Navigation of Small UAVs Operating in Forests

Jack Langelaan[*] and Steve Rock[†]

Stanford University, Stanford, CA 94305, USA

A method for navigation of a small UAV through an unsurveyed environment (e.g. a forest) is presented. In particular the problem of estimating the state of the aircraft and of obstacles in the environment given limited sensors (an inertial measurement unit and a monocular camera) is addressed. The combination of limited observability and the close proximity of the vehicle to the obstacles lead to significant nonlinearities which cause Extended Kalman Filter (EKF) based approaches to fail. This paper presents an implementation of an Unscented Kalman Filter (UKF) to estimate the locations of obstacles and the state of the UAV based on measurements from the IMU and camera. This solution is applicable to navigation of a 6DOF vehicle in a three dimensional environment and allows generic obstacle avoidance routines to be used. Simulation results are presented for a two-dimensional environment which demonstrate: (a) stability and consistency of the UKF implementation and a comparison with an EKF implementation; (b) fusion with a potential field obstacle avoidance algorithm to enable navigation in an unknown two-dimensional environment.

I. Introduction

NAVIGATION THROUGH AN unknown environment is a pivotal problem in the field of mobile robotics. One approach consists of performing two fundamental tasks: generation of a model of the environment (i.e. a map) and using the map to plan a dynamically feasible collision-free trajectory through the environment. Both of these tasks have been the subject of a great deal of research. Much of the research in the area of sensing and world modelling has been conducted on wheeled ground vehicles or underwater vehicles. Research on trajectory planning for both air and ground vehicles has presupposed knowledge both of the environment and of current vehicle state, although that knowledge may be imprecise.

This paper addresses the problem of navigation of a small fixed wing UAV through previously unsurveyed environment (such as a forest). The primary focus is on generation of a model of the environment which is compatible with generic path planning algorithms.

Small UAVs flying among obstacles present unique challenges to the problem of generating models of the environment. The first issue is the limited observability of states. This results from the limited payload capacity of a small UAV, which restricts available sensors to a monocular camera (which provides bearings to obstacles) and a small, low cost inertial measurement unit (IMU). The second issue is the increased effect of non-linearities in the observation model, caused by the close proximity of the UAV to the objects being observed. The third issue is the significant uncertainty in predicted aircraft state, caused by the likelihood of large external disturbances and measurement noise and the drift error induced by biases in the IMU. Note that while a small UAV is likely to be equipped with a GPS receiver, the availability of reliable signals is uncertain in an environment with many obstacles (or below the forest canopy). Direct measurements of vehicle position are therefore unavailable.

Further issues not specific to small UAVs but applicable to fixed-wing UAVs in general are mentioned by Kim.[1] These include highly non-linear system dynamics, the limits imposed on landmark observability by trajectory constraints and the high accelerations and roll rates undergone by the vehicle.

---

[*]Ph.D. Candidate, Department of Aeronautics and Astronautics, Durand Building, Member AIAA.

[†]Professor, Department of Aeronautics and Astronautics, Durand Building, Associate Fellow AIAA.

This paper presents an Unscented Kalman Filter (UKF) based solution to the problem of estimating obstacle locations and vehicle state. The solution is general to a full three dimensional navigation problem and will work with generic path planning algorithms to generate collision–free trajectories. To demonstrate the effectiveness of this approach, results of simulations in two dimensions are presented. The estimator is merged with a potential field obstacle avoidance routine to show navigation through a two dimensional forest.

The remainder of this paper is organized as follows: Section II describes related work; Section III describes the problem of fusing data to perform the required task; Section IV describes the estimator used to perform the sensor fusion; Section V presents results of a Monte Carlo simulation comparing our solution with an EKF implementation and of a simulation showing navigation in an unknown forest; finally Section VI presents conclusions and plans for future work.

II.   Related Work

THERE HAS BEEN a tremendous amount of research relating to the problem of robot navigation. Different aspects of the problem can be loosely grouped into planning, vision augmented inertial navigation and vision based navigation, simultaneous localization and mapping (SLAM) and relative position estimation.

Path planning algorithms seeking to ensure vehicle safety in the presence of limited planning horizons,[2] in the presence of uncertainty in the environment[3] and for dynamic environments[4,5] have been presented, but these all require a priori knowledge of the environment.

Research into vision augmented inertial navigation[6–8] is primarily concerned with estimating the vehicle state by fusing inertial measurements either with bearings to known fiducials or data from optical flow algorithms. Positions of unknown obstacles are not estimated. A system for vision based navigation of an unmanned helicopter is presented by Sinopoli[9] but extensive use is made of a priori information and direct measurements of vehicle state are available through a fused GPS/INS.

Simultaneous Localization and Mapping (SLAM) has been applied both in simulation and on hardware in many scenarios including indoor navigation of small robots,[10] subsea navigation by Autonomous Underwater Vehicles (AUVs)[11] , outdoor navigation by wheeled robots[12] and navigation by aircraft.[1,13] Various algorithms have been developed to enable real-time SLAM in situations where maps contain large numbers of features[14–16] and algorithms have been developed to handle data association[17] and feature initialization.[18]

In most SLAM implementations the available measurements consist of relative measurements (i.e. range and bearing) to features in the environment and vehicle odometry or inertial measurements. The inherent non-linearity of the measurements and of the vehicle dynamics requires implementation of non-linear filters to estimate the vehicle state and map of the environment. In many cases methods based on extended Kalman filters (EKFs) have been successfully implemented. In particular, Kim[1,13] describes an implementation on fixed-wing UAVs. However, these were fairly large aircraft (10kg payload) and range to the objects was available either through laser range finder or from an estimate based on the size of the object in the camera's image plane. Moreover, these aircraft were operated at an altitude of approximately 100m above the objects used as SLAM references. Both the availability of range data and the large distance to the objects used as SLAM references stabilize the EKF.

