

Estimating the Linear and Angular Velocities of an Object in Free Flight

Matthew Gardner Yan-Bin Jia
Department of Computer Science
Iowa State University
Ames, IA 50011, USA
mattga, jia@iastate.edu

Abstract—In this paper, we investigate the problem of estimating the instantaneous position, orientation, velocity, and angular velocity of a rigid body during its free flight. The flight is under the influences of gravity, air drag, and the Magnus force. Two high-speed, stereo vision cameras are synchronized to take images of the flying object. Line features are extracted from each image as observables from a governing system which integrates quaternion-based dynamics with a camera model that deals with lens distortion. An extended Kalman filter (EKF) is employed to carry out the online estimation. Experimental validation using accelerometers and a stepper motor show agreement with estimated velocities.

I. INTRODUCTION

Real-time tracking of the orientation and position of a rigid body moving in the space is an important and challenging problem with applications in space exploration, air traffic control, missile guidance, underwater vehicles, robotics, virtual reality, etc. Techniques for estimating its velocity and angular velocity have relied on either accelerometers or cameras mounted inside the body, or cameras stationed outside.

The first estimation approach measured the accelerations at the mounting points of several linear accelerometers along orthogonal axes, and then either solved a system of kinematic equations for the body's angular acceleration [1], or determined it through optimization [2], [3]. In [4], a magnetometer, accelerometer, and angular rate sensor were employed to obtain measurements from which a quaternion was calculated and supplied to a Kalman filter for polishing. Nevertheless, none of the aforementioned methods addressed how to estimate the object's linear velocity.

Support for this research has been provided by the US National Science Foundation under Grant IIS-1421034. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. The authors would also like to thank Xiaoqian Mu for her help with experiments.

The work [5] showed that the rotation motion of an asteroid about a fixed axis could be recovered on a spaceship from images taken of outside landmarks by combining particle filtering with optimization. The angular velocity, if constant, could be estimated based on line correspondences among the images by combining a least-squares method with interpolation [6].

Vision-based estimation often integrated dual quaternions into an extended Kalman filter (EKF), taking linear features from images generated by either a single camera [7], or a network of cameras followed by optimization to achieve consensus [8]. Since rigid body dynamics were not fully integrated into the system, simulation and experiments were conducted for estimating only constant velocities and angular velocities.

Our work aims to accurately estimate an object's pose and motion during its flight. Accuracy can be difficult to obtain when working with limited camera hardware, wide-angle lenses, and severe effects of aerodynamics. Previous work has not seriously investigated vision-based estimation of angular information under the "full" effects of rigid body dynamics. Since highly accurate state estimation is not required in many robotics tasks, forces of drag and Magnus, or camera lens distortion, are often ignored. Furthermore, past efforts have lacked experiments with low-cost vision systems.

We are interested in tracking the full state of a rigid body in free flight, which includes its position, rotation, velocity, and angular velocity. An EKF is used with system dynamics involving forces of gravity, drag, and Magnus, and a camera model accounting for radial and tangential distortions. Observables are provided to the EKF using line features extracted by two cameras. The cameras are configured in stereo vision with additional work on their calibration.

Section II introduces the dynamics model of flight in a quaternion-based formulation. Section III briefly discusses the camera projection model, the geometry

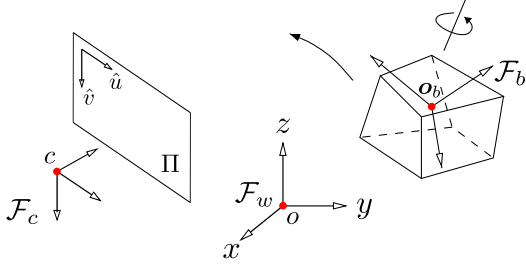


Fig. 1. Flying object and image plane.

of two views of a scene, and a calibration method for estimating all camera parameters. Section IV formulates the EKF and the measurement model. Section V shows results from simulation of various objects, as well as experiments using accelerometers and a stepper motor for validation of a wooden frame and cuboid object, respectively. Section VI discusses of work in the near future.

II. SYSTEM DYNAMICS

Consider a flying object as shown in Figure 1. The object, with known geometry and physical properties, has a body frame \mathcal{F}_b located at its center of mass \mathbf{o}_b , and defined by its principal axes. Under the frame, the angular inertia matrix Q is diagonalized. The frame \mathcal{F}_b has a translation \mathbf{b} from the world frame \mathcal{F}_w . The rotation of \mathcal{F}_b from \mathcal{F}_w is described by a unit quaternion [9]:

$$\mathbf{r} = (r_0 \mathbf{r}^T)^T = (r_0, r_1, r_2, r_3)^T.$$

Let \mathbf{v} be the object's velocity in \mathcal{F}_w , and $\boldsymbol{\omega}$ be its angular velocity in terms of \mathcal{F}_b , that is, relative to a fixed frame instantaneously coinciding with \mathcal{F}_b . The angular velocity is $r\boldsymbol{\omega}r^*$ when expressed in \mathcal{F}_w , where $r^* = (r, -r_i^T)^T$ is the conjugate of r .

