

A Vision Based Relative Navigation Framework for Formation Flight

Daniel B. Wilson, Ali H. Göktoğan and Salah Sukkarieh

Abstract—Unmanned aerial vehicle (UAV) formation flight can vastly increase operational range and persistence through autonomous aerial refuelling or efficient flight on a wingman's wake vortices. Differencing individual UAV state estimates is not sufficiently accurate for close formation operations and must be augmented with vehicle-to-vehicle observations.

To this end, we propose a quaternion based unscented Kalman filter to fuse information from each UAV sensor suite with relative vision observations. The result is a vastly improved relative state estimate that is resilient to brief vision dropouts and degrades gracefully during extended dropouts. Simulated formation flight results validate the approach and provide a numerical analysis of the algorithm performance. Ground based experiments demonstrate the algorithm running in real-time on a dual-UAV system. This represents a significant step towards an airborne implementation.

I. INTRODUCTION

Autonomous tight formation flight of less than one wingspan of separation [1] is a lucrative and worthy challenge because it enables a wide variety of applications, most notably, reduced fuel consumption by utilisation of wake vortices [1] and autonomous aerial refuelling [2]. These unmanned aerial vehicle (UAV) applications can significantly increase the UAV's operational range, endurance and persistence without sacrificing payload space for supplementary fuel.

Successful, sustained tight formation flight requires an accurate and real-time relative state estimate. One method to obtain this estimate is to subtract one vehicle's GPS-based state estimate from the other; the result will be known herein as the raw relative estimate. This approach has merit during high separation formation but the accuracy, particularly in relative position, is in the order of meters which is not sufficiently accurate for close proximity formation flight. The low accuracy is not only attributed to the individual sensor accuracy, but also errors in measurement time synchronisation since absolute measurements are being differenced. This problem also applies to more accurate DGPS systems. Highly dynamic vehicles and sporadic communication dropouts further amplify this effect.

To achieve the necessary accuracy, directly observed relative measurements must be used. Vision is a popular sensor selection in the aerial domain due to its availability, compact size and low weight. Vision techniques that have been

applied in an aerial context include active visual contours [3], silhouette based techniques [4] and feature extraction [2]. The downside to vision is the susceptibility to observation dropouts as a result of occlusion, incorrect feature matching, the target being outside the field of view (FOV) and uncertain lighting conditions. To negate these shortcomings and create a resilient yet accurate relative state estimator, it is important to incorporate constantly available, albeit absolute, information from sensors such as inertial, magnetic, atmospheric and GPS.

To this end, an unscented Kalman filter (UKF) [5] is proposed to fuse vehicle-to-vehicle vision measurements with information from GPS, inertial, magnetic and atmospheric sensors, located on each UAV. A UKF has several advantages over the traditional extended Kalman filter (EKF). It provides at least second-order nonlinear approximation as opposed to the first-order EKF; often difficult derivation of Jacobians is not necessary; the filter is more robust to initial errors and computation can occur in parallel. Resilience to initial error is particularly important because of the large difference in accuracy between the GPS and vision-based measurements. A downside of the UKF is that a quaternion parametrisation of the attitude results in a non-unit quaternion estimate when the mean is computed. A brute force normalisation can be made to work, but is undesirable. Instead, we use generalised Rodrigues parameters (GRPs) to represent the attitude error.

In our scenario, a leader and follower UAV are flying in formation. Visual markers are mounted on the leader at each wingtip, at the top of the tail fin and on the right of the tail plane as shown in Fig. 1. A forward facing camera is mounted on the follower and provides relative measurements to the leader's visual markers.

The estimator was first tested in a high fidelity simulated environment [6] where conditions are repeatable and the ground truth is known. In this way, the algorithm could be validated and analysed quantitatively over multiple runs. These results exhibited compelling performance improvements over both the raw relative estimate and the benchmark vision-only pose estimation algorithm. Further, improvements in the state of each individual vehicle were observed. The algorithm was also implemented on a dual-UAV system to further evaluate the estimator in real-time on an embedded system. Results from these experiments show good agreement with a vision-only relative pose algorithm and have the added benefit of being available when vision is not where the estimate gracefully degrades to the raw relative estimate. It should be noted that the vision information was logged online but processed and fused offline.

The paper is organised as follows. Section II reviews

This work is supported by the Australian Research Council (ARC) Centre of Excellence programme

D. B. Wilson, A. H. Göktoğan & S. Sukkarieh are with the Australian Centre for Field Robotics, Faculty of Aerospace, Mechanical and Mechatronic Engineering, The University of Sydney, NSW, 2006, Australia {d.wilson,a.goktogan,s.sukkarieh} at acfr.usyd.edu.au

related research and orientates the contributions of the paper. The estimation framework is described in Section III and details the vision integration. Simulation and ground-based results are presented and discussed in Section IV before conclusions and suggestions for future work are offered in Section V.

II. RELATED WORK

Vision-based navigation has gained traction recently, particularly in an airborne context, due to the onset of low cost vision sensors and small, embedded processors capable of real-time image processing. Previous work on multi-UAV visual relative navigation can be categorised as vision-only, loosely-coupled and tightly-coupled. Vision-only approaches [7], [3], [8] estimate the relative pose using visual measurements exclusively, typically by solving the PnP problem [9], [10]. Although effective, a minimum of four points are required for a unique solution and discrete switching to a GPS-based estimate is required during visual outages. A comparison to such an algorithm is provided in Section IV. Loosely-coupled methods [11], [12] extend this approach by fusing the vision-only relative pose with other onboard sensors such as GPS and inertial from one or more aircraft. This method avoids discrete switching when vision is unavailable, however a minimum number of features is still required and the observation covariance is difficult to derive.