Relative position estimation for an object pick-up task using only on-board inertial measurements and monocular vision is investigated by Huster.[19] Here EKF based estimators were found to behave erratically and an estimator based on the Unscented Kalman Filter was successfully implemented. This task, however, is fundamentally different from the navigation task in that only a single object is considered and object contact is desired instead of avoided.

While the related work addresses problems similar to that addressed in this paper none address the specific issues which arise due to the limited sensing available to a small UAV and the close proximity of the obstacles as the UAV flies through the environment.

III.   Problem Statement

A.   Navigation Scenario

The scenario considered here consists of a small UAV flying through an unsurveyed forest (Figure 1). An on-board camera obtains bearing measurements to obstacles (tree trunks) and an inertial measurement unit

provides accelerations and angular rates in the body-fixed frame.

Trees are located at $\mathbf{x}_i$ in frame O, an inertial NED frame. A transformation matrix $\mathbf{T}$ resolves a vector in frame O to the body frame B. $\boldsymbol{\omega}$ represents the rotational velocity of the vehicle expressed in the body frame and $[u\ v\ w]^T$ represents the velocity of the aircraft in the body frame. The position of the aircraft is defined by $[X\ Y\ Z]^T$, expressed in frame O.

The problem is to estimate the locations of the tree trunks and of the aircraft given the limited sensor suite. This information can subsequently be used by a path planning algorithm to generate a collision-free trajectory through the forest.

The estimation problem contains two significant non-linearities. The first is due to the rotational degrees of freedom of the UAV; the second is due to the vision system's projection of the three-dimensional

Figure 1. Geometry of navigation problem.

world onto the two-dimensional image plane. Thus the system dynamics and measurement equations are highly non-linear and depend on the actual state of the system.

Since only bearing measurements are available, observability of states is highly dependent on motion of the camera (i.e. the trajectory flown by the UAV). During motion directly towards or away from an object (along the bearing) there is no new information which improves range estimate. Transverse motion is required produce a useful estimate of object position. In the case of obstacle avoidance, transverse motion has the added benefit of ensuring that a collision is avoided, but the presence of multiple obstacles places conflicting demands on the trajectory which must be flown. Additional demands on the trajectory may arise if certain areas are to be avoided or if a final goal location must be reached. However, trajectory-related issues are not addressed in this paper.

To summarize, given a sensor suite limited to a monocular camera and an IMU an estimate of aircraft state and obstacle locations must be generated.
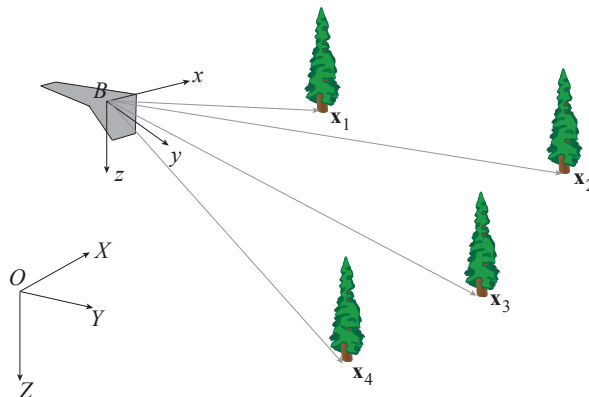
B. System Description

The block diagram in Figure 2 shows a system that uses the given sensors to perform obstacle avoidance. The vector of states which is estimated includes the vehicle state $\mathbf{x}_v$ (which includes position in the inertial frame; orientation with respect to the inertial frame; velocity expressed in the body frame; and IMU biases) and the positions of the tree trunks $\mathbf{x}_o$ (in the inertial frame).

IMU measurements $\mathbf{z}_{imu}$ (acceleration and angular rate in the body frame) and camera measurements $\mathbf{z}_{cam}$ (bearings to obstacles) are fused in the estimator, which computes estimates of the vehicle state $\hat{\mathbf{x}}_v$ and obstacle positions $\hat{\mathbf{x}}_o$ along with the associated covariances. These estimates are given to the trajectory generator, which computes a collision free trajectory to be followed by the aircraft. The difference between the desired trajectory $\mathbf{x}_{v,des}$ and the estimated aircraft state is used by the flight controller to generate control inputs.

