

Vision-Based Precision Approach and Landing for Advanced Air Mobility

Evan Kawamura ^{*}, Keerthana Kannan [†], Thomas Lombaerts [‡], and Corey Ippolito [§]
NASA Ames Research Center, Moffett Field, Mountain View, CA 94035

Advanced Air Mobility (AAM) aircraft require perception systems for precision approach and landing systems (PALS) in urban, suburban, rural, and regional environments. The current state-of-the-art methods approved for automated approach and landing will be difficult to utilize in support of AAM operational concepts. However, there are technology and systems from other applications and lower-TRL research that use vision, IR, radar, and GPS methods to provide baseline perception and sensing requirements for AAM aircraft approach and landing. This paper focuses on vision-based PAL to demonstrate a closed-loop baseline controller while adhering to the Federal Aviation Administration requirements and regulations. The coplanar algorithm determines pose estimation, which feeds into an Extended Kalman filter. Combining IMU with vision creates a sensor fusion navigation solution for GPS-denied environments. The state estimate leads to glideslope and localizer error computations, which will be pertinent for designing and deriving guidance laws and control laws for AAM PALS. The IMU and vision navigation solution provides promising simulation results for AAM PALS, and higher fidelity simulations will include computer graphics rendering and feature correspondence.

I. Introduction

ADVANCED Air Mobility (AAM) assists emerging aviation markets by developing safe air transportation systems in urban, suburban, rural, and regional environments by utilizing revolutionary models to transport people and cargo. These AAM aircraft will benefit the public and air transportation sectors but need autonomy to increase performance capabilities and efficiency. Implementing autonomy onboard AAM aircraft removes the need for human pilots, which increases the payload capacity. AAM aircraft will need an accurate and autonomous approach and landing system onboard to ensure safe landings. AAM operational concepts may struggle with implementing current state-of-the-art methods for automated precision approach and landing (PAL). However, baseline perception and sensing requirements for AAM aircraft approach and landing can draw from existing technology and systems such as vision, IR, radar, glideslope indicators, and GPS.

A. Traditional and Current Landing Systems

There are several types of traditional and current aircraft landing systems. Instrument Landing Systems (ILS) provide aircraft approach navigation aids with three components: glideslope, localizer, and marker beacons. The glideslope provides vertical guidance, the localizer provides horizontal guidance, and the marker beacons provide radio checks during descent. There are also Visual Approach Slope Indicator (VASI) systems, which give approach slope information during aircraft descent. There are many VASI systems, but a common theme is white light beams for flying above the glidepath and red light beams for flying below the glidepath. The Precision Approach Path Indicator (PAPI) systems replace the older VASI systems with a similar strategy: more red than white lights indicates flying below the glidepath, more white than red lights indicates above the glidepath, and an even distribution between the red and white lights shows flying on the glidepath. A glidepath light variation includes tri-color: red for below, green for on, and yellow for above the glidepath. A visual approach variation includes pulsating lights such that pulsating red lights indicate well below the glidepath, pulsating white lights indicate well above the glidepath, and steady red or white indicates smaller glideslope altitude offsets [1]. Ground Based Augmentation Systems (GBAS) have more flexibility and economic benefits than ILS

^{*}Computer Engineer, Intelligent Systems Division, NASA Ames Research Center, Moffett Field, Mountain View, CA 94035, USA.

[†]Computer Engineer, KBR Wyle Services, Intelligent Systems Division, NASA Ames Research Center, Moffett Field, Mountain View, CA 94035, USA.

[‡]Aerospace Research Engineer, KBR Wyle Services, Intelligent Systems Division, NASA Ames Research Center, Moffett Field, Mountain View, CA 94035, USA.

[§]Aerospace Scientist, Intelligent Systems Division, NASA Ames Research Center, Moffett Field, Mountain View, CA 94035, USA

by allowing up to 192 approaches from different approach angles at the same frequency of ILS. However, transitioning from ILS to GBAS at airports will potentially take decades due to operational differences [2].

Some studies utilize GPS and IR beacons for aircraft navigation and landing. One study shows an onboard integrity monitoring and a laser tracker system with accuracies less than 1 ft, and a Kinematic GNSS Landing System (KGLS) provides real-time centimeter-level accuracy with integrity beacons [3]. Flight test results for autoland a Boeing 737 by the Integrity Beacon Landing System (IBLS) have 0.6 ft error or better [4]. A common issue with GPS and integrity beacons is that GPS alone is not sufficient such that a beacon is needed. It is important to note that IR beacons are most effective in lower light environments but limited to weather conditions and ground topography such as vegetation [5]. Another study shows that an autonomous IR landing system for UAVs with one IR source is sufficient for the landing zone [6]. Setting up IR beacons provides waypoints for an efficient guidance system for commercial aircraft between the runway and parking area [7].

B. Vision-Based Navigation

Similar but smaller vehicles such as unmanned aerial vehicles (UAV) have vision-based studies that apply to AAM aircraft precision approach and landing (PAL). Gautam et al. provide a survey of UAV vision-based methods and controllers that has indoor and outdoor landings with pure and pseudo pursuit targets [8]. Utilizing inertial optical flow, a vertical proportional-integral thrust controller, and a pyramidal implementation of the Lucas-Kanade algorithm for hovering and landing [9]. Min et al. demonstrate a feedback guidance law based on time-to-go, velocity, altitude, and gravity computes acceleration commands to change the flight path angle with lateral acceleration constraints for rotary-wing and fixed-wing UAVs [10]. Landing a UAV without GPS can be achieved by inserting computer vision in a feedback control loop to solve the classic ego-motion estimation problem with all the feature points lying on a plane, i.e., landing pad [11].

There are also studies for larger aircraft with vision-based methods. A vision-based closed-loop feedback method occurs in the final approach for eVTOL vehicles and generates accuracies less than 1.5 m when less than 25 m away from the target [12]. A vision-based system for guidance and autonomous safe landing for helicopters in a partially known environment is robust to occlusions and light variations by having a hierarchical controller switch between GPS and vision input data [13]. Equipping an autonomous helicopter with vision in its control loop provides position and velocity estimates of features in an urban environment relative to the helicopter, which leads to guidance and real-time feature tracking [14]. Maritime augmented guidance with integrated controls for carrier approach and recovery enabling technologies (MAGIC CARPET) allows fighter pilots to make simple, precise, repeatable, and consistent landings on aircraft carriers by removing coupled interactions between control inputs [15].

Other studies combine computer vision and Kalman filtering techniques for estimating states. Utilizing a Kalman filter for estimating the camera pose and 3D to 2D line correspondences with pose priors and line features is robust to occlusion and clutter, and a line based pose estimation algorithm for known correspondences has accurate and efficient performance for the model-to-image registration problem [16]. Another study uses an extended Kalman filter to estimate the position of 3D coplanar points when there are four stationary reference points with known coordinates [17]. Combining a landmark-based vision method with an INS to form a loosely-coupled navigation system yields a Kalman filter navigation solution that can operate in GPS-denied environments [18].

C. Computer Vision Research

Some computer vision studies involve extracting the fundamental and essential matrices to determine pose estimation. Estimating the fundamental matrix allows one to compute the essential matrix, which determines the rotation matrix and translation vector between two images. Some methods that estimate the fundamental matrix are the unnormalized and normalized 8-point algorithm, algebraic minimization algorithm, minimizing the Sampson cost function through an iterative-minimization method, and the Gold Standard algorithm [19]. Alternatively, the 5-point algorithm with RANSAC computes the essential matrix directly to obtain the rotation and translation between two images [20]. Having the normalized image coordinates for the two images leads to computation of the essential matrix with four possible solutions for the second camera matrix, which contains the relative rotation and translation [19].

There are also computer vision studies that directly estimate the rotation matrix and translation vector. A least squares fit of two 3D sets of points with SVD yields the relative rotation and translation between the two sets of points [21]. The Pose from Orthography and Scaling with Iterations (POSIT) algorithm determines pose from orthographic projections of an image with noncoplanar feature points of an object [22]. Modifying the POSIT algorithm to scenarios with images containing coplanar points leads to a coplanar POSIT algorithm with a limiting factor that there must be at

least four coplanar points [23]. The SoftPOSIT algorithm combines softassign and pose from orthography and scaling with iterations (POSIT). Softassign solves the image correspondence problem by computing correspondences between object and image points when correspondences are unknown, and POSIT iteratively computes the pose estimation by providing the rotation matrix between the object and camera and translation vector from the camera's center to the object's origin [24].

D. Overview of Work

It is not clear how much overlap exists between current and AAM aircraft PALS due to the lack of available requirements and standards for AAM aircraft. At the time of writing this paper, the FAA has no active Vertiport Design document since FAA AC 150/5390-3: Vertiport Design was canceled in 2010 [25]. However, FAA AC 150/5390-2C: Heliport Design provides adequate estimations of requirements and standards as a guideline for future vertiport requirements and standards [26].

Navigation techniques containing computer vision and vision-based methods might apply to AAM PALS. This paper will provide an initial study on vision-based PALS to determine future PALS research and applications requirements. Thus, the goal of this paper focuses on vision-based PALS to demonstrate an AAM navigation solution for PAL while adhering to the Federal Aviation Administration requirements, regulations, and standards. Overall, as AAM research, operations, and technology progress, NASA will address technical and structural research gaps. At the same time, the FAA has the most critical role of enabling AAM operations with support from state and local governments [27].

The authors of this paper desired to find an easy-to-implement and open-source vision-based algorithm. Initial tests of the open-source SoftPOSIT algorithm * seemed promising but were inadequate in scenarios with coplanar points such as landing lights or markings on the ground or runway. Email correspondence with the second author of Ref. [24] confirmed that the current SoftPOSIT algorithm is not applicable for coplanar points and would have to be modified significantly, which led the authors to download the open-source coplanar POSIT algorithm † described in Ref. [23] for an initial vision-based PAL navigation solution.

Figure 1 shows the proposed software architecture for vision-based AAM PAL. The green blocks work together via post-processing, and the blue blocks are currently under development. This paper shows a post-processed implementation of the green blocks. Solving the feature correspondence and extraction problem will complete the "Image processing (feature correspondence)" blue block. Obtaining the simulated X-Plane telemetry data in real-time to feed into "Vehicle State Estimation Filter" will complete the "Simulated Vehicle Sensor Data Interface" blue block, which currently utilizes simulated telemetry data from the Vertical Motion Simulation (VMS) ‡ and a Cessna plane in the X-Plane flight simulator. Inserting a notification or color scheme like the PAPI or ILS systems will complete the "Pilot GUI Interfaces (PFD)" block to simulate the pilot seeing glideslope and localizer errors. Deriving guidance laws based on the glideslope and localizer errors will complete the "Approach/ Descent Mode Flight Controller" blue block. Finally, creating a main function or wrapper to combine all the vision-based functions and computations into one consolidated package will create the Vision-Based Estimation System.

