AIAA SciTech Forum
12-16 January 2026, Orlando, FL
AIAA SCITECH 2026 Forum

10.2514/6.2026-2571

Localization and Predictive Control for Dynamic Object Interception on Lighter-than-Air UAVs

Matthew Dempsey*, Caeden Taylor†, and Or D. Dantsker‡
Indiana University, Bloomington, IN 47408

Abstract

As the field of aerial unmanned autonomy grows, the requirements for autonomous system capabilities continue to increase in complexity, range, and maneuverability while decreasing required compute resources. In adversarial conditions, systems must additionally track and assess the opposing force's unmanned systems either for avoidance or collision, with current solutions relaying this information to a ground-station when possible. At the Defend The Republic competition, collegiate teams strive to create competitive LTA vehicles with significant physical constraints in a head-to-head adversarial competition as an exploration into the research necessary for autonomous aerial vehicles in contested airspace. This paper proposes a custom method for adversarial vehicle tracking of LTA UAVs relative to the current vehicle position with inertial motion data. Additionally, measurements of other vehicles' speed, position, and orientation are used to continuously track and predict their future position's while not being actively observed.

Nomenclature

DTR	=	Defend the Republic competition	*LTA*	=	Lighter-Than-Air
FOV	=	Field of View	*PID*	=	Proportional-Integral-Derivative
HUD	=	Heads-Up Display	*UAV*	=	Unmanned Aerial Vehicle

I. Introduction

Autonomous aerial robots are increasingly used in inspection, logistics, environmental monitoring, and defense, where they must operate in cluttered, dynamic environments; often without reliable GPS or communication. These scenarios impose strict constraints on vehicle size, weight, and power, limiting the scope of onboard sensors and computing. As mission profiles become more complex and demand longer endurance and persistent situational awareness, these limitations grow more restrictive.

Within this area, LTA UAVs offer a compelling but underexplored alternative. By generating lift through buoyant gas rather than purely rotor thrust, LTA platforms achieve long endurance and stable flight, which are well suited for tasks like communication relays, indoor 3-dimensional navigation, and safe operation near people and structures. However, the limited lifting capacity imposes stringent payload restrictions, forcing trade-offs between sensing, communication, and propulsion components. These characteristics make LTA vehicles an effective test bed for autonomous navigation in constrained adversarial settings. While they can remain airborne for extended periods and tolerate contact with obstacles, LTA's demand ultralight payloads and power efficient control systems.

In this work, the DTR collegiate competition is used as an experimental environment for developing and evaluating predictive vision-based autonomy for LTA UAS. At DTR, teams field autonomous LTA platforms that must capture neutrally buoyant helium balloons and score them through suspended goals in a configuration resembling robotic

*Undergraduate Student, Department of Intelligent Systems Engineering, mattdemp@iu.edu
†Graduate Student, Department of Intelligent Systems Engineering, AIAA Student Member. caedtayl@iu.edu
‡Assistant Professor, Department of Intelligent Systems Engineering, AIAA Member. odantske@iu.edu

"Quidditch". The matches take place in large indoor arenas with multiple free-floating targets, retro-reflective goals mounted near the ceiling, and several vehicles operating concurrently in contested airspace. The competition emphasizes end-to-end system design, including vehicle architecture, perception, communication, and autonomy, under strict limits on helium volume, total vehicle mass, and allowable negative buoyancy. These constraints force explicit trade-offs among sensing payload, onboard and off-board computation, and actuation capability when deploying a fleet of blimps. Figure 1 shows three friendly autonomous agents and one adversarial system during a game at DTR.

Figure 1: Autonomous Blimps designed by Indiana University students (blue) competing at LeHigh University against an opposing team (red) in Defend the Republic.

Autonomy for LTA vehicles in DTR-style environments is still dominated by reactive, camera-frame policies.[1–3] Detections from a forward-facing camera are converted directly into control commands, often through proportional or PID mappings based on pixel offsets, without maintaining a persistent estimate of a target vehicle state in an inertial reference frame. This structure is simple to implement and can perform adequately when a single balloon or goal remains continuously visible. Both onboard and off-board vision pipelines typically inherit this stateless design.[4,5] Onboard implementations often allocate nearly all available computation to running the detector and low-level control, leaving no provision for a predictive inertial reference frame state estimate.[4] Remote-processing architectures, where images are streamed to a ground computer for higher-capacity inference and control, relax onboard computation requirements but introduce variable latency, jitter, and packet loss on the wireless link.[5] In many such systems, control actions still depend directly on the most recent detections, with little tolerance for occlusion, so modest disruptions or brief perception gaps can yield stale commands and loss of coherent autonomous behavior.

Predictive state estimators provide a natural way to exploit the computational advantages of off-board processing while mitigating its communication risks. Kalman filtering[6] and related methods combine motion models with noisy sensor data to produce a stable and consistent estimate of agent and target state, expressed in an inertial frame and accompanied by uncertainty.[7–9] At each step, a prediction advances this state forward in time, and a correction phase incorporates new measurements according to their reliability.[10] When observations are delayed or briefly unavailable, the prediction step alone can predict or interpolate state information so the autonomy stack retains a usable notion of object permanence rather than resetting on each frame.[11] In the context of DTR-style adversarial arenas, this predictive memory model is advantageous due to the combination of intermittent visual contact and time-varying network delay. The architecture developed in this work therefore uses a predictive inertial frame estimator for the off board control loop, maintaining self and adversary state despite occlusions and communication variability. By estimating motion of targets, this architecture enables predictive target interception,[12] rather than reactive control schemes.[13]

This work presents three primary contributions for lighter-than-air vehicle localization and state estimation. First, an ultralight vehicle platform was developed with off-board computation for high level data processing and control. Second, a local-frame state estimation architecture is developed, which fuses inertial IMU data with barometer altitude

information with visual object detection for persistent position estimates. This persistent internal model allows for estimation of objects' positions beyond the sensing range of the developed system. Finally, the resulting framework produces control vectors to navigate towards a future estimate of an objects position and tracking based on estimated linear velocities, resulting in intercept capabilities, and target re-acquisition based on recent system memory.

