AIAA SciTech
5-9 January 2015, Kissimmee, Florida
AIAA Guidance, Navigation, and Control Conference

A Monocular Vision-aided Inertial Navigation System with Improved Numerical Stability

Daniel Magree*, Eric N. Johnson †

This paper develops a monocular vision-aided inertial navigation system based on the factored extended Kalman filter (EKF) proposed by Bierman and Thornton. The simultaneous localization and mapping (SLAM) algorithm measurement update and propagation steps are formulated in terms of the factored covariance matrix $P = UDU^{\mathrm{T}}$, and a novel method for efficiently adding and removing features from the covariance factors is presented. The system is compared to the standard EKF formulation in navigation performance and computational requirements. The proposed method is shown to improve numerical stability with minimal impact on computational requirements. Flight test results are presented which demonstrate navigation performance with a controller in the loop.

I. Introduction

This paper describes a novel vision-aided inertial navigation system (VINS) based on the factored EKF proposed by Bierman and Thornton. The Bierman-Thornton EKF (BTEKF), sometimes known as a UD filter, was developed to improve the numerical stability of the Kalman filter without sacrificing computational performance. This allows the filter to handle poorly conditioned problems with greater reliability than standard EKF implementations. The standard EKF is a widely used tool for aircraft state estimation and visual-inertial navigation in particular. Many visual slam algorithms are based on an underlying EKF algorithm for state and covariance estimation. However, its use in vision-aided navigation is problematic due to the possibility of poor conditioning of the covariance. The visual slam problem is not fully observable, so the state covariance can grow without bound, leading to poorly conditioned operations within the filter. Improving the numerical stability with the BTEKF expands the operational envelope of the system and improves reliability.

The VINS system described here is similar to the familiar EKF-based simultaneous localization and mapping (EKF-SLAM) systems. The state vector is composed of vehicle states and feature locations, and a covariance matrix and state vector are estimated from measurements and model propagation. Our system differs by storing the covariance matrix in factored form, and performs measurement updates, propagations, and feature initialization and removal by modifying the factored covariance matrix. Like many other systems, the number of stored features is limited to ensure the filter operates in constant time.

Next, an overview of the related literature is presented. Section III presents the proposed BTEKF navigation filter. Section IV presents a simulation study comparing the proposed filter to the standard EKF, both in performance and computational load. Close loop performance of the filter is demonstrated with flight test results. Finally, Section V presents conclusions.

*Graduate Research Assistant, Georgia Institute of Technology, dmagree@gatech.edu.
†Associate Professor, Georgia Institute of Technology, eric.johnson@ae.gatech.edu.

II. Related Work

Monocular camera-only SLAM has been a very active research area since the first real-time implementation was introduced by Davison in 2003.[1] The first system was based on the standard EKF, and since then variety of new paradigms have emerged. MonoSLAM,[2] and extentsion of Davison's original work, improved feature initialization and the motion model. Eade and Drummond[3] propose to treat the measurement update as a nonlinear estimation problem, analagous to bundle adjustment with one camera pose and a prior covariance, and build a set of submaps which are then optimized. Both of these systems operate on sparse image features. Klein and Murry[4] propose dividing the mapping and tracking steps into separate threads. This allows a full bundle adjustment to be carried out on a more dense point cloud in non-real time, while tracking continues on the existing map at frame rate.

The system proposed in this paper has most in common with MonoSLAM,[2] in that both use the Kalman filter to estimate a sparse set of features. This paper differs in the use of the factored covariance for improve numerical stability, and the use of inertial sensors for propagation of the motion model. Additionally, this paper looks at the problem in the context of unmanned vehicles, which have fast dynamics and a controller in the loop. This fact becomes relevant because the highly non-linear nature of the visual SLAM problem causes a failure of the separation principle, and the controller feedback loop may cause instability if only considered separately.[5]

Work on the application of VINS to aerospace systems is becoming more common. This is often driven by the need for independence or reduced reliance on GPS for unmanned vehicle systems (UAS). Vision aiding for obstacle avoidance,[6] and for vehicle localization[7] were early efforts in this area. More recently, the work by Weiss et al.[8,9] adapts the PTAM algorithm by Klein and Murray[4] for use on small quadrotor vehicles with an inertial measurement unit as the only additional sensor. They demonstrate highly capable systems with extended closed-loop flight test results. Shen et al.[10] develop a system which is fundamentally monocular, but uses a second camera for feature depth initialization. Other papers such as Leishman et al.[11] and Schmid et al.[12] demonstrate close-loop flight test results with systems using RGB-D vision and stereo vision respectively.

The UD filter used in this paper was first proposed by Bierman and Thornton.[13,14] 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.[15] They demonstrate that the UD filter performs accurately using single precision arithmetic where both the standard EKF and Joseph-stabilized filters fail. Additionally, the UD filter demonstrated computational load which was approximately 50% greater than the standard and Joseph-stabilized, and much less than the square-root filter.[16] 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.

This work builds on previous papers developing the vision-aided inertial navigation system. Wu et al. developed a method for fusing feature information from a monocular vision sensor in an extended Kalman filter framework.[17] 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.[18] The system presented here improves upon Chowdhary et al. by accounting for correlations between the vehicle and feature states, which improves accuracy and consistency, and by using the UD filter for the full system to improve numerical stability.

III. The Bierman-Thornton EKF SLAM Navigation System

This section presents an overview of the Bierman-Thornton visual SLAM navigation system. 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.

The state vector is composed of the vehicle postion, attitude, velocity and IMU bias states. Feature states are added and removed to the estimator, and only a limited number of features are stored to maintain constant time performance. The features are cartesian parameterized.

A. State Definition and Measurement Model

The vehicle model is based on the specific force and angular velocity input from a IMU. The non-linear dynamics of the vehicle are driven by raw IMU input, which is assumed to have a static or slowly evolving bias and corrupted by white Gaussian noise.