The algorithm being proposed in this paper can be categorised as tight-coupled, where the raw visual measurements are fused directly. This adds resilience to visual outages, avoids discrete mode switching and enables information to be extracted from any number of features. Examples of tightly-coupled estimation include work in [13] which implemented a UKF to estimate the relative state, using the follower state and visual azimuth, elevation and target wingtip subtended angle. Fosbury and Crassidis [14] use an EKF to fuse follower sensor measurements with target marker visual measurements, however the target state was assumed known. These works have neglected useful leader and follower sensor information and have instead made assumptions about the vehicle dynamics. A more robust estimator that utilises all available information was proposed in [2] where the indirect form of an EKF was used to tightly fuse leader and follower sensor data with a marker-based vision system. Our work extends this by utilising a better performing unscented framework, incorporating magnetic and atmospheric measurements from both aircraft and demonstrating the algorithm operating in real-time on a payload constrained dual-UAV system.

III. MULTI-VEHICLE RELATIVE NAVIGATION

This section describes the complete multi-vehicle relative navigation algorithm. First, a brief review of single UAV INS/GPS estimation is provided for context, before being extended to the dual-UAV relative navigation case. Then, the integration of tightly-coupled visual measurements is explained which includes the observation model, feature

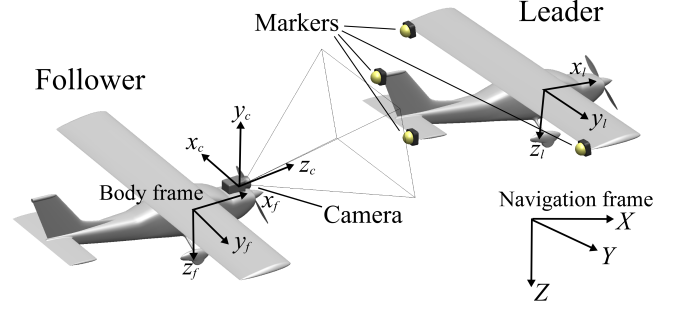


Fig. 1: Leader-follower coordinate frames and the marker based vision system.

correspondence determination and a method to quantify visual observation validity.

A. UAV Navigation

An estimate of a vehicle's state, x , is obtained by fusing sensor measurements with complementary characteristics. This is normally a recursive, two step process composed of prediction and observation. Vehicle specific models can be used in the prediction stage, however UAVs often implement standard inertial mechanisation equations [15]. The inputs to the system, u , are the body frame three-axis acceleration and angular rates, measured by an inertial measurement unit (IMU). Measurements from GPS, magnetometers and atmospheric sensors are used to correct the estimate in the observation stage. In this work a UKF provides the framework and each UAV implements its own instance.

$$x = [P \ V \ q \ a_B \ \omega_B]^T \quad (1)$$

$$u = [\tilde{a}_b \ \tilde{\omega}_{ib}^b]^T \quad (2)$$

The position, $P = [X \ Y \ Z]^T$ and velocity, $V = [v_X \ v_Y \ v_Z]^T$ are expressed in the local tangential or navigation frame, defined in north-east-down coordinates. The origin is the ground station position, P_0 in geodetic coordinates. The quaternion, $q = [q_0 \ q_1 \ q_2 \ q_3]^T$, describes the attitude of the vehicle and forms the rotation matrix, C_b^n which transforms from the body to navigation frame. The definition of C_b^n can be found in [15]. In subsequent sections, b will be replaced with l or f to represent the leader and follower body frames respectively. \tilde{a}_b and $\tilde{\omega}_{ib}^b$ are the raw IMU measurements and a_B and ω_B are the IMU accelerometer and gyro time varying bias'.

State propagation occurs using the inertial navigation prediction model given in Eq. (3).

$$\dot{x} = \begin{bmatrix} \dot{P} \\ \dot{V} \\ \dot{q} \\ \dot{\omega}_B \\ \dot{a}_B \end{bmatrix} = \begin{bmatrix} V \\ C_b^n \tilde{a} - (2\omega_{ie}^n + \omega_{en}^n) \times V + g^E \\ \frac{1}{2} \tilde{\Omega} \tilde{\omega} q \\ n_{\omega_B} \\ n_{a_B} \end{bmatrix} \quad (3)$$

g^E is the Earth gravity vector in the navigation frame and ω_{ie}^n is the rotation of Earth in the navigation frame, as defined in [15]. We assume that the navigation frame is fixed and thus ω_{en}^n is a zeros vector. a_B and ω_B are modelled as a random walk where n_{a_B} and n_{ω_B} are zero-mean Gaussian random variables. \tilde{a} and $\tilde{\omega} = [\tilde{\omega}_p \ \tilde{\omega}_q \ \tilde{\omega}_r]^T$ are the bias and noise corrected IMU measurements. n_a and n_ω are the accelerometer and gyro measurement noise terms. The gyro measurements are also corrected by the Earth's rotation. A full derivation of the INS mechanisation equations can be found in [15].

State observation occurs sequentially to account for differences in sensor sample frequencies. $h^{\text{gps}}[x, k]$ is used when GPS is available whereas $h^{\text{no gps}}[x, k]$ is implemented when magnetometer and pressure observations are available.

$$h^{\text{gps}}[x, k] = \begin{bmatrix} \bar{P}^{\text{gps}} \\ \bar{V}^{\text{gps}} \end{bmatrix} = \begin{bmatrix} P^- + C_b^n r_{\text{gps}} + n_{P_{\text{gps}}} \\ V^- + C_b^n \tilde{\omega} \times r_{\text{gps}} + n_{V_{\text{gps}}} \end{bmatrix} \quad (4)$$

$$h^{\text{no gps}}[x, k] = \begin{bmatrix} \tilde{h} \\ \tilde{\psi} \end{bmatrix} = \begin{bmatrix} h_0 - Z^- + n_h \\ \tan^{-1} \left(\frac{2(q_0 q_3 + q_1 q_2)}{1 - 2(q_2^2 + q_3^2)} \right) + n_\psi \end{bmatrix} \quad (5)$$

Where r_{gps} is the position of the GPS antenna relative to the centre of gravity (CoG). $n_{P_{\text{gps}}}$, $n_{V_{\text{gps}}}$, n_h and n_ψ are the measurement noise terms for the GPS position, GPS velocity, pressure altitude and heading respectively.