Figure 2. System block diagram

C. Sensor Models

The camera is assumed to be fixed to the aircraft with known offset $\triangle\mathbf{s}$ from the CG and known angular offset from the body-fixed frame, defined by a transformation $\mathbf{T}_{cam}$. The camera x-axis is perpendicular to the image plane. Bearings to the ith tree
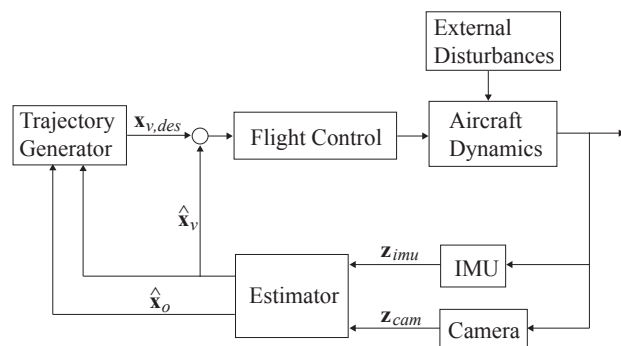
are modelled as:

$$\mathbf{z}_{c,i} = \frac{f}{S_{c_x,i}} \begin{bmatrix} S_{c_y,i} \\ S_{c_z,i} \end{bmatrix} + \mathbf{n}_c \tag{1}$$

The vector $\mathbf{S}_{c,i}$ represents the vector from the camera to the ith tree in camera coordinates.

$$\mathbf{S}_{c,i} = \mathbf{T}_{cam} \left[\mathbf{T} \begin{bmatrix} X_i - X \\ Y_i - Y \\ Z_i - Z \end{bmatrix} - \triangle\mathbf{s} \right] \tag{2}$$

The inertial measurement unit includes accelerometers and rate gyros. The accelerometers measure specific force, which includes the acceleration of the vehicle and the the acceleration due to gravity. The rate gyros measure the rotational velocity of the vehicle. Both sensors include sensor biases and zero mean Gaussian random noise.

$$\mathbf{z}_a = \text{diag}(\boldsymbol{\alpha}) \left[\frac{d}{dt}\mathbf{u}_a - \mathbf{Tg} \right] + \mathbf{b}_a + \mathbf{n}_a \tag{3}$$

$$\mathbf{z}_\omega = \boldsymbol{\omega} + \mathbf{b}_\omega + \mathbf{n}_\omega \tag{4}$$

The accelerometer scale factor correction is represented by $\text{diag}(\boldsymbol{\alpha}) \approx \mathbf{I}_{3\times3}$ and $\mathbf{u}_a$ is the velocity of the accelerometer, which is offset from the aircraft CG by a known amount $\boldsymbol{\rho}$.

Sensor biases and the accelerometer scale factor are assumed to vary by a random walk model with zero mean Gaussian driving terms.

$$\dot{\boldsymbol{\alpha}} = \mathbf{n}_\alpha \tag{5}$$

$$\dot{\mathbf{b}}_a = \mathbf{n}_{b_a} \tag{6}$$

$$\dot{\mathbf{b}}_\omega = \mathbf{n}_{b_\omega} \tag{7}$$

D. Kinematic Model

Aircraft rotations are expressed as Euler angles ϕ, θ, ψ relative to the fixed North-East-Down coordinate frame. Velocities u, v, w are in expressed in the body frame. Given accelerations and angular rates from the IMU the aircraft state can be obtained by integration. Note that sensor noise and biases will lead to unbounded drift if corrective steps are not taken, but fusion of bearings to objects which are known to be fixed can correct for the drift.

$$\dot{X} = \cos\psi\cos\theta\, u + (\cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi)\, v + (\cos\psi\sin\theta\cos\phi)\, w \tag{8}$$

$$\dot{Y} = \sin\psi\cos\theta u + (\sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi)v + (\sin\psi\sin\theta\cos\phi)w \tag{9}$$

$$\dot{Z} = -\sin\theta u + \cos\theta\sin\phi v + \cos\theta\cos\phi w \tag{10}$$

$$\dot{\phi} = (z_p - b_p) + \tan\theta\sin\phi(z_q - b_q) - \tan\theta\cos\phi(z_r - b_r) \tag{11}$$

$$\dot{\theta} = \cos\phi(z_q - b_q) - \sin\phi(z_r - b_r) \tag{12}$$

$$\dot{\psi} = \frac{\sin\phi}{\cos\theta}(z_q - b_q) + \frac{\cos\phi}{\cos\theta}(z_r - b_r) \tag{13}$$

$$\dot{u} = \frac{z_x}{\alpha_x} - g\sin\theta - b_x - (z_q - b_q)w + (z_r - b_r)v - b_x(\rho) \tag{14}$$

$$\dot{v} = \frac{z_y}{\alpha_y} + g\cos\theta\sin\phi - b_y - (z_r - b_r)u + (z_p - b_p)w - b_y(\rho) \tag{15}$$

$$\dot{z} = \frac{z_z}{\alpha_z} + g\cos\theta\cos\phi - b_z - (z_p - b_p)v + (z_q - b_q)u - b_z(\rho) \tag{16}$$

Here $\mathbf{b}(\rho)$ is the bias induced by the (known) offset of the accelerometer from the aircraft CG.

IV. Estimator Design

THE USE OF low-cost inertial sensors complicates system design. Both bias and random noise errors are present and may be quite large in low cost inertial sensors, which can lead to unbounded errors in the integrated quantities. Hence sensor biases must be included in the UAV state vector.

