Visual Pose Estimation for Shipboard Landing of Autonomous Parafoils

Charles W. Hewgley
Department of Electrical and Computer Engineering
United States Naval Academy
Annapolis, USA
hewgley@usna.edu

Roberto Cristi and Oleg A. Yakimenko
Department of Electrical and Computer Engineering
Department of Systems Engineering
Naval Postgraduate School
Monterey, USA
{rcristi,oayakime}@nps.edu

***Abstract*—This paper outlines an investigation into the use of a simple, focal-plane imaging sensor for guidance of an autonomous parafoil system for approach and landing on a moving platform such as a ship underway. The perspective-projective transformation between an object in a three-dimensional world and an image on a two-dimensional plane is analyzed and then formulated using a homogeneous coordinate system. The estimation problem is addressed; specifically, the challenge of dealing with the out-of-frame condition due to parafoil oscillation as it approaches the target. A dual-mode Kalman estimation-scheme is proposed that suspends measurement when the target is out-of-frame, and incorporates a two-view measurement when the target reenters the frame.**

***Keywords*—vision, small UAV, robot**

I. Introduction

Autonomous aerial delivery systems (ADSs) are autonomous vehicles that are deployed from an airborne carrier vehicle and that descend to the earth under a steerable round parachute or rectangular parafoil canopy. These systems are used by the U.S. Army and U.S. Air Force most often for resupply of ground forces. In this way, ADSs represent the modern evolution of aerial delivery, or *airdrop* techniques that came into widespread use during the Second World War.

Modern ADSs have achieved a high degree of accuracy in guiding themselves toward a fixed target on land. In recent conflicts in Iraq and Afghanistan, the U.S. Air Force and U.S. Army have relied on aerial delivery to resupply forward operating bases that are widely dispersed about the region of operations and are often located in rough terrain. A recent survey of research and development efforts, within the U.S. Department of Defense and also within NATO, indicates very active experimentation on aerial delivery systems ranging in weight capacity from 10 lbs up to 42 000 lbs. [1]

The demonstrated utility of precision aerial delivery systems to ground forces has spurred a quickening pace of this field of research; however, the use of precision aerial delivery as a technique for providing swift and flexible logistics delivery to ships underway is an untapped area of investigation. The possible uses of precision aerial delivery in the maritime domain include not only *vertical replenishment*, or resupply of vessels underway, as proposed in previous work by Hewgley and Yakimenko [2], [3], but also as a means for sensor emplacement, both on the ocean, and aboard vessels underway, and as a means to facilitate rapid movement of stores from ship to shore.

Fig. 1. Snowflake prototype ADS. The Snowflake prototype ADS glides toward landing at Camp Roberts, California.

The goal of the research described in this paper is that of enabling an autonomous ADS to land on a moving platform, such as the landing deck of a ship at sea. To accomplish this feat, the ADS must first estimate both the position and velocity of the target, and then plan a trajectory to be followed follow down to the target. Described herein is the development of a novel, dual-rate state-space estimation scheme that enables a moving observer to use measurements from a monocular visual sensor to estimate the position and velocity of a target even when the target is intermittently out of view. A development team consisting of members from Naval Postgraduate School and University of Alabama in Huntsville has built a small prototype ADS called *Snowflake* and conducted a series of flight tests starting in 2008 and continuing to the present day [4]. During some of these tests, the team gathered data to support visual estimation algorithm development.

II. Current Autonomous Shipboard Landing Capability

The idea of autonomous shipboard landing has, until now, been associated only with powered unmanned aircraft systems (UASs). Indeed, autonomous landing and takeoff of rotorcraft

unmanned aircraft system (RUAS) is a current capability possessed by the U.S. Navy, and also available commercially. The U.S. Navy's MQ-8B Fire Scout vertical take-off and landing tactical unmanned aerial vehicle (VTUAV) is designed to take off and land autonomously from certain air-capable ships using a system known as UAV Common Automatic Recovery System (UCARS).

Regarding UCARS, landing guidance is achieved using a dedicated microwave communication link between a sensor package installed on the landing platform and the aircraft. Other instrumentation in development for autonomous landing includes sophisticated carrier-phase, shipboard-relative Global Positioning System (GPS) receivers, and active light detection and ranging (LIDAR) sensors on the aircraft. Visual sensors with computer vision algorithms are lighter, cheaper, consume less power than these alternatives. For a one-way autonomous vehicle such as an ADS, minimizing the expense and weight of the required sensors is of prime importance.

Before proceeding further, the question should be addressed of why monocular vision (one camera) is used and not binocular, or stereo, vision (two cameras). For the application of an ADS attempting to land on a moving target on the ocean's surface, it was presumed that cheaper and more expendable sensors would be preferable to more complex and expensive systems. Binocular, or stereo vision systems, while declining in price, still require baseline calibration between the two visual sensors and more sophisticated video processing than do monocular systems. For these reasons, it was decided that for the purposes of this research, only methods that could be applied to a monocular system would be considered.

A monocular visual system has the additional advantages of being passive and not requiring any supporting equipment to be installed on the target. Furthermore, monocular sensors are inexpensive to implement for rapid prototyping and field testing. The main challenge of the visual estimation portion of the landing task is that an ADS tends to oscillate in yaw in flight. This oscillation can be caused by wind, or by the vehicle dynamics after a control actuation. Rather than adding a complicated panning mount to the visual sensor, an algorithm was developed that uses the motion of the ADS in an advantageous way. Using the inherent oscillatory motion that occurs, the estimation algorithm incorporates additional measurements of target position created from different views of the target.

