Simulated Vision-based Approach and Landing System for Advanced Air Mobility

Evan Kawamura *[1], Chester Dolph †[2], Keerthana Kannan ‡[1], Thomas Lombaerts §[1], and Corey Ippolito ¶[1]
[1]*NASA Ames Research Center, Moffett Field, Mountain View, CA 94035*
[2]*NASA Langley Research Center, 1 Nasa Dr, Hampton, VA 23666*

Advanced Air Mobility (AAM) aircraft require precision approach and landing systems (PALS) in several environments, such as urban, suburban, and rural. It is challenging to implement current state-of-the-art methods approved for automated approach and landing for AAM operations with challenges such as GPS degradation in urban environments and visual navigation aids like the glideslope and localizer being narrow and not allowing alternative incoming landing angles at vertiports. However, existing technology and systems, i.e., the instrument landing system (ILS) with glideslope and localizer indicators that use vision, IR, radar, or GPS methods, provide baseline perception and sensing requirements for AAM aircraft approach and landing. This paper focuses on vision-based PAL and computer vision feature correspondence methods to demonstrate a baseline navigation system while adhering to the Federal Aviation Administration requirements and regulations about heliport design (FAA AC 150/5390-2C), which is one of the closest references for vertiport requirements and regulations. The coplanar pose from orthography and scaling with iterations (COPOSIT) algorithm determines pose estimation, which feeds into an Extended Kalman filter that combines IMU with vision to create a vision-based approach and landing (VAL) sensor fusion navigation solution for GPS-denied environments. The VAL navigation solution provides promising simulation results for AAM PALS with Hough circle detection and feature correspondence, which demonstrate robustness to false positives. This paper incorporates moderately high-fidelity simulations with computer graphics rendering to show a distributed sensor network to track an AAM aircraft during approach and landing to compare with the aircraft's onboard vision-based navigation solution.

I. Introduction

Advanced Air Mobility (AAM) utilizes revolutionary aircraft models from emerging aviation markets to safely transport people and cargo in urban, suburban, rural, and regional environments. These AAM aircraft benefit the public and air transportation sectors, so including autonomy increases their performance capabilities and efficiency. Implementing autonomy onboard replaces human pilots, which increases payload capacity. Without human pilots, AAM aircraft will need remote pilots or a safe, reliable, accurate, and autonomous approach and landing system. Current state-of-the-art automated precision approach and landing (PAL) methods may not be suitable for AAM operations. AAM PAL can derive baseline perception and sensing requirements from existing technology, sensors, and systems such as vision, IR, radar, glideslope indicators, and GPS.

The FAA does not have an approved advisory circular for vertiports. However, one of the closest approved references for vertiport requirements and landing light configuration is their document on heliport design (FAA AC 150/5390-2C [1]. Another reference for vertiports and light configuration is the FAA Engineering Brief #105, Vertiport Design. It provides interim guidance for vertiport design, so requirements and standards will change before generating an advisory circular for vertiport design [2].

*Computer/GNC Engineer, Intelligent Systems Division, NASA Ames Research Center, Moffett Field, Mountain View, CA 94035, USA.
†Aerospace Engineer, Aeronautics Systems Engineering Branch, NASA Langley Research Center, 1 Nasa Dr, Hampton, VA 23666.
‡Software 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

A. Traditional and Current Landing Systems

Several types of traditional and current aircraft landing systems exist for PAL applications. Instrument Landing Systems (ILS) utilize navigation aids with three components: glideslope, localizer, and marker beacons to assist and guide aircraft during approach and landing. The glideslope provides a vertical offset from the nominal glidepath, while the localizer provides a horizontal offset from the nominal glidepath. Finally, the ILS marker beacons provide radio checks throughout the aircraft's descent [3].

Visual Approach Slope Indicator (VASI) systems are another form of visual navigation aids that gives approach slope information during aircraft descent. A common theme for VASI systems utilizes white light beams for flying above the glidepath and red light beams for flying below the glidepath. The Precision Approach Path Indicator (PAPI) systems resemble and replace the older VASI systems. They also have more red than white lights to indicate flying below the glidepath and more white than red lights to indicate above the glidepath. An equal distribution between the red and white lights means flying on the glidepath. There are variations of glidepath lights such as tri-color: red for below, green for on, and yellow for above the glidepath, and pulsating lights with red lights for well below the glidepath, pulsating white lights for well above the glidepath, and steady red or white for smaller glideslope offsets [3].

Ground Based Augmentation Systems (GBAS) provide an alternative landing method to ILS. and the FAA helped validate the International Civil Aviation Organization standards and recommended practices for GBAS Approach Service Type-D for CAT III approaches. Vendors seeking FAA approval for GAST-D GBAS must adhere to these standards, which were approved in 2018 [4]. GBAS has more advantages than ILS with Viseu airfield as a case study. Compared to ILS, GBAS has more straightforward installation methods and is more practical since just one GBAS station can handle several approach angles, i.e., multiple runways at different angles. GBAS also works well in poor weather conditions and allows simultaneous approaches, which increases flexibility and economic benefits for airports. There are some circumstances where ILS performs better than GBAS, such as poor GNSS signals [5]. Even though GBAS has many advantages over ILS, it might not be suitable for AAM due to GPS degradation in urban environments.

Some studies utilize GPS and IR for aircraft navigation and landing. The local area augmentation system improves airborne accuracy to less than 1 m and guarantees aircraft integrity and continuity during approach by augmenting GPS [6]. Autopilot designs based on GPS have advantages over ILS methods, such as being less expensive, more flexible by not being confined to a narrow beam, and three-dimensional GPS sensor measurements work better in an autopilot system [7]. An alternative triangulation method with IR beacons without any mechanical moving components increases system robustness and reduces the complexity [8]. Another study utilizes a 3-point IR-guided landing pad system for autonomously landing unmanned aerial vehicles that contain a monocular camera and an IR filter [9].

Overall, it is difficult to implement traditional visual navigation aids like the glideslope and localizer in confined urban environments because their beams are narrow and not flexible to accommodate alternative incoming landing angles at vertiports [10]. It is also challenging for visual navigation aids to maintain sufficient coverage in a dense city with several buildings, which makes it geometrically complex to maintain a consistent glidepath angle during approach and landing. Therefore, Alternative Precision Navigation and Timing (APNT) solutions are crucial for AAM PAL in urban environments with GPS degradation [11].

B. Computer Vision Research

There are computer vision studies that extract the fundamental and essential matrices for pose estimation. Estimating the fundamental matrix provides information to compute the essential matrix to compute the rotation matrix and translation vector between two images. 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 are some methods that estimate the fundamental matrix [12]. The 5-point algorithm with RANSAC computes the essential matrix to obtain the transformation between two images [13]. Normalizing the image coordinates for two images allows computation of the essential matrix with four possible solutions for the second camera matrix, which contains the relative transformation between the two images [12].

Other computer vision studies directly estimate the rotation matrix and translation vector. The relative transformation between two sets of points can be computed through a least squares fit of two 3D sets of points with SVD [14]. The Pose from Orthography and Scaling with Iterations (POSIT) algorithm uses an image's orthographic projections with noncoplanar feature points of an object [15]. Constraining the POSIT algorithm for images with coplanar points leads to a coplanar POSIT algorithm with a limiting factor of having at least four coplanar points [16]. The SoftPOSIT algorithm merges POSIT with softassign, which solves the image correspondence problem between object and image points when correspondences are unknown. Then, POSIT iteratively computes the pose estimation by providing the

transformation from the camera's center to the object's origin [17].

Other studies combine computer vision with Kalman filtering techniques for estimating states. Combining the coplanar POSIT algorithm's pose estimation with IMU data into an extended Kalman filter (EKF) estimates an AAM aircraft's state during approach and landing [18]. Implementing a Kalman filter that estimates the camera pose and 3D to 2D line correspondences has accurate and efficient performance for the model-to-image registration problem [19]. Another study estimates the position of 3D coplanar points with an EKF when there are four stationary reference points with known coordinates [20]. A landmark-based vision method with an INS forms a loosely-coupled Kalman filter navigation solution that can operate in GPS-denied environments [21].

C. Vision-Based Navigation

There is vision-based navigation research for unmanned aerial vehicles (UAVs). An autonomous vision-based navigation strategy for unmanned aerial vehicles utilizes natural landmarks to obtain a visual memory of the environment [22]. A long-term vision-based navigation system for micro helicopters has different sensor modalities, self-calibrates, and is robust to failure modes [23]. A UAV flying at low altitude builds and updates a virtual 3D model of its environment with onboard cameras to fly from an initial position to a terminal position [24]. Combining stereo odometry with IMU measurements provides autonomous capabilities for a quadrotor to fly inside and outside a building through a window and a door [25].

There are some vision-based navigation methods for AAM. One study analyzes enhanced vision systems for urban air mobility approach and landing operations with simulations and requirements for approach trajectories [26]. An AI-enhanced vision navigation architecture provides compatibility with certification requirements for realistic AAM operations to enable accurate and robust performance [27]. One study demonstrates an eVTOL vehicle landing for the final approach with a vision-based closed-loop feedback method and generates accuracies less than 1.5 m when less than 25 m away from the target [**?**].

D. Overview of Work

This paper presents a novel Alternative Position Navigation and Timing vision-based navigation solution for AAM approach and landing by localizing around known feature points in the scenery. It builds on the work of Ref. [18] by including:

1) moderately high-fidelity simulation for AAM PALS with NASA Reflection software and X-Plane & World Editor
2) computer vision feature correspondence methods
3) distributed sensor network that tracks an AAM aircraft during PAL
4) comparison between the onboard AAM aircraft's navigation solution with the distributed sensor network's tracking solution
5) landing lights on the Fifth & Mission Garage (FMG) rooftop
6) curved approach and landing trajectory from the bay, starting near Oakland Bay Bridge and ending at the rooftop of FMG

Figure 1 shows the proposed architecture diagram. The black boxes about simulated vehicle sensor data interface and image processing feature correspondence are updates since Ref. [18], the green boxes work via post-processing, and the blue boxes about pilot GUI and approach/descent mode flight controller are work-in-progress. Figure 2 shows a high-level block diagram of the Vision-based Approach and Landing System (VALS) with the main components' rates, inputs, and outputs. Figure 3 shows a more detailed block diagram of the VALS. The green blocks are variables or inputs, the blue blocks are functions, and the black blocks contain computed or critical output variables.

II. Kinematics and Dynamics

This subsection discusses the state vector, coordinate frames, kinematics, and dynamics for AAM aircraft and follows Ref. [18].