This work is organized as follows: Section II describes the physical platform and localization framework with object permanence, and how predicted positions are used for navigation. Section III analyzes experimental results from three separate testing scenarios to showcase the prediction and localization capabilities of the complete architecture. Section IV summarizes our contributions and findings from the tested scenarios. Finally, Section V details future directions of this work.

II. System Design

This section details the system architecture designed to enable predictive autonomy on resource-constrained aerial platforms. First, the physical hardware configuration is presented, which prioritizes weight reduction and neutrally buoyant operation. The estimation framework is then defined, detailing the fusion of visual object detection with inertial odometry. This architecture transforms raw sensor data into persistent state estimates, allowing the vehicle to track, maintain memory of, and predict the motion of both itself and dynamic targets. The system diagram of this architecture is given in Fig. 4.

A. Hardware Platform Overview

The platform used consisted of a small, custom-built LTA UAS. It was selected for use due to its ability to provide long-endurance, stable flight suitable for testing predictive algorithms. This configuration enables the vehicle to operate with a minimal on-board payload, focusing only on critical sensing and actuation while offloading intensive processing to a ground station via a low-latency wireless link. Table 1 lists the specific components chosen and their respective weight.

Component	**Weight (g)**
XIAO ESP32S3 Sense w/OV2640 (x1)	8.0
BNO055 & BMP280 BFF Backpack	5.0
Li-Ion Battery (3.7V 350mAh) (x1)	10.0
1S 7A Brushless Motor controller (x3)	8.0
Brushless Motors & Propellers (x3)	10.0
3D Printed Structure	20.0
Total Payload Mass	**61.0**

Table 1: Weight Budget for prediction and localization platform

The mylar balloon used for the vehicle has an effective carrying capacity of 70 grams, making the platform neutrally buoyant with 9 grams of additional mass for wiring, ballast, and cable management. The platform uses light weight and low cost sensing components in order to minimize the required onboard payload. The resulting system uses differential drive with two side-mounted thrusters, and a third rear-mounted thruster for vertical control. Data is transmitted and received via an onboard WiFi link. Fig. 2 is a top-down view of the platform without the metallized mylar balloon or lithium-ion battery.

B. Predictive Autonomy Framework

The predictive autonomy framework fuses sensing, estimation, prediction, and control into a single closed-loop architecture that maintains a continuous internal model of both the vehicle and its targets. Rather than reacting

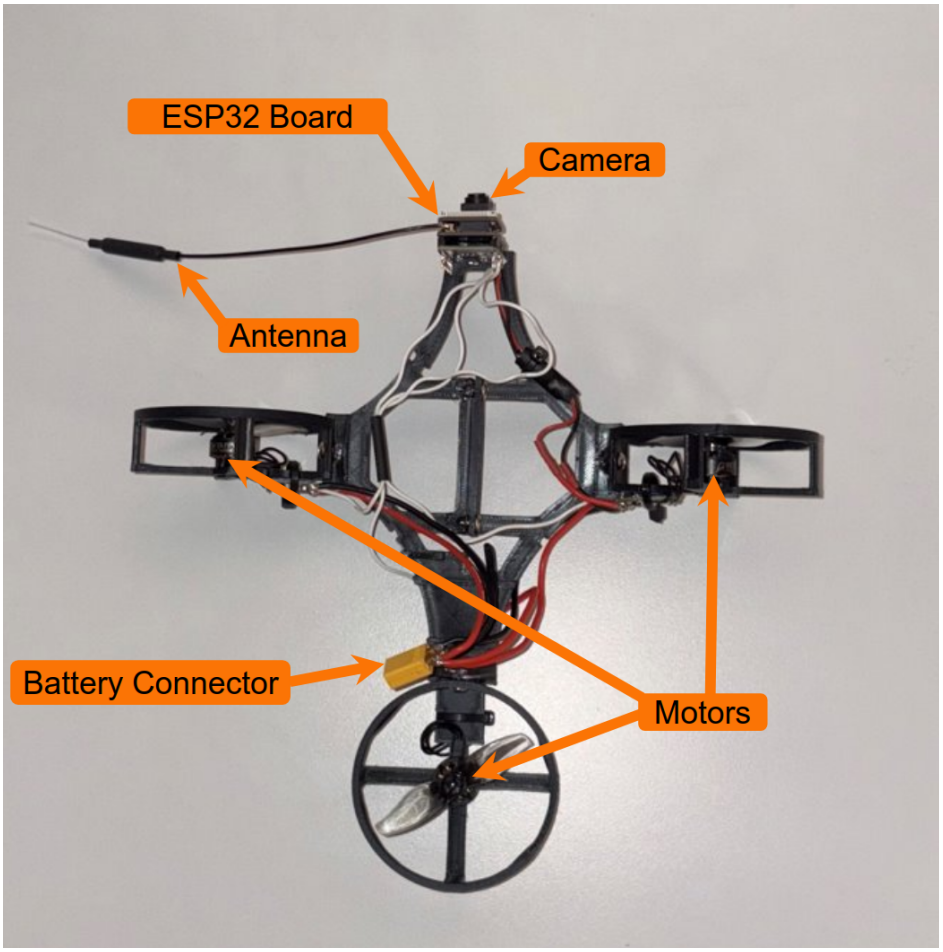

Figure 2: Top-down view of the experimental vehicle platform, not attached to mylar balloon.

independently to each frame, the system propagates state estimates of previously detected target objects forward in time, filling gaps during visual dropouts and uses predicted future states to drive an intercept-oriented controller. All agent and target states are maintained in an inertial reference frame relative to the vehicle and are rendered in an egocentric visualization where the blimp remains fixed at a center fixed location and other entities move relative to it.