The vehicle state is given by the following vector:

$$\hat{x}_a = \begin{bmatrix} \hat{p}^i & \hat{v}^i & \hat{q}^i & \hat{s}_b & \hat{\omega}_b \end{bmatrix}^{\mathrm{T}} \tag{1}$$

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:

$$s_{raw} = a + s_b + L_{bi}g + \eta_a, \tag{2}$$

$$\omega_{raw} = \omega_t + \omega_b + \eta_\omega. \tag{3}$$

where a, and ω_t 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

$$s = s_{raw} - \hat{s}_b, \tag{4}$$

$$\omega = \omega_{raw} - \hat{\omega}_b. \tag{5}$$

The vehicle dynamics are given by the following:

$$\dot{\hat{p}}^i = v^i \tag{6}$$

$$\dot{\hat{v}}^i = L_{ib}s - g = L_{ib}(s_{raw} - \hat{s}_b) - g \tag{7}$$

$$\dot{\hat{q}}^i = \frac{1}{2}\mathcal{Q}(\omega)\hat{q}^i = \frac{1}{2}\mathcal{Q}(\omega_{raw} - \hat{\omega}_b)\hat{q}^i \tag{8}$$

$$\dot{\hat{s}}_b = 0 \tag{9}$$

$$\dot{\hat{\omega}}_b = 0 \tag{10}$$

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[19]

$$\mathcal{Q}([a_1 \quad a_2 \quad a_3]^{\mathrm{T}}) = \begin{bmatrix} 0 & -a_1 & -a_2 & -a_3 \\ a_1 & 0 & a_3 & -a_2 \\ a_2 & -a_3 & 0 & a_1 \\ a_3 & a_2 & -a_1 & 0 \end{bmatrix} \tag{11}$$

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.

$$\delta q = \begin{bmatrix} 1 & \hat{R} \end{bmatrix}^{\mathrm{T}} \tag{12}$$

such that

$$\delta q = \hat{q}_{ref}^{-1} \otimes \hat{q}. \tag{13}$$

Additional details on this formulation can be found in.[20,21]

The minimal vehicle state vector is

$$\hat{x}_v = \begin{bmatrix} \hat{p}^i & \hat{v}^i & \hat{R} & \hat{s}_b & \hat{\omega}_b \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{15} \tag{14}$$

The full state vector of the system is composed of the vehicle state and the feature states:

$$\hat{x} = \begin{bmatrix} \hat{x}_v & \hat{p}_f \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{15+3N_f} \tag{15}$$

$$= \begin{bmatrix} \hat{p}^i & \hat{v}^i & \hat{R} & \hat{s}_b & \hat{\omega}_b & \hat{p}_{f_1} & \dots & \hat{p}_{f_{N_f}} \end{bmatrix}^{\mathrm{T}} \tag{16}$$

where N_f is the number of feature states. The feature states are assumed to be static, and so the process model may be ignored, though a feature state process noise term Q_f may be applied. The covariance of the state vector is given by $P \in \mathbb{R}^{(15+3N_f)\times(15+3N_f)}$, though it not explicitly tracked in the BTEKF. The process noise of the filter is given by $Q = \mathrm{diag}(0, Q_a, Q_\omega, 0, 0, Q_f, \dots, Q_f)$, where diag indicates a block diagonal matrix of the arguments.

A pinhole camera model is used, and for simplicity, it will be assumed that the principle point is in the center of the image plane, and distortion has been compensated for. Additionally, we will treat the the image plane as if it is located in front of the optical center, and align the camera frame x-axis with the optical axis. Given the feature point location $r_f^c = p_f^c - p^c = [X, Y, Z]^{\mathrm{T}}$, the image plane location (u, v) is given by

$$z = \begin{bmatrix} u \\ v \end{bmatrix} = h(x) + \nu = \begin{bmatrix} f_u \frac{Y}{X} + \nu_u \\ f_v \frac{Z}{X} + \nu_v \end{bmatrix} \tag{17}$$

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 1 illustrates the relationship between the inertial, vehicle and camera reference frames.

B. 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 in Grewal and Andrews.[14]

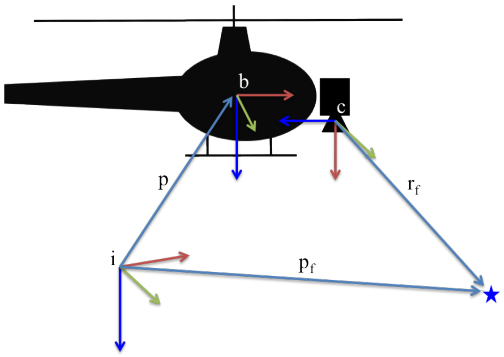


Figure 1. A schematic of the key reference frames used in this presentation: the inertial frame i, the vehicle body frame b, and the camera frame c. In general, the camera and body frame origins can be in different locations, but for simplicity of presentation it is assumed they are co-located.

1. *Thornton propagation equaations*

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

$$P = UDU^{\mathrm{T}}. \tag{18}$$

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

$$P_k(-) = \Phi_{k-1} P_{k-1}(+) \Phi_{k-1}^{\mathrm{T}} + Q_{k-1}. \tag{19}$$

Rewriting this in terms of factors U and D gives

$$U_k(-)D_k(-)U_k(-)^{\mathrm{T}} = \begin{bmatrix} \Phi_{k-1}U_{k-1}(+) & I \end{bmatrix} \begin{bmatrix} D_{k-1}(+) & 0 \\ 0 & Q_{k-1} \end{bmatrix} \begin{bmatrix} U_{k-1}(+)^{\mathrm{T}}\Phi_{k-1}^{\mathrm{T}} \\ I \end{bmatrix}. \tag{20}$$

Factoring terms on the right side,

$$\begin{bmatrix} U_{k-1}(+)^{\mathrm{T}}\Phi_{k-1}^{\mathrm{T}} \\ I \end{bmatrix} = BL \tag{21}$$

where B is an orthogonal matrix and L is lower triangular. Comparing the result to the left side:

$$U_k(-) = L^{\mathrm{T}} \tag{22}$$

$$D_k(-) = B^{\mathrm{T}} \begin{bmatrix} D_{k-1}(+) & 0 \\ 0 & Q_{k-1} \end{bmatrix} B \tag{23}$$

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 Section III.A.

