

TDP: AeroITAM (México), Indoors Category

J. Aranda-Montoya*, A. Cazares-García*, D. Martínez-Tinoco*, N. Medina-García*, E. Medina-Mejía*, M. Méndez-Mejía*, E. Mora-García*, S. Orduña-Ferreira*, M. Trujillo-Flores* and M. Martinez-Ramirez*,†

*Instituto Tecnológico Autónomo de México,

Campus Río Hondo, Departamento de Ingeniería Eléctrica y Electrónica,
Río Hondo 1, Ciudad de México, 01080, México

†Centro de Investigación y de Estudios Avanzados
del Instituto Politécnico Nacional,

Sección de Mecatrónica, Depto. de Ingeniería Eléctrica,
Ciudad de México, 07360, México

ABSTRACT

This work presents the control strategies developed for the indoor category of the International Micro Air Vehicle competition. Two approaches are considered: an open-loop strategy for pre-defined maneuvers, and a vision-based closed-loop controller for autonomous navigation and task execution. The proposed control algorithms are implemented on a DJI Mini 4 Pro platform, with the main processing running on an offboard PC. The results demonstrate the feasibility of integrating vision-based feedback with a commercial drone to address complex competition scenarios.

1 INTRODUCTION

Mexico is located at the intersection of several tectonic plates, which makes it one of the most seismically active regions in the world. Since the devastating 1985 earthquake, the government and society have become more aware of the risks associated with natural disasters and the importance of preparedness. These concerns resurfaced strongly after the major earthquakes of 2017 and 2019, which once again highlighted the vulnerability of Mexico City and surrounding areas. In this context, new technologies, such as aerial vehicles, are increasingly being considered as valuable tools for improving disaster response and resilience.

In recent years, unmanned aerial vehicles (UAVs) have demonstrated their feasibility in dangerous and hard-to-access scenarios, and their potential applications have been widely studied. In [1], rotary-wing UAVs were employed to transport first-aid materials, while in [2] they were used for mapping affected zones and supporting search and rescue operations. Consequently, the use of UAVs in the context of seismic disasters represents a promising approach. Whether by employing vision-based algorithms to detect survivors, or by equipping platforms with microphones to identify distress

calls, UAVs can accelerate the location of victims and reduce the risk of fatalities caused by delays in search and rescue.

Given this context, UAV competitions have emerged as valuable platforms for incorporating natural disaster scenarios into their challenges. These events provide controlled yet demanding environments where teams can design, test, and compare innovative aerial solutions that may later be adapted to real-world disaster response. By reproducing representative tasks—such as navigating confined spaces, identifying victims, or performing precise maneuvers—competitions not only advance the state of aerial robotics but also foster collaboration among students and researchers, ensuring a strong link between academic innovation and practical applications.

For this reason, the IMAV 2025 Competition focuses on the theme of seismic natural disasters. The indoor competition is composed of four tasks. The first task requires the UAV to travel through a tunnel, simulating scenarios where aerial vehicles must navigate confined spaces to locate injured individuals. The second task consists of finding a whiteboard and drawing a line on it, which highlights the precision and stability that a quadrotor must demonstrate in operation. The third task involves detecting a cry for help, reflecting real situations after an earthquake in which rapid victim localization is essential for increasing survival chances. Finally, the fourth task challenges the UAV to land on a moving platform obscured by smoke, a condition that simulates the reduced visibility often present in post-disaster environments.

Together, these tasks aim to replicate realistic challenges encountered in post-seismic scenarios, encouraging the development of innovative aerial solutions that combine autonomy, perception, and resilience. By solving them, research teams not only compete but also contribute to advancing UAV applications in disaster relief operations.

This paper presents the team's technical approach to solving the indoor tasks of IMAV 2025. The proposed strategies rely primarily on vision-based control, complemented by lightweight open-loop controllers, all tested on a laboratory-built replica of the competition arena. The overall design philosophy emphasizes simple, real-time methods that can be implemented reliably under resource-constrained UAV platforms.

*Email address(es): {jaranda4, lcazares, dmarti79, nmedina3, emilio.medina, miguel.mendez.mejia, emoraga2, samuel.orduna, mtrujillo1, marco.martinez}@itam.mx

The remainder of the paper is organized as follows. In Section 2 a brief description of the team, as well as the team's patch, is presented. Section 3 presents the mathematical model of the experimental platform developed by the team. Section 4 describes the proposed approaches to address each task. Finally, Section 5 provides the conclusions and outlines future directions for this work.