In addition to the gravitational force, the object is subject to drag and Magnus forces. Based on the works [10], [11], [12], [14], we approximate the two forces by their coefficients C_d and C_m , which are dependent on the object's shape relative to the flow of air. Let ρ be the air density and A the area of the object's cross section normal to \mathbf{v} . Introduce

$$e_d = \frac{1}{2}\rho AC_d \quad \text{and} \quad e_m = \frac{1}{2}\rho AC_m.$$

Air drag acts opposite to the object's velocity, producing the force

$$\mathbf{f}_d = -e_d\|\mathbf{v}\|\mathbf{v}.$$

Magnus force [13, pp. 16-33] is due to uneven air pressure created by the air flow passing the object's

top and bottom parts as it rotates in the air. This force, traverse to the air flow, is given as

$$\mathbf{f}_m = e_m(r\boldsymbol{\omega}r^*) \times \mathbf{v}.$$

Newton's equation now takes the form:

$$\dot{\mathbf{v}} = \mathbf{g} - e_d\|\mathbf{v}\|\mathbf{v} + e_m\boldsymbol{\omega} \times \mathbf{v}, \quad (1)$$

where \mathbf{g} is the gravitational acceleration vector. It is reasonable to assume that the forces \mathbf{f}_d and \mathbf{f}_m act through the object's center of mass, just like the gravitational force. Then Euler's equation assumes the form $Q\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times Q\boldsymbol{\omega} = 0$, from which we immediately obtain

$$\dot{\boldsymbol{\omega}} = -Q^{-1}(\boldsymbol{\omega} \times Q\boldsymbol{\omega}). \quad (2)$$

The 13-vector $\mathbf{s} = (\mathbf{b}^T, r^T, \mathbf{v}^T, \boldsymbol{\omega}^T)^T$ describes the state of the flying object. The quaternion r describing the object's orientation has the following derivative given in Appendix C of [9]:

$$\dot{r} = \frac{1}{2}r\boldsymbol{\omega}. \quad (3)$$

This, together with $\dot{\mathbf{b}} = \mathbf{v}$, (1), (2), forms a system of nonlinear differential equations below:

$$\frac{d\mathbf{s}}{dt} = \begin{pmatrix} \mathbf{v} \\ \frac{1}{2}r\boldsymbol{\omega} \\ \mathbf{g} - e_d\|\mathbf{v}\|\mathbf{v} + e_m\boldsymbol{\omega} \times \mathbf{v} \\ -Q^{-1}(\boldsymbol{\omega} \times Q\boldsymbol{\omega}) \end{pmatrix} = \mathbf{a}(\mathbf{s}). \quad (4)$$

III. GEMOETRY OF PROJECTION

The object's state is estimated based on measurements extracted from its images taken simultaneously by two cameras. We first focus on one camera to understand the measurement model. Fig. 1 shows the camera's focal point located at \mathbf{c} , where a frame \mathcal{F}_c is set up with its z -axis perpendicular to the image plane. The rotation of \mathcal{F}_c from the world frame \mathcal{F}_w is described by a quaternion r_c . The image plane Π has a local coordinate system with the origin at the upper left corner of the image, the u -axis pointing rightward, and the v -axis pointing downward.

A. Imaging Model

The projection model used is outlined in detail in [15, pp. 16–17]. Let \mathbf{p} be a point on the object, and denote ${}^b\mathbf{p}$, ${}^w\mathbf{p}$ and ${}^c\mathbf{p}$ as its coordinates in the body, world, and camera frames, respectively. They assume the following mappings:

$${}^w\mathbf{p} = \mathbf{b} + r({}^b\mathbf{p})r^*, \quad (5)$$

$${}^c\mathbf{p} = r_c^*({}^w\mathbf{p} - \mathbf{c})r_c. \quad (6)$$

Letting ${}^c\mathbf{p} = ({}^c p_x, {}^c p_y, {}^c p_z)^T$, the pinhole camera model yields the normalized image coordinates

$${}^p\mathbf{p} = \frac{1}{{}^c p_z} \begin{pmatrix} {}^c p_x \\ {}^c p_y \end{pmatrix}. \quad (7)$$