2. *Bierman measurement equations*

The Bierman measurement update equations update the covariance factors to account for new measurement. The equations assume a scalar measurement. If the system has vector measurements, they can be processed

sequentially.[22] In standard EKF, the covariance update is given by

$$P(+) = P(-) - P(-)C^{\mathrm{T}}\left(CP(-)C^{\mathrm{T}} + R\right)^{-1} CP(-). \tag{24}$$

Where $C = \frac{\partial h(x)}{\partial x}_{x=\hat{x}}$ is the measurement Jacobian evaluated at the estimate $\hat{x}$. Rewriting in terms of U and D gives

$$U(+)D(+)U(+)^{\mathrm{T}} \tag{25}$$

$$= U(-)D(-)U(-)^{\mathrm{T}} - \frac{U(-)D(-)U(-)^{\mathrm{T}}C^{\mathrm{T}}CU(-)D(-)U(-)^{\mathrm{T}}}{CU(-)D(-)U(-)^{\mathrm{T}}} \tag{26}$$

$$= U(-)\left[D(-) - \frac{D(-)U(-)^{\mathrm{T}}C^{\mathrm{T}}CU(-)D(-)}{CU(-)D(-)U(-)^{\mathrm{T}}C^{\mathrm{T}} + R}\right]U(-)^{\mathrm{T}}. \tag{27}$$

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

$$D(-) - \frac{D(-)U(-)^{\mathrm{T}}C^{\mathrm{T}}CU(-)D(-)}{CU(-)D(-)U(-)^{\mathrm{T}}C^{\mathrm{T}} + R} = U_a D(+)U_a^{\mathrm{T}}, \tag{28}$$

where U_a is an upper triangular matrix, then inserting in to Eq. (27) gives

$$U(+)D(+)U(+)^{\mathrm{T}} = U(-)\left[U_a D(+)U_a^{\mathrm{T}}\right]U(-)^{\mathrm{T}} \tag{29}$$

$$= \left[U(-)U_a\right]D(+)\left[U(-)U_a\right]^{\mathrm{T}} \tag{30}$$

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

$$U(+) = U(-)U_a \tag{31}$$

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

C. Initialization of New Features

The initialization and removal 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, and the relationship between elements of U and the states is not so clear. 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 the state x full covariance matrix P. 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.

$$\hat{x} = \begin{bmatrix}\hat{x}_1^{\mathrm{T}} & \hat{x}_2^{\mathrm{T}} & \hat{x}_3^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}} \tag{32}$$

$$P = \begin{bmatrix} P_1 & P_{12} & P_{13} \\ P_{12}^{\mathrm{T}} & P_2 & P_{23} \\ P_{13}^{\mathrm{T}} & P_{23}^{\mathrm{T}} & P_3 \end{bmatrix} \tag{33}$$

by extension, the modified Cholesky factors are given by

$$U = \left[\begin{array}{c|c|c} U_1 & U_{12} & U_{13} \\ \hline 0 & U_2 & U_{23} \\ \hline 0 & 0 & U_3 \end{array}\right], \quad D = \left[\begin{array}{c|c|c} D_1 & 0 & 0 \\ \hline 0 & D_2 & 0 \\ \hline 0 & 0 & D_3 \end{array}\right] \tag{34}$$

Let us replace the state $\hat{x}_2$ and its correlation with a new state $\hat{x}_{2_{new}}$ and correlation. The new state, in this context the location of a feature in an image projected into 3D space, is a function of the current state and measurement. Inverting the camera model gives

$$\hat{x}_{2_{new}} = f(\hat{x}_1, \hat{x}_3, z, d) \tag{35}$$

$$= \hat{p}^i + dL_{ic}(\hat{q}^i, \hat{R}) \begin{bmatrix} 1 \\ u/f_u \\ v/f_v \end{bmatrix}, \tag{36}$$

where d is an *a priori* distance along the projection ray necessary for computing the initialization of the state. The expression for the new state vector is given by

$$\hat{x}_{new} = \begin{bmatrix} \hat{x}_1 \\ \hat{x}_{2_{new}} \\ \hat{x}_3 \end{bmatrix} = \begin{bmatrix} \hat{x}_1 \\ f(\hat{x}_1, \hat{x}_3, z, d) \\ \hat{x}_3 \end{bmatrix} \tag{37}$$

The initialization function is in general nonlinear, therefore to calculate the initialization covariance, the function is linearized:

$$\hat{x}_{new} \approx J \begin{bmatrix} \hat{x}_1^{\mathrm{T}} & z^{\mathrm{T}} & d & \hat{x}_3^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{38}$$

where

$$J = \frac{\partial \hat{x}_{new}}{\partial \left[\hat{x}_1^{\mathrm{T}}, z^{\mathrm{T}}, d, \hat{x}_3^{\mathrm{T}}\right]^{\mathrm{T}}} = \left[\begin{array}{ccccc|c|c} & & I_1 & & & 0 & 0 & 0 \\ \hline \frac{\partial \hat{x}_{2_{new}}}{\partial \hat{x}_v} & 0 & \dots & 0 & & \frac{\partial \hat{x}_{2_{new}}}{\partial z} & \frac{\partial \hat{x}_{new}}{\partial d} & 0 \\ \hline & & 0 & & & 0 & 0 & I_3 \end{array}\right] \tag{39}$$