III. Projecting a 3D Space on to a 2D Image Plane

It is well known that the human brain forms a 3D representation of the scene observed by the human visual system using stereo vision. Monocular machine vision systems are known to suffer from a lack of depth perception, and thus need a working mathematical relationship between each object point location in the three-dimensional scene, and the corresponding image point location on the two-dimensional image plane. The perspective-projective transformation (see Schalkoff [5] for an excellent introduction) is a mapping of object coordinates $\mathbf{x}_o \in \mathbb{R}^3$ to image coordinates $\mathbf{x}_i \in \mathbb{R}^2$. Critical to this transformation is the selection of a coordinate system.

A. Using Homogeneous Coordinates

An effective technique for using matrix multiplication to calculate this transformation is the use of *homogeneous coordinates*, by which both object coordinates $\mathbf{x}_o \in \mathbb{R}^3$ and image coordinates $\mathbf{x}_i \in \mathbb{R}^2$ are multiplied by scale factors w_o and w_i, respectively, and these scale factors are appended as an additional element to each of these vectors. An overbar denotes the homogeneous versions of the coordinate vectors, as $\bar{\mathbf{x}}_o \in \mathbb{R}^4$ and $\bar{\mathbf{x}}_i \in \mathbb{R}^3$.

Thus, $\bar{\mathbf{x}}_o$ is a four-dimensional vector describing the coordinates of an object in the 3D world, and $\bar{\mathbf{x}}_i$ is a three-dimensional vector describing the coordinates of an image of that object on the two-dimensional image plane:

$$\bar{\mathbf{x}}_i = \begin{bmatrix} w_i u \\ w_i v \\ w_i \end{bmatrix} \qquad \bar{\mathbf{x}}_o = \begin{bmatrix} w_o x \\ w_o y \\ w_o z \\ w_o \end{bmatrix}, \tag{1}$$

where (u, v) are the coordinates of the image of the object on the 2D image plane.

B. Perspective-Projective Transformation

Assuming that the coordinates of both image and object are defined with respect to the same coordinate axes, and using homogeneous coordinates, the perspective-projective transformation can then be represented by matrix $\mathbf{P} \in \mathbb{R}^{3\times 4}$, such that

$$\bar{\mathbf{x}}_i = \mathbf{P}\bar{\mathbf{x}}_o. \tag{2}$$

Matrix $\mathbf{P}$ from (2) is known as the *projection matrix*.

Consider a coordinate frame with origin O_i located in the image plane, and with mutually orthogonal axes X_i, Y_i, and Z_i as shown in Fig. 2. The coordinates of the object in this frame are (x_o, y_o, z_o). The coordinates of the object's image on the image plane are as follows: coordinate u is the measure of the image position along the image plane Y_i axis, and coordinate v is the measure of the image position along the image plane Z_i axis.

The assumption that $x_o \gg f$ leads to the following approximations:

$$u \approx \frac{f y_o}{x_o} \qquad v \approx \frac{f z_o}{x_o}. \tag{3}$$

which can be cast in matrix form as:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & f & 0 & 0 \\ 0 & 0 & f & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} w' x_o \\ w' y_o \\ w' z_o \\ w' \end{bmatrix}. \tag{4}$$

where the scale factor for the homogeneous coordinates of the object has been normalized using $w' = w_o/w_i$. The matrix in (2) corresponds to $\mathbf{P}$ of (1).

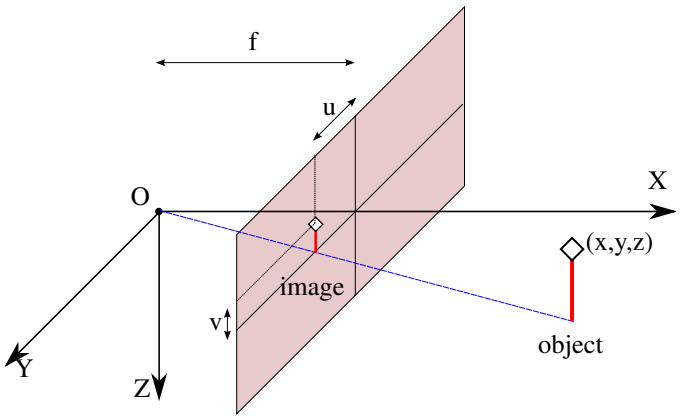


Fig. 2. Geometry of the perspective-projective transformation. The object with coordinates (x, y, z) in the 3D world is projected on to a 2D image plane. The coordinates of the image are (u, v).

C. Frames of Reference

In the following, a *global* coordinate frame will be used to describe objects in the 3D scene. This frame will have a different origin and orientation from the *image* coordinate frame that is aligned with the image plane. For the problem at hand, both the observer and target can be located with respect to the global coordinate frame, and two additional vectors defined: ${}^{\mathrm{g}}\mathbf{p}_T$ is the position vector of the target, and ${}^{\mathrm{g}}\mathbf{d}$ is the displacement, or position, vector of the observer.