Before the raw sensor data is used within the observation models, some preprocessing is required. First, GPS geodetic measurements are converted to the navigation frame using the transformation in [15]. Then, heading, $\tilde{\psi}$ is calculated from the observed magnetic vector, \tilde{H} by first de-rotating through roll, ϕ and pitch, θ using Eq. (6) and then calculating $\tilde{\psi}$ using Eq. (7). A magnetic declination correction accounts for the local difference in magnetic and true north.

$$\begin{aligned} \tilde{M}_x &= \tilde{H}_x \cos \theta + \tilde{H}_y \sin \phi \sin \theta + \tilde{H}_z \cos \phi \sin \theta \\ \tilde{M}_y &= \tilde{H}_y \cos \phi - \tilde{H}_z \sin \phi \end{aligned} \quad (6)$$

$$\tilde{\psi} = \text{atan2} \left(-\tilde{M}_y, \tilde{M}_x \right) + \psi_{\text{dec}} \quad (7)$$

Altitude above mean sea level (MSL), \tilde{h} , is now calculated using the atmospheric pressure. First, the pressure at MSL, p_{MSL} , is estimated using Eq. (8) where p_0 is the initial pressure and h_0 is the initial MSL height, as observed by the GPS. Finally, \tilde{h} is determined using Eq. (9) with L , T_0 , M and R provided in Table I.

$$p_{\text{MSL}} = p_0 \left(1 - \frac{L h_0}{T_0} \right)^{-\frac{g M}{R L}} \quad (8)$$

$$\tilde{h} = \frac{T_0}{L} \left(1 - \left(\frac{\tilde{p}}{p_{\text{MSL}}} \right)^{\frac{R L}{g M}} \right) \quad (9)$$

Equations pertaining to the UKF prediction and update are omitted for brevity, but can be found in [5], [13], [16]. The downside to a quaternion attitude parametrisation in a UKF

Parameter	Description	Value
L	temperature lapse rate	0.0065 K/m
T_0	MSL standard temperature	288.15 K
M	molar mass of dry air	0.0289644 kg/mol
R	universal gas constant	8.31447 J/(mol · K)

TABLE I: The International Standard Atmosphere constants.

is that a unit-norm cannot be guaranteed when computing the mean. To overcome this, the method proposed in [16] was implemented. Here, we replace q with a three-dimensional GRP vector which represents attitude error and is initialised to zero. This formulation has the added benefit of reducing the state dimensionality.

To extend this formulation to the dual-UAV case, the individual vehicle state vectors are simply concatenated to form $x_{l|f}$. Equation (11) is then used to calculate the relative position $P_{l|f}$, relative velocity $V_{l|f}$ and relative quaternion $q_{l|f}$, where \otimes denotes quaternion multiplication. An alternative approach would be to estimate the relative state directly, however simulated results have shown that this formulation performs better. Further, simulated results in Section IV-A show that when coupled with visual measurements, the accuracy of x_l and x_f is improved. The inertial bias states are neglected for computational reasons; instead, each individual UAV UKF estimates these onboard and corrects the sensor measurements prior to being utilised in the relative UKF.

$$x_{l|f} = [x_l \ x_f]^T \quad (10)$$

$$\begin{bmatrix} P_{l|f} \\ V_{l|f} \\ q_{l|f} \end{bmatrix} = \begin{bmatrix} P_l - P_f \\ V_l - V_f \\ q_f^{-1} \otimes q_l \end{bmatrix} \quad (11)$$

State prediction occurs as per Eq. (3) and sensor corrections occur sequentially using the models in Eqs. (4) and (5) when new data is available.

B. Vision Integration

Relative pose estimation using vision sensors has been well researched and many valid approaches exist. Our work employs a feature based method where visual markers of a known configuration are mounted on a leader vehicle and observed by a follower. Using the set of n correspondences between the 3D marker positions, ζ_j^l and the 2D observations $\tilde{\delta}_j$, as well as the camera intrinsic parameters, the relative pose can be calculated directly. This requires $n \geq 3$ for a solution and $n \geq 4$ for a unique solution. A number of algorithms are available to solve this PnP problem, including POSIT [17] which is used as a benchmark in Section IV, the Lu-Hager-Mjolsness algorithm [18] and an efficient approach called EPnP [19].

The downside to this vision only approach is that it fails with incorrect point matching, occlusion or a target outside the FOV. These brief or prolonged measurement dropouts are highly undesirable, particularly during close proximity

operations. Alternatively, one could fuse the pose estimate from one of the aforementioned algorithms with the on-board sensor data in a loosely-coupled arrangement, however preliminary results with a fixed measurement covariance displayed inferior performance to the tightly-coupled equivalent. One reason may be that the measurement covariance is dynamic and a function of the relative pose, in addition to the pixel noise. Deriving an expression for this is difficult.

Instead, we propose a tightly-coupled approach which uses n raw 2D marker observations, $\tilde{\delta}_j = [u_j \ v_j]^T$, $j = 1, \dots, n$. In our case $n = 4$ and $n \geq 3$ is required for observability within the UKF [2], [14]. The expected observations $\bar{\delta}_j$, $j = 1, \dots, m$ are calculated by first transforming ζ_j^l from the leader's body frame to the world frame, ζ_j^f using Eq. (12). In this case the world frame is the follower's body frame.

$$\zeta_j^f = C_n^f (C_l^m \zeta_j^l + P_{l|f}) \quad (12)$$

Next, the vision sensor extrinsic parameters transform ζ_j^f to the camera frame using Eq. (13). $P_{f|c}$ and C_f^c are the translation and rotation from the followers body frame to the camera frame. C_f^c includes both the camera mounting orientation and the axes transformation.