Then, from the definition of covariance, the covariance of the new state vector can be found as a linear transformation of the covariance of $[\hat{x}_1, z, d, \hat{x}_3]^{\mathrm{T}}$, which is known.

$$P_{new} = J \left[\begin{array}{c|ccc|c} P_1 & 0 & 0 & P_{13} \\ \hline 0 & R & 0 & 0 \\ 0 & 0 & \sigma_d^2 & 0 \\ \hline P_{13}^{\mathrm{T}} & 0 & 0 & P_3 \end{array}\right] J^{\mathrm{T}} = J \left[\begin{array}{c|c|c} P_1 & 0 & P_{13} \\ \hline 0 & \bar{R} & 0 \\ \hline P_{13}^{\mathrm{T}} & 0 & P_3 \end{array}\right] J^{\mathrm{T}} \tag{40}$$

$$= J P_o J^{\mathrm{T}} \tag{41}$$

where σ_d is the standard deviation of the *a priori* distance d.

Next, recall that P is stored in its factored form U and D. Rewriting P_o in terms of these factors gives

$$P_o = \left[\begin{array}{c|c|c} U_1 & U_{12} & U_{13} \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & U_3 \end{array}\right] \left[\begin{array}{c|c|c} D_1 & 0 & 0 \\ \hline 0 & D_2 & 0 \\ \hline 0 & 0 & D_3 \end{array}\right] \left[\begin{array}{c|c|c} U_1 & U_{12} & U_{13} \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & U_3 \end{array}\right]^{\mathrm{T}} + \left[\begin{array}{c|c|c} 0 & 0 & 0 \\ \hline 0 & \bar{R} & 0 \\ \hline 0 & 0 & 0 \end{array}\right] \tag{42}$$

$$= \bar{U} D \bar{U}^{\mathrm{T}} + \bar{\bar{R}}. \tag{43}$$

This expression for P_o allows the new covariance to be written in the same form as the Thornton propagation.

$$P_{new} = J\left(\bar{U} D \bar{U}^{\mathrm{T}} + \bar{\bar{R}}\right) J^{\mathrm{T}} \tag{44}$$

$$= \begin{bmatrix} J\bar{U} & J \end{bmatrix} \begin{bmatrix} D & 0 \\ 0 & \bar{\bar{R}} \end{bmatrix} \begin{bmatrix} \bar{U}^{\mathrm{T}} J^{\mathrm{T}} \\ J^{\mathrm{T}} \end{bmatrix}, \tag{45}$$

and the calculation of the U and D factors of the new covariance then becomes a identical factorization problem to that described in Section III.B.1. Decomposing the right-most factor into orthogonal matrix B and lower-triangular matrix L,

$$\begin{bmatrix} U^{\mathrm{T}} J^{\mathrm{T}} \\ J^{\mathrm{T}} \end{bmatrix} = BL, \tag{46}$$

yields new U and D factors,

$$U_{new} = L^{\mathrm{T}} \tag{47}$$

$$D_{new} = B^{\mathrm{T}} \begin{bmatrix} D & 0 \\ 0 & \bar{\bar{R}} \end{bmatrix} B. \tag{48}$$

This method of feature initialization allows the use of the BTEKF implementation without costly reconstruction and re-factorization of the covariance from the factors. Additionally, multiple features can be initialized simultaneously by constructing P_o and J for multiple features. Then the factorization into B and L may be performed once for all features.

IV. Results

This section presents the simulation and flight test results for the fully correlated vision-aided navigation system described in this paper. Validation of the vision-based navigation system was conducted on the GTMax,[23] a 66 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 2 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 for vision-based navigation. A computer with an Intel i7 processor performs all flight-essential processing. When performing vision aided navigation, the computer runs two processes, one for guidance, navigation and control and another for feature and descriptor extraction. Image features and descriptors are generated using the SIFT algorithm,[24] as implemented in OpenCV [a]. 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).

[a] http://opencv.org/

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.

Figure 2. The GTMax helicopter weighs 66 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.

A. Comparison of BTEKF SLAM with Standard EKF SLAM

The purpose of the Bierman-Thornton EKF implementation is to improve the numerical stability of the update and propagation of the filter. To demonstrate the improved numerical properties, the BTEKF was compared to a standard EKF. The standard EKF was implemented using the following covariance propagation equations:

$$\Delta t = t_k - t_{k-1} \tag{49}$$

$$P_k = \Phi(t_k, t_{k-1}) P_{k-1} \Phi(t_k, t_{k-1})^{\mathrm{T}} + Q\Delta t \tag{50}$$

The state derivative given in equations (6-10) is propagated with a second order Runge-Kutta method. The measurement update equations are

$$P(+) = (I - KC)P(-) \tag{51}$$

$$\hat{x}(+) = \hat{x}(-) + K(z - h(\hat{x}(-))) \tag{52}$$

$$K = P(-)C^{\mathrm{T}}(CP(-)C^{\mathrm{T}} + R) \tag{53}$$

The implementation of the propagation equation performed two $N \times N$ matrix multiplications, and all entries of P_k were calculated and the upper and lower triangular values were averaged to ensure symmetry.

In both the BTEKF and standard EKF, the state transition matrix $\Phi(t_k, t_{k-1})$ is approximated with a first order expansion of the linearized dynamics,

$$\Phi(t_k, t_{k-1}) = I + A(t_k - t_{k-1}) + H.O.T \tag{54}$$

where A is the Jacobian of the nonlinear dynamics of equations (6-10).

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 Section III.A. 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 1. Camera and magnetometer sensors were used.

