

Received January 25, 2020, accepted February 8, 2020, date of publication February 14, 2020, date of current version February 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2973970

Directional Statistics for 3D Model-Based UAV Tracking

NUNO PESSANHA SANTOS^{1,2}, VICTOR LOBO², AND ALEXANDRE BERNARDINO¹

¹Department of Electrical and Computer Engineering, Institute for Systems and Robotics (ISR), Instituto Superior Técnico, 1049-001 Lisbon, Portugal

²Portuguese Navy Research Center (CINAV), Escola Naval, 2810-001 Almada, Portugal

Corresponding author: Nuno Pessanha Santos (nuno.pessanha.santos@tecnico.ulisboa.pt)

This work was supported by the national project VOAMAS under Grant 02/SAICT/2017/31172.

ABSTRACT We present a vision system based on a monocular camera to track the 3D position and orientation of an Unmanned Aerial Vehicle (UAV) during the landing process aboard a ship. The proposed method uses a 3D model-based approach based on a Particle Filter (PF) with proposal distributions given by an Unscented Kalman Filter (UKF) for the translational motion and filters based on directional statistics for the rotational motion. Our main contributions are (i) the development of a novel 3D model-based tracking architecture based on directional statistics that can be easily adapted to other tracking problems, and (ii) the development of the Unscented Bingham-Gauss Filter (UBiGaF) for rotation estimation. We show the advantages of using directional statistics based filters on 3D model-based tracking in a series of quantitative tests in a challenging simulation scenario with real video data. The obtained position and angular error are compatible with the automatic landing system requirements when using directional statistics. We obtain lower error when using the UBiGaF scheme for the vast majority of the tested combinations.

INDEX TERMS Unmanned aerial vehicles, computer vision, motion estimation, algorithm design and analysis, military vehicles.

I. INTRODUCTION

The vast majority of the accidents and incidents with Unmanned Aerial Vehicles (UAVs) occur during take-off or landing [1], [2], since these are the most challenging manoeuvres where, typically, an external pilot takes control. Reducing human intervention in these operations increases system reliability and alleviates the use of trained UAV pilots.

We propose one approach to automate the landing manoeuvre by tracking the pose of the UAV with a monocular Red, Green and Blue (RGB) camera located on the Fast Patrol Boat (FPB) with a processing station to perform the needed Computer Vision (CV) processing tasks. A ground-based vision system makes it possible to use more processing power, allowing the use of more computationally intensive methods. On the contrary to systems that embed the sensing and processing on the UAV, our approach does not require any customization of standard Commercial-Off-The-Shelf (COTS) UAVs. The UAV platform is composed of sensors, actuators, communications, and onboard processing, and the Ground Control Station (GCS) is composed of communi-

The associate editor coordinating the review of this manuscript and approving it for publication was Yuming Fang.

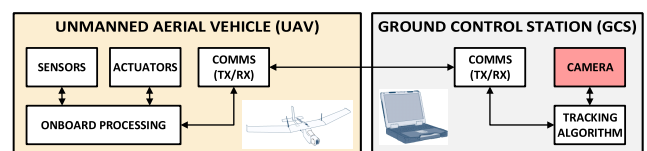


FIGURE 1. General scheme illustration.

cations, a camera, and a processing station with a tracking algorithm (Figure 1). The system observes and obtains the UAV pose and sends that information to the GCS that computes the needed control commands to guide the UAV via radio to perform autonomous landing using a net-based retention system (Figure 2). In our scenarios, the fact that the camera is installed in a moving platform in a maritime environment and the reduced dimensions of the fixed-wing UAV makes the initial detection and the tracking very challenging tasks.

In previous work [3] we have proposed a pose tracking framework using a 3D model-based vision system employing the UAV Computer-Aided Design (CAD) model and an Unscented Particle Filter (UPF) based approach using Gaussian noise in the translational component of motion and



FIGURE 2. Landing area (red rectangle) and used UAV model.

Bingham (Bi) noise in the rotational component, to better model the periodic nature of rotations. However, the Bi noise is specific to the angular position component and does not capture its correlation with the angular velocity. In this paper, we introduce Bingham-Gauss (BiGa) noise to model the full rotational noise, both in its angular position and velocity components. We add new results that show significant performance improvements when using the BiGa model with respect to the Bi model (up to 50% less pose estimation error). In the current implementation, we divide the translation and rotation into two independent motions. In the UAV field, this decoupling has been applied with success to automatic landing using the airborne camera [4], using stereo vision to estimate altitude, attitude and motion [5] and UAV controller design [6]. For filtering the rotational motion, the Gaussian model is not a good approximation in the presence of strong orientation noise. The Bi distribution (Appendix B) is one distribution employed in the field of directional statistics a subdiscipline of statistics that deals with directions. Some examples of data that may be considered as directional statistics are orientations, compass directions and time of day. The BiGa distribution (Appendix D), consists in the product of a Bi distribution and a Gaussian distribution conditioned on the Bi distributed random variables [7]. This distribution allows us to capture dependencies between the angular velocity and the UAV attitude. The motion filtering is made using an Unscented Kalman Filter (UKF) for the translation [8] (Appendix A) and an Unscented Bingham Filter (UBiF) [3], [9]–[12] (Section IV-D) or a new filter using the BiGa distribution, denoted Unscented Bingham-Gauss Filter (UBiGaF) (Section IV-E) for the rotation.

The main contributions of this article are (i) the inclusion of directional statistics and correlation between attitude and angular velocity in the UAV tracking field, (ii) the development and analysis of a novel UBiGaF for the state estimation of rotating objects, (iii) the use of directional statistics in 3D model-based tracking and (iv) the comparison of performance between the UPF, the UBiF and the UBiGaF in a simulation environment using real data.

This article is organized as follows. In Section II, the related work regarding 3D Model-based pose estimation and pose tracking is outlined. In Section III, the overall system proposal is outlined, describing the used target detection and tracking methods. In Section IV, the used motion models and Unscented filters are explained in detail. In Section V, we present the experimental results. Finally, in Section VI, we present the conclusions and provide directions for further research work.

II. RELATED WORK

This section makes a review of the related work with particular focus on the 3D Model-based pose estimation (Section II-A) and pose tracking (Section II-B).

A. 3D MODEL-BASED POSE ESTIMATION

Pose estimation is an important task and has been studied in several fields of science. The class of methods that use the object or environment CAD model to perform pose estimation are called 3D model-based pose estimation. The main applications of 3D model-based pose estimation are Augmented Reality (AuRe) [13], robot manipulation [14], robot navigation [15], object pose tracking [16], among others.

Using the RGB onboard camera and the environment 3D CAD model we can perform feature matching [15] using the moving edges tracker algorithm for pose estimation and combine this information with Inertial Measurement Unit (IMU) data to get the translational velocities needed for the UAV control [17]. In [18], [19], we describe a ground-based vision system that estimates the UAV pose using its CAD model. The vast majority of the ground-based systems are developed for tracking or control without using CAD information. Nowadays all UAVs have their 3D CAD model available, so their pose can be estimated using a class of methods for 3D model-based pose estimation. We propose a 3D model-based ground-based vision system with high processing capability, which allows reducing the UAV size, weight and power requirements. This approach makes it also possible to use standard UAVs equipped with COTS autopilots.

B. POSE TRACKING

Given several measures over time, we can use target specific dynamic models to filter the sensor data using a Kalman Filter (KF) [20], an Extended Kalman Filter (EKF) [21], a UKF [22] or a Particle Filter (PF) [23]. PFs in CV were applied to visual tracking tasks such as ball tracking [24], spherical pendulum tracking [25], human face tracking [26], articulated object tracking [27], vehicle tracking [28] or object tracking [29]. The combination of a PF with a UKF, known as a UPF [30], is described in [31] for visual contour tracking, and in [32] for ground maneuvering target tracking. The UKF generates better proposal distributions for the PF, taking into account the current observations [33].

When using conventional filtering techniques for attitude estimation, we have to consider a small angle assumption to quantify the existing uncertainty [34]. To overcome this, we can use a directional distribution to take into account the periodic nature of the rotation. The Bi distribution [35] is defined directly on the unit multidimensional hypersphere [36]. It has been successfully used to (i) estimate the attitude of a ping pong ball [37], (ii) for estimation of orientations in the 3D space with unit quaternions [38] and (iii) using a UKF based approach using deterministic sampling [9]. Some mixtures and combinations of distributions have also been made to quantify the correlation between



FIGURE 3. UAV model (left) and camera reference frame (right).

Euclidean states (e.g. angular velocities) and the attitude on its manifold. Some of these implementations are the Partially-Conditioned Gaussian Mixtures (PCGM) [34], the Gauss-Bingham (GaBi) [10] and the Bingham-Gauss (BiGa) [7] distributions. These approaches allow us to capture the covariance between the attitude and angular velocity uncertainties.

As described in Section I, we have developed a pose tracking architecture based on a UPF. The main developments regarding our previous work [3], [12] is a comparison among several tracking architectures based on UPFs [8] and the development of a novel UBiGaF able to capture the covariance between attitude and angular velocity.

III. SYSTEM PROPOSAL

This section presents the overall system architecture used to perform UAV tracking. This architecture has been developed and refined in the last few years in a series of publications [3], [8], [12], [18], [19]. In this paper, we propose modifications to the tracking modules using directional statistics.

As described in Section I, we are using a UPF based scheme where each filter particle will represent the UAV state - linear and angular positions and velocities:

$$\mathbf{x}_t = [\mathbf{t}_t^T, \mathbf{r}_t^T]^T \text{ with } \mathbf{t}_t^T = [\mathbf{u}_t^T, \mathbf{v}_t^T] \text{ and } \mathbf{r}_t^T = [\mathbf{q}_t^T, \boldsymbol{\omega}_t^T] \quad (1)$$

where $\mathbf{u}_t^T = [X, Y, Z]$ is the linear position, $\mathbf{v}_t^T = [v_x, v_y, v_z]$ is the linear velocity, $\mathbf{q}_t^T = [q_1, q_2, q_3, q_4]$ (3) is the orientation quaternion, and $\boldsymbol{\omega}_t^T = [\omega_x, \omega_y, \omega_z]$ is the angular velocity according to the camera reference frame described in Figure 3. The state distribution is represented by the particle set \mathbf{x}_t^i . The developed system can be divided into two parts (Figure 4):

- **Detection and Hypotheses generation** (Section III-A);
- **Tracking** (Section III-B).

A. DETECTION AND HYPOTHESES GENERATION

The initial UAV detection is made using a *You Only Look Once* (YOLO) classifier [39] trained on a created synthetic dataset performing transfer learning to real images. After we obtain the Region Of Interest (ROI), we compare it with a pre-trained database of synthetic images of the UAV in multiple poses to retrieve a set of top-ranked poses as described in [18].

B. TRACKING

The standard UPF [30] scheme is divided into (i) initialization, (ii) importance sampling, and (iii) importance weighting and resampling. The initialization is possible using the detection and hypotheses generation stage (Section III-A). The main difference, when compared with the generic UPF,

is in the importance sampling step, where we use a UKF-UBiF/UBiGaF (Section IV) to integrate the current observation and generate a better proposal distribution. The current observation $\mathbf{z}_t = [\mathbf{t}_t^T, \mathbf{q}_t^T]^T$ is given by the maximum likelihood particle according to a color similarity metric [18]. The applied resampling strategy was resampling reallocation [19]. The mode of the distribution, approximated by the particle with the largest importance weight before resampling, is used as the state estimation. If during tracking, we cannot detect the UAV on the frame, all the particles came from the last iteration. The main article focus will be the motion filtering rotational part (Section IV).