1. State Vector

The state vector of the AAM aircraft is:

$$\boldsymbol{s} = [\ E\ N\ U\ v_E\ v_N\ v_U\ \phi\ \theta\ \psi\ \ ]^T. \tag{1}$$

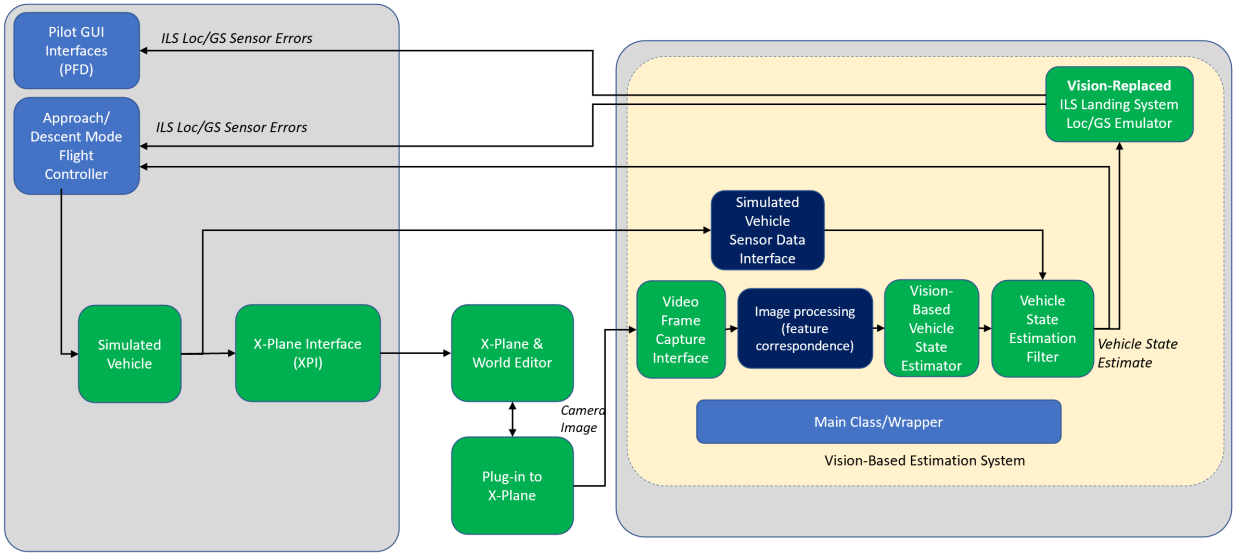


Fig. 1 Proposed Architecture Diagram

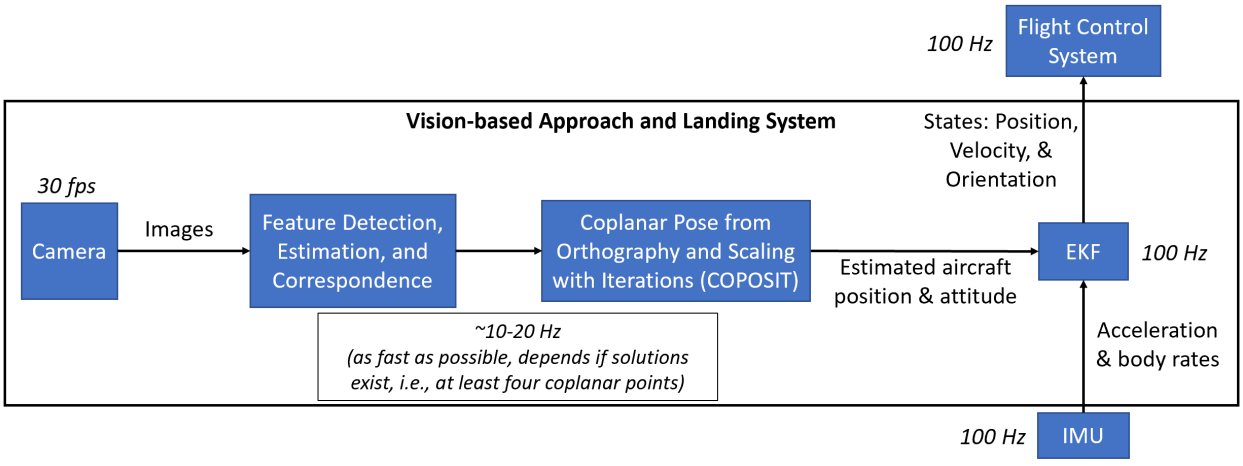


Fig. 2 Vision-based Approach and Landing System (VALS) Block Diagram

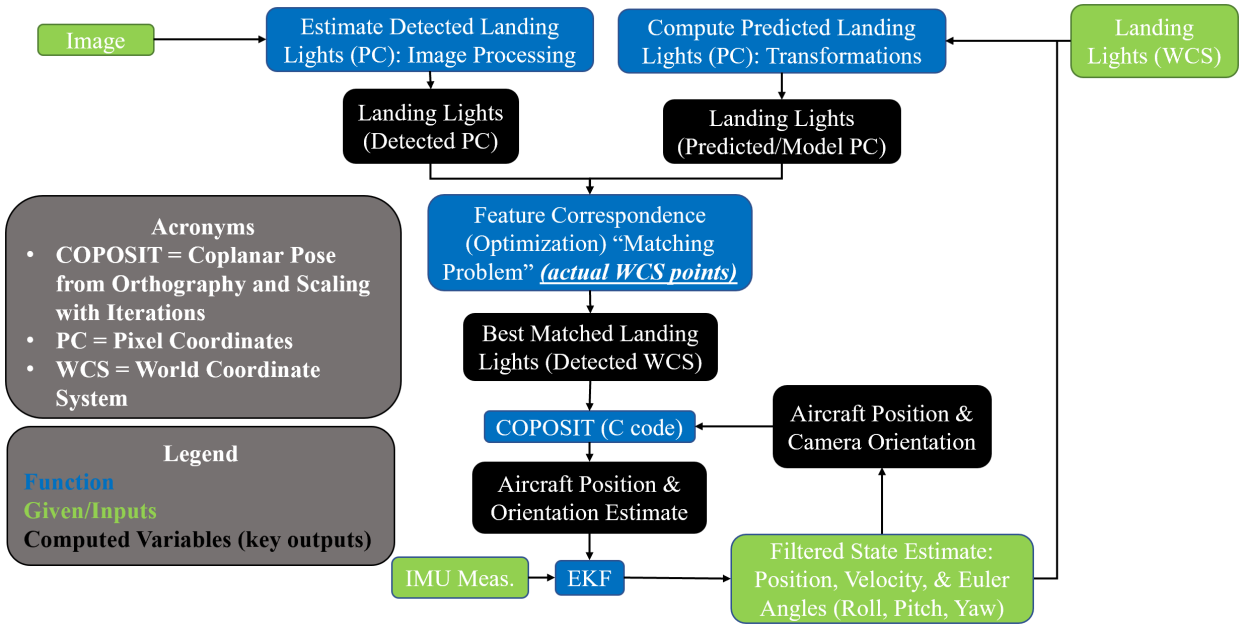


Fig. 3 Vision-based Approach and Landing System (VALS) Detailed Block Diagram

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

$$\boldsymbol{s} = \begin{bmatrix} \boldsymbol{p} & \boldsymbol{v} & \boldsymbol{\Theta} \end{bmatrix}, \tag{2}$$

where each of them is:

$$\boldsymbol{p} = \begin{bmatrix} E & N & U \end{bmatrix}, \quad \boldsymbol{v} = \begin{bmatrix} v_E & v_N & v_U \end{bmatrix}, \quad \boldsymbol{\Theta} = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}. \tag{3}$$

The vector $\boldsymbol{p}$ 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 $\boldsymbol{v}$ are in the inertial frame, while the body frame velocities are: u, v, w. Finally, the Euler angles are the roll, pitch, and yaw angles (ϕ, θ, ψ).

2. Coordinate Frames

The world coordinate system (WCS) is an inertial frame fixed on the ground in which gravity points in the negative U-direction, i.e., down. The vehicle coordinate system (VCS) is on the body frame of 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 4 shows the WCS axes denoted by E, N, U, the VCS axes represented by VCS_x, VCS_y, VCS_z, and the CCS axes marked by CCS_x, CCS_y, CCS_z.

3. Euler Angles

This paper utilizes the direction cosine matrix sequence in Ref. [28] from Appendix C of Ref. [29] to rotate the body-fixed frame (u,v,w) to the inertial ENU frame (WCS) by the roll, pitch, and yaw Euler angles:

$$\mathbf{R} = \begin{bmatrix} s\psi c\theta & c\phi c\psi + s\phi s\psi s\theta & -s\phi c\psi + c\phi s\psi s\theta \\ c\psi c\theta & -c\phi s\psi + s\phi c\psi s\theta & s\phi s\psi + c\phi c\psi s\theta \\ s\theta & -s\phi c\theta & -c\phi c\theta \end{bmatrix}. \tag{4}$$

in which $c\theta$ and $s\theta$ denote $\cos\theta$ and $\sin\theta$ respectively.

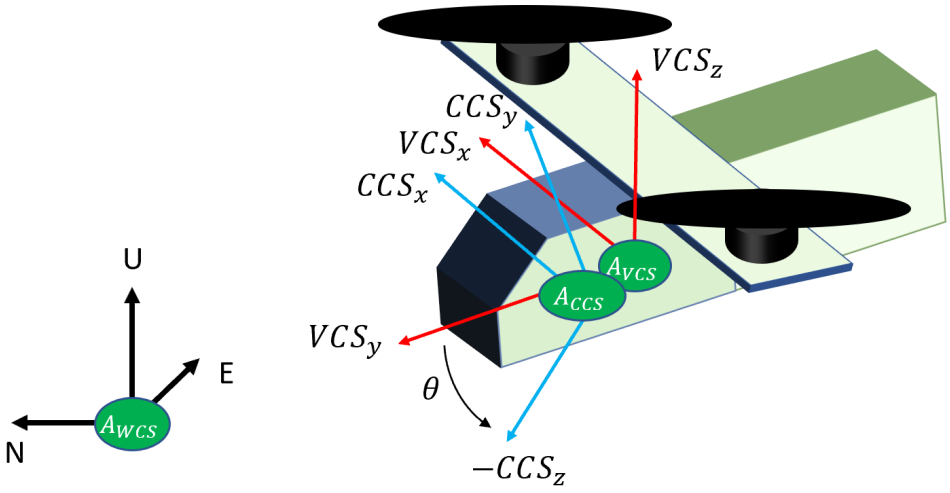


Fig. 4 Inertial, Body, and Camera Frames of AAM Aircraft

Transposing Eqn. (4) yields the rotation from the inertial ENU frame (WCS) to the body-fixed frame (u,v,w). The rotation between the angular velocity and Euler angular rates for the (3-1-2) direction cosine matrix sequence is [30]:

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

in which $\Omega = [rqp]^T$ and $\dot{\Theta} = [\dot{\psi}\dot{\theta}\dot{\phi}]^T$.

