcal{N}(0, R_u)$ and $\nu_v \sim \mathcal{N}(0, R_v)$, $R = \mathrm{diag}(R_u, R_v)$. Figure 2 illustrates the camera model.

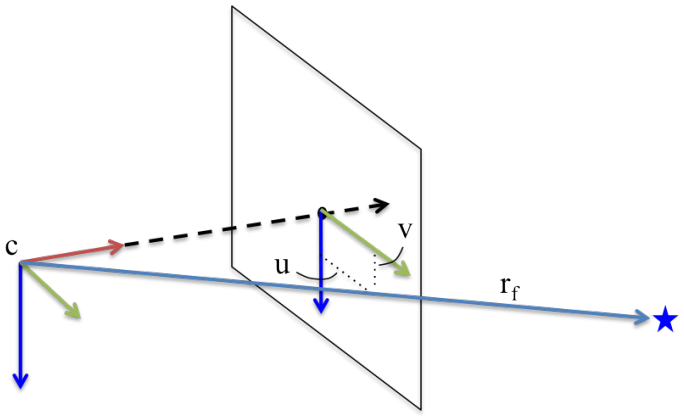


Figure 2: A schematic of the camera model.

2.5 Feature Parameterization

Numerous methods exist for parameterization of feature locations. Perhaps the most obvious one is a Cartesian parameterization, where the feature location is described by its Cartesian coordinates in a local inertial frame. The primary benefit of this method is its simplicity of implementation and clarity. However, alternate parameterizations have been proposed which capture the uncertainty of features better, and allow for infinite uncertainty in depth and points located at infinity. There are a variety of forms, but they all parameterize the location of feature points using inverse distance to the feature. These methods of parameterization are essential for implementations of camera-only visual SLAM that do not delay in initializing points, due to the infinite uncertainty in depth from a single observation [6] [42].

In the applications of the navigation systems presented here, there are less stringent requirements. It is assumed that some form of depth prior is available, such as a sonar or initial terrain height. With the additional information, the question of which parameterization to use must factor not only improved uncertainty characterization, but also greater computational cost of larger feature state vectors and/or propagation of feature states. Therefore both Cartesian parameterization and inverse depth parameterizations were implemented for use in the presented navigation algorithms.

The Cartesian parameterization is given as follows: Let x_{f_j} be the state of feature j. Then the state vector is made up of the coordinates of the feature in the local inertial frame,

$$x_{f_j} = p_{f_j} \in \mathbb{R}^3 \tag{28}$$

The feature location can then be transformed into the camera frame with the following conversion

$$r_f^c = L_{ci}(p_f - p_v) \tag{29}$$

which then can be used in the camera model presented in 2.4.

The inverse depth parameterization used here is taken from [6]. It is an anchored form, in which the location of the feature is described with respect to an anchor point, in this case the optical center of the camera at point initialization. The feature is described with a 6 dimensional vector consisting of the Cartesian coordinates of the anchor point and the modified polar coordinates to the location of the feature, all in the local inertial frame.

$$x_{f_j} = \begin{bmatrix} p_{a_1} & p_{a_2} & p_{a_3} & \theta & \psi & \rho \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^6 \tag{30}$$

The vector $p_a = [p_{a_1} \quad p_{a_2} \quad p_{a_3}]^{\mathrm{T}}$ defines the anchor point, the pair (θ, ψ) are the azimuth and elevation of the ray from the anchor point to the feature, and ρ is the inverse distance of the feature from the anchor point. The $(0, 0)$ axis for the polar coordinates may be chosen arbitrarily, and in this case is chosen to be the direction of the optical axis of the camera when the vehicle is on the ground.

To find the relative feature location from the state vector, the following conversion is made

$$h_f^c = L_{ci}\left[\rho(p_a - p_v) + m(\theta, \psi)\right] \tag{31}$$

where the function $m(\cdot, \cdot)$ maps azimuth and elevation to a unit vector direction. Note that h_f^c is the normalized ray towards the feature point, rather than the vector

between the camera and feature. However, this may still be used in the camera model to compute image location, because all points along the ray map to the same image location. The state vector can be transformed back into Cartesian coordinates (for display or conversion) with the following transformation:

$$p_{f_j} = p_a + \frac{1}{\rho} m(\theta, \psi) \tag{32}$$

The inverse depth parameterization allows features to exist at $\rho = 0$, which is a distance of infinity. Additionally, the uncertainty of ρ can be such that 0 is in a significant portion of the Gaussian distribution, allowing the uncertainty to capture an unbounded distance. The cost of this parameterization is the addition of three states for each feature, which, in a simple implementation, halves the number of features able to be processed for a given amount of computational power.

Other inverse depth parameterizations exist which do not require additional states. For example, modified polar coordinates may be used without the anchor state, and be referenced from the vehicle itself [42]. However, this parameterization has its own computational cost from the requirement of propagating the feature states.

CHAPTER III

STATE ESTIMATION IGNORING FEATURE CORRELATIONS

In this chapter, an estimation algorithm is presented which estimates the vehicle state and the locations of features, assuming no correlation among features and between features and vehicle states. Instead, decoupled covariance matrices are stored for each feature and the vehicle states. The primary advantage of this formulation is the reduction in computational requirements – for example, for Cartesian parameterization, operations are performed on a set of small covariance matrices of size $N_v \times N_v$ and 3×3, rather than one large matrix of size $(N_v + 3N_f) \times (N_v + 3N_f)$. This formulation might also be compared to Rao-Blackwellized particle filter approaches such as FastSLAM [36] [12], with only one particle. The cost of such an approximation is the loss of the relative uncertainty of vehicle and features. The filter no longer captures the fact that the relative position of features and vehicle is known accurately, but the global position is not. This causes the filter to become overconfident.

The visual SLAM navigation system uses an extended Kalman filter to estimate the state of the vehicle and the location of image features in the environment. Images are captured and features are extracted from the images. The features from the first image are used to initialize a database of feature states. When each subsequent image is captured, features are extracted and matched to the points stored in the database, the error between their predicted and measured location is used to update their location and the vehicle state via the extended Kalman filter gain and update equations. Points that are no longer visible due to vehicle motion are discarded and unmatched measurements are initialized into the database. Between image captures,

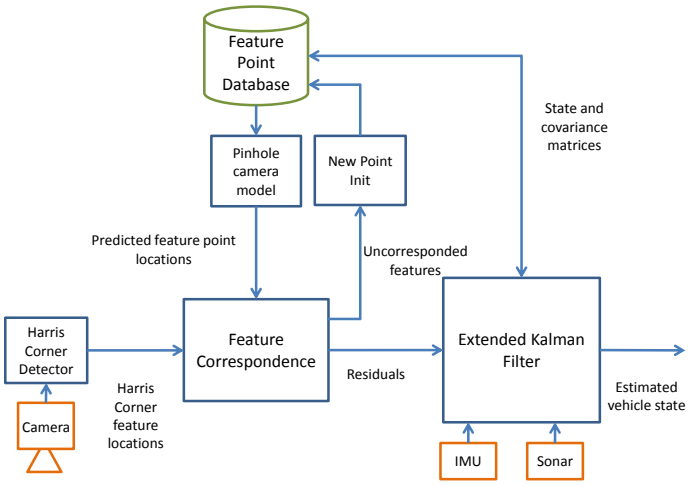


Figure 3: A block diagram describing the navigation system. Pixel location of features are corresponded to predicted locations of database points. The resulting residual is fused with inertial data and sonar altimeter data to estimate the vehicle state, feature locations and associated covariance matrices. Uncorresponded features can be used to update the database as current points leave the field of view. The resulting architecture simultaneously estimates feature point locations and vehicle states without requiring the vehicle to maintain view of particular landmarks.

the vehicle state is propagated by integrating data from the IMU. A schematic of the algorithm is shown in Figure 3. Pseudocode of the algorithm is given in Algorithm 1.

3.1 Propagation of the Estimator

The extended Kalman filter propagates the covariance of the state estimate at each time step using the Jacobian of the dynamic model evaluated at the current state. The nonzero elements of the Jacobian matrix A are given below

$$\frac{\partial \dot{\hat{p}}^i}{\partial \hat{v}^i} = I_{3\times 3}, \qquad \frac{\partial \dot{\hat{v}}^i}{\partial \hat{R}} = -\hat{L}_{ib}\widetilde{s}, \qquad \frac{\partial \dot{\hat{v}}^i}{\partial \hat{s}_b} = -\hat{L}_{ib}, \tag{33}$$
$$\frac{\partial \dot{\hat{R}}}{\partial \hat{R}} = -\widetilde{\omega}, \qquad \frac{\partial \dot{\hat{R}}}{\partial \hat{\omega}_b} = -I_{3\times 3},$$

where $\widetilde{a}$ indicates the skew symmetric matrix composed of the components of a.

The database of feature points are parameterized as either Cartesian coordinates in the inertial frame or as inverse depth points, as described in Section 2.5.

$$\hat{x}_f = \begin{bmatrix} \hat{p}_{f_1} & \dots & \hat{p}_{f_{N_f}} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{N_{fs}N_f} \tag{34}$$

Algorithm 1 Navigation update and confidence-based database management

while new IMU packets are available **do**
  Propagate state and covariance matrix.
  if new image available **then**
    Extract Harris corner features.

    //Point correspondence
    for All database points j **do**
      Project database point onto image plane.
      for All features i **do**
        Calculate Mahalanobis distance $Z_{ij} = e_{ij}^{\mathrm{T}}(C_j P C_j^{\mathrm{T}} + R)^{-1} e_{ij}$.
      end for
    end for
    for All database points j **do**
      Find feature i_{min} such that $\{ Z_{i_{min}j} \leq Z_{ij}$ for all i **and** $Z_{i_{min}j} \leq Z_{max} \}$.
      if feature i_{min} already corresponded to database point j_{prev}
      and $Z_{i_{min}j} < Z_{i_{min}j_{prev}}$ **then**
        Correspond database point j to feature i_{min} and recorrespond database point j_{prev}.
      end if
    end for

    //Kalman filter update
    for All database points j **do**
      if Database point j was corresponded **then**
        Increment confidence in database point j.
        Perform Kalman filter update.
      else
        Decrement confidence in database point j.
      end if
    end for

    //Point initialization
    for All features i **do**
      if Feature i was uncorresponded **then**
        Get inertial frame point location based on altitude, Eq. (44).
        Set confidence of new point based on number of corresponded database points during previous update.
        for All database points j **do**
          if New point confidence > database point confidence **then**
            Insert new point in database location j.
            break
          end if
        end for
      end if
    end for
  end if
end while

where N_{fs} is the size of each feature stat vector, and N_f is the number of stored features. The covariance matrix for each feature point is P_{f_j}. Database points are assumed to be static and have no process noise, therefore no propagation is necessary.

3.2 State and Covariance Update

The measurement model used in the navigation algorithm is the standard pinhole camera model. An undistortion transform is applied to the image features before being used in the algorithm, and so a simple camera model is appropriate. The camera model is described in Section 2.4.

The navigation algorithm performs the state update in a sequential manner. Each feature point is treated as an independent measurement, and correlations between the measurement and state are ignored. At each time step, a subset of the feature database is observed and used to update the state. During each feature update j, the Jacobian of the measurement model, C_k, is calculated at the current state estimate:

$$C_k = \left[\frac{\partial z_j}{\partial \hat{x}_v} \quad \frac{\partial z_j}{\partial \hat{p}_{f_j}} \right]_{(\hat{x}_v, \hat{p}_{f_j}) = (\hat{x}_v, \hat{p}_{f_j})_{k|k-1}} \tag{35}$$

Subscript $k|k-1$ indicates the *a priori* estimate at timestep k. The state and covariance update is calculated according to the familiar EKF update equations. The combined covariance for the state and feature point j is given by

$$P_j = \begin{bmatrix} P_v & 0 \\ 0 & P_{f_j} \end{bmatrix} \tag{36}$$

Then,

$$K_k = P_{jk|k-1} C_k^{\mathrm{T}} \left(C_k P_{jk|k-1} C_k^{\mathrm{T}} + R \right)^{-1} \tag{37}$$

$$\begin{bmatrix} \hat{x}_v \\ \hat{p}_{f_j} \end{bmatrix}_{k|k} = \begin{bmatrix} \hat{x}_v \\ \hat{p}_{f_j} \end{bmatrix}_{k|k-1} + K_k (z_j - h(\hat{x}_{v,k|k-1}, \hat{p}_{f_j k|k-1})) \tag{38}$$

$$P_{jk|k} = (I - K_k C_k) P_{jk|k-1} \tag{39}$$

A note on the notation: the "timestep" k is incremented on each feature update, even though no time has passed while updating measurements from a given frame,

and therefore no propagation step has been carried out. In other words, for a given image, $\begin{bmatrix} \hat{x}_v \\ \hat{p}_{f_j} \end{bmatrix}_{k+1|k} = \begin{bmatrix} \hat{x}_v \\ \hat{p}_{f_j} \end{bmatrix}_{k|k}$ and $P_{v,k+1|k} = P_{v,k|k}$.

3.3 Point Correspondence

Before extracted features can be used in the EKF, it must first be determined which database point they correspond to. The Mahalanobis distance is used to narrow the pool of potential matches, and then a feature descriptor is used to choose the best match from the pool. The use of a descriptor added robustness by limiting the number of false matches, especially in the presence of state error when the nearest statistical point may not be the best.

The feature detector is a modified Harris corner detector, which extracts points based on the image gradient in orthogonal directions[18]. To ensure good feature separation, the detector partitions the image into bins, and sets a maximum number of features in each bin. Also, a minimum distance between features can be set. The feature descriptor used to uniquely identify the point is the pixel intensity in a window surrounding the feature. The feature locations as well as the intensity descriptor is passed to the navigation algorithm in order of their Harris corner score, up to a maximum number.

The search region is determined using the estimated location uncertainty:

$$S^j = (C^j P_j C^{j\mathrm{T}} + R) \tag{40}$$

The statistical distance between measurement z_i and the image plane location of point p_{f_j} is given by

$$e_{ij} \triangleq z_i - h(\hat{x}_v, \hat{p}_{f_j}) \tag{41}$$

$$Z_{ij} = e_{ij}^{\mathrm{T}} (S^j)^{-1} e_{ij} \tag{42}$$

A maximum value, Z_{lim}, is set for Z_{ij} for measurement i and database point j to be considered as possible matches.

All measurements satisfying the Z_{ij} threshold are compared to the database point using their image descriptor. The average difference between pixel intensities is calculated and the pair with the lowest difference, below a maximum threshold, is considered to be corresponded to the database point.

The descriptor from the corresponded measurement is stored with the database point for use in the next iteration. This method, as opposed to retaining the original descriptor for the life of the database point, allows the appearance of the point to change slowly as the camera location changes, and makes it unnecessary to use more complex scale and rotation invariant descriptors.

If a measurement z_i does not meet the conditions to match any database point, it is considered to be a measurement of no currently stored point and is uncorresponded. Uncorresponded points can be used to initialize new database points, as described in the following section.

3.4 Initialization of New Points

New points are initialized from uncorresponded features, and features with the highest corner score are initialized first. Because of the lack of depth information from a single monocular image, additional information is necessary to estimate the measurements' state vector from their two-dimensional image plane location. In many common environments for vehicles with a camera aligned with the body z-axis, it is a reasonable assumption that the points lie on a plane at a distance of the altitude of the vehicle. This assumption would be valid, for instance, for a vehicle flying indoors over an uncluttered floor, or a outdoor vehicle flying over a plane. By initializing points with a suitable uncertainty in altitude, points that are in fact not on the ground plane will converge to the correct altitude if they remain in view [52].

The following steps describe the initialization for Cartesian parameterized features. The inverse depth initialization is similar. Let the direction of the feature in the inertial frame be given by

$$h_f^i = L_{ic} \begin{bmatrix} 1 \\ u/f_u \\ v/f_v \end{bmatrix} \tag{43}$$

Then the 3d location of the new feature is

$$h_f^i \frac{X_j^i}{h_{f_1}^i} = \begin{bmatrix} X_j \\ Y_j \\ Z_j \end{bmatrix}^i = r_{f_j}^i. \tag{44}$$

$$\hat{p}_{f_j} = r_{f_j}^i + \hat{p} \tag{45}$$

Distance X_j^i is assumed to be the altitude of the camera, and image plane locations u and v and focal lengths f_x and f_y are known. The covariance of the new feature is set to fixed value scaled by the distance of the vehicle to that feature:

$$P_{f_j} = L_{ic} \begin{bmatrix} ||r_{f_j}^i||^2 & 0 & 0 \\ 0 & R_0||r_{f_j}^i||^2/f_u^2 & 0 \\ 0 & 0 & R_0||r_{f_j}^i||^2/f_v^2 \end{bmatrix} L_{ic}^{\mathrm{T}}. \tag{46}$$

where P_{f_j} is the 3×3 covariance of point p_{f_j} with itself, and R_0 is a scalar initialization value. All other cross-correlation terms in P are ignored.

