Sensors **2012**, *12*, 9566-9585; doi:10.3390/s120709566

OPEN ACCESS

sensors

ISSN 1424-8220

www.mdpi.com/journal/sensors

Article

Adaptive UAV Attitude Estimation Employing Unscented Kalman Filter, FOAM and Low-Cost MEMS Sensors

Héctor García de Marina [1], Felipe Espinosa [2,*] and Carlos Santos [2]

[1] Discrete Technology & Production Automation Department, University of Groningen, Groningen, 9747 AG, The Netherlands; E-Mail: noeth3r@gmail.com

[2] Electronics Department, University of Alcala, Alcalá de Henares, Madrid 28871, Spain; E-Mails: carlos.santos@depeca.uah.es

* Author to whom correspondence should be addressed; E-Mail: espinosa@depeca.uah.es; Tel.: +34-918-856-545.

Received: 21 May 2012; in revised form: 2 July 2012 / Accepted: 9 July 2012 / Published: 13 July 2012

Abstract: Navigation employing low cost MicroElectroMechanical Systems (MEMS) sensors in Unmanned Aerial Vehicles (UAVs) is an uprising challenge. One important part of this navigation is the right estimation of the attitude angles. Most of the existent algorithms handle the sensor readings in a fixed way, leading to large errors in different mission stages like take-off aerobatic maneuvers. This paper presents an adaptive method to estimate these angles using off-the-shelf components. This paper introduces an Attitude Heading Reference System (AHRS) based on the Unscented Kalman Filter (UKF) using the Fast Optimal Attitude Matrix (FOAM) algorithm as the observation model. The performance of the method is assessed through simulations. Moreover, field experiments are presented using a real fixed-wing UAV. The proposed low cost solution, implemented in a microcontroller, shows a satisfactory real time performance.

Keywords: UAV navigation; attitude estimation; unscented Kalman filter; attitude heading reference system; fast optimal attitude matrix

1. Introduction

Nowadays the industry is employing UAVs for mobile missions, specially in vigilance, monitoring and inspection scenarios [1,2]. The stability and navigation of these autonomous vehicles need knowledge of its attitude angles [3,4]. Although these angles can be measured using a conventional Inertial Navigation System (INS), modern MEMS technologies are offering light and low cost solutions, which are more appropriate for the reduced space available for embedded systems in lightweight UAVs. However, these sensors are less accurate and more noisy than traditional mechanical or optics solutions. These inaccuracies result from the large biases and scale factors due to the sensors themselves and environmental effects such as temperature or mechanical vibrations. Moreover, inertial MEMS sensors do not provide attitude measures directly and the information proffered by them has to be post-processed through data fusion techniques. For this sensor fusion, the common solution in the literature is the Kalman filter. For fitting the highly non-linear kinematics model related to the attitude, the UKF is potentially better solution than the Extended Kalman Filter (EKF). The UKF is a *sigma-point* filter based solution, which is currently gaining growing interest in the attitude estimation area. Space application works based on UKF are showing more robustness and accuracy than the EKF [5]. Although their computational cost is still higher than EKF, there are new sigma-points algorithms aiming to reduce this cost, making it comparable with EKF in attitude estimation [6].

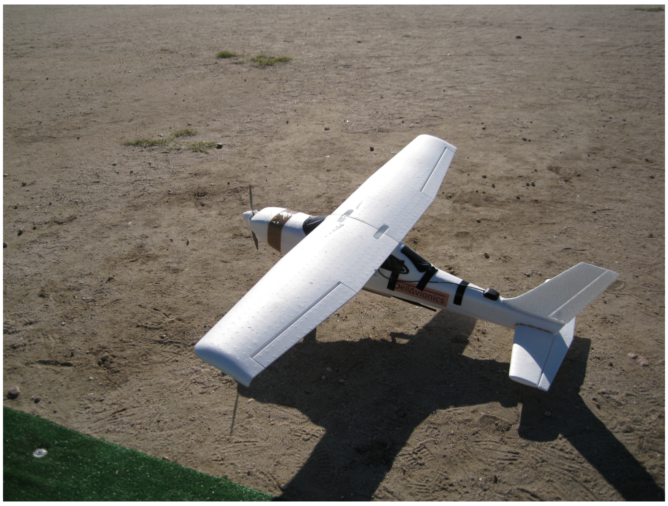
Three attitude angles can be represented in the Direct Cosine Matrix (DCM), which can be observed employing vectorial algorithms solving the Wahba's problem [7], such as Singular Value Decomposition (SVD), Three Axis Attitude Determination (TRIAD) [8] or FOAM [9]. For instance, TRIAD has been applied for attitude estimation in lightweight UAVs in [10], using as observation vectors the accelerometer and magnetometer readings. However, due to environment factors or mission's stage, the trust in sensors has to be adapted in a flexible way according to the actual scenario, therefore the observation model has to be adaptive. This paper introduces FOAM as observation model, which allows to set weights explicitly to the sensor readings, presenting clear advantages in the attitude estimation comparing with previous works that employ TRIAD and other sensor fixed-trust observations models.

Using a quaternion formulation, which is a conventional way to deal with the attitude of aerial vehicles [11], the DCM terms can be easily handled. The quaternion approach is widely used in AHRSs because it avoids the gimbal lock problem. With respect to numerical stability, quaternions are easier to propagate than the angles themselves.

For systematic and non-risky work, a simulation environment has been developed. The core of the simulation is the X-Plane 9 simulator, which is certified by the Federal Aviation Administration (FAA) for subsonic terrestrial flight. For realism [12], perturbations in the form of high-frequency noise, sensor latencies, biases, scale factors and misalignment are considered in the simulator data.

During these simulations a maximum error of 1.0° is imposed on the pitch and roll attitude angles, and a maximum error of 4.0° on the yaw attitude angle. Under those requirements, which are standard in the industry [13], the error tolerances in the different signals are obtained through a Monte Carlo analysis. Finally, the algorithm is tested in a real experiment using a fixed-wing UAV with 2 meters wingspan. The modified aircraft to work as a UAV is the Mentor from Multiplex company. The UAV used in our experiments is shown in Figure 1. An in-house autopilot is used to test the estimation algorithm presented in this paper. The results of the solution are compared with different independent systems.

Figure 1. Mentor Multiplex aircraft modified by the authors to be a UAV.



Summarising, the second section of the paper presents the state of the art about AHRSs. The following section is devoted to the formulation of the problem, including the kinematic model and how to adapt the observation model depending on the UAV situation using the FOAM algorithm. Then comes a section describing the whole solution using the UKF and FOAM. The fifth section deals with data obtained by simulation and it is compared with previous work where TRIAD algorithm was employed. Then, field experiments are presented to fully evaluate the proposed AHRS. In the final section of the paper, some conclusions are drawn.

2. Background and Related Work

The core of a well designed Kalman filter relies on an accurate model of the plant as well as the noise. Fortunately the attitude kinematics based on rate gyros is well known and corresponds to experimental results. However these equations are highly non-linear. This issue leads to the employment of the Unscented Transformation (UT) and the UKF, which have been introduced by Julier and Uhlmann [14]. Navigation algorithms employing this technique were further explained, with examples, by Wan and Merwe [15].

Most of the attitude estimation works trust in sensors in a fixed way throughout vehicle's mission [16], thus not allowing a real time adaptation depending on the actual environment factors, which is a common issue in small and lightweight UAVs.

A well-known problem with gyroscopes is the random walk bias which can not be identified at the calibration process. Nowadays, the random walk bias in gyroscopes based on MEMS technology can not be ignored even for short time missions. Therefore, different sensors have to be used to correct these random walk biases in real time. For instance, in [17] they are corrected using three-axis accelerometers. In [18] the use of eight accelerometers in a new configuration is proposed for measuring angular velocities in small UAVs.

Other approaches rely on magnetometers to estimate the yaw angle in helicopters [19]; or in Global Position System (GPS) to estimate both the position and the attitude in fixed-wing UAVs [20].

Alternatively, other papers propose not to use gyroscope at all in conventional aircrafts, but several GPS receivers instead [21].

The FOAM algorithm was introduced by Markley in [9] to measure the DCM of a spacecraft based on vectorial observations, but in this paper, it is used to estimate the quaternion components of the attitude. Actually, the TRIAD algorithm, introduced by Shuster and Oh in [8], is a particular solution of FOAM. With only two sensors available as sources, FOAM is light enough to be implemented in a microcontroller and a remarkable advantage is that it allows to tune the trust in the involved sensors through weights in execution time.

In the present paper, the authors propose an attitude estimator using the UKF and the FOAM algorithm as adaptive observation model. Gyroscopes, magnetometers and accelerometers based on MEMS technology are involved in the proposed estimator. The validation of the algorithm is done by both simulations and field experiments.