4. Translational Dynamics

This subsection follows the translational dynamics for AAM aircraft in Ref. [18]:

$$\begin{aligned} F_x &= m(\dot{u} + qw - rv) + mg\sin\theta \\ F_y &= m(\dot{v} + ru - pw) - mg\cos\theta\sin\phi \\ F_z &= m(\dot{w} + pv - qu) - mg\cos\theta\cos\phi\ , \end{aligned} \tag{6}$$

such that 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 body frame velocities, p, q, r are the body frame angular velocities, and ϕ, θ, ψ are the roll, pitch, and yaw Euler angles. Modeling specific forces as accelerometer measurements at the AAM aircraft's center of gravity yields the specific aerodynamic forces as: [31]

$$F_x = A_x m\ ,\ F_y = A_y m\ ,\ F_z = A_z m \tag{7}$$

such that A_x, A_y, A_z are the accelerometer measurements at the aircraft's center of gravity. Inserting Eq. (6) into Eq. (7) yields:

$$\begin{aligned} \dot{u} &= A_x - g\sin\theta - qw + rv\ , \\ \dot{v} &= A_y - g\cos\theta\sin\phi - ru + pw\ , \\ \dot{w} &= A_z + g\cos\theta\cos\phi - pv + qu. \end{aligned} \tag{8}$$

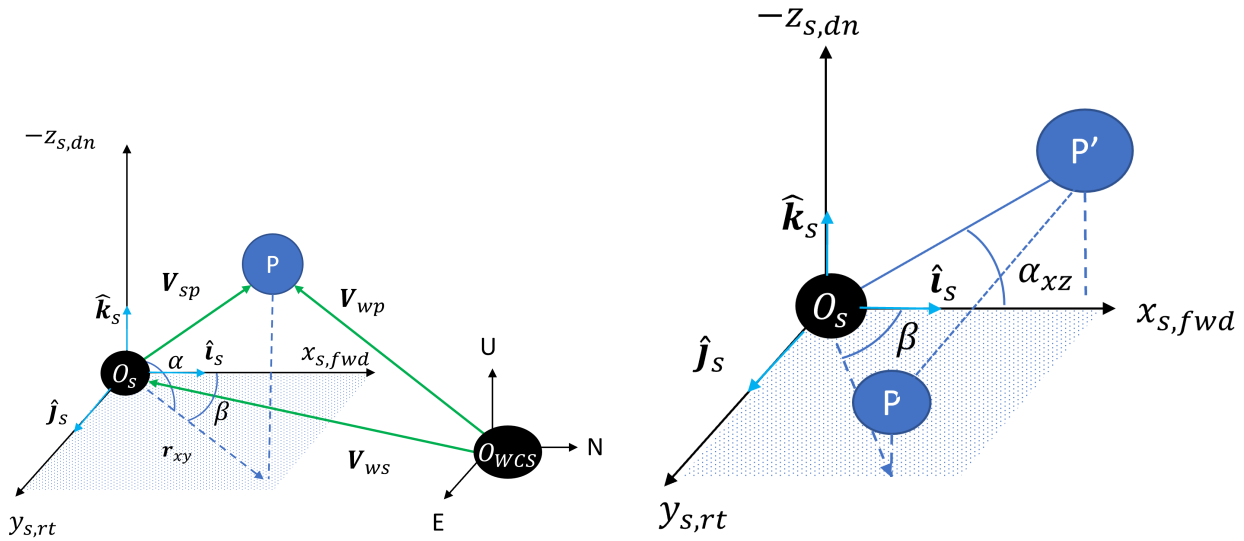
Since Eqn. (8) lacks mass, moments of inertia, and forces, it forms a general set of kinematic equations for all types of AAM aircraft regardless of mass and moments of inertia. Thus, VAL is vehicle agnostic and applies to different AAM aircraft configurations, i.e., multirotor, fixed-wing, lift-plus-cruise, etc.

III. Distributed Cameras: Azimuth and Elevation Angles

Let the East North Up (ENU) world coordinate system, WCS, be defined as an inertial frame fixed on the ground with an origin, O_{WCS}. Let the sensor frame, s, be defined as an x-fwd, y-right, z-down axis system such that its origin O_s is at the center of projection for the sensor with the x-axis pointing in the direction of the sensor, the z-axis pointing down from the point of view of the sensor, and the y axis pointing to the right to complete a right-handed system.

Figure 5a shows the unit vectors, $\hat{\boldsymbol{\imath}}_s, \hat{\boldsymbol{\jmath}}_s, \hat{\boldsymbol{k}}_s$ in the direction of the x, y, z sensor frame directions, respectively. Given a target point P, let $\boldsymbol{V}_{sp}$ be the vector from O_s to the target point P. Let the vector $\boldsymbol{r}_{xy}$ be the projection of $\boldsymbol{V}_{sp}$ to the XY sensor axis plane. Let the azimuth angle β be the angle between $\hat{\boldsymbol{\imath}}_s$ and $\boldsymbol{r}_{xy}$, defined as positive right-hand rotation around $\hat{\boldsymbol{k}}_s$. Let the elevation angle be defined as the angle α between $\boldsymbol{r}_{xy}$ and $\boldsymbol{V}_{sp}$, positive right from $\boldsymbol{r}_{xy}$ to $\boldsymbol{V}sp$ (i.e., positive rotation around the cross-product of $\boldsymbol{r}_{xy} \times \boldsymbol{V}_{sp}$).

Projecting the point onto the X-Z plane yields an elevation angle α_{xz} in the XZ plane, as shown in Figure 5b. This convention resembles the angle of attack and sideslip angles in a body-fixed frame. Ultimately, computing the azimuth



(a) Point in 3D Space: Azimuth and Elevation **(b) Projected Point onto the X-Z Plane: Azimuth and Elevation**

Fig. 5 Azimuth and Elevation Angles

and elevation angles of the target leads to computing the pixel coordinates (see Figure 6). Conversely, it is possible to do the opposite by computing the azimuth and elevation angles from pixel coordinates. Finally, Eqns. (9)-(11) are associated with Fig. 6.

$$\frac{W/2}{r_x} = \tan\frac{FOV_x}{2}\ ,\ r_x = \frac{W/2}{\tan\frac{FOV_x}{2}}(\text{constant}) \tag{9}$$

$$dx = r_x \tan\alpha_{xz} \tag{10}$$

$$u = \frac{W}{2} + dx \tag{11}$$

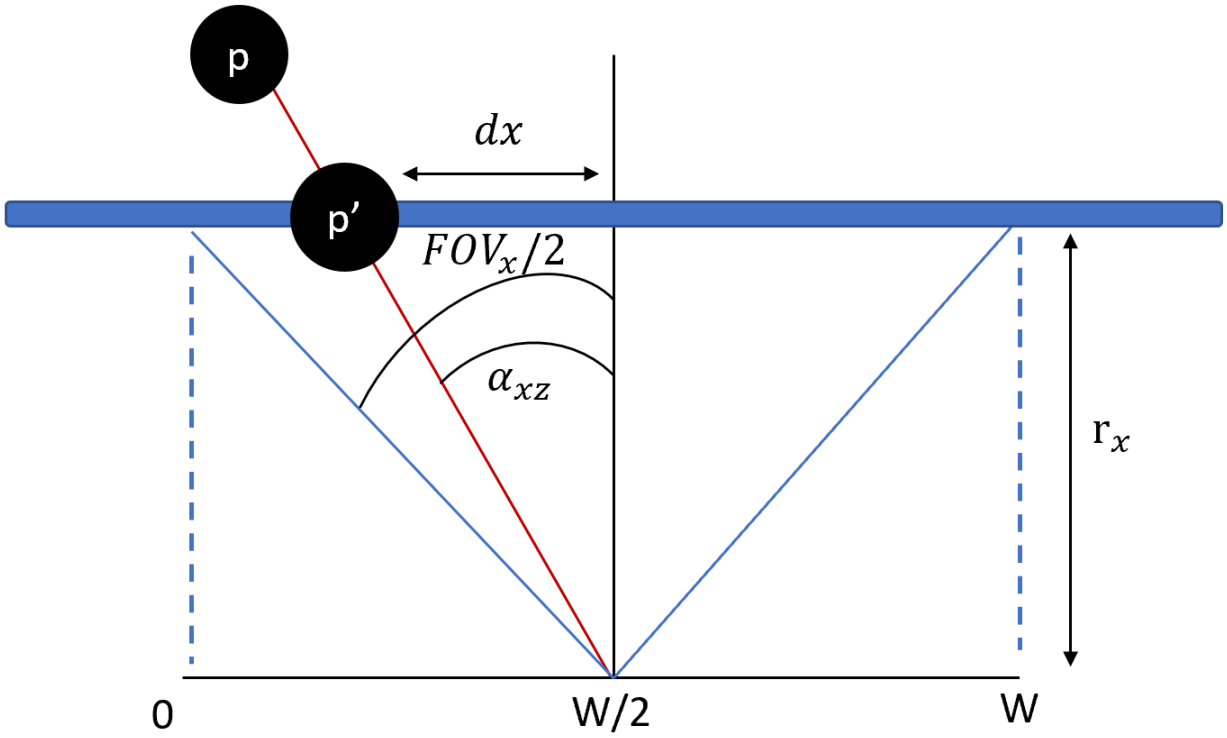


Fig. 6 Azimuth and Elevation Angles to Pixel Coordinates: 1) get the image plane distance, r_x (Eqn. (9)), 2) compute the horizontal pixel offset, dx (Eqn. (10)), 3) compute the horizontal pixels, u (Eqn. (11))

IV. Simulation Setup

A. Camera Model and Parameters

The camera model in this paper utilizes a perspective matrix camera model * to transform camera coordinates to normalized camera coordinates and then to pixel coordinates. Some of the parameters in "Landing Light Configuration" of Ref. [18] are also used in this paper, and Table 1 summarizes the simulated camera parameters utilized for this paper with FOV_h and FOV_v defined as the horizontal and vertical field of view angles, respectively.

Table 1 Camera Model Parameters

Parameter	Image Size	Pitch Angle (°)	Near Plane (m)	Far Plane (m)	FOV_h (°)	FOV_v (°)
Value	1920×1080	-9	1	4000	60	33.75

B. Reflection and X-Plane Interface

The NASA ARC Reflection software contains simulated AAM aircraft models and vertiport approach and landing trajectories. X-Plane is a flight simulator with 3D rendering capabilities and acts as the visualization tool for the simulated AAM aircraft and trajectory generated in Reflection. The "X-Plane Interface (XPI)" block in Fig. 1 establishes a UDP connection with X-Plane for real-time implementation, and more details of the interface between Reflection and X-Plane are in Ref. [32].