The "X-Plane Interface (XPI)" has a UDP connection for real-time implementation to connect to X-Plane but remains offline in this paper because the blue blocks are work-in-progress. The "Xplane & World Editor" and "Plug-in to Xplane" green blocks provide an a priori rendered scenery and environment, and the "Video Frame Capture Interface" is currently a recorded video of the X-Plane flight. The "Vision-Based Vehicle State Estimator" contains the coplanar POSIT algorithm, which provides position and attitude estimation. The "Vehicle State Estimation Filter" block uses an extended Kalman filter to combine VMS IMU measurements and pose from coplanar POSIT to output the state estimate to feed into the "Vision-Replaced ILS Landing System Loc/GS Emulator" block, which computes the glideslope and localizer errors.

This initial vision-based AAM PAL study has the potential to impact many areas. First, it will deliver perception PALS requirements and data sets to other NASA projects and industry partners, identifying gaps in perception, technology, and data for verification and validation and characterization to present to other NASA projects and industry partners. Second, AAM aircraft need safe and accurate PALS to ensure the successful transportation of people and cargo. Third, having autonomous PALS removes the need for AAM pilots, which increases efficiency and payload capacity. Finally, it paves the way for new AAM PALS research activities to enhance future AAM operations.

*<http://www.daniel.umiacs.io/SoftPOSIT.txt>

†http://www.daniel.umiacs.io/Site_2/Code.html

‡<https://www.nasa.gov/ames/vms>

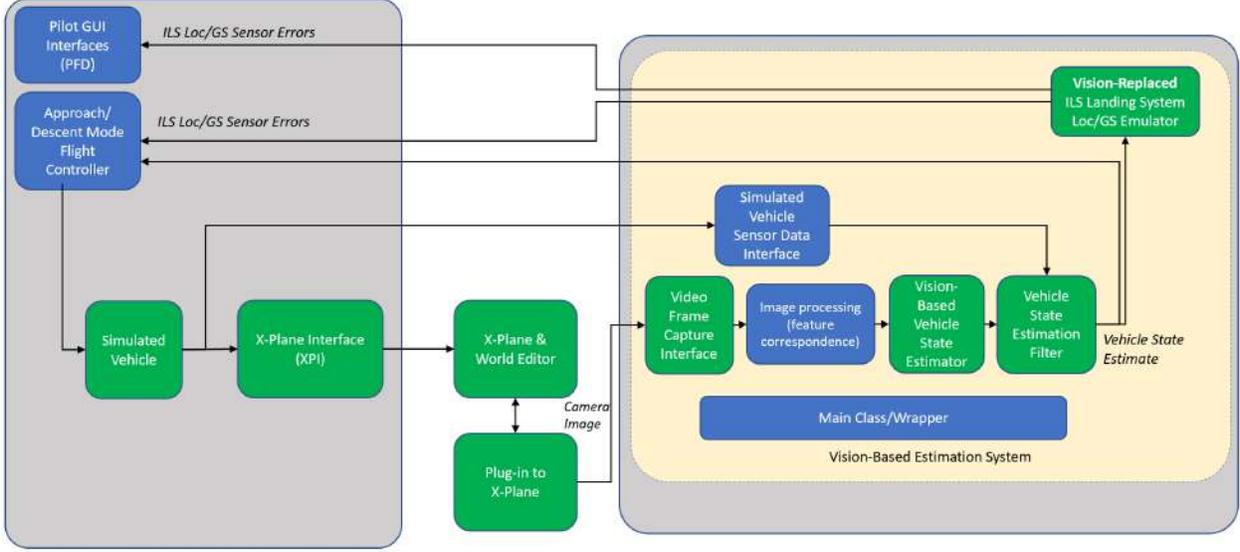


Fig. 1 Proposed Software Architecture Diagram: the green blocks work together via post-processing, and the blue blocks are under development

This paper's organization is as follows. Section II discusses the state vector, coordinate frames, kinematics, and dynamics for AAM aircraft, and section IV provides proposed the approach and landing profile at vertiports. Section V mentions the design and approach for the navigation solution's extended Kalman filter, section VI provides simulation results, and section VII ends with a summary and concluding remarks.

II. Kinematics & Dynamics

This section discusses the state vector, coordinate frames, kinematics, and dynamics for AAM aircraft.

A. State Vector

The state vector of the AAM aircraft is defined as:

$$\mathbf{s} = [E \ N \ U \ v_N \ v_E \ v_U \ \phi \ \theta \ \psi]^T. \quad (1)$$

The state vector \mathbf{s} decomposes to three vectors:

$$\mathbf{s} = [\mathbf{p} \ \mathbf{v} \ \Theta], \quad (2)$$

where each of them is defined as

$$\mathbf{p} = [E \ N \ U], \ \mathbf{v} = [v_x \ v_y \ v_z], \ \Theta = [\phi \ \theta \ \psi]. \quad (3)$$

The vector \mathbf{p} is in East, North, and Up (ENU) coordinates in the inertial frame, fixed on the ground at the helipad landing site. The translational velocities in \mathbf{v} are in the inertial frame. The Euler angles are the roll, pitch, and yaw angles (ϕ, θ, ψ) .

B. Coordinate Frames

The world coordinate system (WCS) is an inertial frame fixed on the ground in which gravity is pointing in the negative U-direction, i.e., down. The vehicle coordinate system (VCS) is on the body frame on the aircraft such that the x-axis points right, the y-axis point forward, and the z-axis points up in the same direction as the motor axes. The camera coordinate system (CCS) has the camera fixed to the aircraft's body, angled down, and pointed in the positive z-axis. Its x-axis points right like in VCS, and the y-axis points down and behind the aircraft. Figure 2 shows the WCS axes

denoted by E, N, U , the VCS axes denoted by VCS_x, VCS_y, VCS_z , and the CCS axes denoted by CCS_x, CCS_y, CCS_z . A rotation matrix following the (3-1-2) sequence is applied to rotate the aircraft from the inertial frame to the body frame [28].

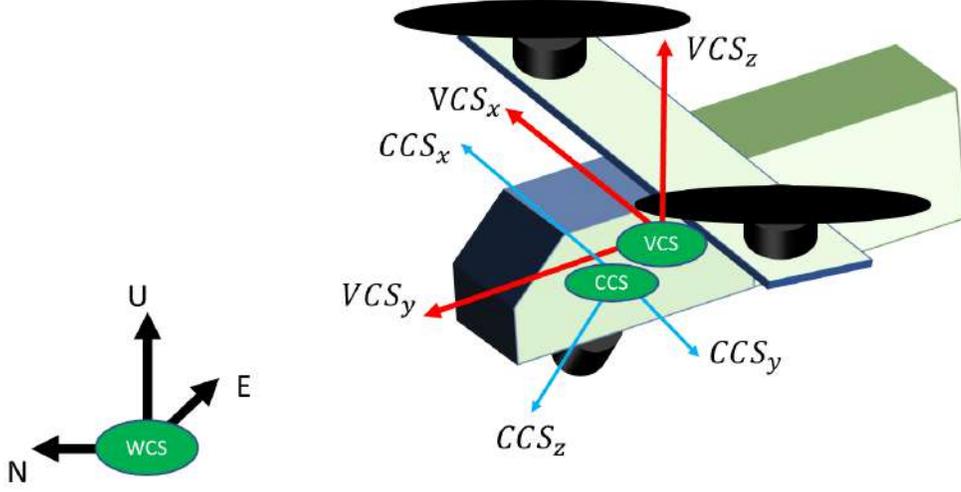


Fig. 2 Inertial, Body, and Camera Frames of AAM Aircraft

C. Euler Angles

This paper utilizes the (3-1-2) sequence of the direction cosine matrix rotates the inertial frame to the body frame through the Euler angles [28]:

$$\mathbf{R} = \mathbf{R}_y(\phi)\mathbf{R}_x(\theta)\mathbf{R}_z(\psi) = \begin{bmatrix} -s\psi s\theta s\phi + c\phi c\psi & s\phi s\theta c\psi + c\phi s\psi & -s\phi c\theta \\ -c\theta s\psi & c\theta c\psi & s\theta \\ c\phi s\theta s\psi + s\phi c\psi & -c\phi s\theta c\psi + s\phi s\psi & c\phi c\theta \end{bmatrix}. \quad (4)$$

where $c\theta$ and $s\theta$ denote $\cos \theta$ and $\sin \theta$ respectively. $\mathbf{R}_z(\psi)$ is the rotation matrix around the z-axis by ψ , $\mathbf{R}_x(\theta)$ is the rotation matrix around the once rotated x-axis by θ , and $\mathbf{R}_y(\phi)$ is the rotation matrix around the twice rotated y-axis by ϕ . The relationship between the angular velocity and Euler angular rates for the (3-1-2) direction cosine matrix sequence is [28]:

$$\boldsymbol{\Omega} = \begin{bmatrix} 0 & \cos \phi & -\cos \theta \sin \phi \\ 1 & 0 & \sin \theta \\ 0 & \sin \phi & \cos \theta \cos \phi \end{bmatrix} \dot{\boldsymbol{\Theta}} \quad (5)$$

such that $\boldsymbol{\Omega} = [rqp]^T$ and $\dot{\boldsymbol{\Theta}} = [\dot{\psi}\dot{\theta}\dot{\phi}]^T$.

D. Position and Velocity

Transposing the direction cosine matrix of the prior subsection relates the time derivative of the inertial position vector and the body frame's velocity vector:

$$\begin{bmatrix} v_N \\ v_E \\ v_U \end{bmatrix} = \begin{bmatrix} -s\psi s\theta s\phi + c\phi c\psi & -c\theta s\psi & c\phi s\theta s\psi + s\phi c\psi \\ s\phi s\theta c\psi + c\phi s\psi & c\theta c\psi & -c\phi s\theta c\psi + s\phi s\psi \\ -s\phi c\theta & s\theta & c\phi c\theta \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (6)$$

E. Translational Dynamics

The general aircraft translational dynamic equations are [29]:

$$\begin{aligned} F_x &= m(\dot{u} + qw - rv) + mg \sin \theta \\ F_y &= m(\dot{v} + ru - pw) - mg \cos \theta \sin \phi \\ F_z &= m(\dot{w} + pv - qu) - mg \cos \theta \cos \phi, \end{aligned} \quad (7)$$

in which m is the mass, g is the acceleration due to gravity, F_x, F_y, F_z are the aerodynamic forces, u, v, w are the velocities in the body frame, p, q, r are the angular velocities in the body frame, and ϕ, θ, ψ are the roll, pitch, and yaw Euler angles. Modeling specific forces as accelerometer measurements at the aircraft's center of gravity measure the specific aerodynamic forces: [29]

$$F_x = A_x m, \quad F_y = A_y m, \quad F_z = A_z m \quad (8)$$

such that A_x, A_y, A_z are the accelerometer measurements at the aircraft's center of gravity. Inserting Eq. (7) into Eq. (8) yields:

$$\begin{aligned} \dot{u} &= A_x - g \sin \theta - qw + rv, \\ \dot{v} &= A_y - g \cos \theta \sin \phi - ru + pw, \\ \dot{w} &= A_z + g \cos \theta \cos \phi - pv + qu, \end{aligned} \quad (9)$$

which removes mass and forms a set of kinematic equations for all types of aircraft regardless of mass.