Brown's distortion model [16] is then used to map the undistorted point ${}^p\mathbf{p}$ to a distorted point ${}^p\tilde{\mathbf{p}}$

$${}^p\tilde{\mathbf{p}} = \delta_r({}^p\mathbf{p}) + \delta_t({}^p\mathbf{p}), \quad (8)$$

where the function δ_r applies radial distortion and δ_t applies tangential distortion. Letting ${}^p\mathbf{p} = ({}^p p_x, {}^p p_y)^T$ and $\rho = \|{}^p\mathbf{p}\|$, the distortion functions are defined as

$$\delta_r({}^p\mathbf{p}) = (1 + k_1\rho^2 + k_2\rho^4 + k_3\rho^6) {}^p\mathbf{p}, \quad (9)$$

$$\delta_t({}^p\mathbf{p}) = \begin{pmatrix} 2h_1({}^p p_x {}^p p_y) + h_2(\rho^2 + 2{}^p p_x^2) \\ h_1(\rho^2 + 2{}^p p_y^2) + 2h_2({}^p p_x {}^p p_y) \end{pmatrix}, \quad (10)$$

for some coefficients k_1, k_2, k_3 of radial distortion, and h_1, h_2 of tangential distortion.

To convert the point to image coordinates, the following intrinsic parameters are needed: scaling parameters α and β , the skew angle θ , and the image center $(u_c, v_c)^T$. The distorted point is then transformed into image coordinates by

$${}^i\mathbf{p} = \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \alpha & -\alpha \cot \theta \\ 0 & \beta / \sin \theta \end{pmatrix} {}^p\tilde{\mathbf{p}} + \begin{pmatrix} u_c \\ v_c \end{pmatrix}. \quad (11)$$

Note that ${}^b\mathbf{p}, {}^w\mathbf{p}$ and ${}^c\mathbf{p}$ are 3-vectors with ${}^b\mathbf{p}$ being determined beforehand (as the position of a marker), while ${}^p\mathbf{p}, {}^p\tilde{\mathbf{p}}$, and ${}^i\mathbf{p}$ are 2-vectors. The sequence of transformations is best summarized as follows:

$${}^b\mathbf{p} \xrightarrow{(5)} {}^w\mathbf{p} \xrightarrow{(6)} {}^c\mathbf{p} \xrightarrow{(7)} {}^p\mathbf{p} \xrightarrow{(8)} {}^p\tilde{\mathbf{p}} \xrightarrow{(11)} {}^i\mathbf{p}. \quad (12)$$

B. Two-view Camera Geometry

The two cameras are placed in a stereo vision configuration to produce different views of a scene. From now on, the single quote will be used to refer to the second camera. The geometry between these views can be described by intersecting lines formed from light rays. These rays originate at a point ${}^w\mathbf{p}$ in the world frame, and pass through the camera centers \mathbf{c} and \mathbf{c}' where they are projected onto the image plane at \mathbf{q} and \mathbf{q}' , respectively¹. These five points form a plane defined by

$$(\mathbf{q} - \mathbf{c}) \cdot [(\mathbf{c}' - \mathbf{c}) \times (\mathbf{q}' - \mathbf{c}')] = 0. \quad (13)$$

The points \mathbf{q} and \mathbf{q}' relative to their respective camera frames represent the perspective projection of ${}^w\mathbf{p}$ up to a scale, and thus can be written as ${}^p\mathbf{p}$ and ${}^p\mathbf{p}'$, respectively. Next, let ${}^c\mathbf{t} = r_c^*(\mathbf{c}' - \mathbf{c})r_c$ be the translation vector in the first camera frame. Denote by ${}^c\mathbf{t} \times$ the skew-symmetric

¹All points have coordinates in the world frame.

matrix that acts through left multiplication as taking a cross product with ${}^c\mathbf{t}$ as the left operand. Let S be the rotation matrix corresponding with the quaternion product $r_c^* r_c'$, where r_c and r_c' describe the rotations of the two camera frames from the world frame. Equation (13) is then rewritten relative to the first camera frame:

$$({}^p\mathbf{p}'^T, 1) \mathcal{E} ({}^p\mathbf{p}^T, 1)^T = 0 \quad (14)$$

where $\mathcal{E} = ({}^c\mathbf{t} \times)S$ [15]. The 3×3 matrix \mathcal{E} , referred to as the essential matrix, is also skew-symmetric and has rank two with two equal non-zero singular values [17], [18].