Next, let the set of Euler angles, (ϕ, θ, ψ), represent a rotation from the global coordinate frame $\{g\}$ to the image coordinate frame $\{i\}$, and let $\mathbf{\Lambda}$ represent a vector with these three angles as elements. Let $\mathbf{R}(\mathbf{\Lambda})$ represent a rotation matrix that rotates $\{i\}$ to align with frame $\{g\}$. The target position in both global and image coordinate frames can now be related mathematically as:

$$ {}^{\mathrm{g}}\mathbf{p}_T = {}^{\mathrm{g}}\mathbf{d} + \mathbf{R}(\mathbf{\Lambda}){}^{\mathrm{i}}\mathbf{p}_T. \tag{5} $$

When homogeneous coordinates are used, the translation and rotation of (5) can be combined in one matrix operation:

$$ \mathbf{T}(\mathbf{\Lambda}, {}^{\mathrm{g}}\mathbf{d}) = \begin{bmatrix} \mathbf{R}(\mathbf{\Lambda}) & {}^{\mathrm{g}}\mathbf{d} \\ \mathbf{0} & 1 \end{bmatrix}. \tag{6} $$

The inverse relationship is:

$$ \mathbf{T}^{-1}(\mathbf{\Lambda}, {}^{\mathrm{g}}\mathbf{d}) = \begin{bmatrix} \mathbf{R}^{\mathrm{T}}(\mathbf{\Lambda}) & -\mathbf{R}^{\mathrm{T}}(\mathbf{\Lambda}){}^{\mathrm{g}}\mathbf{d} \\ \mathbf{0} & 1 \end{bmatrix} \tag{7} $$

which can be verified by computing $\mathbf{T}(\mathbf{\Lambda}, {}^{\mathrm{g}}\mathbf{d})\mathbf{T}^{-1}(\mathbf{\Lambda}, {}^{\mathrm{g}}\mathbf{d}) = \mathbf{I}$. The inverse relationship of (7) can be used to compute coordinates of the target in the image plane give the coordinates of the target in the global frame:

$$ \begin{bmatrix} {}^{\mathrm{i}}x_T \\ {}^{\mathrm{i}}y_T \\ {}^{\mathrm{i}}z_T \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}^{\mathrm{T}}(\mathbf{\Lambda}) & -\mathbf{R}^{\mathrm{T}}(\mathbf{\Lambda}){}^{\mathrm{g}}\mathbf{d} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} {}^{\mathrm{g}}x_T \\ {}^{\mathrm{g}}y_T \\ {}^{\mathrm{g}}z_T \\ 1 \end{bmatrix}. \tag{8} $$

The vector on the left-hand side is expressed in homogeneous coordinates in the image plane reference frame, but it expresses the coordinates of the *target itself* and not of the target's image on the image plane.

D. Transformation of an Object's Global Coordinates

The next step is to combine the projection, rotation, and translation operations into one expression so that the coordinates of the target's image on the image plane can be computed from the target's global coordinates. The sought expression can be made simpler by considering overall problem in terms of an airborne aerial delivery system as the observer, and a ship on the sea surface as the target. The z coordinate of the target *must* be zero if the origin of the global coordinate system is at sea level and the effects of vertical wave motion and the height of the ship's landing area above the sea surface are neglected. This assumption of ${}^{\mathrm{g}}z_T = 0$ allows the third element of the target coordinate vector to be eliminated along with the third column of the rotation and translation matrix. Then, the global coordinates of the target can be computed from the image plane measurements by:

$$ \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & f & 0 & 0 \\ 0 & 0 & f & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \overbrace{\left[\begin{array}{cc|c} \multicolumn{2}{c|}{\multirow{2}{*}{$\mathbf{R}^{\mathrm{T}}(\mathbf{\Lambda})$}} & -\mathbf{R}^{\mathrm{T}}(\mathbf{\Lambda}){}^{\mathrm{g}}\mathbf{d}_o \\ & & \\ \hline 0 & 0 & 1 \end{array}\right]}^{3^{\mathrm{rd}}\text{ column removed}}}_{\mathbf{M}} \begin{bmatrix} w'{}^{\mathrm{g}}x_T \\ w'{}^{\mathrm{g}}y_T \\ w' \end{bmatrix} \tag{9} $$

$$ \begin{bmatrix} w'{}^{\mathrm{g}}x_T \\ w'{}^{\mathrm{g}}y_T \\ w' \end{bmatrix} = \mathbf{M}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{10} $$

where the product of the 3×4 projection matrix and the 4×3 (reduced column) rotation and translation matrix is called $\mathbf{M}$. Equation (10) indicates a noteworthy result: because the target is constrained to a 2D plane, a *reverse projection* is possible. Given coordinates of the target's image on the image plane, the target's physical location on a 2D world surface can be computed.

IV. Target Estimation Algorithm

With a relationship established between the state of the target, and a measurement available to a visual sensor, a state-space-based estimation algorithm is now developed that enables calculation of relative pose between observer and target. The state of the surface target is naturally described in terms of its 2D position and its speed and course. As will be shown, this state vector formulation in terms of (x, y, V, ψ_T) allows measurements from the image plane to be used to compute both the target's location and its orientation.

A. Estimating the Target's Position

The nonlinear vector function $\mathbf{f}(\mathbf{x})$ used to compute the state derivatives is fairly straightforward and is shown here in

terms of the derivatives of the individual states:

$$\dot{x} = V\cos\psi_T \tag{11}$$
$$\dot{y} = V\sin\psi_T \tag{12}$$
$$\dot{V} = 0 \tag{13}$$
$$\dot{\psi_T} = 0. \tag{14}$$

Equation (10) contains a nonlinear relationship between the available image plane measurements and two of the elements of the target's state vector. To deal with this nonlinearity, the measurement equation will be cast as:

$$\mathbf{g}(\mathbf{z}) = \mathbf{H}\mathbf{x} + \boldsymbol{\gamma}. \tag{15}$$

where the vector $\mathbf{g}(\mathbf{z})$ is the pseudomeasurement vector and $\boldsymbol{\gamma}$ is the vector of pseudomeasurement noise.