III. Tentative Vertiport Landing Light Configuration

A. Pre-existing Information and Sources

Since there are no active FAA vertiport design documents, baseline requirements, and light configurations come from the FAA heliport design document, AC 150/5390-2C - Heliport Design. Figure 2-2 of Ref. [26] states that the minimum TLOF and FATO lengths should be one rotor diameter and 1.5 diameters, respectively. However, these minimum lengths depend on helicopters instead of AAM aircraft. Ref. [30] provides a figure and dimensions for a VTOL Cessna model with a 36 ft wingspan, which resembles future lift-plus-cruise AAM aircraft. Replacing rotor diameter with a wingspan in the FAA Heliport Design document gives a minimum TLOF length of 36 ft and a minimum FATO length of 54 ft.

The FAA Heliport Design document provides several figures and types of landing lights for a baseline landing light configuration. Note to readers: figures referenced in this paragraph all refer to Ref. [26] with citations omitted for the reader's convenience. There are sixteen TLOF lights near the inner square's perimeter and twenty-eight FATO lights near the outer square's perimeter (see Fig. 2-29). It also includes five omnidirectional green lights in front of the helipad (see Fig. 2-31). Seven optional TLOF lights are placed vertically on the TLOF, twelve edge bar lights, and twelve wing bar lights (see Fig. 6-2). Finally, there are thirty approach lights in a triangle or Christmas tree configuration (see Fig. 6-31). Ultimately, putting this all together yields one hundred ten lights, which provides a tentative baseline landing light configuration. The following subsection provides the method for determining the light locations.

B. Landing Light Configuration

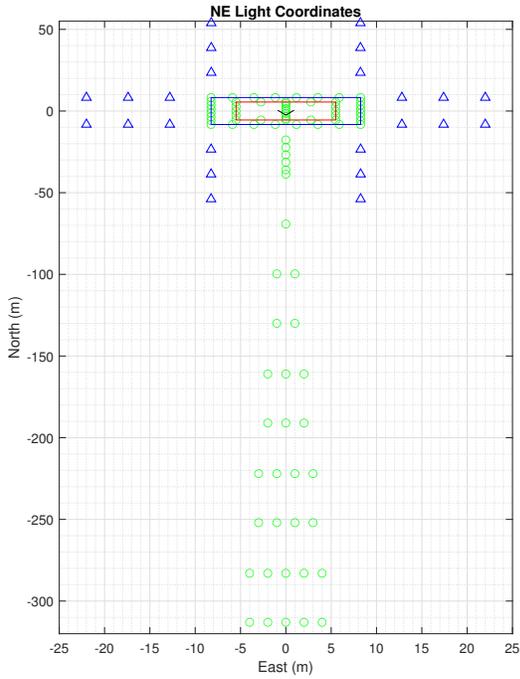
Figure 3a shows the landing lights in the WCS with the origin located at the center of the landing pad. Transforming the landing lights from WCS to CCS and following the light rendering pipeline from Caltech[§] leads to an initial rendering of the landing lights. Fig. 3b shows the landing lights when the aircraft is 950 m south from the center of the landing pad at an altitude of 152 m.

The initial rendering parameters for generating Fig. 3b follows the SL1250 Telephoto 4K resolution day/night lenses for 1/1.7" sensors.[¶] The aircraft has neutral Euler angles, i.e., $\phi = \theta = \psi = 0$, and the camera is tilted down by 9° to align with the 9° glideslope. The camera is 2 m forward from the aircraft's center of gravity, i.e., 2 m positive along the y-axis. The camera's resolution is 4000×2000 pixels, which places the camera's center at (2000, 1000) with the origin at the bottom left corner of the image. The camera's horizontal field of view is 36° , and its vertical field of view is 26° . The focal length ranges from 12 to 50 mm, which yields a focal length of 5263 pixels.^{||} Using the relationship between

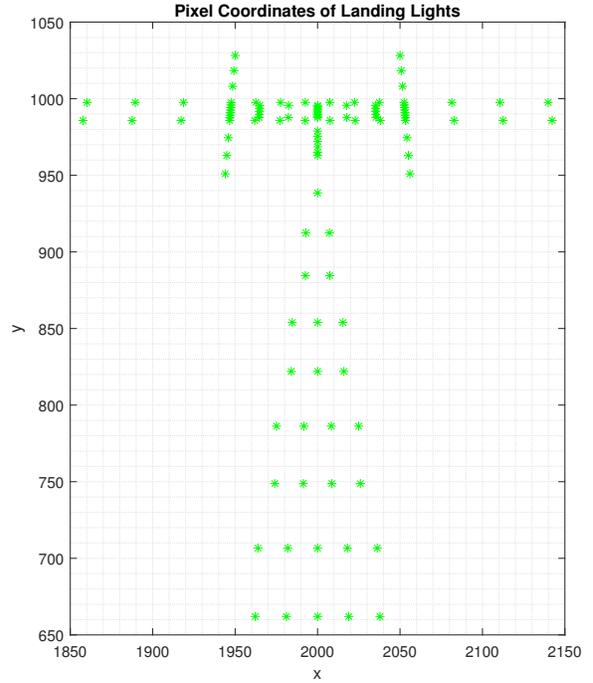
[§]http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/parameters.html

[¶]<https://www.rmaelectronics.com/content/Theia-Technologies-Lenses/SL1250%20spec-short%20form%20150408.pdf>

^{||}<https://answers.opencv.org/question/17076/conversion-focal-distance-from-mm-to-pixels/>



(a) World Coordinate System - North & East



(b) Pixel Coordinates

Fig. 3 Landing Lights in World Coordinates and Pixel Coordinates

the focal length, the sensor width/height, and field of view leads to the computation of the pixels in the horizontal and vertical axes:

$$f_x = \frac{\text{pixelwidth}/2}{\tan FOV_h}, f_y = \frac{\text{pixelheight}/2}{\tan FOV_v} \quad (10)$$

in which FOV_h and FOV_v are the horizontal and vertical fields of view in radians, and f_x and f_y are the horizontal and vertical focal lengths in pixels [31].

IV. Approach and Landing Profile

Figure 4 shows a screenshot of slide 27 from Ref. [32], which presents a tentative approach profile for UAM vertiports. The authors assume that the glidepath, horizontal distances, altitude, and decelerations are in the WCS fixed to a static location such as the touchdown point (TDP). Important AAM aircraft kinematic parameters to note are the glidepath deceleration of $-1.6ft/s^2$, glidepath distance of 3156 ft, initial altitude of 500 ft, and an initial Final Approach Fix (FAF) forward velocity of 70 KIAS pointed towards the TDP. There are no official and active FAA Vertiport Design documents, but the vertiport glidepath provided in Ref. [32] contains the same glideslope angle of 9° in Ref. [25]. Assembling all this information tentatively provides an adequate baseline approach path for vision-based AAM PALS. Figure 5 modifies the vertiport approach profile in Fig. 4 by including the last row of HALS lights and using the last row of HALS lights as the origin instead of the TDP. The flight phases are outlined as:

- A→B: glideslope descent
 - Start: 950 m south of the last row of HALS lights
 - Stop: 100 m south of the last row of HALS lights
- B→C: forward flight phase 1 with POSIT and camera pitched down
 - Start: 100 m south of the last row of HALS lights
 - End: 250 m north of the last row of HALS lights
- C→D: forward flight phase 2 with homography (non-POSIT) and downward facing camera
 - Start: 250 m north of the last row of HALS lights

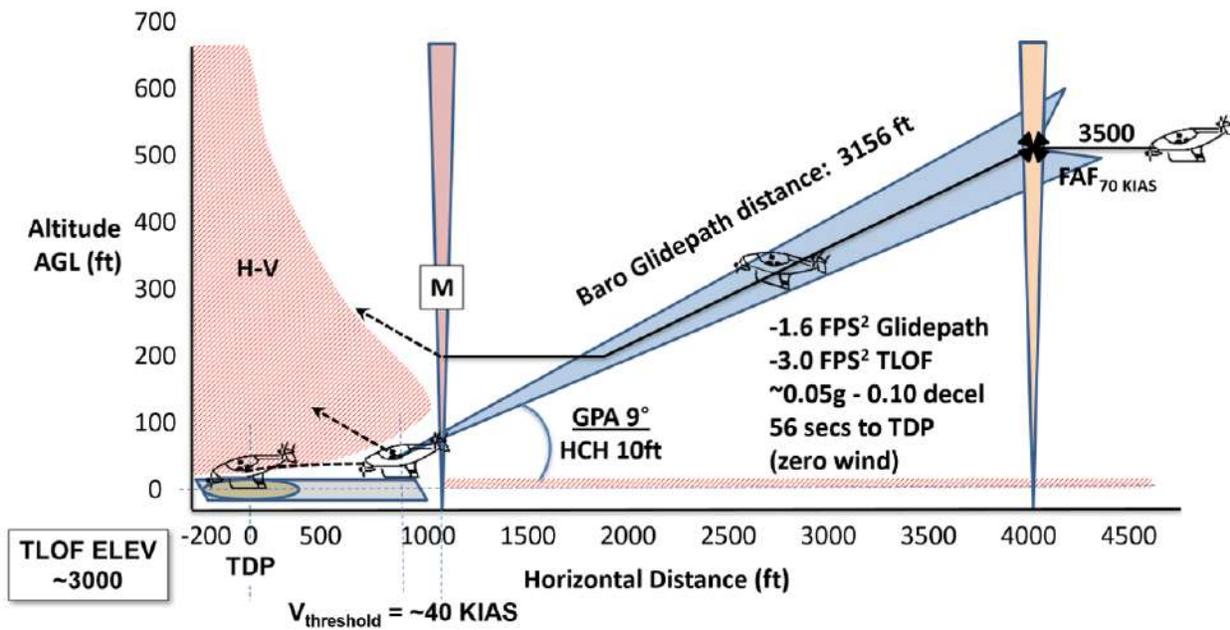


Fig. 4 FAA and the National Campaign ppt slide 27 [32]

