IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 25 November 2025, accepted 14 December 2025, date of publication 22 December 2025,
date of current version 29 December 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3646978

RESEARCH ARTICLE

POF+MADER: Trajectory Planner in Dynamic Environments With Improved Collision Avoidance

SYED IZZAT ULLAH[1], JOSÉ BACA[2], PABLO RANGEL[2], TIANXING CHU[1], AND CARLOS RUBIO-MEDRANO[1]
[1]Department of Computer Science, Texas A&M University-Corpus Christi, Corpus Christi, TX 78412, USA
[2]Department of Engineering, Texas A&M University-Corpus Christi, Corpus Christi, TX 78412, USA

Corresponding author: José Baca (jose.baca@tamucc.edu)

This work was supported in part by the National Science Foundation (NSF) under Grant 2131263, and in part by the Open Access Publication Fund provided by Texas A&M University-Corpus Christi.

ABSTRACT Trajectory planning in dynamic environments is crucial for safe and efficient navigation of UAVs, particularly in scenarios involving unknown dynamic obstacles. The Trajectory Planner in Multiagent and Dynamic Environments (MADER) algorithm has been successfully used for this purpose; however, it assumes perfect knowledge of the future trajectories of the dynamic obstacles. In practical scenarios, the future trajectory of the obstacles may not be known in advance. To address this limitation, we propose POF+MADER, an enhanced version that integrates a probabilistic obstacle filter with adaptive prediction horizons based on obstacle velocity. Our approach integrates a decentralized real-time Kalman Filter-based trajectory estimation with MADER's planning core, eliminating the need for perfect a priori knowledge. We validate our approach through simulations with sensor noise ($\sigma = 0.2 - 0.5m$) to simulate real-world localization uncertainty, with performance benchmarked against the MADER, Robust-MADER, and Ego-Swarm planner baselines. To further demonstrate its practical applicability, we conducted real-world experiments on the Crazyflie UAV platform, comparing its performance to the original MADER algorithm. Results show a 38.75% collision rate reduction in simulation and a 25% reduction in real-world experiments, without a significant increase in average navigation time. Furthermore, even in the presence of real-time sensor-induced noise in obstacle position measurements, our approach consistently outperforms the baseline planners, demonstrating its robustness in trajectory planning under perception uncertainty.

INDEX TERMS Collision avoidance, multi-robot systems, unmanned aerial vehicles (UAVs), autonomous navigation, motion planning, trajectory planning, obstacle tracking.

I. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) have become increasingly prevalent in various applications where autonomous navigation in complex and dynamic environments is essential, such as surveillance, agriculture, package delivery, and disaster response [1], [2], [3]. A crucial aspect of UAV autonomous operation is trajectory planning, which ensures collision-free navigation through static obstacles (e.g., buildings, towers, tree canopies, and wires) and dynamic obstacles (e.g., pedestrians, ground vehicles, balloons, other drones, or even birds). Trajectory planning is especially challenging when the future motion of obstacles is unknown or uncertain.

Dynamic obstacles encountered by UAVs can be broadly categorized as cooperative (sharing trajectory or intent information) or non-cooperative (offering no such information), and as interactive (adapting to the UAV's behavior) or non-interactive (moving independently) [4]. In practical scenarios, UAVs often face non-cooperative, non-interactive obstacles, such as pedestrians, vehicles, unregistered drones, or even birds, whose future trajectories are not available a priori and may be affected by sensor noise and environmental uncertainty.

Various approaches in the literature have been proposed to provide UAV trajectories that handle static obstacles,

The associate editor coordinating the review of this manuscript and approving it for publication was Zheng Chen.

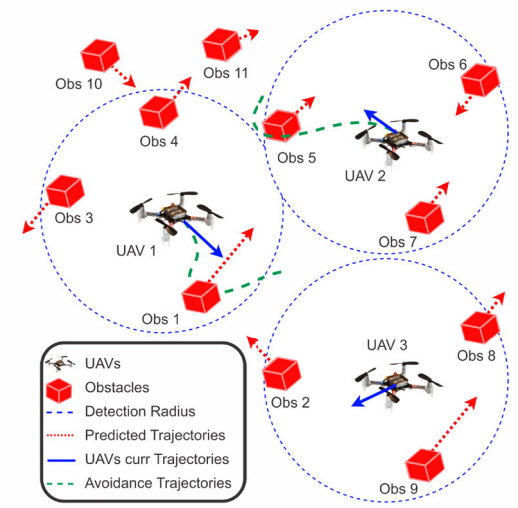


FIGURE 1. A typical multi-UAVs multi-dynamic obstacles scenario where the UAVs detect and estimate the future trajectories of dynamic obstacles within the safety radius of the UAVs to plan obstacle-free trajectories and avoid collisions.

dynamic obstacles, and interactions with other agents [5], [6], [7]. Among these, MADER [8] and its communication-relaxed extension, Robust-MADER (RMADER) [9], have demonstrated strong performance in handling multiagent systems to avoid dynamic obstacles. However, both approaches inherently assume that the future trajectories of dynamic obstacles are perfectly known to the planner. This assumption is rarely satisfied in real-world deployments. Therefore, during trajectory planning, considering the uncertainties of the dynamic obstacle's maneuvers is important for efficiently avoiding collisions. The collision avoidance component is especially important when flying in dynamic environments, to consistently detect the position of dynamic obstacles and estimate their future trajectories, as depicted in Fig. 1.

To alleviate this limitation of MADER [8] and RMADER [9], in this work, we propose a Probabilistic Obstacle Filter (POF) coupled with the MADER and RMADER algorithms, based on the Kalman Filter (KF) [10] to estimate the future trajectories of non-cooperative, non-interactive dynamic obstacles in real-time. We call this extension POF+MADER. Unlike prior approaches that either require perfect knowledge [8], [9] or rely on computationally intensive learning-based predictors [11], POF+MADER adaptively predicts obstacle motion based on observed velocities and uncertainties, and enable robust, real-time trajectory planning on resource-constrained UAV platforms.

The main contributions of this paper can be summarized as follows:

- *Real-time Dynamic Obstacle Prediction:* We introduce a decentralized lightweight probabilistic obstacle filter to track moving obstacles and extrapolate their future trajectories. This relaxes the restrictive assumption of perfect future knowledge in optimization-based trajectory planners such as MADER and RMADER. Our filter dynamically adjusts the prediction horizon based on observed obstacle velocities and incorporates uncertainty into the planning process.
- *Robust Collision Avoidance in Dynamic Environments:* We integrate the probabilistic obstacle filter with MADER's trajectory optimizer to develop POF + MADER (or POF+RMADER if integrated with RMADER), which formulates the predicted obstacle states and their associated uncertainties as dynamic, time-varying constraints within the MADER optimization problem. POF-integrated methods transform the planners from being purely reactive to proactively generating collision-free trajectories that account for most probable future trajectories of the dynamic obstacles. To comprehensively evaluate their robustness, these planners are tested against a representative suite of four motion models: straight-line for predictable paths, sinusoidal for oscillatory maneuvers, vertical wave for altitude variations, and trefoil-knot for complex, looped evasions. These canonical motion models serve as parametric approximations of challenging real-world behaviors, such as the flight patterns of birds or other non-cooperative drones. Across these scenarios, POF-integrated methods demonstrated significantly lower collision rates compared to the baselines.
- *Experimental Validation and Benchmarking:* We provide a comprehensive benchmark of our hybrid architecture in simulation, against the MADER, RMADER, and the state-of-the-art EGO-Swarm [5] baselines. Our analysis also considers sensor noise ($\sigma = 0.2 - 0.5m$) to quantify the system's resilience to real-world perception uncertainty. Furthermore, we demonstrate the practical viability of our lightweight approach through successful hardware implementation on resource-constrained Crazyflie 2.1 UAVs, which shows its applicability beyond simulation.

The remainder of the paper is organized as follows: Section II provides a literature review on multiagent trajectory planning considering obstacle avoidance. Section III provides problem formulation, including the assumptions and constraints associated with dynamic obstacles trajectory prediction. Section IV describes our proposed method, focusing on the integration of our probabilistic filter with optimization-based planners to create a proactive trajectory planner. Section V presents the evaluation and results of our experiments, including a comparison with the baselines. Section VI discusses the performance, robustness, and limitations of the proposed method. Lastly, Section VII discusses the conclusion and future work.

II. LITERATURE REVIEW

Trajectory planning in dynamic environments remains a critical challenge for unmanned aerial vehicles (UAVs) [12], particularly when navigating alongside non-cooperative dynamic obstacles with uncertain motion [13]. One way

to avoid mid-air collisions with dynamic obstacles is to predict their short-term future trajectories [4]. Recent studies have presented several prediction techniques, including model-based [14], [15], [16], learning-based [17], [18], biomimetic [19], [20], and polynomial fitting [21], [22] methods. Model-based methods such as Kalman Filters (KF) estimate future states from past observations and provide fast, low-latency predictions [14], [23]. Learning-based methods, such as reinforcement learning, employ neural networks to capture complex and non-linear obstacle motions, but they require substantial training data and computational resources [21], [24]. Biomimetic methods replicate biological vision, offering quick responses using minimal visual information but usually require significant parameter tuning and lack predictive depth [25]. Polynomial fitting method utilizes polynomial calculations for prediction but is constrained to paths featuring regular and smooth curves [12]. Most of these approaches relied heavily on deterministic assumptions about obstacle trajectories, such as constant-velocity [26], linear acceleration, or an a priori knowledge of obstacles' future trajectories [8], [9], which limit their real-world applicability. For instance, the MADER [8] algorithm provides collision-free multi-agent planning using polyhedral obstacle representations, but assumes perfect a priori knowledge of the dynamic obstacle future trajectory. Non-cooperative obstacles, such as human-operated vehicles, pedestrians, or even birds, do not provide trajectory information to UAVs. Even cooperative obstacles may experience communication delays or failures that make their intended trajectories temporarily unavailable. Following this, RMADER [9], which is an extension of the original MADER algorithm, incorporates delay compensation mechanisms to achieve improved collision avoidance performance even when communication with other UAVs and agents is intermittent. The algorithm maintains computational efficiency while providing theoretical guarantees for collision avoidance under realistic communication constraints. However, it still assumes eventual access to obstacle future trajectories information.

Collision avoidance in multiagent systems often differs whether the method can avoid collision with static obstacles [27], [28], dynamic obstacles [18], [29], or among the agents only [30], [31]. Charbel et al. [32] have proposed a decentralized Model Predictive Control-based trajectory planner that has better performance than MADER in terms of static obstacle avoidance and computation time, however, this approach does not explicitly incorporate dynamic obstacles as part of the collision avoidance scheme. Ego-Swarm [5] is another approach, an online decentralized and asynchronous trajectory planner implemented on both simulation and hardware, and includes capabilities for dynamic obstacle avoidance. In this paper, we have considered it as a baseline for comparison. The Chance Constrained Rapidly-exploring Random Trees (CC-RRT) algorithm [33] is another method that avoids dynamic obstacles by conservatively constraining the likelihood of collisions, accounting for Gaussian noise inherent in both linear system dynamics and the translational predictions of dynamic obstacles, and then plans obstacle-free trajectories.

Trajectory planner for dynamic environments usually relies on sensing and detection technologies [34] that could provide real-time information about dynamic obstacles. Modern UAVs employ a variety of sensing modalities, each with distinct advantages and limitations for dynamic obstacle detection and tracking. Approaches can be broadly divided into vision-based detection [35] and map-based tracking [14]. Vision-based methods use cameras (RGB [35] or depth [36]) or other optical sensors to directly detect obstacles in the UAV's field of view. Some lightweight techniques [14], [36] extract geometric cues (e.g., disparity from stereo or depth cameras) to locate obstacles and track them in real-time. For example, Liu et al. [37] has shown that lightweight object detection algorithms, can operate on UAV platforms to identify and track dynamic obstacles in real-time. Their approach combines visual object detection with Kalman filtering to estimate obstacle states and predict short-term trajectories. The limitation of purely vision methods is that they often cannot distinguish moving obstacles from static ones without temporal consistency checks [38]. Learning-based vision approaches (for example, neural network detectors) [39] are also used to classify and track obstacles, but these tend to be computationally heavy for small-scale UAV onboard use. The second category is map-based approaches [40], [41], [42], which integrate sensor data over time into a world model. For example, a voxel grid map where each cell is marked as occupied (obstacle) or free; by updating this map at high frequency, one can infer which cells have changed and thus detect moving objects [43]. Such methods separate static and dynamic elements, but a drawback is that a voxel map alone does not provide an easy way to predict future motion of dynamic obstacles [14].