Figures 3 and 4 illustrate the operation of the filter, during initialization until 5 seconds, and then in full operation afterwards. Figure 3 shows the root-mean-squared position error for the two filters. It can

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	derived	

Table 1. Initial covariance of the state vector

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 4 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. The case demonstrate a dramatic failure, but it can be assumed that inaccuracies are present in the standard filter even when failure does not occur.

B. Comparison of Computational Load of BTEKF and Standard EKF

An investigation of the computational requirements of the BTEKF and Standard EKF was performed. 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, we think this is a useful comparison.

The propagation portion of the filter was chosen for the comparison, as this is the most frequent operation and thus is often a limiting factor. The average computation time for the covariance propagation was measured over 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 2. The BTEKF propagation was found to be on average slightly more efficient at propagating the covariance than the standard EKF.

Propagation Time	
BTEKF	0.00211 s
Standard EKF	0.00262 s

Table 2. Average computation time for one covariance propagation over 60 s of operation for BTEKF and standard EKF.

C. Simulated Navigation Performance

The BTEKF was evaluated in simulation flying a prescribed trajectory. The GTMax flew four laps of an oval trajectory at 9.1 m/s velocity and an altitude of 30.5 m. 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 were used.[26]

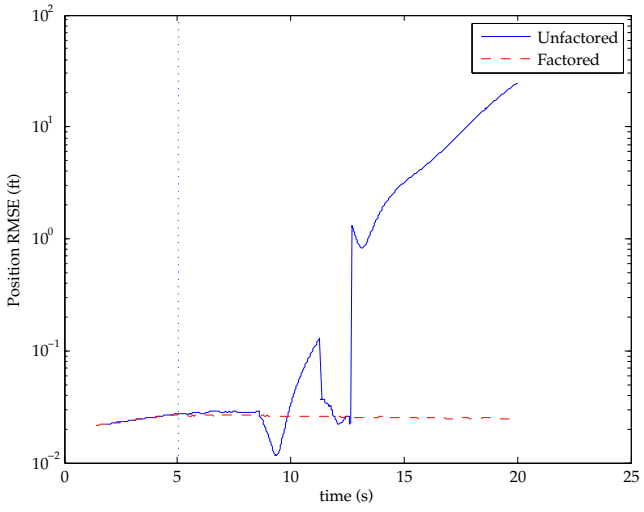


Figure 3. Position RMS error for the standard EKF (unfactored) and the BTEKF (factored). Filter initialization ends at 5 seconds.

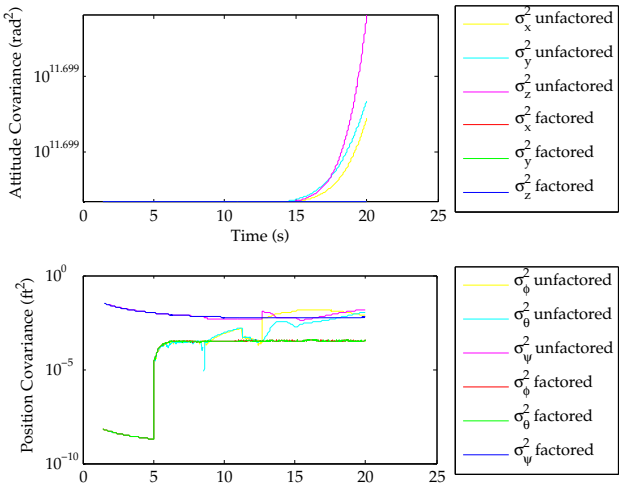


Figure 4. Position and attitude covariance for the standard EKF (unfactored) and the BTEKF (factored). Filter initialization ends at 5 seconds.

Figures 5, 6, and 7 show the results from the simulation. Figures 6 and 7 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 5 shows the trajectory of the vehicle and the navigation estimate. Table 3 shows some performance statistics from the simulated flight.

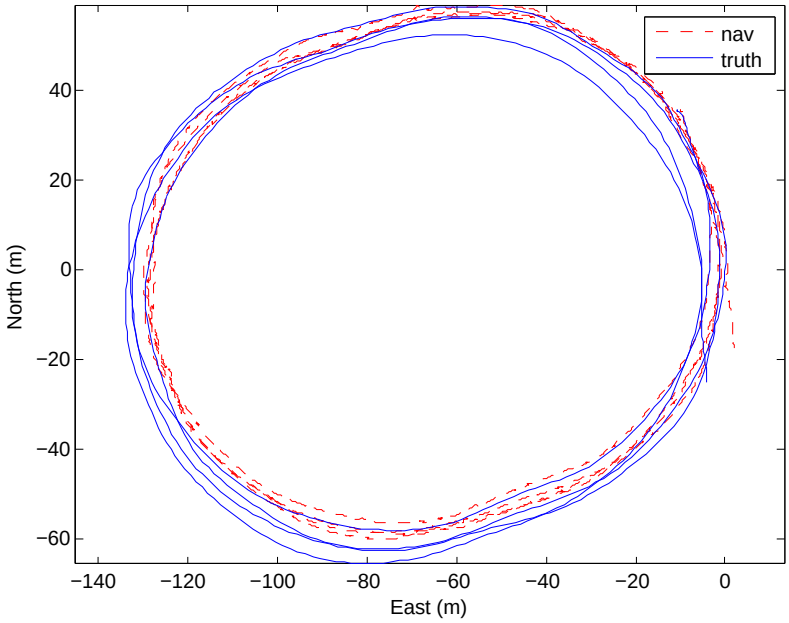


Figure 5. Horizontal position of vehicle as given by vision-based navigation position states (red dashed line) and simulation truth data (blue solid line) of the GTMax during a simulated flight of an oval trajectory. The total distance flown was approximately 1600 m.

