JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 33, No. 2, March–April 2010

Vision-Based Geolocation Tracking System for Uninhabited Aerial Vehicles

Mark E. Campbell*
Cornell University, Ithaca, New York 14853
and
Matthew Wheeler†
The Insitu Group, Bingen Washington 98605

DOI: 10.2514/1.44013

The design and implementation of a vision-based geolocation tracking system for uninhabited aerial vehicles is described. The geolocation tracking system for the uninhabited aerial vehicles includes avionics, a gimballing camera with feedback isolation and command loops, and a ground station. The point of interest is locked in the camera image by maintaining the center of the image frame to frame. An architecture for a geolocation tracking estimator is developed and demonstrated. The estimator has the unique characteristics of being modular so that it can work with different camera systems and different avionics components, compensate for random and bias uncertainties, run in real time, and deliver a consistent estimate of the location and uncertainty of the object being tracked. Flight results using the SeaScan uninhabited aerial vehicles show consistent results for two- and three-dimensional tracking of stationary and moving targets.

Nomenclature

$B_{(\cdot)}$	=	uniform bias distribution for variable $(\cdot)$ of width B
$F_{(\cdot)}$	=	field of view for the $(\cdot)$ axis
$f_{(\cdot)}$	=	dynamics functions for the $(\cdot)$ states
$g_{(\cdot)}$	=	measurement output functions for the $(\cdot)$ measurement
K_{POI}	=	Kalman gain in the geolocation square root sigma point filter estimator
n_a	=	number of states in the augmented system (prediction step of square root sigma point filter)
$n_{\text{C-POI}}$	=	number of measurements of the C-POI
n_{POI}	=	number of states in the point of interest state vector
n_w	=	number of disturbances
n_{2a}	=	number of states in the secondary augmented system (update step of square root sigma point filter)
(p, t, s)	=	pan, tilt, scan of gimbal; components of $\mathbf{x}_{\text{GIM}}$
$p^{\text{SCR}}_{(\cdot)\text{-max}}$	=	maximum number of pixels in the $(\cdot)$ axis of an image
$Q_{(\cdot)}$	=	process noise covariance, where $(\cdot) \in$ [NAV, ATT, GIM, POI]
R_a, R_b	=	Earth equatorial, polar radii
$R_{av(\cdot)}$	=	vector length from center of Earth to the $(\cdot)$ states
$R^{C_2}_{C_1}$	=	rotation matrix from C_1 to C_2
$R_{(\cdot)}$	=	sensor noise covariance, where $(\cdot) \in$ [C-POI, GIM]
S_{POI}	=	Cholesky factor (square root) of the point of interest error covariance matrix
u	=	path variable
$\mathbf{v}_{(\cdot)}$	=	sensor noise, where $(\cdot) \in$ [C-POI, GIM]
W_m, W_c	=	weights for mean, covariance in the square root sigma point filter estimator
$\mathbf{w}_{(\cdot)}$	=	process noise, where $(\cdot) \in$ [NAV, ATT, GIM, POI]
$\mathcal{X}^-$, $\mathcal{X}$, $\mathcal{X}^c$	=	predicted, updated, centralized state sigma points
$\mathbf{x}_{\text{ATT}}$	=	three-dimensional attitude of the uninhabited aerial vehicles
$\mathbf{x}_{\text{GIM}}$	=	three-dimensional pointing of the camera sensor by the gimbal
$\mathbf{x}_{\text{POI}}$	=	state vector of the point of interest
$\hat{\mathbf{x}}_{\text{POI}}$	=	estimate of point of interest state
$\mathbf{x}_{\text{NAV}}$	=	three-dimensional position/navigation of the uninhabited aerial vehicles
$\mathbf{x}_{x2}$	=	secondary states, $=[\mathbf{x}^T_{\text{NAV}}, \mathbf{x}^T_{\text{ATT}}, \mathbf{x}^T_{\text{GIM}}]^T$
$\mathcal{Y}^-$, $\mathcal{Y}$, $\mathcal{Y}^c$	=	predicted, updated, centralized measurement sigma points
$\mathbf{y}^{\text{SCR}}_{\text{C-POI}}$	=	center of the projected point of interest in the camera image, in screen coordinates (pixels)
$\mathbf{y}_{\text{GIM}}$	=	measurement of the three-dimensional pointing gimbal
$\lambda_{(\cdot)}$	=	pixel length scale factor for the $(\cdot)$ axis
$\Phi_{(\cdot)}$	=	characteristic function for Gaussian or uniform distribution
(ϕ, θ, ψ)	=	roll, pitch, yaw of uninhabited aerial vehicles, components of $\mathbf{x}_{\text{ATT}}$
$(\cdot)_a$	=	augmented state/covariance in square root sigma point filter estimator
$(\cdot)_{\text{POI-B}}$	=	variable of point of interest state with added shipset uncertainty bound
$(\cdot)^{\text{coord}}_{\text{type}}$	=	type $\in$ {NAV, ATT, GIM, POI}, coord $\in$ {LLA, ECEF, CAM, ENU, SCR}

Presented as Paper 6246 at the AIAA Guidance, Navigation, and Control Conference, Keystone, CO, 21–24 August 2006; received 24 February 2009; revision received 23 November 2009; accepted for publication 23 November 2009. Copyright © 2009 by Mark Campbell and Matthew Wheeler. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission. Copies of this paper may be made for personal or internal use, on condition that the copier pay the $10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 0731-5090/10 and $10.00 in correspondence with the CCC.

*Associate Professor, Sibley School of Mechanical and Aerospace Engineering; mc288@cornell.edu. Associate Fellow AIAA.

†Special Projects Lead, 118 East Columbia River Way; Matt.Wheeler@insitu.com.

Introduction

THE implementation of uninhabited aerial vehicles (UAVs) is envisioned for a wide variety of missions, such as defense, search and rescue, special emergency services [1], and fire fighting [2]. A recent application that nicely uses groups of UAVs is search and rescue in hurricane season [3]. During hurricane Katrina in September 2005, the Navy lent 10 Evolution-class UAVs with visual cameras to the search process, which were used to relay pictures back

to command. Also, five Fire-Fox-class UAVs with thermal sensors were used to search for survivors [4].

Maturing algorithms and technologies, increased defense funding, and decreased component costs have allowed UAVs to be built at many different levels, from the university level [5,6] to the small industry level [7–11] and, of course, to the sophisticated level of large industry, typically for defense purposes [12,13] such as Predator and Global Hawk. Universities use UAV testbeds to validate research algorithms in a realistic setting. With a typical flight time of $\sim$30 min, however, along with common computational and power constraints that require algorithms to run off-board on the ground, university UAVs can be far from a final industry-built UAV validation.

A key technology in UAV systems currently being explored is the tracking of stationary or moving ground targets using visual cameras for payloads. The tracking system requires the complex integration of several hardware components (camera, UAV, Global Positioning Systems (GPSs), and attitude sensors) and software components (camera image processing, inner loop and path planning control, and estimation software) to develop realistically accurate estimates of the object being tracked. Off-the-shelf digital cameras, which have been developed inexpensively recently by the electronics industry for consumer usage, have enabled both industry and academia to drastically reduce the cost of the UAV and payload system.

The ultimate performance of the vision-based geolocation tracking system for production UAVs is also a challenge because of several key characteristics. First, production UAVs typically use one of several choices for hardware components, such as inertial navigation and camera or gimbal systems. Thus, it is desirable to have a modular geolocation estimator that uses only the output data products of these components (and their uncertainty models), not additional models, such as of the aircraft. This minimizes the changes to the geolocation estimator for each UAV. Second, production UAVs will vary in their characteristics across a shipset (multiple UAVs with the same components), particularly their output data products and biases. Thus, it is desirable to have one estimator that can be used for all UAVs in the same shipset, requiring the need to fuse information about both sensors and bias uncertainties. Third, most small UAVs have small processors because of their limited size and power requirements. Thus, it is desirable to reduce computation in the estimator as much as possible to meet the given requirements. Fourth, the output of the estimator must be statistically accurate in geolocating the point of interest. Although these four characteristics have been posed in general form, specific requirements placed on these characteristics by the actual application will drive the final solution; this paper presents a general form of an estimator that can be adapted to these final requirements.

Several research groups are using UAVs with vision systems, with applications such as road following [14], refueling, and obstacle avoidance [15]. For the target tracking application, Ridley et al. [5] and Grocholsky et al. [16] have implemented UAV systems with cameras using decentralized fusion (information filtering) concepts; the cameras were typically not gimbaled. Several groups have implemented gimbaling camera systems on UAVs [17,18], a few with initial target tracking results [6,19]. Ivey and Johnson have compared different filtering techniques for UAV tracking showing similar results [20]. Kaaniche et al. [21] presented a traffic surveillance approach with UAVs using a graph cut formulation and a verification step. Several groups have also developed tight, fast trajectory planning loops that use vision feedback [22–24]. Most of these works typically only address one or several, but not all, of the aforementioned requirements for vision tracking systems for production UAVs; none address variation of uncertainties across UAVs in the same shipset.

This paper develops and empirically validates a vision-based UAV geolocation system for production UAVs, while addressing each of the aforementioned requirements. The proposed geolocation architecture is decentralized in that it directly uses sensory (or estimation) based data from the UAV, including the attitude and navigation estimates. This decentralized architecture allows the estimator to be easily used with different UAV components, including different UAVs, using the a priori known estimation statistics from the components. A modified version of the square root sigma point filter (SR-SPF) is used [25], as it has been shown to be more accurate than the extended Kalman filter [25], works nicely in real time, and easily integrates additional object (target) models. Two key extensions of the estimator are developed here. First, a modular architecture is developed, including a modular update step that enables the use of UAV data from other software modules and reduces computation. Second, unobservable biases are modeled and integrated with the estimator to produce consistent estimates. Empirical flight data and a truth measure of the point of interest (which is not available during typical flights) are used to model the biases, posed in the form of a uniform (U) density model. Using this model, an additional estimator is added that fuses the estimated mean and covariance from the geolocation estimator with the uniform bias density. The result is a consistent estimate of the object location and uncertainty, even across a range of production UAVs. Flight-test results are shown for stationary and moving targets using the SeaScan UAV, demonstrating consistent results. The SeaScan is a long-endurance (24 + h) UAV developed by the Insitu Group.

The paper is laid out as follows. The next section gives an overview of the decentralized estimation architecture, followed by detailed descriptions of the geolocation models, the square root estimator, and the confidence bound estimator in the presence of uniform bias uncertainties. The subsequent section details the SeaScan UAV and its components. The final section presents a summary of the SeaScan UAV and the flight-test results for stationary and moving objects.

Geolocation Problem

Key elements of the geolocation problem are shown in Fig. 1. A point of interest (POI: object, target, feature, etc.) on the ground is stationary or potentially moving. The aircraft (Fig. 1a) points a camera using a gimbaling payload mount inside the nose of the UAV at the POI. An image of the POI is then projected into the screen (SCR, where y is horizontal and z is vertical) frame, as denoted by C-POI in Fig. 1b. Because the aircraft is moving, and the POI is potentially moving, the camera gimbal must adjust its pointing angles in real time to maintain the C-POI inside the image frame.

The sensor measurement C-POI in Fig. 1b is a single point (the center of the object in this case) inside the SCR frame, in units of pixels. The derivation of this measurement equation occurs later in the paper; to summarize, however, this measurement equation $\mathbf{y}^{\text{SCR}}_{\text{C-POI}}$ is a nonlinear function of the UAV position (NAV), UAV attitude (ATT), POI position, turret gimbal pointing angles (GIM), and sensor noise $\mathbf{v}^{\text{SCR}}_{\text{C-POI}}$:

$$\mathbf{y}^{\text{SCR}}_{\text{C-POI}} = g_{\text{C-POI}}(\mathbf{x}_{\text{NAV}}, \mathbf{x}_{\text{ATT}}, \mathbf{x}_{\text{GIM}}, \mathbf{x}_{\text{POI}}, \mathbf{v}^{\text{SCR}}_{\text{C-POI}}) \tag{1}$$

where $\mathbf{x}_{(\cdot)}$ denotes a specific state vector. The objective of geolocation is then to estimate the position of the POI using this measurement, or $\hat{\mathbf{x}}_{\text{POI}}$. Complicating this problem are uncertainties in the aircraft position and orientation, turret gimbal angles, camera specifications and measurements, and disturbances such as turbulence and engine vibrations.