Even with good tracking, the future trajectories of dynamic obstacles remain uncertain [4]. The planner must therefore account for uncertainty to ensure robust collision avoidance. Amutha et al. [44] have presented a probabilistic framework that explicitly models uncertainty in obstacle motion and UAV dynamics to generate trajectories with probabilistic safety guarantees. Chance-constrained planning formalizes this idea by treating obstacle states as random variables and enforcing that the probability of collision remains below a specified threshold [45], [46]. Another approach is to use online learning [47] or adaptive modeling [48] to tighten the uncertainty bounds. Instead of assuming worst-case behavior (which can be overly conservative), the planner can learn typical obstacle behavior on the fly. For example, Zhou et al. [47] estimate intended control inputs of dynamic obstacles and compute less-conservative forward reachable sets, enabling a robust MPC to keep the ego trajectory outside those sets over the horizon.

The proposed POF+MADER approach addresses the limitations of the existing methods by providing a lightweight, theoretically grounded solution that combines the computational efficiency of Kalman filter-based prediction with the proven collision avoidance capabilities of established optimization methods (e.g., MADER, RMADER). Unlike existing approaches that require perfect a priori knowledge or extensive computational resources, our method enables practical deployment on standard UAV platforms while maintaining formal safety under realistic operational conditions with uncertain dynamic obstacles and limited computational resources.

III. PROBLEM FORMULATION

Let $\mathcal{I} = \{1, \ldots, N\}$ denote the set of N agents, with $\mathbf{p}_i(t) \in \mathbb{R}^3$ representing the position of agent $i \in \mathcal{I}$ at time t. Each agent is modeled as a rigid sphere $S_i \subset \mathbb{R}^3$ of radius $r_i > 0$, enclosing all possible orientations such that the collision geometry remains invariant to rotation. The perception range of agent i is defined as a sphere $S_{pi} \subset \mathbb{R}^3$ of radius $r_{pi} > 0$, centered at $\mathbf{p}_i(t)$, within which the agent can detect obstacles using onboard sensors.

A. DYNAMIC OBSTACLES AND PARTIAL OBSERVABILITY

Let $\mathcal{M} = \{1, \ldots, M\}$ denote the set of M dynamic obstacles, with state vector $\mathbf{x}_m(t) = [\mathbf{q}_m(t)^\top, \dot{\mathbf{q}}_m(t)^\top]^\top \in \mathbb{R}^6$ for obstacle $m \in \mathcal{M}$, where $\mathbf{q}_m(t) \in \mathbb{R}^3$ is the true position and $\dot{\mathbf{q}}_m(t) \in \mathbb{R}^3$ is the velocity at time t. Obstacle positions are not globally known; instead, when an obstacle is within agent i's perception range, the agent receives a noisy measurement of its position $\mathbf{z}_m(t)$, i.e., when

$$\|\mathbf{p}_i(t) - \mathbf{q}_m(t)\| \leq r_{pi} \quad \forall i \in \mathcal{I},\ \forall m \in \mathcal{M},\ \forall t \in [t_{\text{in}}, t_{\text{f}}],$$

where $[t_{\text{in}}, t_{\text{f}}]$ is the planning horizon, and

$$\mathbf{z}_m(t) = \mathbf{q}_m(t) + \mathbf{v}_m(t),$$

where $\mathbf{v}_m(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ is zero-mean Gaussian sensor noise with covariance $\mathbf{R}$.

B. PROBABILISTIC FORECASTING AND TRAJECTORY PLANNING

Given this partial and uncertain information, the overall problem is to: (1) predict the obstacle's future trajectory and (2) plan a safe and efficient path for the agent.

1) FORECASTING OBJECTIVE

Upon detection, each agent i employs a probabilistic filter to estimate the obstacle's current state $\hat{\mathbf{x}}_m(t|t)$ and predict the future trajectory $\hat{\pi}_m(t|t) = \hat{\mathbf{q}}_m(t+k|t)_{k=1}^{L}$ over a horizon L. Let $\pi_m(t)$ represent the actual unobserved trajectory of obstacle m over $[t_{\text{in}}, t_{\text{f}}]$. The primary objective of the filter is to minimize the aggregate prediction error across all obstacles:

$$\min_{\{\hat{\pi}_m\}_{m=1}^{M}} \sum_{m=1}^{M} \int_{t_{\text{in}}}^{t_{\text{f}}} \|\pi_m(t) - \hat{\pi}_m(t)\|^2 \, dt \tag{1}$$

2) PLANNING OBJECTIVE

The primary objective for each agent i is to compute a smooth, dynamically feasible trajectory $\tau_i(t)$ that reaches a goal while ensuring safety. This is typically formulated as a constrained optimization problem that minimizes control effort (e.g., integrated squared jerk):

$$\min_{\tau_i} \int_{t_{\text{in}}}^{t_{\text{f}}} \|\mathbf{j}_i(t)\|^2 \, dt$$

This optimization is subject to kinematic constraints (e.g., maximum velocity and acceleration) and the following collision avoidance constraints based on $\hat{\pi}_m(t)$.

C. SAFETY CONSTRAINTS

To ensure safe navigation, the agent's planned trajectory $\tau_i(t)$ must remain separate from the predicted paths of other agents and obstacles.

1) INTER-AGENT COLLISION AVOIDANCE

For any two distinct agents, their planned trajectories must be disjoint by a safety margin d_{agent}:

$$\bigcup_{i \in \mathcal{I}} \bigcup_{\substack{j \in \mathcal{I} \\ j \neq i}} \left(\tau_i(t) \oplus d_{\text{agent}}\right) \cap \left(\tau_j(t) \oplus d_{\text{agent}}\right) = \emptyset \quad \forall t \in [t_{\text{in}}, t_{\text{f}}] \tag{2}$$

where $d_{\text{agent}} > 0$ is the minimum inter-agent separation (typically $2r_i + \epsilon$).

2) AGENT-OBSTACLE COLLISION AVOIDANCE

The agent must avoid the predicted occupancy of each dynamic obstacle. In other words, the agent's trajectory $\tau_i(t)$ must not intersect with the predicted obstacle trajectory $\hat{\pi}_m(t)$, inflated by a safety margin d_{safe}:

$$\bigcup_{i \in \mathcal{I}} \bigcup_{m \in \mathcal{M}} (\tau_i(t) \oplus d_{\text{safe}}) \cap \hat{\pi}_m(t) = \emptyset \quad \forall t \in [t_{\text{in}}, t_{\text{f}}] \tag{3}$$

where $d_{\text{safe}} > 0$ is the minimum safe separation (sum of agent radius r_i, obstacle radius r_m, and margin $\epsilon > 0$ for uncertainty), and $\oplus d_{\text{safe}}$ denotes the Minkowski sum inflating the trajectory by a sphere of radius d_{safe}.

IV. METHODOLOGY AND INTEGRATION

A. DYNAMIC OBSTACLE DEFINITION AND MODEL

In our framework, dynamic obstacles are defined as non-cooperative and non-interactive moving entities that may intersect with the agent's trajectory, and may pose collision risks. These obstacles do not share intent or trajectory information with the agents and do not alter their motion in response to the agents' behaviors. To evaluate the robustness of our approach across diverse real-world like scenarios, we model the obstacles' motions using four distinct parametric trajectories, each representing varying levels of predictability and complexity. These models are implemented in both simulation and hardware experiments, with obstacle

velocities scaled to challenge the prediction and avoidance capabilities.

The position of obstacle m at time t, denoted by $\mathbf{q}_m(t) \in \mathbb{R}^3$, is governed by one of the following equations, selected per experiment:

- ***Straight line***: This model represents linear, highly predictable trajectories common in open-field navigation or uniform motion. The position evolves as:

$$\mathbf{q}_m(t) = \mathbf{q}_m(t-1) + \dot{\mathbf{q}}_m(t-1) \cdot \Delta t \tag{4}$$

  where $\dot{\mathbf{q}}_m(t-1) \in \mathbb{R}^3$ is the velocity at the previous timestep, and $\Delta t > 0$ is the discrete time step.
- ***Sinusoid wave***: This model captures oscillatory dynamics, mimicking evasive maneuvers or environmental perturbations (e.g., wind effects). The position at time t is defined by:

$$\mathbf{q}_m(t) = \begin{bmatrix} Bt \\ C \\ A\sin(\omega t + \phi) \end{bmatrix} \tag{5}$$

  where $A > 0$ is the amplitude, $\omega > 0$ is the angular frequency, ϕ is the phase shift, B is a constant velocity component along the x-axis, and C is a constant offset along the y-axis.
- ***Trefoil-knot***: Inspired by parametric curves for looped, non-linear trajectories, simulating complex, repeating patterns. The position of the obstacle at time t is given by:

$$\mathbf{q}_m(t) = \begin{bmatrix} \sin(t) + 2\sin(2t) \\ \cos(t) - 2\cos(2t) \\ -\sin(3t) \end{bmatrix} \tag{6}$$

- ***Vertical wave***: This models vertical oscillations, relevant for altitude-varying obstacles like ascending/descending UAVs. The position of the obstacle at time t is given by:

$$\mathbf{q}_m(t) = \begin{bmatrix} x_0 \\ y_0 \\ A\sin(\omega t + \phi) \end{bmatrix} \tag{7}$$

  where x_0 and y_0 are constant positions in the x and y directions, respectively, and A, ω, and ϕ are the amplitude, angular frequency, and phase shift of the sinusoidal motion in the z-direction.

B. PROBABILISTIC OBSTACLE FILTERING

We employ a decentralized Kalman Filter (KF) for real-time trajectory prediction of dynamic obstacles. Each agent runs an independent KF for every obstacle that enters its perception sphere S_{pi} of radius r_{pi}. This agent-centric approach scales with the number of agents and obstacles, as each agent independently manage predictions for obstacles in its local vicinity.

Algorithm 1 describes the implementation of the probabilistic obstacle filter, which takes the agent position $\mathbf{p}_i(t)$, obstacle position measurement $\mathbf{q}_m(t)$, the perception radius of the agent r_{pi}, and a minimum prediction length L_{min}. Once

Algorithm 1 Probabilistic Obstacle Filter

Data: $\bigcup_{i=1}^{N} \mathbf{p}_i(t)$, $\bigcup_{m=1}^{M} \mathbf{q}_m(t)$, perception range r_{pi}, minimum prediction length $L_{\min}$
Result: Predicted trajectories $\hat{\pi}_m(t)$ with covariances for all detected obstacles m

begin
  foreach *agent* $i \in \{1, \ldots, N\}$ **do**
    $D_i(t) \leftarrow \emptyset$;
    foreach *obstacle* $m \in \{1, \ldots, M\}$ **do**
      if $\|\mathbf{p}_i(t) - \mathbf{q}_m(t)\| \le r_{pi}$ **then**
        $D_i(t) \leftarrow D_i(t) \cup \{m\}$;
        Compute $\dot{\mathbf{q}}_m(t)$ from $\mathbf{q}_m(t)$;
        Initialize KF with:
        $\hat{\mathbf{x}}_m(t_{\text{in}}|t_{\text{in}}) = [\mathbf{q}_m(t)^\top, \dot{\mathbf{q}}_m(t)^\top]^\top$,
        $\mathbf{P}_m(t_{\text{in}}|t_{\text{in}}) = \mathbf{P}_0$;
      end
    end
    while $|D_i(t)| > 0$ **do**
      foreach *obstacle* $m \in D_i(t)$ **do**
        Perform KF prediction step:
        $\hat{\mathbf{x}}_m(t|t-1) = \mathbf{A}\hat{\mathbf{x}}_m(t-1|t-1)$,
        $\mathbf{P}_m(t|t-1) = \mathbf{A}\mathbf{P}_m(t-1|t-1)\mathbf{A}^\top + \mathbf{Q}$;
        Update with measurement $\mathbf{z}_m(t) = \mathbf{q}_m(t)$:
        $\mathbf{y}_m(t) = \mathbf{z}_m(t) - \mathbf{H}\hat{\mathbf{x}}_m(t|t-1)$,
        $\mathbf{S}_m(t) = \mathbf{H}\mathbf{P}_m(t|t-1)\mathbf{H}^\top + \mathbf{R}$,
        $\mathbf{K}_m(t) = \mathbf{P}_m(t|t-1)\mathbf{H}^\top\mathbf{S}_m(t)^{-1}$,
        $\hat{\mathbf{x}}_m(t|t) = \hat{\mathbf{x}}_m(t|t-1) + \mathbf{K}_m(t)\mathbf{y}_m(t)$,
        $\mathbf{P}_m(t|t) = (\mathbf{I}_6 - \mathbf{K}_m(t)\mathbf{H})\mathbf{P}_m(t|t-1)$;
        Compute adaptive horizon:
        $L \leftarrow \max\left(L_{\min}, \left\lceil \frac{\|\dot{\mathbf{q}}_m(t|t)\| \cdot T_{\text{plan}}}{d_{\text{step}}} \right\rceil\right)$,
        for $k = 1$ **to** L **do**
          Perform open-loop prediction:
          $\hat{\mathbf{x}}_m(t+k|t) = \mathbf{A}\hat{\mathbf{x}}_m(t+k-1|t)$,
          $\mathbf{P}_m(t+k|t) = \mathbf{A}\mathbf{P}_m(t+k-1|t)\mathbf{A}^\top + \mathbf{Q}$,
          $\hat{\pi}_m(t) \leftarrow \hat{\pi}_m(t) \cup \{\hat{\mathbf{q}}_m(t+k|t)\}$;
        end
      end
      foreach *obstacle* $m \in D_i(t)$ **do**
        if $\|\mathbf{p}_i(t) - \mathbf{q}_m(t)\| > r_{pi}$ **then**
          $D_i(t) \leftarrow D_i(t) \setminus \{m\}$;
          Terminate KF for obstacle m;
        end
      end
      Increment t to next timestep;
    end
  end
end

an obstacle m is detected within this predefined perception radius r_{pi} (i.e., $\|\mathbf{p}_i(t) - \mathbf{q}_m(t)\| \leq r_{pi}$), the tracking system initializes a dedicated KF specifically tailored to the detected obstacle's trajectory prediction (i.e., $\hat{\pi}_m(t) = \{\hat{\mathbf{q}}_m(t+1|t), \ldots, \hat{\mathbf{q}}_m(t+L|t)\}$).