Feature removal from the database is decided by storing a probabilistic measure of the feature's existence. A similar method is used in occupancy grid models [46]. The probability is stored in log-odds form:

$$l_{t,i} = \log\left(\frac{p(f_i|z_{1:t})}{1 - p(f_i|z_{1:t})}\right) \tag{47}$$

where $p(f_i|z_{1:t})$ is the probability of the existence of feature i conditioned on the measurements until time t. Bayesian inference on a binary random variable can be

given in the following log-odds form.

$$l_{t,i} = \log\left(\frac{p(f_i|z_t)}{1-p(f_i|z_t)}\right) + \log\left(\frac{p(f_i|z_{1:t-1})}{1-p(f_i|z_{1:t-1})}\right) - \log\left(\frac{p(f_i)}{1-p(f_i)}\right) \tag{48}$$

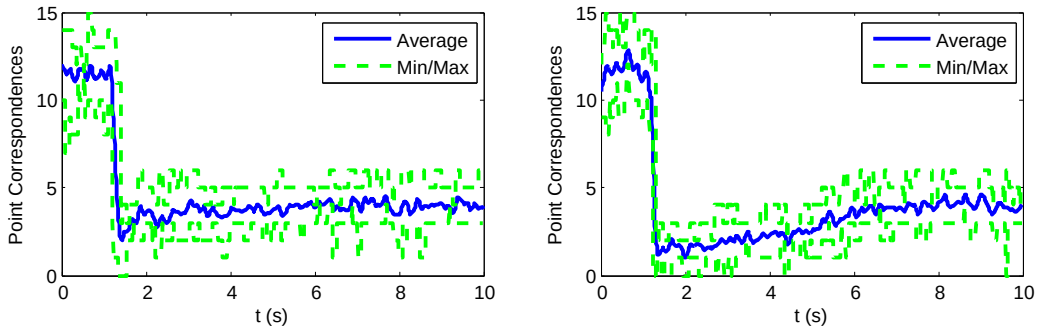
$$= l_{t-1,i} + l_{\text{update}} - l_0 \tag{49}$$

where l_{update} is the inverse measurement model output, $l_{t-1,i}$ is the log-odds of the previous time step, and l_0 is the log-odds prior of the map. l_0 is assumed to be 0, indicating a prior probability of existence of 0.5.

The inverse measurement model is often difficult to determine analytically, so ad hoc methods are sometimes used. In this work, the probability of existence given a feature correspondence, $p(f_i|correspondence) \approx .75$, and the probability of existence given a noncorrespondence, $p(f_i|noncorrespondence) \approx .45$. This leads to log-odds values of 0.92 and -0.184, respectively. By limiting the maximum and minimum value of $l_{t,i}$ to ± 4.6 ($0.01 < p(f) < .99$), a scale is constructed on which a capped, weighted count of correspondences and noncorrespondences is stored for each feature. Without loss of generality, the count, max and min, and update values may be scaled and translated. On a scale from 0 to 100, for example, the values for a correspondence and noncorrespondnence are 10 and -2 respectively.

A sliding threshold is used to determine whether features exist or not, and thus whether they are candidates for replacement. This threshold is set high when few features are being corresponded, so that features are replaced quickly, and low when many features are being corresponded, so that they are replaced slowly. The new features are initialized at the value of the threshold. In this work the threshold is set to 3.7 if 2 or fewer features were corresponded, -3.7 if 10 or more features were corresponded, and 0 otherwise.

To demonstrate the effectiveness of the proposed method, it is compared to a fixed threshold feature removal method. The fixed threshold method is implemented by setting the existence threshold to 0 regardless of the number of features corresponded.



(a) (b)

Figure 4: The number of feature correspondences over time. The left plot data was gathered using the dynamic thresholding. The right plot uses a fixed threshold and is shown for comparison. After the change in image texture at $t = 1$, the dynamic threshold method more quickly removes old points from the database.

This fixed threshold method is equivalent to the method proposed in [9] and [10], with the number of iterations equal to 50 and the percent correspondence equal to 50%.

The simulated vehicle was flown over a highly textured surface which was instantaneously changed to a low textured surface. This is similar to the effect of instantaneous lighting changes, such as a light being switched off in a room. The texture removal was performed 8 times for each method, and the average, minimum and maximum number of feature correspondences is shown in Figure 4. It can be seen that the dynamic threshold does a better job maintaining a steady number of consistent features throughout the test. When the texture is removed, the fixed threshold takes a longer time to throw out the now-unobservable features, as the confidence index for these points fall. In the dynamic thresholding case, the threshold is raised when few features correspond, and so unseen features are removed quickly. The dynamic threshold method allows the navigation algorithm to rapidly adapt to dynamic environment.

3.5 Results

Validation of the vision-based navigation system was conducted on the GTMax [26], a 80 kg modified Yamaha RMAX helicopter UAV with custom avionics and flight software designed by the Georgia Institute of Technology Unmanned Aerial Vehicle Research Facility (UAVRF). Figure 5 shows a picture of the GTMax. The helicopter is outfitted with a variety of sensors including an Inertial Science IMU sampled at 100 Hz, short-range sonar, magnetometer, and differential GPS. The vehicle was equipped with a Prosilica GC 1380 camera with global shutter for vision-based navigation. Two onboard computers with 1.6 GHz Intel Pentium M CPUs process information from these sensors for guidance, navigation and control of the vehicle. For the vision-based navigation described in this paper, the secondary computer performs image acquisition and feature detection, while the primary computer performs the navigation update and handles vehicle guidance and control. All flight results presented below employ the GTMax's baseline adaptive flight controller that has been described in detail in [25].

Simulation results presented below were generated using the Georgia Tech UAV Simulation Tool (GUST). The GUST software package that combines a high-fidelity vehicle and environment model, onboard flight control software, and ground station software. The vehicle model is a six rigid body degree of freedom model with additional engine and rotor dynamics. The vehicle model simulates sensor noise, delay, location, orientation, and actuator dynamics and saturation.

3.5.1 GTMax Flight Test Results

The data presented below show excerpts from a single flight during which there were 16 minutes of continuous operation of the vision-based navigation system (16 minutes is about 40% of the typical endurance of the GTMax). Figure 6 shows the navigation solution during the flight, and the top-down view also shows the location of the vehicle

Figure 5: The GTMax helicopter weighs 80 kg and has a rotor diameter of 3 m. For vision-based navigation, a downward-facing camera is mounted on the nose of the vehicle.

as given by GPS for comparison. Note that the vehicle is not drawn to scale. Speeds as high as 6 m/s were reached and the vehicle flew at multiple altitudes. The flight was carried out over a grassy area with a fence occasionally in view. The satellite imagery in Figure 6 gives a rough indication of the flight area.

For all tests, the camera was mounted to the nose of the aircraft pointing downward, and monochrome images were gathered at approximately 20 Hz and at 320×240 pixel resolution. The best features in each image, with a maximum of 20, were output for use in the navigation algorithm. A maximum of 50 feature locations were stored in the database. Differential GPS data was gathered at 5 Hz and was used for altitude measurement in the navigation solution. The measurement is analogous to using a barometric pressure sensor, in that both measurements are relative to an initial datum, rather than the altitude above ground. Also, the GPS position and velocity output is presented below for comparison to the vision-based navigation in the horizontal plane.

Figures 7 and 8 show the results of a series of translational commands given to the GTMax while using vision-based navigation. Figure 7 shows that the navigation solution corresponds well with the GPS output, despite the vehicle traveling over 90 m each way, and having to replace online the database of feature locations. At time 100-120 seconds, it can be seen that a velocity error causes the vehicle to drift. This

(a)

(b)

Figure 6: The ground station visualization of full 16 minute (40% of aircraft endurance) flight trajectory while using vision-based navigation. GTMax flight test data presented in this paper, except for the landing maneuver, is excerpted from this flight. The navigation output is shown in red, and GPS data from the flight is show in blue, for comparison. Note that the image of the vehicle is not drawn to scale.

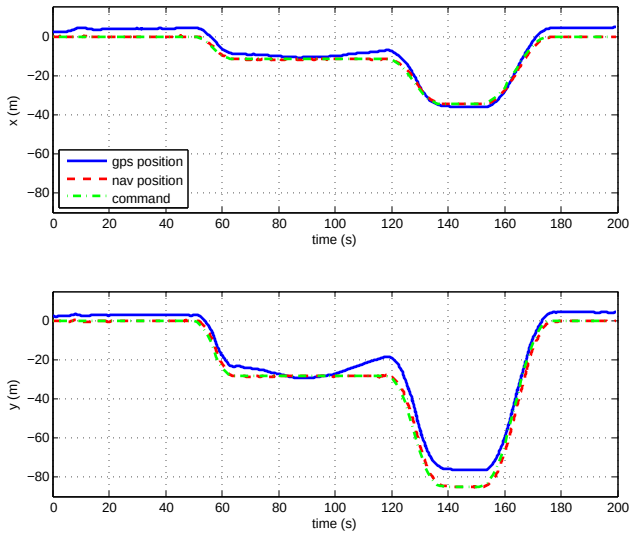


Figure 7: Comparison of vision-based navigation position states (red dashed line) with GPS location data (blue solid line) of the GTMax during translational commands. GPS data was used in navigation solution for altitude only. The commands were given as indicated by the free dash-dot line. The vehicle tracked the commanded positions with 7 m drift of the vision-based navigation solution from GPS position data. These results are part of a 16 minute flight.

drift is possibly caused by mis-corresponded measurement and database features, which can occur over areas with poor texture and poor feature distribution over the image. In poorly textured areas, features are less repeatable from frame to frame, and periodic dropouts of features could cause a database point to be incorrectly matched to a neighboring feature. An absolute position error of 7 m is incurred over the course of the maneuver.

Figures 9 and 10 show the results of a 90 second hover of the GTMax. The vehicle remains within approximately a 1.6 m radius circle around the commanded position.

Figure 11 show the performance of the V-INS solution during autonomous descent and landing. This is the first time an autonomous landing has been performed under vision-based navigation for a vehicle of this class in completely unmapped environments, to the author's knowledge [5]. This maneuver demonstrates that the navigation solution is robust to large altitude changes and corresponding changes in the

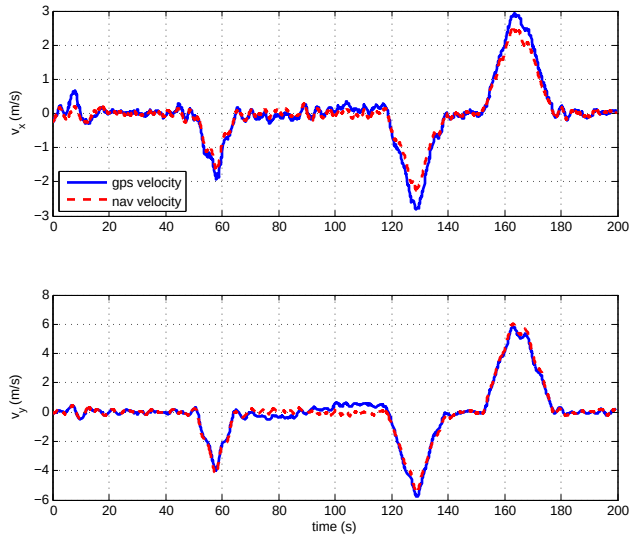


Figure 8: Comparison of vision-based navigation velocity states (red dashed line) with GPS velocity data (blue solid line) of the GTMax during translational input commands. These results are part of a 16 minute flight.

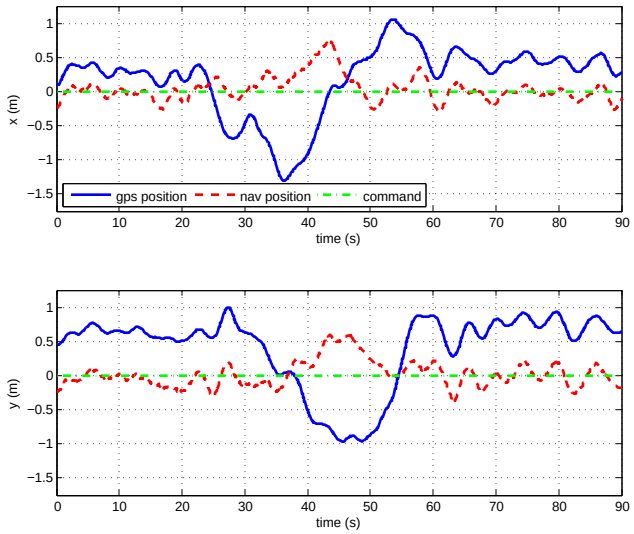


Figure 9: Comparison of vision-based navigation position states (red dashed line) with GPS location data (blue solid line) of the GTMax during a 90 second excerpt of hovering flight approximately 100 ft above ground. GPS data was used in navigation solution for altitude only, and this can easily be replaced with a barometer or active range sensor. The vehicle remained within approximately 1.6 m of the commanded position despite a small amount of drift in the navigation solution from 22 seconds to 60 seconds. These results are part of a 16 minute flight.

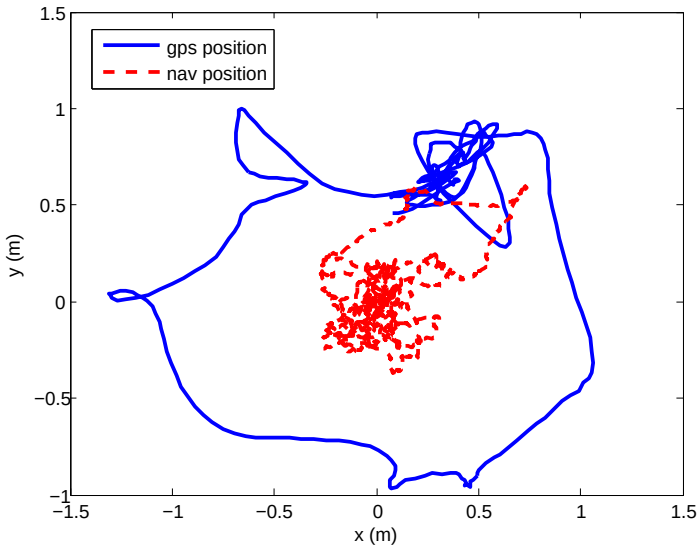


Figure 10: X-Y position of vehicle as given by vision-aided navigation solution (red dashed line) and raw differential GPS data (blue solid line) of the GTMax during a 90 second excerpt from extended hovering flight approximately 100 ft above ground. These results are part of a 16 minute flight.

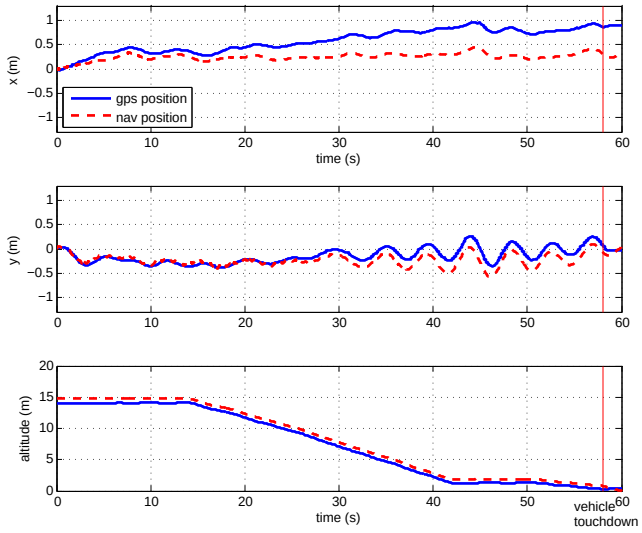


Figure 11: Time history of the position states of the GTMax navigation solution during autonomous descent and landing. The red dashed line gives the navigation solution position, and the blue solid line gives the GPS position. A video of the maneuver can be found at `http://uav.ae.gatech.edu/videos/g120124e1_visionBasedLanding.wmv`.

surface texture beneath the vehicle, and that the solution can be trusted during maneuvers with low velocity error tolerance. A video of the maneuver can be found online at `http://uav.ae.gatech.edu/videos/g120124e1_visionBasedLanding.wmv`.

3.5.2 GTMax Simulation Results

A 350 second simulation was performed to analyze the performance of the navigation system over an area with a non-flat ground plane, and provide further validation of long-distance flight. During this simulation, the vehicle traveled over 560 m at an altitude of 50 m, over a group of buildings up to three stories tall (approximately 10 m). Images were captured from a simulated camera located on the vehicle. An image from the simulation environment, along with the true path of the vehicle and the waypoints of the trajectory, is shown in Figure 12.

Figure 13 show the GPS and navigation output in the x-y plane and with respect to time. The final navigation error over the 560 m flight path was 5 m. Note that no loop-closure was performed, indicating that similar performance can be expected

over non-repeating paths. It can be seen that the navigation solution provides data which ensures the stability of the vehicle during the entire trajectory.