B. Estimating the Target's Orientation

Returning to the assumption that the target is a vessel underway on calm seas, the target can be assumed to be an oblong object constrained to lie horizontally on the world's surface. Also, the target's motion is constrained to be in the direction of its own longitudinal axis, which would lie on a 2D surface within the global coordinate system. Thus, the orientation of the target on a 2D surface in the world (hence its course, ψ_T), may be inferred by the orientation ψ_i of the oblong image on the image plane. Assuming that the ends of the target's image, the bow and stern features, are able to be extracted flawlessly from the target's image, the orientation of the image then can be computed easily.

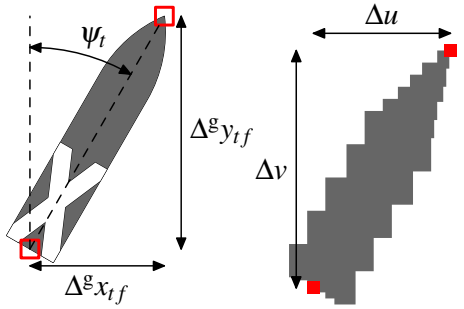
Consider the depiction in Fig. 3 of the target with bow and stern points identified in the global frame (Fig. 3a). Let $\Delta^{\mathrm{g}}x_{T_f}$ and $\Delta^{\mathrm{g}}y_{T_f}$ be the difference in global coordinates between the bow and stern features of the target. Therefore, there are two positions of interest for the target that are contained in the following vectors:

$${}^{\mathrm{g}}\mathbf{p}_{T_1} = \begin{bmatrix} {}^{\mathrm{g}}x_{T_f} \\ {}^{\mathrm{g}}y_{T_f} \\ 0 \\ 1 \end{bmatrix} \quad \text{and} \quad {}^{\mathrm{g}}\mathbf{p}_{T_2} = \begin{bmatrix} {}^{\mathrm{g}}x_{T_f} + \Delta^{\mathrm{g}}x_{T_f} \\ {}^{\mathrm{g}}y_{T_f} + \Delta^{\mathrm{g}}y_{T_f} \\ 0 \\ 1 \end{bmatrix} \tag{16}$$

where $({}^{\mathrm{g}}x_{T_f}, {}^{\mathrm{g}}y_{T_f})$ are the coordinates of one feature, such as the bow of the target ship. The third coordinate of each vector is zero because the entire target ship, including bow and stern, must line on the world surface plane. Each of the vectors in (16) thus loses its third element, and then each serves as input to (9) to compute image plane coordinates (u_1, v_1) and (u_2, v_2) corresponding to the bow and stern locations of the image.

$$\begin{bmatrix} w_1u_1 \\ w_1v_1 \\ w_1 \end{bmatrix} = \mathbf{M}(\boldsymbol{\Lambda}, {}^{\mathrm{g}}\mathbf{d}) \begin{bmatrix} {}^{\mathrm{g}}x_{T_f} \\ {}^{\mathrm{g}}y_{T_f} \\ 1 \end{bmatrix} \tag{17}$$

$$\begin{bmatrix} w_2u_2 \\ w_2v_2 \\ w_2 \end{bmatrix} = \mathbf{M}(\boldsymbol{\Lambda}, {}^{\mathrm{g}}\mathbf{d}) \begin{bmatrix} {}^{\mathrm{g}}x_{T_f} + \Delta^{\mathrm{g}}x_{T_f} \\ {}^{\mathrm{g}}y_{T_f} + \Delta^{\mathrm{g}}y_{T_f} \\ 1 \end{bmatrix}. \tag{18}$$



(a) Actual target in global coordinate frame. (b) Image of target in image plane.

Fig. 3. Target features in global frame (a) and image features on image plane (b). Bow and stern features on the target in the global coordinate frame are related to these same features on the image of the target in the image plane.

In (17) and (18), the notation $\mathbf{M}(\boldsymbol{\Lambda}, {}^{\mathrm{g}}\mathbf{d})$ emphasizes that $\mathbf{M}$ depends only on the location and orientation of the observer, not the target.

Equations (17) and (18) can be resolved together to produce an expression relating the horizontal and vertical differences on the image plane between bow and stern features, and the corresponding differences on a plane in global coordinates for the target's actual bow and stern. The resulting expression is:

$$\begin{bmatrix} w\Delta u \\ w\Delta v \end{bmatrix} = \underbrace{\left\{ \mathbf{M}(1:2, 1:2) - \begin{bmatrix} u \\ v \end{bmatrix} \begin{bmatrix} M(3,1)M(3,2) \end{bmatrix} \right\}}_{\mathbf{S}} \begin{bmatrix} \Delta^{\mathrm{g}}x_{T_f} \\ \Delta^{\mathrm{g}}y_{T_f} \end{bmatrix}. \tag{19}$$

In (19), a MATLAB-like notation is used to indicate a submatrix of $\mathbf{M}$ consisting of the first two rows and first two columns ($\mathbf{M}(1:2, 1:2)$), and also the individual scalar elements of the matrix $\mathbf{M}$, such as the third row, first column ($M(3,1)$). Also in (19), w is a scale factor for homogeneous coordinates, and $\Delta^{\mathrm{g}}x_{T_f}$ and $\Delta^{\mathrm{g}}y_{T_f}$ are meant to convey changes in global coordinates of *target features*, namely the difference between bow and stern positions.