For each detected obstacle m at time t, POF maintains a state vector $\mathbf{x}_m(t) = [\mathbf{q}_m(t)^\top, \dot{\mathbf{q}}_m(t)^\top]^\top \in \mathbb{R}^6$, comprising the 3D position $\mathbf{q}_m(t) \in \mathbb{R}^3$ and velocity $\dot{\mathbf{q}}_m(t) \in \mathbb{R}^3$. We adopt a constant-velocity motion model, which is well-suited for short-term predictions in the absence of intent information from non-cooperative obstacles, and is commonly used in tracking applications. The discrete-time dynamics are given by:

$$\mathbf{x}_m(t+1) = \mathbf{A}\mathbf{x}_m(t) + \mathbf{w}_m(t) \tag{8}$$

where $\mathbf{w}_m(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$ is zero-mean Gaussian process noise with covariance matrix $\mathbf{Q} \in \mathbb{R}^{6\times6}$ that captures the unmodeled accelerations or environmental disturbances. The state transition matrix $\mathbf{A}$ is:

$$\mathbf{A} = \begin{bmatrix} \mathbf{I}_3 & \Delta t \cdot \mathbf{I}_3 \\ \mathbf{0}_{3\times3} & \mathbf{I}_3 \end{bmatrix}$$

where Δt represents the sampling interval, $\mathbf{I}_3$ the 3×3 identity matrix, and $\mathbf{0}_{3\times3}$ the zero matrix. In this formulation, we omit the control input term ($\mathbf{B}\mathbf{u}_m(t)$), as non-cooperative obstacles provide no such information; however, this can be extended for scenarios with partial observability of intent.

Sensor measurements yield noisy position observations $\mathbf{z}_m(t) = \mathbf{q}_m(t) + \mathbf{v}_m(t)$, where $\mathbf{v}_m(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ and $\mathbf{R} \in \mathbb{R}^{3\times3}$ is the measurement covariance which is tuned based on sensor specifications. The linear observation model is:

$$\mathbf{z}_m(t) = \mathbf{H}\mathbf{x}_m(t) + \mathbf{v}_m(t), \quad \mathbf{H} = [\mathbf{I}_3, \mathbf{0}_{3\times3}] \tag{9}$$

The KF recursively computes the minimum mean-squared-error estimate of $\mathbf{x}_m(t)$ and alternates between time prediction and measurement update steps. Starting from an initial state estimate $\hat{\mathbf{x}}_m(0|0)$ (e.g., from the first detection) and covariance $\mathbf{P}_m(0|0)$ (initialized with large diagonals to reflect high initial uncertainty), the prediction step propagates the prior:

$$\hat{\mathbf{x}}_m(t|t-1) = \mathbf{A}\hat{\mathbf{x}}_m(t-1|t-1), \tag{10}$$

$$\mathbf{P}_m(t|t-1) = \mathbf{A}\mathbf{P}_m(t-1|t-1)\mathbf{A}^\top + \mathbf{Q} \tag{11}$$

Upon receiving a measurement $\mathbf{z}_m(t)$, the update incorporates the innovation to yield the posterior:

$$\mathbf{y}_m(t) = \mathbf{z}_m(t) - \mathbf{H}\hat{\mathbf{x}}_m(t|t-1), \tag{12}$$

$$\mathbf{S}_m(t) = \mathbf{H}\mathbf{P}_m(t|t-1)\mathbf{H}^\top + \mathbf{R}, \tag{13}$$

$$\mathbf{K}_m(t) = \mathbf{P}_m(t|t-1)\mathbf{H}^\top\mathbf{S}_m(t)^{-1}, \tag{14}$$

$$\hat{\mathbf{x}}_m(t|t) = \hat{\mathbf{x}}_m(t|t-1) + \mathbf{K}_m(t)\mathbf{y}_m(t), \tag{15}$$

$$\mathbf{P}_m(t|t) = (\mathbf{I}_6 - \mathbf{K}_m(t)\mathbf{H})\mathbf{P}_m(t|t-1) \tag{16}$$

where $\mathbf{K}_m(t)$ is the Kalman gain, optimally blending prediction and measurement based on their relative uncertainties, and $\mathbf{I}_6$ is the 6×6 identity matrix. This formulation converges to accurate estimates as observations accumulate, with the posterior covariance $\mathbf{P}_m(t|t)$ providing a measure of estimation confidence.

For trajectory forecasting, POF extends the filtered state forward over a prediction horizon L in the open-loop mode without assuming future measurements. The multi-step predictions are computed recursively:

$$\hat{\mathbf{x}}_m(t+k|t) = \mathbf{A}\hat{\mathbf{x}}_m(t+k-1|t), \quad k = 1, \ldots, L \tag{17}$$

Initialized from the current posterior $\hat{\mathbf{x}}_m(t|t)$. The corresponding covariances propagate as:

$$\mathbf{P}_m(t+k|t) = \mathbf{A}\mathbf{P}_m(t+k-1|t)\mathbf{A}^\top + \mathbf{Q} \tag{18}$$

To optimize prediction relevance, we adaptively adjust L based on obstacle velocity magnitude $\|\dot{\mathbf{q}}_m(t)\|$:

$$L = \max\left(L_{\min}, \left\lceil \frac{\|\dot{\mathbf{q}}_m(t)\| \cdot T_{\text{plan}}}{d_{\text{step}}} \right\rceil\right) \tag{19}$$

where $L_{\min}$ is a minimum horizon, T_{plan} is the agent's planning timeframe, and d_{step} a spatial discretization step used for collision checks. This ensures longer predictions for faster obstacles and covers potential collision windows while minimizing computational overhead.

To manage multiple dynamic obstacles, each agent i dynamically maintains a set, $D_i(t)$, containing all obstacles currently within its perception sphere. An obstacle m is considered detected and added to this set if its distance to the agent is less than or equal to the perception radius r_{pi}. This condition can be expressed as a binary indicator:

$$y_{i,m}(t) = \begin{cases} 1, & \text{if } \|\mathbf{p}_i(t) - \mathbf{q}_m(t)\| \leq r_{pi} \\ 0, & \text{otherwise} \end{cases} \tag{20}$$

The set of detected obstacles for agent i at time t is therefore formally defined as:

$$D_i(t) = \{m \in M \mid y_{i,m}(t) = 1\} \tag{21}$$

An independent Kalman Filter is instantiated for each obstacle upon entering $D_i(t)$ and is terminated once it leaves. As long as an obstacle $m \in D_i(t)$, its corresponding KF is updated with new measurements at each timestep, and its trajectory is reforecasted according to (17); when $m \notin D_i(t)$ the filter is terminated and removed from memory.

The output of the probabilistic filter is the predicted trajectory, $\hat{\pi}_m(t)$, which is a time-ordered sequence of the obstacle's forecasted future positions over the adaptive horizon L. This sequence is formally defined as:

$\hat{\pi}_m(t) = \{\hat{\mathbf{q}}_m(t|t), \hat{\mathbf{q}}_m(t+1|t), \ldots, \hat{\mathbf{q}}_m(t+L|t)\}$

Each position vector $\hat{\mathbf{q}}_m(t+k|t)$ in this sequence is extracted from the corresponding forecasted 6D state vector $\hat{\mathbf{x}}_m(t+k|t)$. This set of predicted positions is then passed to the optimization-based planner to be incorporated as dynamic, time-varying obstacle-avoidance constraints.

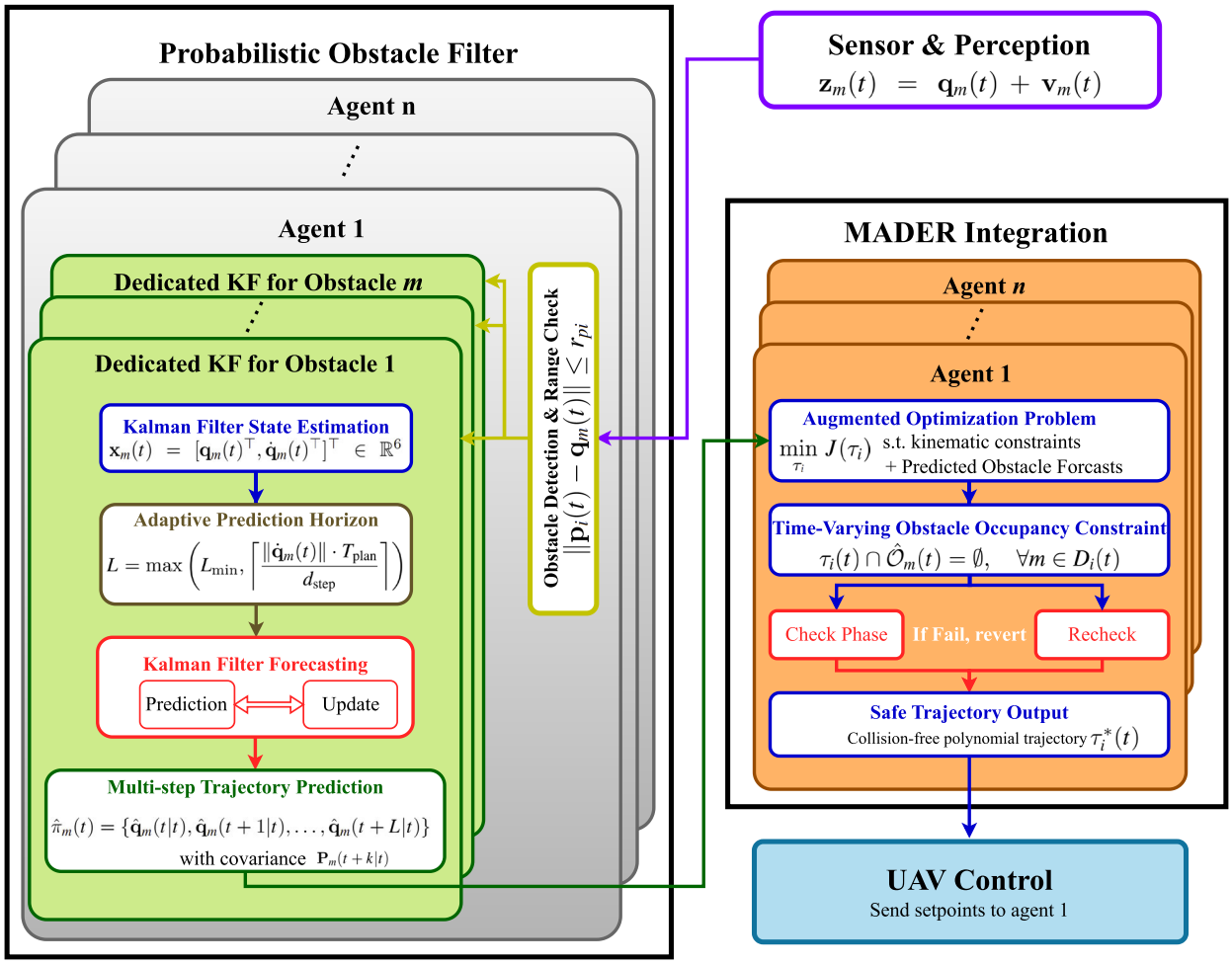


FIGURE 2. The POF+MADER system architecture; integrating decentralized probabilistic obstacle filter with optimization-based planners such as MADER [8]. The system begins by collecting the real-time current positions of both agents and obstacles, and then checks if obstacles are being detected by agents. The POF module uses a decentralized Kalman Filter bank to estimate and forecast the trajectory ($\hat{\pi}_m(t)$) and uncertainty ($P_m(t)$) for each obstacle. These predictions are formulated as dynamic constraints within the MADER optimization problem, which generates a safe, optimal trajectory (τ_i^*) for execution by the UAV's controller, closing the real-time loop.