A. System Equations

The estimator propagates a state vector (which includes aircraft state and obstacle states) and the associated covariance:

$$\hat{\mathbf{x}} = \left[\begin{array}{c} \hat{\mathbf{x}}_v \\ \hat{\mathbf{x}}_o \end{array} \right] \tag{17}$$

$$\mathbf{P} = \left[\begin{array}{cc} \mathbf{P}_{vv} & \mathbf{P}_{vo} \\ \mathbf{P}_{ov} & \mathbf{P}_{oo} \end{array} \right] \tag{18}$$

Note that

$$\hat{\mathbf{x}}_o = \left[\begin{array}{c} \hat{\mathbf{x}}_1 \\ \hat{\mathbf{x}}_2 \\ \vdots \\ \hat{\mathbf{x}}_m \end{array} \right] \tag{19}$$

The vehicle state vector is

$$\hat{\mathbf{x}}_v = \left[\begin{array}{cccccccccccc} X & Y & Z & \phi & \theta & \psi & u & v & w & \boldsymbol{\alpha}^T & \mathbf{b}_a^T & \mathbf{b}_\omega^T \end{array} \right]^T \tag{20}$$

1. Time Update

The time update of the filter is driven by the kinematics of the vehicle (equations 8– 16) and dynamics of the IMU biases (equations 5– 7) with measurements of acceleration and angular rate supplied by the inertial measurement unit.

In discrete form, the system dynamics are

$$\mathbf{x}_{k+1} = \left[\begin{array}{c} \mathbf{x}_{v,k+1} \\ \mathbf{x}_{o,k+1} \end{array} \right] = \left[\begin{array}{c} f(\mathbf{x}_{v,k}, \mathbf{u}_k) \\ \mathbf{x}_{o,k} \end{array} \right] + \left[\begin{array}{c} \mathbf{n}_{ext} \\ 0 \end{array} \right] \tag{21}$$

where

$$\mathbf{u}_k = \left[\begin{array}{c} \mathbf{z}_k \\ \mathbf{n}_{\alpha,k} \\ \mathbf{n}_{b,k} \end{array} \right] \tag{22}$$

Both $\mathbf{u}$ and $\mathbf{n}_{ext}$ are assumed to be Gaussian random variables. Note that measurements from the IMU are likely to be available at a much higher rate than measurements from the camera, allowing the time update to proceed at a higher rate than the vision correction.

The process noise for the time update must still be obtained. In this case process noise arises from the noisy measurements of acceleration and angular rate. We approximate process noise as

$$\mathbf{Q}_k = \left[\begin{array}{cc} \nabla_{\mathbf{u}} f \Sigma_{\mathbf{u}} \nabla_{\mathbf{u}} f^T + \Sigma_{ext} & 0 \\ 0 & 0 \end{array} \right] \tag{23}$$

This is identical to the EKF process noise approximation, and is adequate for the application being considered. Since obstacles (i.e. trees) are assumed to be perfectly stationary only components corresponding to vehicle states have non-zero process noise.

2. Vision Correction

Measurements from the camera are incorporated in the measurement update step of the estimator. The measurement model (equations 1– 2) is also non-linear and has the form

$$\mathbf{z}_{cam,k} = h(\mathbf{x}_{v,k}, \mathbf{x}_{o,k}) + \mathbf{n}_c \tag{24}$$

where $\mathbf{n}_c$ is a Gaussian random variable.

B.   Unscented Kalman Filter Implementation

An extended Kalman filter approximates a non-linear system by linearizing the system equations about the current estimate. However, the states about which the dynamics are linearized are uncertain, leading to potentially significant unmodelled uncertainty in the linearization. An Unscented Kalman Filter[20, 21] instead approximates the probability distribution of the state which is to be estimated. The probability distribution is represented by a set of *Sigma Points* which capture the mean and covariance of a random vector $\mathbf{x} = \mathcal{N}(\hat{\mathbf{x}}, \mathbf{P})$:

$$\mathbf{X} = \left[\begin{array}{ccc} \hat{\mathbf{x}} & \hat{\mathbf{x}} + \eta\sqrt{\mathbf{P}} & \hat{\mathbf{x}} - \eta\sqrt{\mathbf{P}} \end{array} \right] \tag{25}$$

where η is a scale factor and $\sqrt{\mathbf{P}}$ is an orthogonal matrix square root of the covariance of the distribution. Note that this matrix square root is not unique: it can be chosen based on issues such as numerical stability, computational cost or to exploit known structure in the problem (Huster[19] makes use of this property: he chooses sigma points so that the range to the target is always positive). These Sigma Points are propagated through the full non-linear system equations and the estimate mean and covariance are then recovered. This allows the statistics of the time update step and measurement update to be captured more accurately.

This implementation allows generation of a world model for a 6DOF vehicle with a limited sensor suite (IMU and monocular vision only) operating in close proximity to the objects being observed. Note that several issues have not been addressed, including feature initialization, measurement correspondence and feature management to enable real time implementation. These will be addressed in future work.

V.   Simulation Results

WHILE THE SOLUTION above is general to a full three dimensional navigation problem, for ease of visualization simulations were conducted in two dimensions. The system kinematics in 2D are:

$$\dot{X} = \cos\psi u - \sin\psi v \tag{26}$$

$$\dot{Y} = \sin\psi u + \cos\psi v \tag{27}$$

$$\dot{\psi} = z_r - b_r \tag{28}$$

$$\dot{u} = \frac{z_x}{\alpha_x} - b_x + (z_r - b_r)v - b_x(\rho) \tag{29}$$

$$\dot{v} = \frac{z_y}{\alpha_y} - b_y - (z_r - b_r)u - b_y(\rho) \tag{30}$$

The 2D measurement model is

$$z_{c,i} = \arctan\frac{fS_{c_y,i}}{S_{c_x,i}} + n_c \tag{31}$$

where

$$\mathbf{S}_{c,i} = \mathbf{T}_{cam}\left[\mathbf{T}\left[\begin{array}{c} X_i - X \\ Y_i - Y \end{array}\right] - \triangle\mathbf{s}\right] \tag{32}$$

The camera measurement was expressed as a bearing to avoid singularities.