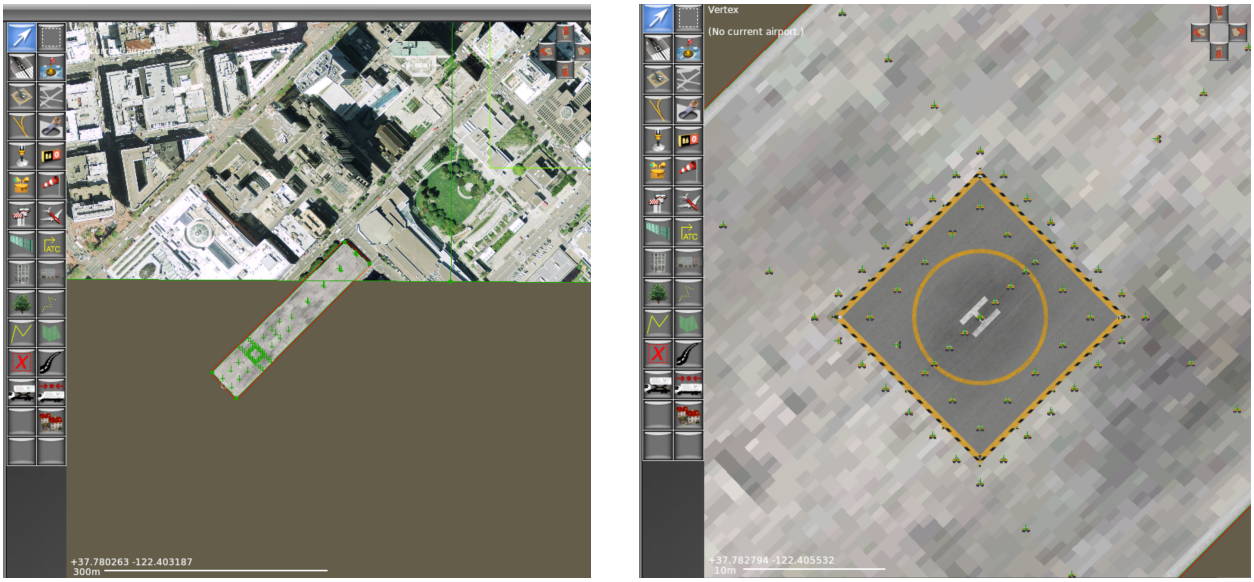
C. X-Plane and World Editor

Figure 7 shows a screenshot of the simulated FMG vertiport, created in X-Plane's 2D airport editor called World Editor for editing and creating customized airports and scenery. The landing lights in World Editor follow the guidelines presented in the FAA Advisory Circular on Heliport Design [1]. Since the FMG vertiport has a limited surface area, it lacks the bottom rows with four and five landing lights of the heliport approach landing system lights. Future iterations

*https://learn.microsoft.com/en-us/windows/win32/opengl/glfrustum

of the simulation can utilize the vertiport landing light configuration tentatively proposed in the FAA Vertiport Design Engineering Brief No. 105 in Ref. [2] or the approved vertiport design advisory circular when it becomes publicly available.

Figure 8 shows an X-Plane simulation of a static ground-based camera that tracks an AAM aircraft. Combining multiple camera fields of view provides visual coverage and overlap for incoming AAM aircraft during approach and landing. More details will be revealed later about distributed cameras for tracking AAM aircraft in subsection VII.C.



(a) FMG building and landing lights with orthophoto (top) **(b) FMG helipad: zoomed in version of a)**

Fig. 7 X-Plane & World Editor: FMG Vertiport

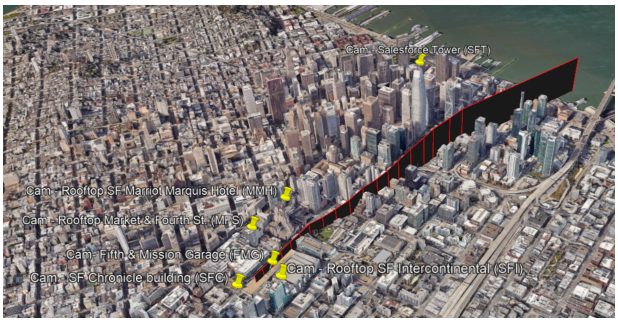
(a) X-Plane Static Camera Test **(b) X-Plane Static Camera Tracking Test**

Fig. 8 X-Plane Static Camera Test: (b) occurs at a different time than (a); (b) shows a red circle and green x to indicate the location of the tracked aircraft and has a black vertical line to indicate how close the tracked aircraft is relative to the vertical center of the image

D. Trajectory and Distributed Cameras

Figure 9 shows the nominal trajectory and distributed cameras in Google Earth. Waypoints were carefully placed throughout the path to avoid buildings in San Francisco, and the X-Plane World Editor rendered the FMG building (see Fig. 7). Table 2 shows the FMG curved trajectory's distributed camera locations and viewing angles.

(a) X-Plane & World Editor: Simulated FMG Trajectory



(b) FMG Curved Trajectory and Distributed Cameras

Fig. 9 Nominal Trajectory: X-Plane Rendering and Google Earth

Table 2 Distributed Camera Locations and Angles

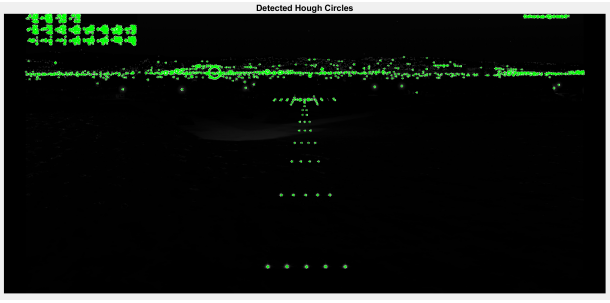
Camera Name	Latitude (°)	Longitude (°)	Abs. Alt. (m)	Heading (°)	Elev. (°)
Fifth & Mission Garage (FMG)	37.783379	-122.405077	26.0	45.0	16.9
San Francisco Chronicle (SFC)	37.782546	-122.406520	34.0	45.0	16.9
San Francisco Intercontinental (SFI)	37.781944	-122.404722	110.0	345	-15.0
Marriot Marquis Hotel (MMH)	37.785278	-122.404167	135	165	-45.0
Market & Fourth Street (MFS)	37.784684	-122.405732	76.0	165	-10.0
Salesforce Tower (SFT)	37.789782	-122.396968	326	225	-10.0

V. Feature Detection and Correspondence

A. Feature Detection Methods

Figure 10 shows preliminary results with Hough circle detection, which has promising results for detecting the circular landing lights. However, it also detects the background lights on the horizon, which is not ideal, as shown in Fig. 10b. Other feature detection methods such as SURF, Harris corner detection, FAST, template matching, and image registration did not detect the landing lights as well as Hough circle detection.

(a) X-Plane: Middle Harbor Shoreline Park



(b) Hough Circle Detection

Fig. 10 Original and Processed Images for Hough Circle Detection

B. Feature Correspondence

Combining Hough circle detection with a custom feature descriptor that finds the closest Hough circle detections to the estimated landing light locations gives better results because it ignores the farther and undesired background lights in the horizon (see Figure 11).

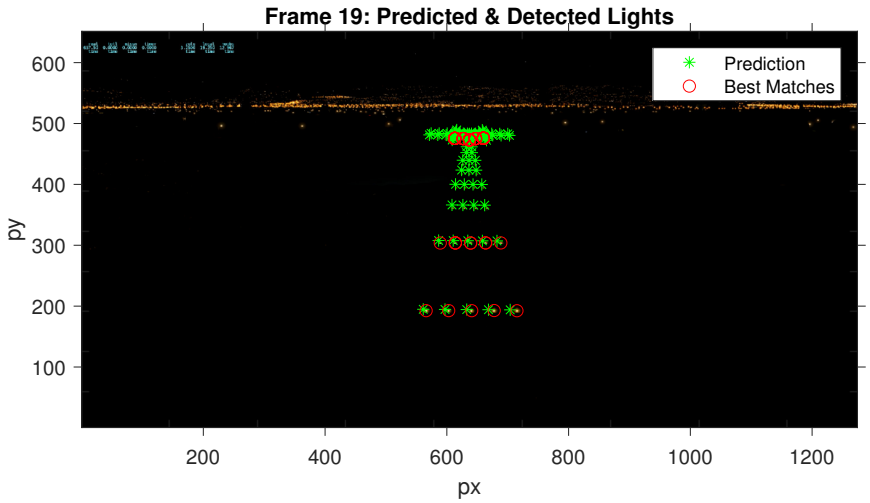


Fig. 11 X-Plane: Middle Harbor Shoreline Park Custom Feature Descriptor for Selecting the Best Detections

After matching the predicted and detected Hough circles in pixel coordinates, the next step is to determine the corresponding world coordinates for those detections based on matching indices:

(1) Compute and store the absolute difference between the predicted and detected landmark.
(2) Compute and store the Euclidean norm between each of the landmarks.
(3) Find the closest match by finding the minimum value of the stored Euclidean norm between each landmark.
(4) Determine the index of the closest match: best match
(5) Store the pixels of the best match.
(6) Apply further pruning based on a predetermined radial pixel. tolerance value, Δp_r.
    (I) Compute and store the distance for each predicted and detected landmark in pixel coordinates.
    (II) Find the shortest distance.
    (III) If the shortest distance is within the predetermined radial pixel tolerance, then store the index.
(7) Finally, match the index with the index for the landmark in world coordinates. Using bipartite graphs provides a visual representation of the pairing of the best matches with lines to show the connections between each pair.

The predictions and detections do not always perfectly align due to the small errors in the camera model. However, as long as they are within the specified pixel radial tolerance, the matches will be adequate. Therefore, the feature correspondence method is robust to false positives because it removes pixels outside the acceptable radial pixel range or keeps the detections close enough to the predictions. In other words, the detections that are close enough to the predicted landing lights may not necessarily be all detected landing lights. However, they will be considered since these detections are close to the predicted landing lights. Figures 15-17a show detections close to the predictions, which are false positives but close enough to be considered as landing lights.

