Probabilistic Modeling and Control for Multi-UAV Search Over Uneven Terrain

Luka Lanča, Karlo Jakac, and Stefan Ivić

Abstract—**This article addresses survey missions involving multiple uncrewed aerial vehicles (UAVs) over complex, varying terrain. The methodology integrates a probabilistic model of target's position uncertainty with UAV flight dynamics, camera properties, and a machine learning-based detection system. It estimates undetected target probability and overall search performance, feeding into a feedback loop that combines 2-D ergodic search with model predictive control (MPC) of UAV altitude and velocity. Trial trajectory optimization accounts for sensing characteristics and operational constraints, producing terrain-aware, collision-free trajectories that balance area coverage with target detection. Simulations demonstrate the integration of MPC and ergodic search, enabling dynamic altitude adjustments to enhance the search performance. The control algorithm operates in real time and performs reliably under uncertainty. Field experiments provided training data, validated the method, and confirmed compliance with motion constraints. Detection rates closely match model predictions, demonstrating stable performance even under significant deviations from ideal conditions. The framework, thus, offers a reliable solution for autonomous multi-UAV search operations in real-world environments.**

Index Terms—**Ergodic search, motion control, multirobot systems, search and rescue robots.**

I. Introduction

The main objective of search and rescue (SAR) operations is to locate lost, missing, or injured individuals as quickly as possible, to minimize their suffering and reduce the risk of environmental effects, injuries, and other hazards. These operations often occur in remote or inaccessible areas, such as wilderness, urban ruins, and the ocean, which makes them particularly challenging. Autonomous UAVs can offer numerous benefits in SAR operations, as they can efficiently and quickly survey a large area at a relatively low cost, which is particularly important when time is of the essence.

While conducting area surveys, keeping the UAV within a designated altitude range is crucial to achieve good ratio between area coverage and image details. The primary objective of the search is speed, prioritizing maximum velocity to cover as much area as possible, which is crucial for SAR operations over endurance. Maximizing velocity while ensuring a collision free path, adhering to desired altitude goals and constraints over uneven terrain while respecting the UAV's technical constraints and capabilities poses a significant challenge. Employing multiple UAVs can further expedite the search process, but it requires high computational efficiency to coordinate the search and ensure safe operation.

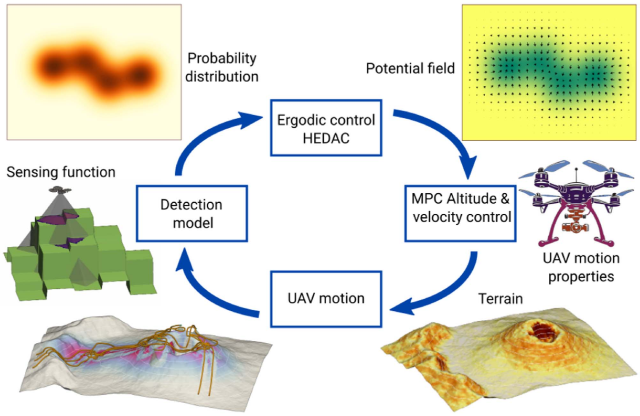


Fig. 1. Streamlined visual representation of the proposed UAV search framework, highlighting its key components and functionality.

The location of the (undetected) target is uncertain, and represented by a probability distribution over the search domain, which evolves dynamically based on the UAVs' sensing effects. These effects involve properties of the sensor and probabilistic performance of the target detection model. The distance between the sensor and the targets effects the detection performance. This poses a challenge when searching over uneven terrain, since the topography needs to be reflected in the UAV flight altitude control, motion constraints, as well as in the probabilistic model of the search.

To address the challenges mentioned, a multiagent ergodic control approach, enabling efficient two-dimensional area search, was paired with model predictive control (MPC) for managing velocity and altitude, allowing for three-dimensional motion control (see Fig. 1). Ergodic control was implemented using the heat equation driven area coverage (HEDAC) method. The sensing and detection model utilize a sensor with a pyramidal field of view (FOV) that simulates an orthophoto camera. The camera and computer vision (CV) detection system are integrated to provide a broad definition of the detection rate, which depends on the distance to the target. Terrain is represented using

Received 23 April 2025; revised 3 October 2025; accepted 17 November 2025. Date of publication 18 December 2025; date of current version 5 January 2026. This work was supported by the Croatian Science Foundation under Project UIP-2020-02-5090. The Python code needed to reproduce this research is available upon request. This article was recommended for publication by Associate Editor D. Panagou and Editor P. Robuffo Giordano upon evaluation of the reviewers' comments. *(Corresponding author: Stefan Ivić.)*

The authors are with the Faculty of Engineering, University of Rijeka, 51000 Rijeka, Croatia (e-mail: stefan.ivic@uniri.hr).

Data are available online at: https://osf.io/t947u/.
Digital Object Identifier 10.1109/TRO.2025.3645884

TABLE I
FEATURE COMPARISON FOR REVIEWED SEARCH METHODS

Study	Method	Ergodic control	Real-time control	Multi-UAV	Velocity control	Probabilistic detection	Terrain following	Collision avoidance
Mavrommati et al., 2017. [37]	MPC	✓	✓	✓	-	✓	-	-
Abraham and Murphey, 2018. [1]	MPC	✓	✓	✓	-		-	✓
Silvagni et al., 2017. [50]	Pattern search	-	✓	-	-	-	✓	-
Tordesillas and How, 2021. [55]	MADER	-	✓	✓	-	-	-	✓
Phung et al., 2021. [41]	PSO	-	-	-	-	-	✓	✓
Hubenko et al., 2011. [18]	SMC	✓	✓	✓	-	-	-	-
Ivić et al., 2022. [24]	HEDAC	✓	✓	✓	-	✓	-	✓
Ivić et al., 2023.[22]	HEDAC	✓	-	✓	-	✓	✓	✓
Proposed	HEDAC + MPC	✓	✓	✓	✓	✓	✓	✓

open-access digital elevation model while accounting for terrain inaccuracies, vegetation height, and human-made structures, ensuring safety. The UAV performance and search success of the proposed method is evaluated through three test cases with varying terrain complexities.

One of the main contributions of this article is the development of an accurate probabilistic model managing the target probability distribution over uneven terrain using arbitrary sensing functions that can be derived from the performance of realistic sensors. Ergodic control employs a convolution of varying base functions reflecting changes in altitude and terrain influencing the sensing. The MPC for altitude and velocity considers UAV properties and motion constraints for efficient and feasible flight. The proposed control framework was tested in realistic search scenarios, demonstrating advantages over baseline methods and robustness to control and localization uncertainties. Finally, the probabilistic model and control framework is validated in field experiments confirming assessed metric for overall detection probability. All simulations inputs and results, terrain and mesh data, collected telemetry and images, trained YOLO detection model, as well as animations of simulations and experiments are available on public OSF repository.

II. Literature Overview

The following review covers current state of research regarding UAV operations, focusing on search mission strategies and their implementations, including considerations, such as altitude control, sensing, and target detection. Conveniently, in Table I, we provide an overview of the most important features of relevant methods for search control.

A. UAV Search

To execute a search with a UAV, it is essential to navigate it effectively through the environment, which involves computing flight trajectories. Depending on the required computation time and mission demands, they can either be computed in advance [2] or calculated in real time [11]. Executing missions with planned trajectories is well-suited for known, static environments [13], [52]. While real-time computation presents significant challenges due to its complexity, it facilitates autonomous decision-making during flight [59].

Several methods for 2-D path planning are outlined in [31]. Optimization techniques, such as particle swarm optimization (PSO), are often used for trajectory generation as presented in [41] where authors compare different variations of PSO for computing three-dimensional trajectories that avoid obstacles and satisfy altitude constraints. Generated trajectories are validated in field test experiments, demonstrating their validity for real-world applications. In addition, the study in [54] presents an intelligent flight task algorithm that employs PSO, skeletonization, and B-spline curves to determine optimal UAV flight routes in complex topographies. MPC, also known as receding horizon control (RHC) can be used for trajectory planning as demonstrated by [8], [29]. It is also a commonly used technique for controlling UAVs in real time. In [45], model predictive contouring control method was utilized to perform time-optimal quadrotor flight through multiple waypoints in a closed loop. It was compared to standard MPC and expert human pilots in a real-world experiment. Real-time trajectory generation system for flight in complex environments is presented in [11] where a quadrotor equipped with onboard sensors collects environmental data and utilizes gradient information to produce smooth, collision-free trajectory through optimization. Further improvements in efficiency and convergence rate are demonstrated in [59] where the proposed method is validated through real-world experiments.

Swarm-based or multiagent approaches are frequently employed to address coverage problems, since they provide more efficient exploration compared to single agent approaches [15]. In a multiagent system, agents can operate autonomously as independent units [4], [20] or in coordinated groups with designated leaders and followers [5], [13]. Multiple agents can either share the entire search domain [19], [37] or it can be divided into distinct subdomains [3], [58]. The system can be centralized [10] or decentralized [1], each with its own benefits and drawbacks. In a decentralized system, the computational load can be distributed among agents, enhancing efficiency and scalability. However, this approach presents greater challenges in terms of information sharing, interagent communication [33], and collision avoidance. A hybrid approach is proposed in [37], where information is shared globally while each agent locally computes its own control actions.

To guarantee safety and successful execution of tasks it is important to consider the surrounding environment and avoid possible collisions. Collision avoidance can be implemented in various ways. The generation of local guidance vector fields for collision avoidance, enabling UAVs traveling at constant velocity to evade both static and moving obstacles, is presented in [35]. In [53], MPC and an artificial potential field are used to navigate fixed-wing UAVs while avoiding collisions by utilizing points of repulsive potential. In multiagent systems, avoiding collisions between agents is essential. Decentralized, asynchronous 3-D trajectory planner for multiagent systems, generating collision free routes in an environment with static and moving obstacles was presented in [55]. Collision avoidance can be implemented in practice by gathering environment information using light detection and ranging and/or sonar sensors [4], [59].

In the context of SAR, UAVs can work independently or conjunction with a team of ground searches as described in [16]. Simulations of UAV swarms aiding SAR missions during natural disasters, such as earthquakes and tsunamis, are presented in [4]. A real-world application of a UAV system for a mountain avalanche SAR scenario is showcased in [50]. Implementation and experimental comparison of different motion planners for the area coverage problem using real fixed-wing UAV is presented in [57]. It uses robot-specific controller, which takes the optimal coverage path and makes the motion plan that satisfies the dynamic constraints of the aircraft. It also compares simulation results and real-world experiments. Another practical implementation is presented in [30] where a multirotor UAV equipped with appropriate sensors is used on a field test for radioactive source search.

B. Velocity and Altitude Control

When performing search missions over relatively large areas, its important to consider the energy limitations and flight range of UAVs, especially when using multirotor UAVs. Each UAV has an optimal velocity for achieving maximum range or endurance [7]. As UAV velocity increases, it becomes more difficult to adhere to constraints like collision avoidance [26]. This underscores the importance of velocity control, especially when the UAV is unable to execute certain maneuvers at its optimal (endurance or range) velocity, allowing it to reduce flight speed during critical maneuvers. Velocity control of a large number of UAVs was implemented in [12] using mid-field game approach in order to bring the energy consumption to a minimum. An example of endurance-constrained multi-UAV planning for disaster area coverage is shown in [52].