IV. MOTION FILTERING

Motion filtering has the objective of using measures affected by uncertainty and noise over time and generate results closer to the real values of the measured quantities. We have considered a constant velocity model with independent linear and angular motions [40]. The translational motion (Section IV-A) filtering will be performed using a UKF (Appendix A) and the rotational motion (Section IV-B) filtering will be performed using a UBiF (Section IV-D) or a UBiGaF (Section IV-E).

A. TRANSLATIONAL TRANSITION MODEL

As we are using a linear model for translation, a simple KF could be applied. To facilitate the transition between the linear and the angular filter formulations, we will use a discrete-time UKF. This is a very common filter, so we remit its description to Appendix A. In discrete time, the evolution of the state is:

$$\mathbf{x}_{t+1}^l = \mathbf{F}^l(\mathbf{x}_t^l, \boldsymbol{\xi}_t^l) = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \Delta t \cdot \mathbf{I}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix} \mathbf{x}_t^l + \boldsymbol{\xi}_t^l \quad (2)$$

where $\mathbf{x}_{t+1}^l = \mathbf{t}_{t+1}$ (1), $\boldsymbol{\xi}_t^l \sim \mathcal{N}(0, \mathbf{Q}_t^l)$ is a Gaussian noise random variable with zero mean and covariance \mathbf{Q}_t^l .

B. ROTATIONAL TRANSITION MODEL

We represent the UAV rotation using a unit quaternion:

$$\mathbf{q} = [\boldsymbol{\rho}^T, q_4]^T \quad (3)$$

where $\mathbf{q} \in S^3 \subset \mathbb{R}^4 : \|\mathbf{q}\| = 1$ with:

$$\boldsymbol{\rho}^T = [q_1, q_2, q_3] = \hat{\mathbf{e}} \sin\left(\frac{\rho}{2}\right) \text{ and } q_4 = \cos\left(\frac{\rho}{2}\right) \quad (4)$$

where $\hat{\mathbf{e}}$ is the axis rotation and ρ is the angle of rotation. In discrete time, the evolution of the state is [8]:

$$\mathbf{x}_{t+1}^r = \mathbf{F}^r(\mathbf{x}_t^r, \boldsymbol{\xi}_t^r) = \begin{bmatrix} \mathbf{q}_t \otimes \delta \mathbf{q}_t^\omega \otimes \delta \mathbf{q}_t^r \\ \boldsymbol{\omega}_t + \boldsymbol{\xi}_t^r \end{bmatrix} \quad (5)$$

where $\mathbf{x}_{t+1}^r = \mathbf{r}_{t+1}$ (1), \otimes represents unit quaternion multiplication (orientations composition), $\boldsymbol{\xi}_t^r \sim \mathcal{N}(0, \mathbf{Q}_t^r)$ is a Gaussian noise random variable with zero mean and covariance \mathbf{Q}_t^r , and $\delta \mathbf{q}_t^\omega$ and $\delta \mathbf{q}_t^r$ are quaternions representing the integration of the effect of the angular velocity and rotation noise assumed constant during a sampling interval Δt :

$$\delta \mathbf{q}_t^\omega = \boldsymbol{\Omega}(\boldsymbol{\omega}_t) \text{ and } \delta \mathbf{q}_t^r = \boldsymbol{\Omega}(\boldsymbol{\xi}_t^r) \quad (6)$$

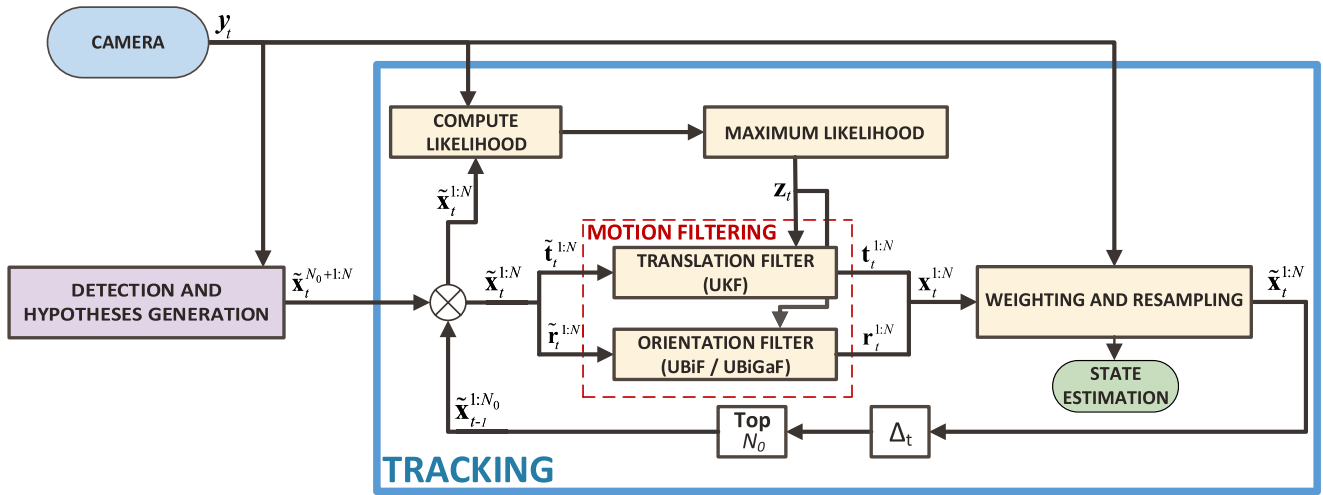


FIGURE 4. System architecture.

with:

$$\Omega(\mathbf{b}) = \left[\frac{\mathbf{b}}{|\mathbf{b}|} \sin\left(\frac{|\mathbf{b}| \Delta t}{2}\right), \cos\left(\frac{|\mathbf{b}| \Delta t}{2}\right) \right] \quad (7)$$

C. OBSERVATION MODEL

The observation model is given by:

$$\mathbf{z}_t = \begin{bmatrix} \mathbf{H}^l(\mathbf{t}_t, \eta_t^l) \\ \mathbf{H}^r(\mathbf{r}_t, \eta_t^r) \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{I}_{4 \times 4} & \mathbf{0}_{3 \times 3} \end{bmatrix} \begin{bmatrix} \mathbf{t}_t + \eta_t^l \\ \mathbf{r}_t \otimes \delta \mathbf{q}_t^r \end{bmatrix} \quad (8)$$

where $\eta_t^l \sim \mathcal{N}(0, \mathbf{R}_t)$ is a Gaussian noise random variable with zero mean and covariance matrix \mathbf{R}_t and $\delta \mathbf{q}_t^r$ is a quaternion representing the integration of the effect of the observation rotation noise in a similar way to (6). When using the UBiF (Section IV-D) or the UBiGaF (Section IV-E), the noise of the rotational components of the observation model is assumed Bi distributed (Appendix B).

D. UNSCENTED BINGHAM FILTER (UBiF)

In a periodic domain like the manifold of orientations in a 3D space, the Gaussian model is not a good approximation, especially in the presence of strong noise. The Bi distribution is an antipodally symmetric distribution that represents a zero-mean Gaussian distribution projected on the unit hypersphere (Appendix B). Since the product of two Bi distributions is closed under multiplication after renormalization (Appendix B-B), we can use the UBiF with an update step directly derived from the Bayes filter formulation. We apply a UBiF to the orientation part of the state vector. As in other filtering frameworks, the used UBiF has two stages: (i) prediction (Section IV-D.1), and (ii) update (Section IV-D.2). The angular velocities will be obtained from the orientation difference between iterations. The UBiF schematic view is described in Figure 5.

1) PREDICTION

The prediction step is described in Algorithm 1. For the rotation case, the system model is given by:

$$\mathbf{q}_t = \mathbf{F}(\mathbf{q}_{t-1}) \otimes \Phi_{t-1} \quad (9)$$

where $\mathbf{q}_{t-1} \sim P_B(\mathbf{M}_{t-1}^e, \mathbf{Z}_{t-1}^e)$ is the orientation at time $t - 1$ and $\Phi_{t-1} \sim P_B(\mathbf{M}_{t-1}^\Phi, \mathbf{Z}_{t-1}^\Phi)$ is the Bi distributed system noise where \mathbf{M} is an orthogonal matrix describing the orientation of the distribution and \mathbf{Z} is the concentration matrix that controls the spread of the distribution around its mean (Appendix B). The system dynamic $\mathbf{F}(\cdot)$ is given by (5).

Algorithm 1 UBiF - Prediction Step

- ▷ **Initialization:** Φ_{t-1} (9)
- ▷ **Inputs:** $\mathbf{M}_{t-1}^e, \mathbf{Z}_{t-1}^e$ (9)
- 1. Approximate the current system state (Appendix B-E): $\mathbf{q}_{t-1} \sim P_B(\mathbf{M}_{t-1}^e, \mathbf{Z}_{t-1}^e)$
- 2. Obtain the points \mathbf{q}_{t-1}^i using deterministic sampling (Appendix C): $\mathbf{q}_{t-1}^i = \mathbf{M}_{t-1}^e \tilde{\mathbf{q}}_{t-1}^i \quad i = 1, \dots, 7$
- 3. Propagate the sigma points \mathbf{q}_t^i using the system model $\mathbf{F}(\cdot)$ (5): $\tilde{\mathbf{q}}_t^i = \mathbf{F}(\mathbf{q}_{t-1}^i) \quad i = 1, \dots, 7$
- 4. Compute the covariance matrix $\mathbf{C}_{\tilde{\mathbf{q}}_t}$ from the sigma points (46): $\mathbf{C}_{\tilde{\mathbf{q}}_t} = \text{Cov}(\tilde{\mathbf{q}}_t)$
- 5. Obtain the covariance matrix \mathbf{C}_Φ from the Bi system noise Φ_t (45): $\mathbf{C}_\Phi = \text{Cov}(\Phi_t)$
- 6. Obtain the covariance matrices composition $\mathbf{C}_{\tilde{\mathbf{q}}_t^\Phi}$ (Appendix B): $\mathbf{C}_{\tilde{\mathbf{q}}_t^\Phi} = \text{Cov}(\tilde{\mathbf{q}}_t \otimes \Phi_t)$
- 7. Estimate the Bi distribution from the obtained covariance (47): $P_B(\bar{\mathbf{M}}_t, \bar{\mathbf{Z}}_t) \sim \text{MLE}(\mathbf{C}_{\tilde{\mathbf{q}}_t^\Phi})$
- ▷ **Outputs:** $\bar{\mathbf{M}}_t, \bar{\mathbf{Z}}_t$

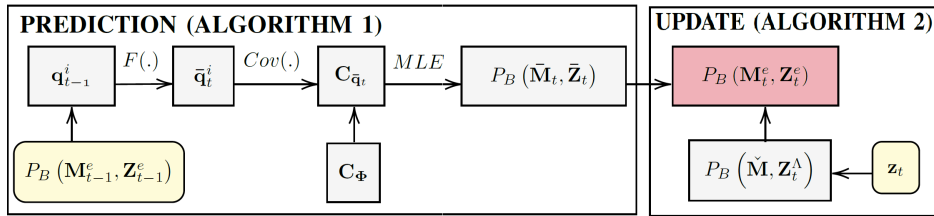


FIGURE 5. UBiF schematic view.