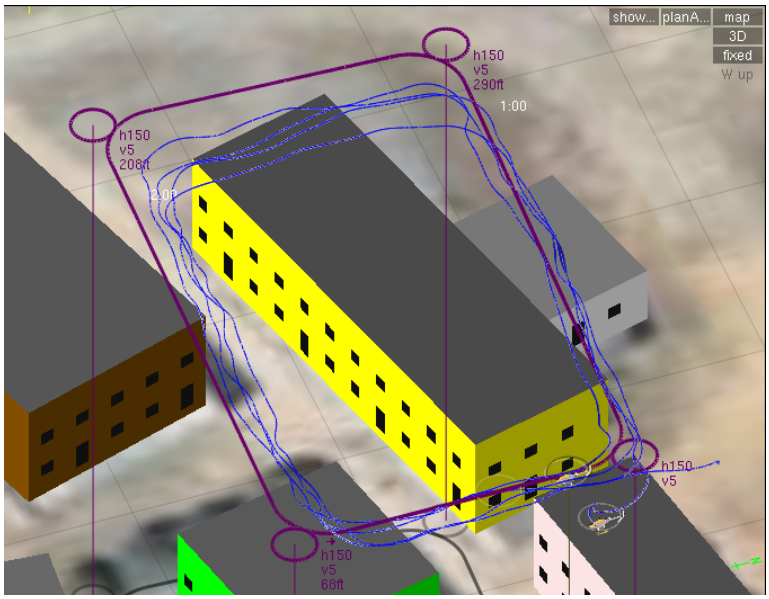
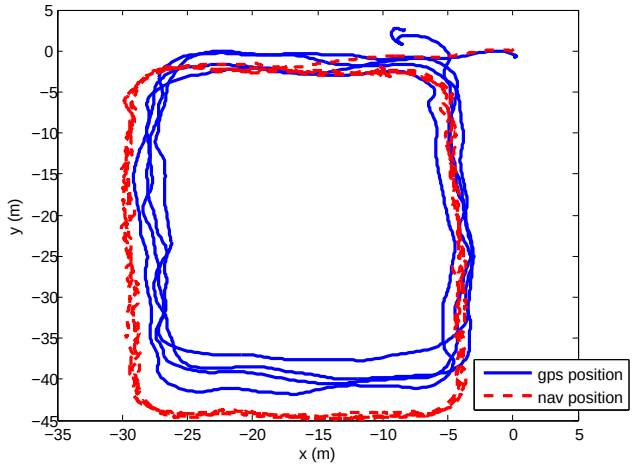
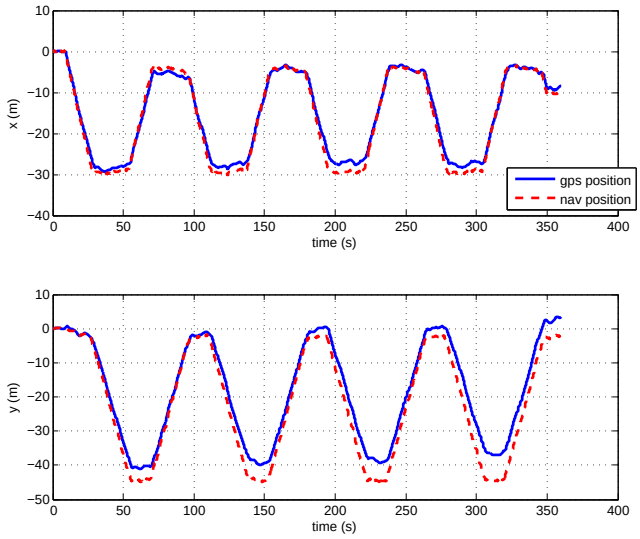


Figure 12: Flight path during a simulated flight over a group of buildings. The vehicle flew at an altitude of 50 m and and buildings were as tall as 10 m. The vehicle flew four times around the path indicated by the purple waypoints, for a total linear distance of 560 m. The true path of the vehicle is shown in blue, and the final error between the true position and navigation solution was 5 m.



(a)



(b)

Figure 13: Plots illustrating the navigation solution (red) and GPS position (blue) during a simulated flight over non-flat ground.

CHAPTER IV

MINIMAL STATE UPDATE FOR MODULAR STATE ESTIMATION

This chapter describes a novel modular filter which allows the previous vision-aided navigation architecture to be used with a wide variety of navigation filters. This architecture makes use of a theorem stating the minimum state information necessary for the architecture, and is algebraically equivalent to standard EKF formulation presented in Chapter 3. This theorem is general and can be applied to any modular filter satisfying the given conditions.

The use of this architecture greatly simplifies integration of the vision-aided navigation component of a complex navigation filter, and allows the vision component to be designed independently of the rest of the filter.

4.1 Equivalent State Update

The following theorem describes how the full vehicle state estimator may be equivalently updated by an update to part of the state and knowledge of the change in state and covariance. Let the probabilistic estimate of the full system, system $\mathcal{A}$, be defined as

$$\mathcal{N}(x_{\mathcal{A}}, P_{\mathcal{A}}). \tag{50}$$

and the estimate of the partial system, system $\mathcal{B}$, be

$$\mathcal{N}(x_{\mathcal{B}}, P_{\mathcal{B}}). \tag{51}$$

The following two assumptions are made about the relationship between systems $\mathcal{A}$ and $\mathcal{B}$:

Assumption 4.1.1 *For a given measurement z, the systems have associated measurement matrices $H_{\mathcal{A}}$ and $H_{\mathcal{B}}$.*

$$z = H_{\mathcal{B}} x_{\mathcal{B}} = H_{\mathcal{A}} x_{\mathcal{A}} \tag{52}$$

Assumption 4.1.2 *The state vector $x_{\mathcal{B}}$ is a linear combination of $x_{\mathcal{A}}$:*

$$x_{\mathcal{B}} = H_c x_{\mathcal{A}} \tag{53}$$

Note that under Assumption 4.1.1 and 4.1.2 the following identities hold:

$$H_{\mathcal{A}} P_{\mathcal{A}} H_{\mathcal{A}}^{\mathrm{T}} = H_{\mathcal{B}} P_{\mathcal{B}} H_{\mathcal{B}}^{\mathrm{T}} \tag{54}$$

$$H_{\mathcal{A}} = H_{\mathcal{B}} H_c \tag{55}$$

The following theorem describes the minimal state update.

Theorem 4.1.1 *Given estimates (50) and (51), and associated measurement matrices, and under assumption 4.1.1 and 4.1.2, then the following Kalman filter update equations*

$$x_{\mathcal{A}}(+) = x_{\mathcal{A}}(-) + P_{\mathcal{A}}(-) H_{\mathcal{A}}^{\mathrm{T}} \left(H_{\mathcal{A}} P_{\mathcal{A}}(-) H_{\mathcal{A}}^{\mathrm{T}} + R \right)^{-1} (z - H_{\mathcal{A}} x_{\mathcal{A}}) \tag{56}$$

$$P_{\mathcal{A}}(+) = P_{\mathcal{A}}(-) - P_{\mathcal{A}}(-) H_{\mathcal{A}}^{\mathrm{T}} \left(H_{\mathcal{A}} P_{\mathcal{A}}(-) H_{\mathcal{A}}^{\mathrm{T}} + R \right)^{-1} H_{\mathcal{A}} P_{\mathcal{A}}(-) \tag{57}$$

Can be equivalently expressed in terms of $\Delta x_{\mathcal{B}}$ and $\Delta P_{\mathcal{B}}$ given by

$$x_{\mathcal{B}}(+) = x_{\mathcal{B}}(-) + P_{\mathcal{B}}(-) H_{\mathcal{B}}^{\mathrm{T}} \left(H_{\mathcal{B}} P_{\mathcal{B}}(-) H_{\mathcal{B}}^{\mathrm{T}} + R \right)^{-1} (z - H_{\mathcal{B}} x_{\mathcal{B}}) \tag{58}$$

$$= x_{\mathcal{B}}(-) + \Delta x_{\mathcal{B}} \tag{59}$$

$$P_{\mathcal{B}}(+) = P_{\mathcal{B}}(-) - P_{\mathcal{B}}(-) H_{\mathcal{B}}^{\mathrm{T}} \left(H_{\mathcal{B}} P_{\mathcal{B}}(-) H_{\mathcal{B}}^{\mathrm{T}} + R \right)^{-1} H_{\mathcal{B}} P_{\mathcal{B}}(-) \tag{60}$$

$$= P_{\mathcal{B}}(-) + \Delta P_{\mathcal{B}} \tag{61}$$

with the following modified update equations:

$$x_{\mathcal{A}}(+) = x_{\mathcal{A}}(-) + P_{\mathcal{A}}(-) \tilde{H}^{\mathrm{T}} \left(\tilde{H} P_{\mathcal{A}}(-) \tilde{H}^{\mathrm{T}} + \tilde{R} \right)^{-1} (\tilde{z} - \tilde{H} x_{\mathcal{A}}) \tag{62}$$

$$P_{\mathcal{A}}(+) = P_{\mathcal{A}}(-) - P_{\mathcal{A}}(-) \tilde{H}^{\mathrm{T}} \left(\tilde{H} P_{\mathcal{A}}(-) \tilde{H}^{\mathrm{T}} + \tilde{R} \right)^{-1} \tilde{H} P_{\mathcal{A}}(-) \tag{63}$$

where

$$\tilde{H} \triangleq L_1^{\mathrm{T}} P_{\mathcal{B}}^{-1}(-) H_c \tag{64}$$

$$\tilde{R} \triangleq -L_1^{\mathrm{T}} P_{\mathcal{B}}^{-1}(-) L_1 - D_1^{-1} \tag{65}$$

$$\tilde{z} \triangleq \tilde{H} x_{\mathcal{A}}(-) - D_1^{-1} L_1^{\mathrm{T}} \Delta x_{\mathcal{B}} \tag{66}$$

$$\Delta P_{\mathcal{B}} = LDL^{\mathrm{T}} = \begin{bmatrix} L_1 & L_2 \end{bmatrix} \begin{bmatrix} D_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} L_1^{\mathrm{T}} \\ L_2^{\mathrm{T}} \end{bmatrix} \tag{67}$$

and LDL^{T} is the diagonalization (eigenvalue decomposition) of symmetric matrix $\Delta P_{\mathcal{B}}$.

Proof.

Consider the desired covariance update

$$P_{\mathcal{A}}(+) = P_{\mathcal{A}}(-) - P_{\mathcal{A}}(-) H_{\mathcal{A}}^{\mathrm{T}} \left(H_{\mathcal{A}} P_{\mathcal{A}}(-) H_{\mathcal{A}}^{\mathrm{T}} + R \right)^{-1} H_{\mathcal{A}} P_{\mathcal{A}}(-) \tag{68}$$

Applying Eq. (54) and (55) and dropping the "(−)" for clarity, gives

$$P_{\mathcal{A}}(+) = P_{\mathcal{A}} - P_{\mathcal{A}} H_c^{\mathrm{T}} H_{\mathcal{B}}^{\mathrm{T}} \left(H_{\mathcal{B}} H_c P_{\mathcal{A}} H_c^{\mathrm{T}} H_{\mathcal{B}}^{\mathrm{T}} + R \right)^{-1} H_{\mathcal{B}} H_c P_{\mathcal{A}} \tag{69}$$

$$= P_{\mathcal{A}} - P_{\mathcal{A}} H_c^{\mathrm{T}} P_{\mathcal{B}}^{-1} P_{\mathcal{B}} H_{\mathcal{B}}^{\mathrm{T}} \left(H_{\mathcal{B}} P_{\mathcal{B}} H_{\mathcal{B}}^{\mathrm{T}} + R \right)^{-1} H_{\mathcal{B}} P_{\mathcal{B}} P_{\mathcal{B}}^{-1} H_c P_{\mathcal{A}} \tag{70}$$

Inserting $\Delta P_{\mathcal{B}}$,

$$P_{\mathcal{A}}(+) = P_{\mathcal{A}} + P_{\mathcal{A}} H_c^{\mathrm{T}} P_{\mathcal{B}}^{-1} \Delta P_{\mathcal{B}} P_{\mathcal{B}}^{-1} H_c P_{\mathcal{A}}. \tag{71}$$

The change in covariance $\Delta P_{\mathcal{B}}$ is symmetric positive semidefinite, so there exists a eigendecomposition $\Delta P_{\mathcal{B}} = LDL^{\mathrm{T}}$ unique up to permutation of columns of L. Furthermore, note that if $\Delta P_{\mathcal{B}}$ is not full rank, then with appropriate order of columns of L,

$$\Delta P_{\mathcal{B}} = LDL^{\mathrm{T}} = \begin{bmatrix} L_1 & L_2 \end{bmatrix} \begin{bmatrix} D_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} L_1^{\mathrm{T}} \\ L_2^{\mathrm{T}} \end{bmatrix} = L_1 D_1 L_1^{\mathrm{T}} \tag{72}$$

where D_1 is full rank. Inserting into Eq. (71),

$$P_{\mathcal{A}}(+) = P_{\mathcal{A}} + P_{\mathcal{A}} H_c^{\mathrm{T}} P_{\mathcal{B}}^{-1} L_1 D_1 L_1^{\mathrm{T}} P_{\mathcal{B}}^{-1} H_c P_{\mathcal{A}}. \tag{73}$$

Now, applying definition (64) and letting

$$D_1 = -\left(\tilde{H} P_{\mathcal{A}} \tilde{H}^{\mathrm{T}} + \tilde{R}\right)^{-1} \tag{74}$$

gives desired form of update in Eq. (60). Solving for $\tilde{R}$,

$$\tilde{R} = -L_1^{\mathrm{T}} P_{\mathcal{B}}^{-1}(-) L_1 - D_1^{-1} \tag{75}$$

Next, consider the state update, and applying Eq. (54), (52), and (55), inserting $\Delta x_{\mathcal{B}}$ and applying definition (64),

$$\begin{aligned}
x_{\mathcal{A}}(+) &= x_{\mathcal{A}} + P_{\mathcal{A}} H_{\mathcal{A}}^{\mathrm{T}} \left(H_{\mathcal{A}} P_{\mathcal{A}} H_{\mathcal{A}}^{\mathrm{T}} + R\right)^{-1} (z - H_{\mathcal{A}} x_{\mathcal{A}}) && (76)\\
&= x_{\mathcal{A}} + P_{\mathcal{A}} H_c^{\mathrm{T}} P_{\mathcal{B}}^{-1} P_{\mathcal{B}} H_{\mathcal{B}}^{\mathrm{T}} \left(H_{\mathcal{B}} P_{\mathcal{B}} H_{\mathcal{B}}^{\mathrm{T}} + R\right)^{-1} (z - H_{\mathcal{B}} x_{\mathcal{B}}) && (77)\\
&= x_{\mathcal{A}} + P_{\mathcal{A}} H_c^{\mathrm{T}} P_{\mathcal{B}}^{-1} \Delta x_{\mathcal{B}} && (78)\\
&= x_{\mathcal{A}} + P_{\mathcal{A}} H_c^{\mathrm{T}} P_{\mathcal{B}}^{-1} L_1 L_1^{\mathrm{T}} \Delta x_{\mathcal{B}} && (79)\\
&= x_{\mathcal{A}} + P_{\mathcal{A}} \tilde{H}^{\mathrm{T}} L_1^{\mathrm{T}} \Delta x_{\mathcal{B}} && (80)
\end{aligned}$$

Setting $L_1^{\mathrm{T}} \Delta x_{\mathcal{B}} = \left(\tilde{H} P_{\mathcal{A}} \tilde{H}^{\mathrm{T}} + \tilde{R}\right)^{-1} (\tilde{z} - \tilde{H} x_{\mathcal{A}})$ gives the desired form of the state update. Then, inserting Eq. (74) and solving for $\tilde{z}$ gives

$$\begin{aligned}
L_1^{\mathrm{T}} \Delta x_{\mathcal{B}} &= -D_1 (\tilde{z} - \tilde{H} x_{\mathcal{A}}) && (81)\\
\implies \tilde{z} &= \tilde{H} x_{\mathcal{A}} - D_1^{-1} L_1^{\mathrm{T}} \Delta x_{\mathcal{B}} && (82)
\end{aligned}$$

■

Theorem 4.1.1 describes the necessary transformations applied to the state and covariance updates, $\Delta x_{\mathcal{B}}$ and $\Delta P_{\mathcal{B}}$, performed on system $\mathcal{B}$, in order to incorporate the update into system $\mathcal{A}$ as a measurement update with a chosen form (within the limits set by assumptions). Application of this theorem allows system $\mathcal{A}$ to be updated

with only limited information about the update of $\mathcal{B}$. System $\mathcal{A}$ needs the following information about the update to $\mathcal{B}$: Updates $\Delta x_{\mathcal{B}}$, $\Delta P_{\mathcal{B}}$, and prior covariance $P_{\mathcal{B}}(-)$. Additionally, system $\mathcal{B}$ can limit the state vector to only those states required by the measurement model.

4.2 Modular Vision-aided Navigation

An example of Theorem 4.1.1 applied to a vision-aided inertial navigation system is presented in the following section. The presentation highlights the key advantage of this architecture; the vision-aiding system system $\mathcal{B}$. is independent of changes in the structure of the primary navigation system, system $\mathcal{A}$. Furthermore, the vision-aiding system update is incorporated into the primary navigation system as a measurement update, and can be designed independently.