Since equation (14) is defined up to scale, the vector ${}^c\mathbf{t}$ can be made to have unit length after introducing a scalar value d to represent the distance between the two cameras. The position of the second camera is then written as

$$\mathbf{c}' = \mathbf{c} + d r_c ({}^c\mathbf{t}) r_c^*. \quad (15)$$

Note that the constraint (14) ignores lens distortion which causes rays of light to refract off the camera lens, producing bent lines that are not coplanar. To accommodate this, image points are first back-projected and undistorted into the ideal pinhole model by solving the system of (8) and (11) for ${}^p\mathbf{p}$ given ${}^i\mathbf{p}$. This is done numerically using the Levenberg-Marquardt algorithm for nonlinear least-squares minimization [19], [20].

C. Calibration

We now present a calibration procedure to accurately estimate parameters of the camera model subject to the two-view geometric constraint. A set of l points ${}^w\mathbf{q}_1, \dots, {}^w\mathbf{q}_l$ are chosen in world coordinates. We obtain their corresponding image coordinates ${}^i\mathbf{q}_1, \dots, {}^i\mathbf{q}_l$ from the first camera, and ${}^i\mathbf{q}'_1, \dots, {}^i\mathbf{q}'_l$ from the second camera. This data is then used to calibrate the cameras together through two steps of minimization where an initial calibration is needed. Single camera calibration methods such as the one by Zhang [21] can provide a good initial estimate.

Introduce the vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}'$ for the two cameras, respectively, each containing ten intrinsic parameters: $u_c, v_c, \alpha, \beta, \theta, k_1, k_2, k_3, h_1, h_2$. The first step is to calculate \mathcal{E} via nonlinear optimization as presented by Hartley and Zisserman [22, pp. 283–284]. First, using the current values of $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}'$, undistort ${}^i\mathbf{q}_j$ and ${}^i\mathbf{q}'_j$ by solving (11) and (8) sequentially to obtain ${}^p\mathbf{q}_j$ and ${}^p\mathbf{q}'_j$, $j = 1 \dots l$. Then, solve the linear system (14) for the essential matrix \mathcal{E} , which has rank 2. Extract from \mathcal{E} the matrix S and unit vector ${}^c\mathbf{t}$ [22, pp. 258–259].

In the second step, for each image point ${}^i\mathbf{q}$, we minimize the *reprojection error* of the measured world

coordinate ${}^w\mathbf{q}$ passed through the camera model summarized by (12). Denote the reprojected points ${}^i\mathbf{p}$ and ${}^i\mathbf{p}'$ as functions of ${}^w\mathbf{q}_j$ by concatenating (6), (7), (8), and (11). Minimization is performed in a least-squares manner with an objective function that combines the reprojection error of both cameras:

$$g(S, {}^c\mathbf{t}; \mathbf{x}) = \sum_{j=1}^l \|\mathbf{q}_j - {}^i\mathbf{p}\|^2 + \sum_{j=1}^l \|\mathbf{q}'_j - {}^i\mathbf{p}'\|^2. \quad (16)$$

The function is minimized over the parameter vector $\mathbf{x} = (\boldsymbol{\lambda}^T, \boldsymbol{\lambda}'^T, r_c, \mathbf{c}, d)^T$ using the Levenberg-Marquardt algorithm. Note that the rotation r'_c and translation \mathbf{c}' of the second camera are omitted as they are calculated respectively from S along with r_c , and from equation (15) along with the parameter d .

Given an initial calibration, we perform the aforementioned steps repeatedly until convergence of the reprojection errors.

IV. STATE ESTIMATION

We use an extended Kalman filter (EKF) [23, p. 405] to observe the state $\mathbf{s} = (\mathbf{b}^T, r^T, \boldsymbol{\omega}^T, \mathbf{v}^T)^T$ of the flying object. The EKF evaluates the state's system dynamics from Section II in continuous-time, while obtaining observables from the cameras at discrete time instants. Denote the function $\mathbf{h}(\mathbf{s})$ as the measurement model to predict observables from the state estimate. The algorithm makes use of the Jacobians of \mathbf{a} and \mathbf{h} :

$$A = \frac{\partial \mathbf{a}}{\partial \mathbf{s}} \quad \text{and} \quad H = \frac{\partial \mathbf{h}}{\partial \mathbf{s}}, \quad (17)$$

which can be derived from equation (4) and the chain of equations summarized by (12), respectively.

The measurement model consists of observables computed from lines along the object's edges. Line features are used as opposed to point features since they are more reliable for pose estimation [7]. Consider three incident edges sharing a vertex on the object.