2) MEASUREMENT UPDATE

The measurement update step is described in Algorithm 2. The measurement model is represented as:

$$\mathbf{z}_t = \mathbf{H}(\mathbf{q}_t) \otimes \Lambda_t \quad (10)$$

where $\mathbf{z}_t \in \mathbb{S}^3$ is the measurement at time t and $\Lambda_t \sim P_B(\mathbf{M}_t^\Lambda, \mathbf{Z}_t^\Lambda)$ is the Bi distributed measurement noise. The current observation \mathbf{z}_t is given by the orientation quaternion of the maximum likelihood particle, as described in Figure 4. Function $\mathbf{H}(\cdot)$ relates the measurement \mathbf{z}_t to the values of the orientation \mathbf{q}_t (identity function in our case). Choosing $\mathbf{M}_t^\Phi = \mathbf{M}_t^\Lambda = \mathbf{I}_{4 \times 4}$ is equivalent to the concept of zero-mean noise in the *Euclidean* space.

To be able to apply the measurement update step (Algorithm 2), the noise is rotated (disturbed) according to the actual measurement \mathbf{z}_t according to (43):

$$\begin{aligned} P(\mathbf{z}_t | \bar{\mathbf{q}}_t') &= P_B(\bar{\mathbf{q}}_t'^{-1}) \otimes \mathbf{z}_t; \mathbf{M}_t^\Lambda, \mathbf{Z}_t^\Lambda \\ &= P_B(\text{diag}(-1, -1, -1, 1) \cdot \bar{\mathbf{q}}_t' \otimes \mathbf{z}_t; \mathbf{M}_t^\Lambda, \mathbf{Z}_t^\Lambda) \\ &= P_B(\bar{\mathbf{q}}_t'; \underbrace{\text{diag}(-1, -1, -1, 1) \cdot \mathbf{M}_t^\Lambda \otimes \mathbf{z}_t, \mathbf{Z}_t^\Lambda}_{\check{\mathbf{M}}}) \\ &= P_B(\check{\mathbf{M}}, \mathbf{Z}_t^\Lambda) \end{aligned} \quad (11)$$

where $P_B(\check{\mathbf{M}}, \mathbf{Z}_t^\Lambda)$ represents the measurement noise $P_B(\mathbf{M}_t^\Lambda, \mathbf{Z}_t^\Lambda)$ with a peak aligned with \mathbf{z}_t and spread \mathbf{Z}_t^Λ .

The estimate is described by $P_B(\mathbf{M}_t^e, \mathbf{Z}_t^e)$, directly obtained by the product of the Bi distributed system state $P_B(\bar{\mathbf{M}}_t, \bar{\mathbf{Z}}_t)$ with the Bi disturbed measure $P_B(\check{\mathbf{M}}, \mathbf{Z}_t^\Lambda)$ (Algorithm 2). The update quaternion \mathbf{q}_t is obtained from the $P_B(\mathbf{M}_t^e, \mathbf{Z}_t^e)$ mode (Algorithm 2). The quaternion error is obtained by multiplying the previous quaternion (\mathbf{q}_{t-1}) with the conjugate of the estimated one ($\bar{\mathbf{q}}_t$). The angular velocities are obtained converting to the angle-axis representation, according to:

$$\omega_x = \frac{2 \cos^{-1}(q_4)}{\Delta t} \times \frac{q_1}{\|\mathbf{q}_e\|} \quad (12)$$

$$\omega_y = \frac{2 \cos^{-1}(q_4)}{\Delta t} \times \frac{q_2}{\|\mathbf{q}_e\|} \quad (13)$$

$$\omega_z = \frac{2 \cos^{-1}(q_4)}{\Delta t} \times \frac{q_3}{\|\mathbf{q}_e\|} \quad (14)$$

where Δt is the sampling interval and $\mathbf{q}_e = \mathbf{q}_{t-1} \otimes \bar{\mathbf{q}}_t = [q_1, q_2, q_3, q_4]^T$ (4).

Algorithm 2 UBiF - Update Step

- ▷ **Initialization:** Λ_t (10)
- ▷ **Inputs:** $\bar{\mathbf{M}}_t, \bar{\mathbf{Z}}_t$ (Algorithm 1) and \mathbf{z}_t (10)
- 1. Disturb the measure \mathbf{z}_t using the Bi noise Λ_t (11): $P_B(\check{\mathbf{M}}, \mathbf{Z}_t^\Lambda)$
- 2. The measurement update can be derived from the *Bayes theorem*: $P(\bar{\mathbf{q}}_t' | \mathbf{z}_t) = C \cdot P(\mathbf{z}_t | \bar{\mathbf{q}}_t') \cdot P(\bar{\mathbf{q}}_t')$
 $P_B(\mathbf{M}_t^e, \mathbf{Z}_t^e) = P_B(\check{\mathbf{M}}, \mathbf{Z}_t^\Lambda) \cdot P_B(\bar{\mathbf{M}}_t, \bar{\mathbf{Z}}_t)$
 where C is a normalization constant;
- 3. The final estimate is then obtained from (42): $P_B(\mathbf{M}_t^e, \mathbf{Z}_t^e) = P_B(\check{\mathbf{M}}, \mathbf{Z}_t^\Lambda) \cdot P_B(\bar{\mathbf{M}}_t, \bar{\mathbf{Z}}_t)$
- ▷ **Outputs:** $\mathbf{M}_t^e, \mathbf{Z}_t^e$

E. UNSCENTED BINGHAM-GAUSS FILTER (UBIGAF)

The UBiF does not quantify the uncertainty of the correlation between angular velocity ω and the quaternion attitude \mathbf{q} on their natural manifold. We will use the BiGa distribution (Appendix D) that allows to capture this correlation in a filtering structure. The BiGa is a distribution that consists in the product of a Bi distribution and a Gaussian distribution conditioned on the Bi distributed random variables. Thus, we propose the Unscented Bingham-Gauss Filter (UBiGaF). As described in the Section IV-D, the multiplication of two Bi distributions is closed under multiplication after renormalization, but the same did not happen using the BiGa distribution since we do not have a closed-form multiplication. To be able to incorporate it on a filtering structure, we developed an update step that was based on the Unscented Transform (UT) [30], [31] with a structure similar to the UKF. The UBiGaF is separated into: (i) prediction (Section IV-E.1), and (ii) update (Section IV-E.2). The UBiGaF schematic view is described in Figure 6.

1) PREDICTION

The prediction step is described in Algorithm 3. The system state at time $t - 1$ is given by:

$$\mathbf{r}_{t-1}^T = [\mathbf{q}_{t-1}^T, \omega_{t-1}^T] \quad (15)$$

where \mathbf{q}_{t-1} is the attitude quaternion and ω_{t-1} is its angular velocity at time instant $t - 1$. The state vector is assumed to be BiGa distributed with parameters defined by \mathbf{m}_{t-1}^ω (60), \mathbf{P}_{t-1}^ω (61), \mathbf{P}_{t-1}^q (62) and $\mathbf{P}_{t-1}^{q\omega}$ (63) at time instant $t - 1$.

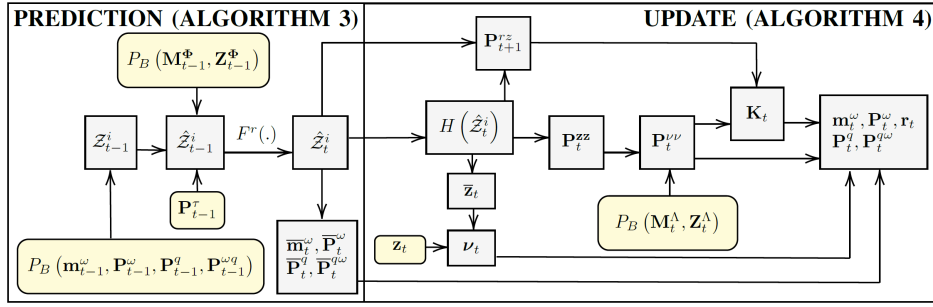


FIGURE 6. UBiGaF schematic view.

The system model is given by:

$$\mathbf{r}_t = \mathbf{F}(\mathbf{r}_{t-1}) \odot \Psi_{t-1} \quad (16)$$

where $\Psi_{t-1} \sim P_{BG}$ with Covariance \mathbf{P}_{t-1}^r for the angular velocity part and Covariance $\Phi_{t-1} \sim \mathbf{P}_{t-1}^\Phi \sim P_B(\mathbf{M}_{t-1}^\Phi, \mathbf{Z}_{t-1}^\Phi)$ for the orientation part, $\mathbf{F}(\cdot)$ is the motion model (5) and \odot represents the BiGa composition obtained using the sigma points representation (Appendix E).

The quaternion motion $\mathbf{q}_M = [q_1, q_2, q_3, q_4]^T$ (4) used to propagate each one of the sigma points $\hat{\mathbf{Z}}_{t-1}^i$ (Algorithm 3) is given by (7):

$$q_1 = \frac{\omega_x}{\|\boldsymbol{\omega}\|} \sin\left(\frac{\|\boldsymbol{\omega}\| \Delta t}{2}\right) \quad (17)$$

$$q_2 = \frac{\omega_y}{\|\boldsymbol{\omega}\|} \sin\left(\frac{\|\boldsymbol{\omega}\| \Delta t}{2}\right) \quad (18)$$

$$q_3 = \frac{\omega_z}{\|\boldsymbol{\omega}\|} \sin\left(\frac{\|\boldsymbol{\omega}\| \Delta t}{2}\right) \quad (19)$$

$$q_4 = \cos\left(\frac{\|\boldsymbol{\omega}\| \Delta t}{2}\right) \quad (20)$$

where $f_\omega(\hat{\mathbf{Z}}_{t-1}^i) = [\omega_x, \omega_y, \omega_z]^T$ and Δt is the sampling interval. We calculate the BiGa parameters from the obtained sigma points after propagation $\hat{\mathbf{Z}}_t^i$ (Appendix E).

After the prediction step (Algorithm 3), the predicted system state is described by a BiGa distribution with parameters defined by $\bar{\mathbf{m}}_t^\omega, \bar{\mathbf{P}}_t^\omega, \bar{\mathbf{P}}_t^q, \bar{\mathbf{P}}_t^{q\omega}$.

2) MEASUREMENT UPDATE

The measurement update step is described in Algorithm 4, and is similar to the used in the UBiF (Section IV-D.2). The measurement model is also represented as:

$$\mathbf{z}_t = \mathbf{H}(\mathbf{q}_t) \otimes \Lambda_t \quad (21)$$

where $\mathbf{z}_t \in \mathbb{S}^3$ is the current observation at time t and $\Lambda_t \sim P_B(\mathbf{M}_t^\Lambda, \mathbf{Z}_t^\Lambda)$ is the Bi distributed measurement noise. Function $\mathbf{H}(\cdot)$ relates the measurement \mathbf{z}_t to the values of the orientation \mathbf{q}_t (identity function in our study case).