Modular Geolocation Estimator

The most straightforward estimator that could be designed for the geolocation problem uses models of the ATT and NAV, GIM, and POI because the output measurement [Eq. (1)] is a function of each of these states. Specifically, a single model and measurement can be written in block form as

$$\begin{bmatrix} \begin{bmatrix} \dot{\mathbf{x}}_{\text{NAV}} \\ \dot{\mathbf{x}}_{\text{ATT}} \\ \dot{\mathbf{x}}_{\text{GIM}} \end{bmatrix} \\ \dot{\mathbf{x}}_{\text{POI}} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} f_{\text{NAV}}(\mathbf{x}_{\text{NAV}}, \mathbf{x}_{\text{ATT}}, \mathbf{w}_{\text{NAV}}) \\ f_{\text{ATT}}(\mathbf{x}_{\text{ATT}}, \mathbf{w}_{\text{ATT}}) \\ f_{\text{GIM}}(\mathbf{x}_{\text{GIM}}, \mathbf{w}_{\text{GIM}}) \end{bmatrix} \\ f_{\text{POI}}(\mathbf{x}_{\text{POI}}, \mathbf{w}_{\text{POI}}) \end{bmatrix} \tag{2}$$

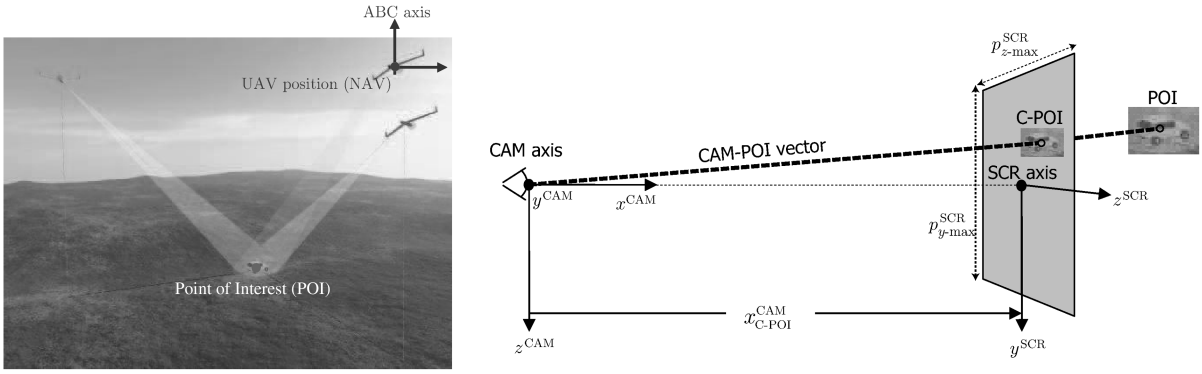

Fig. 1 Overview of the geolocation problem: a) POI on the ground, potentially moving, with a UAV in the air and a camera pointing at the POI; and b) onboard camera view (CAM axis), with the camera projecting the image of the POI onto a screen (SCR), denoted as C-POI. The measurement $\mathbf{y}^{\text{SCR}}_{\text{C-POI}} = [y^{\text{SCR}}_{\text{C-POI}}, z^{\text{SCR}}_{\text{C-POI}}]$ is the center of the C-POI in units of pixels in the SCR coordinate system.

$$\begin{bmatrix} \begin{bmatrix} \mathbf{y}_{\text{NAV}} \\ \mathbf{y}_{\text{ATT}} \\ \mathbf{y}_{\text{GIM}} \end{bmatrix} \\ \mathbf{y}^{\text{SCR}}_{\text{C-POI}} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} g_{\text{NAV}}(\mathbf{x}_{\text{NAV}}, \mathbf{v}_{\text{NAV}}) \\ g_{\text{ATT}}(\mathbf{x}_{\text{ATT}}, \mathbf{v}_{\text{ATT}}) \\ g_{\text{GIM}}(\mathbf{x}_{\text{GIM}}, \mathbf{v}_{\text{GIM}}) \end{bmatrix} \\ g_{\text{C-POI}}(\mathbf{x}_{\text{NAV}}, \mathbf{x}_{\text{ATT}}, \mathbf{x}_{\text{GIM}}, \mathbf{x}_{\text{POI}}, \mathbf{v}^{\text{SCR}}_{\text{C-POI}}) \end{bmatrix} \tag{3}$$

where $\mathbf{w}_{(\cdot)}$, $\mathbf{v}_{(\cdot)}$ denote the process, sensor noises. The dynamics models, $f_{(\cdot)}$, for the NAV, ATT, and GIM states are developed based on known principles and vehicle specifics, and are typically nonlinear; separate tracking models can defined for stationary or moving POIs using f_{POI}. The measurement functions, $g_{(\cdot)}$, typically measure the states, $\mathbf{x}_{(\cdot)}$, and include the nonlinear screen geolocation measurement defined in Eq. (1).

An estimator developed with Eqs. (2) and (3) is referred to here as being "centralized" because it includes models of all components of the system and generates estimates of all states including the aircraft NAV and ATT states. The benefit of this estimator is that it will be the most accurate, as geolocation is truly a function of all of these states. The centralized estimator also develops estimates of the correlations between the states. The drawback of this estimator is that most production UAVs already have sensors and software onboard for estimating aircraft NAV and ATT states. Also, a larger estimator such as this will be more sensitive to tuning. And finally, the computational burden of such a large estimator will be relatively high. In the case of a stationary POI, the estimator will require a model with 19 states.

Because NAV and ATT estimators typically already exist on many production UAVs inside the avionics suite, a modular form of the estimator is proposed here to simplify (and speed up) the estimation process, yet achieve similar performance as compared to the centralized version. The model, in this case, only includes the POI states, whereas the measurement is simply the screen coordinates of the C-POI:

$$\dot{\mathbf{x}}_{\text{POI}} = f_{\text{POI}}(\mathbf{x}_{\text{POI}}, \mathbf{w}_{\text{POI}}) \tag{4}$$

$$\mathbf{y}^{\text{SCR}}_{\text{C-POI}} = g_{\text{C-POI}}\left(\mathbf{x}_{\text{POI}}, \begin{bmatrix} \mathbf{x}_{\text{NAV}} \\ \mathbf{x}_{\text{ATT}} \\ \mathbf{x}_{\text{GIM}} \end{bmatrix}, \mathbf{v}^{\text{SCR}}_{\text{C-POI}}\right) \tag{5}$$

The measurement output is written in block form to denote the "secondary" states explicitly (NAV, ATT, GIM). In the case of the NAV and ATT states, estimates are provided by the onboard avionics, denoted as $(\hat{\mathbf{x}}_{\text{NAV}}, \hat{\mathbf{x}}_{\text{ATT}})$, with square root covariances of $(P_{\text{NAV}}, P_{\text{ATT}})$. In the case of the GIM states, direct sensor measurements from gimbal encoders are available, with sensor noise modeled as Gaussian (G) and white, or $\mathbf{x}_{\text{GIM}} \sim \mathcal{N}(\mathbf{y}_{\text{GIM}}, R_{\text{GIM}})$. The least-squares update step for this estimator is similar to that of a fully centralized geolocation estimator taking into account all system states, but producing only estimates of the POI states. Thus, it is less complicated than, faster than, and less sensitive to tuning of the other estimators.

Figure 2 shows the proposed modular architecture of the geolocation estimator in a block diagram. The model prediction step has reduced to only propagating the POI states; thus, models for aircraft NAV, ATT, and GIM are not required. The update step is more complicated because of the stochastic dependency on the NAV, ATT, and GIM states. In this case, the outputs (estimates, covariances) of the onboard estimators for the NAV and ATT states and the measurements of the GIM pointing are used directly. In the case of a stationary POI, the proposed modular estimator requires only four states (compared to 19 for a centralized solution).

Tracking Models

Bar-Shalom et al. [26] detailed a number of models for target tracking that can be used in Eq. (4). One of the most general forms is a random walk on each velocity state in three dimensions. With no loss of generality, this work assumes that the POI is a stationary or moving vehicle to give more accurate tracking results for these types of POIs. The nonlinear model in this case is written as

$$\begin{bmatrix} \dot{x}^{\text{LLA}}_{\text{POI}} \\ \dot{y}^{\text{LLA}}_{\text{POI}} \\ \dot{z}^{\text{LLA}}_{\text{POI}} \\ \dot{V} \\ \dot{\phi} \\ \dot{\omega} \\ \dot{a} \end{bmatrix} = \begin{bmatrix} VS(\phi)/R_e \\ VC(\phi)/(C(x^{\text{ENU}}_{\text{POI}})R_e) \\ 0 \\ a \\ \omega \\ -\omega/\rho_\omega \\ -a/\rho_a \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ w_\omega \\ w_a \end{bmatrix} \tag{6}$$

where $\mathbf{x}_{\text{POI}} = [x^{\text{LLA}}_{\text{POI}}, y^{\text{LLA}}_{\text{POI}}, z^{\text{LLA}}_{\text{POI}}]^T$ is the three-dimensional position in latitude, longitude, and altitude (LLA, where x is the latitude, y is the longitude, and z is the height) coordinates (typically in deg or rad, deg or rad, and m asl); V, ϕ, ω, and a are the velocity, heading, turn rate, and acceleration, respectively; R_e is the radius of the Earth; $C(\cdot)$ and $S(\cdot)$ denote the $\cos(\cdot)$ and $\sin(\cdot)$, respectively; and $\rho_{(\cdot)}$ are model tuning parameters based on probable motions of the target [26].

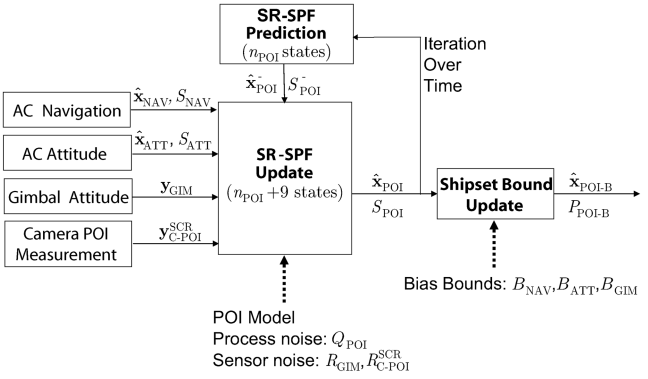

Fig. 2 Block diagram of the modular geolocation estimator. Solid lines indicate real-time data; dashed lines indicate preprogrammed data.

There is process noise $\mathbf{w}_{\text{POI}} = [w_\omega, w_a]^T$ on the acceleration and turn rate dynamic equations that model typical motions of ground objects. For the subsequent discrete implementation of the estimator, these dynamics are discretized and combined with a discrete form of the output equation:

$$\begin{aligned}\mathbf{x}_{\text{POI},k+1} &= \mathbf{x}_{\text{POI},k} + \Delta t(f_{\text{POI}}(\mathbf{x}_{\text{POI},k}) + \mathbf{w}_{\text{POI},k}) \\ &= f^D_{\text{POI}}(\mathbf{x}_{\text{POI},k}, \mathbf{w}_{\text{POI},k})\end{aligned} \tag{7}$$

$$\mathbf{y}^{\text{SCR}}_{\text{C-POI},k+1} = g_{\text{C-POI}}\left(\mathbf{x}_{\text{POI},k+1}, \begin{bmatrix}\mathbf{x}_{\text{NAV},k+1}\\ \mathbf{x}_{\text{ATT},k+1}\\ \mathbf{x}_{\text{GIM},k+1}\end{bmatrix}, \mathbf{v}^{\text{SCR}}_{\text{C-POI},k+1}\right) \tag{8}$$

Note that more precise and typically complex integration routines such as Runga–Kutta could also be used with no loss in generality.

Measurement

The measurement equation, Eqs. (1), (5), and (8), is a nonlinear function of the UAV and POI states, and yields the C-POI location in SCR coordinates, in units of pixels. Figure 1b shows the coordinate systems and notation used to derive the measurement question. For simplicity, it is assumed that the GPS antenna, center of the aircraft axes (ABC, where x is aligned with the fuselage of the UAV and z is vertically down), and camera axes (CAM, where x is the camera line of site and z is vertically down) are all located at the same point (this could be relaxed). It is noted that, in this derivation, k is dropped for notational simplicity.