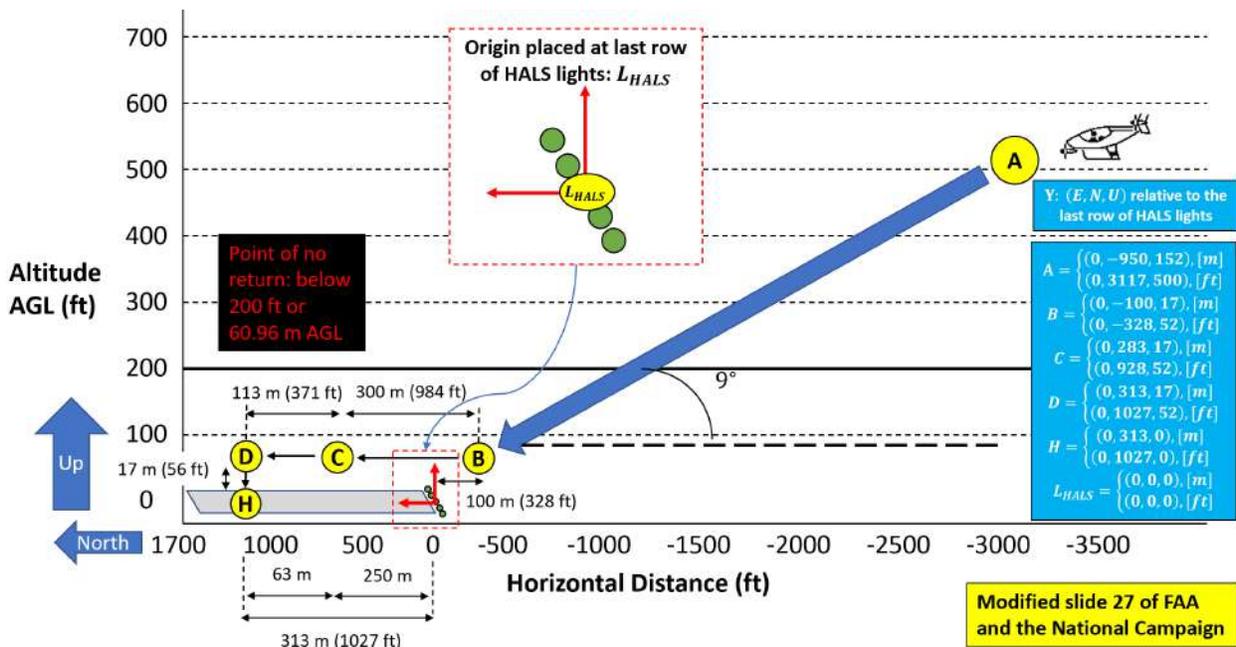


Fig. 5 FAA and the National Campaign ppt: modified slide 27

- End: 318 m north of the last row of HALS lights (68 m north of the helipad's H)
- D→H: vertical descent once centered on the helipad's H

With a camera tilted down at a fixed angle, there will be fewer landing lights within view during forward flight (B→C and C→D), which directly affects the accuracy of the vision-based navigation solution. An insufficient number of features or landing lights will be detrimental and difficult to determine the aircraft's position relative to the TDP. Thus, the considered flight phase in this paper focuses on the glideslope descent (A→B) to ensure all the landing lights are within view.

V. Extended Kalman Filter Design

This paper utilizes a continuous-discrete extended Kalman filter for a stationary system, i.e., \mathbf{F} is time-invariant and follows a similar approach in Ref. [33]-[34]. The input vector, $\mathbf{u} \in \mathbb{R}^6$, includes the IMU measurements in the body frame:

$$\mathbf{u} = \begin{bmatrix} acc_x & acc_y & acc_z & gyro_x & gyro_y & gyro_z \end{bmatrix}. \quad (11)$$

The predicted state is defined as:

$$\hat{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t), \hat{\mathbf{x}}_k(-) = \hat{\mathbf{x}}_k(+) + \int_{k-1}^k \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) dt \quad (12)$$

with $(-)$ to denote before measurements and k represents the k th iteration. The predicted covariance before measurements is defined as:

$$\mathbf{P}_k(-) = \mathbf{\Phi}(k)\mathbf{P}_{k-1}(+)\mathbf{\Phi}(k)^T + \mathbf{\Gamma}(k)\mathbf{Q}(k)\mathbf{\Gamma}(k)^T \quad (13)$$

such that with $(+)$ denotes after measurements. $\mathbf{\Phi}(k)$ and $\mathbf{\Gamma}(k)$ are defined as:

$$\mathbf{\Phi}(k) = e^{\mathbf{F}(k)\Delta t}, \quad \mathbf{\Gamma}(k) = \left(\int \mathbf{\Phi}(k)\Delta t \right) \mathbf{G}(k) \quad (14)$$

The $\mathbf{F}(k)$, $\mathbf{G}(k)$, and $\mathbf{H}(k)$ matrices are defined as:

$$\mathbf{F}(k) = \left. \frac{\partial \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t)}{\partial \mathbf{x}(t)} \right|_{\mathbf{x}=\hat{\mathbf{x}}_k(+)}, \quad \mathbf{G}(k) = \left. \frac{\partial \mathbf{G}(\mathbf{x}(t))}{\partial \mathbf{x}(t)} \right|_{\mathbf{x}=\hat{\mathbf{x}}_k(+)}, \quad \mathbf{H}(k) = \left. \frac{\partial \mathbf{h}(\mathbf{x}(t), \mathbf{u}(t), t)}{\partial \mathbf{x}(t)} \right|_{\mathbf{x}=\hat{\mathbf{x}}_k(+)}. \quad (15)$$

The Kalman gain matrix is:

$$\mathbf{K}_k = \mathbf{P}_k(-)\mathbf{H}_k^T (\mathbf{H}_k\mathbf{P}_k(-)\mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (16)$$

The state estimate update equation is:

$$\hat{\mathbf{x}}_k(+) = \hat{\mathbf{x}}_k(-) + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k(-)) \quad (17)$$

Using the Joseph stabilized version of the covariance measurement update is a more stable and robust formulation such that it guarantees $\mathbf{P}_k(+)$ will be symmetric and positive definite if $\mathbf{P}_k(-)$ is symmetric and positive definite: [35]

$$\mathbf{P}_k(+) = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k(-)(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T. \quad (18)$$

The process and measurement noise covariances are assumed to be constant, diagonal, and utilize the Gaussian distribution, i.e., $\mathbf{Q}, \mathbf{R} \sim N[\mu, \sigma^2]$ with μ as the mean and σ as the standard deviation. The EKF uses the IMU as the input vector and coplanar POSIT for pose measurements, so $\mathbf{Q}, \mathbf{R} \in \mathbb{R}^{6 \times 6}$. Thus, the process noise covariance includes uses the IMU measurement variances:

$$\mathbf{Q} = \text{diag} \left[\sigma_{acc,x}^2 \quad \sigma_{acc,y}^2 \quad \sigma_{acc,z}^2 \quad \sigma_{gyro,x}^2 \quad \sigma_{gyro,y}^2 \quad \sigma_{gyro,z}^2 \right]. \quad (19)$$

The measurement noise covariance utilizes the variances from the coplanar POSIT algorithm:

$$\mathbf{R} = \text{diag} \left[\sigma_{N,POSIT}^2 \quad \sigma_{E,POSIT}^2 \quad \sigma_{U,POSIT}^2 \quad \sigma_{\phi,POSIT}^2 \quad \sigma_{\theta,POSIT}^2 \quad \sigma_{\psi,POSIT}^2 \right]. \quad (20)$$

Figure 6 shows the block diagram of the proposed EKF. The black box with "c2d" utilizes the functions in Eq. (14) to compute $\mathbf{\Phi}$ and $\mathbf{\Gamma}$. The box with "Plant" contains the kinematic equations for predicting the state, $\hat{\mathbf{x}}(-)$, before taking measurements. The solid black lines represent computations for the state, while the dashed blue lines represent computations for covariance. Figure 7 shows the outputs of the IMU and POSIT as \mathbf{u} and \mathbf{z} , respectively, which feed into the EKF as the input vector and measurement vector. The accelerometer and gyroscope measurements in the body frame feed into the input vector, \mathbf{u} , which feeds into the plant and computations for \mathbf{F} and \mathbf{G} with white Gaussian noise, \mathbf{w} .

VI. Simulation Results

A. Initial Coplanar POSIT Algorithm Results

After downloading and altering the coplanar POSIT C code **, initial simulation tests show that coplanar POSIT works for a downward-facing camera tilted down by 9° descending on a 9° glideslope (recall A→B in Fig. 5). It takes approximately 0.027 seconds per iteration to run the coplanar POSIT algorithm, which demonstrates real-time capabilities for onboard implementation in the future.

Tables 1-2 show the initial coplanar POSIT estimations for the position (WCS) in meters with North being the most inaccurate. The camera Euler angles are in degrees with high accuracy ($\psi_{\text{des}} = 0^\circ, \theta_{\text{des}} = -9^\circ, \psi_{\text{des}} = 0^\circ$), especially the camera pitch down angle. Eq. (21) computes the errors between the desired and estimated variables with subscript, des, to denote desired variables subscripts and subscript, est, to denote estimated variables.

$$\begin{aligned}
 \Delta E &= E_{\text{des}} - E_{\text{est}}, \Delta \phi = \phi_{\text{des}} - \phi_{\text{est}} \\
 \Delta N &= N_{\text{des}} - N_{\text{est}}, \Delta \theta = \theta_{\text{des}} - \theta_{\text{est}} \\
 \Delta U &= U_{\text{des}} - U_{\text{est}}, \Delta \psi = \psi_{\text{des}} - \psi_{\text{est}} \\
 \Delta \mathbf{p} &= \sqrt{\Delta N^2 + \Delta E^2 + \Delta U^2}, \Delta \Theta = \sqrt{\Delta \phi^2 + \Delta \theta^2 + \Delta \psi^2}
 \end{aligned} \tag{21}$$

Table 1 Initial Position Estimation Coplanar POSIT Glideslope Test

E_{des}	N_{des}	U_{des}	E_{est}	N_{est}	U_{est}	ΔE	ΔN	ΔU	$\Delta \mathbf{p}$
0	-950	152	0	-948.2	151.84	0	1.794	-0.1564	1.801
0	-850	136	0	-845.9	135.63	0	4.071	-0.3699	4.087
0	-750	120	-0.1434	-745.1	119.46	-0.1434	4.942	-0.5428	4.974
0	-650	104	0.1241	-644.9	103.53	0.1241	5.062	-0.4698	5.086
0	-550	88	0	-548.0	88.07	0	2.026	0.07176	2.027
0	-450	72	-0.0867	-450.4	72.40	-0.08668	-0.4449	0.3966	0.602
0	-350	56	0	-347.9	55.91	0	2.119	-0.0877	2.121
0	-250	40	-0.04761	-247.4	39.95	-0.04761	2.636	-0.0501	2.637
0	-150	24	0	-148.0	23.99	0	1.983	-0.008429	1.983
0	-100	16	-0.01888	-98.1	16.01	-0.01888	1.934	0.00991	1.935