C. POF+MADER INTEGRATION

The integration of the Probabilistic Obstacle Filter (POF) with state-of-the-art optimization-based trajectory planners, such as MADER [8] or RMADER [9], forms the core of our proposed framework, denoted as POF+MADER (or POF+RMADER when extended to the robust variant). This architecture addresses the inherent limitation of assuming perfect a priori knowledge of dynamic obstacle trajectories by substituting real-time probabilistic predictions from POF into the planning pipeline. As illustrated in Fig. 2, the system comprises two interconnected modules: (1) decentralized detection and trajectory forecasting of obstacles entering an agent's perception sphere, executed via Algorithm 1; and (2) augmentation of the trajectory optimization process to incorporate these forecasts as dynamic constraints, detailed in Algorithm 2 and explained in this subsection. This coupling enables agents to generate feasible, collision-free polynomial trajectories that adapt proactively to uncertain obstacle motions, and eventually enhance safety without sacrificing computational tractability.

At the core of POF+MADER lies an extension of MADER's non-linear optimization formulation, which generates smooth, dynamically feasible trajectories represented as piecewise polynomials in the MINVO basis [49]. For agent i, the original MADER problem minimizes a cost function that includes a control effort term (proportional to squared jerk for cubic splines), weighted penalties on deviations from the goal, and adherence to kinematic limits. Specifically, for a clamped uniform B-Spline of degree $p = 3$ (cubic) with $n+1$ control points $\{\mathbf{q}_0, \ldots, \mathbf{q}_n\}^{\text{BS}}$ and knots defining

intervals $j \in J$, the control effort is approximated as:

$$\int_{t_{\text{in}}}^{t_{\text{f}}} \|\mathbf{j}(t)\|^2 \, dt \;\propto \sum_{j \in J} \left\| \mathbf{Q}_j^{\text{BS}} \mathbf{A}_{\text{pos}}^{\text{BS}}(j) \begin{bmatrix} 6 \\ 0 \\ 0 \\ 0 \end{bmatrix} \right\|_2^2$$

where $\mathbf{Q}_j^{\text{BS}}$ are the B-Spline control points for interval j, and $\mathbf{A}_{\text{pos}}^{\text{BS}}(j)$ is the interval-dependent evaluation matrix. The full objective also includes a soft penalty on the final position: $\omega\|\mathbf{q}_{n-2} - \mathbf{g}\|_2^2$, with $\omega \geq 0$, as $\mathbf{q}_{n-2} = \mathbf{q}_{n-1} = \mathbf{q}_n$ under the final stop condition ($\mathbf{v}(t_f) = \mathbf{0}$, $\mathbf{a}(t_f) = \mathbf{0}$). Constraints encompass initial conditions ($\mathbf{x}(t_{\text{in}}) = \mathbf{x}_{\text{in}}$, fixing $\mathbf{q}_0, \mathbf{q}_1, \mathbf{q}_2$), bounds on velocity and acceleration using MINVO-transformed points ($|\mathbf{v}| \leq \mathbf{v}_{\max}$ for $\mathbf{v} \in \mathcal{V}_j^{\text{MV}}$, $|\mathbf{a}_l| \leq \mathbf{a}_{\max}$ for $l \in L \setminus \{n-1, n\}$), confinement to a planning sphere $\mathcal{S}$ of radius r centered at $\mathbf{d}$ ($\|\mathbf{q} - \mathbf{d}\|_2^2 \leq r^2$ for $\mathbf{q} \in \mathcal{Q}_j^{\text{MV}}$), and collision avoidance via separating planes $\boldsymbol{\pi}_{ij}$ (defined by normal $\mathbf{n}_{ij}$ and offset d_{ij}) between the agent's MINVO hulls $\mathcal{Q}_j^{\text{MV}}$ and other agents' committed polyhedra $\mathcal{C}_{ij}$:

$$\mathbf{n}_{ij}^T \mathbf{c} + d_{ij} > 0 \quad \forall \mathbf{c} \in \mathcal{C}_{ij}, \forall i \in I, j \in J \tag{22}$$

$$\mathbf{n}_{ij}^T \mathbf{q} + d_{ij} < 0 \quad \forall \mathbf{q} \in \mathcal{Q}_j^{\text{MV}}, \forall j \in J \tag{23}$$

The total planning time is heuristically set as $t_f - t_{\text{in}} = \|\mathbf{g} - \mathbf{d}\|_2 / v_{\max}$.

In POF+MADER, we augment this formulation by introducing time-varying constraints derived from the predicted obstacle trajectories $\hat{\pi}_m(t)$ output by POF. For each detected obstacle $m \in D_i(t)$, the predicted positions $\hat{\mathbf{q}}_m(t+k|t)$ for $k = 1, \ldots, L$ are inflated by a safety margin ϵ (accounting for UAV size, obstacle bounding box, and prediction uncertainty via covariance ellipsoids from $\mathbf{P}_m(t+k|t)$) to form restricted regions. These regions are approximated as convex polyhedra over discrete time intervals to ensure tractable inclusion in the optimization. The augmented problem becomes:

$$\min_{\tau_i} J(\tau_i)$$

$$\text{s.t.} \begin{cases} \text{kinematic bounds and initial/final conditions,} \\ \tau_i(t) \cap \mathcal{O}_{\text{static}} = \emptyset, \quad \forall t \in [t_0, t_f], \\ \tau_i(t) \cap \tau_j(t) = \emptyset, \quad \forall j \neq i, \forall t \in [t_0, t_f], \\ \tau_i(t) \cap \hat{\mathcal{O}}_m(t) = \emptyset, \quad \forall m \in D_i(t), \forall t \in [t_0, t_f] \end{cases}$$

where $\mathcal{O}_{\text{static}}$ denotes static obstacle sets, $\pi_j(t)$ committed trajectories from other agents (broadcast in MADER's decentralized protocol), and $\hat{\mathcal{O}}_m(t)$ the inflated predicted occupancy of obstacle m, computed as the convex hull of $\hat{\mathbf{q}}_m(t) \oplus B(\epsilon + \sqrt{\lambda_{\max}(\mathbf{P}_m(t))})$, with $B(r)$ a ball of radius r and $\lambda_{\max}$ the maximum eigenvalue of the position submatrix of $\mathbf{P}_m(t)$ for uncertainty-aware buffering. This constraint enforces a minimum separation $d_{\text{safe}} = r_i + r_m + \epsilon$, where r_i and r_m are agent and obstacle radii. To ensure feasibility and robustness, POF+MADER employs a collision check-recheck mechanism, extending MADER's two-phase validation. In the check phase, the optimized trajectory τ_i^* is discretized at fine temporal resolution (e.g., $\Delta t_c = 0.01$ s) and tested for intersections with $\hat{\mathcal{O}}_m(t)$ using efficient separating hyperplane methods for polyhedral sets. Mathematically, for each time t_k and obstacle m, intersection is detected if no hyperplane separates the agent's inflated position $\mathbf{p}_i(t_k) \oplus B(r_i)$ from $\hat{\mathcal{O}}_m(t_k)$. If a potential collision is flagged, the recheck phase refines the test with higher fidelity (e.g., continuous-time overlap via root-finding on distance functions), confirming violations only if the minimum distance falls below d_{safe}:

$$\min_t d(\mathbf{p}_i(t), \hat{\mathbf{q}}_m(t)) < d_{\text{safe}}, \quad t \in [t_k - \delta, t_k + \delta]$$

where $d(\cdot, \cdot)$ is the Euclidean distance, and δ a local interval. This dual-phase approach mitigates false positives from discretization artifacts while maintaining real-time performance. For initial guesses, POF+MADER leverages an adapted version of MADER's Octopus Search algorithm (a variant of A* tailored for B-Splines and dynamic constraints). This search generates candidate control points by sampling velocities satisfying $v_{\max}$ and $a_{\max}$, discarding nodes that violate linear separability with predicted obstacle hulls, sphere bounds, or voxel proximity checks. The resulting feasible path, closest to the goal if exact reachability fails, seeds the non-linear solver, ensuring rapid convergence. Algorithm 2 encapsulates this process: predicted trajectories from POF serve as inputs, and the loop iteratively adjusts $\tau_i(t)$ (the committed segment of τ_i^*) upon collision detection. Unlike the original MADER, which reacts to known obstacle states, POF+MADER anticipates conflicts by leveraging forward-propagated estimates, enabling preemptive rerouting. For instance, if a fast-moving obstacle's prediction indicates an impending intersection, the optimizer biases the solution toward evasive maneuvers, such as altitude changes or lateral deviations, while preserving trajectory continuity through soft penalties on deviations from prior commitments. In the RMADER variant, this integration further benefits from delay-aware broadcasting to ensure predictions remain synchronized across agents despite communication latencies.

D. LOCALIZATION UNCERTAINTY IN OBSTACLE PREDICTION

We conducted real-world experiments using a Crazyflie 2.1 drone within a motion capture system-equipped arena to characterize localization uncertainty in obstacle prediction. A predefined figure-eight trajectory was mathematically defined using a parametric function to represent the ground truth, and the drone was navigated along this path across five trials, as illustrated in Fig. 3(a). The deviation between the predefined positions and the real-time perceived positions, recorded by the motion capture system, was computed to quantify localization errors. These empirical deviations, depicted in Fig. 3(b-d), provide a realistic estimate of localization uncertainty encountered during UAV navigation in dynamic environments.

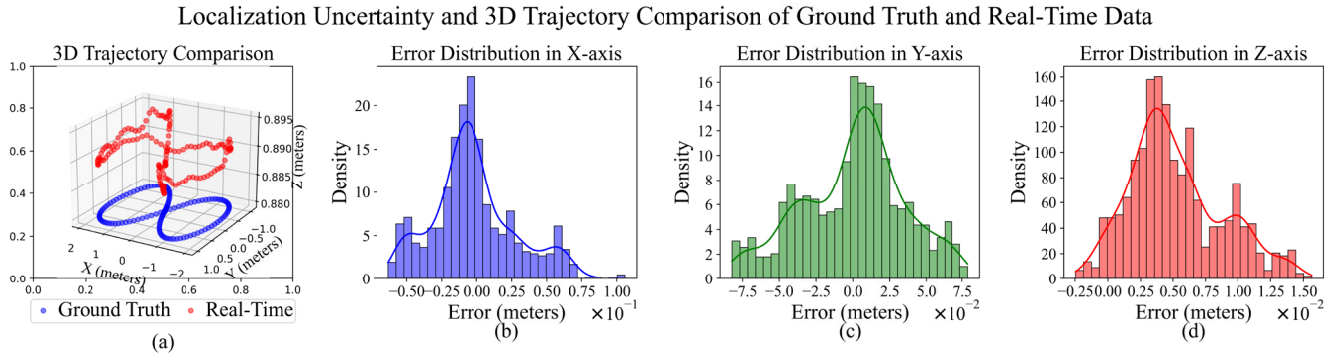


FIGURE 3. Visualization of localization uncertainty and 3D Crazyflie drone trajectory. (a) 3D trajectory plot comparing ground truth and real-time data for one trial, (b) X-axis localization error distribution, (c) Y-axis localization error distribution, and (d) Z-axis localization error distribution.

Algorithm 2 POF+MADER Proactive Trajectory Planner

Data: Current agent state $\mathbf{x}_i(t_{curr})$, goal $\mathbf{g}_i$
Result: Collision-free trajectory segment τ_i^*
begin
  // Get latest forecasts from the filter
  $\{\hat{\mathcal{O}}_m(t)\}_{m\in D_i} \leftarrow$ POF_Module.getPredictions();
  // Find a feasible initial trajectory seed
  $\tau_{seed} \leftarrow$ A-star-Search($\mathbf{x}_i(t_{curr})$, $\mathbf{g}_i$, $\{\hat{\mathcal{O}}_m(t)\}$);
  // Solve the augmented optimization problem
  $\tau_i^* \leftarrow$ NonlinearOptimize(τ_{seed}, $\{\hat{\mathcal{O}}_m(t)\}$);
  // Validate the final trajectory for safety
  if *isCollisionFree*(τ_i^*, $\{\hat{\mathcal{O}}_m(t)\}$) **then**
    return τ_i^*;
  end
  else
    // Use previous trajectory
    return τ_i^{prev};
  end
end

To generalize this uncertainty, we model the observed position of a dynamic obstacle as corrupted by Gaussian noise:

$$\mathbf{q}_m^{\text{noisy}}(t) = \mathbf{q}_m(t) + \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{24}$$

where:

- $\mathbf{q}_m^{\text{noisy}}(t) \in \mathbb{R}^3$ is the noisy observed position of obstacle m at time t,
- $\mathbf{q}_m(t) \in \mathbb{R}^3$ is the true position of the obstacle,
- $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ represents a multivariate Gaussian noise distribution with mean vector $\boldsymbol{\mu}$ and a covariance matrix $\boldsymbol{\Sigma}$.

Empirical analysis of localization uncertainty across the X, Y, and Z dimensions yielded the following statistical parameters, derived from the motion capture data:

- **X**-axis: $\mu_x = 0.0232$ m, $\sigma_x^2 = 0.0197$ m^2,
- **Y**-axis: $\mu_y = 0.0278$ m, $\sigma_y^2 = 0.0215$ m^2,
- **Z**-axis: $\mu_z = 0.0053$ m, $\sigma_z^2 = 0.0034$ m^2.