Assume that the three-dimensional UAV and POI position states are in LLA coordinates, denoted as $\mathbf{x}^{\text{LLA}}_{\text{NAV}} = [x^{\text{LLA}}_{\text{NAV}}, y^{\text{LLA}}_{\text{NAV}}, z^{\text{LLA}}_{\text{NAV}}]^T$ and $\mathbf{x}^{\text{LLA}}_{\text{POI}} = [x^{\text{LLA}}_{\text{POI}}, y^{\text{LLA}}_{\text{POI}}, z^{\text{LLA}}_{\text{POI}}]^T$, respectively. These three-dimensional position states are converted to Earth-center–Earth-fixed (ECEF) coordinates using

$$\begin{aligned}x^{\text{ECEF}}_{\text{NAV}} &= C(y^{\text{LLA}}_{\text{NAV}})C(x^{\text{LLA}}_{\text{NAV}})(z^{\text{LLA}}_{\text{NAV}} + R^2_a/R_{av,\text{NAV}}) \\ y^{\text{ECEF}}_{\text{NAV}} &= S(y^{\text{LLA}}_{\text{NAV}})C(x^{\text{LLA}}_{\text{NAV}})(z^{\text{LLA}}_{\text{NAV}} + R^2_a/R_{av,\text{NAV}}) \\ z^{\text{ECEF}}_{\text{NAV}} &= C(x^{\text{LLA}}_{\text{NAV}})(z^{\text{LLA}}_{\text{NAV}} + R^2_b/R_{av,\text{NAV}})\end{aligned} \tag{9}$$

and

$$\begin{aligned}x^{\text{ECEF}}_{\text{POI}} &= C(y^{\text{LLA}}_{\text{POI}})C(x^{\text{LLA}}_{\text{POI}})(z_{\text{POI}} + R^2_a/R_{av,\text{POI}}) \\ y^{\text{ECEF}}_{\text{POI}} &= S(y^{\text{LLA}}_{\text{POI}})C(x^{\text{LLA}}_{\text{POI}})(z_{\text{POI}} + R^2_a/R_{av,\text{POI}}) \\ z^{\text{ECEF}}_{\text{POI}} &= C(x^{\text{LLA}}_{\text{POI}})(z^{\text{LLA}}_{\text{POI}} + R^2_b/R_{av,\text{POI}})\end{aligned} \tag{10}$$

where R_a and R_b are the equatorial and polar radii, $R_{av,\text{NAV}} = \sqrt{R^2_a C(x^{\text{LLA}}_{\text{NAV}})^2 + R^2_b S(x^{\text{LLA}}_{\text{NAV}})^2}$, and $R_{av,\text{POI}} = \sqrt{R^2_a C(x^{\text{LLA}}_{\text{POI}})^2 + R^2_b S(x^{\text{LLA}}_{\text{POI}})^2}$.

The next step is to recognize that the UAV, C-POI, and POI must all lie in a straight line in any given coordinate system; this is shown in Fig. 1b as the CAM–POI vector. In ECEF coordinates, the projected C-POI in the camera image can be written as a function of a line in the following manner:

$$\begin{bmatrix}x^{\text{ECEF}}_{\text{C-POI}}\\ y^{\text{ECEF}}_{\text{C-POI}}\\ z^{\text{ECEF}}_{\text{C-POI}}\end{bmatrix} = \begin{bmatrix}x^{\text{ECEF}}_{\text{NAV}}\\ y^{\text{ECEF}}_{\text{NAV}}\\ z^{\text{ECEF}}_{\text{NAV}}\end{bmatrix} + u\left(\begin{bmatrix}x^{\text{ECEF}}_{\text{POI}}\\ y^{\text{ECEF}}_{\text{POI}}\\ z^{\text{ECEF}}_{\text{POI}}\end{bmatrix} - \begin{bmatrix}x^{\text{ECEF}}_{\text{NAV}}\\ y^{\text{ECEF}}_{\text{NAV}}\\ z^{\text{ECEF}}_{\text{NAV}}\end{bmatrix}\right) \tag{11}$$

where u is termed the path variable. The projected C-POI in ECEF coordinates can also be written by rotating the C-POI from the CAM axes to the ABC axes to the aircraft in local level East–North-up (ENU) axes and finally to the ECEF coordinate system, or

$$\begin{aligned}\begin{bmatrix}x^{\text{ECEF}}_{\text{C-POI}}\\ y^{\text{ECEF}}_{\text{C-POI}}\\ z^{\text{ECEF}}_{\text{C-POI}}\end{bmatrix} &= R^{\text{ECEF}}_{\text{ENU}} R^{\text{ENU}}_{\text{ABC}} R^{\text{ABC}}_{\text{CAM}} \begin{bmatrix}x^{\text{CAM}}_{\text{C-POI}}\\ y^{\text{CAM}}_{\text{C-POI}}\\ z^{\text{CAM}}_{\text{C-POI}}\end{bmatrix} + \begin{bmatrix}x^{\text{ECEF}}_{\text{NAV}}\\ y^{\text{ECEF}}_{\text{NAV}}\\ z^{\text{ECEF}}_{\text{NAV}}\end{bmatrix} \\ &= R^{\text{ECEF}}_{\text{CAM}} \begin{bmatrix}x^{\text{CAM}}_{\text{C-POI}}\\ y^{\text{CAM}}_{\text{C-POI}}\\ z^{\text{CAM}}_{\text{C-POI}}\end{bmatrix} + \begin{bmatrix}x^{\text{ECEF}}_{\text{NAV}}\\ y^{\text{ECEF}}_{\text{NAV}}\\ z^{\text{ECEF}}_{\text{NAV}}\end{bmatrix}\end{aligned} \tag{12}$$

where the following rotation matrices are defined:

$$R^{\text{ECEF}}_{\text{ENU}} = \begin{bmatrix}-S(y^{\text{LLA}}_{\text{NAV}}) & -S(x^{\text{LLA}}_{\text{NAV}})C(y^{\text{LLA}}_{\text{NAV}}) & C(x^{\text{LLA}}_{\text{NAV}})C(y^{\text{LLA}}_{\text{NAV}}) \\ C(y^{\text{LLA}}_{\text{NAV}}) & -S(x^{\text{LLA}}_{\text{NAV}})S(y^{\text{LLA}}_{\text{NAV}}) & C(x^{\text{LLA}}_{\text{NAV}})S(y^{\text{LLA}}_{\text{NAV}}) \\ 0 & C(x^{\text{LLA}}_{\text{NAV}}) & S(x^{\text{LLA}}_{\text{NAV}})\end{bmatrix}$$

$$R^{\text{ENU}}_{\text{ABC}} = \begin{bmatrix}S(\psi)C(\theta) & C(\phi)C(\psi) + S(\phi)S(\psi)S(\theta) & -S(\phi)C(\psi) + C(\phi)S(\phi)S(\theta) \\ C(\psi)C(\theta) & -C(\phi)S(\psi) + S(\phi)C(\psi)S(\theta) & S(\phi)S(\psi) + C(\phi)C(\psi)S(\theta) \\ S(\theta) & -S(\phi)C(\theta) & -C(\phi)C(\theta)\end{bmatrix}$$

$$R^{\text{ABC}}_{\text{CAM}} = \begin{bmatrix}C(p) & -S(p) & 0\\ S(p) & C(p) & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}C(t) & 0 & -S(t)\\ 0 & 1 & 0\\ S(t) & 0 & C(t)\end{bmatrix}\begin{bmatrix}C(s) & -S(s) & 0\\ S(s) & C(s) & 0\\ 0 & 0 & 1\end{bmatrix}$$

and $\mathbf{x}_{\text{ATT}} = [\phi, \theta, \psi]^T$ denotes the roll, pitch, and yaw of the aircraft, and $\mathbf{x}_{\text{GIM}} = [p, t, s]^T$ denotes the pan, tilt, and scan of the turret gimbal.

Setting Eqs. (11) and (12) equal to each other and multiplying each side by $(R^{\text{ECEF}}_{\text{CAM}})^{-1}$ then yields

$$R^{\text{CAM}}_{\text{ECEF}}\left(\begin{bmatrix}x^{\text{ECEF}}_{\text{POI}}\\ y^{\text{ECEF}}_{\text{POI}}\\ z^{\text{ECEF}}_{\text{POI}}\end{bmatrix} - \begin{bmatrix}x^{\text{ECEF}}_{\text{NAV}}\\ y^{\text{ECEF}}_{\text{NAV}}\\ z^{\text{ECEF}}_{\text{NAV}}\end{bmatrix}\right)u = \begin{bmatrix}x^{\text{CAM}}_{\text{C-POI}}\\ y^{\text{CAM}}_{\text{C-POI}}\\ z^{\text{CAM}}_{\text{C-POI}}\end{bmatrix} \tag{13}$$

In Fig. 1b, $x^{\text{CAM}}_{\text{C-POI}}$ is the linear distance between the camera and image plane, or screen. Because the undistorted full-scale image to the operator is not a requirement, this distance can be set arbitrarily. Defining $x^{\text{CAM}}_{\text{C-POI}} = 1$, the first row of Eq. (13) is used to find the path variable u:

$$u = \left[R^{\text{CAM}}_{\text{ECEF}}(1)\left(\begin{bmatrix}x^{\text{ECEF}}_{\text{POI}}\\ y^{\text{ECEF}}_{\text{POI}}\\ z^{\text{ECEF}}_{\text{POI}}\end{bmatrix} - \begin{bmatrix}x^{\text{ECEF}}_{\text{NAV}}\\ y^{\text{ECEF}}_{\text{NAV}}\\ z^{\text{ECEF}}_{\text{NAV}}\end{bmatrix}\right)\right]^{-1} \tag{14}$$

where $R(i)$ indicates the ith row of R.

Finally, as shown in Fig. 1b, the C-POI in the SCR coordinate system, or the measurement, can be written as a function of the C-POI in CAM coordinates as

$$\mathbf{y}^{\text{SCR}}_{\text{C-POI}} = \begin{bmatrix}x^{\text{SCR}}_{\text{C-POI}}\\ y^{\text{SCR}}_{\text{C-POI}}\end{bmatrix} = \begin{bmatrix}y^{\text{CAM}}_{\text{C-POI}}/\lambda_y\\ z^{\text{CAM}}_{\text{C-POI}}/\lambda_z\end{bmatrix} \tag{15}$$

where λ is the pixel length scale factor, which is written in terms of the camera field of view (F) and maximum pixels of the camera (p) in each axis:

$$\lambda_y = \frac{\tan(F_y/2)}{p^{\text{SCR}}_{y\text{-max}}}, \qquad \lambda_z = \frac{\tan(F_z/2)}{p^{\text{SCR}}_{z\text{-max}}}$$

The measurement output equation is then written in its final form by substituting $y^{\text{CAM}}_{\text{C-POI}}$ and $z^{\text{CAM}}_{\text{C-POI}}$ from the second and third rows of Eq. (13) into Eq. (15):

$$\mathbf{y}^{\text{SCR}}_{\text{C-POI}} = \begin{bmatrix} R^{\text{CAM}}_{\text{ECEF}}(2)/\lambda_y \\ R^{\text{CAM}}_{\text{ECEF}}(3)/\lambda_z \end{bmatrix} u\left(\begin{bmatrix} x^{\text{ECEF}}_{\text{POI}} \\ y^{\text{ECEF}}_{\text{POI}} \\ z^{\text{ECEF}}_{\text{POI}} \end{bmatrix} - \begin{bmatrix} x^{\text{ECEF}}_{\text{NAV}} \\ y^{\text{ECEF}}_{\text{NAV}} \\ z^{\text{ECEF}}_{\text{NAV}} \end{bmatrix} \right) \tag{16}$$

with LLA to ECEF conversions in Eqs. (9) and (10) and the path variable u from Eq. (14).