Table 2 Initial Attitude Estimation Coplanar POSIT Glideslope Test

E_{des}	N_{des}	U_{des}	ϕ_{est}	θ_{est}	ψ_{est}	$\Delta \phi$	$\Delta \theta$	$\Delta \psi$	$\Delta \Theta$
0	-950	152	0.2421	-8.969	-0.2394	-0.2421	-0.03142	0.2394	0.3419
0	-850	136	0.1721	-8.975	-0.1696	-0.1721	-0.02539	0.1696	0.2430
0	-750	120	0.1060	-8.968	-0.1001	-0.1060	-0.03177	0.1001	0.1492
0	-650	104	-0.1605	-8.974	0.1543	0.1605	-0.02584	-0.1543	0.2241
0	-550	88	-0.1087	-9.014	0.1079	0.1087	0.01351	-0.1079	0.1537
0	-450	72	0.0653	-9.031	-0.06225	-0.06534	0.03135	0.06225	0.09554
0	-350	56	-0.0400	-8.986	0.03924	0.03999	-0.01378	-0.03924	0.05770
0	-250	40	-0.0392	-9.000	0.03974	0.03918	-0.000318	-0.03974	0.05580
0	-150	24	-0.0872	-8.995	0.08633	0.08719	-0.00453	-0.08633	0.1228
0	-100	16	-0.00245	-9.002	0.00308	0.00245	0.00195	-0.00308	0.00439

**http://www.daniel.umiacs.io/Site_2/Code.html

For forward flight, initial tests show that coplanar POSIT fails at 29 m (AGL altitude is 16 m) from the TDP because there are less than four coplanar points in the camera's view. Consequently, forward flight (B→C & C→D in Fig. 5) requires alternative methods to determine pose estimation. It is unknown what vision-based methods will be best for forward and is part of future work. The last flight phase, D→H, in Fig. 5 can utilize similar techniques shown in Ref. [12] through a nadir camera and applying SIFT and the homography matrix to match a reference image of the landing pad.

B. EKF Results

The VMS provided telemetry data of a simulated RVLТ quad model for manual approach and landing at NASA Neil Armstrong Flight Research Center (AFRC). Figure 8 shows a screenshot of the video. After modifying the simulated



Fig. 8 Screenshot of VMS Manual Descent at NASA AFRC

VMS telemetry for a 9° glideslope, taking the variances of the accelerometer and gyroscope measurements in the body frame leads to values for the process noise covariance:

$$\mathbf{Q} = \text{diag} \left[1.764 \cdot 10^{-5} \quad 1.8279 \cdot 10^{-8} \quad 3.844 \cdot 10^{-5} \quad 5.29 \cdot 10^{-6} \quad 5.29 \cdot 10^{-6} \quad 1.053 \cdot 10^{-7} \right]. \quad (22)$$

The measurement noise covariance values come from expanding the initial simulated coplanar POSIT test to numerous more iterations during the glideslope descent:

$$\mathbf{R} = \text{diag} \left[2.329 \quad 0.0052 \quad 0.0625 \quad 1.218 \cdot 10^{-7} \quad 3.046 \cdot 10^{-8} \quad 2.467 \cdot 10^{-6} \right]. \quad (23)$$

The initial covariance matrix, \mathbf{P}_0 , contains values of 1000 along the diagonal to provide significant initial uncertainties to ensure quick convergence. The IMU measurements, coplanar POSIT pose measurements, and the EKF time steps all use a timestep of 0.01 seconds to keep time synchronization simple. It takes approximately 0.2 ms per iteration, which demonstrates real-time capabilities for onboard implementation.

1. Covariance

Figure 9 shows the position and velocity error covariances, which quickly converge to zero. Similarly, Fig. 10 shows that the Euler angle error covariances also converge to zero. Since all the state variables' covariances converge to zero, this demonstrates high confidence in the state estimation increases as time marches forward. Quick convergence with small covariance values shows that the proposed EKF design produces confident results.

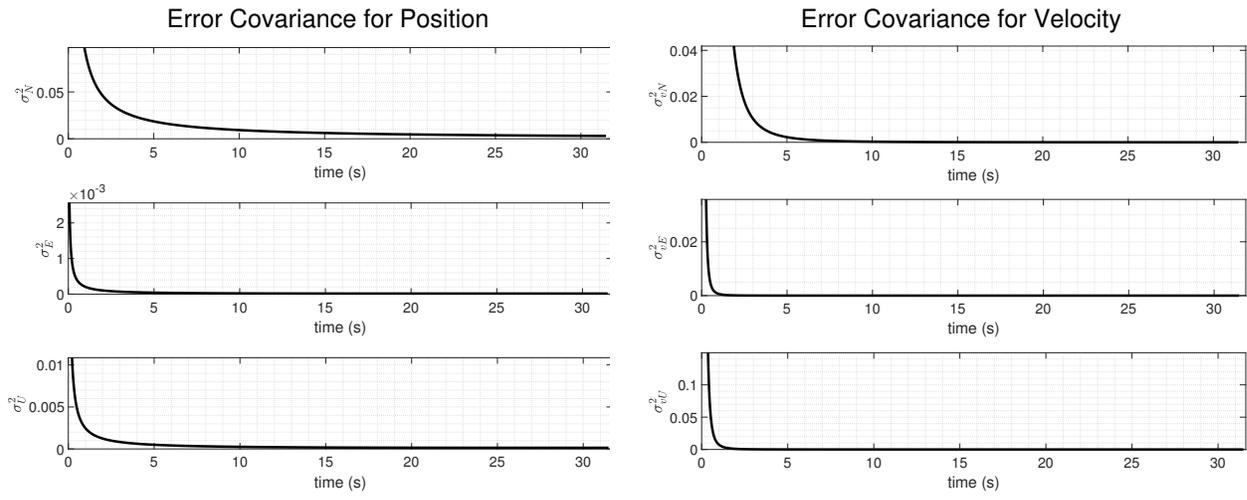


Fig. 9 Error Covariance: Position & Velocity

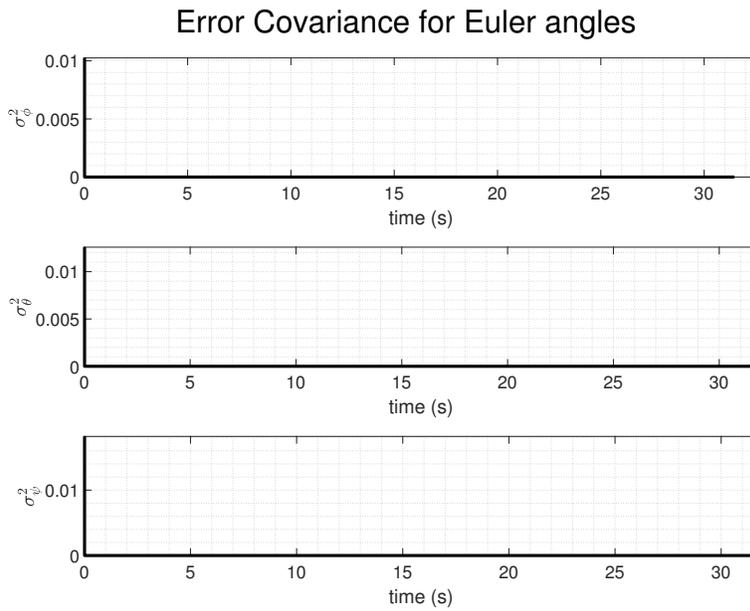


Fig. 10 Error Covariance: Euler Angles

2. State Estimation

Figure 11 shows the position and velocity state estimates, which align closely with the black line, nominal path (A→B in Fig. 5). Figure 12 shows the aircraft Euler angle camera estimates, which have tiny fluctuations in the roll and pitch angles. There is a positive bias in the pitch angle estimation but as much in the roll angle estimation. Overall, the state estimation is fairly accurate despite the North estimation average and standard deviation for v_N . Table 3 shows the state estimation statistics.

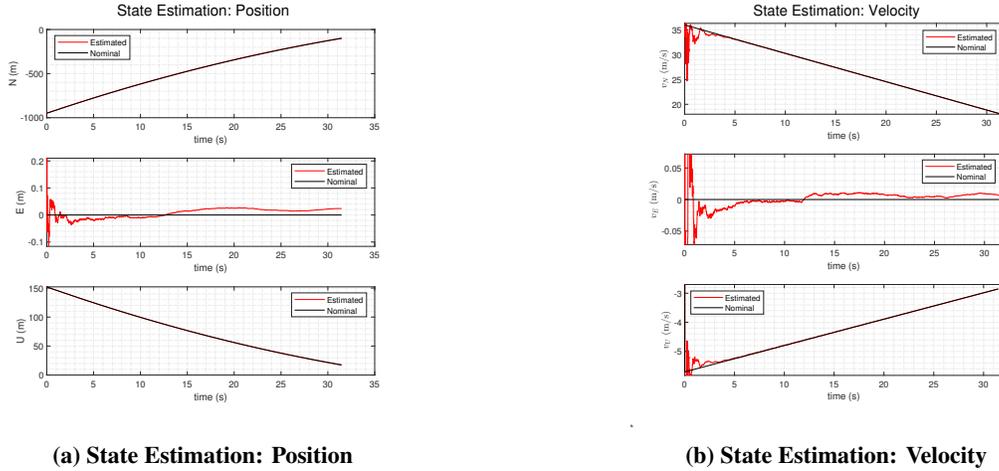


Fig. 11 State Estimation: Position & Velocity

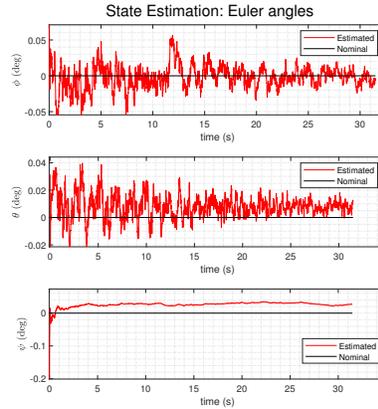


Fig. 12 State Estimation: Euler Angles

Table 3 State Estimation Statistics

	N	E	U	v_N	v_E	v_U	ϕ	θ	ψ
μ	1.9136	0.01690	0.02258	0.2095	0.02556	0.05556	0.000212	0.000165	0.000467
σ	0.2549	0.009755	0.04175	1.299	0.3620	0.5992	0.000204	0.000114	0.000115