The framework is implemented as an off-board control loop connected to the blimp platform. The blimp provides all sensor measurements including: camera images, IMU measurements, and barometer readings. Additionally, it gives actuation commands sent from the ground station to all on-board motors. The ground station performs vision detection on the provided images, agent and target state estimation, short-horizon prediction of target objects, and control computation. An optional real-time heads-up display (HUD) of the inertial reference frame is provided to the operator.

1. State Estimation and Memory Model

The predictive autonomy framework maintains an inertial reference frame from the agent and uses the relative location of all tracked targets to make decisions. The agent's state is represented by its estimated horizontal position, orientation, and velocity. The target's state is represented by it's estimated position and velocity. All motion is modeled in the horizontal plane; altitude is estimated and controlled in a separate vertical PID controller and is not fused into the motion filters. This separation is sufficient for the low-speed, stable flight envelopes with high drag force balloons, and simplifies the motion computation for target bodies.

Each control cycle begins by sampling an image, three-axis acceleration measurements, yaw angle, and altitude. The camera frame is streamed to a ground station, where a trained YOLOv8 object detection model[14] extracts class-specific bounding boxes, confidence scores, and pixel-space offsets. Detections of the configured target balloon class above a confidence threshold, set at 60%, are retained. For each retained detection, the vision geometry produces a distance and a angular position from the camera frame. Given the current yaw and agent position, the target's positional values are rotated into the agent's inertial reference frame and added to the target's respective position estimate. These relative positions serve as noisy measurements for the target-tracking filters, where the noise comes from range estimation and pixel-level bounding box jitter.

The agent's x and y motion is estimated using a two-dimensional velocity filter that integrates measured onboard accelerations into velocities. At each iteration, measured accelerations are rotated into a north oriented local frame of reference using the current yaw. The filter maintains a short sliding window of the most recent accelerations (five samples) and computes the mean acceleration over this window. The average acceleration is compared to a threshold (typically 0.1 m/s^2). When the threshold is met, typically when motors are active, simple integration updates the relative velocity. Otherwise, the filter suppresses further integration and instead applies an exponential decay, so that negligible velocities are clamped to zero. In both cases, estimated agent velocity updates the agent position. Noise and decay parameters are tuned so that intentional motion is captured, while drift from accelerometer bias is minimized due to the short time horizon. This velocity-filter formulation provides a low-complexity alternative to full pose estimation that is well-matched to the slow, smooth dynamics of the blimp observed during its operation.

Target motion is modeled using a set of independent constant-velocity filters and models, one per tracked balloon, each maintaining a state of it's 2-dimensional position and velocity. At each frame, all active filters perform a prediction step, advancing position by the current velocity over the time step under a noise term that captures unmodeled accelerations due to small accelerations and constant drifts. New measurements are then associated to predicted targets using nearest-neighbor matching in the inertial compensated coordinates. Unmatched detections spawn new tracks initialized with zero velocity and high initial uncertainty. Tracks with no associated measurements are propagated forward purely by a constant-velocity model and are removed if they are not seen after a certain amount of time.

This combination of inertial velocity filtering for the agent, constant-velocity tracking for targets, and acceleration-based gating yields a simple linear estimator of the different states of the agent system and the environment. The 2-dimensional assumption and constant-velocity model are sufficient to represent low-speed objectives with modest drift, while the persistent target memory allows the system to maintain unseen targets beyond the platform's sensing capabilities.

2. Control Layer

The control layer uses the agent and target state estimates to generate motor-level commands in either manual or autonomous modes. State estimation and HUD visualization run continuously in both cases. In Manual Control, the operator commands forward–back, left–right, and up–down thrust directly, while the vertical PID controllers are reset each cycle so that integral terms do not accumulate when the human is in the loop. In autopilot, the controller drives all three axes: forward thrust is set by a fixed feed-forward term, lateral thrust is produced by a PID regulator acting on body-frame lateral error, and vertical thrust is governed by a pair of PIDs that switch between vision-based and memory-based operation.

In intercept mode, the framework selects the closest target for intercept and computes a predicted trajectory that intersects the predicted point in the horizontal plane. This is done by giving the agent a known maximum achievable velocity. It then finds the earliest intercept time and position based on current target velocity estimates. If no feasible intercept position is found, the controller falls back to a direct-pursuit approximation that steers toward a point along the target's current direction of travel using a limited time horizon time-of-flight estimate.

Given the point of interception in relative agent centered coordinates, the controller computes an error vector by subtracting the agent position and rotating into the yaw-aligned frame. The component perpendicular to the forward axis serves as the input to a lateral PID that generates left–right thrust, while forward thrust is provided by a constant command tuned for nominal cruising speed. The lateral PID includes integral anti-windup and output clamping to

prevent excessive commands under large lateral errors or temporary loss of visual lock; gains are tuned empirically on the platform to balance responsiveness against overshoot and to remain robust under communication latency and unmodeled disturbances.

Vertical motion is regulated by a dedicated loop with two modes. When a target is visually locked, the controller performs a vision-based altitude adjustment. The set point used is an image-plane vertical offset and the measurement is based on the target's vertical error, driving the vehicle toward a chosen vertical alignment in the camera view. When only unseen tracks remain, the controller switches to a memory-based mode in which the altitude set point is initialized to the altitude at last sighting of the nearest object and then decays smoothly back toward a configurable base altitude over the track's persistence horizon. If no tracks are active, the controller either maintains the base altitude using an altitude PID or leaves the PID effectively neutral so that only a feed-forward thrust term is applied, depending on configuration. Vertical PID integrators are reset whenever the system transitions between a target locked and unseen target states. It is also set whenever manual control is active, ensuring that mode switches do not introduce sudden, integrator-driven changes. Fig. 3 shows the flow of processes during autonomous prediction.