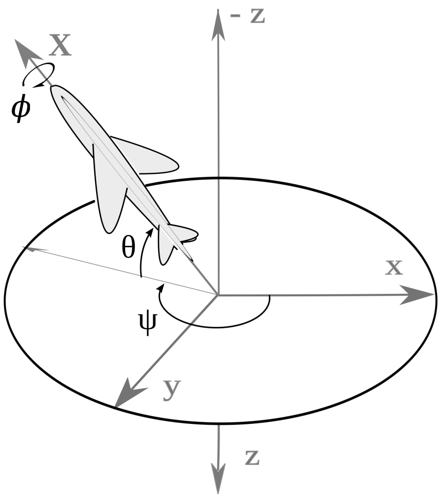
3. Problem Formulation

Despite the specific aerodynamic coefficients of the vehicle, it is possible to determine the attitude angles evolution from a kinematic model. In this section we derive the mathematical formulation of the AHRS problem in a UAV equipped with a three-axis gyroscope, a three-axis accelerometer and a three-axis magnetometer.

3.1. AHRS Kinematic Model

The attitude angles describe the aircraft body-axis orientation in north, east, and down coordinates, *i.e.*, in longitudinal, lateral and normal coordinates, with respect to the local tangent plane to the Earth and true north. Here θ is the pitch angle, ϕ the roll angle and ψ is the yaw angle according to Figure 2. The angular velocity vector expressed in body frame includes three components: P the roll rate, Q the pitch rate and R the yaw rate; and it is related to the Earth frame by the transformation given by the kinematics Equation (1).

Figure 2. Axes and coordinate definitions.



$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \tan\theta\sin\phi & \tan\theta\cos\phi \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{bmatrix} \begin{bmatrix} P \\ Q \\ R \end{bmatrix} \tag{1}$$

Integrating Equation (1) gives numerical instability and could be gimbal locked. For this reason, a quaternion formulation to represent the attitude is preferred:

$$q = q_0 + q_1 i + q_2 j + q_3 k \qquad \sum_{i=0}^{3} q_i = 1 \tag{2}$$

where the quaternion norm is the unity and their components from attitude angles are:

$$q_0 = \cos\phi'\cos\theta'\cos\psi' + \sin\phi'\sin\theta'\sin\psi' \tag{3}$$

$$q_1 = \sin\phi'\cos\theta'\cos\psi' - \cos\phi'\sin\theta'\sin\psi' \tag{4}$$

$$q_2 = \cos\phi'\sin\theta'\cos\psi' + \sin\phi'\cos\theta'\sin\psi' \tag{5}$$

$$q_3 = \cos\phi'\cos\theta'\sin\psi' - \sin\phi'\sin\theta'\cos\psi' \tag{6}$$

where $\phi' = \phi/2$, $\theta' = \theta/2$, and $\psi' = \psi/2$.

The kinematics Equation (1) can be rewritten in linear form using quaternion components:

$$\begin{bmatrix} \dot{q_0} \\ \dot{q_1} \\ \dot{q_2} \\ \dot{q_3} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -P & -Q & -R \\ P & 0 & R & -Q \\ Q & -R & 0 & P \\ R & Q & -P & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{7}$$

The discrete formulation of Equation (7) has a closed form [22], this representation keeps the quaternion norm to the unity and it is numerical stable:

$$q(k+1) = \left(I\cos\frac{\|\Delta\boldsymbol{\omega}\|}{2} + \sin\frac{\|\Delta\boldsymbol{\omega}\|}{2}\|\Delta\boldsymbol{\omega}\|\Omega \right) q(k) \tag{8}$$

where $\|\Delta\boldsymbol{\omega}\| = \frac{1}{2}\sqrt{(P\Delta t)^2 + (Q\Delta t)^2 + (R\Delta t)^2}$, I is the identity matrix, and

$$\Omega = \frac{1}{2} \begin{bmatrix} 0 & -P & -Q & -R \\ P & 0 & R & -Q \\ Q & -R & 0 & P \\ R & Q & -P & 0 \end{bmatrix} \tag{9}$$

Additionally, it is useful to formulate the DCM using quaternion components and the attitude angles from the DCM terms:

$$\text{DCM} \equiv A = \{c_{ij}\} = \begin{bmatrix} A_1 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & A_2 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & A_3 \end{bmatrix} \tag{10}$$

where $A_1 = q_0^2 + q_1^2 - q_2^2 - q_3^2$, $A_2 = q_0^2 - q_1^2 + q_2^2 - q_3^2$, and $A_3 = q_0^2 - q_1^2 - q_2^2 + q_3^2$. Then,

$$\theta = -\arcsin(2(q_1q_3 - q_0q_2)) \tag{11}$$

$$\phi = \operatorname{atan2}(2(q_2q_3 + q_0q_1), q_0^2 - q_1^2 - q_2^2 + q_3^2) \tag{12}$$

$$\psi = \operatorname{atan2}(2(q_1q_2 + q_0q_3), q_0^2 + q_1^2 - q_2^2 - q_3^2) \tag{13}$$

where $\operatorname{atan2}$ is the four-quadrant version of the inverse tangent function, and $\arcsin$ is the arcsine function.

3.2. Gyros Integration Problem

The three-axis gyroscope measures the angular velocities, and for obtaining the attitude angles, the gyros can be integrated using Equation (7). However, even if we ignore the sensor noise, the gyros suffer the random walk bias, increasing their integration error in every step.

Fortunately, for a MEMS gyroscope in normal conditions (not extremal temperature or pressure variation), this random walk varies really slowly, compared with the integration step, throughout the UAV mission. Therefore this bias for the gyroscope can be modeled as:

$$\dot{\mathbf{b}} = 0 \quad \text{with} \quad \mathbf{b} = \begin{bmatrix} b_x & b_y & b_z \end{bmatrix}^T \tag{14}$$

Denoting the angular velocity vector $\boldsymbol{\omega} = \begin{bmatrix} P & Q & R \end{bmatrix}^T$, if the angular velocities from the gyros are $\boldsymbol{\omega}_s$, they can be corrected using the bias as:

$$\boldsymbol{\omega} = \boldsymbol{\omega}_s - \mathbf{b} \tag{15}$$

3.3. The FOAM Algorithm

The Wahba's problem [7] seeks for the optimal orthogonal matrix, which minimizes the next cost function:

$$J(A) = \frac{1}{2}\sum_{k=1}^{N} a_k|\hat{\mathbf{W}}_k - A\hat{\mathbf{V}}_k|^2 \tag{16}$$

where $\hat{\mathbf{W}}_{\mathbf{k}}, k = 1 \ldots N$ is a set of N vectors measured with respect to body frame, $\hat{\mathbf{V}}_{\mathbf{k}}, k = 1 \ldots N$ is a set of N vectors with respect to the navigation frame, and $a_k, k = 1 \ldots N$ are N positive weights.

The cost function Equation (16) can be rewritten as [23]:

$$J(A) = \lambda_0 - g(A) = \lambda_0 - \text{tr}[AB^T] \tag{17}$$

where

$$\lambda_0 = \sum_{k=1}^{N} a_k \quad \text{y} \quad B = \sum_{k=1}^{N} a_k\hat{\mathbf{W}}_k\hat{\mathbf{V}}_k^T \tag{18}$$

and *tr* is the trace operator.

The two observations for the DCM determination in the UAV are the gravity vector and the Earth's magnetic field. The module of the gravity vector is the most stable, but it is not measurable directly; the accelerometers have to be compensated due to inertial effects and vibrations. The Earth's magnetic field is not affected by inertial effects, but it could be perturbed by standing close to the Earth's surface or environment as buildings, power lines, *etc.* Moreover, the Earth's magnetic field model does not have the same accuracy near of the Earth's surface than the gravity model. This is why it is necessary to modify the weights at Equation (16) throughout the flight. FOAM allows to tune these weights in run time.

The optimum A matrix in Equation (16) is based on the next characteristic equation [9]:

$$0 = (\lambda^2 - \|B\|_F^2)^2 - 8\lambda \text{det} B - 4\|\text{adj}B\|_F^2 \tag{19}$$

The attitude matrix is given by:

$$A_{\text{opt}} = (k\lambda_{\text{max}} - \text{det}B)^{-1} \big((B + \lambda_{\text{max}}\text{adj}B^T - BB^TB\big) \tag{20}$$

where $k = \frac{1}{2}(\lambda_{\text{max}}^2 - \|B\|_F^2)$.