Horizontal Error Statistics	
Distance	1575 m
RMS Error	5.5 m
% RMS Error	0.35 %
Final Error	9.7 m
% Final Error	0.61 %

Table 3. Horizontal error statistics for the simulated oval trajectory using vision-aided navigation.

D. Flight Test Results

The BTEKF SLAM navigation system was flight tested on the GTMax platform. The navigation system provided input to the vehicle controller, which tracked trajectory commands from 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. Additionally, the RTK-GPS altitude above a datum was used to simulate the behavior of a pressure altimeter. The horizontal RTK-GPS was recorded for comparison, but was not used in the navigation solution. Harris corner features were used, and an 11×11

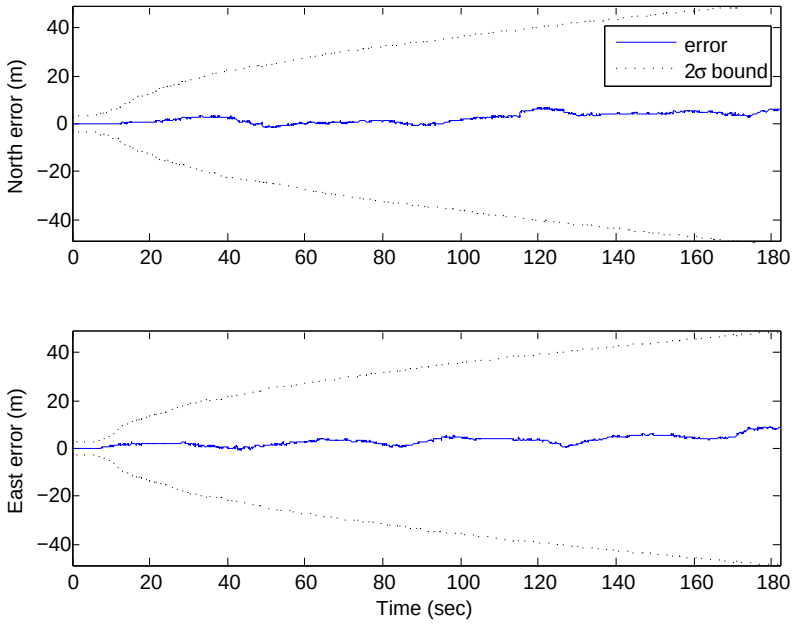


Figure 6. Horizontal position error and 2σ-covariance of vision-aided navigation system of the GTMax during a simulated flight of an oval trajectory. The final horiontal position error was 9.7 m.

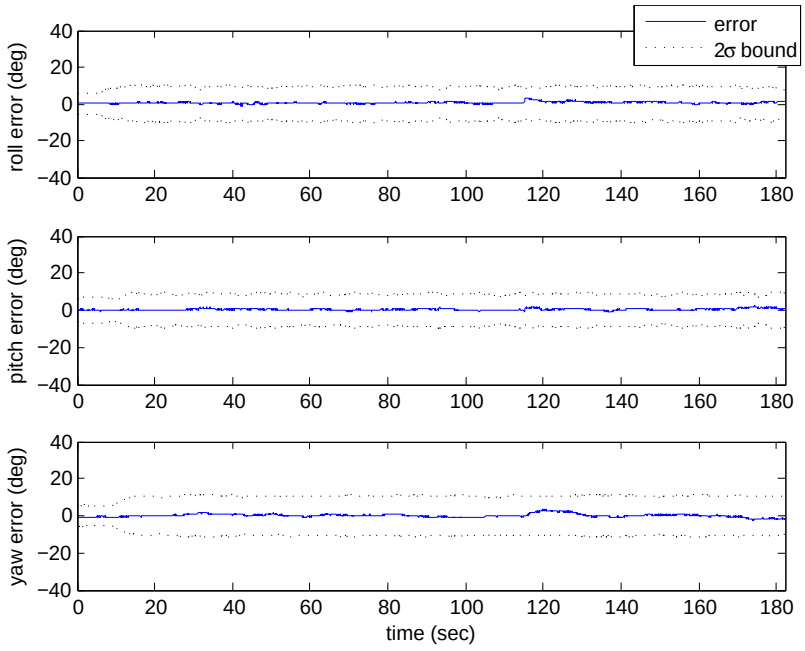


Figure 7. Attitude error and 2σ-covariance of vision-aided navigation system of the GTMax during a simulated flight of an oval trajectory.

pixel window around each feature was used as a descriptor. Figure 8 shows examples of images from the camera used during the test.

Figure 9 shows a image of the planned trajectory as seen from the ground station during the test. Figure 10 shows the navigation solution along with the GPS sensor data. The trajectory of the vehicle agrees with the GPS sensor data in general, but the figure does highlight two sources of error in the system. The first is the seen in the rotation of the navigation solution with respect to the GPS data. This is caused by an error in the magnetometer calibration. Since no absolute yaw angle information is provided by the vision aiding, this introduces a constant yaw angle bias into the solution. Second, the slight scaling of the navigation solution with respect to the GPS data. This is caused by a bias in the initialization distance. Features are initialized with a depth prior based on the datum height, and assume a Gaussian distribution about that height. Variation from this altitude in one direction violates the Gaussian assumption and causes a biased solution.

Figure 11 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. Oscillations in the error plot are caused by the magnetometer bias.