3. Bounded Error

Figure 13 shows the position error bounded by $\pm 2, 3\sigma$, centered around the mean error. The blue lines represent the $\pm 2\sigma$ bounds, while the red lines represent the $\pm 3\sigma$ bounds. An initial glance at the errors seem to yield errors within

the $\pm 2, 3\sigma$ bounds, but zooming in shows that the position errors remain within the $\pm 2, 3\sigma$ for most of the time with small fluctuations outside the boundaries, which demonstrates confidence in the position estimation. Figure 14 shows

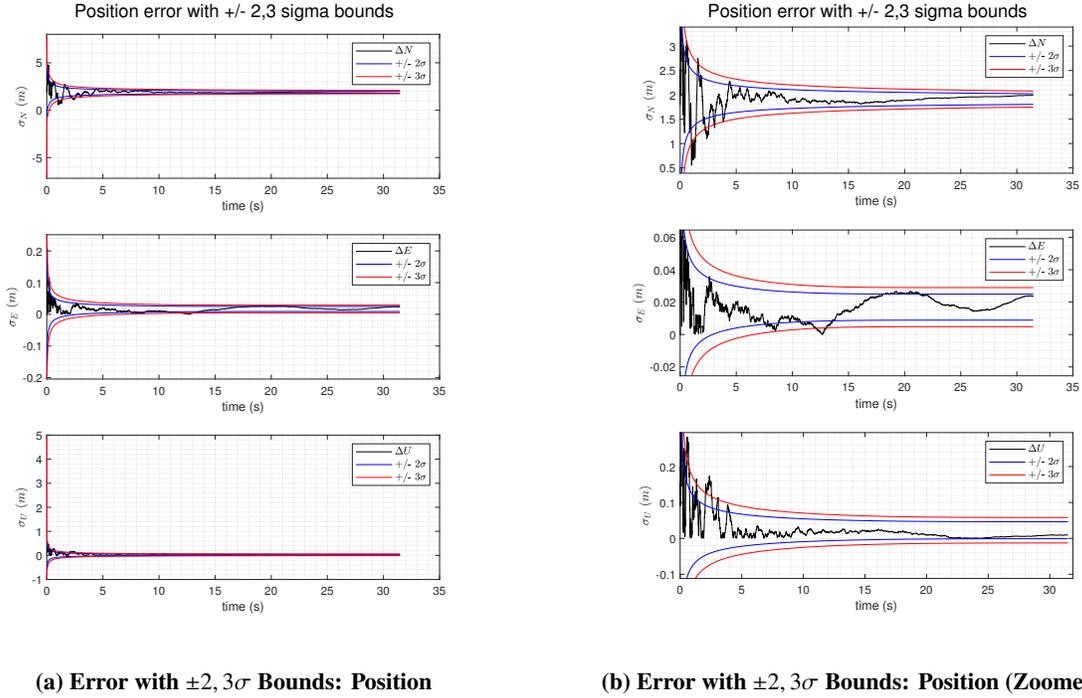


Fig. 13 Error with $\pm 2, 3\sigma$ Bounds: Position (Zoomed In)

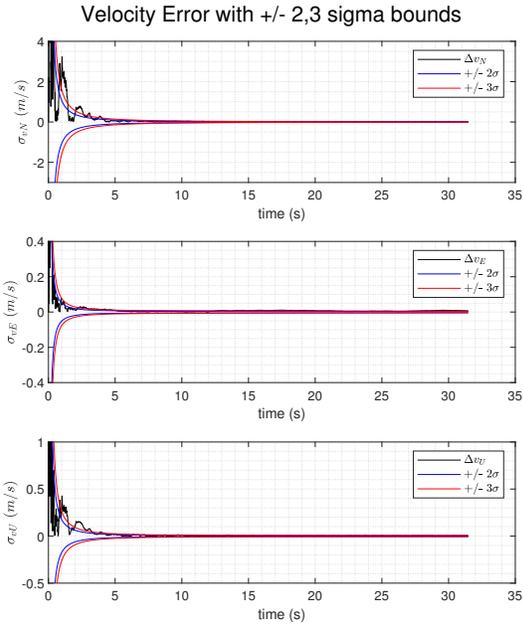
the average velocity error and sigma bounds, which are centered around zero since the mean errors are tiny. A quick glance shows that the mean errors converge quickly. Zooming in shows that the error tends to gravitate towards the $\pm 2, 3\sigma$ bounds with some values outside the $\pm 2, 3\sigma$ bounds. However, the $\pm 2, 3\sigma$ bounds are tiny, so having small errors fall outside these bounds is acceptable. For instance, the last few values for $+3\sigma_{vE}$ are 0.005 m/s , while the error is 0.01 m/s . Figure 15 shows the Euler angle errors and $\pm 2, 3\sigma$ bounds centered around the mean error. Rapid convergence occurs with minimal fluctuations based on the small order of magnitude. Some spikes exceed the $+3\sigma$ bounds for ϕ and θ , but these values are minuscule, i.e., order of 10^{-4} rad . Despite these few spikes, the errors generally stay within the $\pm 2, 3\sigma$ bounds, demonstrating confidence in the Euler angle state estimation computations.

C. Glideslope & Localizer Error

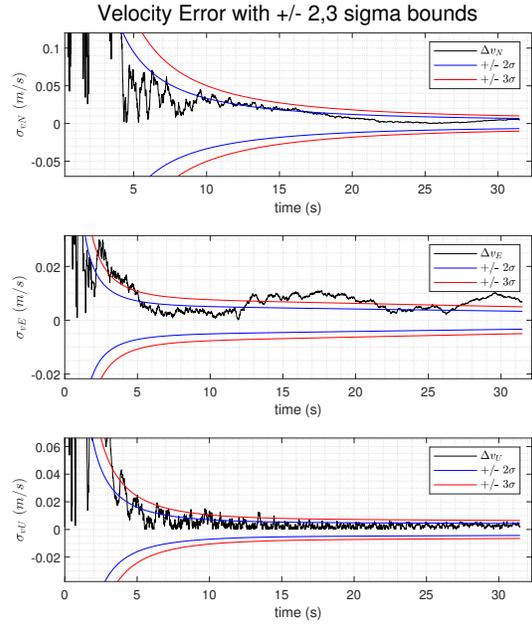
Confident and accurate EKF state estimations lead to accurate glideslope and localizer error computations. Figure 16 depicts a geometric representation of the aircraft's distance from the glidepath to compute the glideslope and localizer error. The 2D back plane is in the UE plane, and the side plane is in the UN plane. The glidepath vector, \mathbf{gp} , is the vector from P to O and yields the shortest path to return to the glidepath because \mathbf{gp} is perpendicular to \mathbf{AO} . It breaks down into three components:

$$\mathbf{gs} = \begin{bmatrix} \mathbf{gp}_{lat,E} & \mathbf{gp}_{lat,N} & \mathbf{gp}_v \end{bmatrix} = \begin{bmatrix} \mathbf{gp}_{lat} & \mathbf{gp}_v \end{bmatrix} \quad (24)$$

in which $\mathbf{gp}_{lat,E}$ is the East component of \mathbf{gp}_{lat} , $\mathbf{gp}_{lat,N}$ is the North component of \mathbf{gp}_{lat} , and \mathbf{gp}_v is the vertical component of \mathbf{gp} . The glideslope error is the magnitude of \mathbf{gp}_v , while the localizer error is the magnitude of $\mathbf{gp}_{lat,E}$. Figures 17a-17b show the 2D view of the aircraft's position, P, relative to the glidepath and the components of \mathbf{gp} .

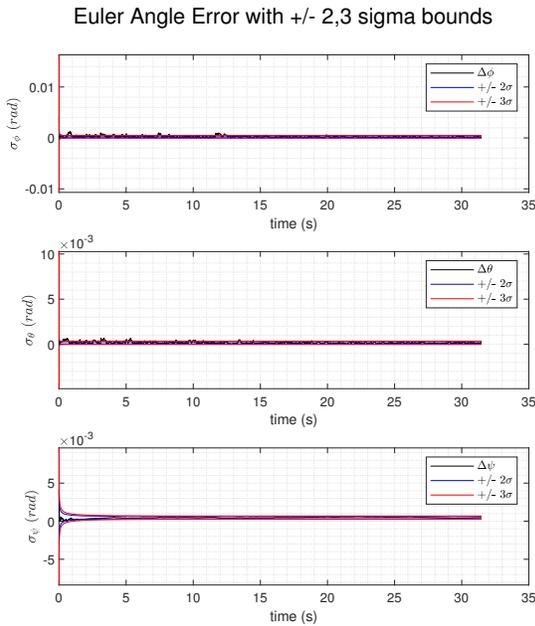


(a) Error with $\pm 2, 3\sigma$ Bounds: Velocity

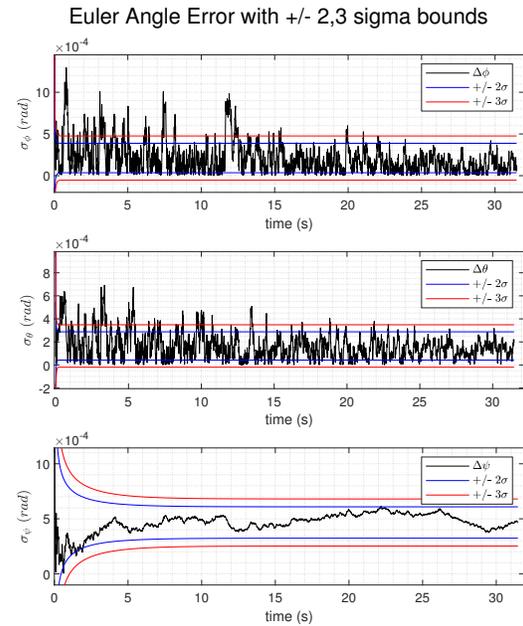


(b) Error with $\pm 2, 3\sigma$ Bounds: Velocity (Zoomed In)

Fig. 14 Error with $\pm 2, 3\sigma$ Bounds: Velocity



(a) Error with $\pm 2, 3\sigma$ Bounds: Euler Angles



(b) Error with $\pm 2, 3\sigma$ Bounds: Euler Angles (Zoomed In)