The primary navigation system state vector is composed of vehicle position, attitude, and velocity as well as sensor states for bias and scale factor estimation. These additional states can be dynamically added to or removed from the state vector in accordance with availability of a particular sensor. Let the state vector be defined as

$$x_{\mathcal{A}} = \begin{bmatrix} p & R & v & \dots & < otherstates > & p_f \end{bmatrix}. \tag{83}$$

The linearized measurement model for feature measurements is dependent on vehicle states alone:

$$H_{\mathcal{A}} = \left[\begin{array}{cccc|c} \frac{\partial z}{\partial p} & \frac{\partial z}{\partial R} & 0 & \dots & 0 & \frac{\partial z}{\partial p_f} \end{array}\right] = \begin{bmatrix} H_{\mathcal{A}_v} & H_{\mathcal{A}_f} \end{bmatrix} \in \mathbb{R}^{2\times N_{\mathcal{A}}} \tag{84}$$

As described int Chapter 3, correlations between the features and the state are ignored, and the covariance of system $\mathcal{A}$ is given by

$$P_{\mathcal{A}} = \begin{bmatrix} P_{\mathcal{A}_v} & 0 \\ 0 & P_{\mathcal{A}_f} \end{bmatrix} \tag{85}$$

The vision module performs the state update described in Chapter 3. Propagation is performed in the primary navigation system and when new vision data becomes

available, the required information is passed to the vision module. The information necessary to perform the update is dependent on the choice of vehicle state vector for the vision module, which in turn is dependent on satisfying Assumptions 4.1.1 and 4.1.2. For the vision module vehicle state vector, we choose only those states from $x_{\mathcal{A}}$ on which the measurement module depends. This gives a state vector and measurement matrix

$$x_{\mathcal{B}} = \begin{bmatrix} p & R & p_f \end{bmatrix} \tag{86}$$

$$H_{\mathcal{B}} = \left[\begin{array}{ccc|c} \frac{\partial z}{\partial p} & \frac{\partial z}{\partial R} & & \frac{\partial z}{\partial p_f} \end{array}\right] = \begin{bmatrix} H_{\mathcal{B}_v} & H_{\mathcal{B}_f} \end{bmatrix} \in \mathbb{R}^{2\times(6+3N_f)} \tag{87}$$

which satisfies Assumption 4.1.1. We now note that Assumption 4.1.2 is satisfied with the following definition of H_c

$$H_c = \left[\begin{array}{ccc|c} I_6 & 0 & \dots & 0 \\ \hline 0 & 0 & \dots & I_{3N_f} \end{array}\right] = \begin{bmatrix} H_{c_v} & 0 \\ 0 & H_{c_f} \end{bmatrix} \in \mathbb{R}^{(6+3N_f)\times N_{\mathcal{A}}} \tag{88}$$

$$x_{\mathcal{B}} = H_c x_{\mathcal{A}} \tag{89}$$

The feature states are updated as before, and the update of $x_{\mathcal{B}}$ and associated covariance $P_{\mathcal{B}}$ is performed as described in Theorem 4.1.1. The following values are passed to the primary navigation system, where the primary state and covariance update is constructed according to Eq. (64)-(67).

$$\Delta P_{\mathcal{B}} = \begin{bmatrix} \Delta P_{\mathcal{B}_v} & 0 \\ 0 & \Delta P_{\mathcal{B}_f} \end{bmatrix} \tag{90}$$

$$P_{\mathcal{B}}(-) = \begin{bmatrix} P_{\mathcal{B}_v}(-) & 0 \\ 0 & P_{\mathcal{B}_f}(-) \end{bmatrix} \tag{91}$$

$$\Delta x_{\mathcal{B}} \tag{92}$$

The corresponding update of system $\mathcal{A}$ then occurs according to Theorem 4.1.1. Due to the neglect of feature-vehicle correlations, only the updates to block diagonal

terms $P_{\mathcal{A}_v}$ and $P_{\mathcal{A}_f}$ need be considered. In terms of variables passed from system $\mathcal{B}$ the updates are given by

$$P_{\mathcal{A}_v}(+) = P_{\mathcal{A}_v}(-) + P_{\mathcal{A}_v}(-)H_{c_v}^{\mathrm{T}}P_{\mathcal{B}_v}^{-1}(-)\Delta P_{\mathcal{B}_v}P_{\mathcal{B}_v}^{-1}(-)H_{c_v}P_{\mathcal{A}_v}(-) \tag{93}$$

$$P_{\mathcal{A}_f}(+) = P_{\mathcal{A}_f}(-) + P_{\mathcal{A}_f}(-)H_{c_f}^{\mathrm{T}}P_{\mathcal{B}_f}^{-1}(-)\Delta P_{\mathcal{B}_f}P_{\mathcal{B}_f}^{-1}(-)H_{c_f}P_{\mathcal{A}_f}(-) \tag{94}$$

Now, noting that by Assumption 4.1.2 the following equivalence holds,

$$P_{\mathcal{B}} = H_c P_{\mathcal{A}} H_c^{\mathrm{T}} \tag{95}$$

we find that the prior covariances for the feature states are equal

$$P_{\mathcal{A}_f}(-) = P_{\mathcal{B}_f}(-) \tag{96}$$

We then apply this to Equation (94).

$$P_{\mathcal{A}_f}(+) = P_{\mathcal{A}_f}(-) + P_{\mathcal{A}_f}(-)H_{c_f}^{\mathrm{T}}P_{\mathcal{B}_f}^{-1}(-)\Delta P_{\mathcal{B}_f}P_{\mathcal{B}_f}^{-1}(-)H_{c_f}P_{\mathcal{A}_f}(-) \tag{97}$$

$$= P_{\mathcal{B}_f}(-) + P_{\mathcal{B}_f}(-)I_{3N_f}P_{\mathcal{B}_f}^{-1}(-)\Delta P_{\mathcal{B}_f}P_{\mathcal{B}_f}^{-1}(-)I_{3N_f}P_{\mathcal{B}_f}(-) \tag{98}$$

$$= P_{\mathcal{B}_f}(-) + \Delta P_{\mathcal{B}_f} \tag{99}$$

Therefore the posterior covariances are also equal

$$P_{\mathcal{A}_f}(+) = P_{\mathcal{B}_f}(+) \tag{100}$$

We only need to store this information once, and so in practice the feature states and covariances are stored and updated only in system $\mathcal{B}$, saving computation time.

Note that the dynamic modification of state vector $x_{\mathcal{A}}$, such as the addition or removal of states, does not affect the design of system $\mathcal{B}$ or the information that is exchanged between the systems. The independence of the the system is a key feature which allows ease of integration into complex filter architectures.

4.3 Results

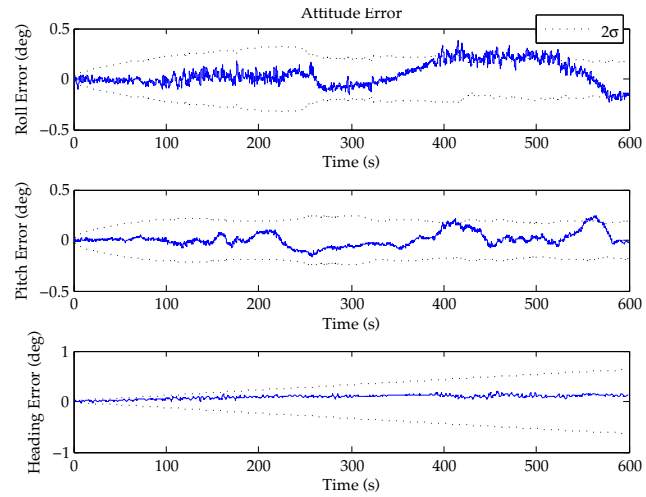
The following section presents results based on recorded sensor data using the modular filter architecture. The sensor data was gathered for the DARPA all-source positioning and navigation (APSN) program, and is from test scenario 19. In addition

to corresponded features, sensor data is was available from an inertial measurement unit, terrain reference altimeter, barometric altimeter, and magnetometer. The IMU is has a nominal acceleration bias of 1 milligee, and a gyroscopic drift of 2 deg/hr.

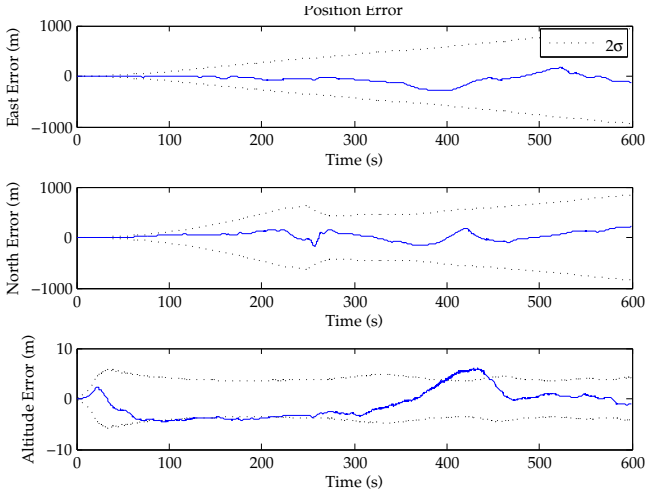
Figures 14 - 16 compare results of the navigation system running with vision updates and without. Figure 14 shows the result of the navigation system running without vision, along with 2σ uncertainty bounds. Figure 15 shows the result of the navigation system running with vision, along with 2σ uncertainty bounds. Note that the uncertainty of the navigation solution is highly overconfident due to the assumption of uncorrelated features and vehicle states, but the estimate is improved compared to the unaided estimate. Figure 16 shows the top-down view of the navigation trajectories and the ground truth estimated from GPS-INS solution. Error statistics from both cases are listed in Table 1.

Table 1: Error statistics for navigation solutions with and without vision-aiding.

Horizontal Error Statistics		
	no vision	vision
Linear distance [m]	38869	38869
RMS 3D Error [m]	136.04	106.57
RMS 3D Error per linear distance	0.35 %	0.27 %

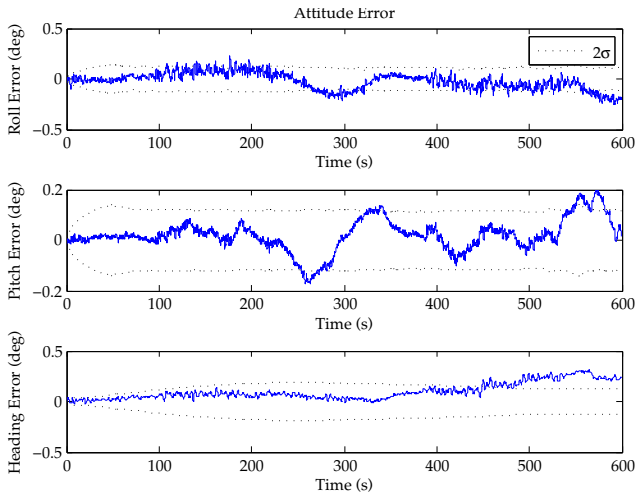


(a)

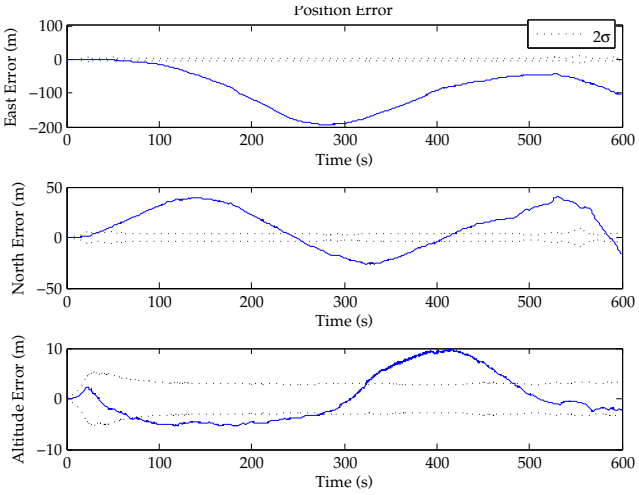


(b)

Figure 14: Attitude error (a) and position error (b) with 2σ bounds of navigation solution without vision aiding.



(a)



(b)

Figure 15: Attitude error (a) and position error (b) with 2σ bounds of navigation solution using modular vision aiding.

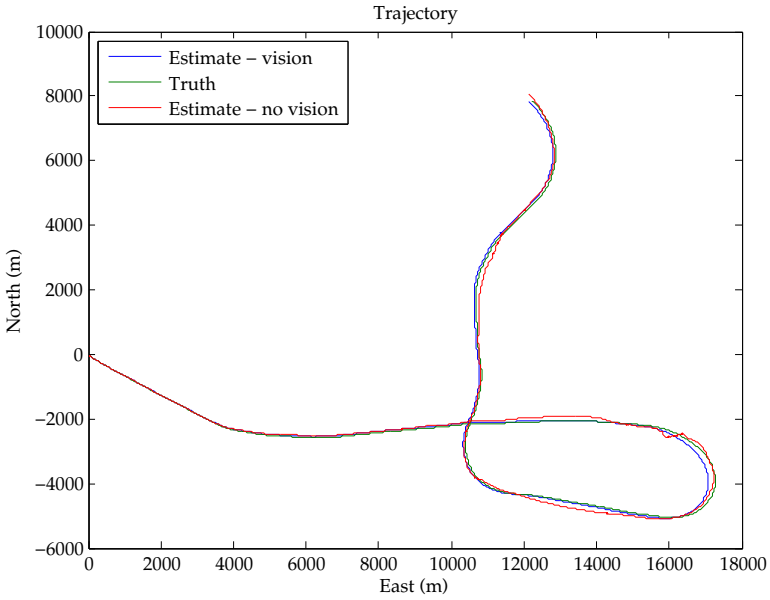


Figure 16: Trajectory of vehicle, with vision-aided and unaided navigation solutions.

CHAPTER V

VISION-AIDED NAVIGATION USING A FACTORED EXTENDED KALMAN FILTER

In this chapter, a state estimation algorithm is presented which captures the correlations between features and vehicle states. In contrast to the previous algorithm, a fully populated covariance matrix for all vehicle and features states is propagated and updated with new sensor information. It is shown that the primary benefit of this approach is a more accurate estimate of the uncertainty, which in turn improves its behavior in the presence of additional sensors. By accurately estimating the growth of position uncertainty as features are added and removed from the state vector, absolute measurements of position, such as from a GPS measurement, will appropriately move the vehicle estimate, the feature estimates, and their associated uncertainties. When feature and vehicle correlations are not tracked, an absolute position update of the vehicle will not modify the feature states, and the poor vehicle covariance estimate can cause slow convergence of the vehicle to the true estimate.

The algorithm is implemented within a Bierman-Thornton EKF (BTEKF) formulation [17], which maintains the covariance of the estimator in a factored form for improved numerical stability. The BTEKF formulation is of importance in this application because the relative nature of vision-aided estimation without known features may cause covariances with large condition number over large time intervals and distances.

5.1 State Vector Definition

For the fully correlated estimation algorithm, the vehicle and feature states are defined as described in Ch. 2. The dynamic model and measurement model are also defined as in Ch. 2. The state vector of the system is composed of the vehicle state and the feature states:

$$\hat{x} = \begin{bmatrix} \hat{x}_v & \hat{p}_f \end{bmatrix}^{\mathrm{T}} \in R^{N_v + N_{f_s} N_f} \tag{101}$$

$$= \begin{bmatrix} \hat{p}^i & \hat{v}^i & \hat{R} & \hat{s}_b & \hat{\omega}_b & \hat{p}_{f_1} & \dots & \hat{p}_{f_{N_f}} \end{bmatrix}^{\mathrm{T}} \tag{102}$$

where N_v and N_f are the number of vehicle states and features, respectively, and N_{f_s} is the number of states to describe a single feature. The covariance of the state vector is given by $P \in R^{(N_v + N_{f_s} N_f) \times (N_v + N_{f_s} N_f)}$, though it is not explicitly tracked in the BTEKF.

5.2 Bierman-Thornton Extended Kalman Filter

The Bierman-Thornton EKF is a modification of the standard EKF formulation which maintains the estimator covariance in a modified Cholesky factorization. It is composed of the Bierman propagation equations and Thornton update equations. Efficient algorithms for computing the resulting covariance factors are available [17].

5.2.1 Thornton Propagation Equations

The BTEKF stores the covariance factors U and D, where U is an upper triangular matrix with unit diagonal entries, and D is a diagonal matrix. The factors relate to the covariance matrix by

$$P = UDU^{\mathrm{T}}. \tag{103}$$

The Thornton propagation equations for timestep $k-1$ to k are given below. Let Q be the process noise and Φ be the state transition matrix for the discrete dynamic

system. Then the propagated covariance is given by

$$P_k(-) = \Phi_{k-1} P_{k-1}(+) \Phi_{k-1}^{\mathrm{T}} + Q_{k-1}. \tag{104}$$

Rewriting this in terms of factors U and D gives

$$U_k(-) D_k(-) U_k(-)^{\mathrm{T}} = \begin{bmatrix} \Phi_{k-1} U_{k-1}(+) & I \end{bmatrix} \begin{bmatrix} D_{k-1}(+) & 0 \\ 0 & Q_{k-1} \end{bmatrix} \begin{bmatrix} U_{k-1}(+)^{\mathrm{T}} \Phi_{k-1}^{\mathrm{T}} \\ I \end{bmatrix}. \tag{105}$$

Factoring terms on the right side,

$$\begin{bmatrix} U_{k-1}(+)^{\mathrm{T}} \Phi_{k-1}^{\mathrm{T}} \\ I \end{bmatrix} = BL \tag{106}$$

where B is an orthogonal matrix and L is lower triangular. Comparing the result to the left side of Eq. (105) gives

$$U_k(-) = L^{\mathrm{T}} \tag{107}$$

$$D_k(-) = B^{\mathrm{T}} \begin{bmatrix} D_{k-1}(+) & 0 \\ 0 & Q_{k-1} \end{bmatrix} B \tag{108}$$

The propagation of factors U and D is therefore a matrix factorization problem. Propagation of the state vector proceeds according to the nonlinear dynamic equations presented in Ch. 2.