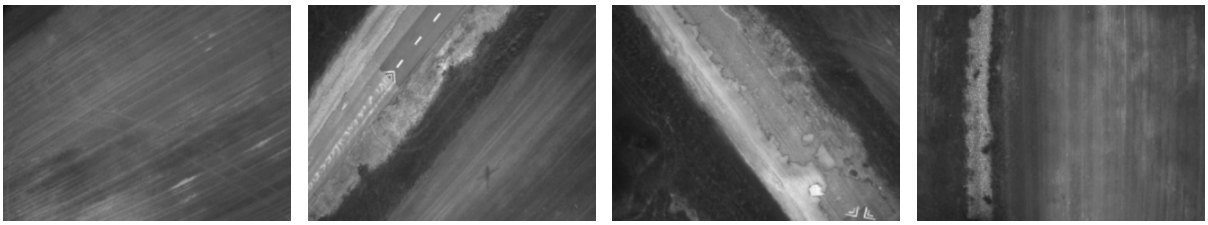
Table 4 gives numerical results from the test.

Flight Test Results	
Time	200 s
Linear Distance	1163 m
Horizontal RMS Error	5.51 m
% RMS Error	0.47 %
Final Error	3.88 m
% Final Error	0.33 %

Table 4. Flight test results for oval trajectory.

V. Conclusion

This paper presents a vision-aided navigation system based on the Bierman-Thornton factored extended Kalman filter (EKF). The filter update and propagation steps are described, and a novel method to initialize and remove states from the filter is presented. Simulation results illustrate the improved numerical stability over the standard EKF formulation and comparable computational requirements. Simulation and flight test with a controller in the loop was performed and the results presented. The navigation system remained within the 2σ bounds over most of the trajectory.



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

Figure 8. Example images from the camera during the flight test.

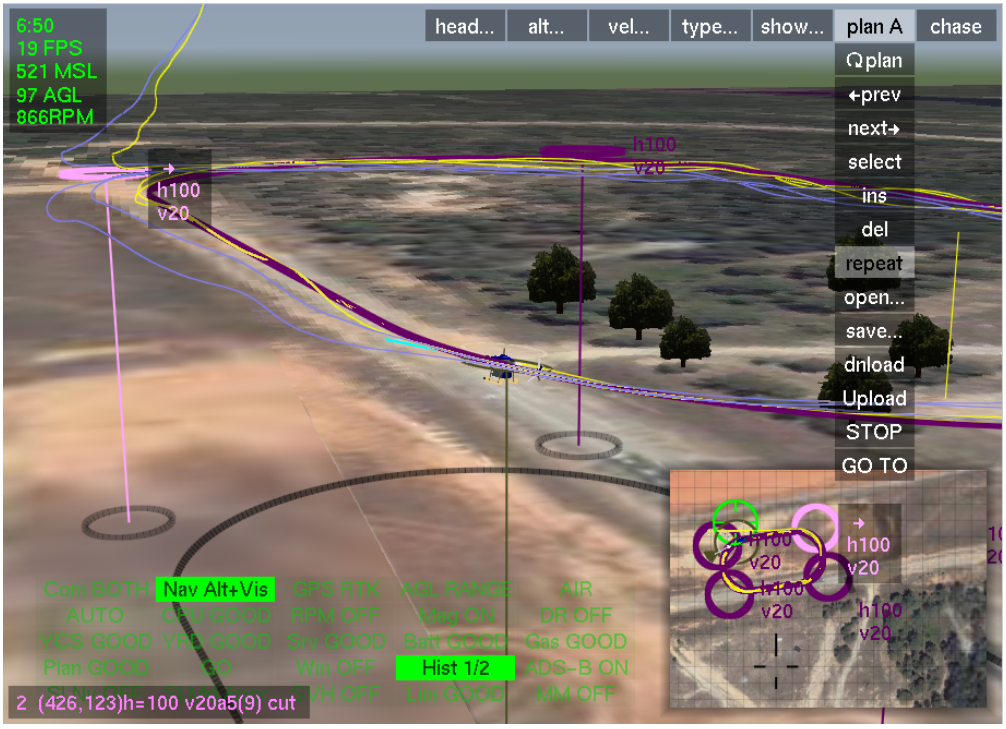


Figure 9. Image from ground control station during flight of oval trajectory. Yellow trace shows the navigation solution. Blue trace shows the GPS data. Purple line indicates the commanded trajectory.

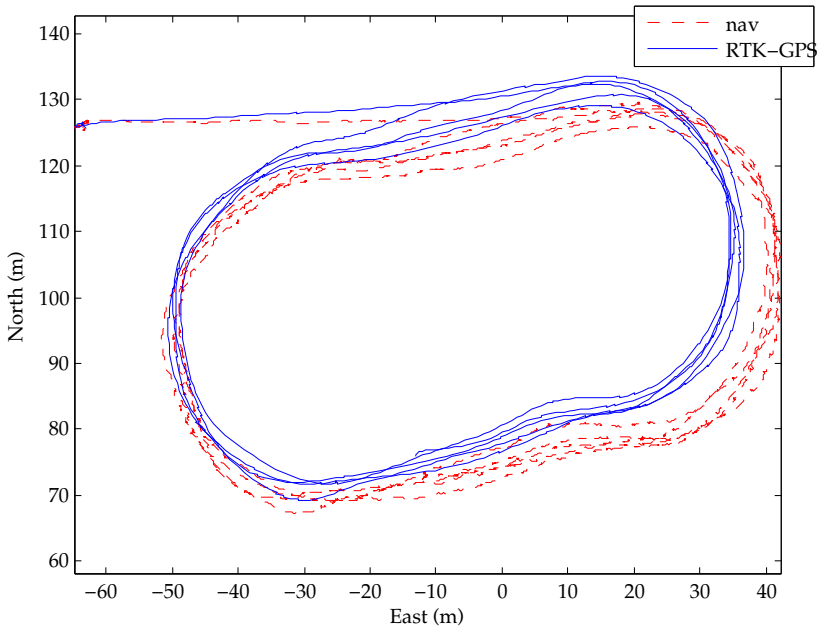


Figure 10. Horizontal navigation solution and RTK-GPS data for autonomous flight with controller in the loop.