Using the BiGa formulation, we can model the full rotational noise, both in its angular position and velocity components. As initially described in Section IV-E, we do not

Algorithm 3 UBiGaF - Prediction Step

▷ **Initialization:** Ψ_{t-1} (16)

▷ **Inputs:** $\mathbf{m}_{t-1}^\omega, \mathbf{P}_{t-1}^\omega, \mathbf{P}_{t-1}^q, \mathbf{P}_{t-1}^{q\omega}$ (16)

1. Approximate the current system state (60, 61, 62, 63):
 $[\mathbf{q}_{t-1}^T, \boldsymbol{\omega}_{t-1}^T] \sim P_{BG}([\mathbf{q}_{t-1}^T, \boldsymbol{\omega}_{t-1}^T]; \mathbf{m}_{t-1}^\omega, \mathbf{P}_{t-1}^\omega, \mathbf{P}_{t-1}^q, \mathbf{P}_{t-1}^{q\omega})$

2. Obtain sigma points \mathbf{Z}_{t-1}^i (Appendix E);

3. Add uncertainty \mathbf{P}_{t-1}^r to the angular velocity covariance:

$$\hat{\mathbf{P}}_{t-1}^\omega = \mathbf{P}_{t-1}^\omega + \mathbf{P}_{t-1}^r$$

4. Add Bi noise Φ_{t-1} to the orientation covariance (Appendix B):

$$\Phi_{t-1}^q = \text{Cov}(\mathbf{P}_{t-1}^q \otimes \Phi_{t-1})$$

5. Add uncertainty $\hat{\mathbf{P}}_{t-1}^\omega$ to the sigma points angular velocity part and Φ_{t-1}^q to the quaternion part originating the sigma points $\hat{\mathbf{Z}}_{t-1}^i$;

6. Propagate each one of the sigma points $\hat{\mathbf{Z}}_{t-1}^i$ (5): $\hat{\mathbf{Z}}_t^i = \mathbf{F}^r(\hat{\mathbf{Z}}_{t-1}^i) = f_q(\hat{\mathbf{Z}}_{t-1}^i) \otimes \mathbf{q}_M$ where f_q is the quaternion part of the considered sigma point and \mathbf{q}_M is the quaternion motion given by the angular part $f_\omega(\hat{\mathbf{Z}}_{t-1}^i)$;

7. Compute $\bar{\mathbf{m}}_t^\omega$ (82), $\bar{\mathbf{P}}_t^\omega$ (83), $\bar{\mathbf{P}}_t^q$ (84) and $\bar{\mathbf{P}}_t^{q\omega}$ (85) from $\hat{\mathbf{Z}}_t^i$.

▷ **Outputs:** $\bar{\mathbf{m}}_t^\omega, \bar{\mathbf{P}}_t^\omega, \bar{\mathbf{P}}_t^q, \bar{\mathbf{P}}_t^{q\omega}, \hat{\mathbf{Z}}_t^i$

have a closed-form multiplication for the product of two BiGa distributions. To be able to deal with that, we have adopted a structure similar to the UKF (Appendix A). Using this filtering structure, we can obtain the *a posteriori* state estimate \mathbf{r}_t and state covariance \mathbf{P}_t^r , as described in Algorithm 4. Contrarily to the UBiF, we do not need to estimate the angular velocities from the quaternion difference since we estimate them directly in the filter.

V. EXPERIMENTAL RESULTS

In this section, we show results of UAV tracking in a simulated environment using real video footage. All the developments were made on a 3.70 GHz Intel i7-8700K Central Processing Unit (CPU) using an NVIDIA Quadro P5000 Graphics Processing Unit (GPU). We have used C++ with

Algorithm 4 UBiGaF - Update Step

- ▷ **Initialization:** Λ_t (21)
- ▷ **Inputs:** $\bar{\mathbf{m}}_t^\omega, \bar{\mathbf{P}}_t^\omega, \bar{\mathbf{P}}_t^q, \hat{\mathbf{Z}}_t^i$ (Algorithm 3) and \mathbf{z}_t (21)
- 1. Predict measurement expected value $\bar{\mathbf{z}}_t$ (32) from $\hat{\mathbf{Z}}_t^i$ angular part;
- 2. Predict measurement covariance \mathbf{P}_t^{zz} (33) from $\hat{\mathbf{Z}}_t^i$ angular part;
- 3. Obtain the innovation \mathbf{v}_t (34);
- 4. Obtain the innovation covariance \mathbf{P}_t^{vv} (35):
 $\mathbf{P}_t^{vv} = Cov(\mathbf{v}_t \otimes \Lambda_t)$
- 5. Computation of the cross-correlation matrix \mathbf{P}_t^{rz} (36);
- 6. Computation of the Kalman gain \mathbf{K}_t (37):
 $\mathbf{K}_t = \mathbf{P}_t^{rz} (\mathbf{P}_t^{vv})^{-1}$
- 7. Update of the *a posteriori* state estimate \mathbf{r}_t (38);
- 8. Retrieve $\mathbf{m}_t^\omega, \mathbf{P}_t^\omega, \mathbf{P}_t^q, \mathbf{P}_t^{q\omega}$ from the state covariance \mathbf{P}_t^r (39).
- ▷ **Outputs:** $\mathbf{r}_t, \mathbf{m}_t^\omega, \mathbf{P}_t^\omega, \mathbf{P}_t^q, \mathbf{P}_t^{q\omega}$



FIGURE 7. Created video environment.

Open source Computer Vision library (OpenCV) and Compute Unified Device Architecture (CUDA). To obtain ground truth information with real images, our simulator renders the UAV CAD model in a real captured video sequence as illustrated in Figure 7. The trajectory of the UAV during the landing sequence can be seen in Figure 8.

A. TESTS DESCRIPTION

We have tested three different filter combinations (see Table 1). In combination 1, we apply a UKF for both translation and rotation filtering [8]. In combination 2, we apply

TABLE 1. Performed combinations description.

Combination	Particle number	Database particles	Filtering
1	N	25	UKF
2	N	25	UKF+UBiF
3	N	25	UKF+UBiGaF

a UKF for the translational motion filtering and a UBiF for the rotational motion filtering [3], [12]. In combination 3, we apply a UKF for the translational motion filtering and the novel UBiGaF for the rotational motion filtering. We also test different particle budgets $N = \{50, 100, 300, 500, 1500\}$. In the first filter iteration, all the particles come from the hypotheses generation scheme (Section III-A) trained in the interval $[-90^\circ, 90^\circ]$ and the velocities are initialized with zero. Every time we capture a new frame, we update the particle vector with 25 new particles coming from the hypotheses generation scheme to reinforce the use of the most recent observation. The time between new frames is $\Delta t \cong 0.034$ seconds. The temporal filters use a Bi process noise P_B^Φ with $\mathbf{M}_t^\Phi = \mathbf{I}_{4 \times 4}$ and $\mathbf{Z}_t^\Phi = \text{diag}(-150, -150, -150, 0)$ and a Bi measurement noise P_B^Λ with $\mathbf{M}_t^\Lambda = \mathbf{I}_{4 \times 4}$ and $\mathbf{Z}_t^\Lambda = \text{diag}(-800, -800, -800, 0)$.

B. PERFORMANCE METRICS

On each iteration, the state estimation (Figure 4) is given by an approximation to the mode of the distribution given by the particle with the highest importance weight before resampling. The translation error was obtained using the *Euclidean* distance, and the rotation error was obtained according to:

$$\delta(R_g, R_r) = \sqrt{\frac{\| \log_m (R_g^T R_r) \|_F^2}{2} \frac{180}{\pi}} \quad [deg] \quad (22)$$

where R_g corresponds to our ground truth rotation matrix and R_r corresponds to the retrieved rotation matrices. We have also obtained the Standard Deviation (SD), the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE).

C. TRANSLATION ERROR ANALYSIS

When we analyze Table 2, we can see that the lowest 90% error interval, RMSE and SD happens when we use, as expected, the highest number of particles. From the tested combinations, UKF+UBiGaF presents the best overall translation performance since the error starts getting close to zero when we get closer to the landing area as plotted in Figure 9. The particle perturbation caused by the correlation between angular velocity and attitude adds more diversity to the set, increasing the precision of the obtained estimate.

D. ROTATION ERROR ANALYSIS

As described in Table 3, the rotation error is higher in the UKF combination where we have the largest 90% error intervals and the highest MAE and SD values. This error is mainly motivated by the poor orientation Gaussian model approximation. When comparing UKF+UBiF and UKF+UBiGaF,

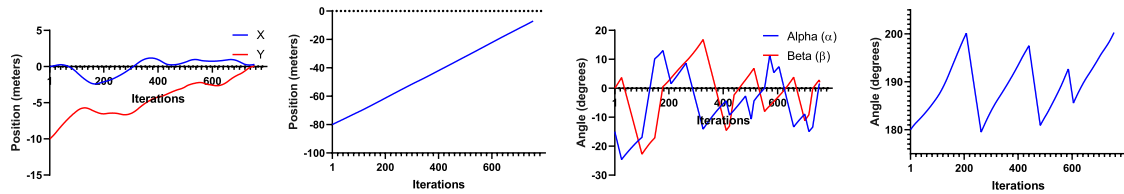


FIGURE 8. Tested landing sequence: $\{X, Y\}$ (left), Z (center left), $\{\alpha, \beta\}$ (center right) and γ (right).

TABLE 2. Translation error (meters).

Combination	Particle number (N)	5% Percentile	25% Percentile	Median	75% Percentile	95% Percentile	MAE	RMSE	SD
UKF	50	0.21	1.17	4.38	8.07	14.80	2.03	1.94	5.18
	100	0.33	1.55	7.78	8.64	18.64	2.25	2.19	5.72
	300	0.26	0.93	3.53	7.50	21.34	2.04	1.88	5.99
	500	0.51	1.31	2.56	6.28	16.10	1.82	1.39	5.19
	1500	1.30	2.99	5.35	7.11	8.90	2.18	0.90	2.94
UKF + UBIF	50	0.20	1.15	4.28	8.41	16.17	2.10	1.98	5.50
	100	0.22	1.42	4.84	9.64	20.72	2.40	2.26	6.40
	300	0.29	1.12	2.99	8.79	19.01	2.07	1.78	5.86
	500	0.54	1.36	3.20	7.17	17.74	1.98	1.29	5.51
	1500	1.37	3.17	5.57	7.40	14.21	2.39	0.70	3.99
UKF + UBIGaF	50	0.29	1.28	4.73	8.87	17.44	2.20	2.14	5.48
	100	0.27	1.39	4.84	9.31	17.99	2.28	2.20	5.80
	300	0.29	1.46	4.87	9.20	19.99	2.39	2.24	6.39
	500	0.34	1.30	4.93	9.10	21.01	2.39	2.28	6.47
	1500	0.31	1.30	4.44	9.86	16.26	2.24	2.09	5.33

TABLE 3. Rotation error (degrees).

Combination	Particle number (N)	5% Percentile	25% Percentile	Median	75% Percentile	95% Percentile	MAE	RMSE	SD
UKF	50	2.26	6.04	10.47	16.79	48.50	7.35	2.24	16.54
	100	2.25	6.26	10.99	19.13	90.36	8.70	1.41	20.76
	300	2.99	6.68	11.40	20.46	117.50	10.12	3.30	24.00
	500	3.91	8.47	13.00	21.03	111.00	9.95	0.77	21.76
	1500	8.07	13.29	18.55	26.51	113.60	12.99	0.51	24.15
UKF + UBIF	50	1.68	4.34	7.95	12.57	24.04	4.44	1.97	6.75
	100	1.53	3.49	6.78	11.79	23.13	4.00	1.82	6.03
	300	2.41	4.64	6.70	12.49	20.05	4.25	1.57	6.12
	500	3.49	7.40	9.98	14.48	24.71	5.49	1.71	7.27
	1500	7.12	11.18	15.98	21.63	35.35	8.50	1.65	10.90
UKF + UBIGaF	50	1.25	4.46	8.63	14.03	23.88	4.58	2.13	7.21
	100	1.62	4.20	8.47	12.83	23.48	4.40	2.24	6.59
	300	1.74	4.06	7.32	13.26	22.37	3.95	1.54	6.05
	500	1.33	3.80	6.98	12.20	21.93	4.05	1.40	6.22
	1500	1.17	3.48	6.73	12.39	20.29	3.90	0.90	6.01

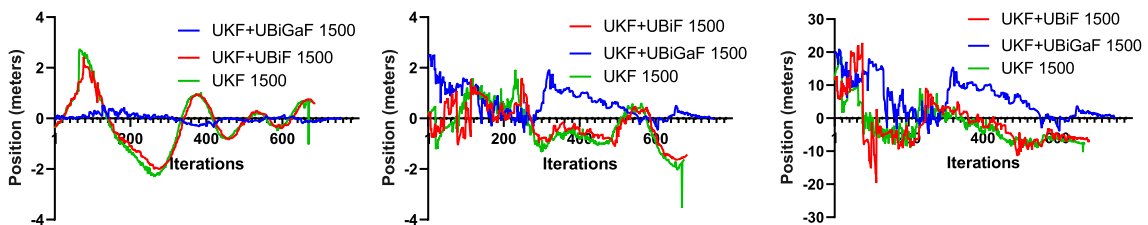


FIGURE 9. Obtained translation error (meters): X (left), Y (center) and Z (right).

the rotation error is higher in the vast majority of combinations for UKF+UBiF. When we increase the number of particles to $N = 1500$, the rotation error in UBIF is higher by approximately 50% when compared with the UBIGaF. In one overall analysis, the obtained rotation error in both

combinations starts getting close to zero when we get closer to the landing area (Figure 10). The rotation error for UKF, UKF+UBiF and UKF+UBiGaF using $N = 1500$ can be seen in Figure 9 and Figure 10. An example of real video tracking for UBIGaF using $N = 1500$ can be seen in Figure 11.