5.2.2 Bierman Measurement Equations

The Bierman measurement update equations update the covariance factors to account for new measurement. The equations assume a scalar measurement. In standard EKF, the covariance update is given by

$$P(+) = P(-) - P(-) C^{\mathrm{T}} \left(C P(-) C^{\mathrm{T}} + R \right)^{-1} C P(-). \tag{109}$$

Rewriting in terms of U and D and dropping the $(-)$ for clarity gives

$$U(+)D(+)U(+)^{\mathrm{T}} \tag{110}$$

$$= UDU^{\mathrm{T}} - \frac{UDU^{\mathrm{T}}C^{\mathrm{T}}CUDU^{\mathrm{T}}}{CUDU^{\mathrm{T}}C^{\mathrm{T}} + R} \tag{111}$$

$$= U\left[D - \frac{DU^{\mathrm{T}}C^{\mathrm{T}}CUD}{CUDU^{\mathrm{T}}C^{\mathrm{T}} + R}\right]U^{\mathrm{T}}. \tag{112}$$

Now, consider the factor in brackets. If this can be factored into its modified Cholesky form,

$$D - \frac{DU^{\mathrm{T}}C^{\mathrm{T}}CUD}{CUDU^{\mathrm{T}}C^{\mathrm{T}} + R} = U_a D(+)U_a^{\mathrm{T}}, \tag{113}$$

where U_a is an upper triangular matrix, then inserting in to Eq. (112) gives

$$U(+)D(+)U(+)^{\mathrm{T}} = U\left[U_a D(+)U_a^{\mathrm{T}}\right]U^{\mathrm{T}} \tag{114}$$

$$= \left[UU_a\right]D(+)\left[UU_a\right]^{\mathrm{T}} \tag{115}$$

and it can be seen that the updated diagonal matrix $D(+)$, and $U(+)$ is given by

$$U(+) = UU_a \tag{116}$$

Additionally, the Kalman gain K is easily computed from $U(-)$ and $D(-)$ and used to update the state vector.

5.3 Marginalization and Initialization in the Bierman-Thornton EKF

The initialization and marginalization of features in the BTEKF is more difficult than the standard EKF formulation due to the factored form of the covariance. In the standard EKF, all terms contributing to the correlation of a particular state are contained in the row and column corresponding to its location in the state vector. However, it will be shown that in factored form, the values in diagonal matrix D contributes to all state correlations. In the following section, the initialization of a new feature and its covariance in the BTEKF is expressed as a factorization problem,

and this problem turns out to be of identical form to the Thornton propagation problem, allowing the same algorithm to be used.

Consider full covariance matrix P for state vector x. Since the initialization occurs during a single timestep k, the subscripts will be omitted. Let x be divided into three parts, and P divided correspondingly.

$$\hat{x} = \begin{bmatrix} \hat{x}_1 & \hat{x}_2 & \hat{x}_3 \end{bmatrix} \tag{117}$$

$$P = \begin{bmatrix} P_1 & P_{12} & P_{13} \\ P_{12}^{\mathrm{T}} & P_2 & P_{23} \\ P_{13}^{\mathrm{T}} & P_{23}^{\mathrm{T}} & P_3 \end{bmatrix} \tag{118}$$

by extension, the modified Cholesky factors are given by

$$U = \left[\begin{array}{c|c|c} U_1 & U_{12} & U_{13} \\ \hline 0 & U_2 & U_{23} \\ \hline 0 & 0 & U_3 \end{array}\right], \quad D = \left[\begin{array}{c|c|c} D_1 & 0 & 0 \\ \hline 0 & D_2 & 0 \\ \hline 0 & 0 & D_3 \end{array}\right] \tag{119}$$

Consider the case where we wish to marginalize and initialize a subset of states, x_2, in the middle of the state vector. We wish to initialize the state vector $x_{new}^{\mathrm{T}} = \begin{bmatrix} x_1^{\mathrm{T}}, & x_{2_{new}}^{\mathrm{T}}, & x_3^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$. Typically, the desired new feature is known in terms of a measurement and possibly a prior.

$$x_{2_{new}} = g(x_1, x_{2_{init}}, x_3) \tag{120}$$

In the particular case of a feature, the function $g(\cdot)$ would represent the inverse camera model, and $x_{2_{init}}$ the feature location in the image, (u, v) and a depth prior, d. Associated with the initialization state $x_{2_{init}}$ is an uncertainty covariance R_{init}. We can now define a state covariance in terms of the marginalized state covariance and R_{init}.

$$P_0 = \begin{bmatrix} P_1 & 0 & P_{13} \\ 0 & 0 & 0 \\ P_{13} & 0 & P_3 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & R_{init} & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{121}$$

Written in terms of covariance factors U and D,

$$P_0 = \left[\begin{array}{c|c|c} U_1 & U_{12} & U_{13} \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & U_3 \end{array}\right] \left[\begin{array}{c|c|c} D_1 & 0 & 0 \\ \hline 0 & D_2 & 0 \\ \hline 0 & 0 & D_3 \end{array}\right] \left[\begin{array}{c|c|c} U_1 & U_{12} & U_{13} \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & U_3 \end{array}\right]^{\mathrm{T}} + \left[\begin{array}{c|c|c} 0 & 0 & 0 \\ \hline 0 & R_{init} & 0 \\ \hline 0 & 0 & 0 \end{array}\right] \tag{122}$$

$$= \bar{U} D \bar{U}^{\mathrm{T}} + \bar{R}_{init} \tag{123}$$

The covariance matrix P_0 corresponds to state vector $x_{init} = \begin{bmatrix} x_1^{\mathrm{T}}, & x_{2_{init}}^{\mathrm{T}}, & x_3^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$. To find the covariance corresponding to x_{new}, the Jacobian of the state transformation is found:

$$J = \frac{\partial x_{new}}{\partial x_{init}} = \left[\begin{array}{c|c|c} I & 0 & 0 \\ \hline \frac{\partial x_{2_{new}}}{\partial x_1} & \frac{\partial x_{2_{new}}}{\partial x_{2_{init}}} & \frac{\partial x_{2_{new}}}{\partial x_3} \\ \hline 0 & 0 & I \end{array}\right] \tag{124}$$

Then the updated covariance factors can be found with the same process as the covariance propagation, described in Section 5.2.1.

$$U_{new} D_{new} U_{new}^{\mathrm{T}} = J \left(\bar{U} D \bar{U}^{\mathrm{T}} + \bar{R}_{init} \right) J^{\mathrm{T}} \tag{125}$$

$$= \bar{J} \bar{D} \bar{J} \tag{126}$$

$$= \underbrace{L^{\mathrm{T}}}_{=U_{new}} \underbrace{B^{\mathrm{T}} \bar{D} B}_{=D_{new}} L \tag{127}$$

where $\bar{J} = \begin{bmatrix} J\bar{U}, & J \end{bmatrix}$, $\bar{D} = diag(D, \bar{R}_{init})$, and B and L are the QR factors of $\bar{J}^{\mathrm{T}}$. The matrices J and $\bar{R}_{init}$ may be created for all initializing features, and the factorization can occur once. This method of feature initialization allows the use of the BTEKF implementation without costly reconstruction and re-factorization of the covariance from the factors.

5.4 Results

This section presents the simulation and flight test results for the fully correlated vision-aided navigation system described in this chapter. First is presented the results

of a Monte Carlo simulation study, which demonstrates the benefit of the Bierman-Thornton EKF visual SLAM implementation over standard and Joseph-form. Next, simulation and flight test results are presented for the GTMax platform, which demonstrate the improved numerical stability, the low position drift, and validate the algorithm in a closed-loop test. Finally, results of an implementation of the system on a small quadrotor are presented.

The GTMax platform and model are used in several of the presented results, and is essentially the same as described in Chapter 3. The most significant difference is the use of a single computer based on the Intel i7 mobile processor. The single computer runs two processes, one for guidance, navigation and control and another for feature and descriptor extraction. Image features and descriptors are generated using SIFT features [31] or a modified Harris corner detector. The simulation platform is described in Chapter 3.

5.4.1 Monte Carlo Analysis of UD and Standard EKF

This section describes a series of simulations comparing the numerical consistency of Bierman-Thornton EKF, Joseph-form EKF, and standard EKF. The simulation scenario in these tests is a downward-facing camera moving above a surface and looking at a grid of feature points. The filters estimate the camera position, attitude and velocity as well as feature locations. Each filter is run with identical input data and the results are compared over a series of progressively more poorly conditioned filtering problems.

Two simulation series are performed: one with a high assumed measurement uncertainty, and another with an low assumed measurement uncertainty. The high measurement uncertainty assumes a camera resolution typically found on today's real-time slam algorithms, and measurement accuracy of extracted feature points to pixel accuracy. The low measurement uncertainty assumes a much larger camera

resolution, approaching the largest resolution easily available on the market today. The low measurement uncertainty cases also assume perfect knowledge of the best linearization point, to exclude failures caused by linearization. The low measurement uncertainty series shows what a filtering algorithm might have to handle from a speculative future sensing platform.

5.4.1.1 Filtering Setup

The standard EKF and Joseph-form EKF are both implemented using the equations outlined in Section 2.1. The state integration was performed using a Adams-Bashforth method, which is similar to the two stage Runge-Kutta.

Measurement updates for all filters are performed using sequential scalar updates. The measurement model assumes a feature noise covariance R, though no noise is applied to the feature point measurements in the simulation. The plant model given by $f(x)$ is a simple propagation of inertial measurement unit data. The IMU noise is drawn from a Gaussian distribution for each of the Monte Carlo runs, but is identical for each filter. The feature correspondence is known.

In the high measurement noise case, the measurement Jacobian C is defined as usual for an EKF,

$$C = \frac{\partial h(x)}{\partial x}_{x=\hat{x}_{k|k-1}} \tag{128}$$

However, it has been shown in Hesch et al. [20] and previous works that the linearization of the Jacobian is often a source of filter overconfidence and inconsistency. Errors in the estimate, and hence the linearization point of the Jacobian, contribute fictitious information to the filter and change the observability properties of the system. In an ideal world, the Jacobians would be evaluated at the true system state, not the state estimate. Of course, this information is not generally available. Instead, the updates and propagation can be constrained such that observability properties match the true observability characteristics of the nonlinear system.

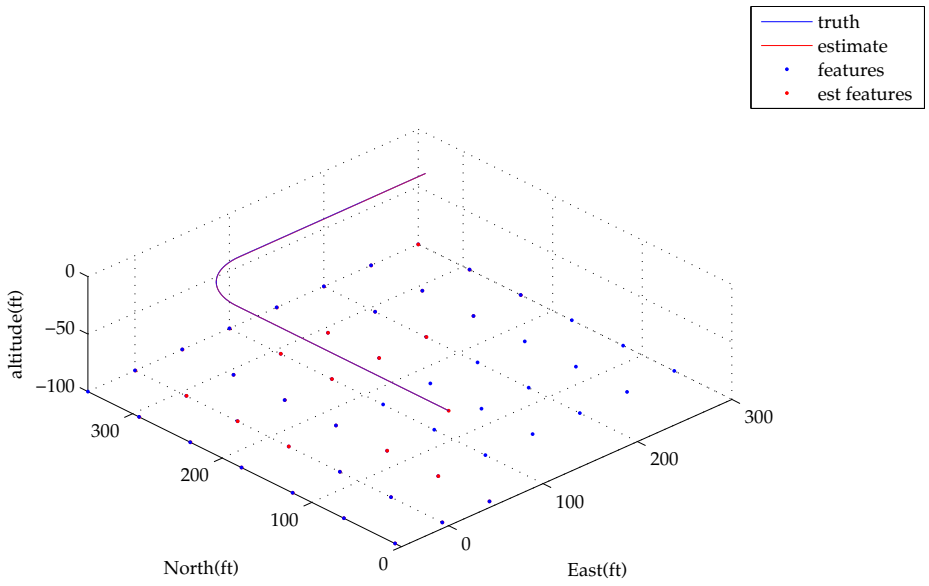


Figure 17: Example trajectory and solution for a single simulation run.

Since observability is not our main focus, we have approximated the technique of Hesch et al. by evaluating our Jacobians at the true state for the low measurement uncertainty series of Monte Carlo tests:

$$C_{ideal} = \frac{\partial h(x)}{\partial x}_{x=x_{k-1}} \tag{129}$$

This allows us to show a best case scenario, where the observability properties are ideal, and the linearization is the best possible.

5.4.1.2 Test environment

The simulated camera follows a trajectory at an altitude of 100 ft over a grid of points. The grid and camera trajectory are shown in Figure 17, along with the navigation result from a single run. The propagation takes place at 100 Hz, and the camera updates at 20 Hz.

For each series, test points were chosen over a range of initial position covariances. At each test point, ten Monte Carlo simulations were run with different noise vectors drawn from the same Gaussian distribution function. The UD filter and standard filter

were run on identical noise vectors and the results recorded. Table 2 summarizes the results of the tests.

Table 2: Summary of results from Monte Carlo comparison of EKF implementations.

Test Series	Position covariance at which failure first occurs, for each EKF implementation		
	Standard	Joseph-form	Proposed
High R: $R = 10^{-6}$	1×10^{11}	$1 \times 10^{10.5}$	$1 \times 10^{12.5}$
Low R: $R = 10^{-10}$	1×10^{6}	1×10^{6}	1×10^{9}

5.4.1.3 Test with high measurement covariance

The results of the high measurement covariance test are presented in Figure 18. A criterion for the failure of the navigation solution was chosen to be an RMS estimation error greater than 1 ft, approximately an order of magnitude greater than the best run. Failure was typically associated with very large RMS error, on the order of 10^3 ft or more. The plots show the percent of the ten Monte Carlo runs that were successful. It can be seen that both the standard and Joseph-form filters deteriorate at approximately the same P_0 value. The Bierman-Thornton EKF remains stable up to P_0 value of approximately 2 orders of magnitude greater.

5.4.1.4 Test with low measurement covariance

The results of the test are presented in Figure 19. Each plot line indicates the percent of successful runs out of the 10 Monte Carlo runs taken for a given initial position covariance, P_0, indicated along the x axis. Note that the values for initial position covariance are shifted down with respect to Figure 18, because failure occurs sooner as the measurement covariance decreases. This indicates that in the future, numerical stability will become more important. Again, it is clearly seen that as the conditioning worsens, the standard EKF and Joseph-form fail earlier than the UD filter, by about three orders of magnitude in initial covariance.

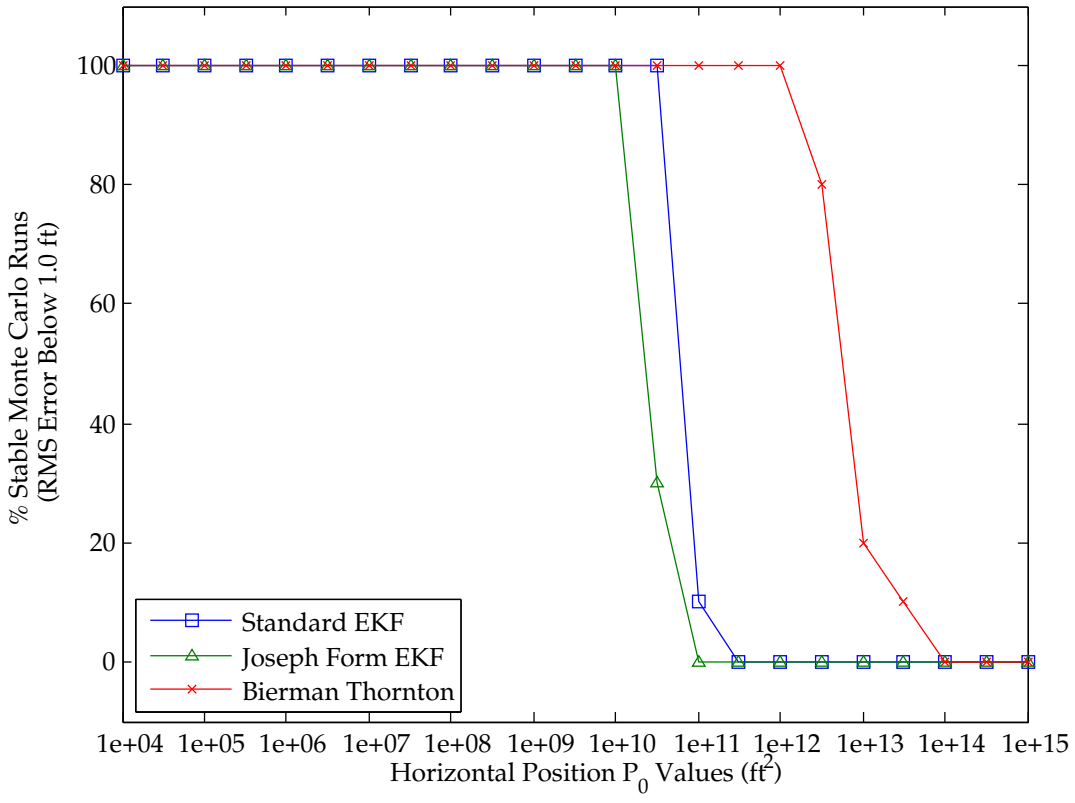


Figure 18: Percent of successful Monte Carlo runs for each initial condition, for the standard EKF, Joseph-form EKF, and the Bierman-Thornton EKF. Measurement standard deviation of .001 of image width.