$$\zeta_j^c = \begin{bmatrix} C_f^c & P_{f|c} \end{bmatrix} \begin{bmatrix} \zeta_j^f \\ 1 \end{bmatrix} \quad (13)$$

$\bar{\delta}_j$ is calculated using K , the camera intrinsic matrix which encapsulates the camera focal length, aspect ratio, principal point and distortion. The final vision measurement model is provided in Eq. (15) and the correction occurs sequentially.

$$\begin{bmatrix} \bar{\delta}_j \\ 1 \end{bmatrix} = K \begin{bmatrix} \zeta_{x_j}^c / \zeta_{z_j}^c \\ \zeta_{y_j}^c / \zeta_{z_j}^c \\ 1 \end{bmatrix} \quad (14)$$

$$\mathbf{h}^{\text{vision}}[x, k] = [\bar{\delta}_1 \ \bar{\delta}_2 \ \dots \ \bar{\delta}_n]^T \quad (15)$$

The vision-based observation model presented in this section updates both the relative position and orientation by assuming correct point matching. This is not always possible, particularly when the target is far and the points are difficult to distinguish from one another. Rather than neglecting such a measurement, a simpler observation model can be utilised to extract $P_{l|f}$ information. Here, the vision observation becomes the average or centroid of $\tilde{\delta}_j$, $j = 1, \dots, n$ as an approximation for the target's CoG and Eq. (12) is replaced with Eq. (16) with $m = 1$. Although relative orientation and range become unobservable, $P_{l|f}$ and $V_{l|f}$ accuracy is improved.

$$\zeta_j^f = C_n^f P_{l|f} \quad (16)$$

C. Point Matching

Before $y_{l|f}^{\text{vision}}$ can be used, we need to match the observed points $\tilde{\delta}_i$ with the projected model points $\bar{\delta}_j$, where the mean of each set is $\bar{\delta}_\mu$ and $\tilde{\delta}_\mu$ respectively. To do this, unique marker characteristics could be used, which may

include colour, size, intensity and frequency. However, in our application we have chosen to use homogeneous visual markers to simplify the MV task and instead use the marker model to match the points. To do this, we use a computationally efficient, deterministic mutual nearest point procedure [20]. Before this is implemented, we eliminate the linear translation between the point sets by subtracting the vector $(\bar{\delta}_\mu - \tilde{\delta}_\mu)$ from $\bar{\delta}_j$. This eliminates errors in relative azimuth, elevation as well as follower attitude and simplifies the matching process. The matrix Θ is then populated with the pixel distances between $\tilde{\delta}_i$ and $\bar{\delta}_j$.

$$\Theta = \begin{bmatrix} d(\bar{\delta}_1, \tilde{\delta}_1) & \dots & d(\bar{\delta}_1, \tilde{\delta}_n) \\ \dots & \ddots & \dots \\ d(\bar{\delta}_m, \tilde{\delta}_1) & \dots & d(\bar{\delta}_m, \tilde{\delta}_n) \end{bmatrix} \quad (17)$$

Where $d(\cdot, \cdot)$ is the linear pixel distance between points. $\Theta_{\text{col}}^{\min}$ and $\Theta_{\text{row}}^{\min}$ are the minimum value of each column and row of Θ , respectively and $\Theta_{\text{col}}^{\text{index}}$ contains the index of the minimum value in each column.

$$\Theta_{\text{col}}^{\min} = [\min(d(\bar{\delta}_i, \tilde{\delta}_1)) \ \dots \ \min(d(\bar{\delta}_i, \tilde{\delta}_m))] \quad (18)$$

$$\Theta_{\text{row}}^{\min} = [\min(d(\bar{\delta}_1, \tilde{\delta}_j)) \ \dots \ \min(d(\bar{\delta}_n, \tilde{\delta}_j))] \quad (18)$$

$$\Theta_{\text{col}}^{\text{index}} = [\text{index}(\Theta_{\text{col}_1}^{\min}) \ \dots \ \text{index}(\Theta_{\text{col}_n}^{\min})]$$

For a point to be valid, it must satisfy Eq. (19), that is to say a valid point in Θ must be the minimum of both its column and row. A threshold of validity can also be implemented to reject outliers and noise.

$$\Theta_{\text{col}}^{\min}[i] = \Theta_{\text{col}}^{\min}[\Theta_{\text{col}}^{\text{index}}[i]] \quad (19)$$

D. Vision Rejection

Empirical results have shown that incorrect point matching quickly leads to filter divergence, from which recovery is difficult. To protect against this, a method is proposed to detect and reject incorrectly matched visual observations.

First, the observed relative orientation is calculated by solving the PnP problem with the current point matching, using POSIT [17] for efficiency. The quaternion error, $q_{\text{error}} = [q_0 \ \varrho]$ between the predicted and observed quaternions is then determined using Eq. (20) and the magnitude of rotation error, α_{error} is calculated using Eq. (21).

$$q_{\text{error}} = q_{l|f}^{-1} \otimes q_{l|f}^{\text{posit}} \quad (20)$$

$$\alpha_{\text{error}} = 2 \cos^{-1} q_0 \quad (21)$$

Observations where α_{error} is above a threshold are rejected. By comparing the observed relative orientation with the predicted relative orientation, we can reject observations that suggest unlikely or impossible instantaneous changes in orientation that disagree with the onboard inertial sensors. This method exploits the fact that incorrect point matching has a much greater effect on the relative orientation than