If only two observations are available, the determinant of B is zero, simplifying the before expression to:

$$\begin{aligned} \lambda_{\text{max}} &= \sqrt{a_1^2 + a_2^2 + 2a_1a_2\left((\mathbf{w_1}\cdot\mathbf{w}_2)(\mathbf{v}_1\cdot\mathbf{v}_2) + |\mathbf{w}_1\times\mathbf{w}_2||\mathbf{v}_1\times\mathbf{v}_2|\right)} && (21) \\ A_{\text{opt}} &= \mathbf{w}_3\mathbf{v}_3^T + (a_1/\lambda_{\text{max}})\left(\mathbf{w}_1\mathbf{v}_1^T + (\mathbf{w}_1\times\mathbf{w}_3)(\mathbf{v}_1\times\mathbf{v}_3)^T\right) \\ &\quad + (a_2/\lambda_{\text{max}})\left(\mathbf{w}_2\mathbf{v}_2^T + (\mathbf{w}_2\times\mathbf{w}_3)(\mathbf{v}_2\times\mathbf{v}_3)^T\right) && (22) \end{aligned}$$

where, $\mathbf{w}_3 = (\mathbf{w}_1\times\mathbf{w}_2)/|\mathbf{w}_1\times\mathbf{w}_2|$, $\mathbf{v}_3 = (\mathbf{v}_1\times\mathbf{v}_2)/|\mathbf{v}_1\times\mathbf{v}_2|$, $\mathbf{w}_{1,2}$ are the measured column vectors with respect to the body frame (accelerometers and magnetometers), and $\mathbf{v}_{1,2}$ are the two reference vectors in navigation frame (Earth's gravity and magnetic field model).

During a flight, sometimes the accelerometers are more reliable than magnetometers and vice versa. The criteria chosen to assign the weights are based on how close is the measured vector's module to the model vector's module. The measured magnetic field module can be perturbed by standing close to the Earth's surface, power lines, *etc.* The module readings from the accelerometer, once the centripetal effect had been compensated, will highlight if linear accelerations are present, such as a take-off or aerobatic maneuver. Considering these effects, the weights can be adapted by:

$$\begin{aligned} a_k =& 1 - \kappa_k\left|1 - \frac{\|\mathbf{v}_k\|}{\|\mathbf{w}_k\|}\right| \\ \text{If} \quad a_k \leq& 0 \quad \text{then} \quad a_k = 0.001 \end{aligned} \tag{23}$$

where κ_k is a gain that indicates the weight sensibility due to the ratio between the module measured and modeled. This gain has been determined experimentally, showing a good behaviour of 1 for the magnetometer, and 2 for the accelerometer. This is because the accelerometer has to be compensated to obtain the gravity vector, while the magnetometer does not need any compensation.

The trust in the two sensors are related between their weights, so we can establish a weight maximum value of 1, and a minimum value of 0.001. Weights must be always positive and different from zero. In the last case, the FOAM algorithm is reduced to the particular case of TRIAD.

4. Unscented Kalman Filter as Sensor Fusion Core

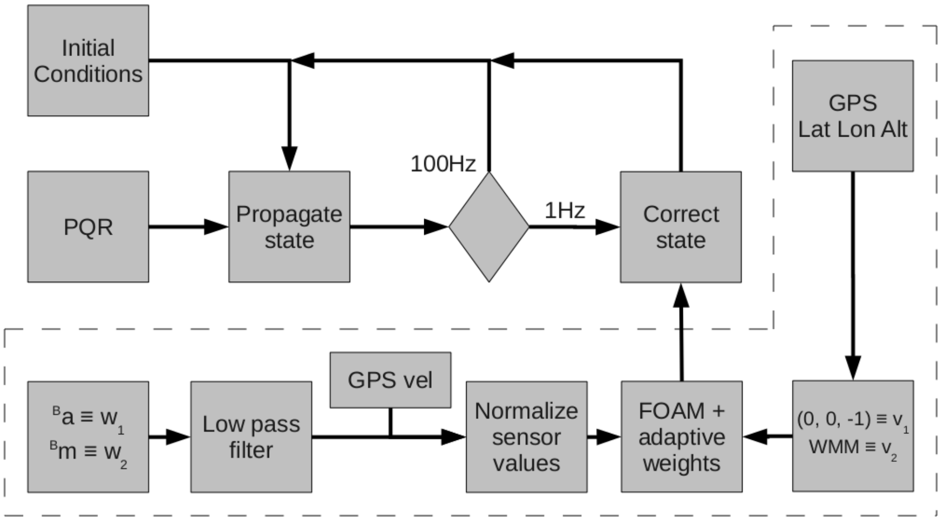
The sensor fusion algorithm is implemented as a two-step *propagator/corrector* filter. It is desirable to run each step as many times as possible, however, the frequencies at which they are going to run are limited by different factors. For the propagation step, the limiting factor is the computation time; we chose a frequency of 100 Hz. This frequency is large enough for our experiments, since we do not expect angular velocities larger than 300 deg/s.

The criteria for selecting the frequency for the correction step are based on the random walk bias of our MEMS gyro, which is 0.007 deg/s (1σ). Assuming that we know perfectly the bias at the beginning, it takes about 9.75 s for diverging 1 degree from the true value. Assuming that the bias is always growing, which could be possible, for the worst case condition (taking 3σ for the random walk) it is obtained:

$$\frac{3 \times 0.007 \text{ deg/s}}{2} x^2 = 1 \text{ deg} \rightarrow x = 9.75 \text{ s} \tag{24}$$

In our case, the correction step is limited by the GPS velocity with respect to ground at 1 Hz sample rate, which fits safely with the before requirement. This GPS velocity is employed to compensate the centripetal acceleration in the accelerometers readings. The diagram of the algorithm is described in Figure 3. It can be noted how the FOAM information feeds the correction step. The area surrounded by dashed lines contains the elements involved in the FOAM calculations. The leftmost block represents an access to the sensors to measure ${}^{b}a$ and ${}^{b}m$, which correspond to the accelerations and magnetic field measurements in the body reference frame respectively. The signals are then filtered and the GPS velocity is used to subtract the centripetal accelerations. On the right side of the diagram, the block denoted as *World Magnetic Model* uses an harmonic spherical model to obtain the magnetic field vector at the position of the UAV. This vector is used as one of the references in the FOAM algorithm. The references vectors are normalized, thus the model of the gravity reference vector is $\begin{bmatrix} 0 & 0 & -1 \end{bmatrix}^T$.

Figure 3. Algorithm block diagram. WMM is the World Magnetic Model and the GPS velocity is employed for subtract the centripetal acceleration.



This section describes the design of the filter under the considerations given above. Subsection 4.1 details the equations involved in the propagation loop whilst Subsection 4.2 deals with those of the correction loop.

4.1. State Vector and Process Model

The state vector selected for the UKF is:

$$x(k) = \begin{bmatrix} q_0(k) & q_1(k) & q_2(k) & q_3(k) & b_x(k) & b_y(k) & b_z(k) \end{bmatrix}^T \tag{25}$$

where q_i are the quaternion components and b_j are the gyroscopes' biases. These are assumed to be Gaussian Random Variables (GRVs). Their process model is given by Equations (8),(14), respectively. This model identifies the gyroscopes' biases while propagates the vehicle's attitude compensating the gyroscopes readings. The detailed description of the UKF at its propagation stage can be found in [10,15].

The discrete-time Equation (8) assumes that the angular velocities remain constant during the discretization period. Hence, its process noise covariance matrix should be close to zero (but not zero). For the simulation and experimental results shown in this paper $Q_q = 1 \times 10^{-6} \cdot I_{4x4}$, where $I_{4\times4}$ is the 4×4 identity matrix, and subscript q refers to the quaternion model.

The process noise covariance associated to the gyroscope biases Q_b is the 3×3 zero matrix. The rationale for this is the same that explains Equation (14) in Subsection 3.2.

4.2. Correction Equations and Observation Model

Even though the FOAM algorithm gives the nine terms of the DCM (see Equation (10)), only four of them are needed to calculate the attitude angles. Hence, the observation function has been designed to measure these four terms.