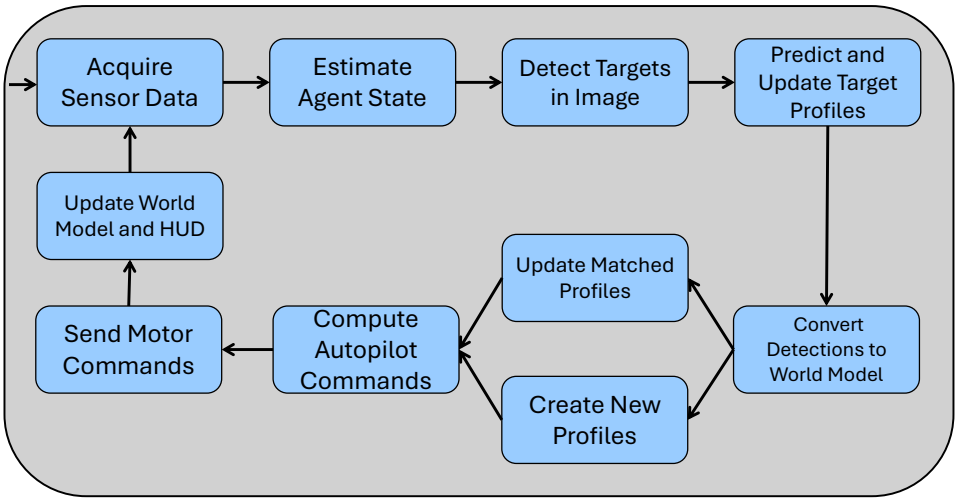

Figure 3: Data flow diagram illustrating the predictive autonomy architecture.

III. Experiments

Flight experiments were conducted to evaluate three core behaviors supported by the predictive autonomy framework. This included a vision-based interception of a stationary target from an offset initial pose, evaluation of the persistence of a static target state through deliberate loss of line-of-sight, and prediction-based tracking and reacquisition of a moving target after it exits the camera field of view. Each behavior was evaluated in a dedicated testing trial, with emphasis on characteristic behavior and failure modes rather than aggregate statistics.

Tests took place in an indoor high-bay warehouse with open floor space and uniform overhead industrial lighting. The flight volume provided several meters of clearance in all directions, allowing the lighter-than-air platform to maneuver in translation and yaw while maintaining safe separation from walls and structural elements. The visual background consisted primarily of exposed walls, shelving, and overhead fixtures, producing moderate clutter without strong specular reflections or outdoor lighting variation.

The vehicle configuration and off-board processing stack matched the platform and predictive autonomy architecture described. A manual mode allowed the operator to issue motor commands for positioning the vehicle, while the full perception, estimation, prediction, and visualization pipelines operated identically to autonomous use. In the stationary intercept scenario, the off-board controller directly commanded motor outputs to close the loop on the estimated target state. In the occlusion and moving-target scenarios, vehicle motion was executed manually while the same predictive internal world model and unseen-tracking mechanisms were used to evaluate estimation and visualization behavior.

A. Scenario 1: Stationary Target Intercept

The first scenario evaluated vision-based intercept of a static target balloon from an offset initial pose under closed-loop off-board control. A helium-filled balloon was suspended approximately 1 m above the floor and 2 m in front of the vehicle. The lighter-than-air platform was manually positioned at a comparable altitude, roughly facing the target with a small yaw offset such that the balloon appeared in the camera field of view but was not centered. Autonomous intercept mode was then engaged, with the off-board controller applying forward thrust and lateral image-based corrections based on the estimated target state for the duration of the trial. Figure 4 shows an overhead mock-diagram of the agent and target's motion over time.

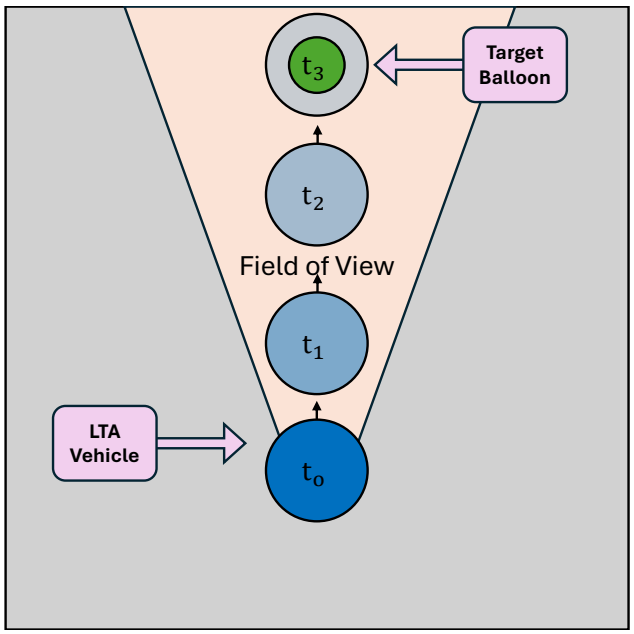

Figure 4: Overhead view of Scenario 1

B. Scenario 2: Static Target Memory Under Loss of Visual Contact

The second scenario evaluated whether a static target could be maintained in the internal model used for target directionality through a deliberate, camera-induced loss of visual contact and then used to recover the correct heading. A balloon was held fixed at a constant position, and the blimp was oriented such that the target appeared near the center of the image. The estimator was allowed to track the balloon until the state converged to a small, stable covariance, after which the operator applied an approximately 90° yaw rotation away from the target with minimal translation, causing the balloon to leave the camera field of view. The vehicle maintained this out-of-frame heading for a short interval, and was then yawed back toward the calculated known bearing based solely on the propagated local-frame estimate. Figure 5 is an overhead representation of the agent turning in place to show the persistence of tracking for the stationary target.