2 TEAM DESCRIPTION

Team AeroITAM represents the Instituto Tecnológico Autónomo de México (ITAM) in the IMAV 2025 competition. The group is composed of undergraduate students from the Department of Electrical Engineering. The team's research focuses on the development of autonomous aerial systems with emphasis on perception and control for operation in challenging environments.

Figure 1 shows the official team patch, which represents our participation in the competition.



Figure 1: AeroITAM team's patch.

3 QUADROTOR'S MATHEMATICAL MODEL

Under the standard assumptions, the quadrotor's kinematic and dynamic model can be described by the following differential equations[3].

$$\begin{aligned} \dot{\chi} &= \mathcal{R}V^b, & m\dot{V}^b &= -m\Omega \times V^b + mg\mathcal{R}^\top e_3 - T_T e_3 \\ \dot{\mathcal{R}} &= \mathcal{R}S(\Omega), & J\dot{\Omega} &= -\Omega \times J\Omega + M^b \end{aligned} \quad (1)$$

with $\chi = [x \ y \ z]^\top$ the quadrotor's translational position,

$$\mathcal{R} = \begin{bmatrix} c_\psi c_\theta & c_\psi s_\theta s_\phi - s_\psi c_\phi & c_\psi c_\theta s_\phi + s_\psi s_\phi \\ s_\psi c_\theta & s_\psi s_\theta s_\phi + c_\psi c_\phi & s_\psi c_\theta s_\phi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix} \in SO(3)$$

with $SO(3) = \{\mathcal{R} \in \mathbb{R}^{3 \times 3} \mid \mathcal{R}^\top \mathcal{R} = I_3, \det(\mathcal{R}) = 1\}$ the rotation matrix parametrized using the Tait-Bryan angles, where ϕ is the roll angle (rotation around the longitudinal axis $0x^b$), θ the pitch angle (rotation around the lateral axis $0y^b$) and ψ the yaw angle (rotation around the vertical axis $0z^b$). $V^b = [u \ v \ w]^\top$ is the quadrotor's translational velocity expressed in body coordinates, m is the quadrotor's mass, g is the gravitational constant, T_T is the total thrust generated by the four rotors, and $e_3 = [0 \ 0 \ 1]^\top$. Moreover, $J = \text{diag}\{J_{xx}, J_{yy}, J_{zz}\}$ is the inertia matrix, $\Omega = [p \ q \ r]^\top$ is the quadrotor's angular velocity expressed in body coordinates and M^b are the control moments produced through the differential thrust generated by the rotors. Given two vectors $a, b \in \mathbb{R}^3$ the matrix $S(a)$ is defined in such a way that $a \times b = S(a)b$.

The quadrotor dynamic model (1) is reduced to a fully actuated kinematic model under the following assumption.

Assumption 1 Consider the quadrotor dynamics (1) in closed-loop with a quadrotor's fast dynamics controller that defines T_T and M_b such that

$$\lim_{t \rightarrow T} u = u_d, \quad \lim_{t \rightarrow T} v = v_d, \quad \lim_{t \rightarrow T} w = w_d, \quad \lim_{t \rightarrow T} r = r_d, \quad (2)$$

for some finite $T > 0$, with u_d, v_d, w_d the desired translational speeds and r_d the desired rotational speed.

Under Assumption 1, each quadrotor dynamics (1) is reduced to the following differential equations

$$\begin{aligned} \dot{\chi}_i &= \bar{R}_i U_i, \\ \dot{\bar{R}}_i &= \bar{R}_i S(r_{d_i}) \end{aligned} \quad (3)$$

with $U_i = [u_{d_i} \ v_{d_i} \ w_{d_i}]^\top$, $\bar{R}_i = \text{diag}\{R_i, 1\}$,

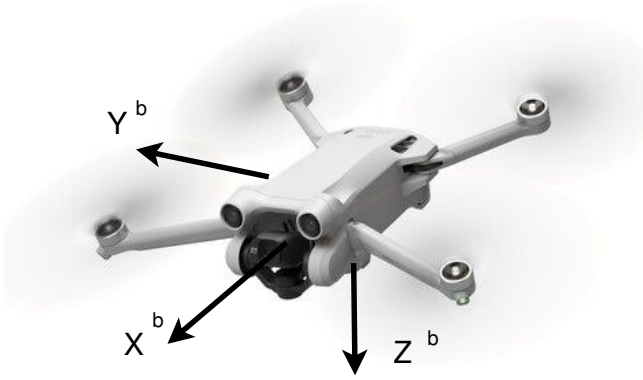
$$\begin{aligned} R_i &= \begin{bmatrix} c_{\psi_i} & -s_{\psi_i} \\ s_{\psi_i} & c_{\psi_i} \end{bmatrix} \in SO(2), \\ S(r_{d_i}) &= \begin{bmatrix} 0 & -r_{d_i} \\ r_{d_i} & 0 \end{bmatrix} \in \mathfrak{so}(2) \end{aligned} \quad (4)$$