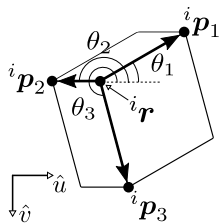


Fig. 2. Edge triple for a cuboid. Each vector forms an angle $\theta \in [0, 2\pi)$ with the x -axis.

We obtain observables from their projections in the two-dimensional image, referred to as the *edge triple*. Fig. 2 shows the image of a cuboid object and its edge triple. Let ${}^i\mathbf{p}_j$, $j = 1, 2, 3$ be the vertices opposite to a shared vertex ${}^i\mathbf{r}$, referred to as the *reference vertex*. Ordering of vertices as well as which vertex becomes the reference can be determined based on which faces are visible. With

known positions ${}^b\mathbf{p}$ in the body frame, the vertices ${}^i\mathbf{p}_j$ and reference vertex ${}^i\mathbf{r}$ are computed from the chain of equations in (12). Each axis is then identified by a line direction

$${}^i\hat{\mathbf{l}}_j = (u_j, v_j)^T = \frac{{}^i\mathbf{p}_j - {}^i\mathbf{r}}{\|{}^i\mathbf{p}_j - {}^i\mathbf{r}\|},$$

for $j = 1, 2, 3$. The phase angles $\theta_j = \text{atan2}(v_j, u_j)$ of these vectors relative to the u -axis are used as observables. The three angles combined with the reference vertex ${}^i\mathbf{r}$ make up the observables of one camera. The measurement vector is then written as

$$\mathbf{y} = ({}^i\mathbf{r}^T, \theta_1, \theta_2, \theta_3, {}^i\mathbf{r}'^T, \theta'_1, \theta'_2, \theta'_3)^T = \mathbf{h}(\mathbf{s}, \boldsymbol{\epsilon}). \quad (18)$$

The measurement error $\boldsymbol{\epsilon}$ follows a normal distribution with zero mean and a covariance matrix R , i.e., $\boldsymbol{\epsilon} \sim (0, R)$.

V. SIMULATION AND EXPERIMENT

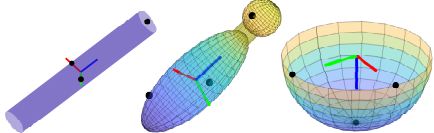
A. Simulation

Simulation is performed to demonstrate accuracy and convergence of the EKF for three objects varying in shape and complexity: a solid rod (radius 3 cm and height 30 cm), bowling pin², and bowl (hollow hemisphere with a radius of 10 cm). All objects have a mass of 0.1 kg.

The drag coefficient C_d is approximated from studies on the aerodynamics of general shapes [24], [25]. Values of C_d and the cross sectional area A are calculated as functions of the velocity direction at each time instant. In the case of the bowl, when air flows against the open side, $C_d = 1.42$, and against the rounded side, $C_d = 0.38$. For orientations between these two, the coefficients are linearly interpolated. Due to a lack of studies on values of C_m for arbitrary objects, it is set to 0.15 for all objects. This was a median value found through experiments to be reasonable for various shapes thrown by hand with a magnitude of angular velocity in the range of 5–25 rad/s.

The EKF is initialized with no information, such that position and velocity estimates started at zero. The state covariance matrix is initialized with diagonal components of 0.01 for positions, 0.005 for rotations, and 0.1 for both linear and angular velocities. The initial true state has a velocity of $(-3.5, 1.5, 3)^T$, and angular velocity of $(-2, -2, -4)^T$. It was propagated by the system dynamics for 200 iterations and with a time interval of 5 ms, totaling 1 s of flight over 4.5 m.

²Composed of an ellipsoidal bottom with radii of 3 cm about the semi-minor axis and 10 cm about the semi-major axis, a cylindrical neck of radius 1.4 cm and height of 2.25 cm, and a spherical top of radius 2.5 cm.



Variable	Errors ($\times 10^{-2}$)			Unit
	Rod	Pin	Bowl	
$\mathbf{b} = \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix}$	-0.16 0.12 0.00	-0.53 0.07 -0.16	-0.24 0.11 -0.03	m
$\mathbf{r} = \begin{pmatrix} r_w \\ r_x \\ r_y \\ r_z \end{pmatrix}$	-0.37 -0.22 0.36 -0.54	-0.99 1.46 0.68 6.63	-0.13 0.80 -0.15 -0.52	-
$\mathbf{v} = \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix}$	0.32 1.07 0.12	0.11 1.65 -0.20	0.27 1.05 0.20	m/s
$\boldsymbol{\omega} = \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix}$	-0.72 -5.13 -0.42	-11.3 23.4 -25.6	-1.48 5.45 -2.62	rad/s

TABLE I
ESTIMATION ERRORS FROM SIMULATION

Table I gives the mean errors for position and velocity estimates from 20 flight instances of each object. Observables are calculated from three fixed body coordinates marked as black dots on the images in the first row. The points are projected onto the image and noise is added according to the covariance matrix R with its diagonal elements set to 2.25 pixels for each component of ${}^i\mathbf{r}$, and 0.001 rad for θ_j , $j = 1, 2, 3$ ³. In the case of the pin, these points form an edge triple that is elongated, spanning primarily the z -axis. Despite this, small error is still obtained, where it is noticeably higher in the angular velocity estimates.