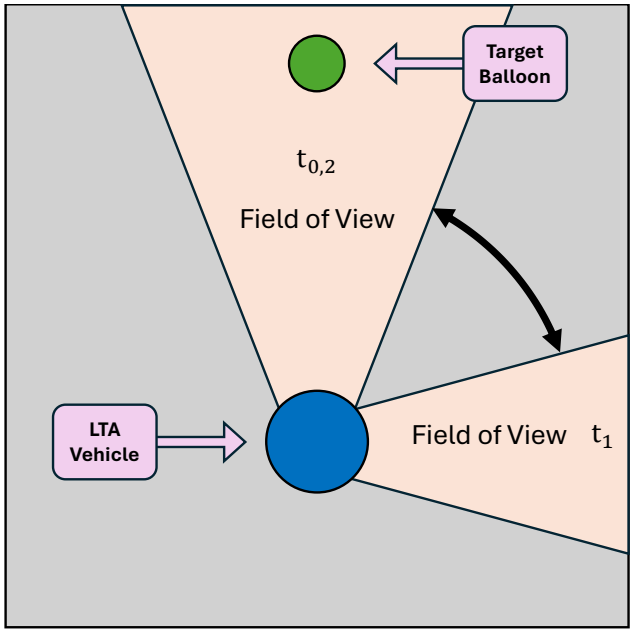

Figure 5: Overhead view of Scenario 2

C. Scenario 3: Moving Target Prediction and Reacquisition

The third scenario evaluated prediction-based tracking and reacquisition of a moving target after it left the camera field of view. A human operator carried the balloon laterally across the front of the blimp at approximately constant speed, producing a smooth, roughly linear trajectory in the horizontal plane at nearly fixed range. The balloon entered the image from one edge, at which point the target filter initialized position and velocity from successive detections and operated in a constant-velocity mode while the vehicle remained approximately stationary in translation. After the balloon exited the camera field of view, the filter propagated the target state without new measurements for a short interval, and the vehicle was then yawed toward the bearing of the predicted target location. Figure 6 represents the agent in a stationary position, with the target moving across the field of view with a continuous prediction beyond the edge of where the camera had last detected the target moving.

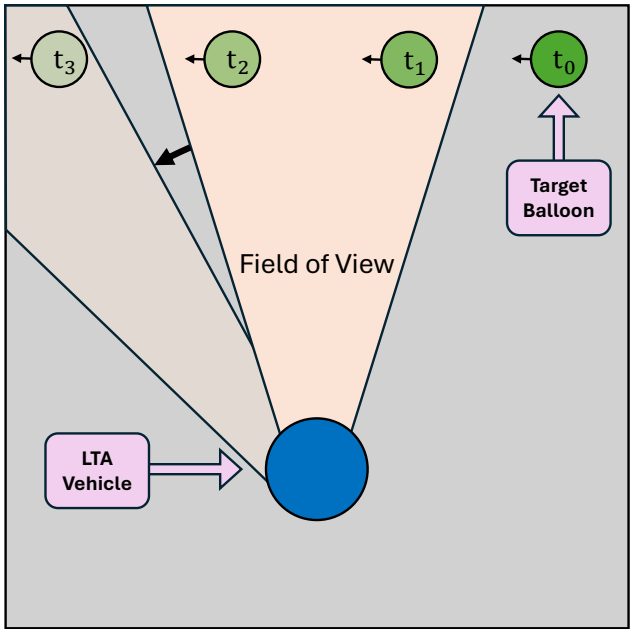

Figure 6: Overhead view of Scenario 3

IV. Results

The inertial reference frame of the agent is displayed through a Pygame-based HUD[15] that provides egocentric situational awareness and supports debugging and tuning. The display renders a yaw-aligned grid centered on the blimp, with the agent fixed at the screen center and all targets and intercept points drawn relative to this origin using the same coordinate transforms employed by the controller. Locked targets appear as solid icons with annotated ranges and dashed range lines; unseen tracks are rendered as semi-transparent markers that fade over their persistence horizon. For the selected target, the predicted intercept point and time-to-intercept are drawn with a distinct icon and labeled trajectory.

HUD panels around the central map summarize derived quantities, including estimated speed, corrected altitude, control mode, number of locked and unseen targets, the distance to the closest target, and the current targeting mode. Additional indicators show the acceleration gate state and whether the system is in manual or autopilot operation. By mirroring the estimator and controller states in real time, the visualization provides an interpretable window into the predictive autonomy loop during bench tests and flight experiments.

A. Scenario 1: Stationary Target Intercept

In the stationary intercept trial, the vision pipeline acquired the balloon immediately after autonomous mode was engaged and initialized a new target track. The intercept controller applied constant forward thrust with lateral image-based PID corrections, producing a slow lateral oscillation rather than a straight-line approach, and a gradual altitude drift caused the vehicle to descend below the target. Despite having no current view of the target, the internal world map allowed the system to continuously estimate the current target position relative to the vehicle, allowing for 'blind' correction and interception without continuous target sensing. This outcome indicates that the predictive state representation and ghost-tracking mechanism can sustain a successful stationary intercept despite under-tuned lateral control, altitude drift, and brief loss of line-of-sight. Fig. 7 shows this intercept from two stationary observing cameras.