Simulations were conducted to evaluate the performance of the estimator (compared with an EKF implementation) and to demonstrate the fusion of the estimator with a preliminary implementation of a potential field obstacle avoidance algorithm.

A.   Estimator Performance

The UAV is flown in a circular trajectory of radius 20m at a velocity of 10m/s through a forest of randomly distributed trees (see figure 3).

A new forest was generated for each run. UAV position and velocity was assumed to be known accurately at the start of each run ($\sigma^2 = 1 \times 10^{-6}$). True IMU biases were not known and were generated at the start of each run. IMU initialization parameters are below.

$$\begin{aligned}
\bar{\alpha}_x &= 1 & \sigma &= 0.01 \\
\bar{\alpha}_y &= 1 & \sigma &= 0.01 \\
\bar{b}_x &= 0.15 & \sigma &= 0.1 \\
\bar{b}_y &= 0.10 & \sigma &= 0.1 \\
\bar{b}_\psi &= 0.004 & \sigma &= 0.02
\end{aligned}$$
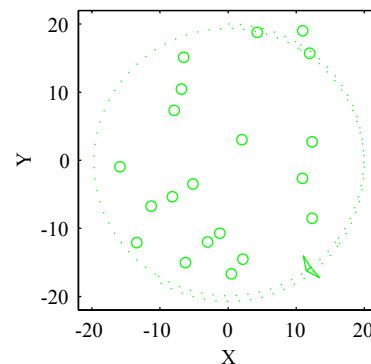
Figure 3. Schematic of a single run of the Monte Carlo simulation. Aircraft speed is 10m/s, turn rate is 1rad/s. The aircraft travels a distance of 200m, approximately 1.6 times around the circular path.

To avoid issues associated with landmark initialization the location of each obstacle was assumed to be known to a standard deviation of 1 meter. This is not in keeping with the stated aim of navigation in a previously unsurveyed environment but the purpose of this Monte Carlo simulation was to investigate the ability of the estimator to generate consistent estimates of the vehicle state and obstacle positions and to provide a comparison with an EKF implementation.

Since the purpose of the estimator is to determine vehicle state and obstacle positions with an accurate assessment of the estimate error, success of a run is determined by comparing the estimated standard deviation of the error with the true standard deviation of the error. These are:

$$\sigma_{k,est} = \sqrt{\text{Tr} P_{kk}} \tag{33}$$

$$\sigma_{k,truth} = \sqrt{(\hat{\mathbf{x}}_k - \mathbf{x}_k)^T(\hat{\mathbf{x}}_k - \mathbf{x}_k)} \tag{34}$$

On average for an unbiased estimator $\sigma_{k,est} = \sigma_{k,truth}$. For each run we compare the mean of $\sigma_{k,est}$ with the mean of $\sigma_{k,truth}$. A run is deemed successful if

$$\frac{\bar{\sigma}_{truth}}{\bar{\sigma}_{est}} < 1.5 \quad \text{where} \quad \bar{\sigma} = \frac{1}{K}\sum_{k=1}^{K}\sigma_k \tag{35}$$

meaning that on average of the run the true standard deviation of the error does not exceed the estimated standard deviation of the error by more than 50%. Results are shown in Table 1.

Table 1. Results of Monte Carlo simulation

| 1000 runs	EKF success	EKF Failure
UKF Success	395	605
UKF Failure	0	0

In 395 cases both the UKF and EKF produced consistent estimates of the full system state. In a further 605 cases the UKF was successful and the EKF was not. By the failure definition used here there were no cases where the UKF produced an inconsistent estimate.

It is also interesting to examine the maximum and minimum standard deviation ratios for each estimator (see Table 2). This gives an indication of the degree of divergence run: the UKF's maximum divergence was 1.48 and the EKF maximum divergence was 42.3.

Table 2. Maximum and minimum standard deviation ratios

1000 runs	max $\frac{\bar{\sigma}_{truth}}{\bar{\sigma}_{est}}$	min $\frac{\bar{\sigma}_{truth}}{\bar{\sigma}_{est}}$
UKF	1.48	0.62
EKF	42.3	0.70

American Institute of Aeronautics and Astronautics Paper 2004-5140

Results are presented graphically in figure 4. Shown is a comparison of the mean estimated standard deviation (over 1000 runs) and the mean true standard deviation of the estimate error. For the UKF the estimated and true standard deviations line up, indicating a consistent unbiased estimator. For the EKF the true standard deviation diverges, indicating an unmodelled bias in the estimator.