B. Experiment

Experiments were performed to demonstrate accuracy and robustness of the estimation algorithm using three methods of validation. Two Ximea MQ022CG-CM high-speed, color cameras were used to capture images simultaneously. Both were equipped with Navitar NMV-6 lenses with a 81.9 degree field of view. The cameras were carefully calibrated using the procedure from Section III-C with data points obtained by images of a 3D calibration cube. An object was then thrown across the camera's view approximately 2.3 m away. The initial estimate of the object's pose was computed by undistorting image points, triangulating the world coordinates, and using Horn's method[26] to estimate orientation.

³Producing measurement error of about ± 4.5 pixels and ± 0.01 rad.

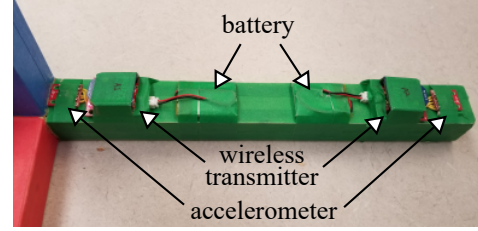


Fig. 3. Single axis of a wooden frame object with accelerometers attached. The y -axis shown (colored green) has one accelerometer at the end of the axis and another near the intersection of the axes. The x (red) and z (blue) axes have one accelerometer each at the ends of their axes. Each axis is 30 cm long and 2.9 cm wide. The object has a mass of 0.31 kg, drag coefficient 1.05, and Magnus coefficient 0.15.

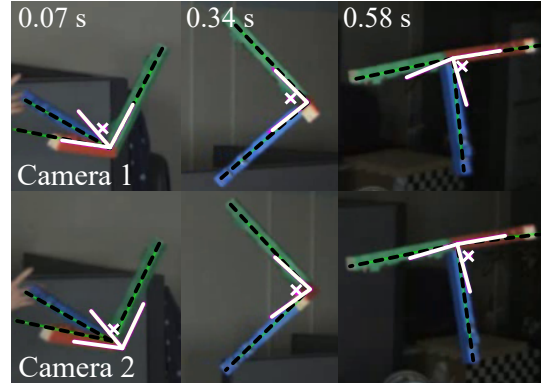


Fig. 4. Cropped images of the wooden frame in flight. Observables from vision are drawn as dashed black lines and predictions from the state estimate as white lines. The object's position is plotted as a white 'x'. Labels at the top indicate the time since the start of estimation.

In the first experiment, a wooden frame object was thrown by hand. Linear accelerations were measured by four tri-axis ADXL335 accelerometers from Analog Devices, capable of measuring ± 29.4 m/s². The data were transmitted wirelessly using four Digi XBee Series 1 modules. Fig. 3 shows the configuration of one axis of the frame. Before the object was thrown, the accelerometers were calibrated by repeatedly measuring voltages corresponding to the force of gravity while laying flat on a table. The object's inertia tensor was also adjusted to account for the accelerometers as point masses. Fig. 4 shows images of the wooden frame in flight. The object was estimated for 0.69 s at a rate of 230 frames per second (fps) for a distance of 2.27 m within the image.

The object's velocities were then calculated in the body frame. The readings \mathbf{a}_i from the i th accelerometer satisfy the kinematics equation $\mathbf{a}_i = \dot{\mathbf{v}} + \dot{\boldsymbol{\omega}} \times \mathbf{r}_i + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}_i)$, where \mathbf{r}_i locates the accelerometer in the body frame. Using four accelerometers, we obtain twelve quadratic equations in total with nine unknowns

in \dot{v} , ω , and $\dot{\omega}$. The equations can instead be written as a linear system with twelve (dependent) unknowns: three for \dot{v} , and nine for a 3×3 matrix W composed from $\dot{\omega}$ and ω . Formulation of the linear system and extraction of ω from W is discussed in [3]. Last, the linear velocity v was obtained by integration over \dot{v} .