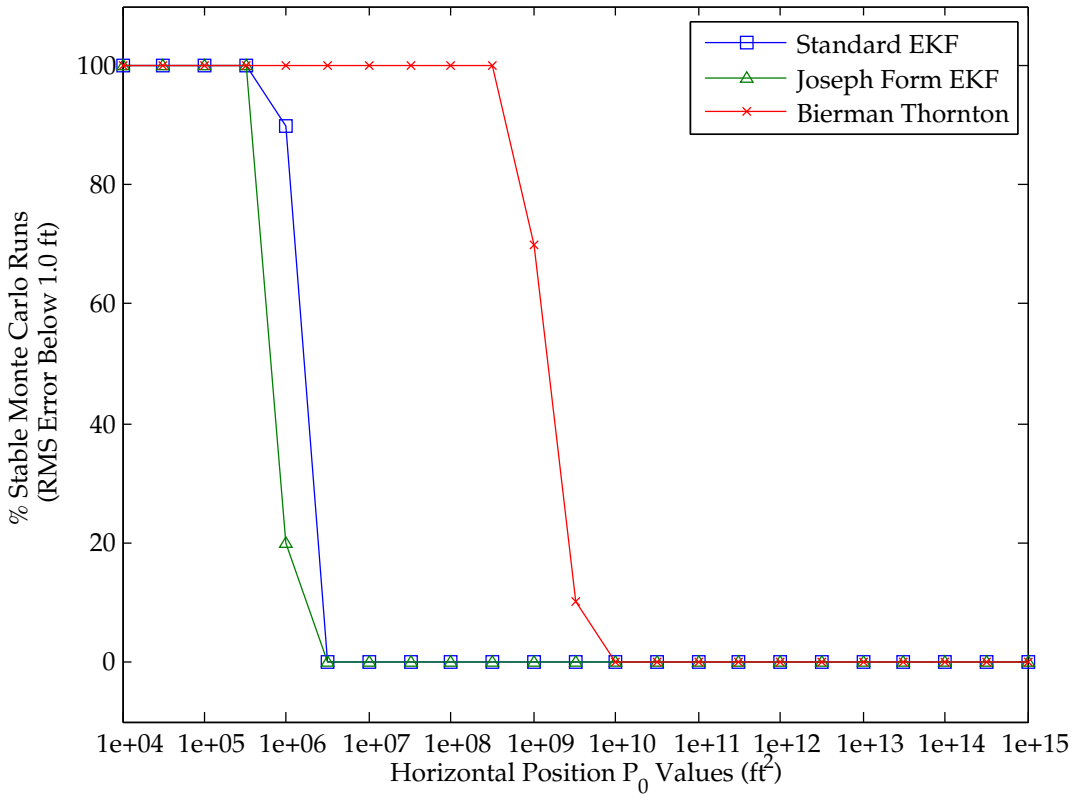


Figure 19: Percent of successful Monte Carlo runs for each initial condition, for the standard EKF, Joseph-form EKF, and the Bierman-Thornton EKF. Measurement standard deviation of .00001 of image width.

5.4.2 Demonstration of Improved Numerical Stability in GUST

To demonstrate the improved numerical properties in a flight-ready executable, the BTEKF was compared to a standard EKF in GUST. The standard EKF was implemented as described in Section 2.1. A state vector of 15 vehicle states and 16 features was used, giving a full state vector of 63 states in the form described in Chapter 5. The filter was initialized with a diagonal matrix P_0, and a diagonal process noise Q was used. The initial covariance is given in Table 3. Camera and magnetometer sensors were used.

Table 3: Initial covariance of the state vector.

Initial Covariance		
ϕ, θ	2×10^{-9}	rad^2
ψ	0.1	rad^2
p_x, p_y, p_x	5×10^{11}	ft^2
v_x, v_y, v_x	0	$(\text{ft/s})^2$
a_{bx}, a_{by}	1	$(\text{ft/s}^2)^2$
a_{bz}	2×10^{-6}	$(\text{ft/s}^2)^2$
ω_b	2×10^{-8}	$(\text{rad/s})^2$
Features	Computed	

Figures 20 and 21 illustrate the operation of the filter, during initialization until 5 seconds, and then in full operation afterwards. Figure 20 shows the root-mean-squared position error for the two filters. It can be seen that the unfactored standard EKF implementation quickly diverges about 5 seconds after the end of the initialization routine, while the factored BTEKF remains accurate. Similarly, in Figure 21 the state variances of the unfactored standard EKF quickly become unreasonable, whereas the factored BTEKF is stable throughout.

The failure of the standard EKF is directly related to numerical problems which do not occur in the BTEKF. Also, the standard EKF fails at an initial position covariance that agrees with the Monte Carlo study presented in Section 5.4.1.

The average computation time for the covariance propagation was measured over

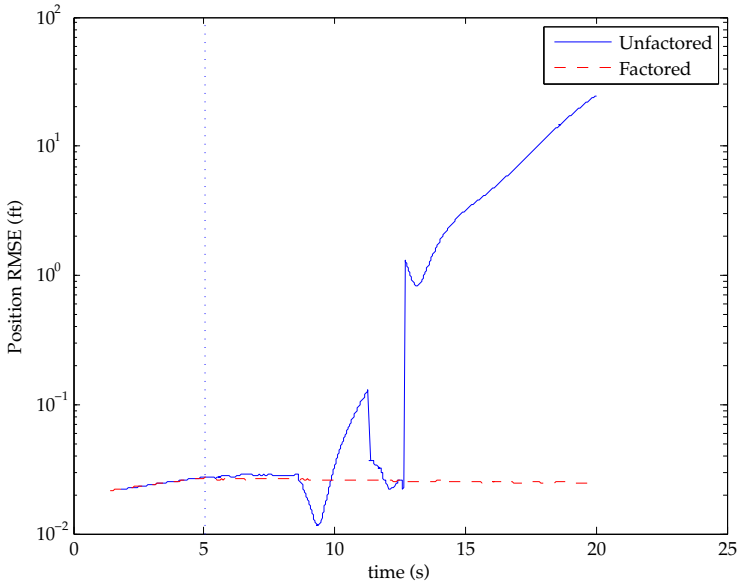


Figure 20: Position RMS error for the standard EKF (unfactored) and the BTEKF (factored). Filter initialization ends at 5 seconds.

60 s of filter operation for both implementations. The simulation was run on a desktop computer with a Core i7 processor. The results are presented in Table 4. The BTEKF propagation was found to be on average slightly more efficient at propagating the covariance than the standard EKF. It should be noted that no special effort was made to make the standard EKF efficient, and that a more in-depth comparison would take greater advantage of sparsity in both the standard EKF and BTEKF algorithms. However, since many implementations of the standard EKF are in precisely this form, it is still a useful comparison.

Table 4: Average computation time over 60 s of operation for one covariance propagation for BTEKF and standard EKF.

Propagation Time	
BTEKF	0.00211 s
Standard EKF	0.00262 s

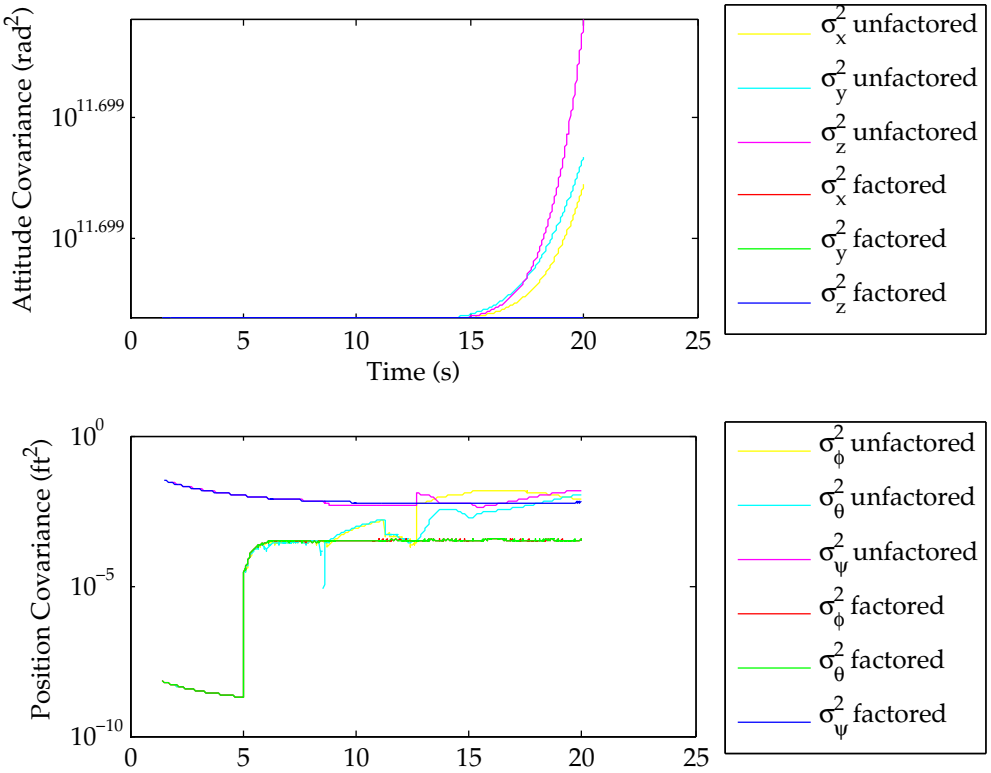


Figure 21: Position and attitude covariance for the standard EKF (unfactored) and the BTEKF (factored). Filter initialization ends at 5 seconds.

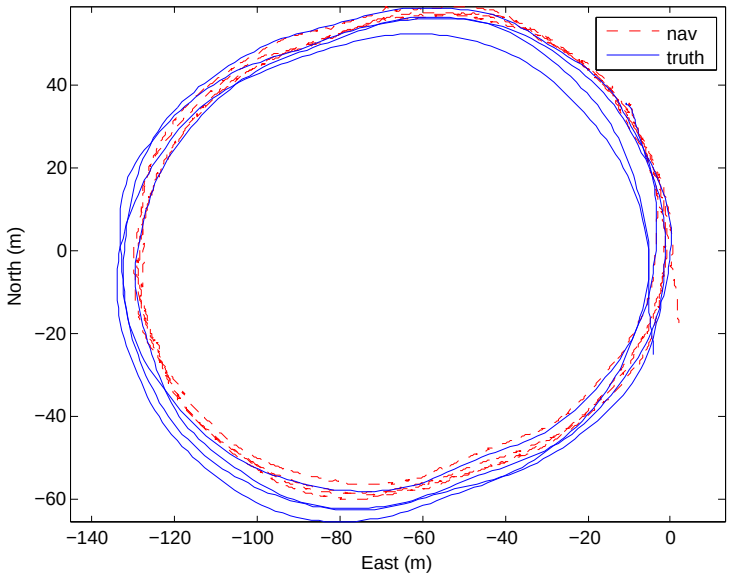


Figure 22: Horizontal position of vehicle as given by vision-based navigation position states (red dashed line) and simulation truth data (blue solid line) of the GTMax during a simulated flight of an oval trajectory. The total distance flown was approximately 1600 m.

5.4.3 Simulated Navigation Performance

The BTEKF was evaluated in simulation flying an oval trajectory. The GTMax flew four laps of an oval trajectory at 30 ft/s velocity and an altitude of 100 ft. The BTEKF vision-aided navigation solution was used in the controller loop in real time. Vision data was simulated by overlaying satellite maps on the simulation ground, and capturing images from rendered graphics from the camera location. SIFT features[31] were used.

Figures 22, 23, and 24 show the results from the simulation. Figures 23 and 24 show the error between the simulation truth states and the navigation output. Also shown are 2σ uncertainty bounds. Is is apparent that the navigation solution remains consistent with the uncertainty throughout the test. Figure 22 shows the trajectory of the vehicle and the navigation estimate. Table 5 shows some performance statistics from the simulated flight.

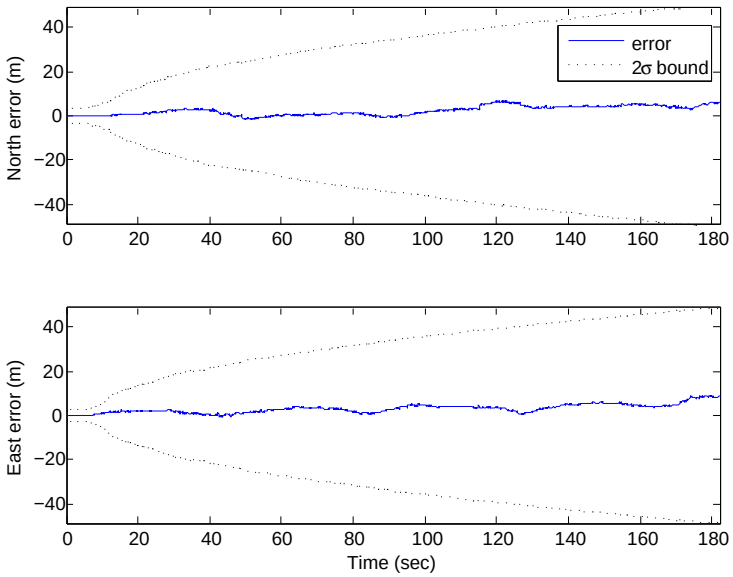


Figure 23: Horizontal position error and 2σ-covariance of vision-aided navigation system of the GTMax during a simulated flight of an oval trajectory. The final horiontal position error was 9.7 m.

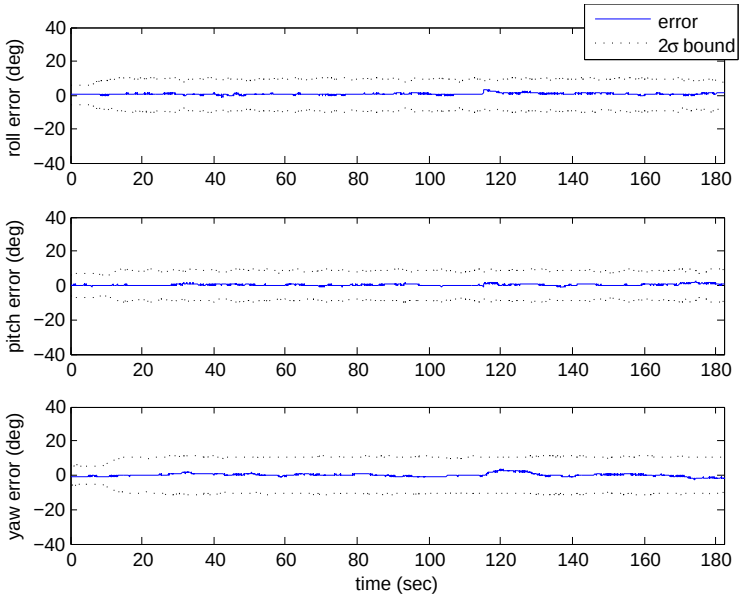


Figure 24: Attitude error and 2σ-covariance of vision-aided navigation system of the GTMax during a simulated flight of an oval trajectory.

Table 5: Horizontal error statistics for the simulated oval trajectory using vision-aided navigation.

Horizontal Error Statistics	
Distance [m]	1575
RMS Error [m]	5.5
RMS Error per linear distance	0.35 %
Final Error [m]	9.7
Final Error per linear distance	0.61 %

5.4.4 Flight Test Results

The BTEKF SLAM navigation system was flight tested on the GTMax platform in three separate tests. The navigation system provided input to the vehicle controller, which tracked an oval trajectory defined by an operator. The navigation system was operated with a controller in the loop, and important validation criteria because of the highly non-linear nature of the SLAM navigation.

The sensors available to the navigation system were as follows: camera, capturing images at 57.66 fps and 320×240 resolution, magnetometer at 10 Hz, IMU at 100 Hz. Also, the absolute altitude of the vehicle was measured either with a barometric pressure sensor or was simulated with differential GPS altitude above a datum. Horizontal differential GPS was recorded for comparison, but was not used in the navigation solution. Harris corner features were used, and an 11×11 pixel window around each feature was used as a descriptor, as described in Section 3.3. Figure 25 shows examples of images from the camera used during the tests.

The quantitative results from three separate tests are shown in Table 6. The results from test 1 and 2 were taken with GPS altitude in the solution, and test 3 used pressure altitude. The error and % error is low and in agreement across all tests.