where

$$\begin{aligned} SO(2) &= \{R_i \in \mathbb{R}^{2 \times 2} \mid R_i^\top R_i = I_2, \det(R_i) = 1\}, \\ \mathfrak{so}(2) &= \left\{ S(\cdot) \in \mathbb{R}^{2 \times 2} \mid S(c) = \begin{bmatrix} 0 & -c \\ c & 0 \end{bmatrix}, \forall c \in \mathbb{R} \right\}. \end{aligned}$$

This reduced model is the one the team will work on, with the platform DJI Mini 3 pro, see Figure 2.

The team is already familiar with this UAV platform, which was initially introduced during their participation in IMAV 2024.

Figure 2: Experimental platform *DJI Mini 3 pro*

4 CONTROL APPROACH

In this section is detailed the approach on each task to solve the problem given.

4.1 Task one - Tunnel

The first task consists of traveling through a tunnel that appears to be constructed from square-shaped frames of a specific color. To detect the correct tunnel, the team employed the OpenCV (CV2) library together with the HSV color space, see Figure 3.

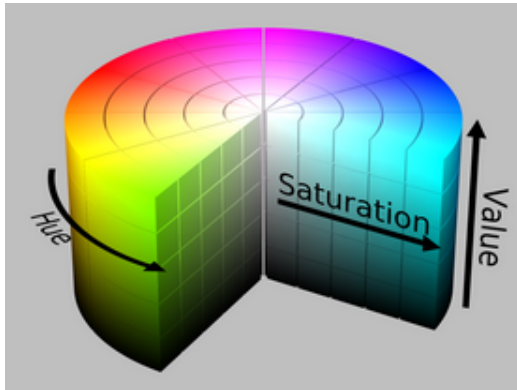


Figure 3: HSV mapping color.

Color detection in HSV is carried out by defining a mask that isolates the pixels within a given hue range corresponding to the target color. The saturation component is used to filter out darker or less vivid colors that may cause ambiguity, while the value component is adjusted to discard overly bright regions such as white tones. This process ensures that only the pixels corresponding to the desired color are retained, providing a reliable input for tunnel detection and navigation.

After applying the color mask, the result is a binary (black-and-white) image where the white pixels correspond

to the detected frame.

The proposed approach uses the control inputs v (lateral velocity) and w (vertical velocity) to align the UAV with the center of the frame. This is achieved by computing the centroid of the white pixels, which represents the average position of the detected frame in the image. The SciPy library provides a convenient “center of mass” function that calculates this centroid efficiently. By comparing the centroid position with the image center, corrective control actions can be applied to guide the UAV toward proper alignment with the tunnel, with the following controller

$$\begin{aligned} v &= -k_1(x_{ci} - x_{com}) \\ w &= -k_2(y_{ci} - y_{com}) \end{aligned} \quad (5)$$

where x_{ci}, y_{ci} is the position of the pixel located on the center of the image, and x_{com}, y_{com} is the position of the calculated center of mass.

However, this approach is effective only when the UAV is positioned directly in front of the frame. If the vehicle is offset or misaligned, an additional controller is required to guide it into the correct position. For this purpose, the remaining control inputs, u (longitudinal velocity) and r (yaw rate), are employed to adjust the UAV’s orientation and lateral displacement until it is properly aligned with the frame.

This alignment is achieved by applying a horizontal edge detector and measuring the distances between the detected borders of the frame in the filtered image. Specifically, the distances between the first two borders and the last two borders are computed, denoted as w_l and w_r , respectively. A yaw controller is then proposed to minimize the difference between these two values, according to

$$r = -k(w_l - w_r) \quad (6)$$

where r represents the yaw control input and k is a proportional gain. This control action drives the UAV to maintain symmetry in the image, effectively aligning it with the frame. In parallel, the lateral velocity input u is employed to regulate the UAV’s position with respect to a fixed distance from the frame. Once alignment and positioning are achieved, the quadrotor proceeds with the tunnel traversal task. Figure 4 presents an image of the algorithm’s test.

4.2 Task two - Draw on the whiteboard

To perform this task, the computer vision library `cv2` in Python was used. This library provides functions to identify ArUco markers and extract important features such as their corners, which we employed to calculate the drone’s viewing angle of the marker.