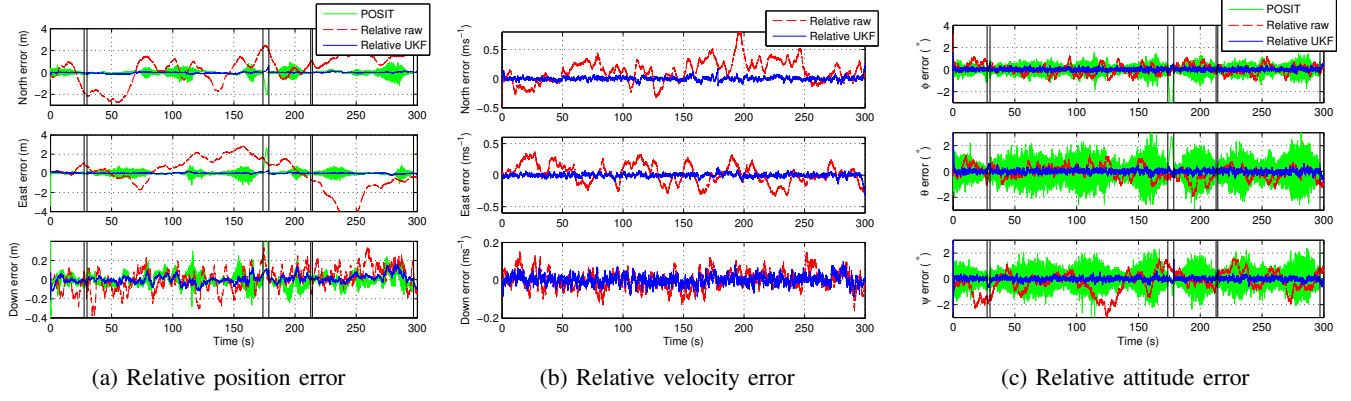


Fig. 2: Raw relative, POSIT and relative UKF estimate accuracy at 10m separation. Vision dropouts are indicated by vertical black lines.

relative position. Results from simulated flights with random point matching showed that a threshold of 20 times the mean α_{error} was conservative.

IV. IMPLEMENTATION

This section begins by presenting results from a high-fidelity multi-UAV 6DOF simulation for algorithm statistical validation and performance evaluation. Simulation is useful because it provides a repeatable environment where the ground truth is known. Here, we see sizeable performance increases in the vision based estimator when compared to the raw estimate and even the vision only relative pose estimation algorithm. Then, results from real-time, ground based experiments are provided. These experiments were designed to isolate the relative navigation problem, and demonstrate the estimation framework functioning in real-time on an embedded system.

A. Simulation Validation

In addition to facilitating algorithm validation and performance testing, the Simulink based multi-UAV simulation was designed to enable rapid algorithm design and deployment and to facilitate collaboration between multiple parties.

Each UAV utilises 6DOF nonlinear fixed wing equations of motion to propagate the state at 100Hz, given the control surface deflections, and external disturbances. The sensors are modelled using environmental and atmospheric models with white Gaussian noise, bias and cross coupling. The GPS model also incorporates Gauss-Markov noise correlation and transport delay. The actuators are modelled with first-order time lag, saturation limits and rate saturation limits. Individual UAV state estimation is handled by a 16-state quaternion UKF; a guidance module implements 'new guidance' [21] to follow a sequence of Dubins paths [22]; and cascaded, physics based PID controllers handle the low level actuation. The guidance, navigation and control (GNC) module operates at 100Hz and is configured to automatically generate C code for use on the embedded system which is detailed in the next section. This feature allows an error-free and seamless transition from the high level graphical

	Raw (1σ)	POSIT (1σ)	$x_{l f}$ (1σ)	Raw impr
$P_{l f}$ (m)				
North	2.069 (0.86)	0.353 (0.15)	0.205 (0.23)	90.1 %
East	1.901 (0.76)	1.017 (0.34)	0.427 (0.24)	77.5 %
Down	0.152 (0.11)	0.123 (0.03)	0.129 (0.07)	15.1 %
$V_{l f}$ (m)				
North	0.203 (0.09)	-	0.216 (0.25)	-6.54 %
East	0.227 (0.18)	-	0.183 (0.18)	19.4 %
Down	0.068 (0.09)	-	0.052 (0.05)	23.5 %
$Q_{l f}$ ($^{\circ}$)				
ϕ	1.148 (0.81)	0.295 (0.01)	0.173 (0.24)	85.0 %
θ	1.095 (1.12)	0.660 (0.02)	0.208 (0.17)	81.0 %
ψ	14.743 (8.90)	0.577 (0.01)	0.443 (0.85)	97.0 %

TABLE II: Raw relative, POSIT and relative UKF estimate RMSE comparison from 100 simulations.

environment to actual implementation. Additional details of this algorithm prototype and deployment workflow can be found in [6].

The vision sensor was modelled with a resolution of 1920×1080 pixels and a FOV of $70^{\circ} \times 42^{\circ}$ at 30 frames per second. A wide FOV is desirable to allow all the visual markers to be observable in a single frame during tight formation flight. The true pixel coordinates were calculated using the true vehicle states and Eqs. (12) to (14). The simulated measurements were then created by adding white noise and placing the points in a random order. Points outside the FOV were set to zero.

The simulated scenario consisted of a leader following a 700×500 m racetrack pattern with rounded corners of radius 100m. The aircraft had a commanded altitude of 100m and commanded airspeed of 25m/s. A follower UAV was commanded to maintain equivalent airspeed and altitude to the leader at a position 10m directly behind the leader. This was done by implementing the relative state estimator using onboard and remote real-time sensor information as well as a forward pointing camera. A simple pursuit based guidance strategy kept the leader within the follower's FOV

sufficiently well, however the actual separation fluctuated between 10m-15m.

Results from 100 simulated missions are summarised numerically in Table II. Here the accuracy of the vision-based relative estimator is compared with the raw estimate and a benchmark vision-only pose estimation algorithm known as POSIT [17]. Although quaternions have been estimated to avoid gimbal lock, attitude results are displayed as Euler angles for convenience. The relative estimator shows significant performance improvements over the raw estimate, particularly in horizontal position and ψ . This can be attributed to the relative inaccuracy of the GPS and magnetometers. As expected, the gains over POSIT are less but remain notable which is likely due to a smoothing effect of the vehicle inertial measurements. Improvements in the vertical position accuracy are small because the atmospheric pressure sensor provides accurate measurements. $V_{l|f}$ improvement are negligible because the GPS velocity measurements are already fairly accurate. A comparison of estimate error from one simulated run is displayed in Figs 2a, 2b and 2c. Here, we see the low precision of POSIT, particularly when observing range. As an aside, a performance comparison with an EKF implementation exhibited minor performance improvement with a favourable initial estimate and larger performance gains when the initial estimate was subject to error. These findings are consistent with [16] and [23].