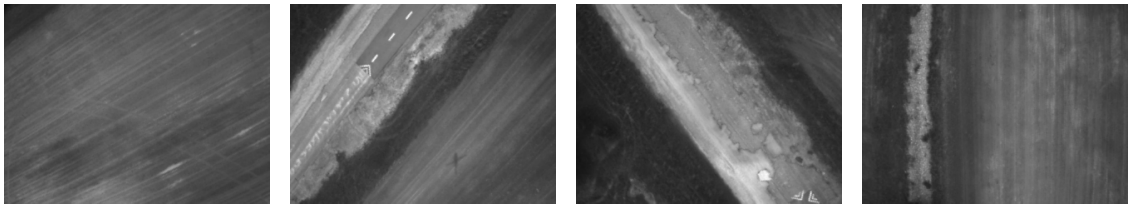
Figures 26, 27, and 28 show results from Test 3. An annotated video of the test can be found at `http://uav.ae.gatech.edu/videos/g150317a1_noGPS.mp4`. Figure 26 shows a image of the planned trajectory as seen from the ground station

during the test. Figure 27 shows the navigation solution along with the GPS sensor data. The trajectory of the vehicle agrees with the GPS sensor data in general with a moderate amount of drift in the solution. Also the results are very similar to what was predicted in simulation in Section 5.4.3 Figure 28 shows the horizontal error plots between the navigation solution and the GPS data, as well as the 2σ error covariance. The altitude of the vehicle above the datum is shown for reference. The error is shown to be consistent with the covariance.

Table 6: GTMax Flight test results for oval trajectory.

GTMax Flight Test Results			
	Test 1 (GPS Altitude)	Test 2 (GPS Altitude)	Test 3 (baro Altitude)
Time [s]	200	155	120
Linear Distance [m]	1163	951	768
Horizontal RMS Error [m]	5.51	5.51	3.09
RMS Error per linear distance	0.47 %	0.58 %	0.40 %
Final Error [m]	3.88	6.35	3.37
Final Error per linear distance	0.33 %	0.67 %	0.44 %
Video	hyperlink[1]		hyperlink[2]

[1] `http://uav.ae.gatech.edu/videos/g140624_visionBased_oval.mp4`
[2] `http://uav.ae.gatech.edu/videos/g150317a1_noGPS.mp4`



(a) (b) (c) (d)

Figure 25: Example images from the camera, showing typical image texture during the flight test.

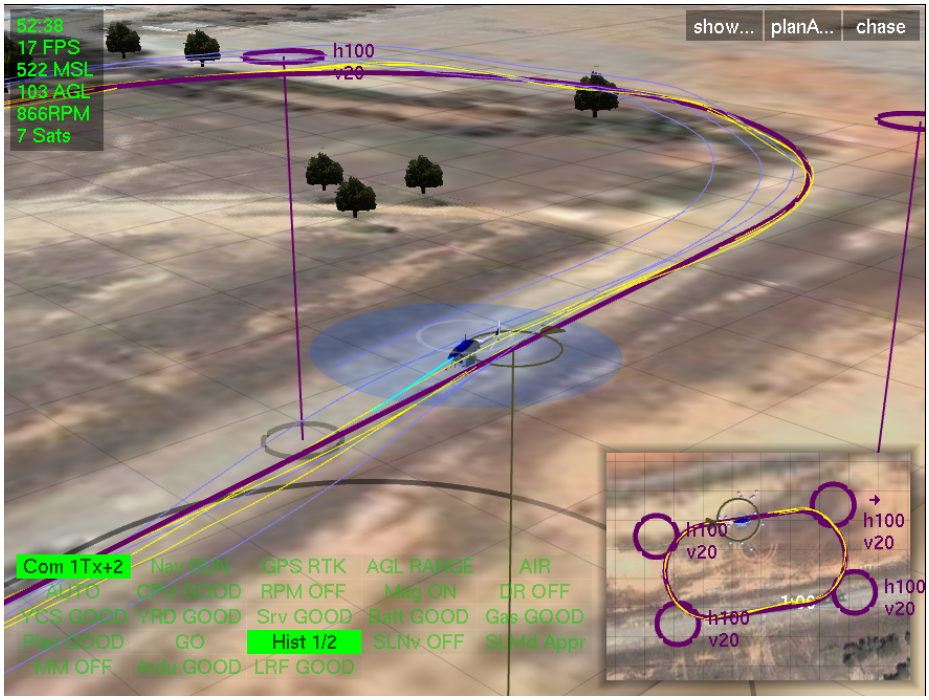


Figure 26: Image from ground control station during flight of oval trajectory. Yellow trace shows the navigation solution. Blue trace shows the GPS data. Purple line indicates the commanded trajectory.

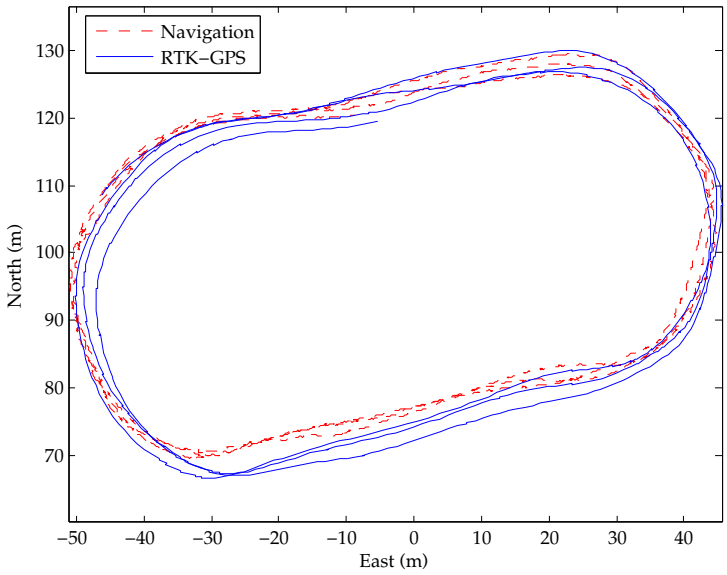


Figure 27: Horizontal navigation solution and differential GPS data for autonomous flight with controller in the loop.

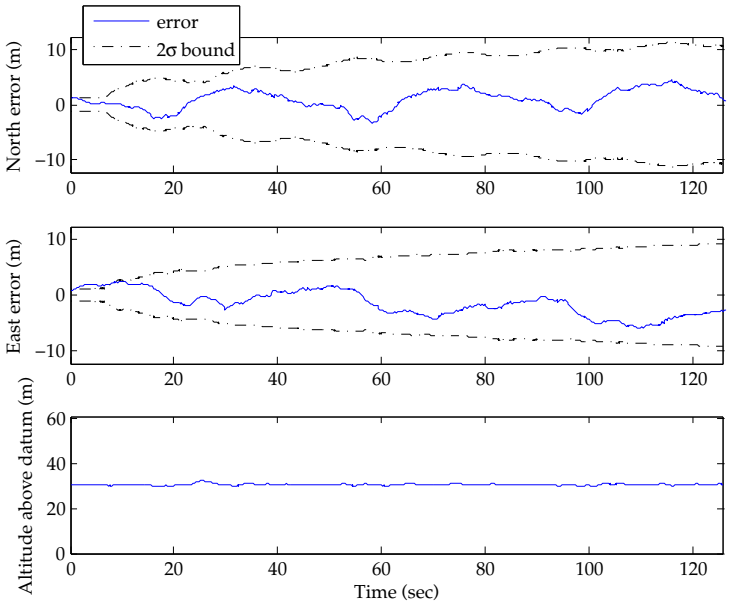


Figure 28: Horizontal position error of navigation solution from GPS truth for autonomous flight with controller in the loop. Altitude is shown for reference.

5.4.5 Miniature Quadrotor Flight Tests

The vision-aided navigation system presented here was also implemented on a small quadrotor platform. The implementation highlights the use of the vision-aided system on a lightweight vehicle with significant payload constraints. The payload constraints limit the amount of onboard processing power and sensors available to the navigation system.

The quadrotor is equipped with three sensors: A MEMS-grade IMU, a monochrome camera, and a sonar. The MEMS-grade IMU is an InvenSense MPU-6050[1], and provides filtered specific force and angular velocity measurements at 100 Hz. The camera is a Point Grey[2] Firefly MV, capturing monochome images in 320×240 resolution at 70 Hz. The sonar is a Maxbotix[3] LV-MaxSonar-EZ4, reporting at 10 Hz with a tested

[1] `http://www.invensense.com`
[2] `http://www.ptgrey.com/`
[3] `http://www.maxbotix.com/`

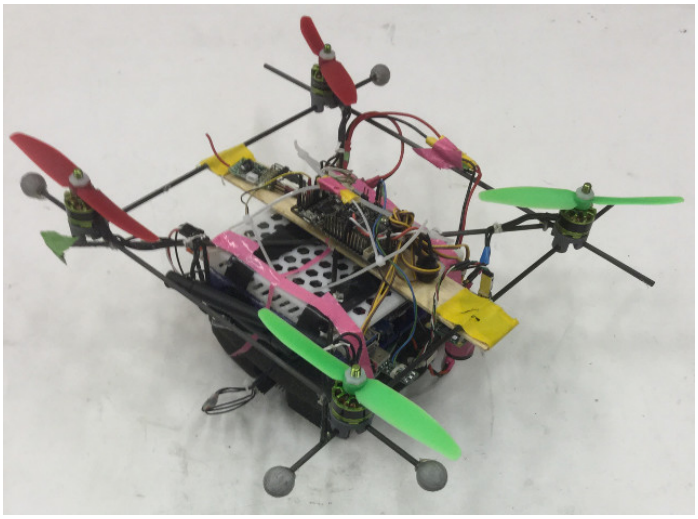


Figure 29: The GTQMini, a quadrotor equipped with an IMU, sonar and camera and a Core i7 based computer weighing less than 600 g.

operational range from 20 cm to over 400 cm. The sensor data is processed onboard the vehicle on a GigaByte[4] Brix BXi7-4500U single board computer with an Intel[5] Core i7-4500U 1.8GHz/3.0GHz processor. The entire vehicle weighs less than 600 g. Figure 29 shows a picture of the vehicle.

The GTQMini was tested in an indoor environment with a motion capture system as ground truth. The vehicle was tasked to fly over 4 laps of an oval trajectory of approximately 1 m by 2 m. The trajectory is illustrated in Figure 30, along with the navigation solution and the commanded trajectory. It is clear from the figure that the majority of the error occurs due to drift in yaw. This is further confirmed in Figure 31. The position error remains within the 2σ error bounds as expected. The yaw error, however, drifts beyond the expected error. This is likely due to the spurious information gain caused by linearizations in the EKF, as identified in Hesch et al. [20]. A similar modification to the one proposed there could be applied.

A second test of the vehicle was carried out to test the behavior of the navigation solution over longer trajectories. Due to the environment, no ground truth data

[4] `http://www.gigabyte.us/`
[5] `http://www.intel.com/`

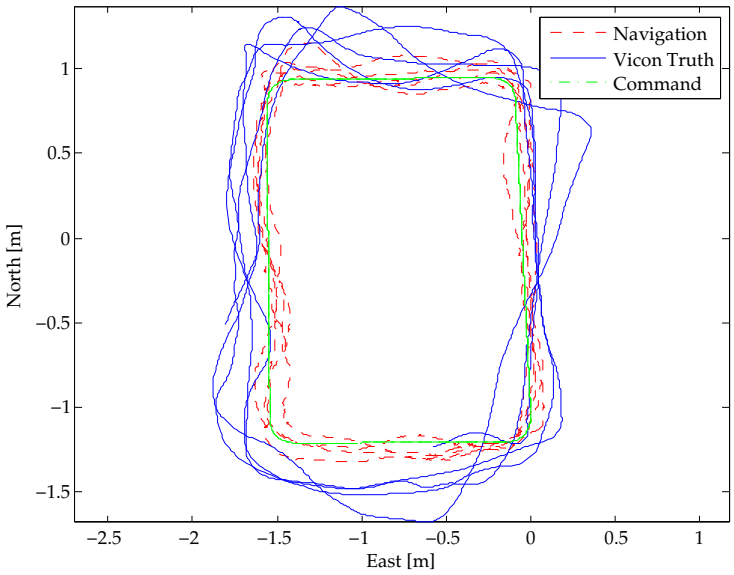


Figure 30: GTQMini trajectory during four laps of a small oval. The majority of the navigation error is caused by yaw drift.

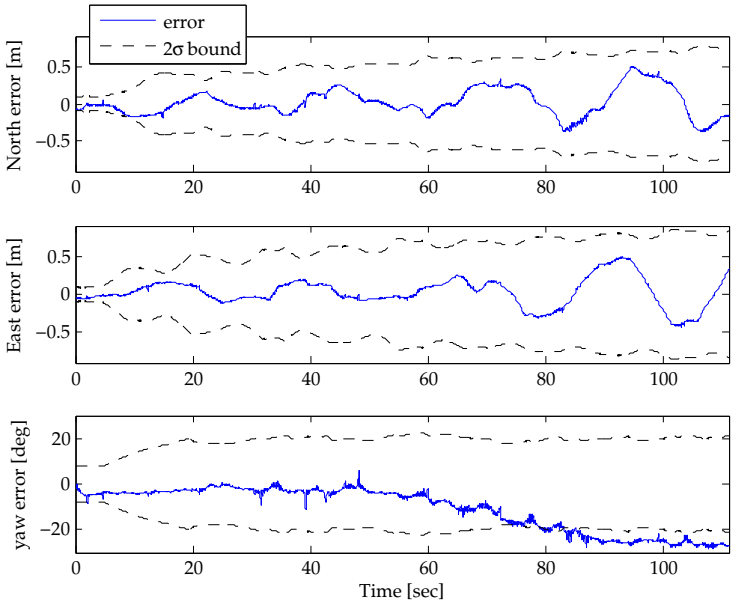


Figure 31: GTQMini error in the unobservable modes of the system, horizontal position and yaw. The horizontal position remains within the 2σ error bounds, but yaw does not.

was available throughout the flight. Instead, the final position was measured and compared to the navigation solution. The trajectory was flown at a speed of 0.9 m/s (3 ft/s) and an altitude of 0.9 m (3 ft). The vehicle was commanded to fly out and back to a waypoint 10 m in front of the vehicle. This trajectory was flown twice, without removing the error between runs. The final Euclidian distance from the starting location after the first and second circuits were 0.6 m and 1.2 m, giving an average error accumulation of 0.6 m per 20 m round trip. Figure 32 shows a frame from the flight test. A video of the flight test can be found at `http://youtu.be/GGqexQy-FgE`.

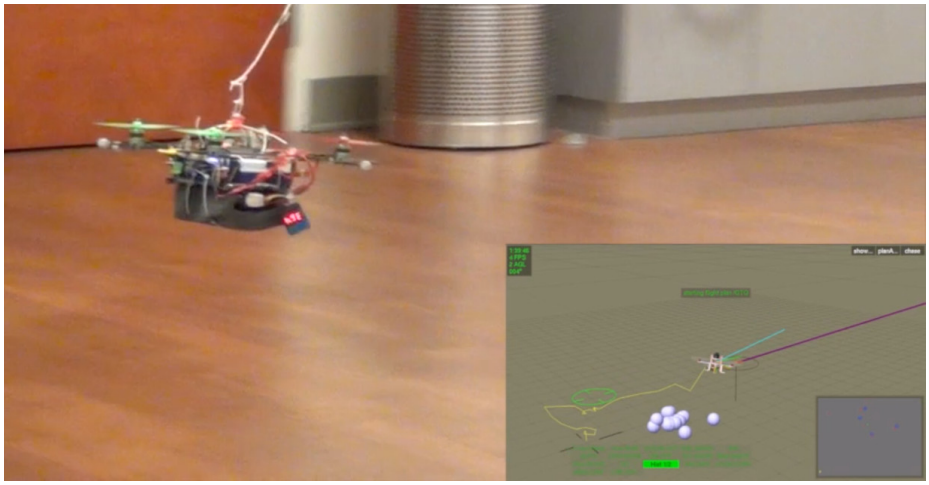


Figure 32: Frame from visual SLAM flight test. Both the flight and the view on the Ground Control Station (GCS) are displayed. The white spheres in the GCS window show the matched features used for the navigation system. A video of the flight test can be found at `http://youtu.be/GGqexQy-FgE`.

CHAPTER VI

CONTRIBUTIONS AND FUTURE RESEARCH

6.1 Contributions

This thesis has presented an uncorrelated V-INS system and evaluated its performance with flight testing. A novel modular implementation of this algorithm was presented which allows integration with a variety of filters. Finally, an algorithm using fully correlated feature and vehicle states was developed, based on the Bierman-Thornton factored EKF. The BTEKF algorithm in particular highlighted the numerical stability of the approach, and illustrated the benefit of accurately accounting for the covariance of vehicle and feature states.