VI. EKF Design

The EKF in this paper follows the EKF design in [18] with IMU measurements in the input vector, $\boldsymbol{u}$, and COPOSIT measurements in the measurement vector, $\boldsymbol{z}$. The EKF runs at 100 Hz and has two possible paths depending on what measurements are available, and Fig. 12 shows a high-level block diagram of the EKF structure for VAL. Ultimately, there are two paths: one with only IMU measurements (dead reckoning, only prediction) and another with IMU and COPOSIT measurements (prediction and correction steps). Its process noise covariance includes the IMU measurement variances:

$$\mathbf{Q} = diag\left[\begin{array}{cccccc}\sigma^2_{acc,x} & \sigma^2_{acc,y} & \sigma^2_{acc,z} & \sigma^2_{gyro,x} & \sigma^2_{gyro,y} & \sigma^2_{gyro,z}\end{array}\right]. \tag{12}$$

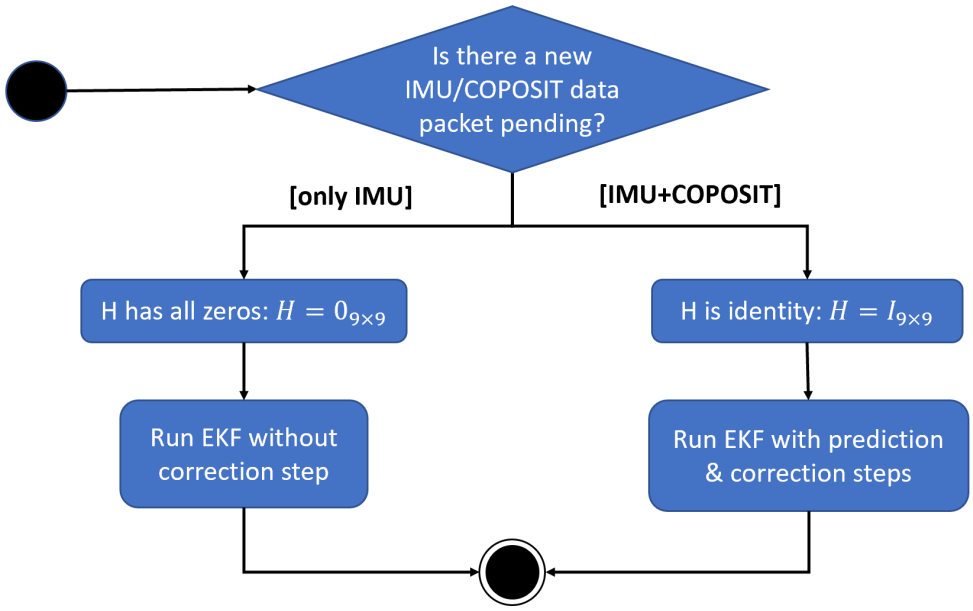


Fig. 12 EKF Flowchart: 1) running the EKF without a correction step and only the prediction step (dead reckoning) occurs since the Kalman gain, K_k, and measurement innovation, v_k, equal zero): $\hat{\boldsymbol{x}}_k(+) = \hat{\boldsymbol{x}}_k(-)$; 2) running the EKF with both prediction and correction steps has nonzero values for the Kalman gain, K_k, and measurement innovation, v_k: $\hat{\boldsymbol{x}}_k(+) = \hat{\boldsymbol{x}}_k(-) + K_k v_k$

The measurement noise covariance utilizes the variances from the coplanar POSIT algorithm and differs from **R** in Ref. [18] by including $\sigma^2_{vE}, \sigma^2_{vN}, \sigma^2_{vU}$:

$$\mathbf{R} = diag\left[\begin{array}{ccccccccc}\sigma_E^2 & \sigma_N^2 & \sigma_U^2 & \sigma_{vE}^2 & \sigma_{vN}^2 & \sigma_{vU}^2 & \sigma_\phi^2 & \sigma_\theta^2 & \sigma_\psi^2\end{array}\right]. \tag{13}$$

The position and velocity measurements are not independent such that the COPOSIT velocity measurements depend on the COPOSIT position measurements via finite difference:

$$\boldsymbol{v}_{coposit,i} = \frac{\boldsymbol{p}_{coposit,i} - \boldsymbol{p}_{coposit,i-1}}{t_i - t_{i-1}}. \tag{14}$$

Preliminary tests and analysis showed divergence and highly inaccurate state estimation using the approach in Ref. [18] without including COPOSIT velocity measurements.

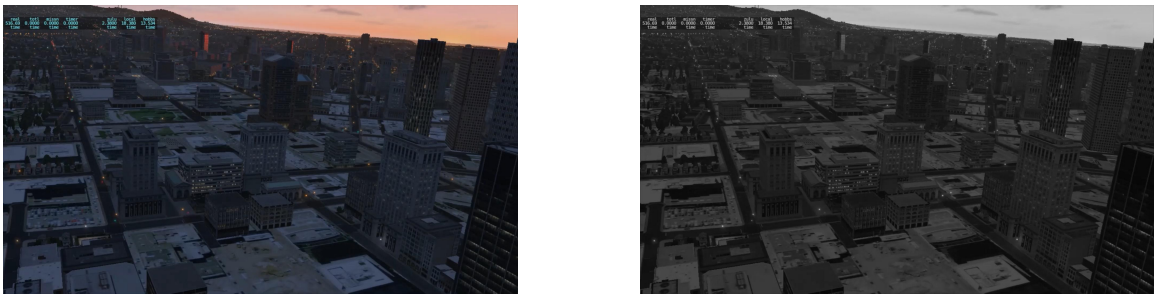
An alternative EKF design applicable for AAM approach and landing utilizes overcorrection during measurement updates. This EKF design has a prediction step independent of the available sensors, but the correction step occurs whenever a sensor measurement becomes available. In the context of this paper, the IMU measurements operate at 100 Hz, so the correction step runs on the noisy IMU measurements with their corresponding measurement matrix, which inevitably causes drift (dead reckoning). However, when COPOSIT measurements become available around 10-20 Hz, the EKF overcorrects the IMU measurement by applying COPOSIT measurements with its corresponding measurement matrix, which eliminates the drift. Overall, this adaptive EKF would work well for APNT AAM navigation whenever onboard sensor measurements become available [33].

VII. Simulation Results

This section shows the simulation results for the curved approach and landing trajectory at FMG (recall figure 9b).

A. VAL Image Processing

Figures 13 - 17 show preliminary results of the VAL image processing pipeline. Figure 13 shows the original and grayscale versions of the first frame in the simulated FMG approach and landing trajectory. The X-Plane overlay is in the top left corner. The most important information is the current UDP time, which needs to sync with the Reflection data to align the frames to obtain COPOSIT measurements that match the simulated IMU measurements.



(a) Original X-Plane Image: 1st Frame **(b) Grayscale Image: 1st Frame**

Fig. 13 X-Plane 1st Frame: Original & Grayscale

For automated time synchronization between the simulated X-Plane images and the telemetry data, the OCR trainer function in MATLAB can classify the timestamps on the static X-Plane overlay in the top left of each frame (see Fig. 14). Alternatively, manual time synchronization or an automated linear fit between the X-Plane and Reflection telemetry timestamps syncs the two datasets to match the rendered images with the simulated onboard telemetry data by matching their GPS coordinates.

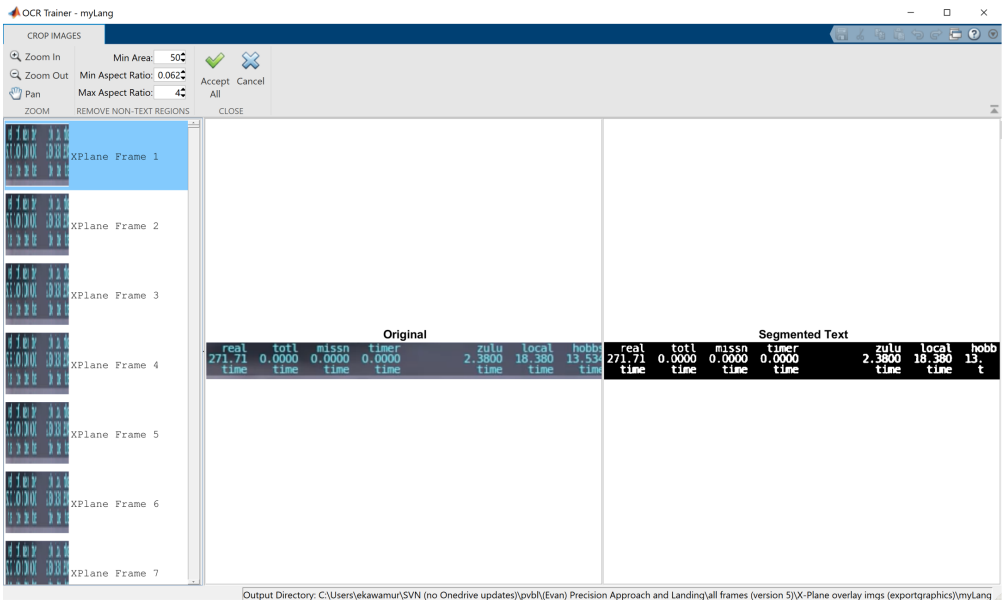
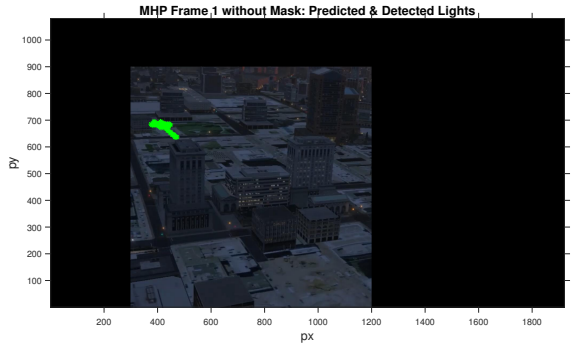
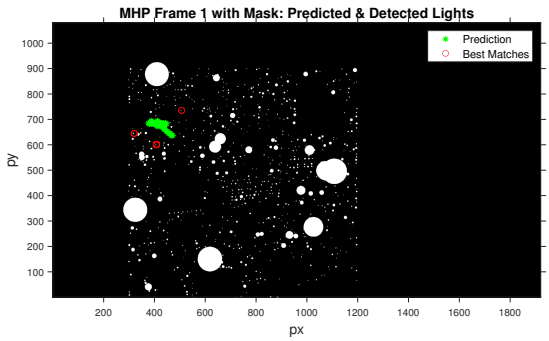


Fig. 14 OCR Trainer: Read and Classify All of X-Plane's Overlay

Figure 15a shows the predicted landing lights with darkened areas in the image to weed out pixels that do not contain landing lights. Figure 15b shows the results from applying Hough circle detection and the closest circles to the predicted landing lights, while Fig. 16a shows a cleaner version of the closest detected Hough circles to the landing lights without the other detected circles.



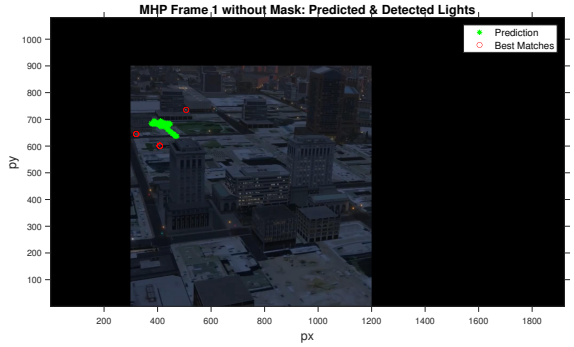
(a) Blackened Image with Predicted Landmarks



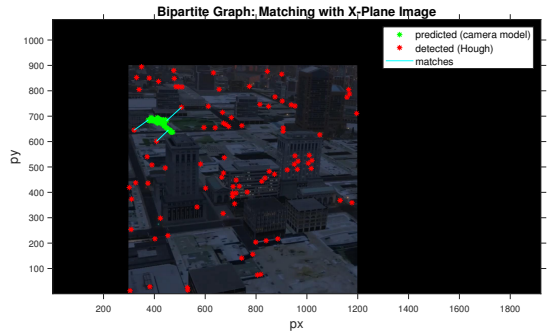
(b) Blackening and Mask with Detected Hough Circles

Fig. 15 Predicted and Detected Landing Lights

Fig. 16b shows the connections between the predicted and closest detected landing lights. Further pruning of the closest matches occurs based on a predetermined radial pixel tolerance value, Δp_r, set to 100 (recall the feature correspondence procedure in subsection V.B).