During the simulated flight, occasional visual dropouts occurred. These are shown in Figs. 2a and 2c by the vertical lines which indicate the beginning and end of the dropout. These vision failures can be attributed to either the markers being outside the followers FOV or the algorithm outlined in Section III-D rejecting the point matching. The duration of these dropouts ranges from less than a second to five seconds. During this time, POSIT completely failed, whereas the relative UKF maintained its accuracy. To further test the algorithm's resilience, the filter was subject to a sustained dropout. The resulting effect on $P_{l|f}$ accuracy is displayed in Fig. 3 where vision is unavailable from 50 seconds onwards. A graceful degradation to the raw relative estimate is observed over 10-15 seconds for the horizontal position components and a few seconds for the vertical component. This difference in degradation time is a result of the higher accuracy and more frequent atmospheric pressure sensor updates, compared to the GPS. At this point, the relative filter approximates the raw relative estimate.

Simulated results also show that the vision integration can also improve individual UAV state estimates. Table III shows consistent improvement in horizontal position accuracy of around 30%. Although the accuracy appears to decrease in vertical position, the change is within 1σ bounds. Further analysis showed that $P_{l|f}$ accuracy only shows marginal degradation when less than four marker observations were available. A single observation still provides useful information. Conversely, $Q_{l|f}$ which quickly becomes unobservable with less than three marker observations.

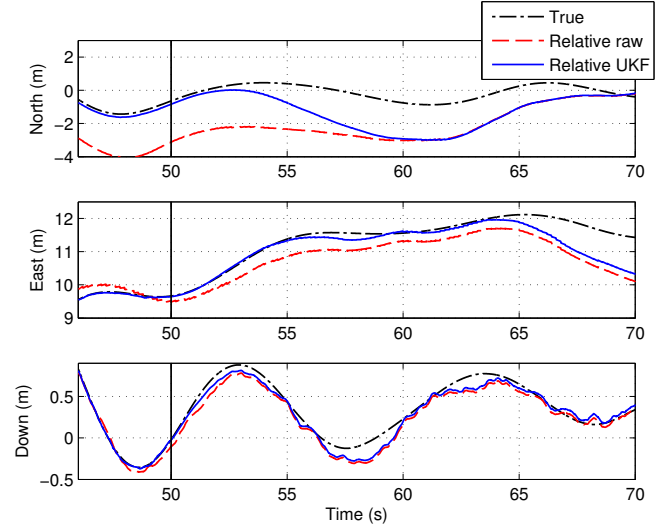


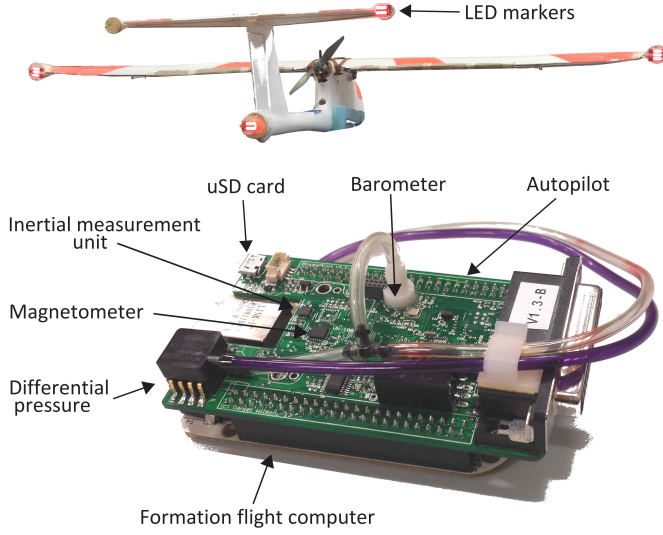
Fig. 3: Graceful degradation of the relative position when vision data is unavailable from 50 seconds.

	Leader	x_l	Impr.	Follower	x_f	Impr.
$P_{l f}$ (m)						
North	1.388	1.022	26.4 %	1.455	1.002	31.2 %
East	1.319	0.917	30.5 %	1.252	0.918	26.7 %
Down	0.153	0.189	-23.2 %	0.077	0.134	-73.2 %
$Q_{l f}$ (°)						
ϕ	0.805	0.486	39.6 %	0.504	0.459	8.91 %
θ	0.978	0.726	25.8 %	0.682	0.694	-1.82 %
ψ	13.834	2.577	81.4 %	5.641	2.512	55.5 %

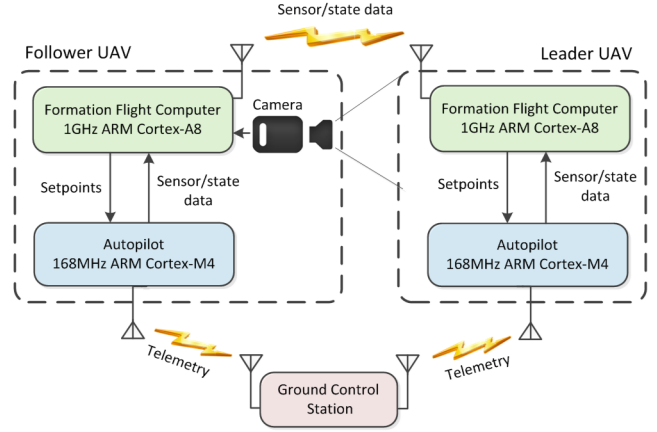
TABLE III: Comparison of individual UAV state estimates with the output of the relative estimator.