These values reflect the mean and variance of position errors in each dimension. These empirically-derived parameters are then used to inform our Probabilistic Obstacle Filter (POF). Specifically, the covariance matrix $\boldsymbol{\Sigma}$ from our model is used to define the measurement noise covariance matrix, $\mathbf{R}$, within the Kalman Filter. By feeding the noisy position data $\mathbf{q}_m^{\text{noisy}}(t)$ as the measurement input $\mathbf{z}_m(t)$ and configuring $\mathbf{R}$ with our real-world data, the POF can accurately estimate the true state $\mathbf{x}_m(t)$ and produce robust future trajectory predictions $\hat{\pi}_m(t) = \{\hat{\mathbf{q}}_m(t|t), \hat{\mathbf{q}}_m(t+1|t), \ldots, \hat{\mathbf{q}}_m(t+L|t)\}$.

V. EXPERIMENTATION AND RESULTS

This section presents the experimental evaluation of the proposed approaches in both simulation and real-world environments. The evaluation includes five algorithms in simulation: the baselines MADER [8], POF+MADER, Robust MADER (RMADER) [9], POF+RMADER, and Ego-Swarm [5]. The real-world evaluation, due to hardware and resource constraints, focuses on MADER and POF+MADER only. All experiments were implemented in the Robot Operating System (ROS) framework, with physical trials conducted on Crazyflie 2.1 drones.

A. EXPERIMENTAL SCENARIOS

In both simulation and real-world tests, the agent was tasked with navigating from a predefined start location to a goal location while avoiding dynamic obstacles. Four representative dynamic obstacle motion models were considered as shown in Fig. 4, and described below:

- **(a) UAV encountering dynamic obstacles moving in the opposite direction along a straight line:** In this experiment, the obstacles move in a straight line.

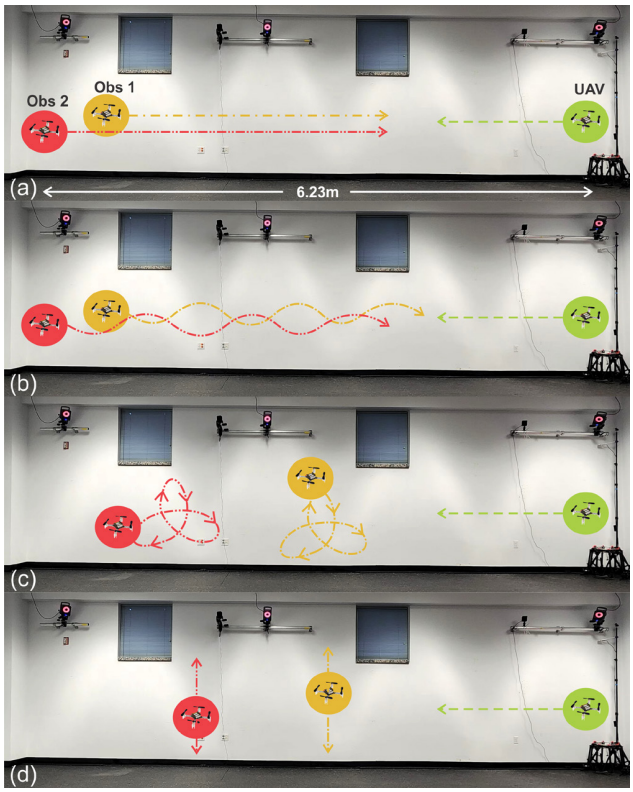


FIGURE 4. Real-world experimental arena equipped with VICON motion capture system, where a Crazyflie drone was used to represent the agent (drone highlighted with green color), and another two drones (highlighted with red and yellow color) were used to emulate dynamic obstacles. The arena consists of a 6.23 *m* length, 3.6 *m* width, and 2.2 *m* height. The four subfigures show experimental setups for different motion models of the dynamic obstacles, such as (a) straight-line, (b) sinusoid, (c) trefoil-knot trajectory, and (d) vertical wave motions.

An agent was tasked to navigate in a way that it had to encounter all of the obstacles and avoid collisions.

- **(b) UAV encountering dynamic obstacles moving in the opposite direction along a sinusoidal wave motion:** The dynamic obstacles exhibit a sinusoidal wave motion, and an agent was tasked to navigate in a way that it has to face at least one dynamic obstacle.
- **(c) UAV encountering dynamic obstacles moving along a trefoil-knot trajectory:** In this experiment, an agent has to fly through multiple dynamic obstacles exhibiting trefoil-knot trajectory. The agent may or may not directly encounter obstacles.
- **(d) UAV encountering dynamic obstacles moving in a vertical wave motion:** In this experiment, an agent has to fly through obstacles oscillating vertically along the z-axis. The min-max of the z-axis for obstacle motion is set so that the probability of interaction with the agent is high.

In simulation, five dynamic obstacles were used; in hardware experiments, two obstacles were deployed due to space and safety constraints.

B. EVALUATION METRICS

To ensure the statistical significance of the results, we employed the following evaluation metrics for all algorithms in both simulation and hardware:

- **Success rate (%age):** The ratio of runs in which the agents successfully navigate from their starting location to the target location without encountering any collisions.
- **Min obs separation (m):** The mean of the shortest Euclidean distances between an agent to any obstacle, recorded in all the 10 runs.
- **Time within the collision (s):** The mean duration during which a collision between an agent and a dynamic obstacle continues to take place, recorded in all 10 runs.
- **Average navigation duration (s):** The mean duration for the agents to navigate from their starting location to their target location across all 10 runs.
- **Average velocity of the drone (m/s):** The average velocity of the agent across all 10 runs in an experiment.
- **Number of collisions per 10 runs:** The total number of collisions in all 10 runs by the obstacles with an inflated agent.

C. SIMULATION SETUP

The simulation experiments were conducted on a laptop with a 12th Gen Intel(R) Core(TM) i7-12700H CPU, running Ubuntu 20.04 and ROS Noetic. The agent was modeled as a sphere $S_i \in \mathbb{R}^3$ of radius 0.15 m, while obstacles were modeled as bounding boxes of size $0.3 \times 0.3 \times 0.3$ m. The agent parameters were set as: maximum velocity 10 m/s, maximum acceleration 20 m/s^2, maximum jerk 30 m/s^3, and perception range 10 m. The start–goal separation was 90 m to ensure that the agent encounters obstacles at its full cruising velocity.

Each algorithm was tested across four obstacle velocities (1, 3, 5, and 10 m/s) and four motion models. For each velocity–motion pair, 10 runs were executed. For instance, we executed 10 simulation runs where the obstacle followed a trefoil knot trajectory at a velocity of 1 m/s. We performed a total of 800 experimental runs in the simulation (i.e., 4 velocities $\times$ 4 motions $\times$ 10 runs $=$ 160 runs per algorithm).

D. SIMULATION RESULTS

The experimental setup, including dimensions, start and goal locations, and velocities of both obstacles and agents, was kept identical for all algorithms. The results, as presented in Table 1 and Fig. 5, highlight the performance differentials between our approach (i.e., POF+MADER and POF+RMADER) and the baselines (i.e., MADER [8], RMADER [9], and Ego-Swarm [5]) algorithms.

1) COMPARISON BETWEEN MADER AND POF+MADER

Across all motion models, POF+MADER substantially outperformed the baseline MADER [8]. The success rate

TABLE 1. Performance differentials of our approach (i.e., POF+MADER and POF+RMADER) and the baselines (i.e., MADER, RMADER, and Ego-Swarm) algorithms across different Obstacle Motions and velocities, in a simulated environment.

Obstacle Motion	Method	Success Rate (% age)				Min. Obstacle Separation (m)				Time Within Collision (s)				Avg Navigation Duration (s)				Avg Velocity of Drone (m/s)				Collisions per 10 Runs			
		1 m/s	3 m/s	5 m/s	10 m/s	1 m/s	3 m/s	5 m/s	10 m/s	1 m/s	3 m/s	5 m/s	10 m/s	1 m/s	3 m/s	5 m/s	10 m/s	1 m/s	3 m/s	5 m/s	10 m/s	1 m/s	3 m/s	5 m/s	10 m/s
Straight line	MADER	30	0	20	20	0.65	0.55	0.47	0.39	0.07	0.07	0.11	0.10	13.32	13.24	13.32	13.21	6.19	6.82	6.77	**6.81**	9	11	10	9
	POF+MADER	**90**	**80**	**80**	**50**	**0.73**	**0.73**	**0.73**	**0.62**	**0.02**	**0.04**	**0.06**	**0.05**	**13.21**	**13.03**	**13.02**	**13.19**	**6.83**	**6.88**	**6.81**	6.79	**1**	**2**	**3**	**6**
	RMADER	50	30	30	0	0.55	0.41	0.50	0.30	0.49	0.27	0.49	0.74	17.72	17.05	17.61	17.43	5.20	5.40	5.22	5.27	11	16	19	34
	POF+RMADER	50	70	60	40	0.43	0.56	0.57	0.50	0.56	0.17	0.18	0.17	17.34	16.54	17.19	16.93	5.29	5.19	5.35	5.42	17	8	8	18
	ego planner	40	40	20	0	0.48	0.41	0.27	0.08	1.47	0.49	0.80	1.53	25.54	26.85	26.64	25.89	4.19	4.67	4.72	4.73	17	11	20	27
Sinusoid	MADER	10	20	0	0	0.60	0.56	0.32	0.29	0.10	0.12	0.19	0.18	13.03	13.02	13.44	**13.07**	**6.88**	**6.90**	6.74	**6.87**	14	11	15	25
	POF+MADER	**70**	**70**	**50**	**40**	**0.70**	**0.72**	**0.71**	**0.67**	**0.05**	**0.03**	**0.06**	**0.07**	**12.97**	**12.98**	**13.21**	13.42	6.35	6.33	**6.87**	6.74	**4**	**3**	**6**	**9**
	RMADER	40	30	30	20	0.50	0.45	0.36	0.45	0.41	0.23	0.55	0.62	17.36	16.85	17.41	17.43	5.31	5.46	5.29	5.29	17	16	24	27
	POF+RMADER	**70**	60	**50**	**40**	0.59	0.59	0.52	0.47	0.13	0.16	0.27	0.32	17.02	16.97	17.69	16.08	5.39	5.43	5.23	5.05	7	11	15	20
	ego planner	50	40	30	10	0.63	0.44	0.51	0.29	2.37	0.70	0.63	0.84	28.57	27.53	26.84	27.20	3.97	4.33	4.91	4.77	10	13	14	27
Vertical wave	MADER	50	30	40	30	0.69	0.46	0.63	0.46	0.17	0.25	0.17	0.15	13.26	**12.96**	**12.96**	**12.96**	**6.80**	**6.92**	**6.93**	**6.83**	8	15	7	10
	POF+MADER	**80**	70	**80**	40	**0.87**	**1.27**	**1.17**	**0.76**	0.09	0.12	**0.11**	0.11	**13.22**	13.50	14.12	13.30	6.79	6.70	6.47	6.35	**2**	6	**2**	8
	RMADER	60	50	30	30	0.67	0.56	0.52	0.36	0.13	0.15	0.24	0.26	16.79	16.93	16.95	17.01	5.46	5.41	5.40	5.39	4	8	14	18
	POF+RMADER	**80**	**80**	60	**50**	0.72	0.85	0.77	0.59	**0.05**	**0.04**	0.13	**0.08**	17.01	17.13	16.95	17.23	5.36	5.32	5.40	5.30	**2**	**2**	6	**7**
	ego planner	70	50	30	20	0.84	0.59	0.44	0.43	0.94	0.23	0.31	0.19	27.20	27.68	28.75	25.14	2.37	2.46	2.41	2.40	3	9	13	16
Trefoil knot	MADER	70	40	60	60	0.78	0.62	0.75	**0.82**	0.05	0.12	0.11	0.07	12.84	**12.86**	**12.81**	**12.82**	6.95	**6.97**	**6.99**	**6.95**	5	7	4	5
	POF+MADER	90	70	70	**70**	0.77	0.76	0.78	**0.82**	0.11	0.08	0.07	0.04	**12.81**	12.98	12.86	12.98	**6.96**	6.69	6.94	6.91	1	3	3	**3**
	RMADER	80	80	80	**70**	0.86	0.72	0.81	0.66	0.09	0.06	0.05	**0.03**	16.73	17.00	16.68	16.60	5.47	5.39	5.48	5.51	3	4	3	**3**
	POF+RMADER	**100**	**100**	**90**	**70**	**1.03**	**1.03**	**0.89**	0.69	**0.00**	**0.00**	**0.02**	0.05	17.34	17.52	17.49	17.31	5.27	5.21	5.21	5.27	**0**	**0**	2	**3**
	ego planner	**100**	80	**90**	50	0.97	0.75	0.83	0.58	**0.00**	0.04	0.08	0.11	25.59	26.01	25.87	26.61	2.37	4.02	4.07	4.09	**0**	2	**1**	7

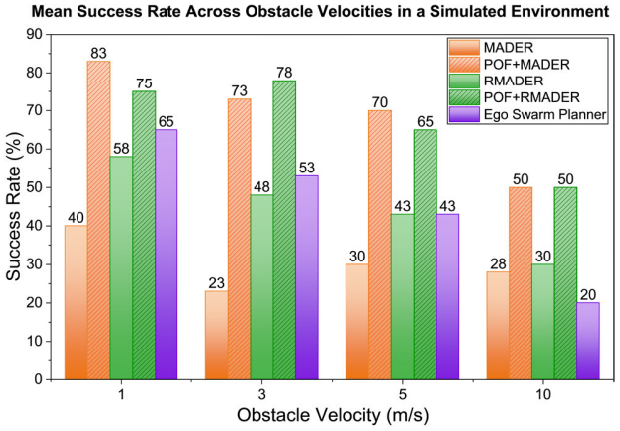


FIGURE 5. Success rate comparison of our approach (i.e., POF+MADER and POF+RMADER) and the baselines (i.e., MADER, RMADER, and Ego-Swarm) algorithms in a simulated environment, with obstacle velocities of 1 m/s, 3 m/s, 5 m/s, and 10 m/s, respectively.

improved from 30% to 68.75% (+38.75%), while the minimum agent–obstacle separation increased from 0.56 m to 0.8 m (+41.7%). Collision statistics show a reduction from 10.3 to 3.9 per 10 runs, and the mean collision duration decreased from 0.13 s to 0.07 s (−45.32%). These gains indicate that POF integration enables more conservative trajectory planning and ensures larger safety margins and drastically lowers collision risk.