The primary objective of UAV search missions is to locate targets by flying over them and detecting them with a sensor. The altitude of the UAV, or the distance between the sensor and the target, greatly influences detection success [16]. When using a camera sensor and a CV model, target detection is particularly affected by altitude [40]. When surveying a flat area, the UAV can fly at a fixed height, maintaining a constant distance from the ground, which can be adjusted for optimal target detection. Solving the coverage problem at fixed flight height was presented in [57], utilizing a single fixed-wing UAV. In a study carried out in [2], RHC was employed to address the cooperative coverage path planning problem while considering collision avoidance constraints for heterogeneous autonomous vehicles operating at fixed flight height.

Altitude control becomes crucial when inspecting uneven, hilly, or sloped terrain, as flying at a fixed height results in altitude changes due to variations in the terrain below. In this context, the UAV's height is relative to sea level, while altitude is relative to the terrain below the UAV. Altitude control serves two main purposes: avoiding collisions by maintaining a safe minimum distance form the terrain and ensuring the optimal altitude for target detection. Coverage task with imposed altitude goals is presented in [13], which focuses on predefined agent trajectories computed using the fast marching method with additional terrain-following functionality. Another approach for generating terrain-following trajectories is presented in [28], which employs neural networks to generate two-dimensional trajectories that meet constraints, such as drone dynamics, minimum, and maximum flight height. Strong focus on terrain following is presented in [26], where an autonomous UAV low altitude flight was executed by utilizing an onboard laser sensor for detecting the environment.

C. Ergodic Search

In search tasks, the target's location is uncertain and represented as a probability distribution over the search domain. Traditional greedy methods focus on high-probability regions, which works well if the belief is accurate but fails when the target lies in low-probability areas. When uncertainty is high, it becomes important to explore the whole distribution, since the target may be anywhere. Searching uniformly is inefficient, so ergodic methods are used instead. These methods guide the search so that, over time, the search effort matches the probability distribution: regions with higher probability are visited more often, while lower probability regions are still explored but less frequently. Formally, a trajectory $\mathbf{y}(\tau)$ is ergodic with respect to a spatial distribution $\nu(\mathbf{f})$ over domain Ω if the time spent along the trajectory matches the spatial probability distribution [47]

$$\lim_{t\to\infty} \frac{1}{t}\int_0^t f(\mathbf{y}(\tau))\,\mathrm{d}\tau = \int_\Omega f(\mathbf{p})\nu(\mathbf{p})\,\mathrm{d}\mathbf{p} \tag{1}$$

for all integrable test functions $f \in L^1(\Omega)$, $\mathbf{p} \in \Omega \in \mathbb{R}^n$. Alternatively, it can be interpreted as driving the difference between the trajectory coverage and the goal distribution to zero in the infinite-time limit [25], [36].

Several benefits of ergodic search are highlighted in [39], which shows its robustness to initial condition configurations and demonstrates that it outperforms alternative approaches in the presence of distractions. For controlling multiagent systems in ergodic exploration, several key approaches stand out: MPC, spectral multiscale coverage (SMC), and HEDAC. Many applications of ergodic control require modifications to the Sobolev-norm-based formulation presented in [36]. However, many of these algorithms, including this one, still indirectly minimize the Sobolev norm, and the resulting motion can be considered ergodic.

Research related to MPC is reviewed in [38]. It can be used to generate ergodic trajectories by incorporating an ergodicity metric into the trajectory generation process. MPC was utilized in [37] to achieve ergodic coverage alongside target localization. Although the computation is distributed among multiple agents, the method is not fully decentralized since every agent needs to communicate with the central transceiver. A similar approach is used in [1], with a modification to the control policy that allows the system to be fully decentralized.

The SMC method, introduced in [36], utilizes Sobolev-norm-based smoothing of the difference between achieved and goal coverage in order to provide directions for search agents. In multiscale adaptive search, presented in [18], SMC was combined with decision-making based on Neyman–Pearson lemma and utilized for solving a 2-D search problem. A modified SMC method (mSMC) was introduced in [21] and applied to the MH370 search simulation.

It can also be beneficial to include objectives beyond ergodicity in the path planning process, as demonstrated in [15], where a multiagent ergodic search trajectory planner is proposed that accounts for multiple objectives, including ergodicity, control energy, and interagent distance (collision avoidance). Multiple objectives in ergodic search were also considered in [44] by introducing multiple information maps. In addition, Prabhakar et al. [42] demonstrated a decentralized system for ergodic control of a swarm that can dynamically adapt to user input and environmental changes

The foundation of this article is the HEDAC algorithm, which was initially introduced in [25] and has since been further improved for search applications incorporating agent sensing and detection [19]. To enhance its capabilities, the finite element method was utilized for solving underlying heat equation in [24]. This integration enabled greater control over the agents' motion, facilitating the handling of irregularly shaped domains and interdomain obstacles without incurring additional computational costs. The versatility of the algorithm has led to its application in various contexts. For instance, it was adopted for multiagent maze exploration [10] and it can be utilized for controlling multiple UAVs performing nonuniform crop spraying [20]. The method was also applied to three-dimensional multi-UAV path planning for the vision inspection of complex structures [22]. In [9], HEDAC was employed in a whole-body ergodic exploration method for robotic manipulators. In [58], it was used to solve a coverage problem by dividing the search region into multiple subregions. In addition, in [34], the algorithm was employed to create artistic portraits by reformulating the coverage problem. In many of the mentioned papers, the superiority of HEDAC variants is demonstrated in comparisons with other ergodic control methods.

D. CV and Sensing

Modern UAVs are typically equipped with RGB cameras for target detection. Rather than manually identifying targets by reviewing images, the process can be automated using CV detection models. In recent years, the field of object detection in CV has gained significant attention, as evidenced by the growing number of publications, available training datasets, and various detection models [60].

In our research, we assume the use of a known detection model with available performance metrics and focus on how to integrate it into the search process. With this approach, the UAVs are essentially guided according to the effectiveness of the detection model, which is reflected in the accurate amount of sensing applied to the target probability distribution.

Since the performance of the detection model correlates with altitude [40], [43], [51], it is important to account for this effect within the sensing system, given the variations in altitude during flight. One possible reason the detection system performs better at certain altitude ranges is the influence of the training data [46], particularly the altitude at which the images were captured. Another factor contributing to detection model inefficiencies may be inaccurate labeling of the training data [17]. As altitude increases, the objects to be detected appear smaller, making it more difficult for humans to recognize them when labeling the training data [56].

The model we used as a reference for the detection performance is described in [40]. Although it is focused on detecting animals, this model provides detailed performance metrics across a wide altitude range, allowing us to associate detection accuracy with altitude. In addition, to target size, potential image degradation, such as blur caused by camera motion, will be considered within the scaling factor to account for its impact on detection performance [32].

III. UAV Motion Model

In order to model the motion of multiple UAVs within a three-dimensional region $\Omega_{\text{3D}} \in \mathbb{R}^3$, where each of the total n UAVs is identified by the index i, we consider the following control variables regulated in time t:

1) velocity intensity $\rho_i(t) \in [0, 1]$;
2) incline angle $\varphi_i(t) \in [\varphi_{\text{min},i}, \varphi_{\text{max},i}]$;
3) yaw angular velocity $\omega_i(t) \in [-\omega_{\text{max},i}, \omega_{\text{max},i}]$.

The incline parameter φ denotes the angle between the velocity vector, which is tangential to the UAV trajectory, and the horizontal plane. It is important to highlight, especially in the context of multirotor UAVs that the incline angle φ, representing the slope of the resulting trajectory, is distinct from the aircraft pitch. Aircraft pitch refers to the angle between the longitudinal axis of the aircraft and the horizontal plane. Furthermore, we do not take into account possible lateral motion resulting from adjustments in multirotor roll.

To authentically incorporate the distinct difference in velocity characteristics UAVs exhibit when flying horizontally, ascending, or descending, we utilize the limit velocity function $v(\varphi)$. It represents the maximal absolute velocity the UAV can achieve for a given incline. The function provides a concise description of the velocity characteristics of the UAV and can be determined experimentally for a specific aircraft.

For simplicity, we approximate the limit velocity function with an (asymmetric) ellipse constructed using three distinctive velocities: maximal horizontal velocity $v_{s,\text{max}} \equiv v(\varphi = 0)$, maximal ascending velocity $v_{z,\text{max}} \equiv v(\varphi_{\text{max}})$, and maximal descending velocity $v_{z,\text{min}} \equiv v(\varphi_{\text{min}})$, as shown in Fig. 2. Depending on the conditions and mission strategy, it may be beneficial to use the optimal range or endurance velocity, considering energy constraints of the UAVs, rather than the actual maximal velocity, which is not energy efficient [7]. If that is the case, $v_{s,\text{max}}$, $v_{z,\text{max}}$, and $v_{z,\text{min}}$ can be set up for maximal range or endurance, as

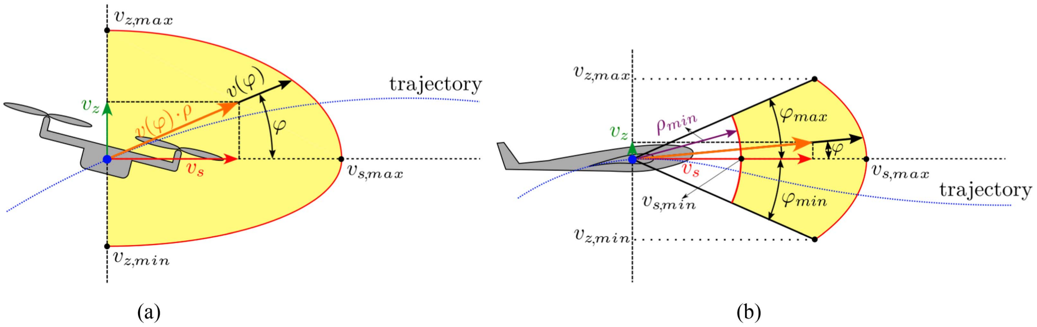


Fig. 2. Definition of velocity components and constraints for (a) multirotor and (b) fixed-wing UAVs. Vertical and horizontal velocity components are depicted by the red and green vectors, respectively, while the orange vector represents the total velocity. The red line illustrates the velocity limits corresponding to various inclines and shaded yellow area represents the feasible region for the tip of the total velocity vector, originating from the center of aircraft's mass. The feasible region for multirotor UAV allowing different dynamics of ascending and descending motions, hence, it is limited with two elliptical arcs defined with $v_{s,\text{max}}$, $v_{z,\text{min},}$ and $v_{z,\text{max},}$ characteristics of the aircraft. The motion of fixed-wing UAV is limited with a circular and an elliptical arc constructed using measurable UAV parameters $v_{s,\text{min}}$ (providing ρ_{min} limit), and $v_{s,\text{max}}$, $v_{z,\text{min}} = -v_{z,\text{max}}$, and $\varphi_{\text{min}} = -\varphi_{\text{max}}$, respectively. From the depicted image, it is evident that the fixed-wing UAV's minimum horizontal velocity can fall below the prescribed value of $v_{s,\text{min}}$. However, this discrepancy is disregarded due to the defined limits of φ, which ensure that the aircraft cannot deviate more than 4% below its minimum horizontal velocity.

opposed to setting them equal to the maximal values attainable by the UAVs.