B. Ground-based Experiments

Ground based experiments were conducted on a dual-UAV system to isolate the relative navigation aspect of the problem and demonstrate the relative estimator operating in real-time on an embedded system. This system consists of two fixed-wing UAVs, an autopilot and formation flight computer (FFC) onboard each aircraft, LED markers on the leader, and a camera on the follower. A overview of this system can be found in Figs. 4a and 4b. The custom designed autopilot utilises an ARM Cortex-M4 microcontroller, running the ChibiOS/RT [24] hard real-time operating system. Making use of the ChibiOS/RT hardware abstraction layer, each autopilot receives data from accelerometers, gyros, magnetometers, GPS, barometer and a differential pressure sensors. This data is utilised within the GNC module, as outlined in the preceding section, to generate appropriate control deflections. A μ SD card allows autopilot configuration files to be read and flight-data to be logged during operation. A 900MHz wireless link to the ground control station facilitates transmission of telemetry and upload of real-time configuration. A custom ground station UI displays all flight data, facilitates data logging, allows onboard parameters to be configured



(a) Leader UAV with LED markers visible, autopilot and the FFC



(b) System architecture

Fig. 4: The dual-UAV system used in the ground experiments.

in-flight and provides a Google Earth visualisation. Onboard switching circuitry allows a ground based security pilot to take control if necessary.

The FFC is built upon a 1GHz ARM Cortex-A8 microcontroller running Linux. The role of the FFC is to estimate the relative state between the aircraft and to guide the follower to rendezvous with the leader. This is done by exchanging sensor data with the remote FFC via 900MHz wireless link; receiving sensor data from the local autopilot via serial communications and receiving visual information from a camera module. This information is then fused using the 20-state relative estimation framework outlined in Section III before appropriate low level commands are sent to the local autopilot. The relative navigation estimation and control algorithms run at 100Hz and are again auto-generated from the simulation environment. All communication, both onboard and airborne, is handled by the open source Mavlink protocol [25].

Since the core hardware and firmware is identical onboard each UAV, either vehicle can act as the leader or follower and its onboard estimate is simply the state of the remote aircraft, relative to itself. The homogeneous nature of the aircraft also means that adding a rearward facing camera to the leader is trivial. It should be noted that the complete system has been validated in autonomous flight tests, albeit not in formation and without online image processing.

The vision system consists of a camera mounted at the front of the follower and four high power LED markers positioned at the extremities of the leader as per Fig. 4a. Timestamped 1080p vision data was logged in real-time onboard the follower FFC. Marker positions were extracted and fused offline using a circular Hough transform and HSV thresholds. An intrinsic camera calibration was carried out

using the Matlab Camera Calibration Toolbox [26]. Work is currently being undertaken to transition the marker extraction work flow to an embedded module, ready for an airborne implementation.

When powered on, the aircraft automatically synchronised the clocks onboard all four processors and then agreed on the ground station position. Once complete, data exchange was initiated and the relative estimator was run on each FFC. Relative position and attitude estimates from one such experiment are shown in Figs. 5a and 5b. Here, we can see good agreement between the vision-only POSIT algorithm and the output of the relative estimator. A slight bias can be observed in the east and ψ components which indicate a slight error in the camera extrinsic calibration. Additionally, we see that POSIT fails between 23-29 seconds because less than four points are available. Here, the relative estimator is able to utilise information from even a single visual marker and only slowly degrades to the raw relative estimate when no visual measurements are available.

V. CONCLUSION

This paper has presented a vision-based unscented filter for UAV formation flight. To enable quaternions to be used with the unscented framework, the attitude error, parametrised by GRPs was estimated rather than the quaternion directly. Simulation results validated the approach and exhibited a significant relative state accuracy over the raw relative estimate and a vision-only pose estimation algorithm. Improvements in the individual UAV state estimate were also observed. Ground based experiments with a dual-UAV system demonstrated the algorithm operating in real-time on an embedded system and also showed resilience to short-term vision outages and graceful degradation over extended periods. Future work will

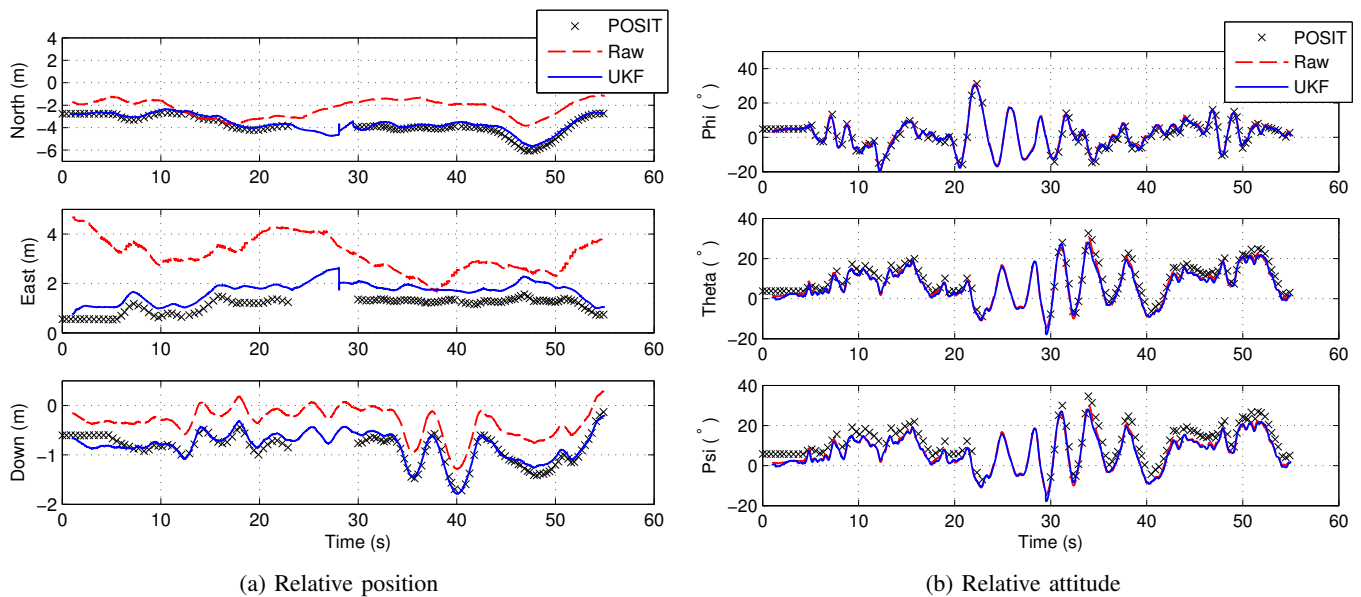


Fig. 5: Vision based relative estimation results from the dual-UAV system.

focus on transitioning the vision processing algorithm to an embedded system and implementing appropriate formation guidance to enable an airborne implementation.