The 2×2 matrix indicated as $\mathbf{S}$ in (19) can be inverted to obtain:

$$\begin{bmatrix} \Delta^{\mathrm{g}}x_{T_f} \\ \Delta^{\mathrm{g}}y_{T_f} \end{bmatrix} = \mathbf{S}^{-1} \begin{bmatrix} w\Delta u \\ w\Delta v \end{bmatrix}. \tag{20}$$

The target's computed course in the global coordinate system then is simply

$$\psi_T = \arctan\left(\frac{\Delta^{\mathrm{g}}x_{T_f}}{\Delta^{\mathrm{g}}y_{T_f}} \right). \tag{21}$$

C. Missing Measurements and Epipolar Geometry

A noteworthy aspect of this particular visual estimation problem is that the camera is fixed to the body of the ADS and rotates with it. These rotations are quite pronounced in flight; therefore, the duration that the target is out-of-frame is significant. An advantage is gained from casting this situation in terms of multiple-view geometry, whereby the instants when the target exits the field of view, and when it re-enters the field of view are characterized as two views of the same scene. Of

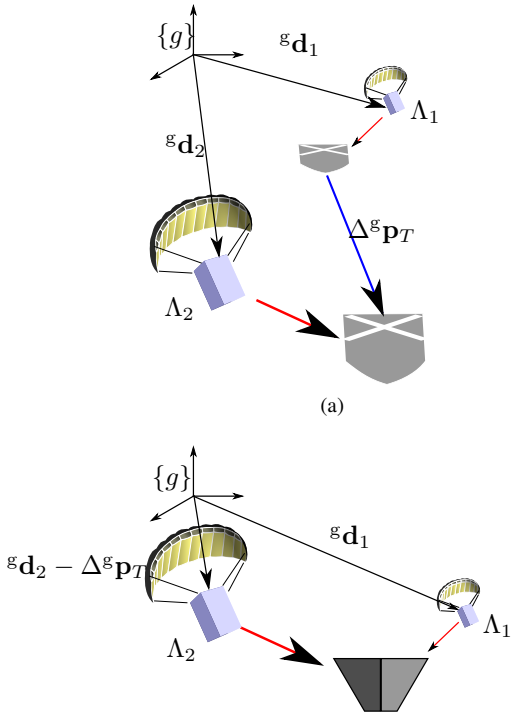


Fig. 4. Two views of the target. Actual observations of the target are separated in time and space (a). Assuming constant target motion, the two views can be brought together (b).

course, the target has moved between these two instants of time, but with the help of the assumption that the target's motion has constant course and speed, the second view of the target (entering frame) can be properly shifted in time so that it coincides with the first view. Assuming that the target image can be condensed to a single point, for example the centroid of the target, then the two views are of the very same point.

Consider two views of the same point target depicted in Fig. 4a. This depiction is of two separate time instants at which the observer positions and orientations with respect to global reference frame $\{g\}$ are $(\mathbf{d}_1, \Lambda_1)$ and $(\mathbf{d}_2, \Lambda_2)$. The target is out of the field of view of the observer between these two time instants, and the distance that the target has traveled over this duration is $\Delta^{g}\mathbf{p}_T$. In Fig. 4b, the second observation has been shifted by the movement $\Delta^{g}\mathbf{p}_T$ of the target during the time that the target was out of view. This calculation is equivalent to retarding the second observation in time because the velocity of the target is assumed to be constant. Thus, Fig. 4b depicts a situation where there are effectively two views of the *same* target.

The epipolar constraint stipulates that when there are two views of the same object in two different image planes, the location if the image in one plane constrains the set of possible image locations in the second image plane, provided that the

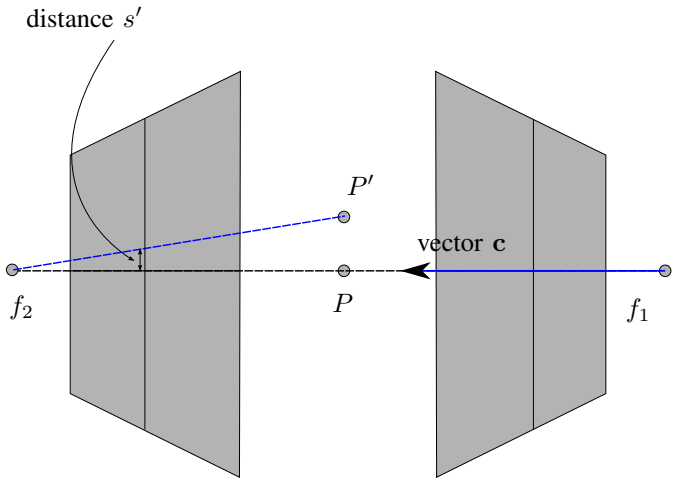


Fig. 5. Epipolar constraint with both image planes. When the target reappears at location P', distance s' is the distance between the location of the image of P' and the epipolar constraint line.

baseline vector between the two image planes is known. In the case at hand, the set of possible image locations is the epipolar line, which is established in the first-in-view image plane by the location of the previous image on the last-in-view image plane. A scalar measurement can be made whose value is the distance between the target centroid, at the instant that the target returns to view, and the epipolar line. Let this measurement be called s'. In Fig. 5, both image planes are shown together, with the left-hand image plane representing last-in-view (target exiting frame), and the right-hand image plane representing first-in-view, or target returning to the frame.