Square Root Sigma Point Filter with a Modular Update

Using the square root implementation of the sigma point filter developed in [25] as a basis, the estimator is extended to have a modular update step for reduced computation and complexity. The key development is the reduction of the prediction step to only include POI states and the use of statistics of the secondary state variables (NAV, ATT, GIM), known in real time from estimators and measurements onboard the aircraft in the update step.

The discrete dynamics and measurement output of the system are defined in Eqs. (7) and (8), with n_{POI} POI states, $n_{\text{C-POI}}$ measurements and sensor noises, and n_w disturbances. The disturbance and sensor noises are assumed to be zero-mean Gaussian processes with covariances of Q_{POI} and $R^{\text{SCR}}_{\text{C-POI}}$, respectively. An initial POI state and lower left triangular, square root factors of the covariance, $\mathbf{x}_{\text{POI},0}$, $S_{\text{POI},0}$, are also assumed.

An augmented state vector is defined first as

$$\mathbf{x}_{a,k} = \begin{bmatrix} \mathbf{x}_{\text{POI},k} \\ \mathbf{w}_{\text{POI},k} \end{bmatrix} \tag{17}$$

The dimension of the augmented state vector is $\mathbf{x}_{a,k} \in \mathbb{R}^{n_a}$, where $n_a = n_{\text{POI}} + n_w$. The initial augmented state estimate and square root covariance are assumed to be

$$\hat{\mathbf{x}}_{a,0} = E[\mathbf{x}_{a,0}] = \begin{bmatrix} \mathbf{x}_{\text{POI},0} \\ 0 \end{bmatrix}, \quad S_{a,0} = (E[(\mathbf{x}_{a,0} - \hat{\mathbf{x}}_{a,0})(\mathbf{x}_{a,0} - \hat{\mathbf{x}}_{a,k})^T])^{1/2} = \begin{bmatrix} S_{\text{POI},0} & 0 \\ 0 & \sqrt{Q_{\text{POI}}} \end{bmatrix} \tag{18}$$

Next, $2n_a + 1$ sigma points are defined:

$$\mathcal{X}_{a,0} = [\,\hat{\mathbf{x}}_{a,0} \quad \hat{\mathbf{x}}_{a,0} \cdot e_{n_a} + \sigma_f S_{a,0} \quad \hat{\mathbf{x}}_{a,0} \cdot e_{n_a} - \sigma_f S_{a,0}\,] = \begin{bmatrix} \mathcal{X}_{\text{POI},0} \\ \mathcal{X}_{w,0} \end{bmatrix} \tag{19}$$

where σ_f is a scaling for the distance of the sigma points from the mean, and e_{n_a} is a $n_a \times 1$ vector of all ones. A set of associated weights is then defined, which can be used to find the sample mean/covariance:

$$W^0_m = \frac{\sigma_f^2 - n_a}{\sigma_f^2}, \qquad W^0_c = \frac{\sigma_f^2 - n_a}{\sigma_f^2} + 3 - \frac{\sigma_f^2}{n_a}$$
$$W = W^i_m = W^i_c = \frac{1}{2\sigma_f^2}, \qquad i = 1, \cdots, 2n_a \tag{20}$$

where m and c denote mean and covariance, respectively.

The modular, square root SPF algorithm is now given using three steps: prediction, modular conversion, and update.

Modular Sigma Point Filter Prediction

The SPF prediction step propagates each of the $2n_a + 1$ sigma points through the nonlinear dynamics [Eq. (21)] and evaluates the sample mean [Eq. (22)] and predicted, centralized sigma points [Eq. (23)]. In the case presented here, the prediction step is based on the POI model given in Eq. (7), where $n_{\text{POI}} = 7$:

$$\mathcal{X}^{i-}_{\text{POI},k+1} = f^D_{\text{POI}}(\mathcal{X}^i_{\text{POI},k}, \mathcal{X}^i_{w,k}), \qquad i = 0, \ldots, 2n_a \tag{21}$$

$$\hat{\mathbf{x}}^{\,-}_{\text{POI},k+1} = \sum_{i=0}^{2n_a} W^i_m \mathcal{X}^{i-}_{\text{POI},k+1} \tag{22}$$

$$[\mathcal{X}^{c0-}_{\text{POI},k+1} \;\; \mathcal{X}^{c-}_{\text{POI},k+1}] = [\mathcal{X}^{0-}_{\text{POI},k+1} - \hat{\mathbf{x}}^-_{\text{POI},k+1} \;\; \mathcal{X}^{1-}_{\text{POI},k+1} - \hat{\mathbf{x}}^-_{\text{POI},k+1} \;\; \cdots \;\; \mathcal{X}^{2n_a-}_{\text{POI},k+1} - \hat{\mathbf{x}}^-_{\text{POI},k+1}] \tag{23}$$

Compared to the centralized case, which performs prediction calculations on $n_a = 18$ states (for $n_{\text{POI}} = 7$), this algorithm performs prediction calculations on only $n_{\text{POI}} = 7$ states.

Modular Sigma Point Filter Conversion

The modular update step requires additional information, namely, estimates and covariances for the NAV, ATT, and GIM states. Therefore, the state vector in Eq. (17) is further augmented to include these states, as

$$\mathbf{x}_{2a,k+1} = \begin{bmatrix} \mathbf{x}_{\text{POI},k+1} \\ \mathbf{v}^{\text{SCR}}_{\text{C-POI},k+1} \\ \mathbf{x}_{x2,k+1} \end{bmatrix} \tag{24}$$

where $\mathbf{x}_{x2,k+1} = [\mathbf{x}^T_{\text{NAV},k+1}, \mathbf{x}^T_{\text{ATT},k+1}, \mathbf{x}^T_{\text{GIM},k+1}]^T$ are secondary states that were not a part of the prediction step. The dimension of the secondary augmented vector is $\mathbf{x}_{2a,k+1} \in \mathbb{R}^{n_{2a} \times 1}$, where $n_{2a} = n_{\text{POI}} + n_{\text{C-POI}} + n_{x2}$ states.

In this application, as shown in Fig. 2, it is assumed that aircraft NAV and ATT estimators exist on the aircraft, yielding state estimate and square root covariance pairs, denoted as $(\hat{\mathbf{x}}_{\text{NAV},k+1}, S_{\text{NAV},k+1})$ and $(\hat{\mathbf{x}}_{\text{ATT},k+1}, S_{\text{ATT},k+1})$, respectively. The gimbal pointing angles are measured directly, yielding a measurement and square root noise covariance, denoted as $(\mathbf{y}_{\text{GIM},k+1}, \sqrt{R_{\text{GIM},k+1}})$. Therefore, the predicted state of the new augmented state vector is

$$\hat{\mathbf{x}}^{\,-}_{2a,k+1} = \begin{bmatrix} \hat{\mathbf{x}}^-_{\text{POI},k+1} \\ \hat{\mathbf{v}}^{\text{SCR},-}_{\text{C-POI},k+1} \\ \hat{\mathbf{x}}^-_{x2,k+1} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{x}}^-_{\text{POI},k+1} \\ 0 \\ \begin{bmatrix} \hat{\mathbf{x}}_{\text{NAV},k+1} \\ \hat{\mathbf{x}}_{\text{ATT},k+1} \\ \mathbf{y}_{\text{GIM},k+1} \end{bmatrix} \end{bmatrix} \tag{25}$$

The square root covariance associated with the secondary states is defined as

$$S^-_{x2,k+1} = \begin{bmatrix} S_{\text{NAV},k+1} & 0 & 0 \\ 0 & S_{\text{ATT},k+1} & 0 \\ 0 & 0 & \sqrt{R_{\text{GIM},k+1}} \end{bmatrix} \tag{26}$$

The centralized, predicted sigma points are now redefined from the first augmented state vector in Eq. (23) to a secondary augmented state vector as

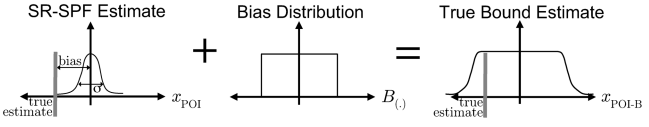

Fig. 3 Bias integration into the estimator to create accurate probability bounds.

$$\begin{bmatrix} \mathcal{X}^{c0-}_{\text{POI},k+1} & \mathcal{X}^{c-}_{\text{POI},k+1} \\ \mathcal{X}^{\text{SCR},c0-}_{\text{C-POI},k+1} & \mathcal{X}^{\text{SCR},c-}_{\text{C-POI},k+1} \\ \mathcal{X}^{c0-}_{x2,k+1} & \mathcal{X}^{c-}_{x2,k+1} \end{bmatrix} = \begin{bmatrix} \mathcal{X}^{c0-}_{\text{POI},k+1} & \mathcal{X}^{c-}_{\text{POI},k+1} & \mathcal{X}^{c0-}_{\text{POI},k+1} \cdot e_{2n_{\text{C-POI}}} & \mathcal{X}^{c0-}_{\text{POI},k+1} \cdot e_{2n_{x2}} \\ 0 & 0 & [+\sqrt{R^{\text{SCR}}_{\text{C-POI},k+1}}, -\sqrt{R^{\text{SCR}}_{\text{C-POI},k+1}}] & 0 \\ 0 & 0 & 0 & [+S^{-}_{x2,k+1}, -S^{-}_{x2,k+1}] \end{bmatrix} \tag{27}$$

Predicted sigma points for the secondary augmented state vector are calculated by adding the mean, or

$$\begin{bmatrix} \mathcal{X}^{-}_{\text{POI},k+1} \\ \mathcal{X}^{\text{SCR},-}_{\text{C-POI},k+1} \\ \mathcal{X}^{-}_{x2,k+1} \end{bmatrix} = \begin{bmatrix} \mathcal{X}^{c0-}_{\text{POI},k+1} + \hat{\mathbf{x}}^{-}_{\text{POI},k+1} & \mathcal{X}^{c-}_{\text{POI},k+1} + \hat{\mathbf{x}}^{-}_{\text{POI},k+1} \cdot e_{2n_{2a}} \\ \mathcal{X}^{\text{SCR},c0-}_{\text{C-POI},k+1} & \mathcal{X}^{\text{SCR},c-}_{\text{C-POI},k+1} \\ \mathcal{X}^{c0-}_{x2,k+1} + \hat{\mathbf{x}}^{-}_{x2,k+1} & \mathcal{X}^{c-}_{x2,k+1} + \hat{\mathbf{x}}^{-}_{x2,k+1} \cdot e_{2n_{2a}} \end{bmatrix} \tag{28}$$

Finally, the sigma point weights are updated to the augmented state vector with a larger number of states in the system. As shown in Eq. (20), only the central (0) sigma point weights must be updated:

$$W^0_m = \frac{\sigma_f^2 - n_{2a}}{\sigma_f^2}, \qquad W^0_c = \frac{\sigma_f^2 - n_{2a}}{\sigma_f^2} + 3 - \frac{\sigma_f^2}{n_{2a}} \tag{29}$$

This step primarily repackages the estimator and data into an appropriate form for the modular update.