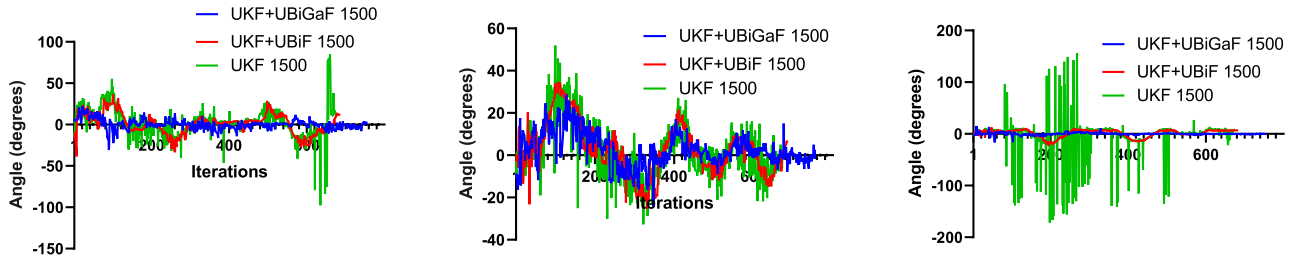


FIGURE 10. Obtained rotation error (degrees): α (left), β (center) and γ (right).

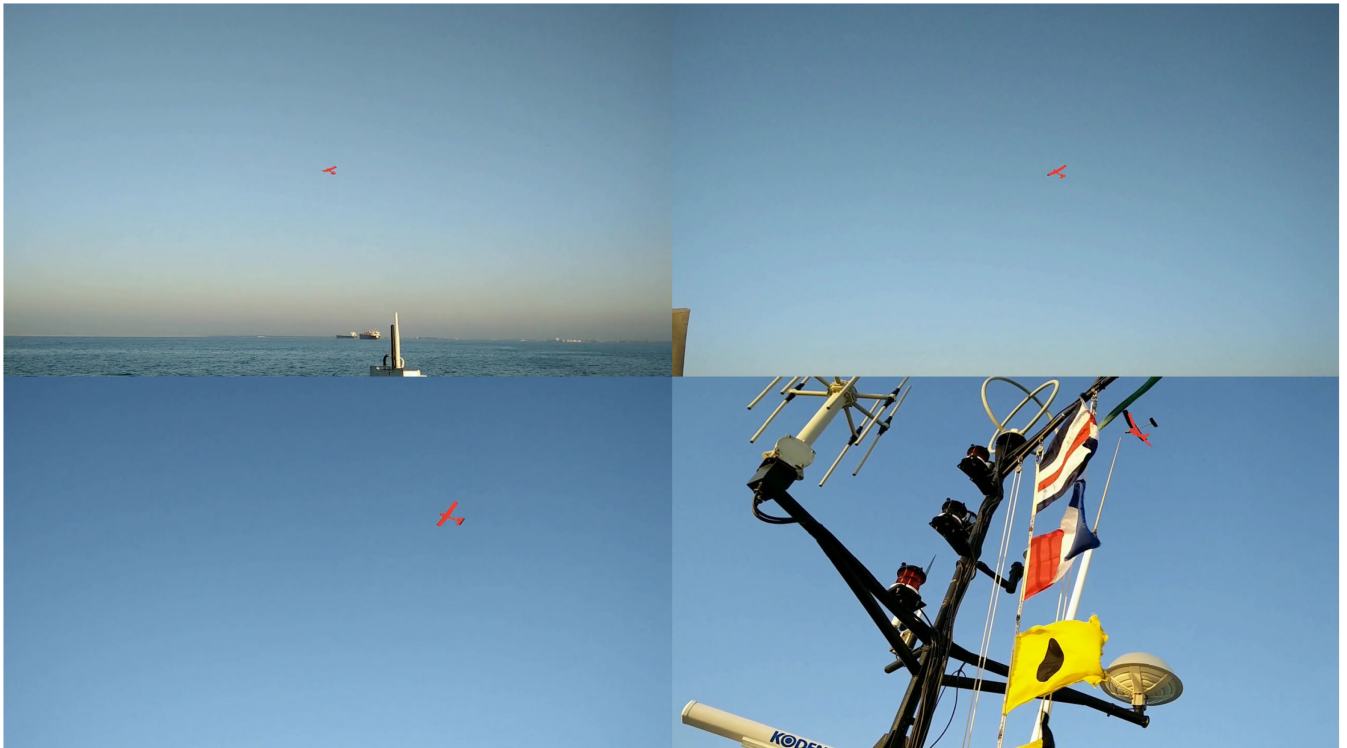


FIGURE 11. Real video estimate example using UKF+UBiGaF using $N = 1500$ (red pixels).

VI. CONCLUSION AND FUTURE WORK

A ground-based 3D model-based approach for UAV tracking using Unscented filters was introduced and tested. The proposed tracking architecture incorporate directional statistics on a UPF scheme, combining a UKF for translational motion filtering with a UBiF or UBiGaF for rotational motion filtering. The developed new filter UBiGaF was able to represent the covariance between the angular position and velocity, obtaining better results than previous approaches. The UBiF rotation error is 50% higher when compared with the UBiGaF when we use 1500 particles. Future work will focus on the inclusion of aerodynamics constraints in the UAV motion model.

APPENDIXES

APPENDIX A

UNSCENTED KALMAN FILTER

In the UKF, a Gaussian approximation to the distributions of the n -dimensional state and process noises are used to generate a set of points (sigma points) that are sufficient

to represent their statistics using a UT. The process noise covariance \mathbf{Q}_{t-1} ($n \times n$ matrix) and the state covariance \mathbf{P}_{t-1} ($n \times n$ matrix) are transformed into a $2n$ set of points $\delta \mathbf{x}_{t-1}(i)$ that represent perturbations to the current state according to:

$$\delta \mathbf{x}_{t-1}(i) = \pm \left(\sqrt{\iota \cdot (\mathbf{P}_{t-1} + \mathbf{Q}_{t-1})} \right)_i \quad i = 1, \dots, 2n \quad (23)$$

the parameter ι is a scaling parameter given by $\iota = \alpha^2(n+k)$, where α is a positive real ($0 \leq \alpha \leq 1$) parameter that controls the high order effects resulting from the existing nonlinearity, k is another real parameter ($k \geq 0$) that will control the distance between the sigma points and their average [41]. The matrices \mathbf{P}_{t-1} and \mathbf{Q}_{t-1} are symmetric and positive definite, so it is possible to use the *Cholesky* decomposition to compute $\sqrt{\iota \cdot (\mathbf{P}_{t-1} + \mathbf{Q}_{t-1})}$ [42]. The computation of the sigma points \mathcal{X}_i is now done by adding directly $\delta \mathbf{x}_t(i)$ to the mean value of the state vector \mathbf{x}_t according to:

$$\mathcal{X}_i = \mathbf{x}_{t-1} + \delta \mathbf{x}_{t-1}(i) \quad i = 1, \dots, 2n \text{ and } \mathcal{X}_0 = \mathbf{x}_t \quad (24)$$

A. PREDICTION

The process model $\mathbf{F}(\cdot)$ is then applied to the obtained sigma points, generating the transformed sigma points:

$$\mathcal{X}'_i = \mathbf{F}(\mathcal{X}_i, 0) \quad i = 0, \dots, 2n \quad (25)$$

No additional noise is considered at this step because the noise was already added at the sigma point's creation step (24). The *a priori* state estimate is obtained calculating the mean of the transformed sigma points \mathcal{X}'_i according to:

$$\bar{\mathbf{x}}_t = \sum_{i=0}^{2n} W_i^m \mathcal{X}'_i \quad (26)$$

The weights are given by [43]:

$$W_0^m = \frac{\lambda}{n + \lambda} \text{ and } W_i^m = W_i^c = \frac{1}{2(n + \lambda)} \quad (27)$$

with λ given by $\lambda = \alpha^2(n + k) - n$.

To estimate the *a priori* state covariance each propagated sigma point is removed from its mean to create the set of error vectors:

$$\delta \bar{\mathbf{x}}_t(i) = \mathcal{X}'_i - \bar{\mathbf{x}}_t \quad (28)$$

then:

$$\mathbf{P}_t^{xx} = \sum_{i=0}^{2n} W_i^c \delta \bar{\mathbf{x}}_t(i) \delta \bar{\mathbf{x}}_t(i)^T \quad (29)$$

where the scaling weights W_i^c are given by (27), except W_0^c alternatively given by [44]:

$$W_0^c = \frac{\lambda}{n + \lambda} + (1 - \alpha^2 + \beta) \quad (30)$$

where β is a non-negative term which incorporates knowledge of the higher order moments(the chosen α and β determine the accuracy of third and higher order moments for non-Gaussian inputs [43]) of the distribution [41].

The transformed sigma points are now projected into the measurement space according to:

$$\mathcal{Z}_i = \mathbf{H}(\mathcal{X}'_i, 0) \quad (31)$$

The measurement expected value is computed as:

$$\bar{\mathbf{z}}_t = \sum_{i=1}^{2n} W_i^m \mathcal{Z}_i \quad (32)$$

B. MEASUREMENT UPDATE

The measurement covariance estimate \mathbf{P}_t^{zz} is given by:

$$\mathbf{P}_t^{zz} = \sum_{i=0}^{2n} W_i^c [\mathcal{Z}_i - \bar{\mathbf{z}}_t][\mathcal{Z}_i - \bar{\mathbf{z}}_t]^T \quad (33)$$

The innovation vector \mathbf{v}_t is obtained comparing the actual measurement \mathbf{z}_t to the measurement estimate $\bar{\mathbf{z}}_t$:

$$\mathbf{v}_t = \mathbf{z}_t - \bar{\mathbf{z}}_t \quad (34)$$

The innovation covariance \mathbf{P}_t^{vv} is obtained adding the measurement noise \mathbf{R}_t to the measurement covariance \mathbf{P}_t^{zz} :

$$\mathbf{P}_t^{vv} = \mathbf{P}_t^{zz} + \mathbf{R}_t \quad (35)$$

The cross-correlation matrix \mathbf{P}_t^{xz} is obtained from \mathcal{Z}_i and \mathcal{X}'_i , according to:

$$\mathbf{P}_t^{xz} = \sum_{i=0}^{2n} W_i^c [\mathcal{X}'_i - \bar{\mathbf{x}}_t][\mathcal{Z}_i - \bar{\mathbf{z}}_t]^T \quad (36)$$