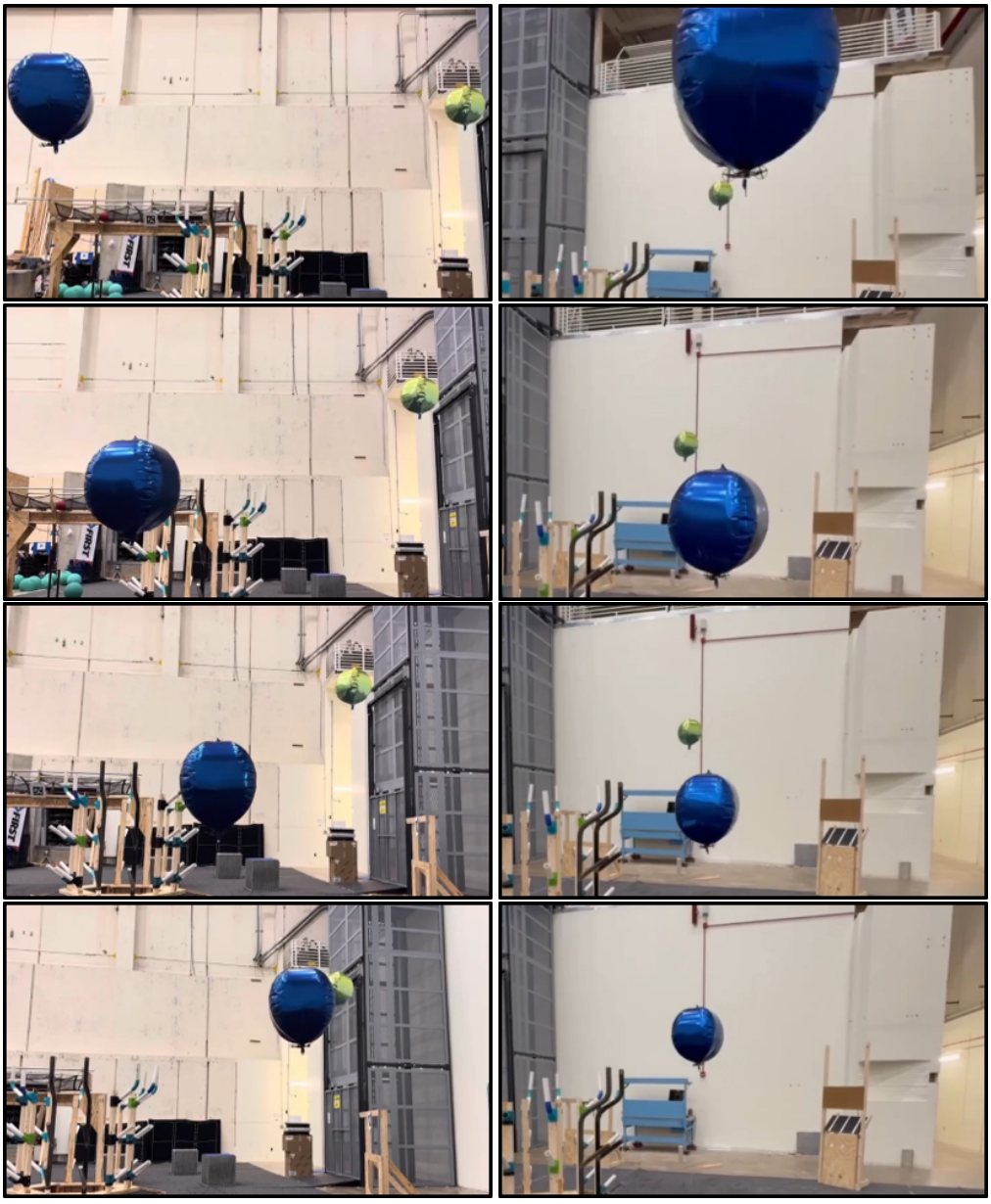

Figure 7: The columns display synchronous Side (Left) and Rear (Right) perspectives, progressing chronologically from top to bottom. (Top Rows) The autonomous agent identifies the target and initiates forward thrust. (Bottom Rows) As the vehicle approaches, the Rear View (Right Column) highlights the vertical drift mentioned in Section IV-A, where the agent descends slightly below the target plane. Despite this deviation and the resulting loss of visual centering, the predictive state estimator maintains a valid intercept solution, resulting in a successful collision in the final frame.

B. Scenario 2: Static Target Memory Under Loss of Visual Contact

In the static-target trial, the estimator first converged to a low-uncertainty initial inertial state for the balloon while it was centered in the image. During the subsequent yaw rotation of 70.8° away from the target, the estimated target position remained fixed in the inertial frame, while in the egocentric HUD the target icon appeared to rotate around the vehicle, consistent with correct decoupling of self-motion and target position. Once the balloon left the camera field of view, detections ceased and the track transitioned from a filled marker to a hollow ghost, with the static-target model propagating the state forward and uncertainty increasing slowly. After a short visual dropout interval, yawing back toward the remembered bearing brought the unseen target toward the forward direction in the HUD. When the camera

re-crossed into line of sight, the balloon reappeared at a bearing closely aligned with the unseen target, was associated with the existing track, and returned to a visually confirmed state. Qualitatively small heading error at reacquisition suggests that the dual-filter inertial frame representation maintains a coherent memory of static targets during short losses of visual contact and supports accurate heading recovery. Fig. 8 depicts the HUD and vehicle camera view seen when tracking the ball during this scenario.