To accurately simulate the movement of various types of UAVs, we introduce additional constraints. When modeling fixed wing UAVs we need to limit the range of φ values to avoid phenomena, such as aircraft stall. This effect usually happens when wings reach an angle of attack around $15^\circ \approx 0.26$ radians, therefore we impose φ limits in $[-0.25, 0.25]$. For multirotor UAVs, we limit the φ parameter from $\varphi_{\text{min}} = -\pi/2$, which represents vertical descent and $\varphi_{\text{max}} = \pi/2$ that corresponds to vertical ascent. By defining φ in this manner, multirotor UAVs gain the capability to execute purely vertical motion, without any horizontal movement.

The velocity intensity ρ indicates the percentage of the limit velocity that is actually utilized. In contrast to multirotor UAVs, fixed-wing UAVs inherently require a certain amount movement to generate lift. Consequently, we introduce a minimum horizontal velocity, denoted as $v_{s,\text{min}}$, to effectively restrict ρ within the range $[\rho_{\text{min}}, \rho_{\text{max}}] = [v_{s,\text{min}}/v_{s,\text{max}}, 1]$. Horizontal and vertical velocities can now be defined as

$$v_{s,i}(\rho_i, \varphi_i) = \rho_i \cdot v_i(\varphi_i) \cdot \cos(\varphi_i) \tag{2}$$

$$v_{z,i}(\rho_i, \varphi_i) = \rho_i \cdot v_i(\varphi_i) \cdot \sin(\varphi_i). \tag{3}$$

Now, the UAV motion can be defined, i.e., the trajectory $\mathbf{X}_i(t) = [x_i(t), y_i(t), z_i(t)] \in \Omega_{\text{3D}}$ that is governed by flight control parameters $\omega_i(t)$, $\rho_i(t)$, $\varphi_i(t)$ can be represented with

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = v_{s,i}(\rho_i, \varphi_i) \cdot \cos\theta_i$$

$$\frac{\mathrm{d}y_i}{\mathrm{d}t} = v_{s,i}(\rho_i, \varphi_i) \cdot \sin\theta_i$$

$$\frac{\mathrm{d}z_i}{\mathrm{d}t} = v_{z,i}(\rho_i, \varphi_i)$$

where θ is the heading angle regulated by the yaw velocity

$$\frac{\mathrm{d}\theta_i}{\mathrm{d}t} = \omega_i(t).$$

The constraints for the yaw angular velocity are defined later in Section V.

IV. Sensing Model and Search Evaluation

Though UAVs operate in three-dimensional space, their primary task is to explore and observe the terrain surface, which can be considered as a problem of two-dimensional ergodic search in the horizontal domain Ω_{2D}. The terrain is defined using a terrain height function $z_T : \Omega_{\text{2D}} \mapsto \mathbb{R}$. In order to define the sensing model, we assume that the UAV trajectory and initial probability distribution $m_0 : \Omega_{\text{2D}} \mapsto \mathbb{R}$ is known. The initial probability distribution is an assessment of spatial probability of targets (or missing persons) on a terrain surface projected to the horizontal plane Ω_{2D} at time $t = 0$.

A. Detection Model for Search Control

In order to simulate the surveillance performance of the UAV, the geometric (FOV) and sensing (detection probability) properties of the sensor need to be considered. Both of this aspects are incorporated in the detection probability density function defined in UAVs local coordinates $\in \mathbf{R}^3$ as

$$\psi_i(\mathbf{R}) = \begin{cases} \Gamma_i(\|\mathbf{R}\|), & \text{if } \mathbf{R} \in \Omega_{\text{FOV},i} \\ 0, & \text{otherwise} \end{cases}$$

where Γ is the sensing function, which encodes the instantaneous detection probability.

Surveillance is performed by a camera sensor that produces a rectangular image that envelops terrain within pyramidal FOV $\Omega_{\text{FOV}} \in \mathbb{R}$. Dependent on the UAV heading direction, FOV is defined with angles γ_1 and γ_2 between two lateral and two longitudinal sides of the pyramid, respectively. Since we are considering uneven terrain and different flight altitudes, the recorded image contains the terrain surface encapsulated by a rectangular pyramid, as shown in Fig. 3. The apex of the pyramid coincides with the camera's position that is located at $\mathbf{X}_i$ and it is the origin for local coordinates $\mathbf{R}$. The detection probability is set to 0 for any point outside the sensor's FOV, but also for

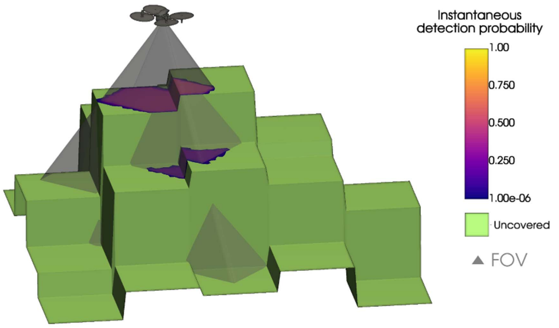


Fig. 3. Single UAV utilizes the detection probability rate function on the terrain surface within its FOV. The detection probability is determined by the distance from the camera sensor and is visually depicted as a gradient spanning from purple to yellow. The unobserved area is depicted in green, while the sensor's FOV is illustrated using a semitransparent gray pyramid. It is important to note that only the areas within the FOV pyramid that have successfully passed the ray trace check are affected by the sensor, not the entire enclosed space.

points outside the line of sight, which is determined with a ray trace check. For the area within the sensor's scope, detection probability rate is calculated based on the sensing function Γ and $\|\mathbf{R}\|$, which equals the distance between the sensor and the observed point.

To determine whether an arbitrary point $\mathbf{p} \in \Omega_{\text{2D}}$ is sensed by the sensor at $\mathbf{X}$ we need to perform a transformation to local coordinates

$$\mathbf{R} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 \\ \sin\theta_i & \cos\theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \left(\mathbf{X}_i - [\mathbf{p}_x, \mathbf{p}_y, z_T(\mathbf{p})]^T\right)$$

which includes three-dimensional translation and a rotation in horizontal plane depending on the UAV heading direction θ. Note that $\mathbf{p}$ is projected on the terrain surface using height coordinate $z_T(\mathbf{p})$.

B. CV Perspective on Human Detection

To establish a clear connection with image detection models, we developed a framework that incorporates a relevant performance metric of a chosen machine learning detection model into our sensing function, which in turn governs the behavior of the search process. In this case, the relevant metric is recall μ, which is the ratio between the number of object instances correctly identified by the detection model and the total number of object instances in the image. Recall can be interpreted as the probability of detecting an object (e.g., a missing person) since it measures the model's ability to find all positive instances when they are present in the image.

Given that we are working with a moving camera, $\mathbf{R}$ varies in time. According to the work in [19] and [27], when considering a time-varying detection rate, the probability of detection can be expressed as

$$p(\mathbf{R}) = 1 - e^{-\int_0^t \psi_i(\mathbf{R})dt}. \tag{4}$$

The detection rate function ψ, represents the sensing effect per unit of time, while CV model recall μ is typically calculated for a set of still images. To relate these values, we introduce t_{scene}, which represents the duration of the current scene, essentially, the time it takes for the scene to change significantly. The detection rate function, when applied to one domain point for the duration of t_{scene}, needs to apply the correct amount of detection probability that corresponds to the recall value. We assume the primary factor causing the scene change is camera movement, rather than object movement within the frame. Therefore, we approximate t_{scene} as the average time it takes for an arbitrary point in the domain to enter and exit the camera's FOV during the search. To calculate t_{scene}, we assume that the UAV is flying with over a flat surface at a constant altitude equal to the target search altitude h_{goal}, and we estimate the UAV's average horizontal velocity $v_{s,\text{avg}}$ during the search as 70% of its maximum horizontal velocity, $v_{s,\text{max}}$. It can be formulated as

$$t_{\text{scene},i} = \frac{2 \cdot h_{\text{goal},i} \cdot \tan\frac{\gamma_{2,i}}{2}}{v_{s,\text{avg},i}}. \tag{5}$$

Assuming the detection rate remains constant over small time intervals (on the scale of t_{scene}) and considering that recall values vary with altitude [40], we can define recall μ from (4) as

$$\mu(||\mathbf{R}||) = 1 - e^{-\Gamma(||\mathbf{R}||)\cdot t_{\text{scene}}}. \tag{6}$$

With know recall values, the sensing function can be obtained from (6) using

$$\Gamma(||\mathbf{R}||) = -\frac{\ln(1 - \mu(||\mathbf{R}||))}{t_{\text{scene}}}. \tag{7}$$

This model can be also applied for discrete sensing, when recording image snapshots at some given interval. If n_i images are recorded during a single scene, the detection rate (sensing) function must be applied for t_{scene}/n_i time to the area of the search domain covered by the images in order to achieve the same result as continuous sensing.

In addition, if objects within the camera frame are moving and the camera movement is not the only factor causing scene change, t_{scene} could be adjusted to account for that.

C. Search Effectiveness

The overall detection effect of all n UAVs involved in the search is expressed with coverage density

$$c(\mathbf{p}, t) = \sum_{i=1}^{n} \int_0^t \psi_i(\mathbf{R}(\mathbf{X_i}(t), \mathbf{p}))dt$$

which evaluates detection probability along their paths.

The probability of undetected target presence at any point $\mathbf{p}$ and time t is determined with

$$m(\mathbf{p}, t) = m_0(\mathbf{p}) \cdot e^{-c(\mathbf{p},t)}$$

by combining initial target presence probability and conducted sensing after time t at the same location. To evaluate the overall search success we integrate m over the entire domain to obtain the survey accomplishment

$$\eta(t) = 1 - \int_{\Omega_{\text{2D}}} m\,(\mathbf{p}, t)\,d\mathbf{p}.$$

Since the search time window is undefined, the minimization of m to zero is considered in infinity limit

$$\lim_{t\to\infty}\int_{\Omega_{2D}} m(\mathbf{p},t)\,\mathrm{d}\mathbf{p} = 0. \tag{8}$$

This exponential formulation of the minimizing field $m = m_0\exp(-c)$ is alternative to original formulation $m = m_0 - c$ but still aligns with asymptotic definition of ergodicity (1). This allows for use of ergodic control methods to solve the search problem [19].

The local minimization of m is achieved by a moving sensor that lowers the probability of an undetected target in the inspected region in proportion to the sensor's detection rate. Consequently, the flight control system needs to be designed to dynamically adjust parameters $\omega(t)$, $\rho(t)$, and $\varphi(t)$ with the objective in (8).

We should note that the search is obviously not infinite in provided simulation and experimental scenarios, but the total time of the search is unknown to the control algorithm.

V. Horizontal Search Control

The three-dimensional trajectory of the UAV is governed by the combined influence of horizontal and vertical movements, allowing us to calculate them separately. We begin by computing the UAV's horizontal path, which represents its movement within the search domain Ω_{2D}. Following that, we perform altitude and velocity optimization to determine the final trajectory and complete control of UAVs as presented in the next section.

A. Ergodic Control

Effective horizontal movement is crucial for a successful search, as it maximizes the UAVs' search performance according to the target probability distribution. We rely on ergodic control designed to satisfy the objective given by (8) and implemented using the HEDAC method [19]. The method treats the probability of undetected targets as a heat source and uses the heat equation to generate a temperature field whose gradient directs exploration agents. The method essentially minimizes the source term of the stationary heat equation by cooling the heat source in the areas the agents explore. To provide a more general perspective, the temperature is interpreted as a potential field $u(\mathbf{p},t)$ that attracts search agents. It is computed by solving the partial differential equation

$$\alpha\cdot\Delta u(\mathbf{p},t) = \beta\cdot u(\mathbf{p},t) - m(\mathbf{p},t)$$

using the boundary condition

$$\frac{\partial u}{\partial \mathbf{n}} = 0$$

where $\mathbf{n}$ denotes the outward normal to the domain boundary $\partial\Omega_{\text{2D}}$. HEDAC parameters $\alpha > 0$ and $\beta > 0$ can be adjusted to achieve different search behavior. α governs the smoothness of the probability field and, therefore, dictates whether the search focus is local or global, while β serves as a numerical stability factor and has a weaker effect on the search performance.