The improvement, however, introduces a modest cost in efficiency. Average navigation time rose from 13.07 s to 13.18 s (+0.8%), with a corresponding decrease in average drone velocity. The effect is most apparent at higher obstacle velocities ($\geq$ 3 m/s), where MADER tends to attempt direct paths toward the goal and suffers collisions, whereas POF+MADER actively adjusts to predicted obstacle motion.

2) COMPARISON BETWEEN RMADER AND POF+RMADER

The integration of POF also substantially improved the robustness of RMADER. Success rate improved from 44.38% in RMADER [9] to 66.88% (+22.5%), while minimum obstacle separation increased from 0.54 m to 0.68 m (+24.42%). Collisions, which averaged 13.8 per 10 runs in RMADER, were reduced to 7.9 under POF+RMADER, with collision duration reduced from 0.3 s to nearly 0.15 s. Importantly, these safety gains were achieved without increasing average navigation time or reducing mean velocity. This shows that POF integration strengthens RMADER's robustness without compromising efficiency. This is particularly valuable in high-speed sinusoidal and trefoil-knot cases, where baseline RMADER experienced more failures.

3) COMPARISON BETWEEN POF-BASED METHODS AND EGO-SWARM

The Ego-Swarm planner, which is optimized for aggressive, high-speed trajectories, struggled with reliability in our test scenarios mentioned in subsection V-A, successfully completed only 45% of trials. Both POF+MADER and POF+RMADER outperformed in terms of safety and reliability, when compared to the Ego-Swarm [5] baseline. Ego-Swarm achieved an average navigation time (26.7 s) with mean velocity (3.8 m/s). Safety margins also improved markedly: from 0.53 m in Ego-Swarm to 0.8 m (+50.0%) in POF+MADER and 0.68 m (+26.5%) in POF+RMADER.

Ego-Swarm recorded the highest collision statistics, averaging 11.9 collisions per 10 runs with a mean collision duration of 0.67 s. In comparison, POF+MADER reduced collisions to 3.9 with an average duration of 0.09 s, while POF+RMADER reduced collisions to 7.9. These results indicate that Ego-Swarm's aggressive trajectory generation favors speed but compromises safety. By contrast,

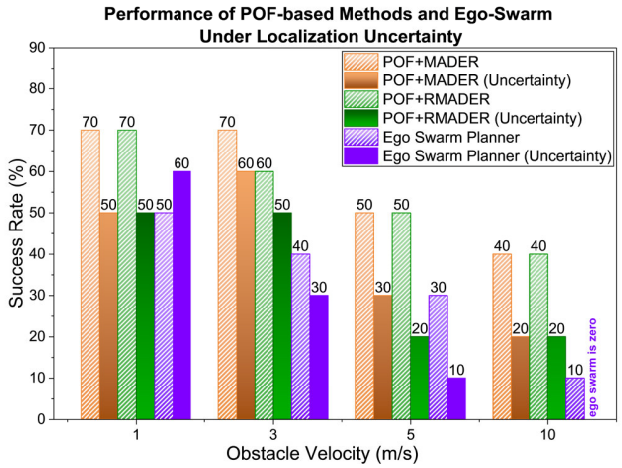


FIGURE 6. Success rate comparison of our approach (i.e., POF+MADER and POF+RMADER) and baselines (i.e., MADER [8], RMADER [9], and Ego-Swarm [5], under localization uncertainty in simulation, with obstacle velocities of 1 m/s, 3 m/s, 5 m/s, and 10 m/s, respectively.

POF-based methods achieve a more balanced outcome, trading slight increases in navigation time for substantial improvements in robustness. This makes them more appropriate for safety-critical UAV operations in dynamic environments.

4) PERFORMANCE UNDER LOCALIZATION UNCERTAINTY

To evaluate robustness against imperfect state estimation, we examined the impact of localization uncertainty on all algorithms under the sinusoidal motion model. This scenario was chosen for two reasons. First, both straight-line and vertical wave motions are predominantly confined to a single axis, which limits the complexity of their trajectories and makes them less susceptible to the effects of localization errors. In contrast, both trefoil-knot and sinusoid motions exhibit complex trajectories; however, the sinusoidal motion is characterized by continuous lateral oscillations and a less predictable periodicity, as opposed to the structured and repeating loops of the trefoil-knot. This intrinsic unpredictability in the lateral plane makes the sinusoidal motion particularly sensitive to localization uncertainty. Second, from the baselines results in Table 1, sinusoidal motion represented one of the most challenging cases, where baseline methods such as MADER and Ego-Swarm recorded their lowest success rates, whereas POF-based methods maintained significant improvements. This observation motivated our decision to further challenge POF-based methods by introducing localization uncertainty within the context of a sinusoidal obstacle trajectory.

The experimental setup remained identical to Section V-C, with dynamic obstacles following sinusoidal paths at velocities of 1, 3, 5, and 10 m/s. At each velocity, 10 runs were conducted with localization uncertainty applied to obstacle positions. The comparative outcomes are summarized in Table 2.

As shown in Fig. 6 and Table 2, the results show that localization uncertainty degrades the performance of all algorithms, but to varying degrees. MADER and Ego-Swarm experienced the sharpest drops in success rate, confirming their limited resilience to state estimation errors. RMADER showed moderate robustness, but still suffered from non-negligible collision rates. In contrast, POF-integrated approaches remained consistently superior: POF+MADER retained a success rate 17.5% lower than in the nominal case yet still outperformed MADER by a wide margin, while POF+RMADER sustained the highest overall reliability with only minor reductions in performance. Importantly, both POF-based methods preserved significantly larger obstacle separation distances than the baselines, which indicates that POF-integration substantially enhances robustness by maintaining conservative safety buffers and collision-free operation in most trials, under localization uncertainties.

E. REAL-WORLD EXPERIMENTS

To validate implementation feasibility beyond simulation, we conducted hardware experiments using Crazyflie 2.1 UAVs in a $6.23 \times 3.6 \times 2.2$ m indoor arena equipped with a VICON motion capture system. In each experiment, one Crazyflie acted as the navigating agent while two additional Crazyflies emulated dynamic obstacles. Due to the limited onboard computation of the Crazyflie platform, trajectory planning was executed in real-time on a host computer, with commands transmitted to the drones via Crazyradio PA. Reflective markers were mounted on all drones for motion capture, and the Crazyswarm library [50] was used to stream control inputs from the host to the UAVs. An illustration of the experimental setup is shown in Fig. 4.

The real-world evaluation was performed only on MADER and POF+MADER because of two main considerations: (i) MADER serves as the standard baseline in decentralized trajectory planning, which makes it the most appropriate reference for hardware validation; and (ii) POF+MADER is our proposed extension, enabling direct assessment of probabilistic filtering under realistic sensing, communication, and actuation conditions. With the hardware experiments, we want proof-of-concept feasibility and validation of POF integration in real-world on physical UAVs.

For the dynamic obstacles, we use the same four types of trajectories (i.e., straight line, sinusoid, vertical wave, and trefoil-knot trajectories), as shown in Fig. 4. For each trajectory, 10 runs were performed at obstacle velocities of 0.5, 1.0, and 1.5 m/s, resulting in 120 trials per algorithm. An experiment is considered successful if the agent navigates from their start location to the goal location without colliding with the obstacles. For safety purposes, the boundaries of the obstacles are inflated. Specifically, the actual diameter of the obstacle is 0.15 m, and for collision avoidance, this was inflated to 0.3 m (i.e., a 100% increase in size). A collision is considered to have occurred if an agent enters this inflated

TABLE 2. Comparison of POF-based methods and the baselines (under localization uncertainty).

Obstacle Motion	Method	Success Rate (% age)				Min. Obstacle Separation (m)				Time Within Collision (s)				Avg Navigation Duration (s)				Avg Velocity of Drone (m/s)				Collisions per 10 Runs			
		1 m/s	3 m/s	5 m/s	10 m/s	1 m/s	3 m/s	5 m/s	10 m/s	1 m/s	3 m/s	5 m/s	10 m/s	1 m/s	3 m/s	5 m/s	10 m/s	1 m/s	3 m/s	5 m/s	10 m/s	1 m/s	3 m/s	5 m/s	10 m/s
Sinusoid	MADER	10	20	0	0	0.60	0.56	0.32	0.29	0.10	0.12	0.19	0.18	13.03	13.02	13.44	**13.07**	**6.88**	**6.90**	6.74	**6.87**	14	11	15	25
	POF+MADER	**70**	**70**	**50**	**40**	**0.70**	**0.72**	**0.71**	**0.67**	**0.05**	**0.03**	**0.06**	**0.07**	**12.97**	**12.98**	**13.21**	13.42	6.35	6.33	**6.87**	6.74	**4**	**3**	**6**	**9**
	RMADER	40	30	30	20	0.50	0.45	0.36	0.45	0.41	0.23	0.55	0.62	17.36	16.85	17.41	17.43	5.31	5.46	5.29	5.29	17	16	24	27
	POF+RMADER	**70**	60	**50**	**40**	0.59	0.59	0.52	0.47	0.13	0.16	0.27	0.32	17.02	16.97	17.69	16.08	5.39	5.43	5.23	5.05	7	11	15	20
	ego planner	50	40	30	10	0.63	0.44	0.51	0.29	2.37	0.70	0.63	0.84	28.57	27.53	26.84	27.20	3.97	4.33	4.91	4.77	10	13	14	27

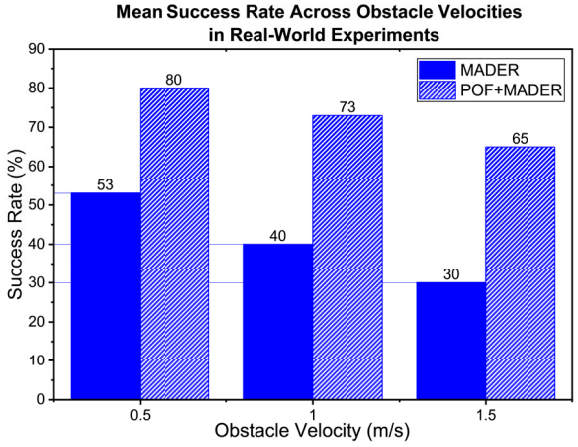


FIGURE 7. Performance evaluation of MADER and POF+MADER in a real-world environment, with obstacle velocities of 0.5m/s, 1.0m/s, and 1.5m/s, respectively.

boundary, which ensures a safer buffer zone around the obstacle.

F. REAL-WORLD RESULTS

Table 3 and Fig. 7 present the results. POF+MADER consistently achieved higher success rates, fewer collisions, and greater minimum separation than MADER, validating the safety improvements observed in simulation. For example, in sinusoidal motion at 1.5 m/s, POF+MADER achieved a 90% success rate compared to 65% for MADER.

VI. DISCUSSION

Our experimental evaluation, presented in Section V, provided strong evidence that integrating a probabilistic obstacle filter (POF) into optimization-based planners like MADER and RMADER yields significant gains in robustness. This section moves beyond the quantitative results to interpret their deeper implications, discussing the fundamental advantages of our proactive planning architecture, the reasons for its resilience to uncertainty, and the practical considerations for real-world deployment.

A. THE ADVANTAGE OF PROACTIVE PLANNING UNDER UNCERTAINTY

A core insight from our comparative analysis is the fundamental advantage of proactive planning over purely reactive strategies. The baseline planners, particularly MADER and Ego-Swarm, perform adequately in simple scenarios but exhibit brittle performance as obstacle dynamics become more complex or perception becomes noisy. This is because their planning horizon is limited; they react to the current state of the world, leaving little time to generate smooth, safe maneuvers in response to fast-approaching or erratically moving obstacles.