It is crucial to determine whether the drone is correctly positioned in front of the ArUco marker to ensure stability when drawing on the whiteboard. The corners obtained from the marker were expressed by the program in rectangular coordinates, which we used as a reference to calculate the norm of the marker’s sides.

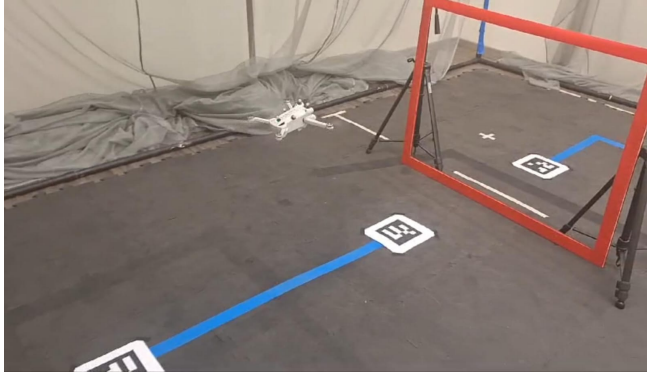


Figure 4: Tunnel task test.

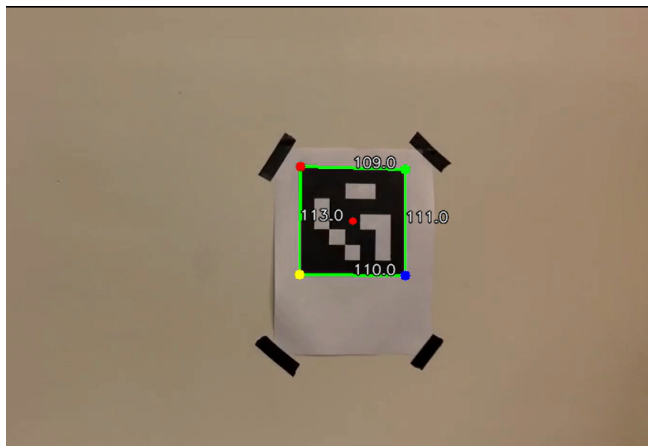


Figure 5: Real-time calculation of norms and positioning.

By using the calculated norms, we were able to determine the drone's relative position with respect to the marker and design a control system to ensure the correct alignment. The positioning equations were based on maintaining equality between the two lateral norms (ls and rs), while adjusting the drone's altitude by comparing the upper norm (us) with the desired upper side (ds). We did not use the lower side for comparison because, due to the drone's stability systems, it cannot maintain a tilted vision along the vertical axis. The implemented equations are as follows:

$$u = -\frac{1}{4}(ls - ld) \quad (7)$$

$$r = (rs - ls) \times 5 \quad (8)$$

Once the drone was properly positioned, we implemented basic translation controls based on desired norms, which we previously determined through experiments, to establish how much the drone should advance before initiating the planned drawing routine on the whiteboard, an image of the experiment is shown in Figure 6.

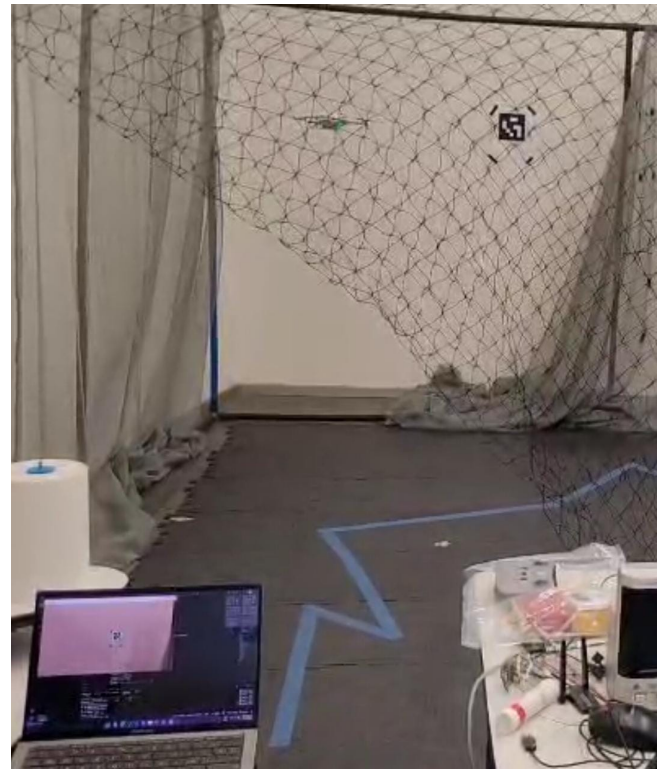


Figure 6: Experiment to test the vision base algorithm to center an ArUco.