The geometry has been contrived to be simple in Fig. 5, where f_1 and f_2 are the focal points of the two image planes, and $\mathbf{c}$ is the vector along the line joining the two focal points. Point P is the target's centroid position at the instant of leaving the frame, and it is coplanar with line $\mathbf{c}$ in a perfectly horizontal plane in this depiction. Point P' represents the target's centroid position at the instant of re-entering the frame. Ideally, because the returning-to-frame view has been shifted in time to coincide with the leaving frame view, points P and P' should also be coincident. Figure 5 depicts the case where P and P' are not coincident, and scalar distance s' is shown as the distance in the image plane between the aforementioned epipolar constraint line, and the intersection point of the image plane itself and the line of sight between f_2 to P'.

Another interpretation of the epipolar constraint is that it stipulates that the lines of sight $\mathbf{v}_1$ and $\mathbf{v}_2$, which are rays from each of the focal points toward the target, must both be coplanar with baseline $\mathbf{c}$ joining the two focal points. Thus, the scalar triple product of $\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{c}$ should be zero; therefore, in a perfect situation, distance s' should be zero:

$$\mathbf{v}_1^{\mathrm{T}}(\mathbf{c} \times \mathbf{v}_2) = 0 = s'. \tag{22}$$

For implementation, scalar distance s' is split into two components by splitting the baseline vector $\mathbf{c}$ into two compo-

nents: one depending only on observer motion, and the other depending only on target motion. Let these two components of baseline vector $\mathbf{c}$ be labeled $\mathbf{c}_1$ and $\mathbf{c}_2$, respectively. The change results in the two quantities used in the estimation algorithm:

$$\mathbf{v}_1^{\mathrm{T}}\left[(\mathbf{c}_1+\mathbf{c}_2)\times\mathbf{v}_2\right]=0 \tag{23}$$

$$\underbrace{\mathbf{v}_1^{\mathrm{T}}(\mathbf{c}_1\times\mathbf{v}_2)}_{s}=\underbrace{\mathbf{v}_1^{\mathrm{T}}(\mathbf{v}_2\times\mathbf{c}_2)}_{\hat{s}} \tag{24}$$

where scalar value $\hat{s}$ is called the predicted scalar measurement and is calculated based on the component containing the target's motion, and scalar value s is called the scalar measurement and is calculated from component $\mathbf{c}_1$ that depends on the observer's motion.

The vector cross-products in (24) must have both operands expressed in the same coordinate frame. To this end, rotation matrices serve to transform vectors from one coordinate frame to another. Note that the baseline vector $\mathbf{c}$ and the lines-of-sight vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ are all free vectors, so locations of the coordinate frame origins do not matter, and no translations are required. The full expressions for scalar measurement s and predicted scalar value $\hat{s}$ that include these rotation matrices are:

$$s=\mathbf{v}_1^{\mathrm{T}}\left\{\mathbf{R}^{\mathrm{T}}(\mathbf{\Lambda}_1)\mathbf{R}(\mathbf{\Lambda}_2)\left[\mathbf{R}^{\mathrm{T}}(\mathbf{\Lambda}_2)\left({}^{\mathrm{g}}\mathbf{d}_1-{}^{\mathrm{g}}\mathbf{d}_2\right)\times\mathbf{v}_2\right]\right\} \tag{25}$$

$$\hat{s}=\mathbf{v}_1^{\mathrm{T}}\left\{\mathbf{R}^{\mathrm{T}}(\mathbf{\Lambda}_1)\mathbf{R}(\mathbf{\Lambda}_2)\left[\mathbf{v}_2\times\mathbf{R}^{\mathrm{T}}(\mathbf{\Lambda}_2)\Delta{}^{\mathrm{g}}\mathbf{p}_T\right]\right\}. \tag{26}$$

Predicted scalar value $\hat{s}$ can be computed as the multiplication of row vector $\mathbf{h}^{\mathrm{T}}$ and the target velocity extracted from estimated state vector $\hat{\mathbf{x}}$:

$$\hat{s}=\underbrace{\mathbf{v}_1^{\mathrm{T}}\mathbf{R}^{\mathrm{T}}(\mathbf{\Lambda}_1)\mathbf{R}(\mathbf{\Lambda}_2)\mathbf{S}(\mathbf{v}_2)\mathbf{R}^{\mathrm{T}}(\mathbf{\Lambda}_2)}_{\mathbf{h}^{\mathrm{T}}}\mathbf{R}_t(\psi)\begin{bmatrix}0&0&1&0\\0&0&0&0\\0&0&0&0\end{bmatrix}\Delta t\hat{\mathbf{x}} \tag{27}$$

where time interval Δt is defined as a multiple of the fundamental sampling rate: $N\cdot T_s$ and matrix $\mathbf{R}_t(\psi)$ resolves the target's course into 2D planar global coordinates. A compact expression for predicted scalar value $\hat{s}$ can be written as:

$$\hat{s}=\mathbf{h}^{\mathrm{T}}NT_s\begin{bmatrix}{}^{\mathrm{g}}V_x\\{}^{\mathrm{g}}V_y\\0\end{bmatrix}, \tag{28}$$

where, in this expression, ${}^{\mathrm{g}}V_x$ and ${}^{\mathrm{g}}V_y$ are two components of the target's velocity vector expressed in the global coordinate frame.

D. State-Space Estimation

In contrast to the conventional Kalman estimation algorithm, this additional measurement using the epipolar constraint uses additional information at the time the target returns to view. The conventional Kalman estimation algorithm does not use this additional information, but instead must set the measurement error covariance matrix $\mathbf{R}$ to infinity when the target is not in view. By so doing, the traditional Kalman estimator ignores all measurements during the out-of-view period and instead uses only the most recently-computed position, course, and speed of the target to predict its future position while running the estimation algorithm with no *new* data. This process is also known as *dead-reckoning* the target position.