Modular Sigma Point Filter Update

The modular SPF update step computes the Kalman gain, updated mean [Eq. (33)], and updated, square root covariance [Eq. (34)] for the original n_{POI} states *only* [Eq. (32)]. First, the predicted output sigma points are calculated, along with the mean predicted output:

$$\mathcal{Y}^{\text{SCR},i-}_{\text{C-POI},k+1} = g_{\text{C-POI}}(\mathcal{X}^{i-}_{\text{POI},k+1}, \mathcal{X}^{i-}_{x2,k+1}, \mathcal{X}^{\text{SCR},i-}_{\text{C-POI},k+1})$$
$$i = 0, \cdots, 2n_{2a} \qquad \hat{\mathbf{y}}^{\text{SCR},-}_{\text{C-POI},k+1} = \sum_{i=0}^{2n_{2a}} W^i_m \mathcal{Y}^{\text{SCR},i-}_{\text{C-POI},k+1} \tag{30}$$

The centralized output sigma points can then be defined as

$$\begin{bmatrix} \mathcal{Y}^{\text{SCR},c0-}_{\text{C-POI},k+1} & \mathcal{Y}^{\text{SCR},c-}_{\text{C-POI},k+1} \end{bmatrix} = \begin{bmatrix} \mathcal{Y}^{\text{SCR},0-}_{\text{C-POI},k+1} & -\hat{\mathbf{y}}^{\text{SCR},-}_{\text{C-POI},k+1} & \mathcal{Y}^{\text{SCR},1-}_{\text{C-POI},k+1} & -\hat{\mathbf{y}}^{\text{SCR},-}_{\text{C-POI},k+1} & \cdots & \mathcal{Y}^{\text{SCR},2n_{2a}-}_{\text{C-POI},k+1} & -\hat{\mathbf{y}}^{\text{SCR},-}_{\text{C-POI},k+1} \end{bmatrix} \tag{31}$$

The Kalman gain is calculated for the POI states *only* by using the centralized, predicted state and output sigma points:

$$K_{\text{POI},k+1} = \left[(\mathcal{X}^{c-}_{\text{POI},k+1})(\mathcal{Y}^{\text{SCR},c-}_{\text{C-POI},k+1})^T + \frac{W^0_c}{W} (\mathcal{X}^{c0-}_{\text{POI},k+1})(\mathcal{Y}^{\text{SCR},c0-}_{\text{C-POI},k+1})^T \right] \left[(\mathcal{Y}^{\text{SCR},c-}_{\text{C-POI},k+1})(\mathcal{Y}^{\text{SCR},c-}_{\text{C-POI},k+1})^T + \frac{W^0_c}{W} (\mathcal{Y}^{\text{SCR},c0-}_{\text{C-POI},k+1})(\mathcal{Y}^{\text{SCR},c0-}_{\text{C-POI},k+1})^T \right]^{-1} \tag{32}$$

Note that the modular Kalman gain is $n_{\text{POI}} \times n_{\text{C-POI}}$ rather than $n_a \times n_{\text{C-POI}}$ in the centralized case.

The updated state estimate and square root covariance are then given as

$$\hat{\mathbf{x}}_{\text{POI},k+1} = \hat{\mathbf{x}}^{-}_{\text{POI},k+1} + K_{\text{POI},k+1}(\mathbf{y}^{\text{SCR}}_{\text{C-POI},k+1} - \hat{\mathbf{y}}^{\text{SCR},-}_{\text{C-POI},k+1}) \tag{33}$$

$$S_{\text{POI},k+1} = \text{up}\{\sqrt{W} \cdot \text{orth}\{(\mathcal{X}^{c-}_{\text{POI},k+1} - K_{\text{POI},k+1}\mathcal{Y}^{\text{SCR},c-}_{\text{C-POI},k+1})\}, (\mathcal{X}^{c0-}_{\text{POI},k+1} - K_{\text{POI},k+1}\mathcal{Y}^{\text{SCR},c0-}_{\text{C-POI},k+1})\sqrt{|W^0_c|}, \text{sgn}(W^0_c)\} \tag{34}$$

Note that the QR decomposition, which is the most costly computation in the filter, is performed on a $n_{\text{POI}} \times (2n_{2a} + 1)$ matrix rather than on a $n_{2a} \times (2n_{2a} + 1)$ matrix in the centralized case.

Finally, the augmented state and square root covariance and the augmented sigma points are recalculated:

$$\hat{\mathbf{x}}_{a,k+1} = \begin{bmatrix} \hat{\mathbf{x}}_{\text{POI},k+1} \\ 0 \end{bmatrix}, \qquad S_{a,k+1} = \begin{bmatrix} S_{\text{POI},k+1} & 0 \\ 0 & \sqrt{Q_{\text{POI}}} \end{bmatrix} \tag{35}$$

$$\mathcal{X}_{a,k+1} = [\hat{\mathbf{x}}_{a,k+1} \quad \hat{\mathbf{x}}_{a,k+1} \cdot e_{n_a} + \sigma_f S_{a,k+1} \quad \hat{\mathbf{x}}_{a,k+1} \cdot e_{n_a} - \sigma_f S_{a,k+1}] = \begin{bmatrix} \mathcal{X}_{\text{POI},k+1} \\ \mathcal{X}_{w,k+1} \end{bmatrix} \tag{36}$$

and the process repeats.

Shipset Estimates: Estimation with Bias Uncertainties

The bias errors typically cannot be estimated online without a calibration maneuver at the start of each flight and a specialized truth measure for the general moving POI case. The biases could be estimated for the stationary POI case because their assumed dynamics (typically constants) can be differentiated from that of the POI states [27]. For the moving POI case, however, the bias states are nearly unobservable; physically, the estimator cannot tell the difference between a bias and a POI movement. Because the biases are not observable in the data, they translate directly through the estimator, adding a bias error in the tracking performance of the estimator.

The bias errors are not known a priori, as they are typically caused by mounting uncertainties or attitude estimation errors. Experience in evaluating these uncertainties across a set of UAVs by the Insitu Group has shown that these uncertainties are not well modeled as a Gaussian distribution. For example, a 0 deg mounting bias is no more likely than a 1 deg bias. For these reasons, the unobservable biases are modeled here as uniform random variables.

With the definition of the bias errors as uniform random variables that are unobservable to the sensor measurements, the approach here is to statistically fuse the geolocation estimates with the uniform bias density model after each cycle of the estimator. The hypothesis is that this fused density model, although not Gaussian, is a better representation of the estimator's accuracy across all UAVs in a single shipset. A conceptual view of this is shown in Fig. 3.

More specifically, the a posteriori state estimate of the SR-SPF geolocation estimator is normal and will be fused with uniform density representations for all biases. It is known from the central limit theorem that when many uniform bias errors exist, the summation of these random variables approaches a Gaussian density

[26]. However, in this application, the output density can be dominated by one, or a few, of the biases, thus requiring a more formal treatment of the fusion process.

To start, it is recognized that the set of measurement equations [Eqs. (9–15)] can be rewritten in the form

$$\mathbf{x}_{\text{POI}} = g_{\text{POI}}\left(\mathbf{y}^{\text{SCR}}_{\text{C-POI}}, \begin{bmatrix}\mathbf{x}_{\text{NAV}}\\ \mathbf{x}_{\text{ATT}}\\ \mathbf{x}_{\text{GIM}}\end{bmatrix}, \mathbf{v}^{\text{SCR}}_{\text{C-POI}}\right) \tag{37}$$

This equation can be expanded about the current SR-SPF estimate of the POI, along with the NAV and ATT estimates, and GIM measurement, to yield

$$[\mathbf{x}_{\text{POI}}]_{\text{SHIP}} = [\mathbf{x}_{\text{POI}}]_{\text{SPF}} + \left.\frac{\partial g_{\text{POI}}}{\partial \mathbf{x}_{\text{NAV}}}\right|_{\mathbf{x}_{\text{NAV}}=\hat{\mathbf{x}}_{\text{NAV}}} \cdot \delta\mathbf{x}_{\text{NAV-B}} + \left.\frac{\partial g_{\text{POI}}}{\partial \mathbf{x}_{\text{ATT}}}\right|_{\mathbf{x}_{\text{ATT}}=\hat{\mathbf{x}}_{\text{ATT}}} \cdot \delta\mathbf{x}_{\text{ATT-B}} + \left.\frac{\partial g_{\text{POI}}}{\partial \mathbf{x}_{\text{GIM}}}\right|_{\mathbf{x}_{\text{GIM}}=\mathbf{y}_{\text{GIM}}} \cdot \delta\mathbf{x}_{\text{GIM-B}} \tag{38}$$

where $[\mathbf{x}_{\text{POI}}]_{\text{SPF}} \sim \mathcal{N}(\hat{\mathbf{x}}_{\text{POI}}, P_{\text{POI}})$ is the output of the SR-SPF geolocation estimator, $P_{\text{POI}} = S^T_{\text{POI}} S_{\text{POI}}$ is the POI covariance, and $\delta\mathbf{x}_{(\cdot)\text{-B}}$ are unobservable bias uncertainties modeled as zero-mean, uniform distributions, $\delta\mathbf{x}_{(\cdot)\text{-B}} \sim \mathcal{U}(\mathbf{0}, B_{(\cdot)})$, where the variable ranges from $-B_{(\cdot)}$ to $+B_{(\cdot)}$ with a probability of $1/(2B_{(\cdot)})$.

The multiplicative factors on the perturbation terms in Eq. (39) are linearized matrices in an output equation, or

$$C_{\text{POI},(\cdot)} = \left.\frac{\partial g_{\text{POI}}}{\partial(\cdot)}\right|_{(\cdot)\in\{\hat{\mathbf{x}}_{\text{POI}}, \hat{\mathbf{x}}_{\text{NAV}}, \hat{\mathbf{x}}_{\text{ATT}}, \mathbf{y}_{\text{GIM}}\}} \tag{39}$$

The linearized output matrix can be rewritten as a function of the cross and full covariance matrices, which could then further be defined in terms of the current, centralized sigma points from Eq. (27). Dropping the time index k for clarity, this is written as

$$C_{\text{POI},(\cdot)} = [P^-_{(\cdot),\text{POI}}]^T P^{-1}_{(\cdot),(\cdot)} = \left[\mathcal{X}^c_{\text{POI}}(\mathcal{X}^c_{(\cdot)})^T + \frac{W^c_0}{W}\mathcal{X}^{c0}_{\text{POI}}(\mathcal{X}^{c0}_{(\cdot)})^T\right] \cdot \left[\mathcal{X}^c_{(\cdot)}(\mathcal{X}^c_{(\cdot)})^T + \frac{W^c_0}{W}\mathcal{X}^{c0}_{(\cdot)}(\mathcal{X}^{c0}_{(\cdot)})^T\right]^{-1} \tag{40}$$

Equation (39) then simplifies to

$$[\mathbf{x}_{\text{POI}}]_{\text{SHIP}} = [\mathbf{x}_{\text{POI}}]_{\text{SPF}} + C_{\text{POI,NAV}} \cdot \delta\mathbf{x}_{\text{NAV-B}} + C_{\text{POI,ATT}} \cdot \delta\mathbf{x}_{\text{ATT-B}} + C_{\text{POI,GIM}} \cdot \delta\mathbf{x}_{\text{GIM-B}} \tag{41}$$

Equation (41) presents the POI state as a vector of n_{POI} random variables for each degree of freedom. Each variable is a single Gaussian random variable, added to a summation of m uniform random variables; $m = 9$ in this case. For the jth POI state, this is written as

$$[\mathbf{x}^j_{\text{POI}}]_{\text{SHIP}} = [\mathbf{x}^j_{\text{POI}}]_{\text{SPF}} + \sum_{\ell=1}^{m} C^{j,\ell}_{\text{POI}} \delta x^\ell_{\text{B}} \tag{42}$$

where $C^{j,\ell}_{\text{POI}}$ and δx^ℓ_{B} are the appropriate entries of $C_{\text{POI},(\cdot)}$ and $\delta\mathbf{x}_{(\cdot)\text{-B}}$.

The shipset density can be solved formally by using the characteristic functions of each random variable. For a zero-mean, Gaussian random variable with standard deviation σ, the characteristic function is given as

$$\Phi_G(\sigma^2, \omega) = e^{-\sigma^2\omega^2/2} \tag{43}$$

where, for the jth POI random variable, the standard deviation is found from the jth diagonal element of the covariance matrix, or $\sigma = \sqrt{P^{j,j}_{\text{POI}}}$. Similarly, for a zero-mean, uniform random variable with a width of $\pm b$, the characteristic function is

$$\Phi_U(b, \omega) = \frac{1}{b\omega}\sin(b\omega) \tag{44}$$

where, for the ℓth bias, the bound is given as $b = B_\ell$.