REFERENCES

- [1] R. J. Sattigeri, "Adaptive estimation and control with application to vision-based autonomous formation flight," Ph.D. dissertation, 2007.
- [2] W. R. Williamson, G. J. Glenn, V. T. Dang, J. L. Speyer, S. M. Stecko, and J. M. Takacs, "Sensor fusion applied to autonomous aerial refueling," *Journal of Guidance, Control, and Dynamics*, vol. 32, no. 1, pp. 262–275, 2009.
- [3] J. Doebbler, J. Valasek, M. J. Monda, and H. S., "Boom and receptacle autonomous air refueling using a visual pressure snake optical sensor," in *AIAA Atmospheric Flight Mechanics Conference and Exhibit*. American Institute of Aeronautics and Astronautics, 2006.
- [4] S. M. Khansari-Zadeh and F. Saghafi, "Vision-based navigation in autonomous close proximity operations using neural networks," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 47, no. 2, pp. 864–883, 2011.
- [5] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, "A new approach for filtering nonlinear systems," in *American Control Conference, Proceedings of the 1995*, vol. 3, 2012, pp. 1628–1632 vol.3.
- [6] D. B. Wilson, A. H. Goktogan, and S. Sukkarieh, "UAV rendezvous: From concept to flight test," in *Australasian Conference on Robotics and Automation (ACRA)*, 2012.
- [7] Z. Mahboubi, Z. Kolter, T. Wang, G. Bower, and A. Y. Ng, "Camera based localization for autonomous uav formation flight," in *Proceedings of the AIAA@ Infotech Conference*.
- [8] J. Valasek, K. Gunnam, J. Kimmitt, J. L. Junkins, D. Hughes, and M. D. Tandale, "Vision-based sensor and navigation system for autonomous air refueling," *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 5, pp. 979–989, 2005.
- [9] K. E. Wenzel, A. Masselli, and A. Zell, "Visual tracking and following of a quadcopter by another quadcopter," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, 2012, pp. 4993–4998.
- [10] R. Mati, L. Pollini, A. Lunghi, M. Innocenti, and G. Campa, "Vision-based autonomous probe and drogue aerial refueling," in *Control and Automation, 2006. MED'06. 14th Mediterranean Conference on*. IEEE, pp. 1–6.
- [11] M. Mammarella, G. Campa, M. R. Napolitano, M. L. Fravolini, Y. Gu, and M. G. Perhinschi, "Machine vision/gps integration using ekf for the uav aerial refueling problem," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 38, no. 6, pp. 791–801, 2008.
- [12] C. Giampiero, F. Mario Luca, F. Antonio, N. Marcello, S. Brad, and P. Mario, *Autonomous Aerial Refueling for UAVs Using a Combined GPS-Machine Vision Guidance*, ser. Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics, 2004, doi:10.2514/6.2004-5350.
- [13] S. Oh and E. N. Johnson, "Relative motion estimation for vision-based formation flight using unscented kalman filter," in *AIAA Guidance, Navigation and Control Conference and Exhibit*. AIAA, 2007.
- [14] A. Fosbury and J. Crassidis, "Relative navigation of air vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 4, pp. 824–834, 2008.
- [15] R. M. Rogers, *Applied Mathematics in Integrated Navigation Systems*, 3rd ed., ser. AIAA Education Series. AIAA, 2007.
- [16] J. Crassidis and F. Markley, "Unscented filtering for spacecraft attitude estimation," in *AIAA Guidance, Navigation, and Control Conference*. AIAA.
- [17] D. F. Dementhon and L. S. Davis, "Model-based object pose in 25 lines of code," *International Journal of Computer Vision*, vol. 15, no. 1-2, pp. 123–141, 1995.
- [18] C. P. Lu, G. D. Hager, and E. Mjolsness, "Fast and globally convergent pose estimation from video images," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 6, pp. 610–622, 2000.
- [19] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnnp: An accurate o(n) solution to the pnp problem," *International Journal of Computer Vision*, vol. 81, no. 2, pp. 155–166, 2009.
- [20] M. Mammarella, G. Campa, M. R. Napolitano, and M. L. Fravolini, "Comparison of point matching algorithms for the uav aerial refueling problem," *Machine Vision and Applications*, vol. 21, no. 3, pp. 241–251, 2010.
- [21] S. Park, J. Deyst, and J. P. How, "A new nonlinear guidance logic for trajectory tracking," in *Proceedings of the AIAA Guidance, Navigation and Control Conference*. Citeseer, 2004.
- [22] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, vol. 79, no. 3, pp. 497–516, 1957.
- [23] D. R. Lee and H. Pernicka, "Vision-based relative state estimation using the unscented kalman filter," *International Journal of Aeronautical and Space Sciences*, vol. 12, no. 1, pp. 24–36, 2011.
- [24] G. Sirio, "Chibios/rt," 2013. [Online]. Available: <http://www.chibios.org>
- [25] L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, "The pixhawk open-source computer vision framework for mavs," 2011.
- [26] J. Bouguet, "Camera calibration toolbox for matlab," 2009. [Online]. Available: <http://www.vision.caltech.edu/bouguetj/calib.doc/>