To determine the pitch and roll angles, the terms are the X and Y components of the Z earth vector expressed in the body frame:

$$^{b}Z_{Ex} \equiv c_{13} = 2(q_1 q_3 - q_0 q_2) \tag{26}$$

$$^{b}Z_{Ey} \equiv c_{23} = 2(q_2 q_3 + q_0 q_1) \tag{27}$$

For the yaw angle, the terms are the X and Y components of the X body vector expressed in the Earth reference frame:

$$^{E}X_{bx} \equiv c_{11} = (q_0^2 + q_1^2 - q_2^2 - q_3^2) \tag{28}$$

$$^{E}X_{by} \equiv c_{12} = 2(q_1 q_2 + q_0 q_3) \tag{29}$$

Therefore, the observation model is given by the following Equation:

$$H(x_k) = \begin{bmatrix} c_{13} & c_{23} & c_{11} & c_{12} \end{bmatrix}^T \tag{30}$$

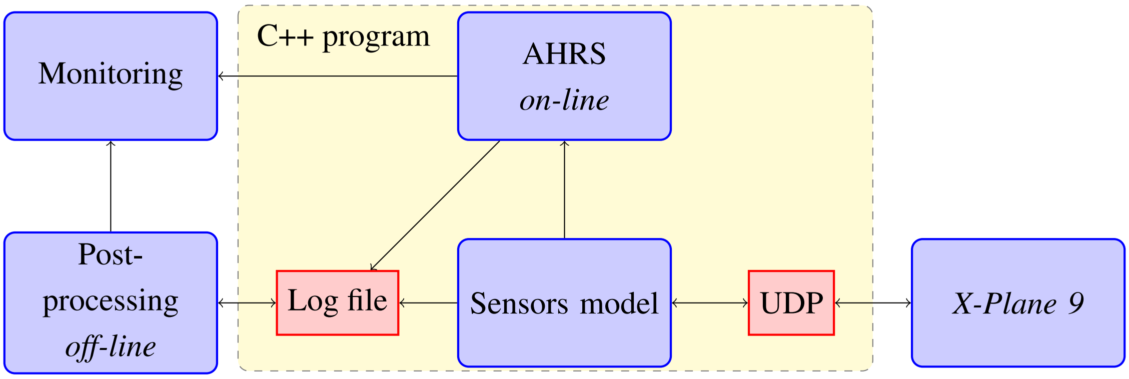
According to the FOAM algorithm two vector pairs are needed to compute the terms of the DCM. Each pair consists of a measure or observation and a reference vector. In our case, these pairs are the magnetic field, and the acceleration of the UAV. The value of the measurement noise covariance can be derived from the nominal values of the involved sensor errors. This derivation is described thoroughly in [8].

As in the propagation stage, the detailed UKF correction stage can be found in [10,15].

5. Simulations Results

Since real experiments might imply crashes, some previous simulations have been carried out. For this, we developed a simulation framework, which consists of four different parts. The core of the simulation is the X-Plane 9 software. The other three parts are: the plug-in code for X-Plane 9, the realistic model of the sensors, and finally, the *online or off-line* processing of the sensors with the algorithm. The idea is to integrate a six-degrees-of-freedom aerodynamic model, provided by X-Plane with a realistic model of the sensors we are using. The plug-in code is just an UDP interface to link each other. Figure 4 shows the diagram block implemented to the *on-line* processing and for the post-processing.

Figure 4. Block diagram of the simulation environment.



X-Plane 9 includes different aircraft models. Its default radio control model is very similar to our UAV, so no additional modifications are needed. This is the model used in the simulations of this section. Because the airplane model is manually piloted, the navigation is not in any closed-loop; therefore if a tuning process is required, it could be done *off-line* with any software tool such as SciLab or Matlab. Also, the interface can be easily modified for Hardware in the Loop purposes, the class responsible for logging only has to write to a Serial Port instead of a file.

The purpose of the simulations is to study the effects of sensor noise, bias, latencies and filter convergence. Therefore, the model of the sensors focuses on these aspects:

- The GPS signal is delayed 1 s.
- The signals from the gyroscopes are biased with angular random walk, and corrupted with white noise.
- Accelerometers are biased with angular random walk and corrupted with colored noise, focusing in high frequencies.

- Magnetometers are biased with angular random walk and corrupted with white noise.
- Filter initial conditions are far from the actual initial values.

A first simulation challenge is to study the tolerances of the AHRS to bias and noise magnitude. According to standard procedures for Guidance and Control in aeronautics, a maximum error of $1.0°$ in the estimation of pitch and roll angle, and $4.0°$ for yaw angle, are imposed. It is assumed that with this error it is possible to do closed-loop control. This is covered in the first part of this section. The second part assesses the performance of the UKF using real values for the biases and noises magnitude of our sensors in a scenario where aerobatics are performed.

5.1. Sensor Error Tolerances in the AHRS

A Monte Carlo analysis of the tolerances was made, supported by a batch of simulation experiments. Each of the experiments specifies different values of biases and noise magnitudes, which were drawn from a Gaussian distribution. The values are shown in Table 1 versus the biases and random errors present in the UAV sensors of our real experiments, which are ADIS16405 from Analog Devices for IMU and magnetometers, and Lassen IQ from Trimble for the GPS.

Table 1. UKF tolerances and actual onboard sensors: Maximum bias and random error standard deviation. It can be noted how the algorithm based on UKF meets the onboard values.

Measurement	**UKF tolerance**	**Sensor Bias error**	**UKF tolerance**	**Sensor Random Error**
Roll rate, P	±11.38°/s	±3.00°/s	±8.82°/s	±1.00°/s
Pitch rate, Q	±11.38°/s	±3.00°/s	±8.82°/s	±1.00°/s
Yaw rate, R	±11.38°/s	±3.00°/s	±5.32°/s	±1.00°/s
Accelerometers	±0.3 m/s^2	±0.05 m/s^2	±0.7 m/s^2	±0.009 m/s^2
Magnetometers	±10.63 mG	±4.00 mG	±22.53 mG	±1.25 mG
GPS Velocity	±2.46 m/s	±0.5 m/s	±2.35 m/s	±1.5 m/s

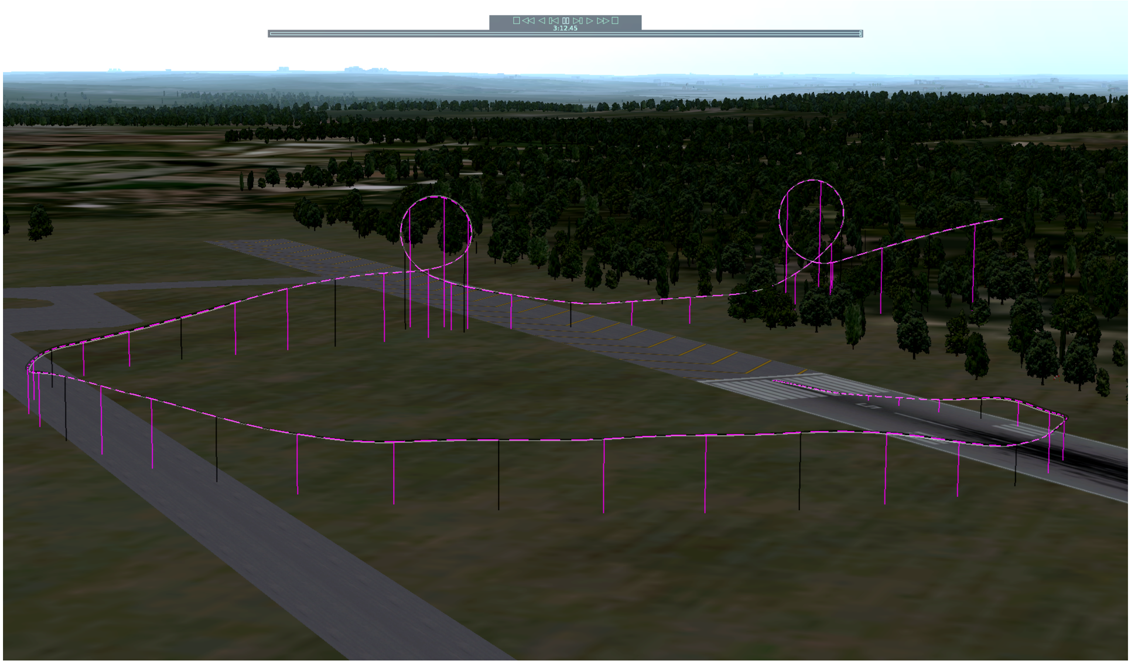
As it can be appreciated the algorithm is more sensitive to errors in R than it is to errors in P and Q. This is because the information in the yaw angle is only provided by one of the sensors, the magnetometer. In contrast, the information in pitch and roll angles are provided by the accelerometer and the magnetometer.

The results show that the attitude estimator algorithm tolerances are large enough to embrace the actual UAV sensors on board. Therefore, the algorithm is suitable for being used with MEMS sensors.

5.2. FOAM and TRIAD Comparison in an Aerobatics Simulation Taking Onboard MEMS Sensors Error Values

In this section we compare the FOAM algorithm against TRIAD in an aerobatics scenario. Figure 5 shows this scenario performing two loopings. The filter is reset just at take-off time, so there is not a stationary situation for helping to filter convergence.