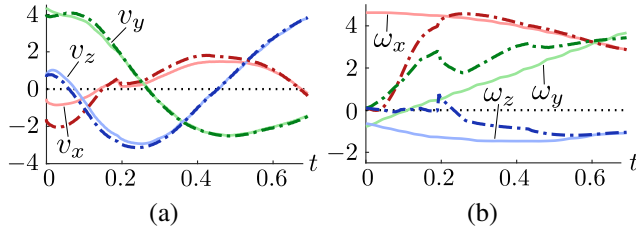


Fig. 5. Plots of the estimated and observed (a) linear and (b) angular velocities. Velocity readings from accelerometers in the x , y , and z directions are plotted as solid lines in light red, green, and blue, respectively. The estimated velocities are plotted as dashed lines in a darker red, green, and blue.

Fig. 5 plots the measured and estimated values of velocity and angular velocity. The final estimation errors at 0.69 s were $(0.036, 0.089, 0.085)^T$ for velocity, and $(0.011, 0.230, -0.032)^T$ for angular velocity. In addition, another EKF was formulated without drag and Magnus forces and used to estimate the same flight. The estimation errors for this EKF were $(0.076, 0.098, 0.137)^T$ for velocity, and $(0.013, 0.223, -0.031)^T$ for angular velocity. Note that in Fig. 4, only two axes are detected at 0.34 s due to occlusion of the third. Regardless, the EKF continues to converge from 0.25 s to 0.45 s while observables are missing, although some errors in ω_y and ω_z can be seen to accumulate.

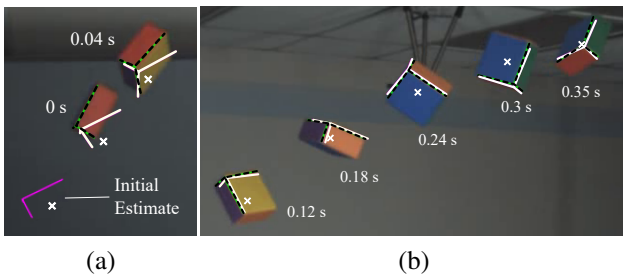


Fig. 6. Images from the first camera of the flying cuboid. The object from different images are superposed at the corresponding location in a single image. (a) shows the estimate early on and (b) shows after a few iterations of the EKF.

Next, a hollow, plastic cuboid was thrown and images were captured and processed at a rate of 120 fps. The cuboid had each face colored differently to allow for identifying which faces, edges, and vertices were visible in each image. Fig. 6 shows two images containing

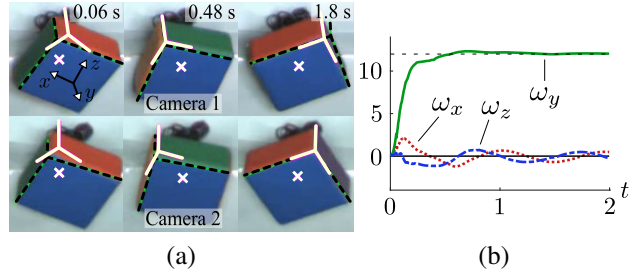


Fig. 7. (a) Images of a hollow cuboid mounted on a stepper motor spinning at 12 rad/s. The cuboid has sides of lengths 3.1 and 6.4 cm, mass 0.103 kg, drag coefficient 0.9, and Magnus coefficient 0.15. (b) Angular velocity estimates, where the dashed black line marks 12 rad/s.

poses of the object at different time instants. The initial estimate and early iterations are shown in (a), and later iterations in (b). The edge triple is drawn in black, and its position predicted from the pose estimate is drawn in white.

Last, the same cuboid object was fixed to a stepper motor along its y -axis, and set to spin at a rate of 12 rad/s. The object was imaged for 2 s at a rate of 500 fps, and estimated by an EKF with the dynamics of linear velocity removed. Fig 7(a) shows images of the cuboid rotating. For 182 out of the 1000 images, observables were not visible, but estimation continued undeterred. The estimated angular velocities are plotted in Fig 7(b). The estimate reached 12 rad/s about the y -axis at 0.5 s with errors of $(-0.99, -0.04, -0.73)^T$, and converged during the remaining 1.5 s with a mean error of $(-0.04, 0.07, -0.11)^T$.

VI. DISCUSSION

Accurate, high-speed motion estimation of an object flying and tumbling in space can be a daunting task for two fixed cameras. Modeling of flight dynamics can be inaccurate due to air drag and Magnus effects. Cameras can have significant nonlinear distortion that effects accuracy of features extracted from the object. We have presented an approach that demonstrates high accuracy of both linear and angular position and velocity by formulating an EKF with a complete model of system dynamics and camera projection.