To maintain optimal stability during the drawing process, a mechanical system was designed consisting of a support that mimics a marker, with a tip and an ink reservoir. This support includes a mechanism that absorbs the impact when

the tip touches the whiteboard, while still providing sufficient rigidity to allow the drone to produce stable and visible strokes.

The design of the device has been progressively improved through several iterations and tests carried out over the past weeks. Figure 7 illustrates one of the intermediate prototypes, while Figure 8 presents the most recent iteration.

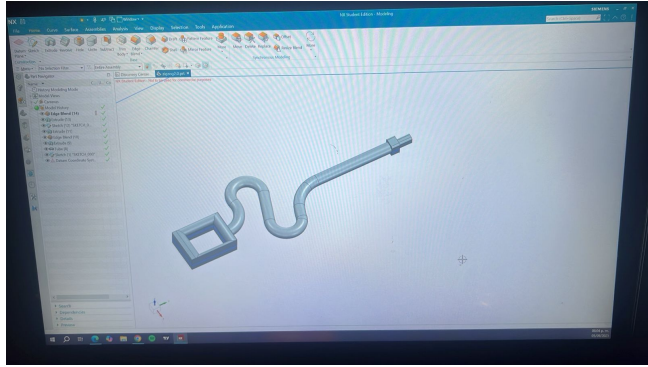


Figure 7: First approach of the marker holder.

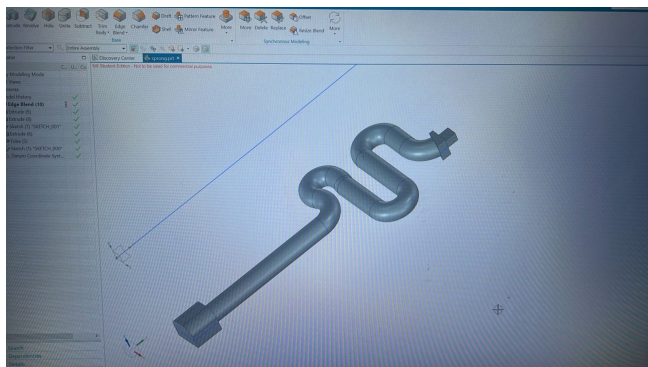


Figure 8: Marker mimic shape for impact absorbing.

Additionally, this system was designed to be extremely lightweight, since the use of commercial drones imposes restrictions on payload capacity. All components were 3D-printed using TPU plastic filaments to ensure stability, with a hollow structure to meet the weight requirements observed during testing.

4.3 Task three - Find the cry for help sound and deliver a med-kit

Since the only available information about the task was the noise level in decibels, a sound level meter was developed to measure it. The sound level meter was built using KY-038 sound sensors, Arduino, and other electronic components.

This device allows us to determine when the sound pressure exceeds a threshold value, which we set to 70 dB in this case. However, it cannot identify the position of the noise

source, as it captures the overall sound pressure of the surrounding environment. To address this limitation, we modified the KY-038 sensor's microphone so that it would change from an omnidirectional type to a shotgun type.

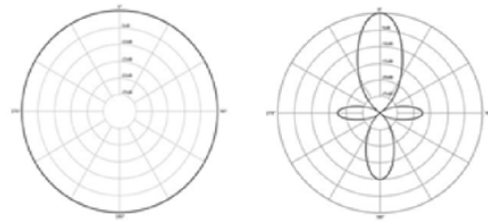


Figure 9: Omnidirectional and shotgun polar curve.

The microphones in the sensors are usually electret microphones because they are inexpensive. The only drawback is that they capture sound omnidirectionally. To modify their polar pattern to behave like a shotgun microphone, we implemented a structure that isolates sound, allowing the microphone to capture vibrations only from the front.

We modified the microphone to build an environmental scanning system similar to acoustic localization, but in this case it was passive rather than active. To achieve this, the microphone with its isolating structure was mounted on a servomotor, enabling sound mapping of the environment with the drone's front aligned to the north. This way, if a sound exceeding the threshold is detected anywhere in the room, we can calculate the angular difference with respect to the drone's front and simply rotate it.



Figure 10: Abstraction of an acoustic radar.

Once the sound is localized, the drone will head toward its direction. To obtain an approximate position of the noise source, a camera-based identification system was used to recognize the target—in this case, the speaker—and drop the aid

package.