To ascertain the travel direction, we introduce the vector field

$$\mathbf{u}(\mathbf{p},t) = \frac{\nabla u(\mathbf{p},t)}{|\nabla u(\mathbf{p},t)|}$$

which equals the normalized gradient of the potential field u.

The angle between the current direction vector $\mathbf{v}_i(t)$ and vector $\mathbf{u}(x_i, y_i, t)$ provides a change of heading that should be realized in the control step Δt. To alter the heading, the UAV pivots around its yaw axis, utilizing yaw angular velocity

$$\omega_i = \frac{1}{\Delta t}\arccos\left(\frac{\mathbf{v}_i(t)\cdot\mathbf{u}(x_i,y_i,t)}{|\mathbf{v}_i(t)|}\right).$$

Each UAV is characterized by maximal achievable angular velocity denoted as ω_{lim}, representing its technical limitation. Therefore, we require that $|\omega| \leq \omega_{\text{lim}}$.

Horizontal velocity is determined by testing various velocity values along a specific path segment (described later in Section VI-A). To guarantee the feasibility of movement along the path regardless of the velocity used, we enforce a path curvature constraint defined by the minimal turning radius R_{min}. The highest ω value is attained when moving at maximum horizontal velocity along a path characterized by a curvature of R_{min}.

Considering the requirements for ω, we define the maximal yaw angular velocity value ω_{max} and impose a constraint

$$|\omega| \leq \omega_{\text{max}} = \min\left(\omega_{\text{lim}}, \frac{v_{s,\text{max}}}{R_{\text{min}}}\right). \tag{9}$$

B. Collision Avoidance

The final step is to check and adapt ω values to ensure a collision free path. The procedure is analogue to previous work [24] with the only difference being that the collision is checked with bounding circles B^+ and B^- instead of clearing circles C^+ and C^-, which were previously used. Since UAV velocities are variable and unknown at this stage, the positions of the clearing circles can differ based on different horizontal velocities. Consequently, it is necessary to examine collisions for all potential locations of the clearing circles, therefore, we use the bounding circle which contains all possible clearing circles, as shown in Fig. 4. In summary, we first check all UAVs for collision using their initial ω_i. Only if there is a collision, ω_i values are optimized in range $[-\omega_{\text{max},i}, \omega_{\text{max},i}]$. Bounds of the optimization range represent possible escape yaw angular velocity $\omega_{\text{esc},i}$ values, which are utilized during the collision avoidance (or escape) maneuver while the path executed using one of the bounds has to be collision free at any given moment. The optimization goal is to acquire ω_i values that are closest to the initial ω_i values (computed from $\mathbf{v}_i(t)$ and $\mathbf{u}(x_i, y_i, t)$) but do not produce collision.

After the final ω_i values are attained, we analyze whether the right or left side of the UAV is available for the escape maneuver (whether B^+ or B^- is collision free) and determine $\omega_{\text{esc},i}$ values. Escape yaw angular velocity can be either $-\omega_{\text{max},i}$ or $\omega_{\text{max},i}$, so we choose the value closer to final ω_i that yields a collision free route.

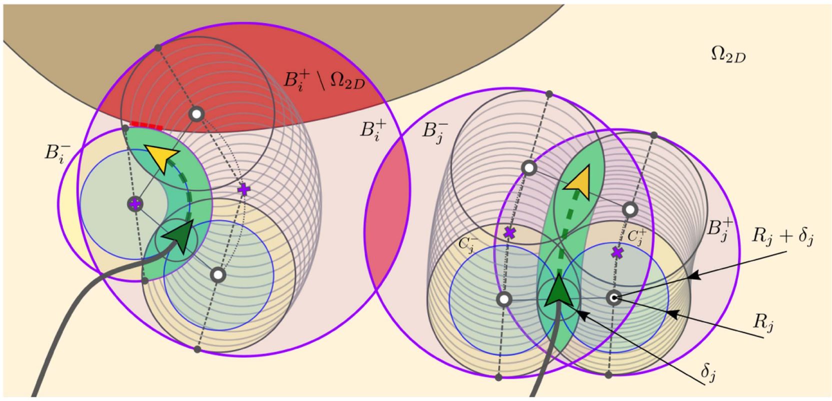


Fig. 4. Two UAVs performing a collision check using bounding circles B^+ and B^- shown in purple. The current aircraft position is indicated by the green arrow, while the yellow arrow represents the next position for $v_s = v_{s,\max}$ and ω that is currently being tested for collision. The blue circles depict radical "escape routes" that are executed using $\pm\omega_{\max}$, while the yellow shaded circles represent previously used clearing circles C^+ and C^-. The gray circles illustrate a subset of potential clearing circle positions corresponding to various horizontal velocities. The clearing circles for $v_s = 0$ (current position) and $v_s = v_{s,\max}$ are internally tangent to the bounding circle which shares its center (shown as purple "+") with the clearing circle for $v_s = v_{s,\max}/2$. The area shaded in green should always be collision free to satisfy minimum clearance δ constraint.

VI. Velocity and Altitude Control

UAV altitude control aims to maintain a prescribed goal distance above uneven terrain to ensure adequate area coverage and detection rates, while complying with a minimum-altitude constraint. The feasibility of the 3-D trajectory requires compliance with UAV motion constraints. Complementing altitude adjustments, velocity control ensures feasible UAV flight.

Velocity and altitude control are achieved using MPC technique, which determines ρ and φ regimes that produce an optimal predicted flight for time window $[t, t + \tau_{\max}]$. By employing the horizontal control method discussed in the preceding section, we generate a two-dimensional predicted path (projected to the horizontal plane) and acquire the terrain's elevation profile along this trajectory. To achieve predefined goals and adhere to constraints, an optimal flight path is established by adjusting the ρ and φ regimes within the time window of $[t, t + \tau_{\max}]$.

A. Trial Trajectories and Trial Control Functions

First, we introduce a predicted path in the horizontal plane, achievable using maximum velocity ($\rho = 1$) and without vertical movement ($\varphi = 0$), which is defined with

$$\frac{\mathrm{d}\bar{x}_i}{\mathrm{d}\tau} = v_{s,\max,i} \cdot \cos\bar{\theta}_i$$
$$\frac{\mathrm{d}\bar{y}_i}{\mathrm{d}\tau} = v_{s,\max,i} \cdot \sin\bar{\theta}_i$$
$$\frac{\mathrm{d}\bar{\theta}_i}{\mathrm{d}\tau} = \bar{\omega}_i.$$

The UAV trial yaw angular velocity is calculated from unchanging (in time window $[0, \tau_{\max}]$) potential field u, obtained at time t, using

$$\bar{\omega}_i(\tau) = \frac{1}{\Delta t} \arccos\left(\frac{\bar{\mathbf{v}}_i(\tau) \cdot \mathbf{u}(\bar{x}_i(\tau), \bar{y}_i(\tau), t)}{|\bar{\mathbf{v}}_i(\tau)|}\right).$$

The calculated $\bar{\omega}_i(\tau)$ is verified to ensure constraint (9) and collision avoidance for $[0, \Delta t]$.

For convenience, a path length function $\bar{s}(\tau)$ can be easily obtained from

$$\frac{\mathrm{d}\bar{s}}{\mathrm{d}\tau} = v_{s,\max,i}$$

as $\bar{s} = v_{s,\max} \cdot \tau$ that allows the trajectory parametrization $\bar{x}_i(s)$ and $\bar{y}_i(s)$.

According to $\tilde{\rho}$ and $\tilde{\varphi}$, we define trial trajectory length function $\tilde{s}$ from

$$\frac{\mathrm{d}\tilde{s}_i}{\mathrm{d}\tau} = v_{s,i}\left(\tilde{\rho}_i(\tau), \tilde{\varphi}_i(\tau)\right)$$

and utilize it along with above mentioned length parametrization of $\bar{x}$ and $\bar{y}$ to determine a trial trajectory in time window $[0, \tau_{\max}]$ as

$$\tilde{x}_i(\tau) = \bar{x}_i\left(\tilde{s}_i(\tau)\right)$$
$$\tilde{y}_i(\tau) = \bar{y}_i\left(\tilde{s}_i(\tau)\right).$$

Note that UAV trial trajectory, in general, does not pass the entire available predicted path in $[0, \tau_{\max}]$ due to possible partial use of velocity intensity ($\rho(\tau) < 1$) or possible ascending or descending trial flight regimes ($\varphi(\tau) \neq 0$). The trial yaw angular velocity can be determined using the parametrization $\bar{\omega}_i(s)$ as

$$\tilde{\omega}_i(\tau) = \frac{v_{s,i}(\tilde{\rho}_i, \tilde{\varphi}_i)}{v_{s,\max,i}} \cdot \bar{\omega}_i(s(\tau))$$

which ensures that the trial trajectory passes over predicted path and preserves the same curvature, but not producing the same length, regardless of $\tilde{\rho}$ and $\tilde{\varphi}$ being employed.

The vertical component of a trial trajectory is also defined using the trial regimes $\tilde{\rho}$ and $\tilde{\varphi}$ as

$$\frac{\mathrm{d}\tilde{z}_i}{\mathrm{d}\tau} = v_{z,i}(\tilde{\rho}_i, \tilde{\varphi}_i).$$

Trial velocity intensity and incline angle function, $\tilde{\rho}(\tau)$ and $\tilde{\varphi}(\tau)$, respectively, are subjected to the optimization in the MPC framework, unlike trial yaw angular velocity $\tilde{\omega}$ that is obtained from the gradient of the current potential field and corrected according to $\tilde{\rho}(\tau)$ and $\tilde{\varphi}(\tau)$.

B. Optimization Problem Formulation

We seek for an optimal trial flight regime in time window $[t, t + \tau_{\max}]$ in order to determine velocity and altitude control regulated by ρ and φ at time t. In order to assemble the optimization problem, $\tilde{\rho}$ and $\tilde{\varphi}$ need to be parameterized. These functions are obtained through quadratic polynomial interpolation defined with three specific points: $\tau_0 = 0$ (known current UAV state at time t), $\tau_1 = \frac{\tau_{\max}}{2}$, and $\tau_2 = \tau_{\max}$. While the trial control functions

$$\tilde{\rho}(\tau_0) = \rho(t)$$

$$\tilde{\varphi}(\tau_0) = \varphi(t)$$

are known at the (current) time t, we define the optimization vector $\mathbf{W} \in \mathbb{R}^4$ as values of $\tilde{\rho}$ and $\tilde{\varphi}$ at τ_1 and τ_2, which is expressed as

$$\mathbf{W}_i \equiv [\tilde{\rho}_i(\tau_1),\ \tilde{\rho}_i(\tau_2),\ \tilde{\varphi}_i(\tau_1),\ \tilde{\varphi}_i(\tau_2)]^T .$$

Since the trial control functions are functions of relative time τ, but in terms of the optimization, they are also defined by (functions of) the optimization vector $\mathbf{W}_i$, for the sake of simplicity and readability, we introduce the notation

$$(\cdot)\big\lfloor_{\mathbf{W}_i}(\tau)$$

which means that any trial function $(\cdot)$ is function of relative time τ and optimization vector $\mathbf{W}_i$.