The estimator using the epipolar measurement is also not using measurements when the target is out of view, but the reason in this case is that the estimator is extending its sampling interval forward in time until the target returns to view. The algorithm will not make any more computations until that time. When the target returns to view, the error signal in the Kalman state update equation using this epipolar measurement is actually:

$$e=s-\hat{s}=s-\mathbf{h}^{\mathrm{T}}NT_s\begin{bmatrix}{}^{\mathrm{g}}V_x\\{}^{\mathrm{g}}V_y\\0\end{bmatrix} \tag{29}$$

instead of

$$\mathbf{e}=\mathbf{g}(\mathbf{z})-\mathbf{H}\hat{\mathbf{x}} \tag{30}$$

as it would be during normal, target-in-view processing.

In this way, the state space estimation algorithm has two different models, or modes: one mode for the target in view, and one for the target out of view. The mode for which the target is out of view has a variable sampling rate because it takes one sample at the instant that the target is leaving the image frame, and another sample at the instant that the target returns to the image frame. For the time interval between these two instants, the estimator is neither taking samples nor performing computations. Both modes, or models, are represented below:

$$\begin{cases}\mathbf{x}[n+1]=\mathbf{\Phi}(T_s)\mathbf{x}[n]+\mathbf{G}\mathbf{w}[n] & \text{in view}\\ \mathbf{x}[n+1]=\mathbf{\Phi}(NT_s)\mathbf{x}[n]+\mathbf{G}\mathbf{w}[n] & \text{returning to view}\\ \mathbf{g}(\mathbf{z}[n])=\mathbf{H}\mathbf{x}[n]+\boldsymbol{\gamma}[n] & \text{in view}\\ s[n]=\mathbf{h}^{\mathrm{T}}NT_s\begin{bmatrix}{}^{\mathrm{g}}V_x\\{}^{\mathrm{g}}V_y\\0\end{bmatrix}+\boldsymbol{\gamma}[n] & \text{returning to view}\end{cases} \tag{31}$$

Note that the state equations were written in terms of discrete-time state transition matrix $\mathbf{\Phi}$, even though in practice, the state update is calculated using the nonlinear state update function $\mathbf{f}$. Furthermore, the state transition matrix is written with the sampling interval T_s (sometimes multiplied by number of out-of-view intervals N) as an explicit argument so that the two rates of the two modes are clearly shown.

V. Simulation Results

A simulation was designed using Simulink to evaluate the benefit of the epipolar constraint measurement over the method of simply using the target's dead-reckoning (DR) position when it is not in view. A very convenient artificial motion was programmed for the observer in this simulation to achieve a wide separation between the last-in-view and return-to-view focal point positions. In Fig. 6, the observer is moving North to South and crossing the target's course while its field of

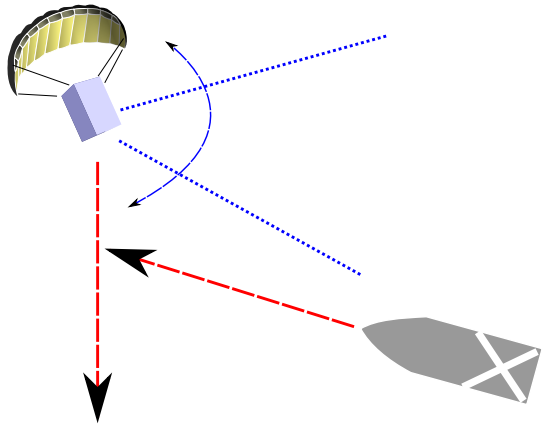


Fig. 6. Geometry of simulation to evaluate epipolar measurement. The observer moves across the target's course so that last-in-view and return-to-view locations are widely separated.

regard rotates in the horizontal plane. For this simulation, the observer was programmed to turn away from the target after a 3 s period of observation, then turn back toward the target after a time delay of 130 s, resulting in approximately two minutes during which the target was out of view.

The parameters that were adjusted in this model were measurement noise and measurement error covariance matrix $\mathbf{R}$. The objective of this set of simulations was to determine whether the use of the epipolar measurement at the back-in-frame instant conferred any advantage over an estimator that simply ignored missing measurements when the target was out of view, and resumed normal processing when the target returned to view—the DR method.

The parameter that was varied for the first set of simulations was measurement noise. A nominal set of assumed measurement noise standard deviation values was chosen for the physical sensors on the autopilot such as the GPS and the attitude sensors, and the vector of these values was labeled $\bar{\boldsymbol{\sigma}}$. The elements of vector $\bar{\boldsymbol{\sigma}}$ correspond to the assumed standard deviations of noise applied to each raw sensor measurement such as observer GPS position, observer orientation, and the target's image position on the image plane. For the simulation runs, both the measurement error covariance matrix $\mathbf{R}$ as well as the actual added measurement noise, were calculated using multiples of the values of the elements of $\bar{\boldsymbol{\sigma}}$.

The first set of simulation runs in this section was conducted while varying the added and assumed measurement noise from zero to $2\bar{\boldsymbol{\sigma}}$ to $4\bar{\boldsymbol{\sigma}}$. For the first run with actual added measurement noise set to zero, measurement error covariance matrix $\mathbf{R}$ was still computed with values of $1\bar{\boldsymbol{\sigma}}$ so that $\mathbf{R}$ would not be a zero matrix. For all these simulations, actual added process noise was kept at zero to isolate the effect of measurement noise. From the simulation output, position and velocity estimation errors are shown in Table I at the instant that the target returns to view. Along with the estimation error value, the standard deviation of the estimate is represented as the square root of the value in the appropriate element of the state estimation error covariance matrix $\mathbf{P}$.