The aid package was built following the competition's size requirements. To release it, we implemented a mechanism attached to the bottom of the drone. This mechanism holds the package with a rubber band, which is kept under tension by a cord tied to a servomotor. The design is simple: once the drone identifies the target, a command is sent to the Arduino (previously integrated into the system), which activates the servomotor. The servo then pulls the cord, releasing the rubber band's tension and allowing the package to drop onto the target.

4.4 Task four - Landing on a moving platform

The objective is to perform a smooth autonomous landing on a platform that moves with variable speed and heading in the horizontal plane. The environment exhibits transient visual degradation (smoke/partial occlusions), so the solution must be robust to momentary measurement dropouts and non-Gaussian visual noise. Quantitative targets are set as follows:

- Touchdown lateral error ≤ 0.10 m and yaw error ≤ 5 deg.
- Vertical speed at touchdown ≤ 0.3 m/s.
- Tolerance to loss of visual measurements up to 0.5 s without loss of stability.

4.4.1 Functional architecture and data flows

The architecture follows a GNC (*Guidance–Navigation–Control*) scheme with safety logic (conceptual figure):

1. **Perception:** detection/tracking of a planar fiducial marker on the platform; relative pose estimation via PnP/homography.
2. **Navigation (State estimation):** multi-sensor fusion with an Error-State Extended Kalman Filter (ESEKF) that integrates inertial/VIO odometry of the Unmanned Aerial Vehicle (UAV), barometric altitude, and relative visual measurements.
3. **Guidance:** generation of velocity references $\mathbf{v}^c$ and yaw rate $\dot{\psi}^c$ as a function of the relative error, with descent profiles depending on altitude and lateral error.
4. **Control:** the UAV internal autopilot closes the attitude/altitude loops; the mission provides body/earth-frame velocity setpoints.
5. **Safety:** statistical *gating* (Mahalanobis distance) for measurement rejection, target-loss detection, and abort/recovery rules to a safe altitude.

4.4.2 Reference frames and relative kinematics

Frames are defined as follows: inertial $\mathcal{I}$ (NED), UAV body $\mathcal{B}$, camera $\mathcal{C}$, and platform $\mathcal{L}$ (z -axis normal to the marker plane). Rotations are represented by matrices $\mathbf{R}_{AB} \in \text{SO}(3)$ (from frame B to A) and unit quaternions q_{AB} when convenient.

Let $\mathbf{p}_d, \mathbf{v}_d$ be the UAV position and velocity in $\mathcal{I}$, and $\mathbf{p}_\ell, \mathbf{v}_\ell$ those of the platform. The central mission variable is the *relative position* expressed in the platform frame $\mathcal{L}$:

$$\mathbf{r} := \mathbf{R}_{\mathcal{L}\mathcal{I}}(\mathbf{p}_d - \mathbf{p}_\ell) \in \mathbb{R}^3, \quad \dot{\mathbf{r}} = \mathbf{v}_r. \quad (9)$$

Over landing time horizons, a CV/CA (constant velocity/acceleration) model is adopted for the **relative dynamics**:

$$\dot{\mathbf{r}} = \mathbf{v}_r, \quad \dot{\mathbf{v}}_r = \mathbf{w}_a, \quad (10)$$

$$\dot{\psi}_r = \omega_\psi, \quad \dot{\omega}_\psi = w_\omega, \quad (11)$$

with $\mathbf{w}_a \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_a)$ and $w_\omega \sim \mathcal{N}(0, \sigma_\omega^2)$ modeling unmeasured platform accelerations/rotations and perturbations.

Discretization with sampling period T_s yields the state $\mathbf{x}_k = [\mathbf{r}^\top, \mathbf{v}_r^\top, \psi_r, \omega_\psi]^\top$ and the piecewise-linear dynamics

$$\mathbf{x}_{k+1} = \mathbf{F} \mathbf{x}_k + \mathbf{G} \mathbf{w}_k, \quad \mathbf{F} = \begin{bmatrix} \mathbf{I}_3 & T_s \mathbf{I}_3 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_3 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 1 & T_s \\ \mathbf{0} & \mathbf{0} & 0 & 1 \end{bmatrix}, \quad (12)$$

$$\mathbf{Q} = \text{blkdiag}(\mathbf{Q}_r, \mathbf{Q}_v, \sigma_\psi^2, \sigma_\omega^2)$$