Figure 5. Aerobatics scenario with two *loopings*.



For this simulation, the real biases and noise magnitudes were extracted from our sensors' data sheet and they can be consulted in Table 1.

Figure 6. Estimation of the biases of the gyroscopes, employing TRIAD at the left and FOAM at the right during the aerobatics simulation shown in Figure 5.

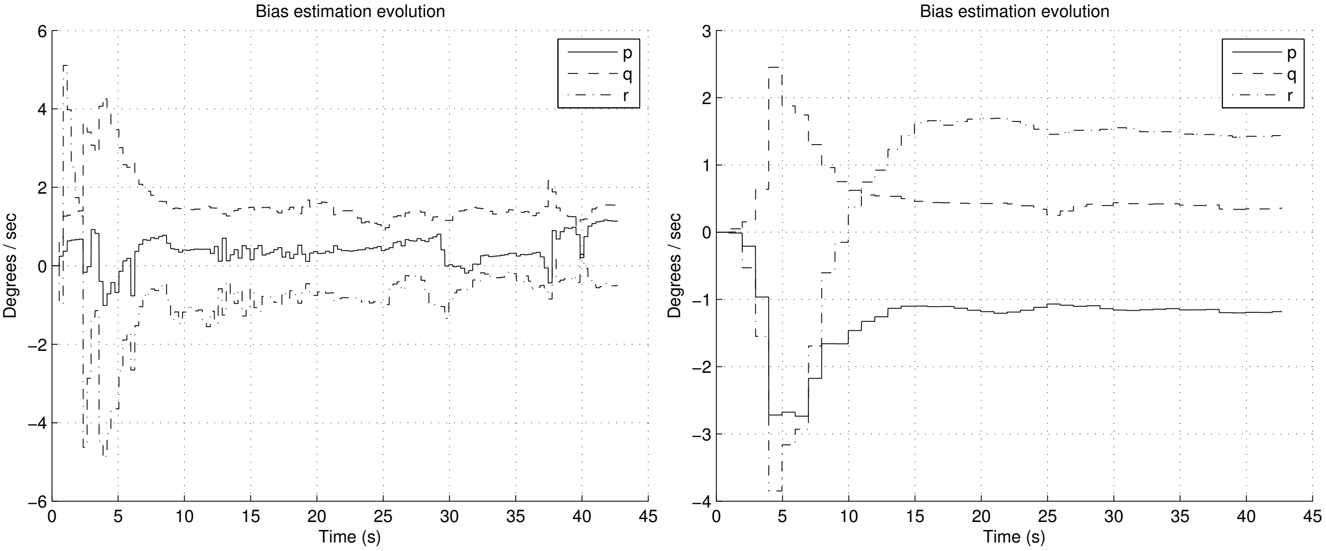


Figure 6 shows the estimation of the biases. It can be noted how the convergence is smoother with FOAM in the right graph than with TRIAD in the left graph. Moreover, since in TRIAD the accelerometers are always trusted in a fixed way, wrong accelerometers corrections lead to a wrong estimation of the biases, which finally gives a less confident estimation in the attitude angles as result. This is shown in Figures 7 and 8, after the convergence time (around the fifteenth second) in the left graphs, the true and estimated value do not converge properly due to the wrong gyroscopes' bias estimation. It can be noted that the right graphs have better RMSE, highlighting that the RMSE also includes the transitory period.

Figure 7. Pitch angle estimation during the aerobatics, employing TRIAD in the left and FOAM in the right during the aerobatics simulation shown in Figure 5.

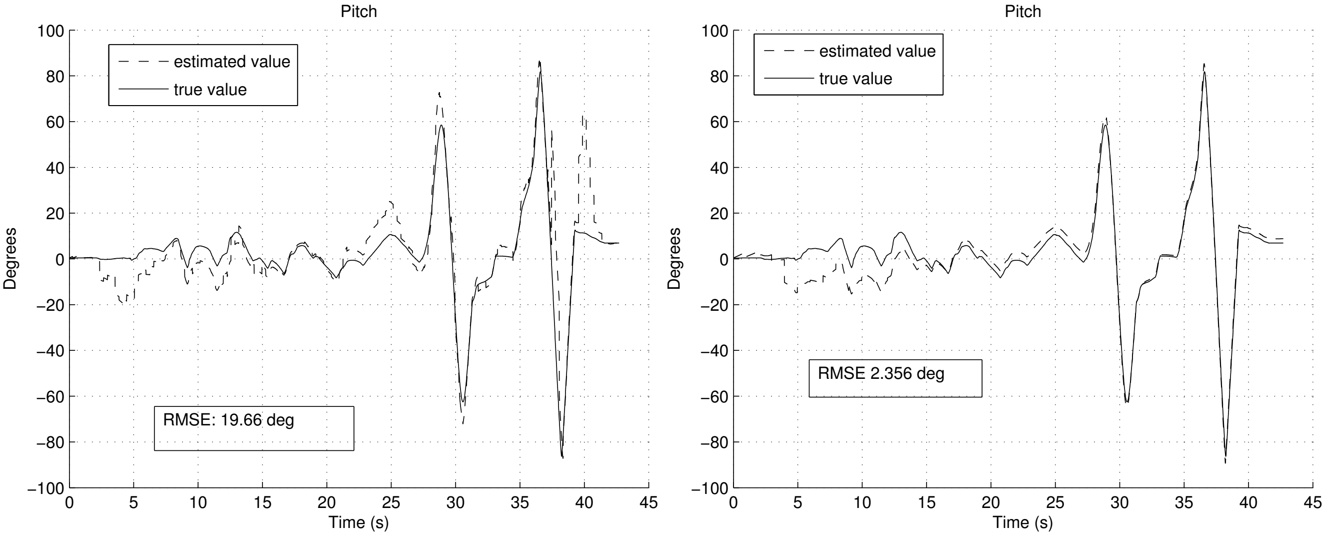
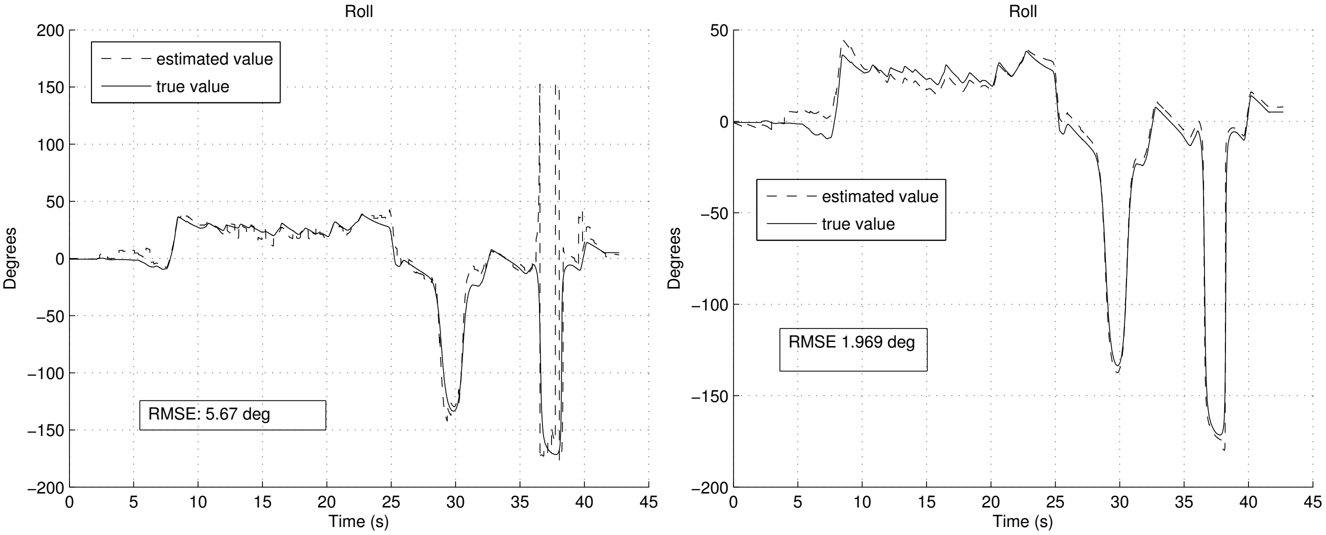
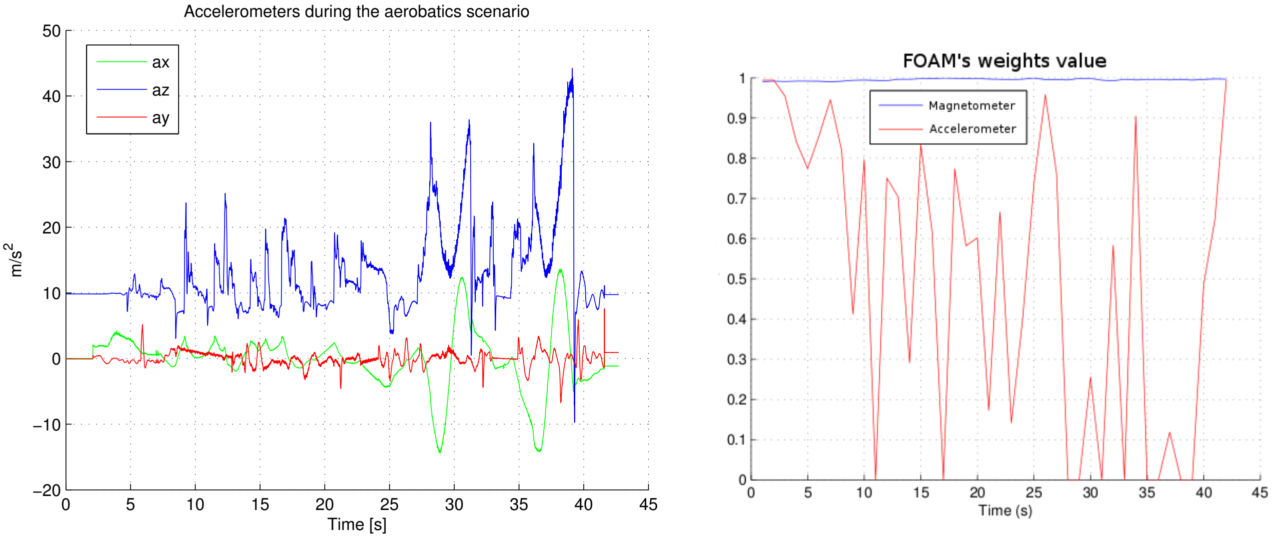