(a) EKF implementation

(b) UKF implementation
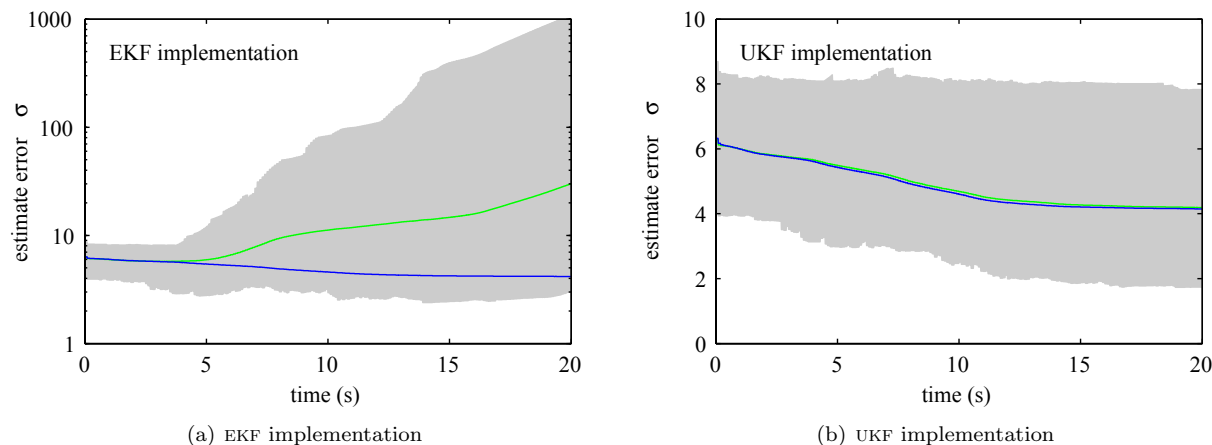
Figure 4. Comparison of estimated standard deviation of the estimate error and true standard deviation of the estimate error for the ekf implementation (left) and ukf implementation (right). Note the difference in scales. Results are from a 1000 run Monte Carlo simulation. The mean of the true standard deviation of estimate error is shown in green, mean of estimated standard deviation of estimate error is shown in blue. The shaded area represents the envelope of the true error variance over 1000 runs.

Figure 5 shows vehicle position estimate error for a single representative run. The vehicle state estimated by the UKF (shown in blue) remains within the 3σ bounds for the entire run while the state estimated by the EKF implementation (shown in red) diverges.

B. Obstacle Avoidance

To demonstrate the effectiveness of the UKF-based estimator with obstacle avoidance a potential field algorithm is linked with the estimator. Inputs to the aircraft (in this 2D case, yaw rate) are generated based on the distance between the obstacles' 3σ error ellipsoid and a circle of radius 1 meter around the aircraft. If the distance is zero (i.e. the ellipsoid and the circle overlap) maximum yaw rate is commanded.

Figure 6 shows a series of pictures taken from a single run. The UAV begins at $X = -30$ flying at 10m/s in the positive X direction. The camera has a 90° field of view and the optical axis is aligned with the UAV's body x axis. Maximum camera range is 25m and measurement error 1σ is 1.5°. IMU biases are initialized as

$$
\begin{bmatrix} \alpha_x \\ \alpha_y \\ b_x \\ b_y \\ b_\psi \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0.15 \\ 0.1 \\ 0.004 \end{bmatrix} \qquad \mathbf{P}_{0|0} = \begin{bmatrix} 0.01^2 & 0 & 0 & 0 & 0 \\ 0 & 0.01^2 & 0 & 0 & 0 \\ 0 & 0 & 0.1^2 & 0 & 0 \\ 0 & 0 & 0 & 0.1^2 & 0 \\ 0 & 0 & 0 & 0 & 0.02^2 \end{bmatrix}
$$

True IMU biases were randomly generated at the beginning of the run based on the statistics above.

The forest consists of 20 trees randomly distributed in the region defined by $-20 \le X \le 20$ and $-15 \le Y \le 15$. Tree locations were initialized with 1σ error of 2m in X and Y.

Because of the camera's limited range and field of view not all trees are visible at all times. However, in the particular case illustrated in figure 6 all trees are eventually seen by the UAV. Note also that the closer a tree is to the path of the UAVthe more accurately its position is determined. This permits more precise navigation as the UAVapproaches obstacles.