4.4.3 Visual measurement model and planar geometry

The platform carries a square planar marker of side L on the plane $z_{\mathcal{L}} = 0$. With a calibrated camera (intrinsic matrix $\mathbf{K}$), the projection of coplanar points satisfies a homography $\hat{\mathbf{u}} \sim \mathbf{H} \hat{\mathbf{p}}$ (homogeneous coordinates). The relative pose $\{\mathbf{r}, \psi_r\}$ is estimated via PnP on marker corners or directly with *solvePnP* using the known 3D outline. The discrete **measurement model** used by the fusion is:

$$\mathbf{z}_k^{(\text{vis})} = [x \ y \ z \ \psi_r]^\top + \mathbf{v}_k^{(\text{vis})}, \quad \mathbf{v}_k^{(\text{vis})} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{\text{vis}}), \quad (13)$$

complemented by barometric altitude z and/or a laser altimeter when available:

$$z_k^{(\text{alt})} = z_k + v_k^{(\text{alt})}, \quad v_k^{(\text{alt})} \sim \mathcal{N}(0, \sigma_{\text{alt}}^2). \quad (14)$$

To be robust against smoke/occlusions, the visual pipeline applies adaptive histogram equalization (CLAHE), glare suppression, and geometric validation (quadrilateral consistency and corner ordering). Statistical acceptance of a measurement is decided via Mahalanobis *gating*.

4.4.4 State estimation via ESEKF

An ESEKF is adopted for its computational efficiency and ability to handle moderate nonlinearities in the observation

model 13. The nominal state follows 12; updates correct the linearized error $\delta \mathbf{x}$. Given the measurement model $\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k$ with $\mathbf{h}(\mathbf{x}) = [x, y, z, \psi_r]^\top$, linearization yields $\mathbf{H}_k = \frac{\partial \mathbf{h}}{\partial \mathbf{x}}|_{\hat{\mathbf{x}}_k^-}$, which in this case is a selection of identity columns.

Prediction

$$\hat{\mathbf{x}}_k^- = \mathbf{F} \hat{\mathbf{x}}_{k-1}, \quad (15)$$

$$\mathbf{P}_k^- = \mathbf{F} \mathbf{P}_{k-1} \mathbf{F}^\top + \mathbf{Q}. \quad (16)$$

Update With innovation $\boldsymbol{\nu}_k = \mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-)$ and innovation covariance $\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^\top + \mathbf{R}$, the filter computes

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^\top \mathbf{S}_k^{-1}, \quad (17)$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \boldsymbol{\nu}_k, \quad (18)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^-. \quad (19)$$

Robust gating and outlier rejection The Mahalanobis metric

$$d_k^2 = \boldsymbol{\nu}_k^\top \mathbf{S}_k^{-1} \boldsymbol{\nu}_k \quad (20)$$

is compared against a threshold γ (a χ^2 quantile) to decide acceptance; if $d_k^2 > \gamma$, the measurement is discarded and the filter continues in *hold* (prediction-only) mode. Additionally, a process-noise *decay*/inflation on $\mathbf{Q}$ is used to increase uncertainty in the absence of measurements and allow smooth re-coupling when the target reappears.

Notes on alternatives An Unscented Kalman Filter (UKF) is viable when the observation is strongly nonlinear (e.g., poses derived directly from homographies without explicit decomposition), at a higher computational cost. In our tests, the local linearity of 13 allows the ESEKF to achieve equivalent performance with lower latency.

4.4.5 Guidance and control for approach-descent

The mission uses a **kinematic guidance law** that generates velocity setpoints in the platform frame $\mathcal{L}$, later mapped to the UAV control frame. Define the error $\mathbf{e}_p = [x \ y]^\top$ and its derivative $\mathbf{e}_v = [\dot{x} \ \dot{y}]^\top$. The desired yaw reference is $\psi_r^* = 0$ (alignment with the marker).

Lateral control law

$$\mathbf{v}_{xy}^c = -\text{sat}_{v_{\max}}(\mathbf{K}_p \mathbf{e}_p), \quad (21)$$

$$\dot{\psi}^c = -k_\psi \psi_r, \quad \psi_r \in (-\pi, \pi]. \quad (22)$$

Gains $\mathbf{K}_p = \text{diag}(k_x, k_y)$ and $\mathbf{K}_d = \text{diag}(d_x, d_y)$ are tuned by pole placement or LQR on the constant-velocity lateral model. Saturation enforces safety limits and prevents *windup*.

Error-dependent descent profile The vertical component follows a *schedule* as a function of altitude z and the normed lateral error $\|\mathbf{e}_p\|$:

$$v_z^c = -\min\{v_{z,\max}, a_z z + b_z\} \cdot \sigma\left(\frac{\varepsilon - \|\mathbf{e}_p\|}{\delta}\right), \quad (23)$$

with $\sigma(\cdot)$ a smooth sigmoid that disables descent when the lateral error exceeds ε (hysteresis δ), encouraging *lateral stabilization first*.