TABLE I
ESTIMATION ERRORS WITH LONG OUT-OF-VIEW DURATION.

Position estimation error and standard deviation/m			
estimator	amount of measurement noise		
	none	$2\bar{\boldsymbol{\sigma}}$	$4\bar{\boldsymbol{\sigma}}$
extended/DR	16.264	11.768	32.313
$\sigma =$	27.525	56.608	97.779
extended/epi	14.023	26.938	14.501
$\sigma =$	59.808	92.540	151.764
Velocity estimation error and standard deviation/m/s			
estimator	amount of measurement noise		
	none	$2\bar{\boldsymbol{\sigma}}$	$4\bar{\boldsymbol{\sigma}}$
extended/DR	−0.113	0.125	0.312
$\sigma =$	0.394	0.534	0.805
extended/epi	−0.086	−0.115	−0.016
$\sigma =$	0.304	0.585	1.074

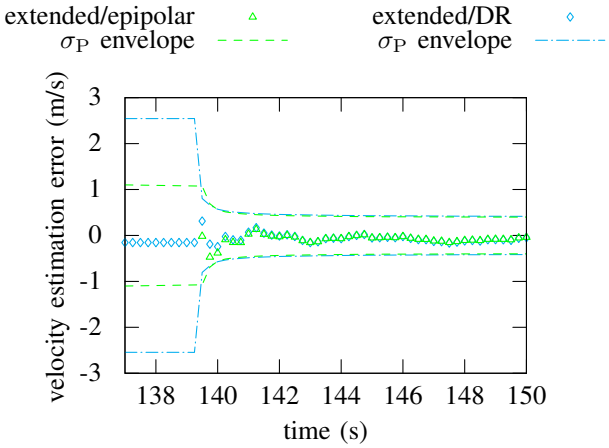


Fig. 7. Velocity estimation error after target returns to view. For this simulation, the target was out of view until 139.5 s of simulation time.

In Table I, the velocity estimation error of the extended Kalman filter (EKF) estimator using the epipolar measurement is lower in all three cases than that of the EKF using DR; however, the computed state estimation error values are higher in two cases than those of the EKF DR estimator.

A history of estimates from the back-in-view instant to the end of the simulation is shown in Fig. 7; this figure depicts the velocity estimation error from the same simulation run whose data was used to construct Table I. In Fig. 7, only the end of the simulation is shown, with the back-in-view instant occurring at 139.5 s of simulation time. For the back-in-view instant, the estimator using the epipolar measurement has a smaller velocity estimation error; although, both estimators converge to small levels of error within 2 s.

To explore further the effect of increasing measurement noise on velocity estimation error, another set of simulations was executed, this time with matrix $\mathbf{R}$ computed with $1\bar{\boldsymbol{\sigma}}$ values, but actual added measurement noise ranging from $5\bar{\boldsymbol{\sigma}}$ to $20\bar{\boldsymbol{\sigma}}$. Velocity estimation errors for these cases are shown in

TABLE II
VELOCITY ESTIMATION ERROR WITH HIGH MEASUREMENT NOISE.

Velocity estimation error and standard deviation/m/s estimator	amount of measurement noise		
	$5\bar{\sigma}$	$10\bar{\sigma}$	$20\bar{\sigma}$
extended/DR	0.185	0.393	0.942
$\sigma =$	0.438	0.434	0.419
extended/epi	-0.322	-0.631	-0.609
$\sigma =$	0.306	0.333	0.372

Table II. The velocity estimation errors shown therein for the EKF epipolar estimator are generally increasing with increasing measurement noise, although not monotonically. Considering Tables I and II together, it is apparent that the EKF using the epipolar measurement has a smaller velocity estimation error in four of the six cases presented.

VI. CONCLUSION

The output of these simulations indicates that the use of the epipolar measurement does confer an advantage to an estimator that makes use of the additional information that the underlying two-view geometry provides. This advantage appears to remain even for high levels of measurement noise. Future plans include the hardware implementation of a machine vision system and the estimator described herein into the Snowflake prototype ADS for flight testing.

REFERENCES

[1] R. Benney, M. Henry, K. Lafond, and A. Meloni, "DoD new JPADS programs and NATO activities," in *Proceedings of the 20th Aerodynamic Decelerator Systems Technology Conference*. Seattle, WA: AIAA, 4–7 May 2009, paper AIAA 2009-2952.

[2] C. W. Hewgley and O. A. Yakimenko, "Precision guided airdrop for vertical replenishment of naval vessels," in *Proceedings of the 20th Aerodynamic Decelerator Systems Technology Conference*. Seattle, WA: AIAA, 4–7 May 2009, paper AIAA-2009-2995.

[3] C. W. Hewgley, O. A. Yakimenko, and N. J. Slegers, "Shipboard landing challenges for autonomous parafoils," in *Proceedings of the 21st Aerodynamic Decelerator Systems Technology Conference*. Dublin, Ireland: AIAA, 23–26 May 2011, paper AIAA 2011-2573, pp. 846–855.

[4] O. A. Yakimenko, N. J. Slegers, and R. Tiaden, "Development and testing of the miniature aerial delivery system snowflake," in *Proceedings of the 20th Aerodynamic Decelerator Systems Technology Conference*. Seattle, WA: AIAA, 4–7 May 2009, paper AIAA-2009-2980.

[5] R. J. Schalkoff, *Digital image processing and computer vision*. New York, NY: John Wiley & Sons, Inc., 1989.