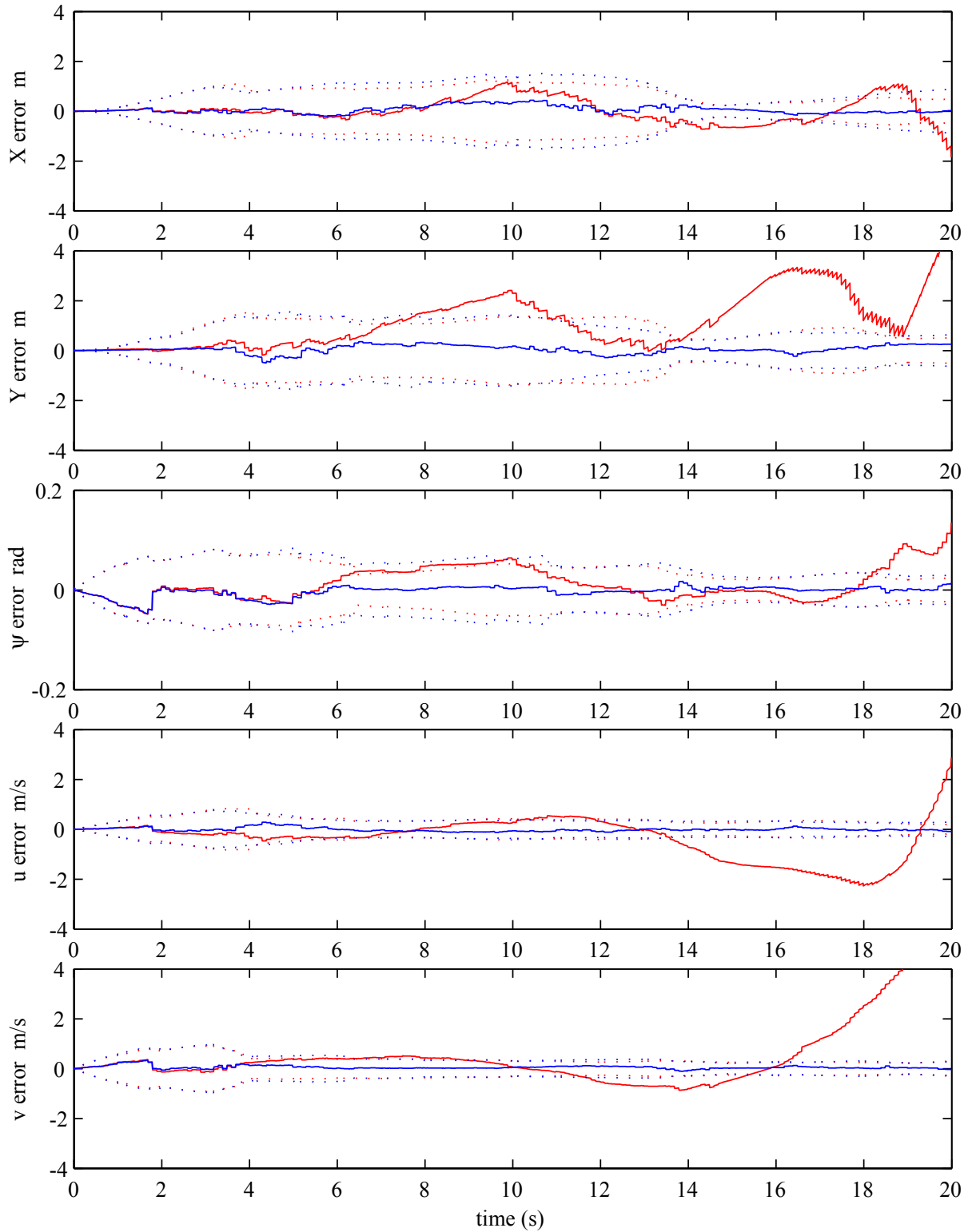American Institute of Aeronautics and Astronautics Paper 2004-5140

Figure 5. Aircraft estimated position and velocity error for a representative run. EKF error is shown in solid red, UKF error is shown in blue. 3σ bounds are shown as dotted red and blue lines for the EKF and UKF, respectively. Note the initial good agreement between the EKF and UKF implementations, although the EKF implementation quickly diverges.

American Institute of Aeronautics and Astronautics Paper 2004-5140

(a) t=0 seconds

(b) t=2 seconds

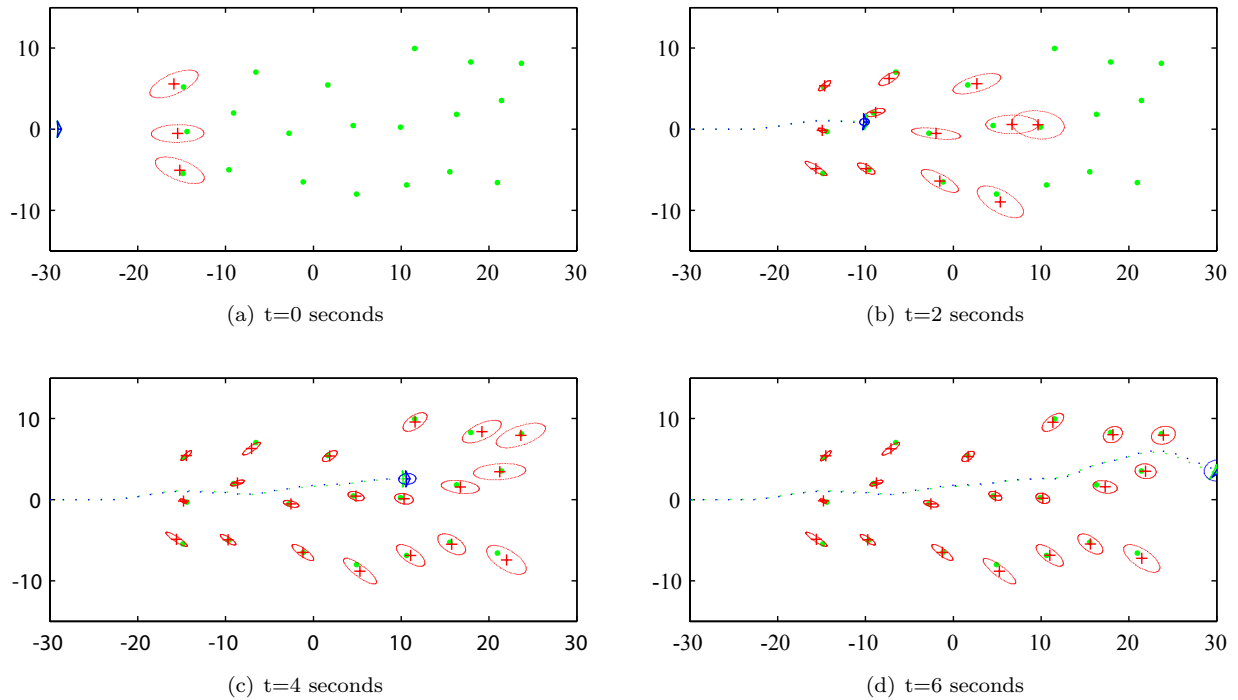(c) t=4 seconds

(d) t=6 seconds

Figure 6. Obstacle avoidance. True tree locations are shown as green dots, estimated tree locations are shown as red crosses with a surrounding 3σ error ellipsoid. Estimated tree locations are only shown for trees that have come into view of the camera. True UAV location is shown in green with a dotted line representing the path. Estimated UAV location is shown in blue with the 3σ error ellipsoid. The blue dotted line represents the estimated path of the UAV. In this case aircraft control input (turn rate) was determined based on the distance between a circle of radius 1m around the estimated aircraft position and the 3σ uncertainty ellipsoid around each tree.