Fig. 15 Error with $\pm 2, 3\sigma$ Bounds: Euler Angles

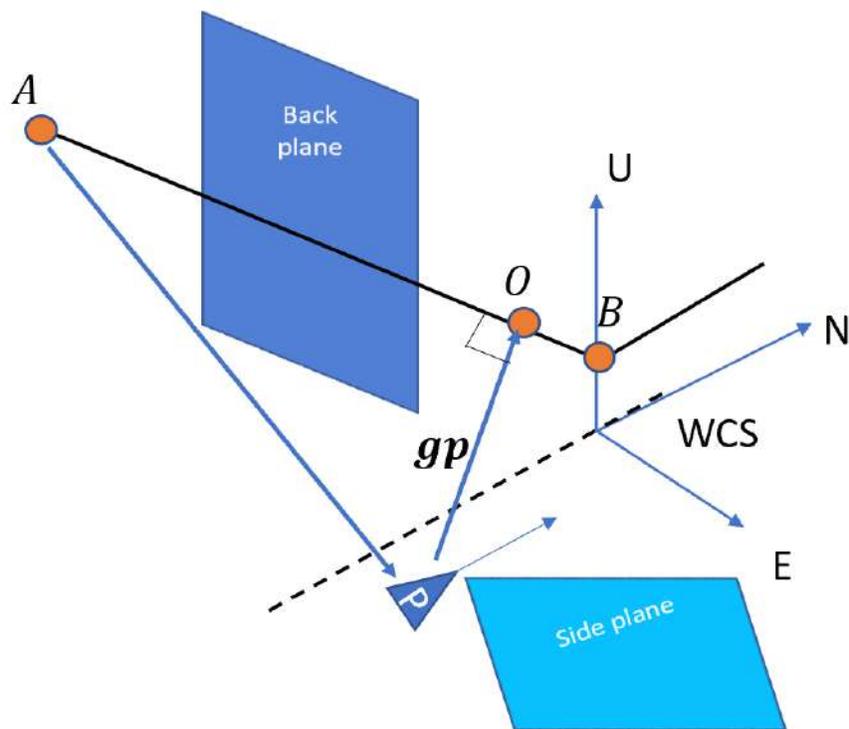
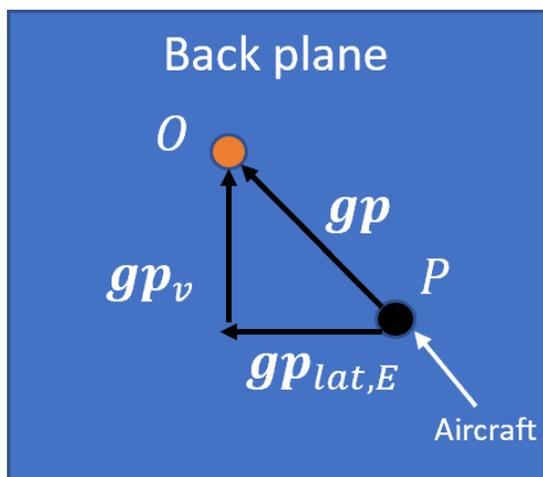
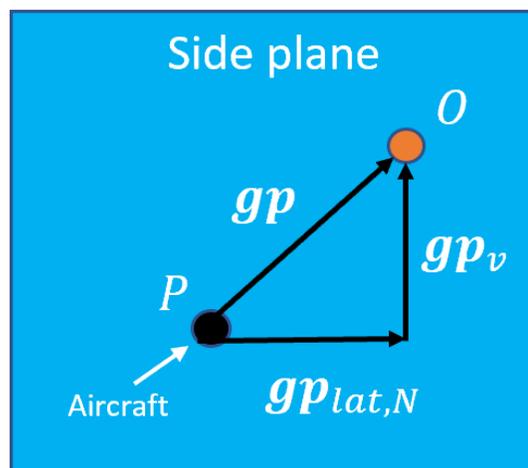


Fig. 16 Glideslope/Localizer Diagram (WCS): P is the aircraft, A is a fixed point at the top of the glideslope, B is a fixed point at the bottom of the glideslope, and O is the closest point on the glidepath from P. The points, A and B, have the same WCS coordinates as in Fig. 5.



(a) Glideslope/Localizer Diagram - Back Plane



(b) Glideslope/Localizer Diagram - Side Plane

Fig. 17 Glideslope/Localizer Diagrams

Figure 18 shows a block diagram of the glideslope and localizer error computations. First, the vector, \mathbf{AB} is constant and computed along with its unit vector, \mathbf{ab} . Then, the vector, \mathbf{AP} can be calculated based on the WCS position estimate from the EKF state estimation results. Next, the vector, \mathbf{AO} can be computed by:

$$\mathbf{AO} = (\mathbf{AP} \cdot \mathbf{ab})(\mathbf{ab}). \quad (25)$$

Then, the glidepath vector, \mathbf{gp} is the difference between \mathbf{AO} and \mathbf{AP} :

$$\mathbf{gp} = \mathbf{AO} - \mathbf{AP}. \quad (26)$$

The glidepath vector's vertical component can then be computed by dotting with $\hat{\mathbf{k}} = [0 \ 0 \ 1]^T$:

$$\mathbf{gp}_v = (\mathbf{gp} \cdot \hat{\mathbf{k}})\hat{\mathbf{k}}, \quad (27)$$

which yields the glideslope error. Finally, the localizer error comes from the computation of the glidepath vector's lateral component:

$$\mathbf{gp}_{lat} = \mathbf{gp} - \mathbf{gp}_v. \quad (28)$$

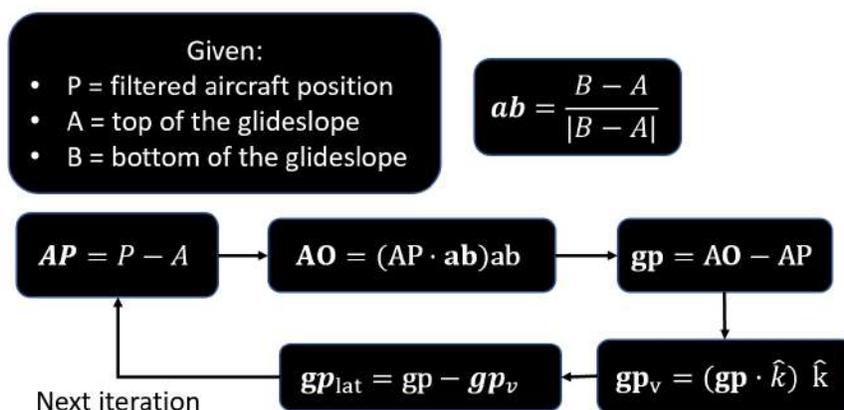


Fig. 18 Glideslope/Localizer - Flow Chart

Figure 19 shows the glideslope and localizer errors based on the EKF position state estimates. The glideslope error diverges due to minor errors that accumulate over time. The localizer error has some minor fluctuations but tends to be very small due to the highly accurate East estimations. This paper does not include guidance laws during descent to steer the aircraft back onto the nominal glidepath utilizing feedback control. This paper focuses on navigation, so future work involves deriving and implementing guidance laws to bring the aircraft back onto the glidepath.

D. X-Plane Simulation

As nighttime AAM operations and technologies progress forward, future studies may need to monitor health and alertness for long-term exposure to the vertiport landing lights [36]. X-Plane is a flight simulator software with aircraft, airports, and scenery packages to resemble mock flights in the real world. X-Plane has a 2D airport editor called World Editor, allowing users to create and modify airports and scenery. Importing orthophotos of tiles around the globe into World Editor and then exporting them into X-Plane provides higher quality images of the scenery. Importing orthophotos and the landing light configuration into World Editor and X-Plane provides a high-fidelity AAM PAL simulation. The NASA Data & Reasoning Fabric (DRF) ^{††} selected eleven vertiport locations for their simulations, and two of them are Fifth and Mission Garage (San Francisco) and Middle Harbor Shoreline Park (Oakland). Figure 20a shows an example of importing an orthophoto of a portion of Middle Harbor Shoreline Park (Oakland, CA) into World Editor. See Fig. 20b for rendering the landing light configuration as green plus signs in World Editor. Figure 22 shows an X-Plane 9° camera tilt down view at daytime and nighttime view of the descent at Middle Harbor Shoreline Park, starting in San Francisco at Fifth and Mission Garage (see Fig. 21 for their locations relative to the San Francisco International Airport

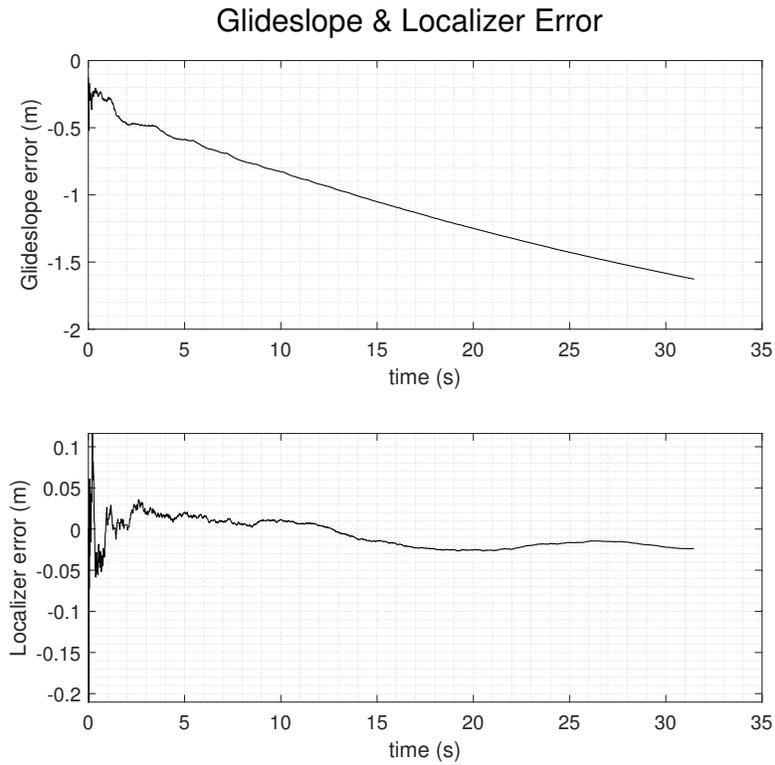


Fig. 19 Glideslope & Localizer Error vs. Time



(a) X-Plane World Editor Middle Harbor Shoreline Park Orthophoto



(b) X-Plane World Editor Middle Harbor Shoreline Park Veriport Lights and Helipad (green plus signs)

Fig. 20 X-Plane Middle Harbor Shoreline Park Veriport Landing Simulation



Fig. 21 Fifth & Mission Garage and Middle Harbor Shoreline Park DRF Vertiports Relative to the San Francisco International Airport (Google Earth)

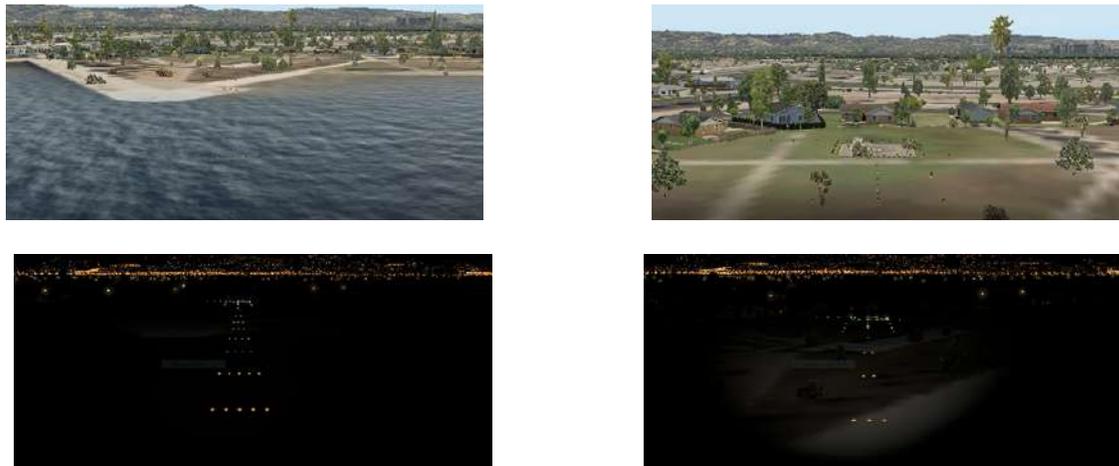


Fig. 22 X-Plane Day & Night Approach and Landing at Middle Harbor Shoreline Park, Oakland, CA

in Google Earth). Future work includes determining the camera tilt down specifications and conducting further analysis and investigation on adequate computer vision methods to determine feature correspondence, i.e., finding the lights in pixel coordinates in the images to feed into coplanar POSIT.