We set two objectives for the flight regime: maximizing UAV velocity and providing the UAV altitude closest possible to the goal search altitude h_{goal}. Maximizing UAV velocity eventually leads to shorter inspection time, and this is accomplished with introduction of the minimization objective

$$o_{v,i}(\mathbf{W}_i) = 1 - \frac{1}{\tau_{\max}} \int_0^{\tau_{\max}} \tilde{\rho}_i\big\lfloor_{\mathbf{W}_i}(\tau)\,\mathrm{d}\tau. \tag{10}$$

To define second objective we introduce the relative terrain height function

$$\tilde{z}_{T,i}\big\lfloor_{\mathbf{W}_i}(\tau) = z_T\left(\tilde{x}_i\big\lfloor_{\mathbf{W}_i}(\tau),\ \tilde{y}_i\big\lfloor_{\mathbf{W}_i}(\tau)\right).$$

The proximity of the trial altitude function $\tilde{z} - \tilde{z}_T$ to the goal altitude h_{goal} is determined by the altitude objective

$$o_{h,i}(\mathbf{W}_i) = \frac{\int_0^{\tau_{\max}} \left|\tilde{z}_i\big\lfloor_{\mathbf{W}_i}(\tau) - \tilde{z}_{T,i}\big\lfloor_{\mathbf{W}_i}(\tau) - h_{\text{goal},i}\right| \mathrm{d}\tau}{h_{\text{goal},i}\cdot\tau_{\max}}. \tag{11}$$

Both velocity and altitude objectives are normed by their definitions (10) and (11), respectively. Hence, the two objectives are combined by addition, without using weight factors since they have the same significance

$$o_i(\mathbf{W}_i) = o_{v,i}(\mathbf{W}_i) + o_{h,i}(\mathbf{W}_i).$$

We define the constraints of a trial trajectory in the nonequality form $c(\mathbf{W}) \le 0$. In order to guarantee that the UAV cannot go below the specified minimum altitude $h_{\min}$, we impose a minimum altitude optimization constraint

$$\tilde{z}_i\big\lfloor_{\mathbf{W}_i}(\tau) - \tilde{z}_{T,i}\big\lfloor_{\mathbf{W}_i}(\tau) \ge h_{\min,i}. \tag{12}$$

It needs to be viable for every τ, which is validated using

$$c_{h,i}(\mathbf{W}_i) = \frac{\int_0^{\tau_{\max}} \max\left\{h_{\min,i} - \tilde{z}_i\big\lfloor_{\mathbf{W}_i}(\tau) + \tilde{z}_{T,i}\big\lfloor_{\mathbf{W}_i}(\tau),\ 0\right\} \mathrm{d}\tau}{h_{\min,i}\cdot\tau_{\max}}.$$

To ensure compliance with the velocity and acceleration specification, their constraints need to be satisfied, as defined with

$$v_{s,\min,i} \le \tilde{v}_{s,i}\big\lfloor_{\mathbf{W}_i}(\tau) \le v_{s,\max,i} \tag{13}$$

$$v_{z,\min,i} \le \tilde{v}_{z,i}\big\lfloor_{\mathbf{W}_i}(\tau) \le v_{z,\max,i} \tag{14}$$

where $\tilde{v}_{s,i}\big\lfloor_{\mathbf{W}_i}(\tau)$ and $\tilde{v}_{z,i}\big\lfloor_{\mathbf{W}_i}(\tau)$ are horizontal and vertical trial velocity functions obtained with (2) and (3), respectively. Evaluation of horizontal velocity constraints is performed using

$$c_{vs\min,i}(\mathbf{W}_i) = \int_0^{\tau_{\max}} \frac{\max\left\{v_{s,\min,i} - \tilde{v}_{s,i}\big\lfloor_{\mathbf{W}_i}(\tau),\ 0\right\} \mathrm{d}\tau}{v_{s,\min,i}\cdot\tau_{\max}}$$

$$c_{vs\max,i}(\mathbf{W}_i) = \int_0^{\tau_{\max}} \frac{\max\left\{\tilde{v}_{s,i}\big\lfloor_{\mathbf{W}_i}(\tau) - v_{s,\max,i},\ 0\right\} \mathrm{d}\tau}{v_{s,\max,i}\cdot\tau_{\max}}$$

and analogously for vertical velocity constraints using

$$c_{vz\min,i}(\mathbf{W}_i) = \int_0^{\tau_{\max}} \frac{\max\left\{v_{z,\min,i} - \tilde{v}_{z,i}\big\lfloor_{\mathbf{W}_i}(\tau),\ 0\right\} \mathrm{d}\tau}{v_{z,\min,i}\cdot\tau_{\max}}$$

$$c_{vz\max,i}(\mathbf{W}_i) = \int_0^{\tau_{\max}} \frac{\max\left\{\tilde{v}_{z,i}\big\lfloor_{\mathbf{W}_i}(\tau) - v_{z,\max,i},\ 0\right\} \mathrm{d}\tau}{v_{z,\max,i}\cdot\tau_{\max}}.$$

The UAV acceleration is also considered in trial trajectories. Horizontal and vertical accelerations depend on properties and technical abilities of the UAV, such as vehicle mass and inertia, and achievable upthrust and pitch in low-level UAV control. The most simple acceleration and deceleration constraints are imposed to horizontal and vertical UAV motion using

$$a_{s,\min,i} \le \tilde{a}_{s,i}\big\lfloor_{\mathbf{W}_i}(\tau) \le a_{s,\max,i} \tag{15}$$

$$a_{z,\min,i} \le \tilde{a}_{z,i}\big\lfloor_{\mathbf{W}_i}(\tau) \le a_{z,\max,i}. \tag{16}$$

Acceleration trial functions $\tilde{a}_{s,i}\big\lfloor_{\mathbf{W}_i}(\tau)$ and $\tilde{a}_{z,i}\big\lfloor_{\mathbf{W}_i}(\tau)$ are calculated via numerical differentiation of corresponding horizontal and vertical velocities $\tilde{v}_{s,i}\big\lfloor_{\mathbf{W}_i}(\tau)$ and $\tilde{v}_{z,i}\big\lfloor_{\mathbf{W}_i}(\tau)$, respectively.

The horizontal acceleration and deceleration constraints are evaluated with

$$c_{as\min,i}(\mathbf{W}_i) = \int_0^{\tau_{\max}} \frac{\max\left\{a_{s,\min,i} - \tilde{a}_{s,i}\big\lfloor_{\mathbf{W}_i}(\tau),\ 0\right\} \mathrm{d}\tau}{a_{s,\min,i}\cdot\tau_{\max}}$$

$$c_{a s\max,i}(\mathbf{W}_i) = \int_0^{\tau_{\max}} \frac{\max\left\{\tilde{a}_{s,i}\big\lfloor_{\mathbf{W}_i}(\tau) - a_{s,\max,i},\, 0\right\} \mathrm{d}\tau}{a_{s,\max,i} \cdot \tau_{\max}}$$

and the vertical acceleration and deceleration constraints are evaluated analogously with

$$c_{a z\min,i}(\mathbf{W}_i) = \int_0^{\tau_{\max}} \frac{\max\left\{a_{z,\min,i} - \tilde{a}_{z,i}\big\lfloor_{\mathbf{W}_i}(\tau),\, 0\right\} \mathrm{d}\tau}{a_{z,\min,i} \cdot \tau_{\max}}$$

$$c_{a z\max,i}(\mathbf{W}_i) = \int_0^{\tau_{\max}} \frac{\max\left\{\tilde{a}_{z,i}\big\lfloor_{\mathbf{W}_i}(\tau) - a_{z,\max,i},\, 0\right\} \mathrm{d}\tau}{a_{z,\max,i} \cdot \tau_{\max}}.$$

C. Solving MPC Optimization

Each optimization is started with a specific initial optimization vector $\mathbf{W}_0$, which is chosen between

$$\mathbf{W}_{0a} = (1, 1, 0, 0)$$
$$\mathbf{W}_{0b} = (1, 1, 1, 1)$$
$$\mathbf{W}_{0c} = (0.5, 0.5, 1, 1).$$

We check possible initial vectors in order from a to c and the first one that returns a feasible solution is selected as $\mathbf{W}_0$. $\mathbf{W}_{0a}$ encourages the utilization of maximal forward velocity in the horizontal search direction, while $\mathbf{W}_{0b}$ and $\mathbf{W}_{0c}$ advocate increase in altitude with different velocity intensity. Fig. 5 displays an illustration of the optimization process and possible initial optimization vectors. If there is no feasible solution, UAV skips the optimization process and starts the collision avoidance procedure (described as follows).

Optimization is performed using a modification of the GPS-MADS method [6], called multiscale grid search implemented in *Indago* Python module [23]. The number of optimization iterations is set to 30 while the stopping criteria are regulated by setting the maximum number of stalled iterations to 10 and target fitness to 10^{-3}. Using the optimized values $\mathbf{W}_{\text{opt}}$ we obtain the candidate control values for velocity intensity

$$\rho_{\text{opt},i} = \tilde{\rho}_i\big\lfloor_{\mathbf{W}_{\text{opt},i}}(\Delta t)$$

incline

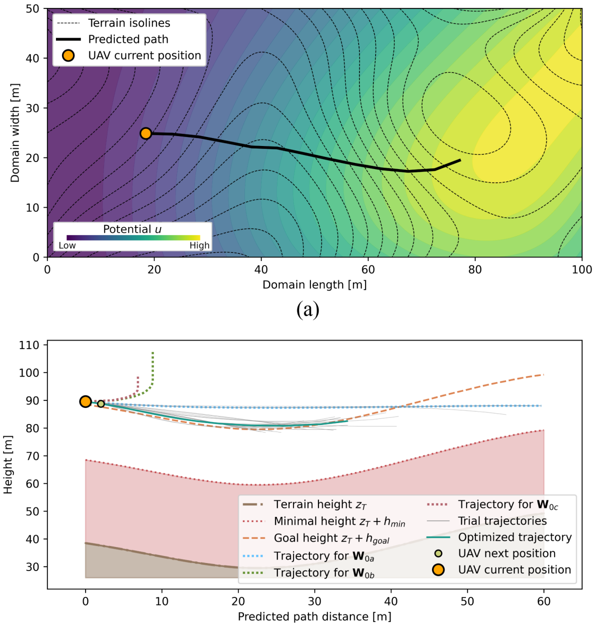
$$\varphi_{\text{opt},i} = \tilde{\varphi}_i\big\lfloor_{\mathbf{W}_{\text{opt},i}}(\Delta t)$$

and yaw angular velocity

$$\omega_{\text{opt},i} = \tilde{\omega}_i\big\lfloor_{\mathbf{W}_{\text{opt},i}}(\Delta t).$$

D. Validating Escape Maneuver and Establishing Control Parameters

Although optimal flight control parameters $\rho_{\text{opt},i}$, $\varphi_{\text{opt},i}$, and $\omega_{\text{opt},i}$ are determined in MPC optimization, they do not guarantee a possibility of collision-free trajectory (realized with escape maneuver) for the next time step. Due to high computational demands, considering this constraint in MPC optimization is not feasible for achieving real-time UAV control. Hence, the obtained MPC flight control parameters need to be verified and corrected if needed before they are enforced to UAVs.