(a) Predictions and Detections with Blackening: 1st Frame

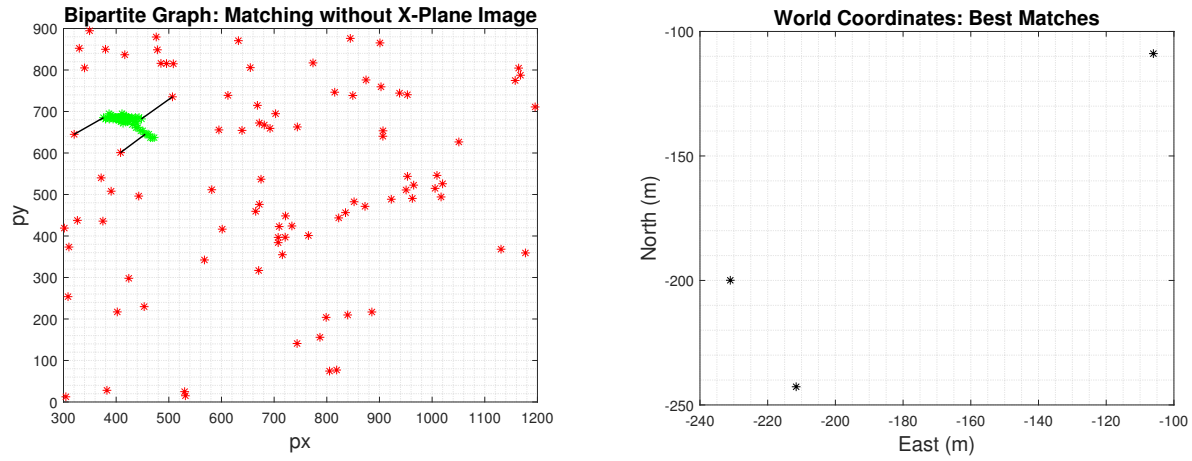


(b) Bipartite Graph: Closest Predictions and Detections

Fig. 16 Bipartite Graph Matching Between Predicted and Detected Landing Lights

Fig. 17 shows the best matches in pixel coordinates mapped to world coordinates based on matching indices between the predicted and detected landmarks. There are only three detections in this frame, so the EKF would apply only a prediction step without COPOSIT measurements for the correction step.

Figure 18 shows the number of detections throughout the simulated FMG approach and landing. Since there are at least four detections (coplanar points) for most of the trajectory, many COPOSIT measurements will provide correction steps in the EKF. More COPOSIT measurements ultimately lead to accurate state estimation due to more correction steps.



(a) Bipartite Graph: Closest Predictions and Detections

(b) Best Matches in World Coordinates

Fig. 17 Best Matches: Bipartite Graph (pixel coordinates) & World Coordinates: 1st Frame

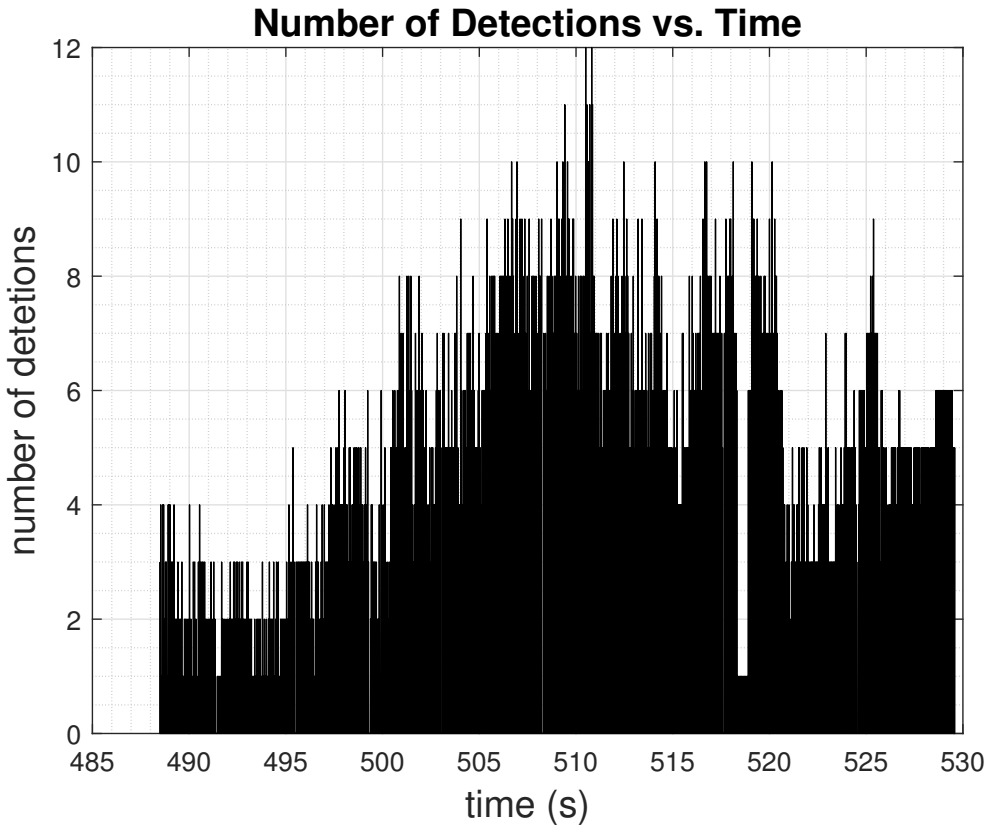


Fig. 18 Number of Detections vs. Time: COPOSIT requires at least four coplanar points (detections), so the times with less than four detections cause the state estimation to drift. However, there are short gaps between COPOSIT measurements, which cause brief moments of drift.

B. VAL EKF

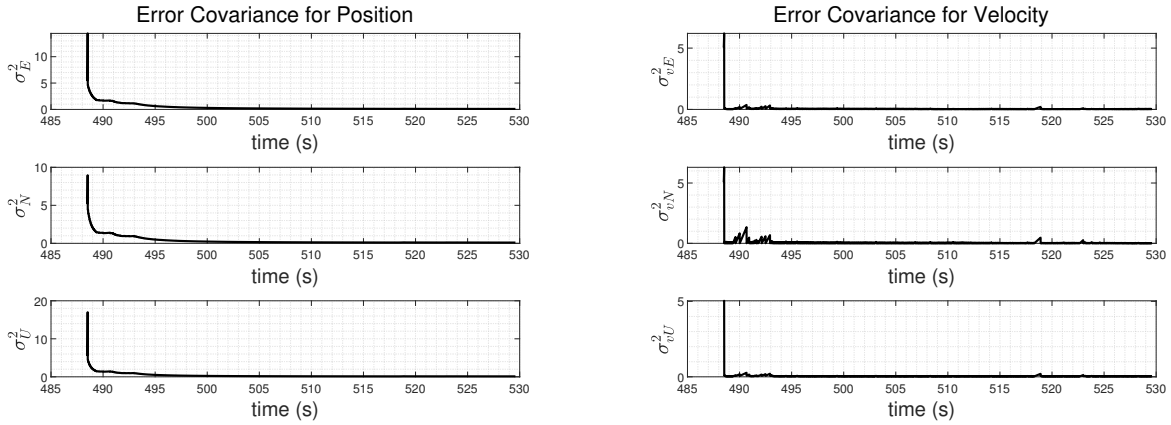
The process and measurement noise covariances ($\mathbf{Q}$, $\mathbf{R}$) values are:

$$\begin{aligned}\mathbf{Q} &= diag\begin{bmatrix} 11.56 & 16 & 25 & 0.00164 & 0.04 & 0.25 \end{bmatrix}, \\ \mathbf{R} &= diag\begin{bmatrix} 49 & 36 & 36 & 0.01 & 0.0999 & 0.0999 & 0.0001 & 0.0001 & 0.0001 \end{bmatrix}.\end{aligned} \tag{15}$$

The average EKF runtime is 2.3 milliseconds, while the feature detection and correspondence steps take about 5.5 seconds. For real-time implementation, VAL requires faster feature detection and correspondence steps to run in real time, especially the Hough circle detection step, which takes approximately 5.1 seconds.

1. Error Covariance

Figures 19-20 show that the error covariance quickly converges due to COPOSIT measurements with some minor spikes during brief moments without COPOSIT measurements, but the error covariances converge. Overall, the convergence of the error covariances demonstrates high confidence and low uncertainty in the state estimation.



(a) Error Covariance: Position **(b) Error Covariance: Velocity**

Fig. 19 Error Covariance: Position & Velocity

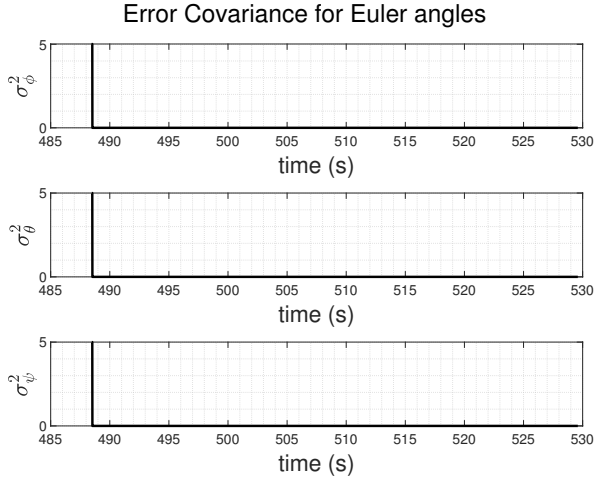
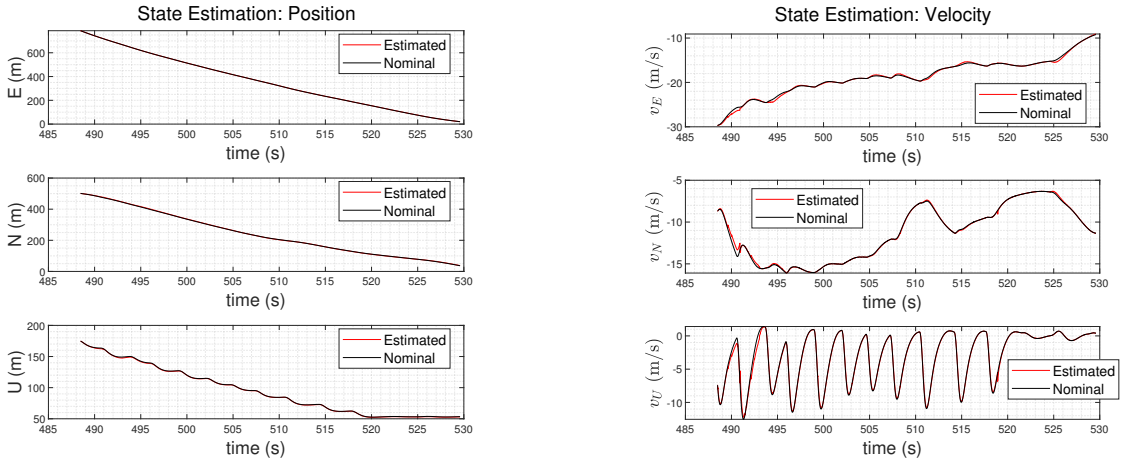


Fig. 20 Error Covariance: Euler Angles

2. State Estimation

Figure 21-22 show the state estimates, which closely align with the nominal states. Minor fluctuations occur due to the lack of COPOSIT measurements, but the estimates quickly return to the nominal states when COPOSIT measurements become available.