The characteristic equation of the total (tracking + bias) density from Eq. (42) is then simply a multiplication of all individual characteristic equations, or

$$\Phi^j_{\text{SHIP}} = \Phi_G(P^{j,j}_{\text{POI}}, \omega)\prod_{\ell=1}^{m} C^{j,\ell}_{\text{POI}}\Phi_U(B_\ell, \omega) \tag{45}$$

The shipset (tracking + bias) density can then be recovered by taking the inverse Fourier transform, or

$$[\mathbf{x}^j_{\text{POI}}]_{\text{SHIP}} \sim \mathcal{F}^{-1}\left(\Phi_G(P^{j,j}_{\text{POI}}, \omega)\prod_{\ell=1}^{m} C^{j,\ell}_{\text{POI}}\Phi_U(B_\ell, \omega)\right) \tag{46}$$

Solving Eq. (46) for the distribution of $[\mathbf{x}^j_{\text{POI}}]_{\text{SHIP}}$ results in a summation of weighted error functions (erf: twice the integral of the Gaussian distribution). In the SeaScan application presented here, the distribution is typically dominated by a Gaussian distribution and one uniform distribution. In this case, given the jth POI state modeled as a Gaussian variable and $m = 1$ uniform bias variable, the total density is given as

$$h([\mathbf{x}^j_{\text{POI}}]_{\text{SHIP}}) \approx \frac{-1}{4B_1}\text{erf}\left(\frac{[\mathbf{x}^j_{\text{POI}}]_{\text{SHIP}} - B_1}{\sqrt{2}P^{j,j}_{\text{POI}}}\right) + \frac{1}{4B_1}\text{erf}\left(\frac{[\mathbf{x}_{j,\text{POI}}]_{\text{SHIP}} + B_1}{\sqrt{2}P^{j,j}_{\text{POI}}}\right) \tag{47}$$

The shipset confidence bound is then found by calculating the state $[\mathbf{x}^j_{\text{POI}}]_{\text{SHIP}}$, where the probability is given by $\mathcal{P}_{\text{conf}}$. This is written as

$$\mathcal{P}_{\text{conf}} = \int_{-[\mathbf{x}^j_{\text{POI}}]_{\text{SHIP}}}^{+[\mathbf{x}^j_{\text{POI}}]_{\text{SHIP}}} h([\mathbf{x}^j_{\text{POI}}]_{\text{SHIP}})\, \text{d}[\mathbf{x}^j_{\text{POI}}]_{\text{SHIP}} \tag{48}$$

Although there is no closed form for this integral, it can be calculated very quickly numerically and, therefore, can be used for online calculation of POI state estimate confidence bounds.

Seascan Uninhabited Aerial Vehicle

The SeaScan is a long-endurance (24 + h) UAV developed by the Insitu Group. The UAV has an SE555 processor board to provide inner-loop control and data management. Flight-path characteristics can be determined from preprogrammed or in-flight commands, and include constant radius/altitude orbits, bowtie patterns, and way point following (from the ground, onboard memory, or a payload processor). A common operational approach for geolocation is to have the operator select an altitude and use a joystick to center an orbit over the POI. If the POI is moving, the operator then continues to update the UAV orbit center.

The air-to-ground communication is a data link used to communicate aircraft status, control, and mission data, as well as to relay messages from payload modules. The avionics-to-payload communication is a data link used to send sensor reports to the payload and receive commands from the payload. The onboard sensors include roll, pitch, and yaw rate gyros; vertical, lateral, and longitudinal accelerometers; external temperature sensors; relative pressures of pitot, alpha, beta, and gamma for wind axes estimation; and absolute pressures of barometric and manifold.

The SE555 board communicates serially with an onboard video turret, issuing camera positioning commands as well as aircraft attitude and stabilization data. The video signal bypasses the SE555 board and is sent directly to a ground receiver via an onboard radio frequency link. The SE555 board also communicates serially with an onboard GPS receiver, receiving differential GPS data from the ground to improve position and velocity solutions.

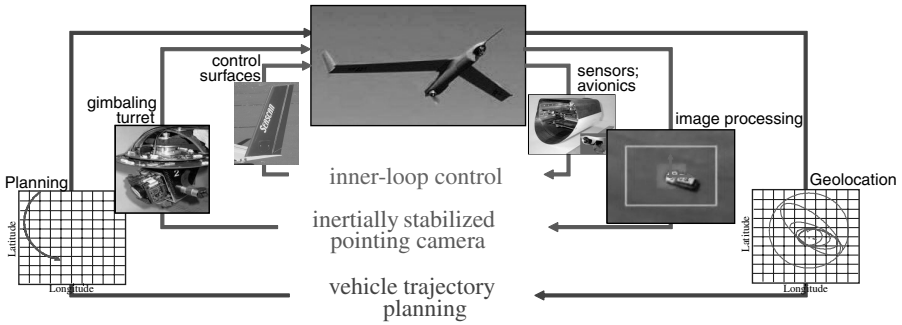

Fig. 4 Control loops for geolocation using the SeaScan UAV; hardware components of the SeaScan UAV are also shown.

Camera and Image Software

The SeaScan contains a digital video camera integrated into an inertially stabilized pan/tilt nose turret. The camera has an acuity ∼50% better than that of the unaided eye at the telescopic end, and it can resolve POIs such as small boats and logs from 5 miles away. The operator can command the camera to pan back and forth for wide-area search or to remain locked onto a POI while the aircraft maneuvers; the latter mode is used here.

Ground software processes the images from the camera. When the user selects a POI for geolocation, the gimbaling turret and ground software attempt to maintain the POI in the center of the frame, from frame to frame. Therefore, the "measurement" of the POI is assumed to be at the center of the image frame, as shown in Fig. 4.

Hierarchical Control Loops of the SeaScan

To maintain robustness in UAV/turret/tracking components for geolocation, the measurement/estimation and control/planning software for the SeaScan has been decomposed into three loops, as shown in Fig. 4. The inner-loop control is typical of an aircraft: the onboard sensors and avionics are used to calculate required control surface deflections of the vehicle. A reference trajectory (way points based on UAV orbits, other patterns) is used to guide this control loop.

The middle loop inertially stabilizes the turret/gimballing camera. Because variations from image to image are very fast compared to the geolocation (position) tracking, it is more robust to decouple these loops. Insitu has several approaches for "sensing" from the image in this loop, including pixel tracking [28], blob and particle filter tracking [29], and optical flow [30]. In this experiment, pixel tracking was used, in which a subset of the pixels from one frame to the next are matched to calculate the shift in the image. With this calculation, along with an estimate of the time delay for this calculation, the axes of the turret are then commanded to keep the C-POI within the image center as well as reject disturbances (such as from engine vibration).

The outer loop is the planning loop, in which the SeaScan plans its orbit (or way points) to improve geolocation performance. This is a slower control loop than the inertially stabilized camera and can be coupled to other vehicles or to operator commands. During the experiments described here, only orbit commands were used and operators continually adjusted the center of the UAV orbit to maintain its location over the target. There are a variety of factors in UAV planning that could be used to improve geolocation performance; an overview of some of these factors along with experimental results with respect to the SeaScan is given in [31,32].

Geolocation Uncertainties

Uncertainties in geolocation can be attributed to many factors. This is most easily seen by examining the sensor measurement, Eq. (1), in which the measured screen coordinates of the POI are a nonlinear function of the UAV position and attitude, turret attitude, and image processing. Uncertainties in any/all of these can affect the uncertainty in the tracking estimator. In addition, these uncertainties can arise during the mission, from mission to mission, or from UAV to UAV, especially in production UAVs.

Because the SeaScan and ScanEagle from the Insitu Group are production UAVs, numerous calibration and flight tests have led to an understanding of the categories of uncertainties within this family of UAVs. Table 1 shows a summary of the uncertainty sources, along with the type of uncertainty (process noise, sensor noise, or bias). To quantify these uncertainties, especially those with bias characteristics, a series of tests were performed. Examples of tests include ground calibration of many UAVs, flight testing using video and known ground features (and accurate differential GPS), and wind-tunnel tests.

Uncertainties with sensor noises are nicely approximated as zero mean and Gaussian, and can typically be quantified/verified through calibration and statistical analyses. Similarly, process noises such as engine vibration and turbulence can be estimated through wind-tunnel and flight tests. Sensor biases and time delays, some of which manifest themselves as a bias and are therefore modeled here similarly, are more challenging. These biases are caused by several sources, such as using nonideal sensors, mounting alignment errors, image processing delays on the ground, and nonideal attitude and navigation estimation hardware and software. In addition, these factors can also vary from UAV to UAV.

Some biases have been shown by the Insitu Group to be nearly constant for a single UAV, such as alignment and time delays, whereas others have been shown to be slowly varying, such as attitude and navigation biases. The constant biases could be evaluated in-flight through a calibration maneuver, or on the ground. But, these tests are challenging because the constant biases would have to be clearly delineated from the slowly time-varying biases. Finally, tests performed by the Insitu Group have shown a distribution of bias uncertainties across multiple UAVs to be non-Gaussian, that is, there is not one very likely estimate of the biases. Therefore, a uniform distribution of the biases is assumed. Because of International Traffic in Arms Regulations (ITAR) restrictions, the true values of these statistics unfortunately cannot be published. However, as an intuitive

Table 1 Uncertainties in the geolocation problem

	System variables affected			Type of uncertainty		
Source	NAV	ATT	GIM	Process noise	Sensor noise	Sensor bias
Aircraft avionics	—	√	—	—	√	√
Turret gyro	—	—	√	—	√	√
GPS	√	—	—	—	√	√
Turbulence	—	√	—	√	—	—
Engine vibrations	—	√	—	√	—	—
Camera alignment	—	—	√	—	—	√
Time delays	√	√	√	—	√	√

example, consider a 1 deg bias in pointing (either from the gimbal or aircraft) and a UAV flying in a 500 m orbit at an altitude of 500 m. In this case, the constant bias will cause a 12 m error in the POI estimate, which oscillates between the two axes of the ground plane. Larger (and more) biases scale this error higher.

Experimental Flight-Test Results Using the Seascan UAV

A set of flight tests were performed on 6 October 2005. A ground "truth" for the POI, denoted as $\mathbf{x}_{\text{TRU}}$, was set up using a GPS antenna and receiver in a car near the flight-test range. Over 2 h of flight tests were recorded for both stationary and moving POIs. Telemetry data were saved for the aircraft and the camera data, as well as the truth. The following is a set of estimation results based on this flight-test data. Figures plot the estimation errors, denoted as $\mathbf{e}_{\text{SPF}} = \hat{\mathbf{x}} - \mathbf{x}_{\text{TRU}}$; thus, an error of zero indicates an estimator performing very close to the truth. Also in the figures, "SPF 2σ bound" refers to the $\mathcal{P} = 0.95$ probability bound, as defined from the square root error covariance derived from Eq. (34); "SPF 2σ bound + bias" refers to the $\mathcal{P} = 0.95$ probability bound, as defined from the shipset distribution and probability calculations in Eqs. (46) and (48). It is noted that the specific performance data are left off due to ITAR restrictions, but the implementation and relative comparisons are still clear.

Stationary Point of Interest: Two-Dimensional Estimator

Figure 5 shows the tracking two-dimensional (latitude and longitude) estimates for a stationary POI using the developed geolocation estimator and a terrain altitude model, that is, the POI altitude is assumed to be measured directly. The standard deviation of the estimation error over one full orbit was $0.29L$. For the SPF bound case, the average bound width was $0.65L$, and the true POI location stays within the bounds only 28.9% of the time. For the SPF bound + bias case, the average bound width was $4.5L$, and the true POI location stays within the bounds 95.8% of the time.

The following observations can be made. First, the true POI location stays within the bounds as defined by the SPF bound + bias estimator 95.8% of the time, which is consistent with the estimator and density models. If only the SPF bounds are used, the true POI location stays within the bounds only 28.9% of the time, thus yielding inconsistent results. However, the estimates would be consistent within the SR-SPF framework because the biases are unobservable. Second, whereas the SPF bounds remain small and relatively constant, the bounds in the SPF bound + bias estimator are much larger and vary by nearly an order of magnitude over the 2 min data set. These bounds become their largest at $\approx$40 s and smallest after $\approx$80 s, a factor of $\approx$5 difference. Third, although not shown, a fully centralized geolocation estimator was also implemented using aircraft navigation models, thus evaluating the correlations that are not captured in the proposed modular estimator. The performance for the centralized estimator is nearly identical to the corresponding plots, with an error standard deviation from the modular case of $0.0012L$, which is less than 1% of the tracking error. Thus, this shows that the decentralized implementation is adequate, and performance is dominated more by sensor uncertainties than a time-based correlation of UAV states. Fourth, these results show that the true location always stays inside the largest bounds, but the bias error is considerable. Finally, Fig. 5 also shows the real-time data, which uses geometry and an assumed POI altitude to estimate the POI location; this estimate has a large, time-varying error below 40 s, which diminishes to a much smaller error between 80 and 100 s.