In contrast, our POF-integrated methods succeed by effectively "looking into the future." POF provides the planner with a high-fidelity forecast of where obstacles are most likely to be, giving the optimizer a crucial "heads-up." This allows it to generate evasive trajectories that are not just reactive, but preemptive. The most compelling finding is the nature of the trade-off this provides: our results show nearly 39% improvement in success rate for a negligible 0.8% increase in navigation time. This demonstrates that the computational cost of filtering and prediction is far outweighed by the gains in safety and reliability, representing a highly favorable trade-off for any safety-critical application.

B. STATE ESTIMATION AS A FOUNDATION FOR ROBUST PLANNING

The performance of our system under injected sensor noise reveals another key principle: robust planning is critically dependent on robust state estimation. The sharp decline in performance for the baseline planners under localization uncertainty highlights their sensitivity to noisy inputs. A planner without a filtering layer is forced to react to every spurious measurement, which can lead to jerky, inefficient, or unsafe trajectories.

The resilience of POF+MADER and POF+RMADER stems directly from the POF's ability to "see through the noise." By maintaining a probabilistic belief over the obstacle's state (including its velocity and covariance), the filter effectively smoothes the raw measurements and provides a stable, physically plausible estimate to the planner. The planner, therefore, operates on a clean, reliable stream of data, allowing it to make more consistent and accurate decisions. This confirms that for real-world robotics, where perfect perception is impossible, tightly coupling the planning and estimation processes is essential for achieving robust autonomy.

C. LIMITATIONS AND EXTENSIONS

The combination of simulation and hardware results supports the feasibility of POF integration for real-world UAV navigation. Its lightweight nature makes this advanced

TABLE 3. Real-world Comparison of POF+MADER with MADER.

Obstacle Motion	Method	Success Rate (% age)			Min. Obstacle Separation (m)			Time Within Collision (s)			Avg Navigation Duration (s)			Avg Velocity of Drone (m/s)			Collisions per 10 Runs		
		1 m/s	3 m/s	5 m/s	1 m/s	3 m/s	5 m/s	1 m/s	3 m/s	5 m/s	1 m/s	3 m/s	5 m/s	1 m/s	3 m/s	5 m/s	1 m/s	3 m/s	5 m/s
Straight line	MADER	50	40	0	0.18	0.15	0.14	0.96	1.29	1.81	13.60	**13.53**	**14.39**	**0.61**	0.62	0.63	5	6	11
	POF+MADER	**90**	**80**	**70**	**0.30**	**0.27**	**0.24**	**0.19**	**0.41**	**0.49**	**13.38**	13.67	13.42	0.62	**0.61**	**0.62**	**1**	**2**	**4**
Sinusoid	MADER	60	50	40	0.19	0.17	0.15	0.67	2.66	2.71	**12.95**	**13.26**	**13.14**	0.64	**0.61**	**0.62**	4	10	12
	POF+MADER	**90**	**80**	**70**	**0.31**	**0.23**	**0.19**	**0.51**	**0.37**	**1.36**	13.33	13.31	13.27	**0.62**	0.62	**0.62**	**2**	**3**	**6**
Vertical wave	MADER	20	10	10	0.16	0.12	0.13	2.14	3.06	3.07	13.08	**13.42**	13.70	0.59	0.61	0.61	9	10	12
	POF+MADER	**60**	**50**	**40**	**0.19**	**0.22**	**0.16**	**1.02**	**1.66**	**1.90**	**13.07**	13.68	**13.50**	**0.58**	**0.61**	**0.61**	**4**	**5**	**8**
Trefoil knot	MADER	**80**	60	70	0.26	0.22	0.30	**0.41**	1.43	0.92	13.60	**13.57**	13.32	**0.60**	**0.61**	**0.62**	**2**	5	3
	POF+MADER	**80**	**80**	**80**	**0.43**	**0.37**	**0.46**	0.51	**0.48**	**0.88**	**13.11**	12.60	**13.05**	0.61	0.63	0.63	**2**	**2**	**2**

predictive capability accessible to small aerial platforms without requiring expensive, high-power onboard computers. However, the current implementation operates under certain assumptions that define its scope and potential directions for extension.

1) CURRENT LIMITATIONS

First, we employ a constant-velocity motion model that proved effective for non-cooperative obstacles in our experiments. This model supports computational efficiency and short-term prediction but complex maneuvers such as coordinated turns or evasive actions may require more advanced models.

Second, the Gaussian noise model effectively represents motion-capture and typical IMU-based localization uncertainties, but real-world sensors may exhibit non-Gaussian characteristics, including multi-modal distributions from vision-based detectors or heavy-tailed distributions from intermittent failures.

Third, hardware validation was conducted on Crazyflie 2.1 platforms in controlled indoor environments with limited dynamic obstacles. Although this provides compelling proof-of-concept feasibility, outdoor scenarios introduce GPS degradation, wind disturbances, and larger obstacle populations that exceed this constrained operational envelope.

Based on these limitations, several extensions would broaden POF+MADER's applicability to complex real-world deployments.

2) MULTI-AGENT COORDINATION EXTENSIONS

The current decentralized architecture treats each agent's obstacle tracking independently. However, in scenarios where multiple agents have overlapping perception ranges, cooperative estimation strategies could substantially improve prediction accuracy. Distributed filtering approaches, such as consensus-based estimation in sensor networks [51], could be used to fuse obstacle measurements across agents to reduce uncertainty bounds. This collaborative sensing would be particularly valuable in GPS-denied environments where individual sensor reliability degrades due to occlusions.

Agents can further optimize performance by sharing learned obstacle behavior patterns via distributed databases. When one agent observes recurring motion characteristics (such as pedestrian crosswalk behavior or vehicle lane-following patterns), this information could propagate to other swarm members, enabling more accurate prediction than generic constant-velocity models provide.

3) HETEROGENEOUS OBSTACLE TYPES HANDLING

Real-world environments contain diverse obstacle types with distinct motion characteristics. Integrating upstream obstacle classification would enable motion model selection tailored to obstacle type. For instance, pedestrian detection could invoke Social Force Models [52], which capture goal-directed behavior and social interactions, and vehicle detection would trigger car-following models such as the Intelligent Driver Model [53] that represent acceleration patterns and lane constraints.

To handle abrupt maneuver changes without explicit type information, Interacting Multiple Model (IMM) filters [54] could be used to maintain parallel motion hypotheses (constant velocity, coordinated turn, sudden stop) with dynamically updated probabilities based on measurement likelihood. This multi-hypothesis approach provides robustness to unpredictable behaviors such as birds' evasive responses or vehicles' emergency braking, common in outdoor scenarios. Additionally, extending obstacle representation from our spherical models to oriented bounding boxes would enable tighter collision constraints for elongated obstacles and potentially reduce planning conservatism.

4) OUTDOOR ENVIRONMENT ADAPTATION

Outdoor deployment introduces systematic challenges addressable through targeted extensions. Wind disturbances violate constant-velocity assumptions over longer horizons; incorporating measured or estimated wind conditions into the process noise covariance (Q matrix) would maintain prediction fidelity and explicitly represent increased uncertainty in planning constraints. For navigation in GPS-denied areas, integrating POF with vision-based SLAM systems would

require obstacle tracking and ego-motion estimation to be coupled.

These extensions build upon the modular POF+MADER foundation rather than requiring a fundamental redesign. Future work will prioritize heterogeneous classification and multi-model estimation to directly address current modeling limitations and enable outdoor validation.

VII. CONCLUSION AND FUTURE WORK

In this work, we presented a novel hybrid architecture, POF+MADER, that successfully bridges the critical gap between the need for proactive planning and the reality of perception uncertainty in dynamic environments. By integrating a lightweight, real-time probabilistic obstacle filter into the core of state-of-the-art multi-agent trajectory planners, we have eliminated their restrictive reliance on a priori knowledge of obstacle trajectories. The proposed approach transforms these planners into robust, predictive systems capable of generating safer paths by explicitly modeling and reacting to the forecasted motion of unknown dynamic obstacles.

Comprehensive validation through simulation and hardware experimentation shows the advantages of the proposed proactive POF-integrated methods. In simulation benchmarks, POF-based methods outperformed baseline methods in terms of safety and reliability, as demonstrated by collision rate reduction of up to 39% without compromising navigational efficiency. The practical feasibility of this lightweight architecture is then confirmed through its successful deployment on Crazyflie 2.1 UAVs. In these hardware trials, POF+MADER consistently outperformed MADER under realistic sensing and communication conditions, confirming the practical applicability of the approach.

POF-based methods presents advantages, limitations, and areas for future direction, as outlined in subsection VI-C such as integrating more advanced motion models and scaling the validation to larger and outdoor aerial platforms.

ACKNOWLEDGMENT

This article solely reflects the opinions and conclusions of its authors and not NSF or any other entity.

REFERENCES

[1] M. Hassanalian and A. Abdelkefi, "Classifications, applications, and design challenges of drones: A review," *Prog. Aerosp. Sci.*, vol. 91, pp. 99–131, May 2017.

[2] J. Baca, S. I. Ullah, and P. Rangel, "Coaxial modular aerial system and the reconfiguration applications," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2023, pp. 11929–11935.

[3] J. Horyna, T. Baca, V. Walter, D. Albani, D. Hert, E. Ferrante, and M. Saska, "Decentralized swarms of unmanned aerial vehicles for search and rescue operations without explicit communication," *Auto. Robots*, vol. 47, no. 1, pp. 77–93, Jan. 2023.

[4] B. Senbaslar and G. S. Sukhatme, "DREAM: Decentralized real-time asynchronous probabilistic trajectory planning for collision-free multi-robot navigation in cluttered environments," *IEEE Trans. Robot.*, vol. 41, pp. 573–592, 2025.

[5] X. Zhou, J. Zhu, H. Zhou, C. Xu, and F. Gao, "EGO-swarm: A fully autonomous and decentralized quadrotor swarm system in cluttered environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 4101–4107.

[6] J. Park, D. Kim, G. C. Kim, D. Oh, and H. J. Kim, "Online distributed trajectory planning for quadrotor swarm with feasibility guarantee using linear safe corridor," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 4869–4876, Apr. 2022.

[7] Z. Liu, B. Chen, H. Zhou, G. Koushik, M. Hebert, and D. Zhao, "MAPPER: Multi-agent path planning with evolutionary reinforcement learning in mixed dynamic environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 11748–11754.

[8] J. Tordesillas and J. P. How, "MADER: Trajectory planner in multiagent and dynamic environments," *IEEE Trans. Robot.*, vol. 38, no. 1, pp. 463–476, Feb. 2022.

[9] K. Kondo, J. Tordesillas, R. Figueroa, J. Rached, J. Merkel, P. C. Lusk, and J. P. How, "Robust MADER: Decentralized and asynchronous multiagent trajectory planner robust to communication delay," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2023, pp. 1687–1693.

[10] F. F. Lizzio, M. Bugaj, J. Rostás, and S. Primatesta, "Comparison of multiple models in decentralized target estimation by a UAV swarm," *Drones*, vol. 8, no. 1, p. 5, Dec. 2023.

[11] M. Matsushima, T. Hashimoto, and F. Miyazaki, "Learning to the robot table tennis task-ball control & rally with a human," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, Oct. 2003, pp. 2962–2969.

[12] P. Shukla, S. Shukla, and A. Kumar Singh, "Trajectory-prediction techniques for unmanned aerial vehicles (UAVs): A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 27, no. 3, pp. 1867–1910, Jun. 2025.

[13] L. Lu, G. Fasano, A. Carrio, M. Lei, H. Bavle, and P. Campoy, "A comprehensive survey on non-cooperative collision avoidance for micro aerial vehicles: Sensing and obstacle detection," *J. Field Robot.*, vol. 40, no. 6, pp. 1697–1720, Sep. 2023.

[14] Z. Xu, X. Zhan, B. Chen, Y. Xiu, C. Yang, and K. Shimada, "A real-time dynamic obstacle tracking and mapping system for UAV navigation and collision avoidance with an RGB-D camera," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2023, pp. 10645–10651.

[15] M. Lu, H. Chen, and P. Lu, "Perception and avoidance of multiple small fast moving objects for quadrotors with only low-cost RGBD camera," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 11657–11664, Oct. 2022.

[16] Z. Xu, D. Deng, Y. Dong, and K. Shimada, "DPMPC-planner: A real-time UAV trajectory planning framework for complex static environments with dynamic obstacles," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 250–256.

[17] M. Zhang, B. Pang, C. Yan, M. Feroskhan, and C. Lv, "Real-time avoidance of obstacles and emergent geo-fences for urban air mobility using deep reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 26, no. 11, pp. 20091–20108, Nov. 2025.

[18] J. Tordesillas and J. P. How, "Deep-PANTHER: Learning-based perception-aware trajectory planner in dynamic environments," *IEEE Robot. Autom. Lett.*, vol. 8, no. 3, pp. 1399–1406, Mar. 2023.