(a) State Estimation: Position **(b) State Estimation: Velocity**

Fig. 21 State Estimation: Position & Velocity

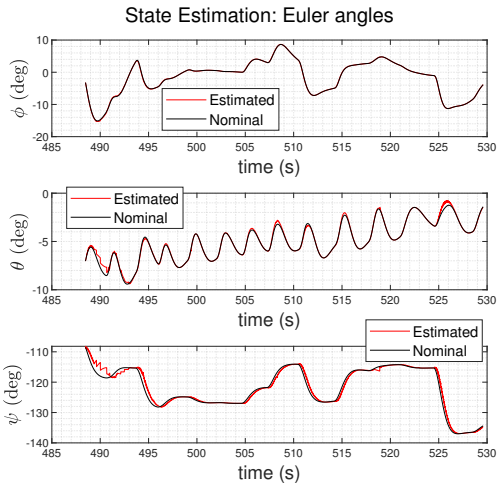


Fig. 22 State Estimation: Euler Angles

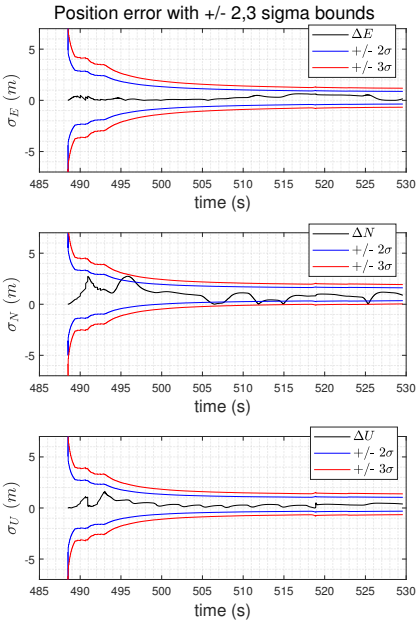
Table 3 shows the error statistics for the state variables. Overall, frequent detections of at least four coplanar distributed landmarks lead to higher accuracy for state estimation. The position estimation has submeter accuracy, so it beats GPS accuracy, which ranges within a few meters.

Table 3 State Estimation Statistics of the Error

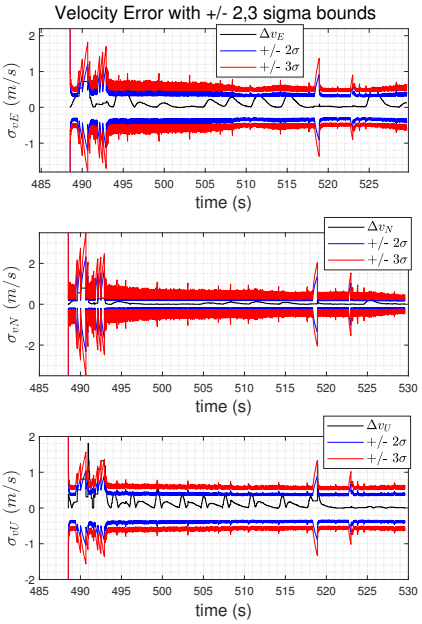
	E (m)	N (m)	U (m)	v_E (m/s)	v_N (m/s)	v_U (m/s)	ϕ (rad)	θ (rad)	ψ (rad)
μ	0.256	0.980	0.365	0.135	0.067	0.162	0.000856	0.00156	0.0116
σ	0.203	0.576	0.253	0.169	0.135	0.241	0.00115	0.00265	0.0151

3. Bounded Error

Figures 23-24 show the position, velocity, and Euler angle errors with $\pm 2, 3\sigma$ (blue, red) centered around the mean error. There are fluctuations throughout the trajectory, but the error remains bounded most of the time, demonstrating confidence in the state estimation results.



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



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

Fig. 23 Error with $\pm 2, 3\sigma$ Bounds: Position & Velocity

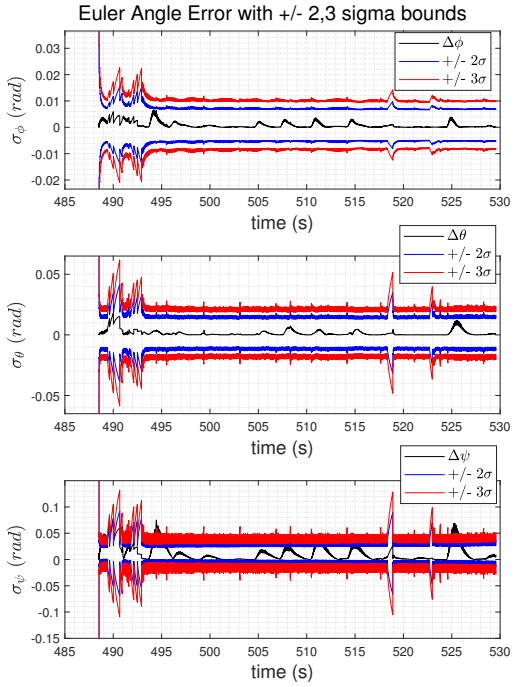
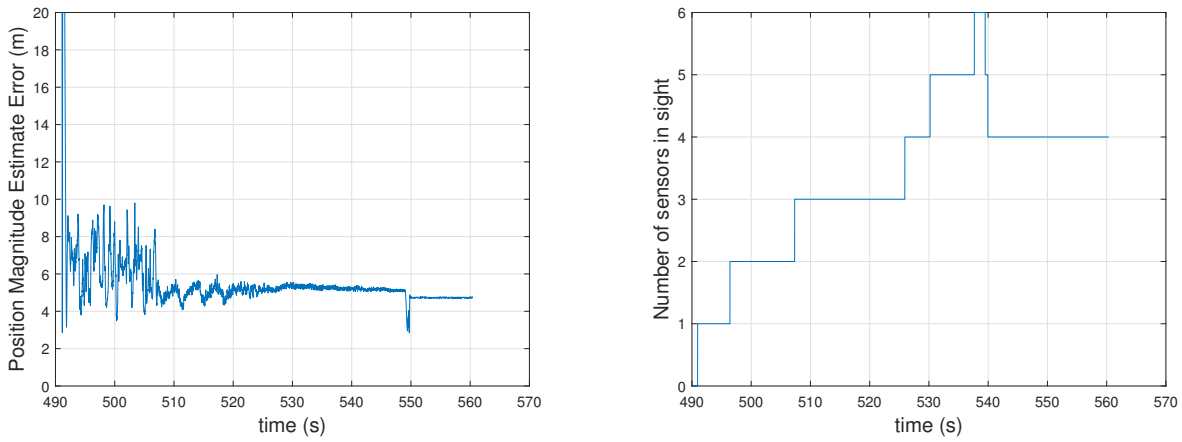


Fig. 24 Error with $\pm 2, 3\sigma$ Bounds: Euler Angles

C. Distributed Cameras for Tracking

A ground-based distributed sensing framework yields a passive camera tracking navigation solution for incoming AAM aircraft. Communication between the AAM aircraft and the ground-based distributed sensor network can increase the certainty of the AAM aircraft's state estimation. The ground-based distributed sensor network also acts as a failsafe or backup navigation solution if the AAM aircraft experiences sensor impairments or malfunctions. Coincidentally, at least four coplanar points and four ground-based sensors are good minimums to produce accurate estimates. Future work may include distributed ground-based sensor degradation and the impacts on the tracking estimation. For more details on the distributed camera EKF design, see Ref. [34].

Figure 25 shows a fixed-angled camera tracking test of rendered X-Plane images as a simulated aircraft flies within the cameras' views. When there are fewer than four independent cameras, the estimated error increases and gets noisier when there are fewer than three independent cameras.



(a) Ground-Based Distributed Camera Fit Error

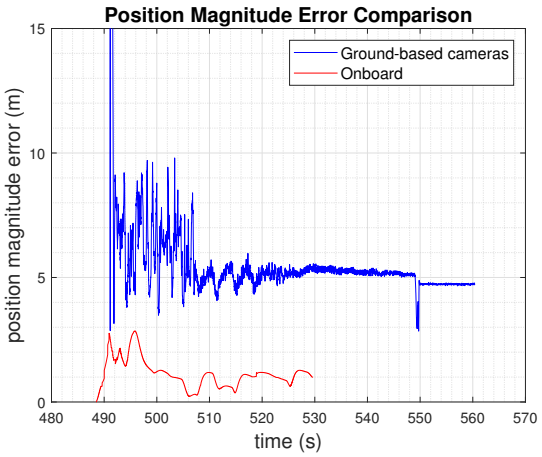
(b) Number of Cameras with Target in Sight

Fig. 25 X-Plane Static Camera Test: (b) occurs at a different time than (a); (b) shows a red circle and green x to indicate the location of the tracked aircraft and has a black vertical line to indicate how close the tracked aircraft is relative to the vertical center of the image

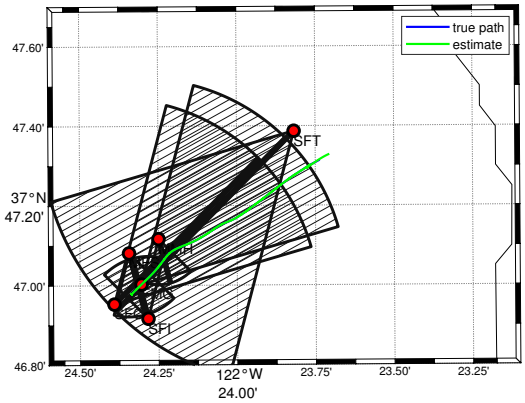
Ultimately, having a ground-based distributed sensor network can provide AAM pilots with a backup navigation solution as an automitigation contingency plan during landing [35]. Normal operations can provide periodic checks to ensure the onboard and ground-based position estimations agree.

Figure 26a compares the position magnitude error between the ground-based distributed cameras and the onboard navigation solutions. The red line in Fig. 26a is the Euclidean norm of the black lines (ENU error) from Fig. 23a, while the blue line is the Euclidean norm of the estimated position from the ground-based distributed cameras. The onboard solution (red line) stops about 20 seconds earlier than the ground-based distributed camera navigation solution because it does not include the landing phase. The AAM aircraft's camera would lose the landing lights in its field of view as it flies over the landing lights, which would cause the EKF to diverge due to a lack of COPOSIT measurements. However, the distributed cameras would maintain visual coverage of the aircraft during these final moments (see Fig. 26b for the distributed camera fields of views at the simulated FMG vertiport). Future work can investigate the divergence associated with a lack of COPOSIT measurements towards the end of the approach and landing trajectory.