(a)

(b)

Fig. 5. Visual representation of path prediction and the optimization process. (a) Predicted path is formed starting from the current position of the UAV, simulating maximal achievable horizontal movement for the number of prediction steps n_{pts}, along the gradient of the potential field u. Terrain height z_T is then probed along the length of the predicted path and used to construct minimal $(z_T + h_{\min})$ and goal $(z_T + h_{\text{goal}})$ height curves. (b) Optimization process initiates at the first feasible regime among $\mathbf{W}_{0a}$, $\mathbf{W}_{0b}$, $\mathbf{W}_{0c}$, following the order as listed. It balances between velocity maximization and goal altitude adherence objectives to generate the optimal flight regime for n_{pts} time steps while taking velocity, acceleration, and minimal height constraints into consideration. The UAV then utilizes the optimized flight regime for Δts, consequently, it moves to the new position (marked with a lime green colored dot) where the complete process repeats.

The escape maneuver is a specific UAV motion that aims to provide following flight features.

1) Escape maneuver can be started from any starting time given any locations, state, and control parameters of the UAV. Here, we consider UAVs at time $t + \Delta t$ with optimal control parameters and, if needed, at current time t with current control parameters.
2) The projection of the escape trajectory to the horizontal plane is circular arc (extendable to full circle) with radius equal to $\pm R_{\min,i}$, while adjusting $\omega_{\text{esc},i}$. This ensures the resulting arc overlaps with one of the escape circles defined in Section V-B.
3) The UAV decelerate in the most agile feasible regime, i.e., $\rho_{\text{em},i} \to \rho_{\min,i}$.
4) The UAV ascends in the most agile feasible regime, i.e., $\varphi_{\text{em},i} \to \varphi_{\max,i}$.
5) Escape route feasibility is determined by before-mentioned constraints: minimal altitude (12), horizontal and vertical velocity (13) and (14), horizontal, and vertical acceleration (15) and (16).
6) The escape motion is considered until the horizontal velocity drops to zero or until full escape circle is passed.

TABLE II
TEST CASES PARAMETERS AND DATA

Test case parameters	Plastic world	Mt. Vesuvius	Star dunes	Učka Mission 1 / Učka Mission 2	Units
Domain size	0.72	7.44	7.5	0.95	km^2
Number of mesh nodes	8380	21 825	21 946	11 505	-
Number of mesh elements	16 300	43 098	43 340	22 658	-
Elevation difference	421	608	221.4	108.8	m
Alpha α	1000	2300	2900	5000	-
Beta β	0.1	1	1	0.1	-
Time step Δt	1	2	2	3	s
Search duration	30	60	60	25.26/19.36	min
Number of UAVs A	3	3	0	0	-
Number of UAVs B	0	2	0	0	-
Number of UAVs C	0	0	2	0	-
Number of UAVs M210	0	0	0	2/1	-
Number of UAVs Mavic 2	0	0	0	0/1	-

TABLE III
MOTION, VISION/SENSING, AND CONTROL UAV PARAMETERS USED IN SIMULATED SEARCH SCENARIOS

UAV parameters	UAV A/ UAV B	UAV C	M210/ Mavic 2	Units
Type	Multi-rotor	Fixed-Wing	Multi-rotor	-
Min turning radius R_{min}	25	100	30	m
Min clearance distance δ	7	60	50	m
Min search altitude h_{min}	30	100	40	m
Goal search altitude h_{goal}	50/100	300	60	m
Max horizontal velocity $v_{s,\text{max}}$	10	15	8	m/s
Min horizontal velocity $v_{s,\text{min}}$	0	5	0	m/s
Max ascending velocity $v_{z,\text{max}}$	5	1.2	5/3	m/s
Max descending velocity $v_{z,\text{min}}$	-3	-1.2	-3/-2	m/s
Max horizontal acceleration $a_{s,\text{max}}$	2	2	2	m/s^2
Min horizontal acceleration $a_{s,\text{min}}$	-3.6	-2	-3.6	m/s^2
Max vertical acceleration $a_{z,\text{max}}$	2.8	1	2.8	m/s^2
Min vertical acceleration $a_{z,\text{min}}$	-2	-1	-2	m/s^2
Min incline φ_{min}	-90	13.5	-90	°
Max incline φ_{max}	90	13.5	90	°
Camera FOV γ_1	62.8/33.94	23	64.7/72.5	°
Camera FOV γ_2	37.9/19.48	13.06	39.2/57.58	°
Sensing function Γ	Γ_A/Γ_B	Γ_C	$\Gamma_{M210}/$ Γ_{Mavic2}	-
Predicted path length n_{pts}	25	30	5	time steps

If a feasible escape maneuver is achievable for optimal control parameters at time $t + \Delta t$, the optimal control parameters

$$\rho_i(t + \Delta t) = \rho_{\text{opt},i}$$
$$\varphi_i(t + \Delta t) = \varphi_{\text{opt},i}$$
$$\omega_i(t + \Delta t) = \omega_{\text{opt},i}$$

can be applied to UAV.

If there is no feasible escape maneuver at time $t + \Delta t$, indicating currently obtained control parameters are not valid, the UAV immediately starts the escape maneuver using parameters

$$\rho_i(t + \Delta t) = \rho_{\text{em},i}(t + \Delta t)$$
$$\varphi_i(t + \Delta t) = \varphi_{\text{em},i}(t + \Delta t)$$
$$\omega_i(t + \Delta t) = \omega_{\text{em},i}(t + \Delta t)$$

obtained at $t + \Delta t$ (for verified escape maneuver simulated from time t).

VII. RESULTS

In this section, the algorithm was evaluated in three simulated search scenarios, with a detailed description of the UAVs used, their parameters, and domain characteristics provided. In addition, a framework was established to enable the integration of a CV detection model into the sensing function utilized by the proposed method.

Results and animations for all test cases are available on the open science framework repository: https://osf.io/t947u/.

A. Tested Search Scenarios

For conducting search simulations, we generated two-dimensional meshes with triangular elements. Each node within the mesh includes terrain elevation data. Domain and mesh details can be found in Table II along with method parameters and number of specific UAVs used.

In order to test the method, we designed three test cases with different terrain complexity and size. Each case uses a specific configuration of different UAVs, described in Table III.

Multirotor UAVs (A, B) are modeled based on technical specifications and tests performed with *DJI Matrice 210 V2* aircraft. We carried out an experiment to confirm or conclude maximal horizontal, ascending, and descending velocity and minimal and maximal value of horizontal and vertical acceleration, which we used. It is important to note that despite both UAVs being modeled similarly, they were regulated to conduct searches at varying altitudes according to their technical characteristics. The deliberate difference in altitude between A, employing a goal altitude of 50 m above ground level and a camera with no zoom, and B, utilizing a goal altitude of 100 m while incorporating and simulating the use of a camera with 2 × optical zoom, leads to a unique FOV and sensing function combination for each UAV type. Characteristics of the fixed-wing UAV (C) are roughly estimated to portray a realistic aircraft, which is significantly more constrained compared to multirotor UAVs. Its design targets the use in large domains with lower incline values and smoother terrain. It uses the same camera sensor as the multirotor UAVs, utilizing 3× optical zoom, with adjustments the FOV and sensing function.

B. Adjusting Sensing Functions

For each camera sensor configuration that was used, we constructed an accompanying sensing function referring to YOLOv4 detection model data provided in [40]. The referenced work focuses on animal (large mammal) detection in images captured by a UAV camera and provides a range of detection model metrics, including recall, in relation to the altitude at which the images were taken.

CV detection systems are typically effective at detecting humans in UAV-captured images. However, their performance often declines in real-world applications due to various technical and environmental factors. Motion or focus blur, poor lighting, weather-related haze, variations in surrounding vegetation, and other natural elements impact the detection rate. We utilized the recall values from [40], which were obtained for large mammals,

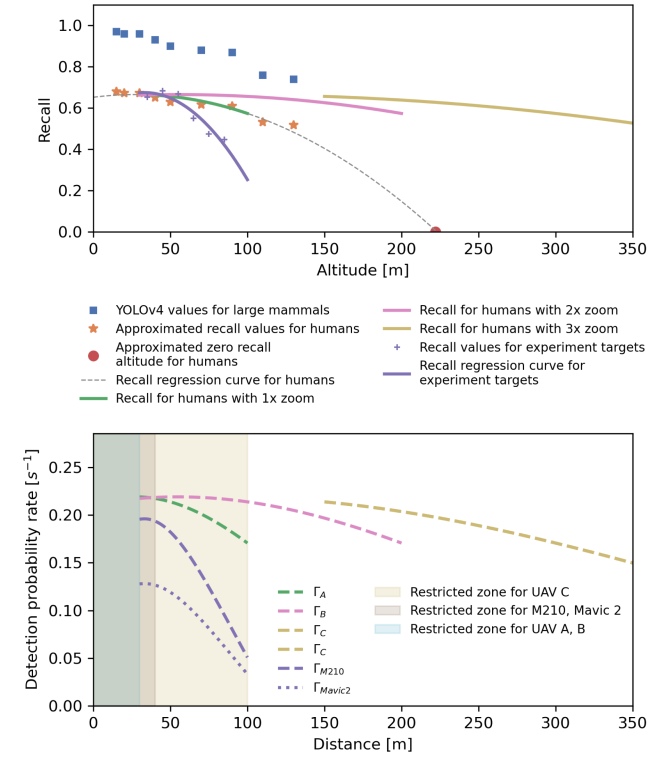


Fig. 6. This illustration depicts the relationship between altitude and recall, along with the corresponding sensing functions utilized in the test cases and experimental validation. Recall-altitude values for large mammals, as reported in [40], are represented by blue squares, while scaled values for humans are indicated by orange stars, and values for experiment targets are depicted with purple pluses. The approximate altitude at which human targets are undetectable is marked with a red circle. Computed recall-altitude curves for humans at various zoom levels, as well as for experiment targets, are shown as solid lines in different colors. The corresponding sensing functions derived from recall-altitude curves are represented by dashed/dotted lines in matching colors. When evaluating the sensing functions, we focus on the distance between the sensor and the observed point instead of the altitude at which the image was captured, ensuring accurate sensing even in irregular terrain.

and approximately reduced them by 30% to account for human detection and other aggravating factors. Both the original and adjusted recall-altitude values are shown in Fig. 6. We introduced an additional altitude point where the recall is assumed 0, indicating that detection is not possible. Based on findings from [56], if a person in the image is smaller than 5 × 5 pixels, it will be considered undetectable. This nondetection is attributed to the model's inability to recognize such small objects, which could be due to the difficulty majority of people have in identifying small objects (people) in images [56], leading to a lack of labeling in the training data for the CV detection model.

To determine the altitude where recall drops to 0, we assume an average human occupies an area of approximately 0.5 × 0.25 m when viewed from above. We then calculate the altitude at which the person would occupy 25 pixels in a 4K resolution and 16:9 aspect ratio image. For simplicity, we use a total of 25 pixels instead of a 5 × 5 pixel area, as this approach simplifies the calculation by not accounting for the person's orientation within the image. Considering the FOV angles for the UAV A, as defined in Table III, equipped with a 1 × zoom camera (no zoom), we determined that detection is not possible when the image is captured from an altitude above 222 m.