Figure 8. Roll angle estimation during the aerobatics, employing TRIAD in the left and FOAM in the right during the aerobatics simulation shown in Figure 5.



Right graph in Figure 9 shows the evolution of the FOAM's weights during the aerobatics. As it can be appreciated, accelerometer weight has a low value when an aerobatic is being performed. The aerobatic can be analyzed in the left graph in Figure 9, where the norm of the three accelerations are not the gravity value. This adaptation in weights avoids disturbing the estimation of the roll and pitch angles by the AHRS.

Figure 9. Accelerometers readings and evolution of the FOAM weights during the aerobatics simulation shown in Figure 5.



However, the final bias values even in FOAM are not stationary due to the random walk noise in gyroscopes. Nevertheless, it can be noted how this random walk evolution is slower than the system dynamics.

The RMSE shown in Figures 7 and 8 includes the convergence time. Despite the convergence time (from fifteenth second), using the UKF with the FOAM the RMSE is 0.974° and 0.873° respectively. Therefore, the established requirements in the earlier subsection are achieved, even in aerobatics scenario.

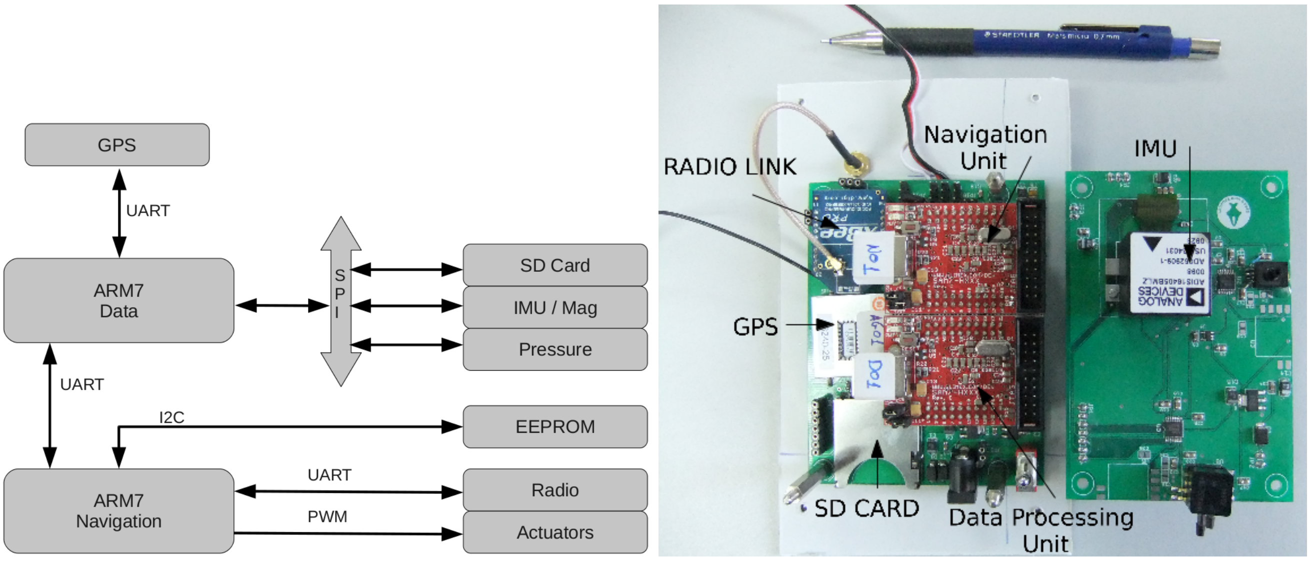
During the simulations, although it is not very clear in the graphs, it is observed that the proposed solution with FOAM does not introduce big delays.

6. Field Experiment Results

Two real experiments were carried out using the small fixed-wing UAV shown in Figure 1. The first one is a take off. This scenario is representative because the accelerometers are measuring not only the gravity but also the linear acceleration, vibrations due to the ground, lift forces, *etc*. The second experiment is a circular flight with the presence of wind gusts as perturbations.

An on-board autopilot hardware has been designed and built for this UAV. Figure 10 shows a functional diagram of the autopilot and its hardware implementation.

Figure 10. Functional diagram of the autopilot and its hardware implementation.

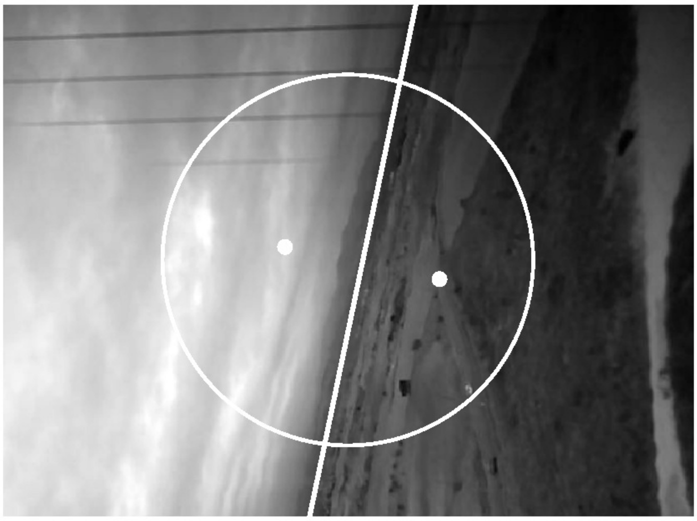


Two ARM7 microcontrollers are assigned for sensor data handling and navigation algorithms. Both are connected with a UART channel. Flight data are measured by different sensors: a GPS receiver, an Inertial Measurement Unit (IMU) with magnetometers, and four pressure sensors. One of the pressure sensors is used as a barometer for altitude measurement and the rest, one per axis, are connected to Pitot tubes for air-speed measurement. Sensor data are stored in an SD card for experiment post-processing analysis.

The objective of the experiments is to assess the accuracy of the estimation of each of the three attitude angles during a non-stationary flight such as the take-off, and to assess the filter convergence during a long flight. A ground station was built to receive data from the UAV using a radio link. The flight is manually controlled using a conventional radio control unit.

For validation purposes, data coming from independent sensors (not used by the estimation algorithm) have been considered. For the roll angle, a computer vision system is used. The GPS velocity is used to validate the yaw angle.

Figure 11. One of the frames processed by the vision system capturing the roll angle.



The vision system uses a small camera attached to the UAV. An algorithm was developed to obtain the roll angle from the video measuring the slope of the horizon. The algorithm is based on [24–26]. Figure 11 shows one of the frames taken during the flight and processed by the vision system. The results of this system have an uncertainty of ±4°. This systematic error comes from the statistical processing of the video frames; in this way, the roll angle can be determined directly by hand from the horizon.