The Kalman gain is then computed from:

$$\mathbf{K}_t = \mathbf{P}_t^{xz}(\mathbf{P}_t^{vv})^{-1} \quad (37)$$

Finally, the *a posteriori* state estimate is obtained according to:

$$\mathbf{x}_t = \bar{\mathbf{x}}_t + \mathbf{K}_t \mathbf{v}_t \quad (38)$$

and the state covariance \mathbf{P}_t is given by:

$$\mathbf{P}_t = \mathbf{P}_t^{xx} - \mathbf{K}_t \mathbf{P}_t^{vv} \mathbf{K}_t^T \quad (39)$$

**APPENDIX B
BINGHAM DISTRIBUTION**

The Bi distribution is an antipodally symmetric distribution (opposite points on S have equal probability) that represents a zero-mean Gaussian distribution in \mathbb{R}^d projected on the unit hypersphere S^{d-1} [9], [35]. The Probability Density Function (PDF) for the Bi distribution is obtained by [35]:

$$P_B(\mathbf{q}; \mathbf{M}, \mathbf{Z}) = \frac{1}{F(\mathbf{Z})} \exp(\mathbf{q}^T \mathbf{M} \mathbf{Z} \mathbf{M}^T \mathbf{q}) \quad (40)$$

where $\mathbf{q} \in S^{d-1} \subset \mathbb{R}^d : \|\mathbf{q}\| = 1$ is a unit vector (when using quaternions $d = 4$), $\mathbf{M} \in \mathbb{R}^{d \times d}$ is an orthogonal matrix (a square matrix whose columns and rows are orthonormal vectors) describing the orientation of the distribution, $F(\mathbf{Z})$ is the normalization constant and $\mathbf{Z} = \text{diag}(z_1, z_2, \dots, z_{d-1}, 0)$ with nondecreasing negative diagonal elements is the concentration matrix that controls the spread of the distribution around its mean. Adding a multiple of the identity matrix to \mathbf{Z} or changing the order of a column of \mathbf{M} and the corresponding \mathbf{Z} columns do not change the distribution, so we can force the last entry of \mathbf{Z} to be zero for computational simplicity, and because of this, the last column of \mathbf{M} represents the distribution mode [35], [38].

A. NORMALIZATION CONSTANT

The main difficulty in the utilization of the Bi distribution consists in the computation of the normalization constant since the distribution must integrate to one over its domain. The normalization constant is obtained by:

$$F(\mathbf{Z}) = \int_{S^{d-1}} \exp(\mathbf{q}^T \mathbf{M} \mathbf{Z} \mathbf{M}^T \mathbf{q}) d\mathbf{q} = \int_{S^{d-1}} \exp(\mathbf{q}^T \mathbf{Z} \mathbf{q}) d\mathbf{q} \quad (41)$$

where $F(\mathbf{Z})$ does not depend on the matrix \mathbf{M} , since the orientation of the distribution peaks does not change its value. Since we are using this distribution in a real-time approach, we choose to interpolate tabulated values from a precomputed lookup table for computational efficiency.

B. PRODUCT

The product of two Bi distributions is closed under multiplication after renormalization and is given by [37]:

$$P_{B_1}(\mathbf{M}_1, \mathbf{Z}_1) \cdot P_{B_2}(\mathbf{M}_2, \mathbf{Z}_2) = \frac{1}{F(\mathbf{Z})} \exp(\mathbf{q}^T \mathbf{M} \mathbf{Z} \mathbf{M}^T \mathbf{q}) \quad (42)$$

where $F(\mathbf{Z})$ is the new normalization constant, \mathbf{M} are the unit eigenvectors and \mathbf{D} are the eigenvalues on the diagonal in ascending order of $(\mathbf{M}_1 \mathbf{Z}_1 \mathbf{M}_1^T + \mathbf{M}_2 \mathbf{Z}_2 \mathbf{M}_2^T)$, $\mathbf{Z} = \mathbf{D} - \lambda_{dd} \mathbf{I}_{d \times d}$ and λ_{dd} is the largest eigenvalue.

C. ROTATION

When we are in S^3 is possible to change (rotate) the orientation of a Bi distribution $P_B(\mathbf{q}; \mathbf{M}, \mathbf{Z})$ (40) by a fixed quaternion $\mathbf{g} \in S^3$ according to [9], [11], [38]:

$$P_B(\mathbf{r}; \mathbf{M} \otimes \mathbf{g}, \mathbf{Z}) \quad \text{when} \quad \mathbf{r} = \mathbf{q} \otimes \mathbf{g} \quad (43)$$

where \otimes represents the composition of orientations, $\mathbf{M} \otimes \mathbf{g} \equiv [\mathbf{m}_1 \otimes \mathbf{g}, \mathbf{m}_2 \otimes \mathbf{g}, \mathbf{m}_3 \otimes \mathbf{g}, \mathbf{m}_4 \otimes \mathbf{g}]$ and \mathbf{m} are the columns of \mathbf{M} . Since the quaternion multiplication is not commutative we have that $\mathbf{r} \sim P_B(\mathbf{g} \otimes \mathbf{M}, \mathbf{Z})$ when $\mathbf{r} = \mathbf{g} \otimes \mathbf{q}$.

D. COVARIANCE

The covariance is a sufficient statistics for the Bi distribution since the Bi distribution is the maximum entropy distribution on the hypersphere which matches the sample inertia matrix [45]. The covariance of the Bi PDF is given by [35]:

$$\text{Cov}(\mathbf{q}) = E(\mathbf{q}\mathbf{q}^T) - (E(\mathbf{q}))^2 = E(\mathbf{q}\mathbf{q}^T) \quad (44)$$

where $(E(\mathbf{q}))^2 = 0$ is a consequence of the antipodal symmetry and $E(\mathbf{q}\mathbf{q}^T)$ is given by:

$$\mathbf{M} \cdot \text{diag} \left(\frac{\frac{d}{dz_1} F(\mathbf{Z})}{F(\mathbf{Z})}, \dots, 1 - \frac{\sum_{i=1}^{d-1} \frac{d}{dz_i} F(\mathbf{Z})}{F(\mathbf{Z})} \right) \cdot \mathbf{M}^T \quad (45)$$

where the values of the gradient of F with respect to \mathbf{Z} are precomputed and accessed by interpolation as made for the normalization constant.

The covariance of the composition (composition is a directional analog to the addition of random vectors in linear space) of two Bi distributions can be obtained using the method of moments [11]. Using this method, we can approximate the resulting composition covariance directly from their covariance matrices combination.

E. INFERENCE

It is possible to estimate the parameters of a Bi distribution which approximates a set of samples [35]. The inertia matrix for a set of N samples $\mathbf{q} = [\mathbf{q}_1, \dots, \mathbf{q}_N]$ is given by [35]:

$$\mathbf{S} = \frac{1}{N} \sum_{i=1}^N \mathbf{q}_i \mathbf{q}_i^T \quad (46)$$

The Maximum Likelihood Estimation (MLE) $\hat{\mathbf{M}}$ for a set of samples is an eigenvalue problem since the columns of $\hat{\mathbf{M}}$ are

eigenvectors κ of \mathbf{S} [35]. The MLE $\hat{\mathbf{Z}}$ can be found setting the partial log-likelihood function on \mathbf{Z} to zero. This leads to:

$$\frac{\frac{d}{dz_j} F(\mathbf{Z})}{F(\mathbf{Z})} = \frac{1}{N} \sum_{i=1}^N (\kappa_j^T \mathbf{q}_i)^2 = \kappa_j^T \mathbf{S} \kappa_j = 0 \quad (47)$$

where κ_j are the eigenvectors of \mathbf{S} . This calculation can be made using the Constrained Optimization BY Linear Approximations (COBYLA) algorithm [46].

F. SAMPLING

Is hard to sample directly from the Bi distribution because of the normalization constant. To solve this problem is used a *Metropolis-Hasting* sampler [47] with proposal distribution given by a projected zero-mean Gaussian with covariance \mathbf{S} (either from (45) or (46)) and target distribution provided by the Bi density [11], [35].

APPENDIX C

BINGHAM FILTER SIGMA POINTS CREATION

The created sigma points follow the same principle as used in the UT applied in the UKF for the translation case (Appendix A). We need to use $4d - 2$ samples that correspond in this case to fourteen samples ($d = 4$). Since the distribution is antipodally symmetric, it is sufficient to consider only one pole (adapting the respective weights). The canonical sigma points are given by:

$$\tilde{\mathbf{q}}^{1,2} = [\pm \sin \alpha_1, 0, 0, \cos \alpha_1]^T \quad (48)$$

$$\tilde{\mathbf{q}}^{3,4} = [0, \pm \sin \alpha_2, 0, \cos \alpha_2]^T \quad (49)$$

$$\tilde{\mathbf{q}}^{5,6} = [0, 0, \pm \sin \alpha_3, \cos \alpha_3]^T \quad (50)$$

$$\tilde{\mathbf{q}}^7 = [0, 0, 0, 1]^T \quad (51)$$

where $\tilde{\mathbf{q}}^7$ is the sample located on the pole (identity quaternion). The canonical distribution will be employed since it simplifies the needed mathematical approach because the parameters will be dimensionless.

The covariance of the estimated Bi distribution is obtained by (45):

$$E_{P_B} \{ \mathbf{x}_t \mathbf{x}_t^T \} = \mathbf{M} \cdot \text{diag}(f_1, f_2, f_3, f_4) \cdot \mathbf{M}^T \quad (52)$$

The deviation for each one of the canonical sigma points is obtained from α_i :

$$\alpha_i = \sin^{-1} \sqrt{\left(\frac{3f_i}{3f_i + \left(1 - \frac{1}{N}\right)f_4} \right)} \quad (53)$$

The weights of the sigma points are given by:

$$w^{1,2} = \frac{f_1 + \frac{1 - \frac{1}{N}f_4}{3}}{4} \quad (54)$$

$$w^{3,4} = \frac{f_2 + \frac{1 - \frac{1}{N}f_4}{3}}{4} \quad (55)$$

$$w^{5,6} = \frac{f_3 + \frac{1 - \frac{1}{N}f_4}{3}}{4} \quad (56)$$

$$w^7 = \frac{f_4}{N} \quad (57)$$

where N is equal to the number of used sigma points and w^7 is the weight for the central sigma point. Each canonical sigma point $\tilde{\mathbf{q}}$ is multiplied by \mathbf{M} (in the UBiF we use $\mathbf{M}_i^e \tilde{\mathbf{q}}$ as described in Section IV-D) originating the set of sigma points \mathbf{q} that represent our Bi distribution P_B . The sigma point propagation is made adding a quaternion motion based on the angular velocities with noise to each one of the sigma points.

APPENDIX D BINGHAM-GAUSS

Is possible to use the definition of conditional probability to construct a distribution that consists in the product of a Bi distribution and a Gaussian distribution conditioned on the Bi distributed random variables. The BiGa PDF is given by [7]:

$$P_{BG}(\mathbf{q}, \boldsymbol{\omega}; \mathbf{M}, \mathbf{Z}, \mathbf{m}_\omega, \mathbf{P}_q, \mathbf{P}_\omega, \mathbf{P}_{q\omega}) \\ = P_G(\boldsymbol{\omega}; \mathbf{m}_\omega + \mathbf{P}_{q\omega}^T \mathbf{P}_q^{-1} \mathbf{q}, \mathbf{P}_\omega - \mathbf{P}_{q\omega}^T \mathbf{P}_q^{-1} \mathbf{P}_{q\omega}) P_B(\mathbf{q}; \mathbf{M}, \mathbf{Z}) \quad (58)$$

where \mathbf{M} and \mathbf{Z} are the orientation matrix and the matrix of concentration parameters of the Bi part defined by \mathbf{P}_q (46) and P_G is a Gaussian PDF given by:

$$P_G(\boldsymbol{\omega}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2) = \frac{1}{\sqrt{2\pi\boldsymbol{\sigma}^2}} e^{-\left(\frac{\boldsymbol{\omega}-\boldsymbol{\mu}}{\boldsymbol{\sigma}}\right)^2} \quad (59)$$

with mean $\boldsymbol{\mu} = \mathbf{m}_\omega + \mathbf{P}_{q\omega}^T \mathbf{P}_q^{-1} \mathbf{q}$ and variance $\boldsymbol{\sigma}^2 = \mathbf{P}_\omega - \mathbf{P}_{q\omega}^T \mathbf{P}_q^{-1} \mathbf{P}_{q\omega}$ as shown in (58).