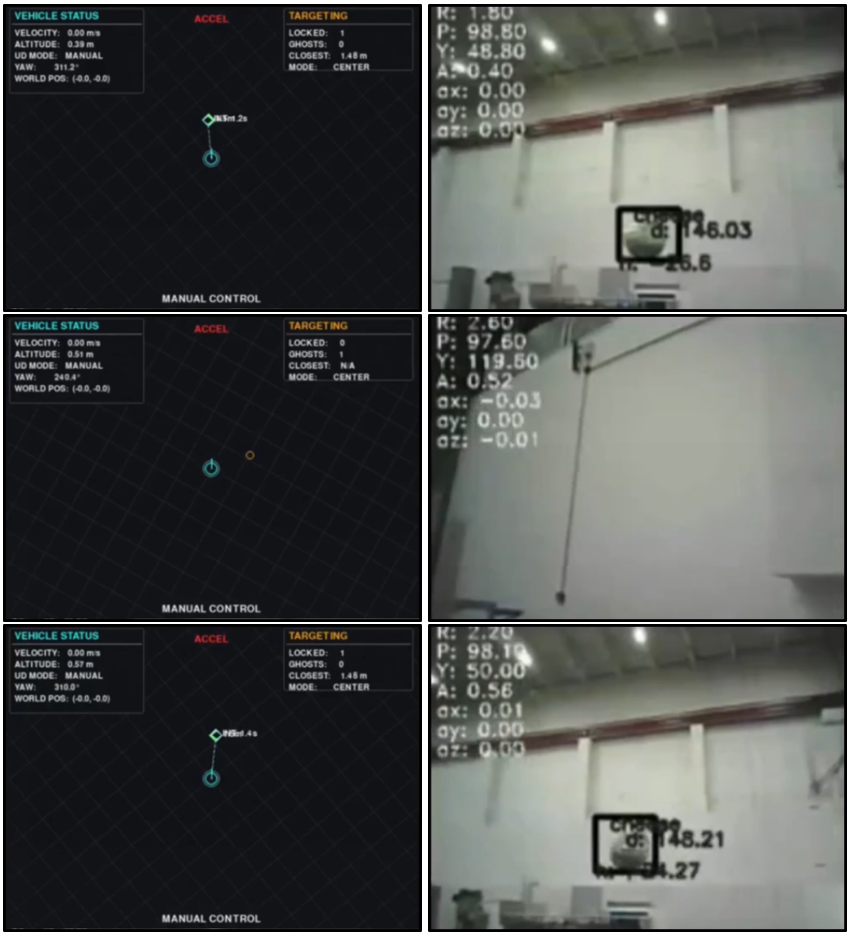

Figure 8: The internal world model (HUD, Left) mirrors the real-time camera feed (Right). (Top Row) The estimator converges on a stable world-frame position for the target (solid diamond). (Middle Row) The vehicle yaws approximately 90° away. The track transitions to an unseen state (hollow marker), maintaining its relative coordinates in the estimator despite the loss of visual contact. (Bottom Row). Upon rotating back to the remembered bearing, the target is immediately re-acquired near the unseen target's location.

C. Scenario 3: Moving Target Prediction and Reacquisition

In the moving-target trial, the balloon traversed laterally across the front of the vehicle at approximately constant speed while the blimp remained nearly stationary in translation. As the balloon entered the image, the target filter initialized position and velocity from successive detections and converged to a constant-velocity estimate; the HUD displayed the estimated position together with a projected intercept vector and nominal time-to-intercept as diagnostic overlays. When the balloon exited the camera field of view, the track transitioned to an unseen state and the Kalman filter propagated the target forward along the estimated velocity, causing the hollow marker to continue moving across the HUD despite the absence of new measurements. After a brief prediction-only interval, the vehicle was yawed toward the predicted bearing, and the real balloon reentered the image near the unseen position. The estimator associated the new detection with the existing track and resumed visually confirmed tracking with only minor bearing correction. These observations

indicate that the constant-velocity predictor and data-association mechanism can again bridge short gaps in observation for slowly moving targets, preserving a coherent track and enabling rapid reacquisition near the predicted line of sight. Figure 9 shows both the predicted relative map HUD and the detected camera frame during this scenario depicting when the ball is sighted, tracked, lost, and reacquired.

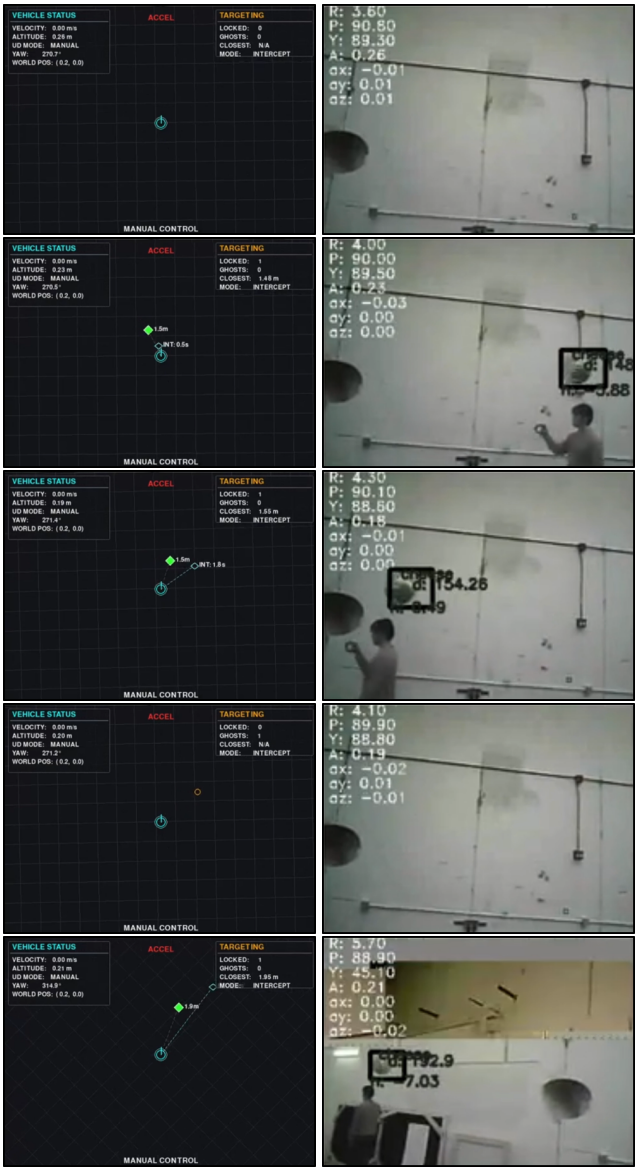

Figure 9: The inertial reference frame (HUD, Left) mirrors the real-time camera feed (Right). (Rows 1–3) The target enters the field of view from the right; the vision system initializes a track (solid diamond) and estimates the target's leftward velocity. (Row 4) The target exits the camera's field of view to the left. The system transitions the track to an unseen state (hollow marker) and propagates its position forward based on the estimated velocity. (Row 5) The vehicle yaws left toward the predicted bearing. The target is immediately re-acquired (solid diamond) near the unseen target's location, confirming the accuracy of the blind prediction.