Additional effort will be spent on validation, including experiments with objects that experience larger forces of drag and Magnus (e.g. a bowl or soccer ball) in the near future. Also, further estimation of drag and Magnus coefficients awaits investigation. Last, obtaining more calibration data may be necessary to ensure that camera parameters are well estimated for all subspaces containing the object's trajectory.

REFERENCES

- [1] A. J. Padgaonkar, K. W. Krieger, and A. I. King. Measurement of angular acceleration of a rigid body using linear accelerometers. *ASME J. Appl. Mech.*, vol. 42, no. 3, pp. 552–556, 1975.
- [2] P. Cardou and J. Angeles. Angular velocity estimation from the angular acceleration matrix. *ASME J. Appl. Mech.*, vol. 75, no. 2, pp. 021003-1, 2008.
- [3] P. Cardou, G. Fournier, and P. Gagnon. A nonlinear program for angular-velocity estimation from centripetal accelerations measurements. *IEEE/ASME Trans. Mechatronics*, vol. 16, no. 5, pp. 932–944, 2011.
- [4] J. L. Marins, X. Yun, E. R. Bachmann, R. B. McGhee, and M. J. Zyda. An extended Kalman filter for quaternion-based orientation estimation using MARG sensors. in *Proc. IEEE/RSJ Int. Conf. on Intell. Robots Syst.*, 2011, pp. 2003–2011.
- [5] N. Takeishi, T. Yairi, Y. Tsuda, F. Terui, N. Ogawa, and Y. Mimasu. Simultaneous estimation of shape and motion of an asteroid for automatic navigation. in *Proc IEEE Int. Conf. Robot. Autom.*, 2015.
- [6] M. Li and B. K. Ghosh. Estimation of angular velocity of a moving object using line correspondences. In *Proc. Am. Control Conf.*, 1993, pp. 746–750.
- [7] J. S. Goddard and M. A. Abidi. Pose and motion estimation using dual quaternion-based extended Kalman filtering. in *Proc. SPIE 4298*, vol. 3313, pp. 189–200, 1998.
- [8] Y. Zu, C. Sun, and R. Dai. Distributed estimation for spatial rigid motion based on dual quaternions. in *Proc. IEEE Conf. Decis. Control*, 2014, pp. 334–339.
- [9] Y.-B. Jia. Quaternion and rotation. <http://web.cs.ias.tate.edu/~cs577/quaternion.pdf>.
- [10] L. J. Briggs. Effect of spin and speed on the lateral deflection (curve) of a baseball; and the magnus effect for smooth spheres. *Am. J. Physics*, vol. 27, no. 8, pp. 589–596, 1959.
- [11] R. G. Watts and R. Ferrer. The lateral force on a spinning sphere: Aerodynamics of a curve ball. *Am. J. Phys.*, vol. 55, no. 1, pp. 40–44, 1987.
- [12] G. S. Sawicki, M. Hubbard, and W. Stronge. Reply to ‘comment on how to hit home runs: Optimum baseball bat swing parameter for maximum range trajectories’. *Am. J. Phys.*, vol. 73, pp. 185–189, 2005.
- [13] L. J. Clancy. *Aerodynamics*. Pitman Publishing Ltd., 1975.
- [14] A. M. Nathan. The effect of spin on the flight of a baseball. *Am. J. Phys.*, vol. 76, no. 2, pp. 119–124, 2008.
- [15] D. Forsyth and J. Ponce. *Computer vision: A Modern Approach*. Prentice-Hall, 2001.
- [16] D. C. Brown. Close-range camera calibration. *Photogrammetric Engineering*, vol. 37, no. 8, pp. 855–866, 1991.
- [17] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. In *Nature*, vol. 293, no. 5828, pp. 133–135, 1981.
- [18] T. S. Huang and O. D. Faugeras. Some Properties of the E Matrix in Two-View Motion Estimation. *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 12, pp. 1310–1312, 1989.
- [19] K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, vol. 2, no. 2, pp. 164–168, 1944.
- [20] D. W. Marquardt. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *SIAM J. Appl. Math.*, vol. 11, no. 2, pp. 431–441, 1963.
- [21] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, 2000.
- [22] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.
- [23] Dan Simon. *Optimal State Estimation*. John Wiley & Sons, Inc., 2006.
- [24] S. Hoerner. Fluid-Dynamic Drag, 1965. *Bricktown, NJ: Published by the author*, 1992.
- [25] Actions, Structural Design. Part 2: Wind Actions. *AS/NZS 1170*, 2002.
- [26] Berthold K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of Optical Society of America A*, vol. 4, no. 4, pp. 629–642, 1987.