A. DISTRIBUTION PARAMETERS

The parameters \mathbf{m}_ω , \mathbf{P}_ω , \mathbf{P}_q and $\mathbf{P}_{q\omega}$ (58) are given by:

$$\mathbf{m}_\omega = E_{P_{BG}}[\boldsymbol{\omega}] \in \mathbb{R}^r \quad (60)$$

$$\mathbf{P}_\omega = E_{P_{BG}}[(\boldsymbol{\omega} - \mathbf{m}_\omega)(\boldsymbol{\omega} - \mathbf{m}_\omega)^T] \in \mathbb{R}^{r \times r} \quad (61)$$

$$\mathbf{P}_q = E_{P_{BG}}[\mathbf{q}\mathbf{q}^T] \in \mathbb{R}^{d \times d} \quad (62)$$

$$\mathbf{P}_{q\omega} = E_{P_{BG}}[\mathbf{q}(\boldsymbol{\omega} - \mathbf{m}_\omega)^T] \in \mathbb{R}^{d \times r} \quad (63)$$

B. ANTIPODAL SYMMETRY

We have to guarantee that this distribution is antipodally symmetric with \mathbf{q} and $-\mathbf{q}$ representing the same attitude. The PDF described in (58) needs to be divided into two hemispheres of the unit hypersphere to guarantee that condition [7]. The BiGa PDF becomes represented as:

$$P_{BG} = \begin{cases} P_{BG}(\mathbf{q}, \boldsymbol{\omega}; \mathbf{m}_\omega, \mathbf{P}_\omega, \mathbf{P}_q, \mathbf{P}_{q\omega}) & \mathbf{q} \in \mathbb{S}^+ \\ P_{BG}(\mathbf{q}, \boldsymbol{\omega}; \mathbf{m}_\omega, \mathbf{P}_\omega, \mathbf{P}_q, -\mathbf{P}_{q\omega}) & \mathbf{q} \in \mathbb{S}^- \end{cases} \quad (64)$$

where \mathbb{S}^+ and \mathbb{S}^- represent the hemispheres. For each \mathbf{q} its position on the hypersphere is obtained analyzing its last nonzero element. If it is negative the quaternion belongs to \mathbb{S}^- otherwise belongs to \mathbb{S}^+ .

APPENDIX E BINGHAM-GAUSS FILTER SIGMA POINTS CREATION

We follow the same approach to create the canonical sigma points as described in Appendix C for the Bi case but adding a

Euclidean part describing the angular velocity. The canonical sigma points that represent the deviation for the *Euclidean* part are defined as [10]:

$$\tilde{\mathbf{r}}^{1,2} = \left[\overbrace{[[0, 0, 0, 1]^T]}^{\tilde{\mathbf{q}}^{1,2}}, \overbrace{[\pm\delta, 0, 0]^T]}^{\tilde{\boldsymbol{\omega}}^{1,2}} \right]^T \quad (65)$$

$$\tilde{\mathbf{r}}^{3,4} = \left[\overbrace{[[0, 0, 0, 1]^T]}^{\tilde{\mathbf{q}}^{3,4}}, \overbrace{[0, \pm\delta, 0]^T]}^{\tilde{\boldsymbol{\omega}}^{3,4}} \right]^T \quad (66)$$

$$\tilde{\mathbf{r}}^{5,6} = \left[\overbrace{[[0, 0, 0, 1]^T]}^{\tilde{\mathbf{q}}^{5,6}}, \overbrace{[0, 0, \pm\delta]^T]}^{\tilde{\boldsymbol{\omega}}^{5,6}} \right]^T \quad (67)$$

The angular deviations in the first three states of the attitude quaternion are introduced while the *Euclidean* part is held constant at zero to guarantee that the perturbed quaternion remains on the unit hypersphere according to [10]:

$$\tilde{\mathbf{r}}^{7,8} = \left[\overbrace{[[\pm \sin \alpha_1, 0, 0, \cos \alpha_1]^T]}^{\tilde{\mathbf{q}}^{7,8}}, \overbrace{[0, 0, 0]^T]}^{\tilde{\boldsymbol{\omega}}^{7,8}} \right]^T \quad (68)$$

$$\tilde{\mathbf{r}}^{9,10} = \left[\overbrace{[[0, \pm \sin \alpha_2, 0, \cos \alpha_2]^T]}^{\tilde{\mathbf{q}}^{9,10}}, \overbrace{[0, 0, 0]^T]}^{\tilde{\boldsymbol{\omega}}^{9,10}} \right]^T \quad (69)$$

$$\tilde{\mathbf{r}}^{11,12} = \left[\overbrace{[[0, 0, \pm \sin \alpha_3, \cos \alpha_3]^T]}^{\tilde{\mathbf{q}}^{11,12}}, \overbrace{[0, 0, 0]^T]}^{\tilde{\boldsymbol{\omega}}^{11,12}} \right]^T \quad (70)$$

The central sigma point is given by:

$$\tilde{\mathbf{r}}^{13} = \left[\overbrace{[[0, 0, 0, 1]^T]}^{\tilde{\mathbf{q}}^{13}}, \overbrace{[0, 0, 0]^T]}^{\tilde{\boldsymbol{\omega}}^{13}} \right]^T \quad (71)$$

The parameters α_i and δ are given by:

$$\alpha_i = \sin^{-1} \left(\sqrt{\frac{f_i}{\boldsymbol{\omega}_{\mathbf{B}_i}}} \right) \quad \text{and} \quad \delta = \sqrt{\frac{r}{\boldsymbol{\omega}_G}} \quad (72)$$

The weights for the sigma points one to six are given by:

$$w^{1,\dots,6} = \frac{\boldsymbol{\omega}_G}{4r} = \frac{2rf_{s+1}}{N4r} = \frac{2f_{s+1}}{4(2r + 2s + 1)} \quad (73)$$

where r and s are equal to three in this case, N is equal to thirteen (the number of sigma points) and f_{s+1} is obtained analyzing the second moment of the canonical BiGa distribution (the zeroth and first moment is one and zero respectively) according to:

$$E_{P_{BG}}\{zz^T\} = \text{diag}[[1 \ 1 \ 1] [f_1 f_2 f_3 f_4]] \quad (74)$$

where the covariance can be obtained as described in (45) for the Bi part of the BiGa distribution alone. The weights for the sigma points seven to twelve are given by:

$$w^{7,8} = \frac{\boldsymbol{\omega}_{\mathbf{B}_1}}{4} = \frac{f_1 + \frac{1-\frac{1}{N}-\frac{2r}{N}f_4}{s}}{4} \quad (75)$$

$$w^{9,10} = \frac{\boldsymbol{\omega}_{\mathbf{B}_2}}{4} = \frac{f_2 + \frac{1-\frac{1}{N}-\frac{2r}{N}f_4}{s}}{4} \quad (76)$$

$$w^{11,12} = \frac{\boldsymbol{\omega}_{\mathbf{B}_3}}{4} = \frac{f_3 + \frac{1-\frac{1}{N}-\frac{2r}{N}f_4}{s}}{4} \quad (77)$$

The weight for the central sigma point is given by:

$$w^{13} = \frac{\omega_C}{2} = \frac{f_4}{2N} \quad (78)$$

Each sigma point is transformed from the canonical representation using the following relations [7]:

$$\mathbf{q} = M\mathbf{p} \quad (79)$$

$$\boldsymbol{\omega} = \sqrt{\mathbf{P}_\omega + \mathbf{P}_{\mathbf{q}\omega}^T \mathbf{P}_{\mathbf{q}}^{-1} \mathbf{P}_{\mathbf{q}\omega}} \mathbb{Z} + \mathbf{P}_{\mathbf{q}\omega}^T \mathbf{P}_{\mathbf{q}}^{-1} M\mathbf{p} + \mathbf{m}_\omega \quad \mathbf{q} \in \mathbb{S}^{s+} \quad (80)$$

$$\boldsymbol{\omega} = \sqrt{\mathbf{P}_\omega + \mathbf{P}_{\mathbf{q}\omega}^T \mathbf{P}_{\mathbf{q}}^{-1} \mathbf{P}_{\mathbf{q}\omega}} \mathbb{Z} - \mathbf{P}_{\mathbf{q}\omega}^T \mathbf{P}_{\mathbf{q}}^{-1} M\mathbf{p} + \mathbf{m}_\omega \quad \mathbf{q} \in \mathbb{S}^{s-} \quad (81)$$

where \mathbf{p} corresponds to the quaternion part and \mathbb{Z} correspond to the *Euclidean* part of the created canonical sigma points \mathbf{z} originating the \mathcal{Z} sigma points. This transformation is similar to what is performed in the Bi case but now taking into account the *Euclidean* part of the sigma point vector as seen in (80) and (81).

The parameters \mathbf{m}_ω , \mathbf{P}_ω , $\mathbf{P}_{\mathbf{q}}$ and $\mathbf{P}_{\mathbf{q}\omega}$ as described in (60) to (63) can be obtained from the sigma points according to:

$$\mathbf{m}_\omega \approx 2 \sum_{i=1}^N w^i f_\omega(\mathbf{z}^i) \quad (82)$$

$$\mathbf{P}_\omega \approx 2 \sum_{i=1}^N w^i (f_\omega(\mathbf{z}^i) - \mathbf{m}_\omega) (f_\omega(\mathbf{z}^i) - \mathbf{m}_\omega)^T \quad (83)$$

$$\mathbf{P}_{\mathbf{q}} \approx 2 \sum_{i=1}^N w^i f_{\mathbf{q}}(\mathbf{z}^i) f_{\mathbf{q}}(\mathbf{z}^i)^T \quad (84)$$

$$\mathbf{P}_{\mathbf{q}\omega} \approx 2 \sum_{i=1}^N w^i f_{\mathbf{q}}(\mathbf{z}^i) (f_\omega(\mathbf{z}^i) - \mathbf{m}_\omega)^T \quad (85)$$

where \mathbf{z}^i is the sigma point i , f_ω is the angular velocity part of the considered sigma point and $f_{\mathbf{q}}$ is the quaternion part of the considered sigma point.

REFERENCES

[1] K. W. Williams, "A summary of unmanned aircraft accident/incident data: Human factors implications," Civil Aersp. Med. Inst., Federal Aviation Admin., Washington, DC, USA, Tech. Rep. DOT/FAA/AM-04/24, 2004.

[2] G. Wild, J. Murray, and G. Baxter, "Exploring civil drone accidents and incidents to help prevent potential air disasters," *Aerospace*, vol. 3, no. 3, p. 22, Jul. 2016.

[3] N. P. Santos, V. Lobo, and A. Bernardino, "Unmanned aerial vehicle tracking using a particle filter based approach," in *Proc. IEEE Underwater Technol.(UT)*, Kaohsiung, Taiwan, Apr. 2019, pp. 1–10.