[19] S. Tanaka, A. Asignacion, T. Nakata, S. Suzuki, and H. Liu, "Review of biomimetic approaches for drones," *Drones*, vol. 6, no. 11, p. 320, Oct. 2022.

[20] S. Poudel, M. Y. Arafat, and S. Moh, "Bio-inspired optimization-based path planning algorithms in unmanned aerial vehicles: A survey," *Sensors*, vol. 23, no. 6, p. 3051, Mar. 2023.

[21] Z. Xu, Y. Xiu, X. Zhan, B. Chen, and K. Shimada, "Vision-aided UAV navigation and dynamic obstacle avoidance using gradient-based B-spline trajectory optimization," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2023, pp. 1214–1220.

[22] E. Aldao, L. González-Desantos, H. Michinel, and H. González-Jorge, "UAV obstacle avoidance algorithm to navigate in dynamic building environments," *Drones*, vol. 6, no. 1, p. 16, Jan. 2022.

[23] W. Xia, F. Song, and Z. Peng, "Dynamic obstacle perception technology for UAVs based on LiDAR," *Drones*, vol. 9, no. 8, p. 540, Jul. 2025.

[24] S. Hai, X. Na, Z. Feng, J. Shi, and Q. Sun, "PENC: A predictive-estimative nonlinear control framework for robust target tracking of fixed-wing UAVs in complex urban environments," *Sci. Rep.*, vol. 15, no. 1, Aug. 2025, Art. no. 29753.

[25] W. Dai, Z. Wang, and C.-C. Ding, "Biomimetic swarm-based UAV collision avoidance control," in *Proc. Int. Conf. Comput. Commun. Netw.*, 2025, pp. 705–718.

[26] T. Wakabayashi, Y. Suzuki, and S. Suzuki, "Dynamic obstacle avoidance for multi-rotor UAV using chance-constraints based on obstacle velocity," *Robot. Auto. Syst.*, vol. 160, Feb. 2023, Art. no. 104320.

[27] J. Park, Y. Lee, I. Jang, and H. J. Kim, "Decentralized trajectory planning for quadrotor swarm in cluttered environments with goal convergence guarantee," *Int. J. Robot. Res.*, vol. 44, no. 8, pp. 1336–1359, Jul. 2025.

[28] P. Chen, Y. Jiang, Y. Dang, T. Yu, and R. Liang, "Real-time efficient trajectory planning for quadrotor based on hard constraints," *J. Intell. Robotic Syst.*, vol. 105, no. 3, p. 52, Jul. 2022.

[29] T. Wang, L. Yang, Y. Chang, Z. Huang, H. Jiang, and Y. Zheng, "A review of dynamic obstacle avoidance for unmanned aerial vehicles (UAVs)," in *Proc. 7th Int. Symp. Auto. Syst. (ISAS)*, May 2024, pp. 1–6.

[30] K. Tamanakijprasart, S. Mondal, and A. Tsourdos, "Multi-agent path planning in dynamic environment with communication disruptions," in *Proc. Int. Conf. Control, Autom. Diagnosis (ICCAD)*, Jul. 2025, pp. 1–6.

[31] L. Ren, M. Li, S. Fan, Y. Zhang, F. Yu, and J. Yang, "Cooperative control method of multi-agent formation and obstacle avoidance," in *Proc. IEEE Int. Conf. Mechatronics Autom. (ICMA)*, Jun. 2025, pp. 1066–1072.

[32] C. Toumieh and A. Lambert, "Decentralized multi-agent planning using model predictive control and time-aware safe corridors," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 11110–11117, Oct. 2022.

[33] B. Luders, M. Kothari, and J. P. How, "Chance constrained RRT for probabilistic robustness to environmental uncertainty," in *Proc. AIAA Guid., Navigat., Control Conf.*, 2010, pp. 1–16.

[34] M. Garvanova, I. Garvanov, D. Borissova, N. Kerimbayev, and Z. Menlibay, "Uav detection and recognition technologies," in *Proc. Int. Conf. Telecommun. Remote Sens.* Cham, Switzerland: Springer, 2024, pp. 1–18.

[35] H.-C. Chen, "Monocular vision-based obstacle detection and avoidance for a multicopter," *IEEE Access*, vol. 7, pp. 167869–167883, 2019.

[36] J. Lin, H. Zhu, and J. Alonso-Mora, "Robust vision-based obstacle avoidance for micro aerial vehicles in dynamic environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 2682–2688.

[37] J. Zhong, M. Li, Y. Chen, Z. Wei, F. Yang, and H. Shen, "A safer vision-based autonomous planning system for quadrotor UAVs with dynamic obstacle trajectory prediction and its application with LLMs," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. Workshops (WACVW)*, Jan. 2024, pp. 920–929.

[38] P. Kumar, D. Makwana, O. Susladkar, A. Mittal, and P. K. Kalra, "MOVES: Movable and moving LiDAR scene segmentation in label-free settings using static reconstruction," *Pattern Recognit.*, vol. 155, Nov. 2024, Art. no. 110651.

[39] T. Eppenberger, G. Cesari, M. Dymczyk, R. Siegwart, and R. Dubé, "Leveraging stereo-camera data for real-time dynamic obstacle detection and tracking," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 10528–10535.

[40] X. Liu, G. V. Nardari, F. Cladera, Y. Tao, A. Zhou, T. Donnelly, C. Qu, S. W. Chen, R. A. F. Romero, C. J. Taylor, and V. Kumar, "Large-scale autonomous flight with real-time semantic SLAM under dense forest canopy," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 5512–5519, Apr. 2022.

[41] S. I. Ullah and A. Muhammad, "Autonomous navigation and mapping of water channels in a simulated environment using micro-aerial vehicles," in *Proc. Int. Conf. Robot. Autom. Ind. (ICRAI)*, Mar. 2023, pp. 1–7.

[42] G. Chen, W. Dong, P. Peng, J. Alonso-Mora, and X. Zhu, "Continuous occupancy mapping in dynamic environments using particles," *IEEE Trans. Robot.*, vol. 40, pp. 64–84, 2024.

[43] G. Chen, W. Dong, X. Sheng, X. Zhu, and H. Ding, "An active sense and avoid system for flying robots in dynamic environments," *IEEE/ASME Trans. Mechatronics*, vol. 26, no. 2, pp. 668–678, Apr. 2021.

[44] S. Amutha, S. V. Ramanan, S. Ganesan, B. Leena, J. Akshya, M. D. Choudhry, and M. Sundarrajan, "Crow search algorithm for energy-efficient UAV path planning in large-scale aerial surveys," in *Proc. 3rd Int. Conf. Device Intell., Comput. Commun. Technol. (DICCT)*, Mar. 2025, pp. 556–561.

[45] Y. K. Nakka, A. Liu, G. Shi, A. Anandkumar, Y. Yue, and S.-J. Chung, "Chance-constrained trajectory optimization for safe exploration and learning of nonlinear systems," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 389–396, Apr. 2021.

[46] E. Olcay, H. Meeß, and G. Elger, "Dynamic obstacle avoidance for UAVs using MPC and GP-based motion forecast," in *Proc. Eur. Control Conf. (ECC)*, Jun. 2024, pp. 1024–1031.

[47] J. Zhou, Y. Gao, O. Johansson, B. Olofsson, and E. Frisk, "Robust predictive motion planning by learning obstacle uncertainty," *IEEE Trans. Control Syst. Technol.*, vol. 33, no. 3, pp. 1006–1020, May 2025.

[48] S. Argiliana, E. Ekawati, and F. Mukhlish, "Adaptive strategies for dynamic obstacle avoidance and formation control in multi-agent drone systems: A review," *J. Robot. Control (JRC)*, vol. 6, no. 4, pp. 1710–1720, 2025.

[49] J. Tordesillas and J. P. How, "MINVO basis: Finding simplexes with minimum volume enclosing polynomial curves," *Computer-Aided Design*, vol. 151, Oct. 2022, Art. no. 103341.

[50] J. A. Preiss, W. Honig, G. S. Sukhatme, and N. Ayanian, "Crazyswarm: A large nano-quadcopter swarm," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 3299–3304.

[51] C. Gao, Z. Wang, J. Hu, Y. Liu, and X. He, "Consensus-based distributed state estimation over sensor networks with encoding-decoding scheme: Accommodating bandwidth constraints," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 6, pp. 4051–4064, Nov. 2022.

[52] M. M. Rashid, M. Seyedi, and S. Jung, "Simulation of pedestrian interaction with autonomous vehicles via social force model," *Simul. Model. Pract. Theory*, vol. 132, Apr. 2024, Art. no. 102901.

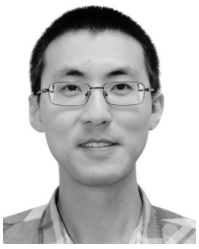
[53] T. T. Zhang, P. J. Jin, S. T. McQuade, A. Bayen, and B. Piccoli, "Car-following models: A multidisciplinary review," *IEEE Trans. Intell. Vehicles*, vol. 10, no. 1, pp. 92–116, Jan. 2025.

[54] I. H. Lee and C. G. Park, "An improved interacting multiple model algorithm with adaptive transition probability matrix based on the situation," *Int. J. Control, Autom. Syst.*, vol. 21, no. 10, pp. 3299–3312, Oct. 2023.

SYED IZZAT ULLAH received the B.S. degree in telecommunication engineering from the Balochistan University of IT, Engineering, and Management Sciences, Pakistan, in 2016, and the M.S. degree in electrical engineering (robotics and control systems) from Lahore University of Management Sciences (LUMS), Pakistan, in 2019. He is currently pursuing the Ph.D. degree in computer science with Texas A&M University-Corpus Christi, Corpus Christi, TX, USA. His research has been presented at leading robotics conferences and has held research positions at the National Center of Robotics and Automation, Pakistan, and as a Visiting Researcher with the Robotics Research Laboratory, TU Kaiserslautern, Germany. His research interests include risk-aware multi-robot motion planning, integrating reinforcement learning, transfer learning, and optimization to improve autonomous navigation in dynamic and uncertain environments.

JOSÉ BACA received the B.S. degree in electrical engineering from Instituto Tecnológico de Matamoros, Mexico, the M.Sc. degree in mechatronics from the University of Applied Sciences in Aachen, Germany, and the Ph.D. degree in automation and robotics from Universidad Politécnica de Madrid, Spain. He was a Postdoctoral Researcher with the Computer Science Department, University of Nebraska at Omaha, USA. He is currently an Associate Professor with the Department of Engineering, Texas A&M University-Corpus Christi, USA. He has organized and co-chaired international conferences and workshops and has been involved in projects funded by agencies, such as DoD, NSF, USDA, and NASA. His research interests include the development of coordination and control strategies for unmanned autonomous systems and the integration of modular systems across different domains, such as robotics, search and rescue, coastal resilience, space, industry, agriculture, healthcare, and education.

PABLO RANGEL received the Ph.D. degree in electrical and computer engineering from The University of Texas at El Paso, in 2017. He is currently an Assistant Professor with Texas A&M University-Corpus Christi. He has been the Chief Operations Officer of the Laboratory for Industrial Metrology and Automation (LIMA) on Biomedical Engineering Projects. He has also been a Research Associate with the Regional Cyber and Energy Security (RCES) Center on Cyber Security Projects. In addition, he has extensive research experience on UAV systems. His current research interests include test and evaluation of autonomous systems, unmanned aircraft systems (UAS), multi-agent, cyber-physical systems, systems engineering, robotics, modern control theory, mechatronics, fuzzy logic, biomedical engineering, wireless communications, and cybersecurity.



TIANXING CHU received the Ph.D. degree in photogrammetry and remote sensing from Peking University, Beijing, China, in 2012. He is currently an Associate Professor with Texas A&M University-Corpus Christi (TAMU-CC), Corpus Christi, TX, USA. Prior to his current role, he was with TAMU-CC as an Assistant Professor, from 2018 to 2025. His current research interests include emergent remote sensing and geomatics techniques, ubiquitous navigation and positioning, indoor simultaneous localization and mapping, and machine learning applied to geospatial applications.

CARLOS RUBIO-MEDRANO received the master's degree from The University of Texas at El Paso and the Ph.D. degree from Arizona State University. He is currently an Assistant Professor of computer science with Texas A&M University-Corpus Christi (TAMU-CC), where he leads the Cybersecurity Research and Innovation Laboratory (CSRIL). He has substantial experience in topics related to cyber risks and protections for emerging technologies, such as generative artificial intelligence, cyber-physical systems, and mobile augmented reality. His research work has been published in top venues in the cybersecurity field, and the undergraduate and graduate students under his mentorship have received several awards at the local, state, and national levels. He is the Principal Investigator or Co-Principal Investigator of various funded research projects sponsored by the National Science Foundation (NSF), the United States Department of Transportation, and Google Inc. He was recently recognized as a Cybersecurity Fellow by the Computing Alliance of Hispanic Serving Institutions (CAHSI), an INCLUDES Alliance sponsored by NSF.