In particular, the following contributions were made:

i) Development of a practical EKF-based V-INS which ignores feature-vehicle correlations. A vision algorithm with low computational requirements is demonstrated that is capable of stabilizing the helicopter over long periods and long trajectories. Correlations between features and vehicle state are ignored, which degrades consistency but maintains usable estimates and low computational burden. Key improvements to feature correspondence and database management are described.

ii) Development and application of a theorem allowing the integration of Kalman filter updates with limited shared information. The theorem describes a novel way of combining two separate filters such that the combination is equivalent to a single standard implementation. This allows greater flexibility in integration of filter modules. The demonstration of it in a modular filter architecture validates the approach.

iii) Development of a novel implementation of a fully correlated V-INS in a factored EKF formulation, with efficient feature marginalization and initialization. The use of fully correlated feature and vehicle states dramatically improves the consistency of the filter, especially in position. This improvement allows better integration with additional absolute sensors which might be found on a UAV, such as GPS. The improved consistency also allows the position uncertainty to grow arbitrarily large over time, as should be the case in a relative navigation framework. The implementation in a factored framework improves the numerical stability of the filter even in the presence of the large position uncertainty, and does so with minimal additional computational load compared with a standard EKF implementation.

iv) Monte Carlo analysis of the numerical stability of the proposed algorithm, and demonstration in flight code. Numerous tests were performed to quantify the benefits of the factored EKF. The Monte Carlo analysis demonstrates a 2 to 3 order of magnitude improvement in the size of the position uncertainty over the standard and Joseph-form EKF. In addition, the demonstration in the flight code shows that this improvement carries over to real-world scenarios. Finally, the recorded propagation times show that the factored framework poses minimal additional computational burden over a naive standard implementation, agreeing with the literature.

v) Simulation and flight test results showing the performance of the navigation algorithms with a controller in the loop on multiple vehicles, indoors and outdoors. Simulations were performed to validate all proposed algorithms. The navigation algorithms were operated both open loop (in the case of the modular filter in Chapter 4) and, more importantly, with a controller in the loop. The use of a controller in the loop is an important validation criteria due to the nonlinearity

of the estimation problem and the potential for controller-navigation coupling. Flight test results were presented for three extended flights of the GTMax vehicle. The navigation system had less than 1% RMS error in all tests. Finally the same navigation system was implemented on a small quadrotor vehicle and demonstrated in indoor flight.

6.2 Conclusions

This thesis has described several algorithms for performing monocular vision-aided inertial navigation on a UAV. This research advances the state of the art by improving the reliability and flexibility of V-INS. Reliable GPS-denied navigation will open new frontiers of applications for UAVs, allowing them to take on more tasks that are dull, difficult, dangerous, or impossible for human actors.

Several conclusions can be drawn from the results of this thesis. In particular, the contrast between the algorithm with ignored correlations, described in Chapter 3 and the algorithm with full correlations, described in Chapter 5, illustrates the roll of correlations in the navigation system. Correlations between features and vehicle states capture the relative nature of the navigation system. The navigation system estimates the vehicle state relative to the features, but no absolute position information is gained from feature measurements. It is the correlations between features and vehicles which account for this fact.

Marginalization and initialization in the BTEKF was shown to have a convenient form. The form allows for efficient implementation of adding and removing features within the visual SLAM system, but it is not limited to this application. The form may be leveraged for modifying the state and covariance vector for whatever purpose.

The results of the Monte Carlo analysis of numerical stability illustrates the importance of the use of stable algorithms. This is especially true as sensing platforms improve and measurement covariances decrease.

Finally, the practicality of vision-aided inertial navigation for UAVs is demonstrated in a variety of real-world scenarios. The contributions of this thesis increases the ease of implementation on practical systems, improve the robustness of visual navigation, and improves the compatibility with other sensors. Immediate use of information to update both features and vehicle eliminate delay in creation of map and exploration.

6.3 Future Work

There are topics that are considered promising future directions to pursue.

i) Evaluation of alternate filtering techniques. The EKF is one among many nonlinear filtering paradigms, and it is by no means guaranteed to be the best for the V-INS problem. In fact, there are good reasons to believe that, due to the highly nonlinear measurement function, that other filter designs or nonlinear optimization techniques may be better suited to the problem. The sigma-point EKF in particular is of interest, and has been used successfully in loosely-coupled V-INS systems in the past[3]. Also, comparison to other numerically stable methods such as the square root information filter and square root filter would be beneficial.

ii) Extension of marginalization and initialization techniques to other decompositions and applications. The ideas used in processing the marginalization and initialization of features can be applied generally to modifications of the U and D factors in the Bierman-Thornton EKF. Extensions for combining operations, such as initialization and propagation, may yield some computational benefit. It may also be possible to extend these results to other factorizations, such as the Potter square root filter or the square root information filter.

iii) Application of observability constraints to the BTEKF estimator. It has been

shown [21, 20, 35] using various observability analysis techniques that V-INS systems are not fully observable, and that designing the estimator to account for this fact can improve estimator consistency. Implementation on a UAV with a controller in the loop would validate these advantages on a typical system.

iv) Investigation of batch methods such as bundle adjustment. Full bundle adjustment, using all locations and feature observations, provides the optimal solution to the SLAM problem but is typically computationally intractable for real-time systems[47]. Methods have been proposed to approximate the full bundle adjustment problem by using key frames and performing the mapping in non real-time [27, 51], or to perform bundle adjustment in real time with one keyframe and an uncertainty prior [13]. Applying non-linear optimization techniques has the potential to improve consistency and reduce error.

v) Investigation of techniques to improve V-INS over long time periods and distances. A variety of methods for sub-map joining and optimization have been proposed in the computer vision community [13, 56], but few have been implemented on aerial vehicles or vehicles with controllers in the loop. Incorporating a secondary optimization over a collection of submaps and poses could improve accuracy of the navigation system, allow for loop closures, and generate maps of the environment which would aid in mission tasks.

The research goals presented here will significantly advance the state of the art in monocular vision-aided inertial navigation for UAVs.

REFERENCES

[1] Bierman, G. J. and Thornton, C. L., "Numerical comparison of kalman filter algorithms: Orbit determination case study," *Automatica*, vol. 13, no. 1, pp. 23 – 35, 1977. 6

[2] Castellanos, J. A., Tardos, J., and Schmidt, G., "Building a global map of the environment of a mobile robot: the importance of correlations," in *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 2, pp. 1053–1059 vol.2, Apr 1997. 6

[3] Chambers, A., Scherer, S., Yoder, L., Jain, S., Nuske, S., and Singh, S., "Robust multi-sensor fusion for micro aerial vehicle navigation in gps-degraded/denied environments," in *American Control Conference (ACC), 2014*, pp. 1892–1899, June 2014. 10, 78

[4] Chiu, H.-P., Das, A., Miller, P., Samarasekera, S., and Kumar, R., "Precise vision-aided aerial navigation," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pp. 688–695, Sept 2014. 11

[5] Chowdhary, G., Johnson, E. N., Magree, D., Wu, A., and Shein, A., "Gps-denied indoor and outdoor monocular vision aided navigation and control of unmanned aircraft," *Journal of Field Robotics*, vol. 30, no. 3, pp. 415–438, 2013. 11, 34

[6] Civera, J., Davison, A., and Montiel, J., "Inverse depth parametrization for monocular slam," *Robotics, IEEE Transactions on*, vol. 24, pp. 932 –945, oct. 2008. 4, 19, 20

[7] Civera, J., Grasa, O. G., Davison, A. J., and Montiel, J. M. M., "1-point ransac for extended kalman filtering: Application to real-time structure from motion and visual odometry," *Journal of Field Robotics*, vol. 27, no. 5, pp. 609–631, 2010. 4

[8] Corke, P., Lobo, J., and Dias, J., "An introduction to inertial and visual sensing," *The International Journal of Robotics Research*, vol. 26, no. 6, pp. 519–535, 2007. 5

[9] Davison, A., "Real-time simultaneous localisation and mapping with a single camera," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pp. 1403 –1410 vol.2, oct. 2003. 4, 30

[10] Davison, A., Reid, I., Molton, N., and Stasse, O., "Monoslam: Real-time single camera slam," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, pp. 1052 –1067, June 2007. 4, 30

[11] Dellaert, F. and Kaess, M., "Square root sam: Simultaneous localization and mapping via square root information smoothing," *The International Journal of Robotics Research*, vol. 25, no. 12, pp. 1181–1203, 2006. 7

[12] Eade, E. and Drummond, T., "Scalable monocular slam," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1, pp. 469 – 476, june 2006. 5, 22

[13] Eade, E. and Drummond, T., "Monocular slam as a graph of coalesced observations," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pp. 1–8, Oct 2007. 79

[14] Faessler, M., Fontana, F., Forster, C., Mueggler, E., Pizzoli, M., and Scaramuzza, D., "Autonomous, vision-based flight and live dense 3d mapping with a quadrotor micro aerial vehicle," *Journal of Field Robotics*, pp. n/a–n/a, 2015. 9

[15] Forster, C., Pizzoli, M., and Scaramuzza, D., "Svo: Fast semi-direct monocular visual odometry," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pp. 15–22, May 2014. 9

[16] Gelb, A., *Applied Optimal Estimation*. MIT Press, 1974. 13, 15

[17] Grewal, M. S. and Andrews, A. P., *Kalman Filtering: Theory and Practice Using MATLAB*. John Wiley and Sons, Inc., 2002. 6, 13, 51, 52

[18] Harris, C. and Stephens, M., "A combined corner and edge detector," in *Alvey vision conference*, vol. 15, pp. 147–151, Manchester, UK, 1988. 26

[19] Heng, L., Honegger, D., Lee, G. H., Meier, L., Tanskanen, P., Fraundorfer, F., and Pollefeys, M., "Autonomous visual mapping and exploration with a micro aerial vehicle," *Journal of Field Robotics*, vol. 31, no. 4, pp. 654–675, 2014. 9, 10

[20] Hesch, J., Kottas, D., Bowman, S., and Roumeliotis, S., "Consistency analysis and improvement of vision-aided inertial navigation," *Robotics, IEEE Transactions on*, vol. 30, pp. 158–176, Feb 2014. 6, 58, 72, 78

[21] Hesch, J. A., Kottas, D. G., Bowman, S. L., and Roumeliotis, S. I., "Camera-imu-based localization: Observability analysis and consistency improvement," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 182–201, 2014. 6, 78

[22] Honegger, D., Meier, L., Tanskanen, P., and Pollefeys, M., "An open source and open hardware embedded metric optical flow cmos camera for indoor and outdoor applications," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 1736–1741, May 2013. 5

[23] Huang, G. P., Mourikis, A. I., and Roumeliotis, S. I., "Observability-based rules for designing consistent ekf slam estimators," *The International Journal of Robotics Research*, vol. 29, no. 5, pp. 502–528, 2010. 6

[24] Indelman, V., Williams, S., Kaess, M., and Dellaert, F., "Factor graph based incremental smoothing in inertial navigation systems," in *Information Fusion (FUSION), 2012 15th International Conference on*, pp. 2154–2161, July 2012. 5, 9

[25] Johnson, E. N. and Kannan, S. K., "Adaptive Trajectory Control for Autonomous Helicopters," *Journal of Guidance, Control, and Dynamics*, vol. 28, pp. 524–538, May 2005. 31

[26] Johnson, E. N. and Schrage, D. P., "System integration and operation of a research unmanned aerial vehicle," *AIAA Journal of Aerospace Computing, Information and Communication*, vol. 1, pp. 5–18, Jan 2004. 31

[27] Klein, G. and Murray, D., "Parallel tracking and mapping for small ar workspaces," in *Mixed and Augmented Reality, 2007. 6th IEEE and ACM International Symposium on*, pp. 225 –234, nov. 2007. 5, 8, 79

[28] Langelaan, J., "State estimation for autonomous flight in cluttered environments," *AIAA Journal of Guidance Navigation and Control*, vol. 30, no. 5, pp. 1414–1426, 2007. 8

[29] Lefferts, E. J., Markley, F. L., and Shuster, M., "Kalman filtering for spacecraft attitude estimation," *Journal of Guidance, Control, and Dynamics*, vol. 5, no. 5, 1982. 18

[30] Leishman, R. C., McLain, T. W., and Beard, R. W., "Relative navigation approach for vision-based aerial gps-denied navigation," *Journal of Intelligent & Robotic Systems*, vol. 74, no. 1-2, pp. 97–111, 2014. 7

[31] Lowe, D., "Object recognition from local scale-invariant features," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 2, pp. 1150 –1157 vol.2, 1999. 57, 66

[32] Ma, Y., Soatto, S., Kosecka, J., and Sastry, S., *An Invitation to 3D Vision: From Images to Geometric Models*. Springer Verlag, 2003. 18

[33] Magree, D. P. and Johnson, E. N., "Performance of a monocular vision-aided inertial navigation system for a small uav," in *AIAA Guidance, Navigation, and Control (GNC) Conference*, American Institute of Aeronautics and Astronautics, August 2013. 8

[34] Magree, D. P. and Johnson, E. N., "Monocular vision-aided inertial navigation with improved numerical stability," in *AIAA Guidance, Navigation, and Control (GNC) Conference*, American Institute of Aeronautics and Astronautics, January 2015. 5, 7

[35] Martinelli, A., "Vision and imu data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination," *Robotics, IEEE Transactions on*, vol. 28, pp. 44–60, Feb 2012. 6, 78

[36] Montemerlo, M., Thrun, S., Koller, D., and Wegbreit, B., "Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges," in *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, 2003. 22

[37] Mostegel, C., Wendel, A., and Bischof, H., "Active monocular localization: Towards autonomous monocular exploration for multirotor mavs," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pp. 3848–3855, May 2014. 9

[38] Scherer, S., Rehder, J., Achar, S., Cover, H., Chambers, A., Nuske, S., and Singh, S., "River mapping from a flying robot: state estimation, river detection, and obstacle mapping," *Autonomous Robots*, vol. 33, pp. 189–214, 2012. 10.1007/s10514-012-9293-0. 10

[39] Schmid, K., Ruess, F., and Burschka, D., "Local reference filter for life-long vision aided inertial navigation," in *Information Fusion (FUSION), 2014 17th International Conference on*, pp. 1–8, July 2014. 7

[40] Schmid, K., Lutz, P., Tomi, T., Mair, E., and Hirschmller, H., "Autonomous vision-based micro air vehicle for indoor and outdoor navigation," *Journal of Field Robotics*, pp. n/a–n/a, 2014. 9

[41] Shen, S., Mulgaonkar, Y., Michael, N., and Kumar, V., "Vision-based state estimation and trajectory control towards high-speed flight with a quadrotor," *Proceedings of Robotics: Science and Systems IX*, 2013. 5, 10

[42] Sol, J., Vidal-Calleja, T., Civera, J., and Montiel, J., "Impact of landmark parametrization on monocular ekf-slam with points and lines," *International Journal of Computer Vision*, vol. 97, no. 3, pp. 339–368, 2012. 19, 21

[43] Stevens, B. L. and Lewis, F. L., *Aircraft Control and Simulation*. John Wiley and Sons, 2003. 17

[44] Strasdat, H., Montiel, J. M. M., and Davison, A., "Real-time monocular slam: Why filter?," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 2657–2664, May 2010. 5

[45] Thornton, C. and Bierman, G., "Givens transformation techniques for kalman filtering," *Acta Astronautica*, vol. 4, no. 78, pp. 847 – 863, 1977. 6

[46] Thrun, S., Burgard, W., and Fox, D., *Probabilistic robotics*. MIT Press, 2005. 28

[47] Triggs, B., McLauchlan, P. F., Hartley, R. I., and Fitzgibbon, A. W., "Bundle adjustment a modern synthesis," in *Vision Algorithms: Theory and Practice* (Triggs, B., Zisserman, A., and Szeliski, R., eds.), vol. 1883 of *Lecture Notes in Computer Science*, pp. 298–372, Springer Berlin Heidelberg, 2000. 79

[48] van Dalen, G., Magree, D. P., and Johnson, E. N., "Absolute localization using image alignment and particle filtering," in *Intelligent Robots and Systems (IROS 2015), 2015 IEEE/RSJ International Conference on*, SUBMITTED 2015. 11

[49] Veth, M. J., *Fusion of Imaging and Inertial Sensors for Navigation.* PhD thesis, Air Force Institute of Technology, Wright-Patterson AFB, OH, Sept 2006. 8

[50] Watanabe, Y., Calise, A. J., and Johnson, E. N., "Vision-Based Obstacle Avoidance for UAVs," in *AIAA Guidance, Navigation, and Control Conference*, no. August, pp. 1–11, 2007. 8

[51] Weiss, S., Achtelik, M. W., Lynen, S., Achtelik, M. C., Kneip, L., Chli, M., and Siegwart, R., "Monocular vision for long-term micro aerial vehicle state estimation: A compendium," *Journal of Field Robotics*, vol. 30, no. 5, pp. 803–831, 2013. 5, 8, 79

[52] Wu, A., *Vision-bsed navigation and mapping for flight in GPS-denied environments.* PhD thesis, Georgia Institute of Technology, School of Aerospace Engineering, Atlanta, GA 30332, dec 2010. 11, 18, 27

[53] Wu, A. and Johnson, E. N., "Autonomous flight in gps-denied environments using monocular vision and inertial sensors," in *Infotech@AIAA*, (Atlanta, GA), AIAA, April 2010. 11

[54] Wu, A. D., Johnson, E. N., Kaess, M., and Dellaert, F., "Autonomous Flight in GPS-Denied Environments Using Monocular Vision and Inertial Sensors," *Journal of Aerospace Information Systems*, vol. 10, pp. 172–186, Apr. 2013. 11

[55] Wu, A. D., Johnson, E. N., and Proctor, A. A., "Vision-Aided Inertial Navigation for Flight Control," *AIAA journal of aerospace computing, information, and communication*, vol. 2, no. September, pp. 348–360, 2005. 8

[56] Zhao, L., Huang, S., and Dissanayake, G., "Linear slam: A linear solution to the feature-based and pose graph slam based on submap joining," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pp. 24–30, Nov 2013. 79