[4] J. R. Azinheira and P. Rives, "Image-based visual servoing for vanishing features and ground lines tracking: Application to a UAV automatic landing," *Int. J. Optomechatronics*, vol. 2, no. 3, pp. 275–295, Sep. 2008.

[5] D. Eynard, P. Vasseur, C. Demonceaux, and V. Frémont, "Real time UAV altitude, attitude and motion estimation from hybrid stereovision," *Auto. Robots*, vol. 33, nos. 1–2, pp. 157–172, Aug. 2012.

[6] N. Metni, T. Hamel, and F. Derkx, "Visual tracking control of aerial robotic systems with adaptive depth estimation," in *Proc. 44th IEEE Conf. Decis. Control*, Oct. 2006, pp. 6078–6084.

[7] J. Darling and K. J. DeMars, "The bingham-gauss mixture filter for pose estimation," in *Proc. AIAA/AAS Astrodynamics Spec. Conf.*, Sep. 2016, p. 5631.

[8] N. P. Santos, V. Lobo, and A. Bernardino, "A ground-based vision system for UaV tracking," in *Proc. OCEANS* Genova, Italy, May 2015, pp. 1–9.

[9] I. Gilitschenski, G. Kurz, S. J. Julier, and U. D. Hanebeck, "Unscented orientation estimation based on the bingham distribution," *IEEE Trans. Autom. Control*, vol. 61, no. 1, pp. 172–177, Jan. 2016.

[10] J. E. Darling and K. J. DeMars, "Rigid body attitude uncertainty propagation using the gauss-bingham distribution," in *Proc. AAS/AIAA Space Flight Mech. Meeting*, 2015, pp. 15–347.

[11] J. Glover and L. P. Kaelbling, "Tracking 3-D rotations with the quaternion Bingham filter," *Comput. Sci. Artif. Intell. Lab.*, Cambridge, MA, USA, Tech. Rep. MIT-CSAIL-TR-2013-005, 2013.

[12] N. P. Santos, V. Lobo, and A. Bernardino, "3D model-based estimation for UAV tracking," in *Proc. OCEANS MTS/IEEE*, Charleston, SC, USA, Oct. 2018.

[13] B.-K. Seo, J. Park, and J.-I. Park, "3-D visual tracking for mobile augmented reality applications," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2011, pp. 1–4.

[14] P. Vicente, L. Jamone, and A. Bernardino, "Robotic hand pose estimation based on stereo vision and GPU-enabled internal graphical simulation," *J. Intell. Robot. Syst.*, vol. 83, nos. 3–4, pp. 339–358, Sep. 2016.

[15] J. Li-Chee-Ming and C. Armenakis, "A feasibility study on using ViSP'S 3D model-based tracker for UAV pose estimation in outdoor environments," in *Proc. Int. Archives Photogram., Remote Sens. Spatial Inf. Sci.*, vol. 40, no. 1, p. 329, 2015.

[16] Z. Cao, Y. Sheikh, and N. K. Banerjee, "Real-time scalable 6DOF pose estimation for textureless objects," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 2441–2448.

[17] C. Teuliere, L. Eck, E. Marchand, and N. Guenard, "3D model-based tracking for UAV position control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2010, pp. 1084–1089.

[18] N. P. Santos, F. Melicio, V. Lobo, and A. Bernardino, "A ground-based vision system for UAV pose estimation," *Int. J. Mechatronics Robot. (IJMR)*, vol. 1, no. 4, p. 7, 2014. [Online]. Available: <http://ojs.unsysdigital.com/index.php/ijrm/article/view/180>

[19] N. P. Santos, V. Lobo, and A. Bernardino, "Particle filtering based optimization applied to 3D model-based estimation for UAV pose estimation," in *Proc. OCEANS*, Aberdeen, U.K., Jun. 2017, pp. 1–10.

[20] E. J. Leffens, F. L. Markley, and M. D. Shuster, "Kalman filtering for spacecraft attitude estimation," *J. Guid., Control, Dyn.*, vol. 5, no. 5, pp. 417–429, Sep. 1982.

[21] F. L. Markley, N. Berman, and U. Shaked, "Deterministic EKF-like estimator for spacecraft attitude estimation," in *Proc. Amer. Control Conf. (ACC)*, Aug. 2005, pp. 247–251.

[22] E. A. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proc. IEEE Adapt. Syst. Signal Process., Commun., Control Symp.*, 2000, pp. 153–158.

[23] Y. Cheng and J. Crassidis, "Particle filtering for sequential spacecraft attitude estimation," in *Proc. AIAA Guid., Navigat., Control Conf. Exhibit*, Aug. 2004, p. 5337.

[24] Y. Xia and X.-J. Wu, "Adaptive ball particle filter and its application to visual tracking," *IETE Tech. Rev.*, vol. 32, no. 6, pp. 462–470, Nov. 2015.

[25] T. A. Myhre and O. Egeland, "Parameter estimation for visual tracking of a spherical pendulum with particle filter," in *Proc. IEEE Int. Conf. Multisensor Fusion Integr. Intell. Syst. (MFI)*, Sep. 2015, pp. 116–121.

[26] C. Chang and R. Ansari, "Kernel particle filter for visual tracking," *IEEE Signal Process. Lett.*, vol. 12, no. 3, pp. 242–245, Mar. 2005.

[27] C. Gonzales and S. Dubuisson, "Combinatorial resampling particle filter: An effective and efficient method for articulated object tracking," *Int. J. Comput. Vis.*, vol. 112, no. 3, pp. 255–284, May 2015.

[28] Y.-M. Chan, S.-S. Huang, L.-C. Fu, P.-Y. Hsiao, and M.-F. Lo, "Vehicle detection and tracking under various lighting conditions using a particle filter," *IET Intell. Transp. Syst.*, vol. 6, no. 1, p. 1, 2012.

[29] B. Sugandi, H. Kim, J. K. Tan, and S. Ishikawa, "Object tracking based on color information employing particle filter algorithm," in *Object Tracking*. Rijeka, Croatia: InTech, 2011.

[30] Y. Rui and Y. Chen, "Better proposal distributions: Object tracking using unscented particle filter," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Aug. 2005, p. 2.

[31] P. Li, T. Zhang, and A. E. C. Pece, "Visual contour tracking based on particle filters," *Image Vis. Comput.*, vol. 21, no. 1, pp. 111–123, Jan. 2003.

[32] R.-H. Guo and Z. Qin, "An unscented particle filter for ground maneuvering target tracking," *J. Zhejiang Univ.-Sci. A*, vol. 8, no. 10, pp. 1588–1595, Oct. 2007.

[33] W. Guo, C. Han, and M. Lei, "Improved unscented particle filter for nonlinear Bayesian estimation," in *Proc. 10th Int. Conf. Inf. Fusion*, 2007, pp. 1–6.

- [34] J. E. Darling and K. J. DeMars, "Uncertainty propagation of correlated quaternion and Euclidean states using partially-conditioned Gaussian mixtures," in *Proc. 19th Int. Conf. Inf. Fusion (FUSION)*, 2016, pp. 1805–1812.
- [35] C. Bingham, "An antipodally symmetric distribution on the sphere," *Ann. Statist.*, vol. 2, no. 6, pp. 1201–1225, Nov. 1974.
- [36] I. Gilitschenski, G. Kurz, S. J. Julier, and U. D. Hanebeck, "Efficient Bingham filtering based on saddlepoint approximations," in *Proc. Int. Conf. Multisensor Fusion Inf. Integr. Intell. Syst. (MFI)*, Sep. 2014, pp. 1–7.
- [37] J. Glover and L. P. Kaelbling, "Tracking the spin on a ping pong ball with the quaternion bingham filter," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 4133–4140.
- [38] G. Kurz, I. Gilitschenski, S. Julier, and U. D. Hanebeck, "Recursive bingham filter for directional estimation involving 180 degree symmetry," *J. Adv. Inf. Fusion*, vol. 9, no. 2, pp. 90–105, 2014.
- [39] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: <http://arxiv.org/abs/1804.02767>
- [40] A. J. Haug, *Bayesian Estimation Tracking: A Practical Guide*. Hoboken, NJ, USA: Wiley, 2012.
- [41] A. Doucet, N. De Freitas, K. Murphy, and S. Russell, "Rao-blackwellised particle filtering for dynamic Bayesian networks," in *Proc. 16th Conf. Uncertainty Artif. Intell.* San Mateo, CA, USA: Morgan Kaufmann, 2000, pp. 176–183.
- [42] N. J. Higham, "Analysis of the Cholesky decomposition of a semi-definite matrix," in *Reliable Numerical Computation*. Oxford, U.K.: Oxford Univ. Press, 1990, pp. 161–185.
- [43] S. S. Haykin, *Kalman Filtering Neural Networks*. Hoboken, NJ, USA: Wiley, 2001.
- [44] Y.-J. Cheon and J.-H. Kim, "Unscented filtering in a unit quaternion space for spacecraft attitude estimation," in *Proc. IEEE Int. Symp. Ind. Electron.*, Jun. 2007, pp. 66–71.
- [45] K. Mardia, *Characterizations of Directional Distributions*. Dordrecht, The Netherlands: Springer, 1975, pp. 365–385.
- [46] M. J. D. Powell, "Direct search algorithms for optimization calculations," *Acta Numerica*, vol. 7, pp. 287–336, Jan. 1998.
- [47] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, Apr. 1970.



NUNO PESSANHA SANTOS received the M.Sc. degree in naval military sciences in weapons and electronics engineering from the Portuguese Naval Academy, Alfeite, Portugal, in 2010, and the M.Sc. degree in electrical engineering (specialization in automation and industrial electronics) from the High Institute of Engineering of Lisbon, Lisbon, Portugal. He is currently pursuing the Ph.D. degree with the Instituto Superior Técnico (IST), Lisbon University, Portugal. He currently works as an Electrical Engineer Officer at the Portuguese Navy Ships Directorate and is an Associate Researcher at the Portuguese Navy Research Center (CINAV). His primary research interests focus on applications of computer vision, machine learning, motion estimation, and algorithm design applied to advanced robotics.



VICTOR LOBO received the Ph.D. degree in 2003. He was the Director of the Portuguese Navy Research Center (CINAV), from 2012 to 2019. He is currently a Full Professor with the Portuguese Naval Academy, Alfeite, Portugal, and an invited Full Professor with the Information Management School, Nova University, Lisbon, Portugal. He has participated in several national and international research projects as a Principal Investigator and Technical Manager. He has graduated four Ph.D. students and more than 60 M.Sc. students. He published more than 100 research papers in peer-reviewed journals and peer-reviewed conferences. His primary research interests focus on data mining, machine learning, and advanced robotics.



ALEXANDRE BERNARDINO received the Ph.D. degree in 2004. He is currently an Associate Professor with the Department of Electrical and Computer Engineering and a Senior Researcher with the Computer and Robot Vision Laboratory, Institute for Systems and Robotics, Instituto Superior Técnico (IST), and the faculty of engineering at Lisbon University. He has participated in several national and international research projects as a Principal Investigator and Technical Manager. He has graduated 12 Ph.D. students and more than 80 M.Sc. students. He published more than 40 research articles in peer-reviewed journals and more than 100 papers on peer-reviewed conferences in the field of robotics, vision, and cognitive systems. His primary research interests focus on the application of computer vision, machine learning, cognitive science, and control theory to advanced robotics and automation systems. He was a Co-Supervisor of the Ph.D. Thesis that received the IBM Prize 2014 and the Supervisor of the Best Robotics Portuguese M.Sc. Thesis Award of 2012. He is currently the Chair of the IEEE Portugal Robotics and Automation Chapter. He is an Associate Editor of the journal *Frontiers in Robotics and AI* and major robotics conferences (ICRA and IROS).

...