The left graph in Figure 12 shows the roll angle evolution during the take off experiment. The vision algorithm indicates an initial roll angle of $-5°$ due to the runway inclination, which coincides with the estimated by the algorithm. This scenario involves high accelerations and it is described in detail in Figure 13. The results of the comparison are clearly satisfactory, both algorithms follow the same tendency and magnitude order, highlighting that the roll estimation has not been affected by the presence of high accelerometer readings.

Figure 12. Roll angle comparison between vision system and UKF estimation and Pitch angle at the landing time after ten minutes of flight.

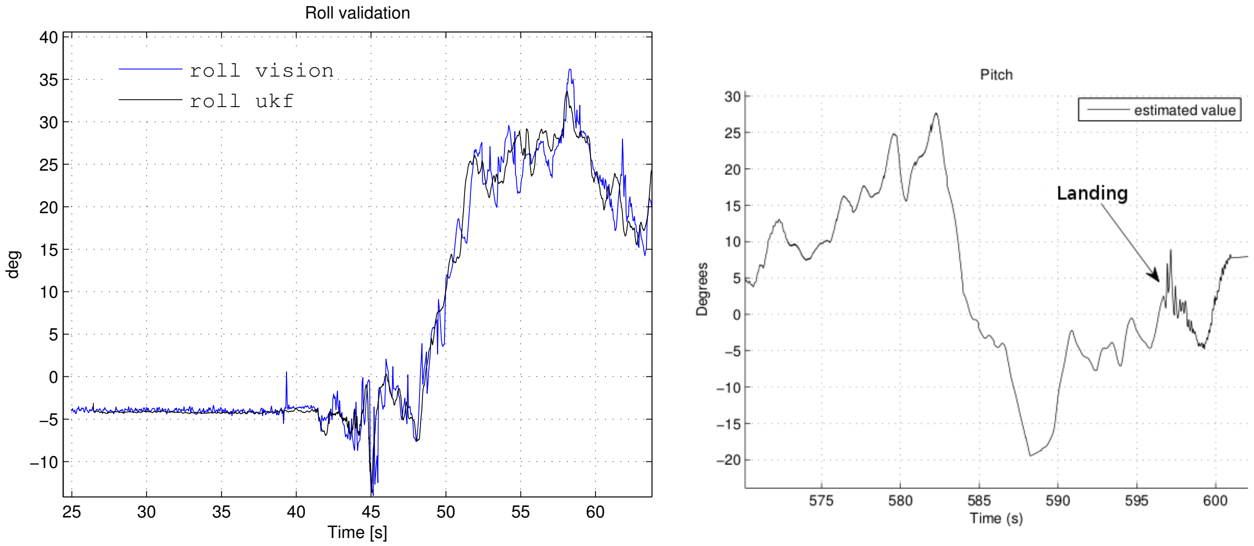
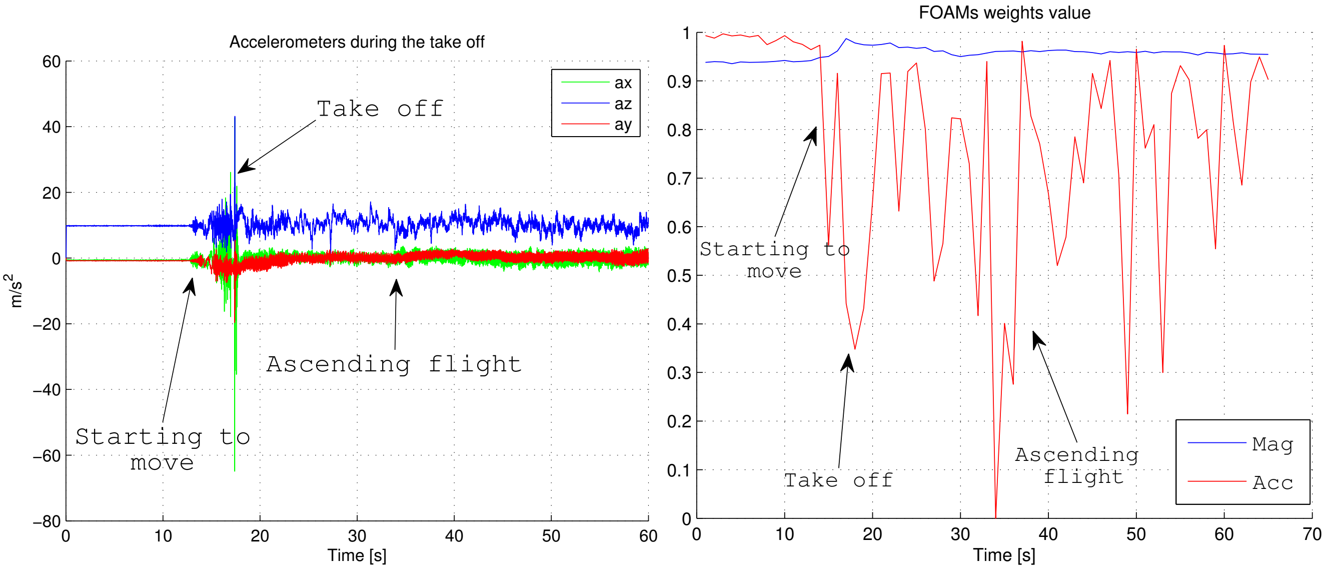


Figure 13. Accelerometers readings and FOAM weights during taking off.



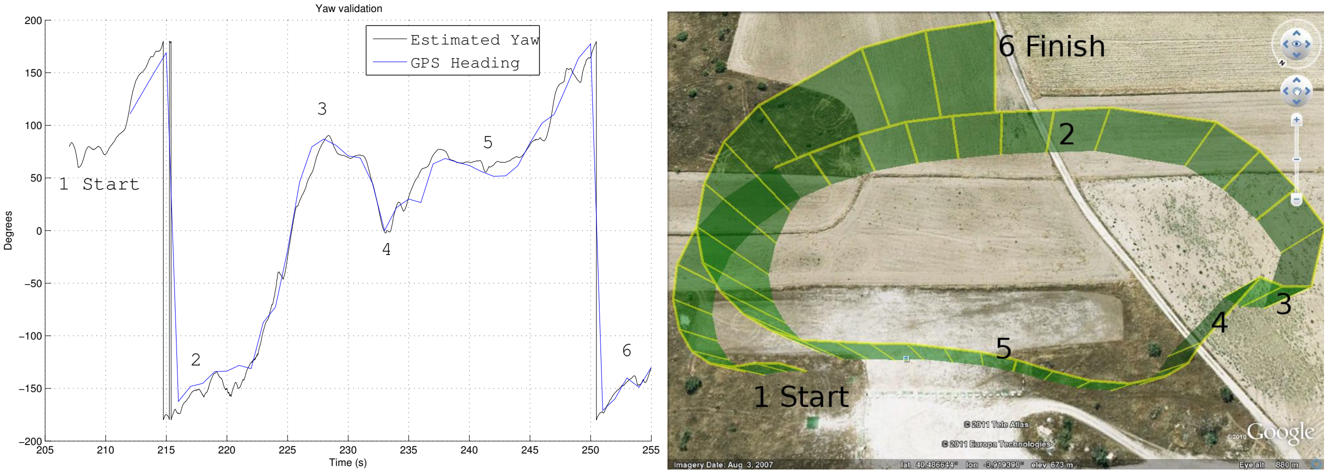
The right graph in Figure 12 shows the vehicle landing after ten minutes of flight. As it can be appreciated the AHRS integrity is held throughout the flight, estimating the same pitch value at the end of the flight compared with the start, which was closer to 7.5°. Thus, the biases have been estimated rightly.

Figure 13 shows the acceleration and the FOAM's weight evolution during the taking off maneuver. Although they describe the same take off maneuver in Figure 12, the time axis is biased with respect to the vision algorithm because both systems have been started at different times. Before the twelfth second, the UAV has been in a steady position where the accelerometers only measure the gravity. Then at the twelfth second, the vehicle begins to accelerate for reaching enough velocity for the take off. Note that all the vibrations are due to a rugged runway at this stage. At the seventeenth second the UAV pilot *pulls up* the elevator for lifting the vehicle, and at this stage the vehicle suffers a high acceleration maneuver, close to 4 g. We can see how the weights at the start are more confident when the accelerometer is only measuring the gravity (nonlinear acceleration is present), and the minimum peaks are reached at the *pull-up* stages during the take off and different ascending moments, due to the manual control of the vehicle. The weight related to the magnetometer differs from the modeled one due to the Earth's surface and power lines next to the runway. Once the UAV gains altitude, this weight becomes more confident. Therefore, the AHRS can better adapt for different situations than keeping the trust in sensors constant.