VII. Conclusion

This paper demonstrates a vision-based navigation solution for AAM PAL by fusing VMS IMU telemetry data and the coplanar POSIT algorithm in an extended Kalman filter. State estimation results demonstrate high accuracy, quick covariance convergence shows increased confidence in the state estimation, and short runtime offers AAM a baseline and real-time vision-based navigation solution for glideslope descent. Future work includes augmenting the high-fidelity PAL simulation in X-Plane and World Editor with guidance laws for steering the aircraft back onto the glidepath. Other future work includes feature correspondence to determine the landing lights in pixel coordinates to insert into coplanar POSIT and the other blue blocks in Fig. 1, which will lead to real-time integrated guidance, navigation, and control simulation for AAM PAL.

^{††}<https://drf.nasa.gov/>

Acknowledgments

The authors are very grateful to Daniel DeMenthon for the insightful and helpful discussions regarding the POSIT and SoftPOSIT algorithms. The authors would also like to thank Uland Wong and Xavier Bouyssounouse from NASA Ames Research Center for the insightful and helpful discussions regarding computer vision and rendering. Finally, the authors would like to thank these NASA Ames Research Center projects: Vertical Motion Simulator (VMS) for sharing simulated AAM aircraft data and the Data & Reasoning Fabric (DRF) project for sharing vertiport locations and trajectory planning.

References

- [1] Clearman, B., *Transportation-Markings Database: Aeronautical Navigation Aids*, Mount Angel Abbey, 2001.
- [2] Wang, Z., Wang, S., Zhu, Y., and Xin, P., "Assessment of ionospheric gradient impacts on ground-based augmentation system (GBAS) data in Guangdong province, China," *Sensors*, Vol. 17, No. 10, 2017, p. 2313.
- [3] Cohen, C. E., Pervan, B. S., Lawrence, D. G., Cobb, H. S., Powell, J. D., and Parkinson, B. W., "Real-time flight testing using integrity beacons for GPS category III precision landing," *Navigation, Journal of the Institute of Navigation*, Vol. 41, No. 2, 1994, pp. 145–158.
- [4] Cohen, C. E., Cobb, H. S., Lawrence, D. G., Pervan, B. S., Powell, J. D., Parkinson, B. W., Aubrey, G. J., Loewe, W., Ormiston, D., McNally, B. D., et al., "Autoland a 737 using GPS integrity beacons," *Navigation, Journal of the Institute of Navigation*, Vol. 42, No. 3, 1995, pp. 467–486.
- [5] Mesloh, C., Henych, M., Wolf, R., and Gallatin, K., "Infrared beacon evaluation: applications for law enforcement," *Annotation*, 2008.
- [6] Nowak, E., Gupta, K., and Najjaran, H., "Development of a plug-and-play infrared landing system for multirotor unmanned aerial vehicles," *2017 14th Conference on Computer and Robot Vision (CRV)*, IEEE, 2017, pp. 256–260.
- [7] Javed, M. I., Afghani, S., and Shabbir, J., "Post Landing Automated Guidance System of Commercial Aircrafts using Infrared Beacons," *International Journal of Science and Technology*, Vol. 3, No. 4, 2013.
- [8] Gautam, A., Sujit, P., and Saripalli, S., "A survey of autonomous landing techniques for UAVs," *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, 2014, pp. 1210–1218.
- [9] Herisse, B., Russotto, F.-X., Hamel, T., and Mahony, R., "Hovering flight and vertical landing control of a VTOL unmanned aerial vehicle using optical flow," *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2008, pp. 801–806.
- [10] Min, B.-M., Tahk, M.-J., Shim, H.-C. D., and Bang, H.-C., "Guidance law for vision-based automatic landing of UAV," *International Journal of Aeronautical and Space Sciences*, Vol. 8, No. 1, 2007, pp. 46–53.
- [11] Shakernia, O., Ma, Y., Koo, T. J., and Sastry, S., "Landing an unmanned air vehicle: Vision based motion estimation and nonlinear control," *Asian Journal of Control*, Vol. 1, No. 3, 1999, pp. 128–145.
- [12] Ye, S., Wan, Z., Zeng, L., Li, C., and Zhang, Y., "A vision-based navigation method for eVTOL final approach in urban air mobility (UAM)," *2020 4th CAA International Conference on Vehicular Control and Intelligence (CVCI)*, IEEE, 2020, pp. 645–649.
- [13] Cesetti, A., Frontoni, E., Mancini, A., Zingaretti, P., and Longhi, S., "A vision-based guidance system for UAV navigation and safe landing using natural landmarks," *Journal of Intelligent and Robotic Systems*, Vol. 57, No. 1, 2010, pp. 233–257.
- [14] Mejias, L., Campoy, P., Saripalli, S., and Sukhatme, G. S., "A visual servoing approach for tracking features in urban areas using an autonomous helicopter," *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, IEEE, 2006, pp. 2503–2508.
- [15] Denham, J. W., "Project MAGIC CARPET: "Advanced Controls and Displays for Precision Carrier Landings",," *54th AIAA Aerospace Sciences Meeting*, 2016, p. 1770.
- [16] Zhang, X., Zhang, Z., Li, Y., Zhu, X., Yu, Q., and Ou, J., "Robust camera pose estimation from unknown or known line correspondences," *Applied Optics*, Vol. 51, No. 7, 2012, pp. 936–948.

- [17] Gurdjos, P., Dalle, P., and Castan, S., “Tracking 3D coplanar points in the invariant perspective coordinates plane,” *Proceedings of 13th International Conference on Pattern Recognition*, Vol. 1, IEEE, 1996, pp. 493–497.
- [18] Kim, Y., and Hwang, D.-H., “Loosely-coupled vision/INS integrated navigation system,” *Journal of Positioning, Navigation, and Timing*, Vol. 6, No. 2, 2017, pp. 59–70.
- [19] Hartley, R., and Zisserman, A., *Multiple View Geometry in Computer Vision*, 2nd ed., Cambridge University Press, 2003.
- [20] Li, H., and Hartley, R., “Five-point motion estimation made easy,” *18th International Conference on Pattern Recognition (ICPR’06)*, Vol. 1, IEEE, 2006, pp. 630–633.
- [21] Arun, K. S., Huang, T. S., and Blostein, S. D., “Least-squares fitting of two 3-D point sets,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, , No. 5, 1987, pp. 698–700.
- [22] DeMenthon, D. F., and Davis, L. S., “Model-based object pose in 25 lines of code,” *International Journal of Computer Vision*, Vol. 15, No. 1-2, 1995, pp. 123–141.
- [23] Oberkampf, D., DeMenthon, D. F., and Davis, L. S., “Iterative pose estimation using coplanar feature points,” *Computer Vision and Image Understanding*, Vol. 63, No. 3, 1996, pp. 495–511.
- [24] David, P., Dementhon, D., Duraiswami, R., and Samet, H., “SoftPOSIT: Simultaneous pose and correspondence determination,” *International Journal of Computer Vision*, Vol. 59, No. 3, 2004, pp. 259–284.
- [25] Federal Aviation Administration, “AC 150/5390-3 (Cancelled) - Vertiport Design,” U.S. Department of Transportation, 2010. URL https://www.faa.gov/documentLibrary/media/advisory_circular/150-5390-3/150_5390_3.pdf.
- [26] Federal Aviation Administration, “AC 150/5390-2C - Heliport Design,” U.S. Department of Transportation, 2012. URL https://www.faa.gov/documentLibrary/media/Advisory_Circular/150_5390_2c.pdf.
- [27] National Academies of Sciences, Engineering, and Medicine, *Advancing Aerial Mobility: A National Blueprint*, The National Academies Press, 2020.
- [28] Schaub, H., and Junkins, J. L., *Analytical Mechanics of Space Systems*, 4th ed., AIAA, 2018.
- [29] Chu, P., Mulder, J. A. B., and Breeman, J., “Real-time identification of aircraft physical models for fault tolerant flight control,” *Fault Tolerant Flight Control*, Springer, 2010, pp. 129–155.
- [30] Lombaerts, T., Kaneshige, J., Schuet, S., Aponso, B. L., Shish, K. H., and Hardy, G., “Dynamic Inversion based Full Envelope Flight Control for an eVTOL Vehicle using a Unified Framework,” *AIAA Scitech 2020 Forum*, 2020, p. 1619.
- [31] Szeliski, R., *Computer Vision: Algorithms and Applications*, Springer Science & Business Media, 2010.
- [32] Webber, D., and Zahn, D., “FAA and the National Campaign,” [Powerpoint], 2021.
- [33] Stepanyan, V., Lombaerts, T., Shish, K. H., and Cramer, N. B., “Adaptive Multi-Sensor Information Fusion For Autonomous Urban Air Mobility Operations,” *AIAA Scitech 2021 Forum*, 2021, p. 1115.
- [34] Gelb, A., *Applied Optimal Estimation*, MIT press, 1974.
- [35] Simon, D., *Optimal State Estimation: Kalman, H ∞ , and Nonlinear Approaches*, John Wiley & Sons, 2006.
- [36] Bhagavathula, R., Gibbons, R., Hanifin, J., and Brainard, G., “LED Roadway Lighting: Impact on Driver Sleep Health and Alertness,” *NCHRP Research Report*, , No. 968, 2021.