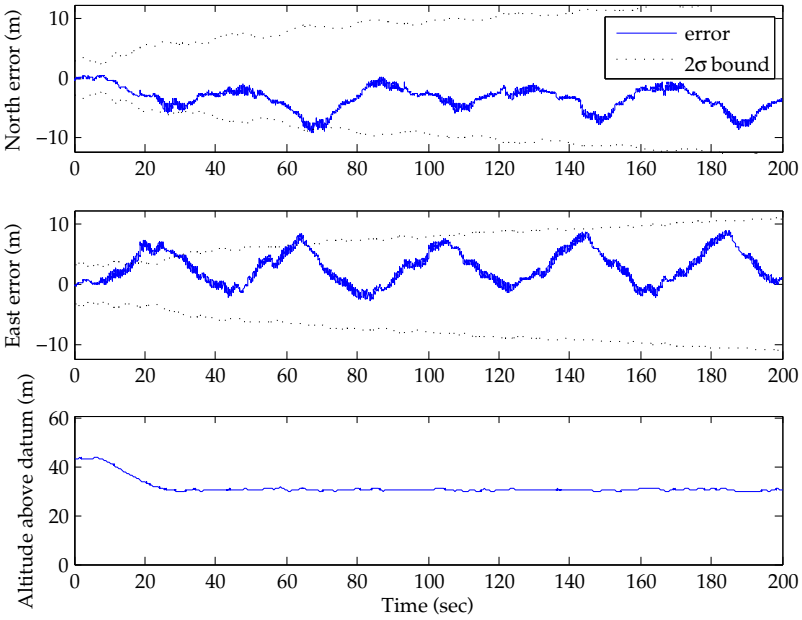


Figure 11. Horizontal position error of navigation solution from GPS truth for autonomous flight with controller in the loop. Altitude is shown for reference.

Acknowledgments

Many thanks to Henrik Christophersen, Jeong Hur, John Mooney, and Dmitry Bershadsky for help with flight testing.

References

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

[2]A. Davison, I. Reid, N. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 6, pp. 1052 –1067, June 2007.

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

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

[5]D. P. Magree and E. N. Johnson, "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. [Online]. Available: http://dx.doi.org/10.2514/6.2013-4699

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

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

[8]S. Weiss, D. Scaramuzza, and R. Siegwart, "Monocular-slambased navigation for autonomous micro helicopters in gps-denied environments," *Journal of Field Robotics*, vol. 28, no. 6, pp. 854–874, 2011. [Online]. Available: http://dx.doi.org/10.1002/rob.20412

[9]S. Weiss, M. W. Achtelik, S. Lynen, M. C. Achtelik, L. Kneip, M. Chli, and R. Siegwart, "Monocular vision for

long-term micro aerial vehicle state estimation: A compendium," *Journal of Field Robotics*, vol. 30, no. 5, pp. 803–831, 2013. [Online]. Available: http://dx.doi.org/10.1002/rob.21466

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

[11]R. C. Leishman, T. W. McLain, and R. W. Beard, "Relative navigation approach for vision-based aerial gps-denied navigation," *Journal of Intelligent & Robotic Systems*, vol. 74, no. 1-2, pp. 97–111, 2014. [Online]. Available: http://dx.doi.org/10.1007/s10846-013-9914-7

[12]K. Schmid, P. Lutz, T. Tomi, E. Mair, and H. Hirschmller, "Autonomous vision-based micro air vehicle for indoor and outdoor navigation," *Journal of Field Robotics*, pp. n/a–n/a, 2014. [Online]. Available: http://dx.doi.org/10.1002/rob.21506

[13]C. Thornton and G. Bierman, "Givens transformation techniques for kalman filtering," *Acta Astronautica*, vol. 4, no. 78, pp. 847 – 863, 1977. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0094576577900170

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

[15]G. J. Bierman and C. L. Thornton, "Numerical comparison of kalman filter algorithms: Orbit determination case study," *Automatica*, vol. 13, no. 1, pp. 23 – 35, 1977. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0005109877900061

[16]G. J. Bierman, "Measurement updating using the u-d factorization," *Automatica*, vol. 12, no. 4, pp. 375 – 382, 1976. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0005109876900583

[17]A. D. Wu, E. N. Johnson, M. Kaess, and F. Dellaert, "Autonomous Flight in GPS-Denied Environments Using Monocular Vision and Inertial Sensors," *Journal of Aerospace Information Systems*, vol. 10, no. 4, pp. 172–186, Apr. 2013. [Online]. Available: http://arc.aiaa.org/doi/abs/10.2514/1.I010023

[18]G. Chowdhary, E. N. Johnson, D. Magree, A. Wu, and A. Shein, "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. [Online]. Available: http://dx.doi.org/10.1002/rob.21454

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

[20]A. Wu, "Vision-bsed navigation and mapping for flight in gps-denied environments," Ph.D. dissertation, Georgia Institute of Technology, School of Aerospace Engineering, Atlanta, GA 30332, dec 2010. [Online]. Available: http://hdl.handle.net/1853/37281

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

[22]A. Gelb, *Applied Optimal Estimation*. MIT Press, 1974.

[23]E. N. Johnson and D. P. Schrage, "System integration and operation of a research unmanned aerial vehicle," *AIAA Journal of Aerospace Computing, Information and Communication*, vol. 1, no. 1, pp. 5–18, Jan 2004.

[24]D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004. [Online]. Available: http://dx.doi.org/10.1023/B%3AVISI.0000029664.99615.94

[25]E. N. Johnson and S. K. Kannan, "Adaptive Trajectory Control for Autonomous Helicopters," *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 3, pp. 524–538, May 2005. [Online]. Available: http://arc.aiaa.org/doi/abs/10.2514/1.6271

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