Since we are implementing altitude control, the zero-recall point is not highly relevant in our case, as it falls outside our target sensing altitude range. However, it is important to define this point for comparison purposes, as we are evaluating the efficiency of our method against one that lacks altitude control but implements the same sensing method.

To finally acquire the sensing function, we applied (7), where the recall function $\mu(||\mathbf{R}||)$ was obtained through quadratic interpolation of recall-altitude points. We assumed that $||\mathbf{R}|$ is approximately equal to UAV's altitude. Different optical zoom values can be also used, but this requires adjusting both the recall function and the camera's FOV. When referring to "zoom," we specifically mean optical zoom, as it magnifies the image using the camera's lens without compromising resolution, unlike digital zoom. Optical zoom preserves image quality and has a similar effect to changing altitude, as both modify the FOV and the level of detail captured in the image. For a zoom level of z, the adjusted recall function becomes $\mu(||\mathbf{R}||/z)$, and the corresponding FOV angles are calculated using

$$\gamma_z = 2 \cdot \arctan\left(\frac{\tan\frac{\gamma}{2}}{z}\right)$$

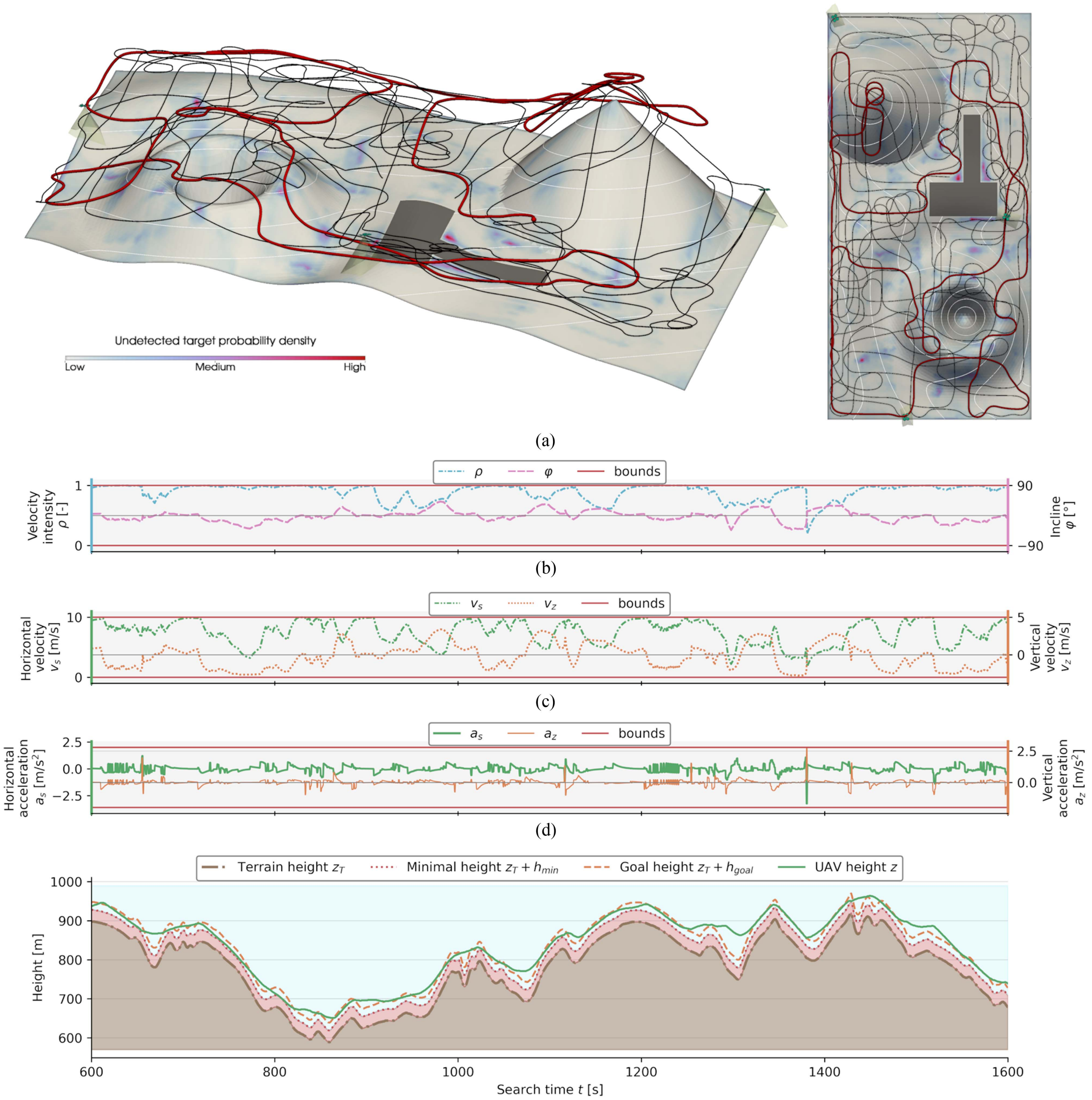
where γ represents an FOV angle with 1 × zoom and γ_z represents the corresponding FOV angle with zoom level z. Sensing functions for all UAVs are presented in Fig. 6, along with interpolated recall-altitude curves for humans at different optical zoom levels.

C. Simulations Setup

Simulations were performed on a machine with 3.7 GHz base CPU clock. Array manipulation along with other numerical tools form *NumPy* and interpolation tools from *SciPy* were used. Meshes were generated using *Gmsh* [14], while *Netgen/NGSolve* [48], [49] was employed to solve the finite element system for obtaining the potential field. Flight visualizations and analysis plots are made using *Matplotlib* and *PyVista*.

D. Plastic World

The first test is performed on a synthetically generated domain called "plastic world." The enveloped terrain is purposely constructed to be simple and exaggerated, yet providing specific features of natural terrain, in order to demonstrate the robustness of the algorithm. Search is conducted collaboratively using three multirotor UAVs. The search domain and accompanying UAV trajectories after survey completion, along with the analysis of control parameters and the associated trajectory performed by one UAV during a 1000 s time window, are presented in Fig. 7. After evaluating the trajectory, it is evident that all control and flight parameter constraints have been fulfilled. The survey accomplishment metric reached 98% within 30 min of the search simulation, as illustrated in Fig. 11. The rectangular no-fly zone is avoided by all UAVs respecting assigned clearance distance



(a)

(b)

(c)

(d)

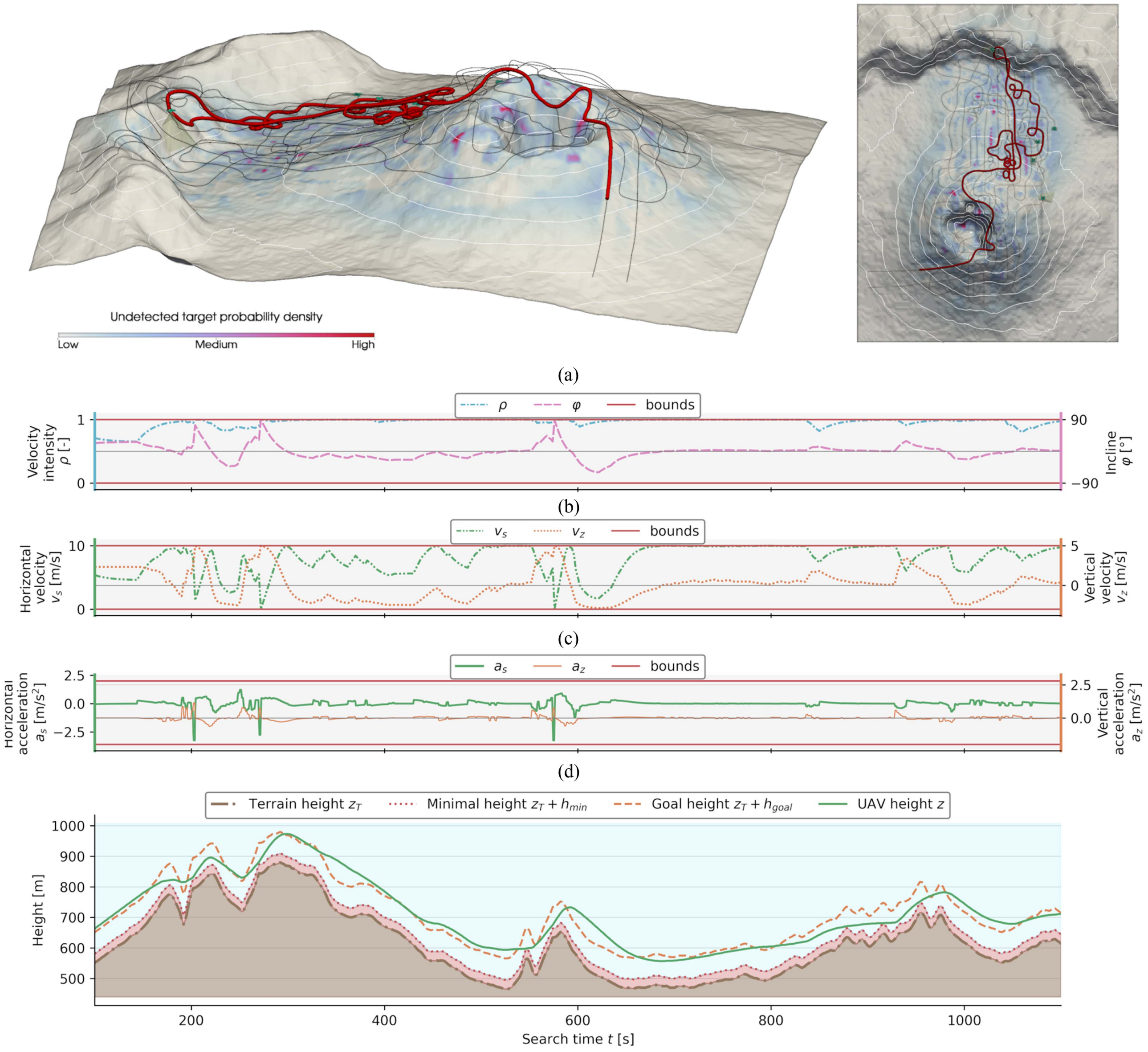
(e)

Fig. 7. (a) Plastic world case survey simulation after 30 min of search time including UAVs' trajectories with an in depth parameter analysis for the trajectory marked in red. (b) High aircraft maneuverability is deducted from frequent variations of control parameters which lead to (c) velocity and (d) acceleration jumps. (e) Height analysis of the red trajectory which is performed by an UAV with a goal altitude of 50 m and a minimum altitude of 30 m. The UAV flies relatively closely to the goal altitude except upon flying over terrain depressions, when it sometimes chooses to favor velocity over altitude (shown at about 680, 1080, and 1300 s), which helps in momentum preservation and results in a smoother trajectory. The inclination for terrain adherence and velocity maximization is evident on elevation peaks, where altitude drops below the goal value (sometimes even up to minimal altitude) and then after crossing the peak UAV ascends back to goal value which results in smoother trajectories, allowing for faster flight.

while the area closely surrounding it has been successfully explored.