Frame transport and setpoints Setpoints are generated in $\mathcal{L}$ and transformed to the frame required by the UAV velocity controller (*earth/body*) using $\mathbf{R}_{\mathcal{L}}$ and the global yaw estimate. *Rate limits* and low-pass filtering are applied to respect the autopilot bandwidths.

4.4.6 Mode logic and safety

Three modes are used: (i) **Search/Lock**: flight at a safe height z_{safe} until the marker is detected with $d_k^2 \leq \gamma$ for N_{lock} samples; (ii) **Approach**: active lateral control with descent inhibited until $\|\mathbf{e}_p\| \leq \varepsilon_1$; (iii) **Descent**: vertical profile enabled and tighter limits on $\|\mathbf{e}_p\|$ and $|\psi_r|$. Abort rule: if the measurement is lost for $> T_{\text{lost}}$ or $d_k^2 > \gamma$ repeatedly, switch to climb to z_{safe} and *loiter*.

4.4.7 Handling latencies and multi-rate sensing

Sensors operate at different rates (camera ~ 30 Hz, barometer ~ 50 Hz, VIO/IMU ≥ 100 Hz). The ESEKF runs at the highest available rate; asynchronous measurements are incorporated via precise timestamping and interim propagation. For a fixed visual latency τ , first-order state extrapolation to the present is used: $\hat{\mathbf{x}}(t) \approx \hat{\mathbf{x}}(t - \tau) + \dot{\hat{\mathbf{x}}}(t - \tau) \tau$ prior to the update.

4.4.8 Specific implementation and command channels

The deployment integrates with the UAV high-level control interface, sending horizontal/vertical velocity and yaw-rate setpoints to the inner attitude loop. ESEKF states and covariances are published to the guidance layer; *gating* and mode logic reside in the same process to minimize latency.

4.4.9 Robustness considerations under smoke and occlusions

To mitigate smoke and glare artifacts, the visual pipeline includes: (i) temporal pre-filtering and block-wise *CLAH*; (ii) geometric marker validation (side ratios, convexity, orientation); (iii) EKF-prediction-driven tracking to bound the ROI; (iv) Mahalanobis-distance rejection and *lost-reacquire* tracking. Combined with the multi-rate ESEKF, these measures preserve stability during brief visual dropouts.

5 CONCLUSION

This Technical Description Paper has presented the team's approach to the indoor event of the IMAV 2025 competition. Each of the four tasks posed distinct challenges that required tailored strategies, ranging from vision-based controllers to open-loop control methods. By combining these approaches, the team sought to balance robustness, precision, and adaptability under the diverse conditions represented by the competition tasks.

All controllers have been extensively tested on a replica of the competition arena built in the institute's laboratory. This testing environment has allowed the team to evaluate system performance, identify limitations, and progressively refine the control strategies. Although the replica provides a valuable preparation tool, the team acknowledges that the actual competition setting will inevitably present new difficulties, such as lighting variations, communication delays, or unexpected environmental disturbances.

For this reason, ongoing efforts are dedicated to improving the controllers and enhancing their resilience to unmodeled scenarios. Future work will focus on integrating adaptive mechanisms, optimizing parameters for increased reliability, and reinforcing the UAV's perception capabilities to better cope with real-world uncertainties. With these continuous improvements, the team aims not only to perform successfully in the competition but also to contribute to the broader development of UAV technologies applicable to disaster response and other critical missions.

ACKNOWLEDGEMENTS

All student members of the team gratefully acknowledge the Instituto Tecnológico Autónomo de México for the financial support provided, which has made it possible to prepare for and participate in the IMAV 2025 competition.

REFERENCES

- [1] Koki Yakushiji, Hiroshi Fujita, Mikio Murata, Naoki Hiroi, Yuuichi Hamabe, and Fumiatsu Yakushiji. Short-range transportation using unmanned aerial vehicles (uavs) during disasters in japan. *Drones*, 4(4), 2020.
- [2] Balmukund Mishra, Deepak Garg, Pratik Narang, and Vipul Mishra. Drone-surveillance for search and rescue in natural disaster. *Computer Communications*, 156:1–10, 2020.
- [3] Eugene A Morelli and Vladislav Klein. *Aircraft system identification: theory and practice*, volume 2. Sunflyte Enterprises Williamsburg, VA, 2016.