To understand the time-varying error and bounds, consider Fig. 6, which shows a plot of the UAV trajectory between times of 40 and 80 s, along with several snapshots of the two-dimensional location and 95% uncertainty bounds (the initial uncertainty ellipse is also shown). Early in the trajectory, the UAV is further (in two dimensions) from the POI, and the ellipse is elongated in the line of site direction because of a lack of observability with the vision sensor. Near the end of the trajectory, the UAV is closer to the POI (in two dimensions), and the uncertainty ellipsoid has shrunk further and in both directions. Thus, a large reason that the tracking error and uncertainty ellipsoids shrink in the 80 + s range is due to the UAV being at a closer proximity to the POI (in two dimensions). A second reason is that, while the errors will vary as a function of the range between the UAV and POI, they also vary across the latitudinal and longitudinal directions as the UAV orbits about the POI. Thus, even for an orbit that is centered about the POI, the geolocation estimator errors in latitude and longitude will vary over time.

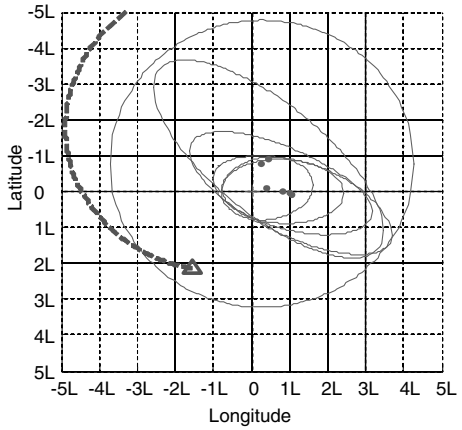

Fig. 6 UAV trajectory (dashed line ending in the triangle) and evolution of the two-dimensional uncertainty ellipses and estimates from the two-dimensional geolocation estimator (centered dots and shrinking ellipses); corresponds to the time window of 40–80 s in Fig. 5.

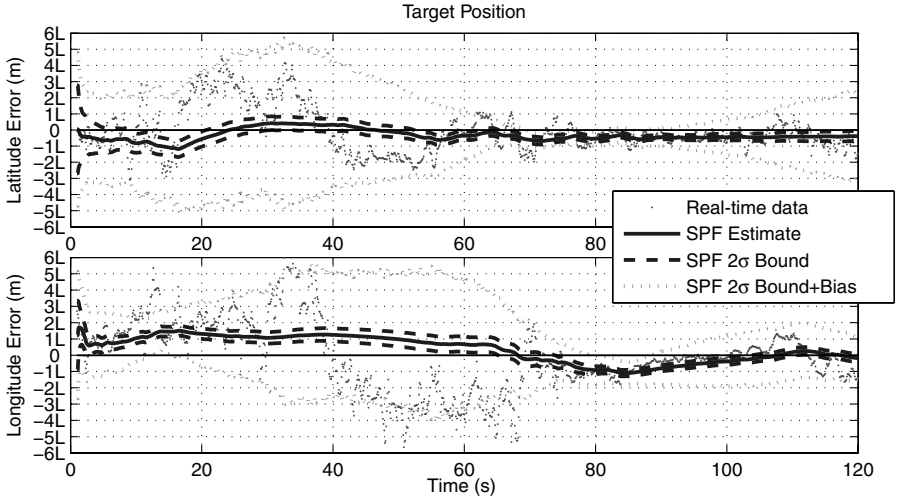

Fig. 5 Geolocation estimator (two-dimensional) performance for a stationary POI; POI altitude is assumed to be known.

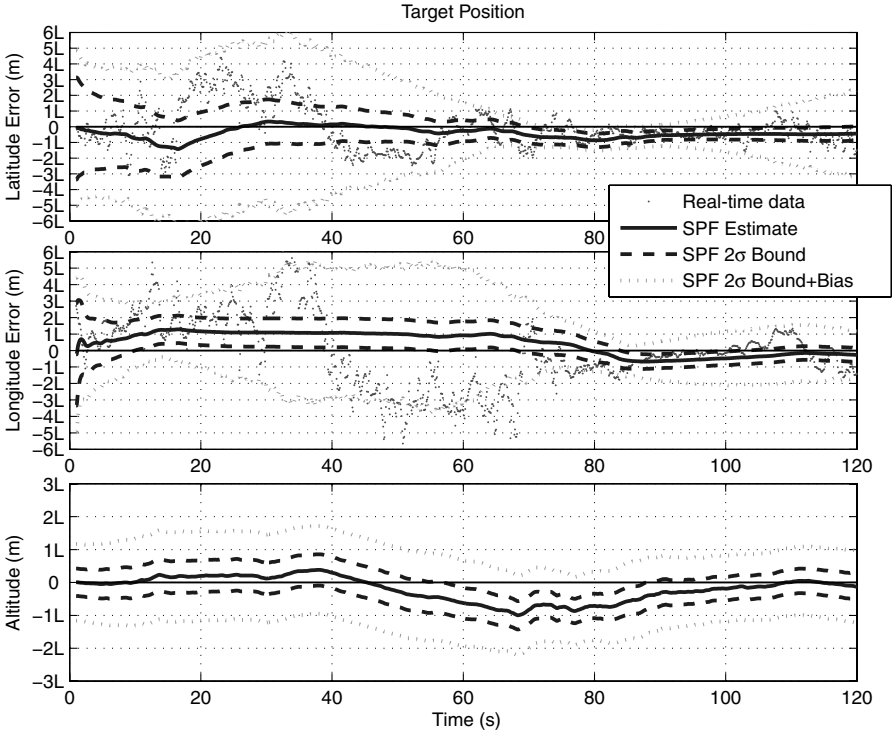

Fig. 7 Geolocation estimator (three-dimensional) performance for a stationary POI; POI altitude is estimated.

Stationary Point of Interest: Three-Dimensional Estimator

The second case considered is identical to that of the previous section, but now the POI altitude is estimated rather than measured from a terrain model. This is a particularly challenging estimation problem because if both the UAV and POI were stationary, it would be impossible to estimate the three-dimensional location of the POI because the system is not observable. The estimation of all three states of the POI can only be done over time, as the UAV moves for more viewing angles.

Figure 7 shows the estimation results for the three-dimensional geolocation estimator with altitude estimation. The standard deviation of the error over one full orbit was $0.36L$. For the SPF bound case, the average bound width was $1.4L$, and the true POI location stays within the bounds only 28.9% of the time. For the SPF bound + bias bound estimator, the average bound width was $4.9L$, and the true POI location stays within the bounds 95.4% of the time.

The characteristics of the estimator are similar to the two-dimensional version, including time-varying bounds roughly correlated with relative proximity between the UAV and POI. In the <20 s range, the performance of the three-dimensional estimator is slightly worse due to the lack of observability. In the $80+$ s range, the accuracy of the tracking and bounds are the best. The SPF bounds are clearly larger for the three-dimensional case, as there is an additional uncertain variable to estimate (POI altitude). Specifically, during this 2 min data set, the SPF bounds are 2.1 times larger in the in the three-dimensional case, whereas the bounds for the SPF bound + bias case are only 1.1 times larger in the three-dimensional case.

Moving Point of Interest: Two-Dimensional Estimator

The structure of the estimator (one piece of code with a configuration file, separate files for different models, user inputs) enables the easy integration of other options. One such option is to geolocate a moving POI, which is implemented by using a different model in the estimator. The case presented here is a POI moving at a constant heading and constant velocity, which is followed by decelerate, stop,

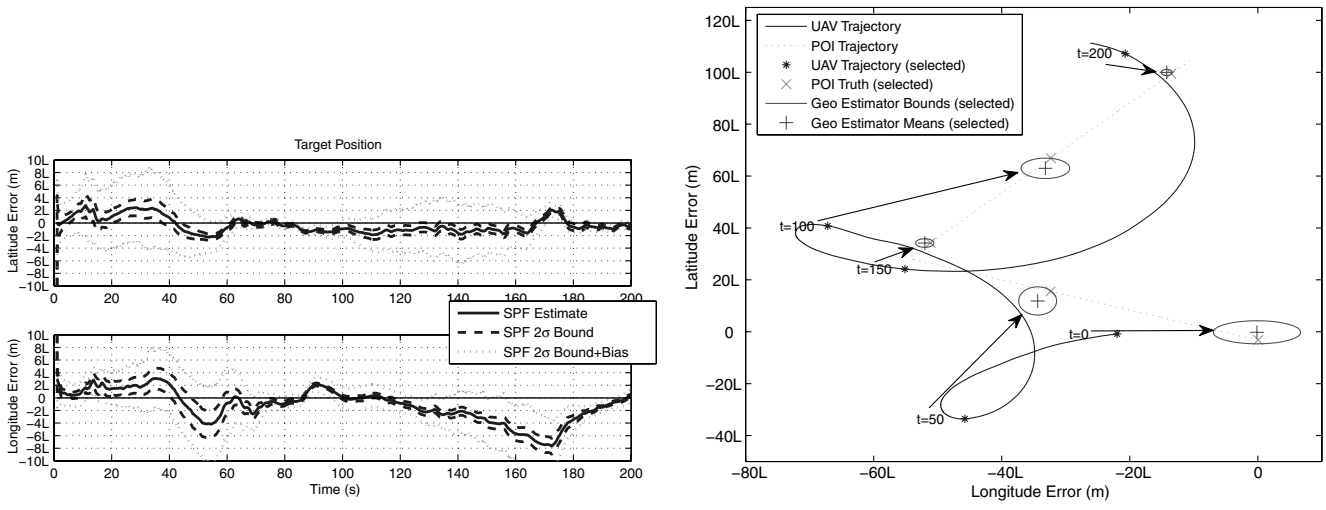

Fig. 8 Geolocation estimator (two dimension) for a moving POI with the following motion: constant velocity moving straight, decelerate, stop, turn right, accelerate, constant velocity moving straight: a) POI latitude and longitude estimation, where POI altitude is assumed to be known; and b) UAV and POI trajectory plot, with several snapshots showing performance and evolution of the results.

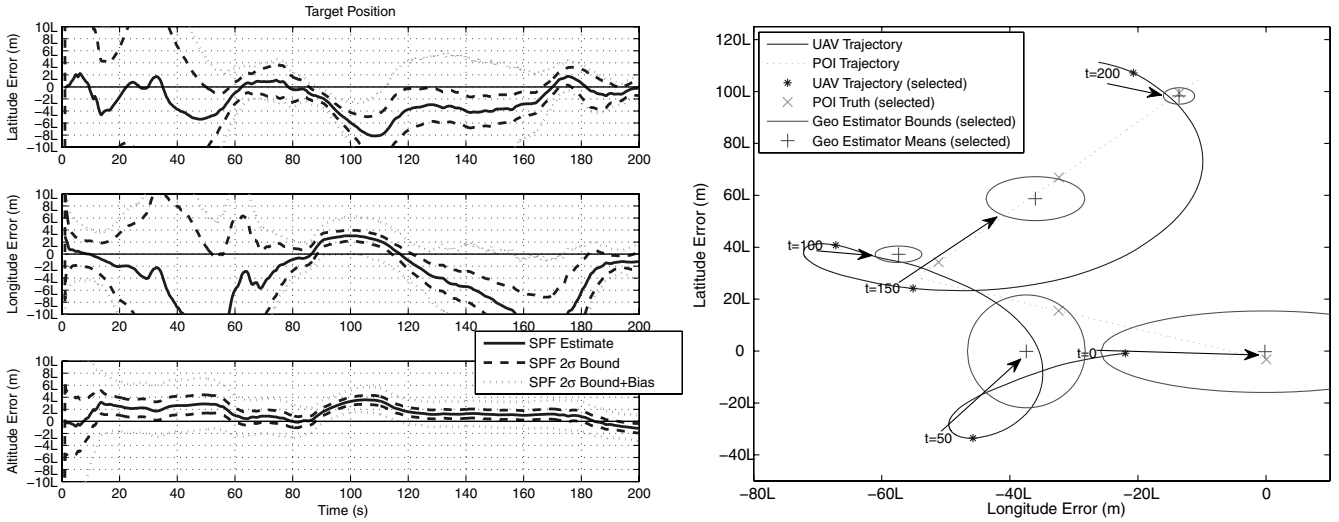

Fig. 9 Geolocation estimator (three dimensions) for a moving POI with the following motion: constant velocity moving straight, decelerate, stop, turn right, accelerate, constant velocity moving straight: a) POI latitude, longitude, and altitude estimation; and b) UAV and POI trajectory plot, with several snapshots showing performance and evolution of the results.

turn right, accelerate, constant velocity, and constant heading. Experimentally, the UAV orbit center was updated by the operator as the POI moved. The POI was maintained within the camera field of view during the experiment; thus, the specific "control" of the UAV was not a factor (only knowledge of the UAV state was a factor in geolocation). In this case, a seven-state POI model was used to capture more complex dynamics of the POI, including turns and acceleration/deceleration.