V. Conclusions and Future Work

This work presents a predictive autonomy framework for lighter-than-air vehicles that maintains a persistent estimate of the agent location relative to the surrounding target locations using low-cost sensing and off-board computation. The

architecture fuses inertial, altitude, yaw, and vision-based range/bearing measurements into planar position and velocity estimates for the vehicle and each tracked balloon, and represents them in an egocentric HUD that displays this data to the user in real time. Through three experimental scenarios: stationary target interception, static target memory under deliberate occlusion, and prediction-based reacquisition of a moving target, the framework demonstrated that a simple constant-velocity model coupled with acceleration-gated velocity estimation can sustain coherent tracks and provide usable intercept solutions even with loss of visual contact and modest control imperfections. These results indicate that predictive local-frame estimation is a practical way to provide a frame of reference and resilient target tracking despite jitter and temporary loss of visual contact in vision-based control of LTA platforms operating in DTR-style environments.

Future work will first focus on systematically improving the closed-loop behavior of the system, including structured tuning of PID gains and estimator parameters to ensure robust performance across variations in latency, target motion, and vehicle dynamics. Additionally, a target priority scoring system that ranks balloons based on distance, relative motion, estimated time-to-intercept, and tracking confidence/occlusion history will be implemented next. In parallel, we aim to integrate this target-selection layer with inter-vehicle communication and world-frame localization for cooperative target selection. Finally, we plan to design quantitative benchmarks, including tracking error, intercept success rate, time-to-intercept, and stability metrics under competitive environments and static controlled conditions to rigorously evaluate the framework.

References

[1]Xu, J., Vu, T., D'Antonio, D. S., and Saldaña, D., "MochiSwarm: A testbed for robotic blimps in realistic environments," 2025.

[2]Mathew, J. P., Karri, D., Yang, J., Zhu, K., Gautam, Y., Nojima-Schmunk, K., Shishika, D., Yao, N., and Nowzari, C., "Lighter-Than-Air Autonomous Ball Capture and Scoring Robot – Design, Development, and Deployment," *arXiv preprint arXiv:2309.06352*, 2023.

[3]Messinger, S., *Modeling, Adaptive Control, and Flight Testing of a Lighter-than-Air Vehicle Validated Using System Identification*, Master's thesis, The Pennsylvania State University, 2022, Master's Thesis.

[4]Widjaja, M., Taylor, C., Meighan, A., and Dantsker, O., "Viability of NPU-Equipped SBCs for Real-Time Onboard Vision Autonomy on Lighter-Than-Air UAVs," *AIAA Aviation Forum*, 2025, AIAA Paper 2025-3546, AIAA Aviation Forum 2025, Las Vegas, NV.

[5]Taylor, C., Widjaja, M., and Dantsker, O. D., "Remotely-Processed Vision-Based Control of Autonomous Lighter-Than-Air UAVs With Real-Time Constraints," AIAA Paper 2025-1344, AIAA SciTech Forum 2025, Orlando, FL. 2025.

[6]Welch, G. and Bishop, G., "An Introduction to the Kalman Filter," Tech. Rep. TR 95-041, University of North Carolina at Chapel Hill, Department of Computer Science, Chapel Hill, NC, 2001.

[7]Yi, P., Jin, G., and Wang, W., "Game Theoretic Non-cooperative Dynamic Target Tracking for Directional Sensing-Enabled Unmanned Aerial Vehicles," *Advanced Intelligent Systems*, Vol. 6, No. 10, 2024, pp. 2300725.

[8]Kalantar, B., Mansor, S. B., Abdul Halin, A., Shafri, H. Z. M., and Zand, M., "Multiple Moving Object Detection From UAV Videos Using Trajectories of Matched Regional Adjacency Graphs," *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 55, No. 9, 2017, pp. 5198–5213.

[9]Conte, C., de Alteriis, G., Schiano Lo Moriello, R., Accardo, D., and Rufino, G., "Drone Trajectory Segmentation for Real-Time and Adaptive Time-Of-Flight Prediction," *Drones*, Vol. 5, No. 3, 2021.

[10]Shukla, P., Shukla, S., and Kumar Singh, A., "Trajectory-Prediction Techniques for Unmanned Aerial Vehicles (UAVs): A Comprehensive Survey," *IEEE Communications Surveys Tutorials*, Vol. 27, No. 3, 2025, pp. 1867–1910.

[11]Yang, K., Zhu, M., Guo, X., Zhang, Y., and Zhou, Y., "Stochastic Potential Game-Based Target Tracking and Encirclement Approach for Multiple Unmanned Aerial Vehicles System," *Drones*, Vol. 9, No. 2, 2025, pp. 103, Copyright - © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/). Notwithstanding the ProQuest Terms and Conditions, you may use this content in accordance with the terms of the License; Last updated - 2025-02-25.

[12]Oliveira, T., Aguiar, A. P., and Encarnação, P., "Moving Path Following for Unmanned Aerial Vehicles With Applications to Single and Multiple Target Tracking Problems," *IEEE Transactions on Robotics*, Vol. 32, No. 5, 2016, pp. 1062–1078.

[13]Russell, J. S., Ye, M., Anderson, B. D. O., Hmam, H., and Sarunic, P., "Cooperative Localization of a GPS-Denied UAV Using Direction-of-Arrival Measurements," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 56, No. 3, 2020, pp. 1966–1978.

[14]Ultralytics, "Model Prediction with YOLOv8," `https://docs.ultralytics.com/modes/predict/`.

[15]Developers, P., "Pygame Documentation," `https://www.pygame.org/docs/`, Accessed: 2025-12-06.