American Institute of Aeronautics and Astronautics Paper 2004-5140

VI. Summary

THIS PAPER HAS presented an algorithm for generating a world model which is applicable to the problem of navigating a small UAV through a previously unsurveyed environment (such as a forest).

The solution is general to navigation in a three dimensional environment. Results of simulations conducted in two dimensions show that while an EKF implementation diverges, the UKF based implementation generates consistent estimates of the state of the vehicle and positions of the obstacles, and that this estimator can be merged with an obstacle avoidance algorithm to permit navigation in previously unsurveyed environments.

We have constructed a small (5 foot wingspan) UAV to enable hardware demonstration of the techniques described here. Results will be presented in future work.

References

[1] Kim, J. H. and Sukkarieh, S., "Airborne Simultaneous Localisation and Map Building," *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, Taipei, Taiwan, 2003.

[2] Schouwenaars, T., Feron, E., and How, J., "Safe Receding Horizon Path Planning for Autonomous Vehicles," *40th Allerton Conference on Communication, Control and Computing*, October 2002.

[3] Jun, M. and D'Andrea, R., *Cooperative Control: Models, Applications and Algorithms*, chap. 6, Kluwer, 2002.

[4] Jun, M., Chaudry, A., and D'Andrea, R., "The Navigation of Autonomous Vehicles in Uncertain Dynamic Environments: A Case Study," *IEEE Conference on Decision and Control (CDC)*, IEEE, Las Vegas, NV, 2002.

[5] Hsu, D., Kindel, R., Latombe, J. C., and Rock, S., "Randomized Kinodynamic Motion Planning with Moving Obstacles," *International Journal of Robotics Research*, Vol. 21, No. 3, March 2002, pp. 233–255.

[6] Roumeliotis, S. I., Johnson, A. E., and Montgomery, J. F., "Augmenting Inertial Navigation with Image-Based Motion Estimation," *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, Washington, DC, 2002.

[7] Lobo, J. and Dias, J., "Integration of Inertial Information with Vision," *24th Conference of IEEE Industrial Electronics Society (IECON)*, IEEE, Aachen, Germany, 1998, pp. 1263–1267.

[8] Netter, T. and Franceschini, N., "A Robotic Aircraft that Follows Terrain Using a Neuromorphic Eye," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Lausanne, Switzerland, 2002.

[9] Sinopoli, B., Micheli, M., Donato, G., and Koo, T. J., "Vision Based Navigation for an Unmanned Aerial Vehicle," *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2001.

[10] Thrun, S., Burgard, W., and Fox, D., "A Real-Time Algorithm for Mobile Robot Mapping with Applications to Multi-Robot and 3D Mapping," *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, San Francisco, CA, 2000.

[11] Williams, S. B., Dissanayake, G., and Durrant-Whyte, H., "Field Deployment of the Simultaneous Localisation and Mapping Algorithm," *15th IFAC World Congress on Automatic Control*, Barcelona, Spain, June 2002.

[12] Liu, Y. and Thrun, S., "Results for Outdoor-SLAM using Sparse Extended Information Filters," *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, Taipei, Taiwan, 2003.

[13] Kim, J. H. and Sukkarieh, S., "Recasting SLAM—Towards Improving Efficiency and Platform Independency," *11th International Symposium of Robotics Research (ISRR)*, Siena, Italy, 2003.

[14] Guivant, J. E. and Nebot, E. M., "Optimization of the Simultaneous Localization and Map-Building Algorithm for Real-Time Implementation," *IEEE Transactions on Robotics and Automation*, Vol. 17, No. 3, June 2001, pp. 242–257.

[15] Dissanayake, G., Williams, S. B., Durrant-Whyte, H., and Bailey, T., "Map Management for Efficient Simultaneous Localization and Mapping (SLAM)," *Autonomous Robots 12*, 2002, pp. 267–286.

[16] Thrun, S., Koller, D., Ghahmarani, Z., and Durrant-Whyte, H., "SLAM Updates Require Constant Time," Tech. rep., Carnegie Mellon University, 2002.

[17] Montemerlo, M. and Thrun, S., "Simultaneous Localization and Mapping with Unknown Data Association Using Fast-SLAM," *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, Taipei, Taiwan, 2003.

[18] Bailey, T., "Constrained Initialisation for Bearing-Only SLAM," *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, Taipei, Taiwan, 2003.

[19] Huster, A. and Rock, S., "Relative Position Sensing by Fusing Monocular Vision and Inertial Rate Sensors," *11th International Conference on Advanced Robotics (ICAR)*, 2003.

[20] Julier, S., Uhlmann, J., and Durrant-Whyte, H. F., "A New Method for the Nonlinear Transformation of Means and Covariances in Filters and Estimators," *IEEE Transactions on Automatic Control*, Vol. 45, No. 3, March 2000, pp. 477–482.

[21] van der Merwe, R. and Wan, E., "The Square-Root Unscented Kalman Filter for State and Parameter-Estimation," *IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, Salt Lake City, UT, 2001.