Overall, the onboard AAM aircraft estimation is more accurate because it has immediate information from the IMU and COPOSIT measurements. The ground-based distributed cameras lack in situ information, so they are inherently more inaccurate. Even though the backup plan with the ground-based distributed camera framework may not have the best accuracy for estimating the AAM aircraft's position, it is adequate and within a reasonable range for AAM position estimation.



(a) Position Magnitude Error Comparison



(b) Map of the Distributed Cameras and Fields of View

Fig. 26 Position Magnitude Error Comparison and Distributed Camera Fields of View: a) the onboard AAM aircraft navigation solution ends earlier than the ground-based distributed camera navigation solution because it would lose landmarks in the onboard camera's field of view; (b) the black cones show the visual coverage of the cameras (red dots) along the trajectory

VIII. Conclusion

The simulated VAL system for PAL demonstrates promising results for AAM approach and landing scenarios. Accurate state estimation, i.e., beats GPS with submeter position accuracy, through frequent COPOSIT measurements provides several correction steps in the EKF to eliminate drift and divergence. Hough circle detection reliably yields at least four detections of the distributed landing lights at the vertiport for most of the approach and landing. The feature correspondence method demonstrates robustness to false positives, i.e., detections that are not landing lights if the detections are close enough to the predictions. Including finite differenced velocity COPOSIT measurements based on the COPOSIT position measurements provides accurate state estimation and convergence even though they are not independent. Consistently detecting at least four coplanar points (distributed landing lights) generates several COPOSIT measurements, which provide several correction steps for the EKF to yield accurate state estimation. VAL's runtime is around 5.5 seconds with the limiting factor being the feature detection and correspondence steps. Therefore, real-time implementation requires faster feature detection and correspondence steps, so future work can consider improving the runtime for the feature detection and correspondence steps.

Comparing the onboard and ground-based navigation solutions shows that the onboard position estimation is more accurate than the ground-based position estimation due to in situ measurements. The ground-based navigation solution is a viable backup plan if the AAM aircraft has any sensor degradation or malfunctions. Alternatively, the AAM aircraft could send its in situ measurements to the ground-based distributed sensor framework to increase the accuracy of the ground-based position estimation. Maintaining communication between the ground-based and airborne-based navigation solutions increases the confidence in the state estimation, which helps ensure a safe approach and landing.

This initial vision-based PAL study has the potential to impact many areas. First, it delivers perception PALS requirements and datasets to other NASA projects and industry partners for verification, validation, and characterization, which also identifies gaps in current AAM perception technology. Second, AAM aircraft require safe and accurate PALS to transport people and cargo successfully. Third, removing pilots with autonomous PALS increases efficiency and payload capacity. Finally, this initial vision-based PAL study paves the way for new AAM PALS research activities to enhance future AAM operations and research.

Acknowledgements

The authors would like to thank the Transformative Tools and Technologies (TTT) Project under the NASA Aeronautics Research Mission Directorate (ARMD) for funding this research.

References

[1] 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.

[2] Bassey, R., "Engineering Brief No. 105, Vertiport Design," U.S. Department of Transportation, 2022. URL https://www.faa.gov/sites/faa.gov/files/2022-09/eb-105-vertiports.pdf.

[3] Clearman, B., *Transportation-Markings Database: Aeronautical Navigation Aids*, Mount Angel Abbey, 2001.

[4] "Satellite Navigation - GBAS - How It Works," U.S. Department of Transportation, ???? URL https://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/navservices/gnss/laas/howitworks.

[5] Oliveira, P., Silva, J., and Soares, P., "A Comparative Study Between ILS and GBAS Approaches: The Case of Viseu Airfield," *Journal of Airline and Airport Management*, Vol. 10, No. 2, 2020, pp. 65–75. https://doi.org/http://dx.doi.org/10.3926/jairm.168.

[6] Enge, P., "Local Area Augmentation of GPS for the Precision Approach of Aircraft," *Proceedings of the IEEE*, Vol. 87, No. 1, 1999, pp. 111–132. https://doi.org/10.1109/5.736345.

[7] Parkinson, B., O'Connor, M., and Fitzgibbon, K., "Aircraft Automatic Approach and Landing Using GPS," *Global Positioning System: Theory & Applications*, 1996, pp. 397–425. https://doi.org/10.2514/5.9781600866395.0397.0425.

[8] Ciezkowski, M., "Triangulation Positioning System Based on a Static IR Beacon-Receiver System," *2017 22nd International Conference on Methods and Models in Automation and Robotics (MMAR)*, IEEE, 2017, pp. 84–88. https://doi.org/10.1109/MMAR.2017.8046803.

[9] Mondal, M., Shidlovskiy, S., Shashev, D., and Okunsky, M., "Autonomous Infrared Guided Landing System for Unmanned Aerial Vehicles," *International Conference on Distributed Computer and Communication Networks*, Springer, 2021, pp. 387–395. https://doi.org/10.1007/978-3-030-97110-6_30.

[10] Peisen, D., and Sawyer, B., "Heliport/Vertiport MLS Precision Approaches," U.S. Department of Transportation, 1994. URL http://www.tc.faa.gov/its/worldpac/techrpt/rd94-23.pdf.

[11] Leman, P., Gill, K., and Merkle, M., "Concept of Operations for NextGen Alternative Position, Navigation, and Timing (APNT)," U.S. Department of Transportation, 2012. URL https://www.faa.gov/sites/faa.gov/files/about/office_org/headquarters_offices/ato/20120319_APNT_CONOPS_FINAL.pdf.

[12] Hartley, R., and Zisserman, A., *Multiple View Geometry in Computer Vision*, 2nd ed., Cambridge University Press, 2003.

[13] 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. https://doi.org/10.1109/ICPR.2006.579.

[14] 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. https://doi.org/10.1109/TPAMI.1987.4767965.

[15] DeMenthon, D. F., and Davis, L. S., "Model-Based Object Ppose in 25 Lines of Code," *International Journal of Computer Vision*, Vol. 15, No. 1-2, 1995, pp. 123–141. https://doi.org/10.1007/BF01450852.

[16] 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. https://doi.org/10.1006/cviu.1996.0037.

[17] 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. https://doi.org/10.1023/B:VISI.0000025800.10423.1f.

[18] Kawamura, E., Kannan, K., Lombaerts, T., and Ippolito, C. A., "Vision-Based Precision Approach and Landing for Advanced Air Mobility," *AIAA SciTech 2022 Forum*, AIAA 2022-0497, 2022. https://doi.org/10.2514/6.2022-0497.

[19] 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. https://doi.org/10.1364/AO.51.000936.

[20] 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. https://doi.org/10.1109/ICPR.1996.546075.

[21] 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. https://doi.org/10.11003/JPNT.2017.6.2.59.

[22] Courbon, J., Mezouar, Y., Guénard, N., and Martinet, P., "Vision-Based Navigation of Unmanned Aerial Vehicles," *Control Engineering Practice*, Vol. 18, No. 7, 2010, pp. 789–799. https://doi.org/10.1016/j.conengprac.2010.03.004.

[23] Weiss, S. M., "Vision Based Navigation for Micro Helicopters," Ph.D. thesis, ETH Zurich, 2012. URL https://www.research-collection.ethz.ch/bitstream/handle/20.500.11850/52698/eth-5889-02.pdf.

[24] Sinopoli, B., Micheli, M., Donato, G., and Koo, T.-J., "Vision Based Navigation for an Unmanned Aerial Vehicle," *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, Vol. 2, IEEE, 2001, pp. 1757–1764. https://doi.org/10.1109/ROBOT.2001.932864.

[25] Schmid, K., Tomic, T., Ruess, F., Hirschmüller, H., and Suppa, M., "Stereo Vision Based Indoor/Outdoor Navigation for Flying Robots," *2013 IEEE/RSJ international conference on intelligent robots and systems*, IEEE, 2013, pp. 3955–3962. https://doi.org/10.1109/IROS.2013.6696922.

[26] Veneruso, P., Opromolla, R., Fasano, G., Burgio, G., Gentile, G., and Tiana, C., "Extending Enhanced Visual Operations to Urban Air Mobility: Requirements and Approaches," *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*, IEEE, 2021, pp. 1–9. https://doi.org/10.1109/DASC52595.2021.9594381.

[27] Escudero, N., Costas, P., Hardt, M. W., and Inalhan, G., "Machine Learning Based Visual Navigation System Architecture for Aam Operations with A Discussion on its Certifiability," *2022 Integrated Communication, Navigation and Surveillance Conference (ICNS)*, IEEE, 2022, pp. 1–15. https://doi.org/10.1109/ICNS54818.2022.9771519.

[28] Lombaerts, T., Shish, K. H., Keller, G., Stepanyan, V., Cramer, N. B., and Ippolito, C. A., "Adaptive Multi-Sensor Fusion Based Object Tracking for Autonomous Urban Air Mobility Operations," *AIAA SciTech 2022 Forum*, 2022, p. 0362. https://doi.org/10.2514/6.2022-0362.

[29] Grewal, M. S., Weill, L. R., and Andrews, A. P., *Global Positioning Systems, Inertial Navigation, and Integration*, John Wiley & Sons, 2007.

[30] Schaub, H., and Junkins, J. L., *Analytical Mechanics of Space Systems*, 4$^{\text{th}}$ ed., AIAA, 2018.

[31] 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. https://doi.org/10.1007/978-3-642-11690-2_4.

[32] Kannan, K., Baculi, J., Lombaerts, T., Kawamura, E., Gorospe, G., Holforty, W., Ippolito, C., Stepanyan, V., Dolph, C., and Brown, N., "A Simulation Architecture for Air Traffic Over Urban Environments Supporting Autonomy Research in Advanced Air Mobility," *AIAA SciTech 2023 Forum*, AIAA-2023-0895, 2023. https://doi.org/10.2514/6.2023-0895.

[33] Stepanyan, V., Kannan, K., Kawamura, E., Lombaerts, T., and Ippolito, C., "Target Tracking with Distributed Sensing and Optimal Data Migration," *AIAA SciTech 2023 Forum*, AIAA-2023-2194, 2023. https://doi.org/10.2514/6.2023-2194.

[34] Lombaerts, T., Kannan, K., Dolph, C., Stepanyan, V., George, G., and Ippolito, C., "Distributed Ground Sensor Fusion Based Object Tracking for Autonomous Advanced Air Mobility Operations," *AIAA SciTech 2023 Forum*, AIAA-2023-0896, 2023. https://doi.org/10.2514/6.2023-0896.

[35] Kim, J., Sharma, P., Atkins, E., Neogi, N., Dill, E., and Young, S., "Assured Contingency Landing Management for Advanced Air Mobility," *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*, IEEE, 2021, pp. 1–12. https://doi.org/10.1109/DASC52595.2021.9594498.