The second experiment took place under adverse atmospheric conditions, in particular, there were gusts of wind. So when there is strong crosswind, the yaw angle differs from the heading of the plane. Even if there is no wind, the plane slides due to its inertia when it turns. The heading is measured using the GPS velocity with an uncertainty of ±10°. It can be noted that the GPS velocity is only used to subtract the centripetal contribution of the accelerometers in the estimation algorithm. Therefore, it can be taken as an independent system for validation purposes.

Figure 14 shows a comparison of the yaw angle estimated and the heading measured by the GPS. The figure corresponds to a one and a half turn, and the different stages are marked with numbers in each figure. It can be noted how the North points to the bottom left corner.

Figure 14. Yaw angle comparison between GPS system and UKF estimation.



During the area marked with "1", the UAV faces the wind, therefore the GPS footprints are closer each other. The area marked with "2" shows the effect of crosswind; it biases the yaw angle and the heading, which is known as to have a non-zero *side slip* angle. In other words, the UAV does not move in the direction it points to. In the area marked with "4", the end of the first turn appears closer to 0° (pointing to the North) in heading and yaw angle. Finally, there is a clear correspondence between the GPS readings and the algorithm estimation, following the same tendency during the flight.

The sensors used in the field experiments do not allow to have an independent measurement for the pitch angle validation. However, due to the formulation of the algorithm using quaternions, the pitch angle is strongly coupled to the roll and yaw angles. Therefore, it can be assumed that if the roll and yaw angles are both correctly estimated, the pitch angle is correctly estimated too.

7. Conclusions

This paper considered the navigation problem of attitude estimation of a light-weight UAV. Due to the severely limited free space of this kind of vehicles, MEMS accelerometers and gyroscopes are only suitable for being on board.

Previous studies in the attitude determination context give the same confidence to the sensors throughout the whole UAV mission. This is unsuitable for missions that mix both aerobatic and non-aerobatic maneuvers. A common solution in the satellite attitude estimation is the FOAM algorithm and it has been used as the observation model in the UKF framework due to the highly non-linear kinematic model of aircraft attitude. An adaptive algorithm to determine the weights in FOAM algorithm has been presented, showing reliable results of the attitude along the different phases of a flight.

The performance of the algorithm has been validated by simulation and assessed using field experiments. By means of independent sensors it has been checked that the estimation algorithm gives good results, confirming the suitable quality of the estimations. The algorithm is light enough to be run on an onboard system with minimal electronic resources. The experimental results give confidence to pursue next works concerning automatic control implementation.

References

1. Kontoes, C.; Keramitsoglou, I.; Sifakis, N.; Konstantinidis, P. SITHON: An airborne fire detection system compliant with operational tactical requirements. *Sensors* **2009**, *9*, 1204–1220.
2. Cole, D.; Thompson, P.; Göktogan, A.H.; Sukkarieh, S. System development and demonstration of a cooperative UAV team for mapping and tracking. *Int. J. Robot. Res.* **2010**, *29*, 1371–1399.
3. Chan, W.L.; Hsiao, F.B. Implementation of the rauch-tung-striebel smoother for sensor compatibility correction of a fixed-wing unmanned air vehicle. *Sensors* **2011**, *11*, 3738–3764.
4. Heredia, G.; Caballero, F.; Maza, I.; Merino, L.; Viguria, A.; Ollero, A. Multi-Unmanned Aerial Vehicle (UAV) cooperative fault detection employing Differential Global Positioning (DGPS), inertial and vision sensors. *Sensors* **2009**, *9*, 7566–7579.
5. Crassidis, J.L.; Markley, F.L. Unscented filtering for spacecraft attitude estimation. *J. Guid. Control Dyn.* **2003**, *26*, 536–542.
6. Fan, C.; You, Z. Highly efficient sigma point filter for spacecraft attitude and rate estimation. *Math. Probl. Eng.* **2009**, doi:10.1155/2009/507370.
7. Wahba, G. Problem 65-1: A Least Square Estimate of Spacecraft Attitude. *SIAM Rev.* **1965**, *7*, 409.

8. Shuster, M.D.; Oh, S.D. Three axis attitude determination from vector observations. *J. Guid. Control* **1981**, *4*, 70–77.
9. Markley, F. Attitude determination using vector observations: A fast optimal matrix algorithm. *J. Astronaut. Sci.* **1993**, *41*, 261–280.
10. de Marina, H.G.; Pereda, F.; Sierra, J.M.G.; Espinosa, F. UAV attitude estimation using unscented kalman filter and TRIAD. *IEEE Trans. Ind. Electron.* **2012**, *59*, 4465–4474 .
11. Špinka, O.; Holub, O.; Hanzálek, Z. Low-cost reconfigurable control system for small UAVs. *IEEE Trans. Ind. Electron.* **2011**, *58*, 880–889.
12. Crassidis, J.L. Sigma-point Kalman filtering for integrated GPS and inertial navigation. *IEEE Trans. Aerosp. Electron. Syst.* **2006**, *42*, 750–756.
13. Stephen A.W.; Mike, F.; Logan, B. Development of a Closed-Loop Strap Down Attitude System for an Ultrahigh Altitude Flight Experiment. In *Proceedings of AIAA-1997-537, Aerospace Sciences Meeting and Exhibit, 35th*, Reno, NV, USA, 6–9 January 1997.
14. Julier, S.J.; Uhlmann, J.K. A New Extension of the Kalman Filter to Nonlinear Systems. In *Proceedings of the International Symposuim Aerospace/Defense Sensing, Simulation and Controls*, Orlando, FL, USA, April 1997; pp. 182–193.
15. Wan, E.; van der Merwe, R. The Unscencted Kalman Filter. In *Kalman Filtering and Neural Networks*; Haykin, S., Ed.; Wiley: Weinheim, Germany, 2001; Chapter 7.
16. Hasan, A.M.; Samsudin, K.; Ramli, A.R.; Azmir, R.S.; Ismaeel, S.A. A review of navigation systems (integration and algorithms). *Aust. J. Basic Appl. Sci.* **2009**, *3*, 943–959.
17. Du, D.; Liu, L.; Du, X. A Low-Cost Attitude Estimation System for UAV Application. In *Proceedings of the Chinese Control and Decision Conference (CCDC)*, Xuzhou, Jiangsu, China, 26–28 May 2010; pp. 4489–4492.
18. Song, L.; Huang, C.-Q.; Weng, X.-W.; Wu, W.-C. Research On a New Eight-Accelerometer Configuration for Attitude Angle Calculation of UAV. In *Proceedings of the Chinese Control and Decision Conference (CCDC)*, Xuzhou, Jiangsu, China, 26–28 May 2010; pp. 4119–4123.
19. Wu, Y.-L.; Wang, T.-M.; Liang, J.-H.; Wang, C.-L.; Zhang, C. Attitude Estimation for Small Helicopter Using Extended Kalman Filter. In *Proceedings of the IEEE Conference on Robotics, Automation and Mechatronics*, Chengdu, China, 21–24 September 2008; pp. 577–581.
20. Jang, J.S.; Liccardo, D. Small UAV automation using MEMS. *Aeros. Electron. Syst. Mag.* **2007**, *22*, 30–34.
21. Gebre-Egziabher, D.; Elkaim, G.; Powell, J.; Parkinson, B. A Gyro-Free Quaternion-Based Attitude Determination System Suitable for Implementation Using Low Cost Sensors. In *Proceedings of the IEEE Position Location and Navigation Symposium*, San Diego, CA, USA, 13–16 March 2000; pp. 185–192.
22. Vaganay, J.; Aldon, M.; Fournier, A. Mobile Robot Attitude Estimation by Fusion of Inertial Data. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Atlanta, GA, USA, 2–6 May 1993; Volume 1, pp. 277–282.
23. Markley, F.; Mortari, M. Quaternion attitude estimation using vector measurements. *J. Astronaut. Sci.* **2000**, *48*, 70–77.

24. Bazin, J.C.; Kweon, I.; Demonceaux, C.; Vasseur, P. UAV Attitude Estimation by Vanishing Points in Catadioptric Images. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Pasadena, CA, USA, 19–23 May 2008; pp. 2743–2749.
25. Yuan, H.Z.; Zhang, X.Q.; Feng, Z.L. Horizon Detection in Foggy Aerial Image. In *Proceedings of the International Conference on Image Analysis and Signal Processing (IASP)*, Zhejiang, China, 9–11 April 2010; pp. 191–194.
26. Pereira, G.; Iscold, P.; Torres, L. Airplane attitude estimation using computer vision: Simple method and actual experiments. *Electron. Lett.* **2008**, *44*, 1303–1304.