Figure 8 shows the results for the geolocation estimator and a moving POI, when a POI altitude model is assumed. Figure 8a shows the latitude and longitude tracking results, quoted in terms of an error from the true moving POI location. The standard deviation of the error over one full orbit was $1.64L$. For the SPF bound case, the average bound width was $1.74L$, and the true POI location stays within the bounds only 24.4% of the time. For the SPF bound + bias case, the average bound width was $5.5L$, and the true POI location stays within the bounds 95.1% of the time.

Comparing with the stationary case (Fig. 5), the geolocation results are slightly worse for the moving POI case. This is intuitively correct because there are more states to estimate (seven), and the process noise covariance is larger to be able to track changes in the moving POI. In the 0–40 s range, the estimates are larger than at later times. This is a result of the relative proximity of the UAV to the POI (similar to the stationary POI case), but also because of inaccuracies in the initial guess. The initial velocity was assumed to be zero (stationary, because of lack of information), but, in fact, the initial velocity was much faster. Note that the two-dimensional geolocation estimator also performs quite well even during the deceleration and turn, which occurred near the time of 85 s. Although not shown, the POI velocity and heading estimate results are similar: larger errors initially, but converging over time, and capturing the deceleration and turn well.

Figure 8b shows a trajectory plot of the UAV and moving POI, along with several time snapshots (at 0, 50, 100, 150, and 200 s) of the results to understand the estimator performance. As with the stationary POI case, the POI uncertainty bounds decrease by large factors when the proximity of the UAV to POI is smaller. Note also that the true location is inside the 95% confidence bounds at all times except in the 170 s range; this was due to the POI moving to the very edge of the camera box, and the operator had to nudge the POI back to the center, thus showing the sensitivity of the estimator to the camera tracking software.

Moving Point of Interest: Three-Dimensional Estimator

Figure 9 shows the results for the three-dimensional geolocation estimator and a moving POI, with the POI altitude estimated. The standard deviation of the error over one full orbit was $2.11L$. For the SPF bound case, the average bound width was $2.26L$, and the true POI location stays within the bounds only 21.2% of the time. For the SPF + bias bound case, the average bound width was $6.2L$, and the true POI location stays within the bounds 91.2% of the time.

In this case, performance is degraded from the two-dimensional case ($\approx$40%), which is qualitatively similar to the stationary results. There are, however, several notable elements of the estimator. First, the three-dimensional estimator performs relatively poorly up to 60 s; compared to the previous case, it is clear that having a POI altitude estimate is quite helpful during the initialization of the estimator. Second, the three-dimensional estimator does not perform well near the turn, where the truth for all three states (latitude, longitude, and altitude) falls outside the bounds of the SPF bound + bias case. The velocity estimate, which is not shown, is similar in that the three-dimensional estimator and has difficulty tracking the velocity during deceleration and acceleration. It appears that the tracking and bias uncertainties, when POI altitude must also be estimated, influence the performance for this specific SeaScan case; with smaller tracking and bias uncertainties, the ability to more accurately estimate the POI maneuver would increase. The heading estimate, which is also not shown, is tracked well in this case.

Conclusions

A modular geolocation estimator has been presented for production UAVs with vision sensors. The estimator has a modular update, which enables an easy update in cases in which components are changed (such as a new avionics system or a different sensor). A novel bias estimation approach has been developed that models unobservable, potentially time-varying biases using uniform distributions and fuses these with the output of the geolocation estimator to produce theoretically sound, consistent geolocation estimates. Flight tests were run using the SeaScan UAV; the geolocation tracking system for the uninhabited aerial vehicles includes avionics, a gimballing camera with feedback isolation and command loops, and a ground station. Stationary and moving targets were evaluated and equipped with GPS for ground truth. Both two- and three-dimensional geolocation estimators were evaluated, with most results producing consistent estimates. Results for the SeaScan case indicate that the geolocation performance is largely affected by the biases, particularly in attitude and camera angles. Geolocation performance improved by up to a factor of five when the UAV was overhead of the target, as opposed to being further away in two dimensions (latitude and longitude). Inclusion of a measurement of the target altitude

decreases sensitivity, and increases performance, both in terms of convergence time and final uncertainty.

References

[1] Pack, D. J., York, G. W. P., Sward, R. E., and Cooper, S. D., "Searching, Detecting, and Monitoring Emergency Sites Using Multiple Unmanned Aerial Vehicles," AIAA Paper 2005-7031, 2005.

[2] Argrow, B., Lawrence, D., and Rasmussen, E., "UAV Systems for Sensor Dispersal, Telemetry, and Visualization in Hazardous Environments," AIAA Paper 2005-1237, 2005.

[3] Murphy, R., Stover, S., and Choset, H., "Lessons Learned on the Uses of Unmanned Vehicles from the 2004 Florida Hurricane Season," Association for Unmanned Vehicle Systems International, Arlington TX, 2005 pp. 1075–1105.

[4] Murphy, R., "Unmanned Vehicles for the Rest of Us: $U \times V$s for Homeland Security Versus Homeland Defense," *Workshop on Human Factors of UAVs*, Cognitive Engineering Research Inst., Mesa, AZ, 2006.

[5] Ridley, M., Nettleton, E., Sukkarieh, S., and Durrant-Whyte, H., "Tracking in Decentralised Air-Ground Sensing Networks," *International Conference on Information Fusion*, Vol. 1, Inst. of Electrical and Electronics Engineers, Piscataway, NJ, 2002, pp. 616–623.

[6] Quigley, M., Goodrich, M., Griffiths, S., Eldredge, A., and Beard, R. W., "Target Acquisition, Localization, and Surveillance Using a Fixed-Wing Mini-UAV and Gimbaled Camera," *IEEE International Conference on Robotics and Automation*, IEEE Publications, Piscataway, NJ, 2005, pp. 2600–2605.

[7] Holland, G., McGeer, T., and Youngren, H., "Autonomous Aerosondes for Economical Atmospheric Soundings Anywhere on the Globe," *Bulletin of the American Meteorological Society*, Vol. 73, No. 12, 1992, pp. 1987–1998.
doi:10.1175/1520-0477(1992)073<1987:AAFEAS>2.0.CO;2

[8] McGeer, T., and Vagners, J., "Historic Crossing: An Unmanned Aircraft's Atlantic Flight," *GPS World*, Vol. 10, No. 2, Feb. 1999, pp. 24–30.

[9] Campbell, M. E., Han, J., Lee, J., Scholte, E., and Ousingsawat, J., "Validation of Active State Model Based Control Using the SeaScan UAV," AIAA Paper 2003-6540, 2003.

[10] Campbell, M., Lee, J., Scholte, E., and Rathbun, D., "Simulation and Flight Test of Autonomous Aircraft Estimation, Planning, and Control Algorithms," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 6, 2007, p. 1597.
doi:10.2514/1.29719

[11] Campbell, M., and Whitacre, W., "Cooperative Tracking Using Vision Measurements on SeaScan UAVs," *IEEE Transactions on Control Systems Technology*, Vol. 15, No. 4, 2007, pp. 613–626.
doi:10.1109/TCST.2007.899177

[12] Goodman, G., "Payloads on Military UAVs Getting Increasingly Sophisticated," *Space News*, Vol. 16, 2005, p. 16.

[13] Weatherington, D. D., "DoD UAV Roadmap," Department of Defense, Tech. Rept. 2002-2027, 2003.

[14] Frew, E., McGee, T., Kim, Z., Xiao, X., Jackson, S., Morimoto, M., Rathinam, S., Padial, J., and Sengupta, R., "Vision-Based Road-Following Using a Small Autonomous Aircraft," *IEEE Aerospace Conference*, IEEE Publications, Piscataway, NJ, 2004, pp. 3006–3015.

[15] Langelaan, J., and Rock, S., "Towards Autonomous UAV Flight in Forests," AIAA Paper 2005-5870, 2005.

[16] Grocholsky, B., Bayraktar, S., Kumar, V., and Pappas, G., "UAV and UGV Collaboration for Active Ground Feature Search and Localization," AIAA Paper 2004-6565, 2004.

[17] Hsiao, F. B., Chien, Y.-H., Liu, T.-L., Lee, M. T., Chang, W.-Y., Han, S. Y., and Wang, Y.-H., "A Novel Unmanned Aerial Vehicle System with Autonomous Flight and Auto-Lockup Capability," AIAA Paper 2005-1050, 2005.

[18] Jakobsen, O. C., and Johnson, E. N., "Control Architecture for a UAV Mounted Pan/Tilt/Roll Camera Gimbal," AIAA Paper 2005-7145, 2005.

[19] Wang, I. H., Dobrokhodov, V. N., Kaminer, I. I., and Jones, K. D., "On Vision-Based Target Tracking and Range Estimation for Small UAVs," AIAA Paper 2005-6401, 2005.

[20] Ivey, G., and Johnson, E., "Investigation of Methods for Target State Estimation Using Vision Sensors," AIAA Paper 2005-6067, 2005.

[21] Kaaniche, K., Champion, B., Pegar, C., and Vasseur, P., "A Vision Algorithm for Dynamic Detection of Moving Vehicles with a UAV," *IEEE International Conference on Robotics and Automation*, IEEE Publications, Piscataway, NJ, 2005, pp. 1878–1883.

[22] Dobrokhodov, V. N., Kaminer, I. I., Jones, K. D., and Ghabcheloo, R., "Vision-Based Tracking and Motion Estimation for Moving Targets Using Small UAVs," AIAA Paper 2006-6606, 2006.

[23] Stepanyan, V., and Hovakimyan, N., "Visual Tracking of a Maneuvering Target," AIAA Paper 2006-6717, 2006.

[24] Ivey, G., and Johnson, E., "Investigation of Methods for Simultaneous Localization and Mapping Using Vision Sensors," AIAA Paper 2006-6578, 2006.

[25] Brunke, S., and Campbell, M. E., "Square Root Sigma Point Filtering for Aerodynamic Model Estimation," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 2, 2004, pp. 314–317.
doi:10.2514/1.1021

[26] Bar-Shalom, Y., Li, X. R., and Kirubarajan, T., *Estimation with Application to Tracking and Navigation*, Wiley, New York, 2001, pp. 71, 267–275.

[27] Whitacre, W., and Campbell, M., "Cooperative Geolocation and Sensor Bias Estimation for UAVs with Articulating Cameras," AIAA Paper 2009-6220, 2009.

[28] Boyd, J., Meloche, J., and Vardi, Y., "Statistical Tracking in Video Traffic Surveillance," *International Conference on Computer Vision*, Inst. of Electrical and Electronics Engineers, Piscataway, NJ, 1999, pp. 163–168.

[29] Isard, M., and Blake, A., "CONDENSATION—Conditional Density Propagation for Visual Tracking," *International Journal of Computer Vision*, Vol. 29, No. 1, 1998, pp. 5–28.
doi:10.1023/A:1008078328650

[30] Horn, B. K. P., and Rhunck, B. G., "Determining Optical Flow," *Artificial Intelligence*, Vol. 17, 1981, pp. 185–203.
doi:10.1016/0004-3702(81)90024-2

[31] Stevenson, D., Wheeler, M., Campbell, M., Whitacre, W., Rysdyk, R., and Wise, R., "Experiments in Cooperative Tracking of Moving Targets by a Team of Autonomous UAVs," AIAA Paper 2007-6756, 2007.

[32] Whitacre, W., and Campbell, M., "Information-Theoretic Optimization of Periodic Orbits for Persistent Cooperative Geolocation," AIAA Paper 2008-7020, 2008.