E. Mount Vesuvius

The second test case is a survey simulation of Mount Vesuvius, which is situated near Naples, Italy. We chose to survey this area only with multirotor UAVs because of their high maneuverability, hence, they could easily search volcanic cone and crater. In this case, we demonstrated nonhomogeneous swarm support by using 3 UAVs flying at an altitude of 50 m with one type of camera sensor and 2 UAVs flying at an altitude of 100 m were we simulated the use of 2 × optical zoom (the sensor has narrower FOV and higher detection sensing



(a)

(b)

(c)

(d)

(e)

Fig. 8. Mount Vesuvius case survey simulation with 5 cooperating UAVs after 1400 s. (a) The 3-D scene contains the terrain colored in reference to the undetected target probability with gray representing the lowest and red representing the highest value. The black lines represent UAVs' trajectories and the red highlighted part, which is executed by an UAV with a goal altitude of 100 m and a minimum altitude of 30 m, is analyzed in following graphs. It is clear that (b) the control parameters, (c) velocities, and (d) accelerations stay within the prescribed limits and correspond to each other. (e) Graph showcases an example of trajectory smoothing and altitude variation dampening effect at elevation peaks (from 850 to 1000 s). Rather than strictly adhering to the goal altitude when encountering consecutive elevation peaks, the UAV efficiently navigates through these peaks, maintaining high velocity and achieving a smooth trajectory. When flying over a sharp altitude peak, as shown around 550 s, the UAV generally exploits the full permitted altitude range, descending to the minimum allowed altitude at the peak's center and then ascending back to goal altitude. This results in a notably smoother trajectory, enhancing flight efficiency.

function was used). The search mission was simulated for 60 min. The inspection was completed without any collisions or breaching velocity, acceleration, or altitude constraints. The survey accomplishment metric reached 96%, which is evident on Fig. 11. The proposed motion control successfully balances between maximizing velocity and ideal altitude objectives while complying to all constraints. A greater difference between the minimal and goal altitudes gives the UAV more flight range, allowing it to fly at higher velocities while sacrificing proximity to the target altitude. However, it achieves a smooth flight trajectory, as shown in Fig. 8(b)–(e), which provides a detailed analysis of a selected segment of the UAV flight.

To demonstrate a realistic search scenario involving a restricted area, we conducted another survey simulation of Mount Vesuvius with the identical case configuration. The only difference between the cases is that we declared the volcano crater a no-fly zone. Visualizations in Figs. 8(a) and 9 exhibit simulation results with and without the no-fly zone, respectively. They showcase the terrain, undetected target probability distribution and UAVs' trajectories. Restricting the area of the search does

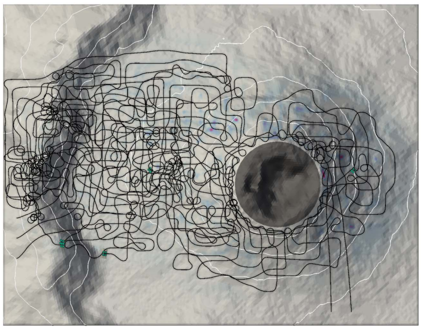


Fig. 9. Mount Vesuvius survey with incorporated no-fly zone after 60 min of search. The effect of the restriction on the final trajectories highlights the UAVs' compliance with the imposed constraint.

not have a significant qualitative impact on exploration, although, due to the excluded surface area, a more detailed search of the remaining zones is achieved in the same amount of time.

F. Star Dunes

For the third test, we performed a search simulation of a sandy desert in Algeria. We took a 7.5 km^2 area that contains star dunes, which are dunes that form in sandy deserts where the wind direction frequently changes. They can grow to considerable heights and are generally taller than other types of sand dunes, which is why they were chosen for altitude control demonstration. The search was conducted for 60 min with the control time step of 2 s. Compared to UAV (A), which used a $1 \times$ zoom camera sensor at a goal altitude of 50 m, the fix-wing UAV (C) used in this test case employs the same sensor but with $3 \times$ zoom and a goal altitude six times higher. This configuration covers a larger area but with reduced sensing intensity. By using only two UAVs, we managed to achieve survey accomplishment of 77% (see Fig. 11). Fig. 10 illustrates the search domain, the UAVs, their postsurvey trajectories, and an analysis of the highlighted trajectory displaying 1000 s of flight time. The analyzed trajectory covers nearly the entire domain. During its flight, the UAV prioritizes maximizing velocity while making altitude adjustments in order to stay close to the target altitude and avoid the minimal altitude zone.

G. Search Control Evaluation

For comparison, we conducted search simulations on the same test cases using four other methods. The HEDAC method without altitude and velocity control is used to show the benefits of altitude control. Other three methods are equipped with the proposed MPC: lawnmower search, and two spectral ergodic control methods, namely SMC [36] and mSMC [21]. Note that the SMC and mSMC methods were utilized only on cases involving rectangular domains without obstacles.

For the lawnmower MPC method, we relaxed the acceleration constraints for some of the UAVs to facilitate completion of the search with the same predicted path length, making the results comparable. In addition, we allowed collisions between UAVs and approximately adjusted the starting configuration to achieve the best coverage. A challenge with this algorithm is the variable horizontal velocity, which makes the final path length uncertain. This uncertainty complicates the setup of the starting configuration, potentially leading to issues, such as the UAVs leaving the search domain, failing to cover the entire area or interfering with another UAV. All that considered, our HEDAC implementation with MPC outperformed the lawnmower MPC method in terms of search performance, as evident in Fig. 11.

The HEDAC search at fixed altitude (without MPC control) was conducted at an altitude equal to highest point of the terrain plus the minimum search altitude. In cases with significant variation between the highest and lowest terrain point (plastic world and Mount Vesuvius), the method stalled and failed to explore the whole domain, highlighting the importance of altitude control implementation. In the scenario where the change in terrain elevation is less pronounced (Star dunes), the method did not stall, but still performed worse than the method with MPC, as shown in Fig. 11.

The original formulation of the SMC [36] method was employed, with the survey goal focused on minimizing $m_0 - c$. However, this method demonstrated lower performance compared to both the proposed approach and the lawnmower method. As a result, we further tested the mSMC method [21], which was specifically designed to address the same problem as the proposed method, the minimization of m. In the mSMC formulation, the exponent was adjusted to $1/2$, in contrast to the SMC's exponent of 3/2. The performance of the mSMC method was comparable to that of the proposed method, though slightly inferior as displayed in Fig. 11.

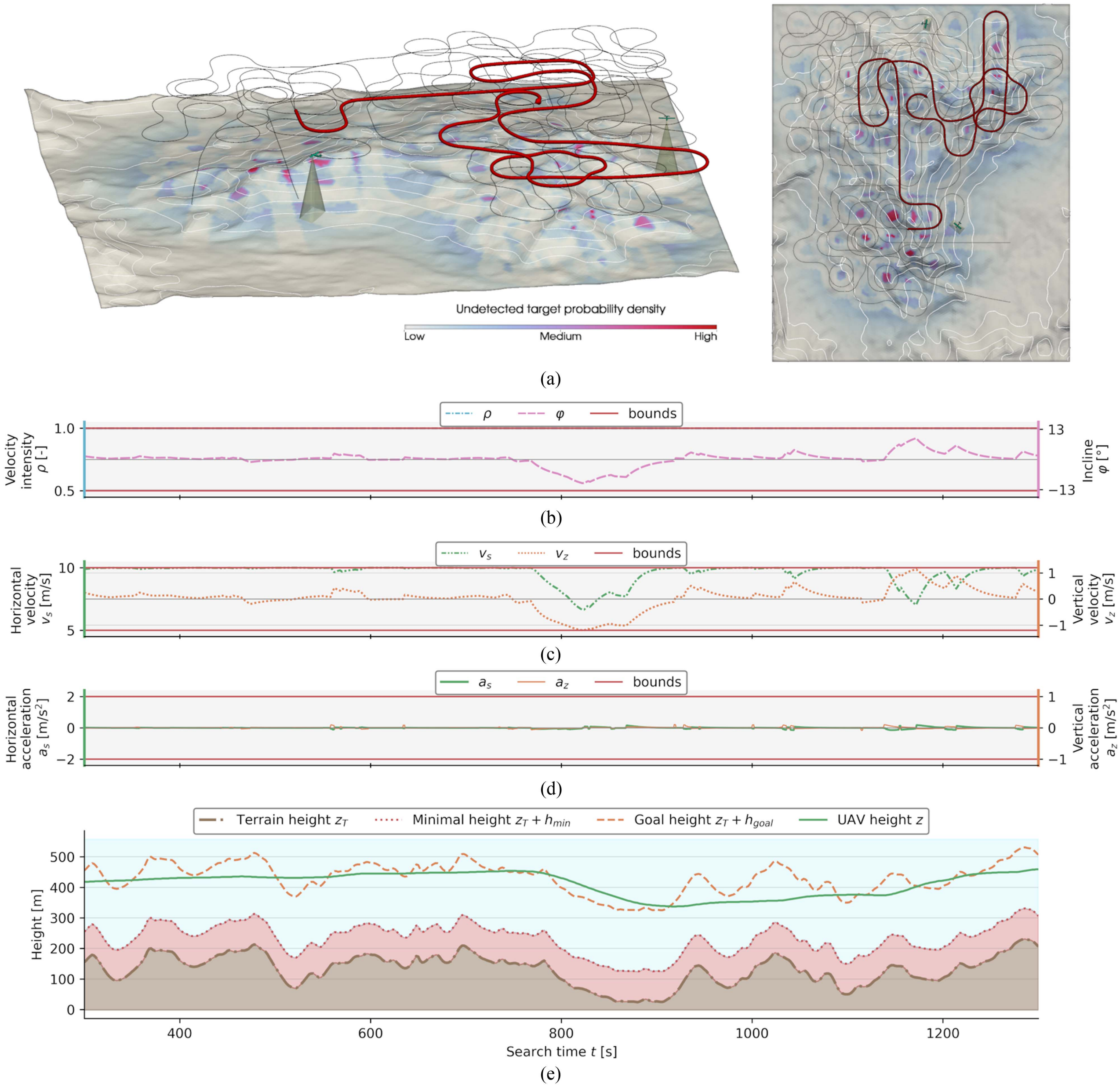
Survey accomplishment and computation time analysis were performed for all three cases and are shown in Fig. 11. With the selected UAV configurations and HEDAC parameters, as shown in Table II, the Plastic world, Mount Vesuvius and the Star dunes case achieved survey accomplishment rates of 98%, 96%, and 77%, respectively, within the specified search time. Computation time for each individual time step is lower than the time step Δt used in the calculation, which is a good indication for successful real-world application.

VIII. Experimental Validation

Prior to experimental validation, we assessed how real-world uncertainties would impact the method by conducting a robustness analysis, available in the supplementary materials.

The proposed method was tested in a real-world search scenario on Mount Učka, Croatia. Two missions with different UAV configurations were conducted, with each mission concluding upon the first instance of battery depletion among the UAVs. The UAVs used in the experiment were the DJI Matrice 210v2, equipped with a DJI X5S camera (5280 × 2970 px resolution), and the DJI Mavic 2 Enterprise Dual (with the RGB camera resolution of 4056 × 3040). The search and domain parameters are detailed in Table II, while the technical specifications of the UAVs are provided in Table III. The experiment site, along with several of the UAVs used, is shown in Fig. 12.

The objective of the missions was to detect and locate custom targets made out of 0.5 × 0.5 m cardboard sheets. A total of



(a)

(b)

(c)

(d)

(e)

Fig. 10. Star dunes case inspection after 60 min. (a) The 3-D scene displays the terrain with colors matching undetected target probability, with gray being the lowest and red the highest value. Black lines depict aircraft trajectories and the highlighted red trajectory section is further analyzed. (b) As evident from control parameters graph, the aircraft tends to keep its moving intensity at the maximum. (c) Velocity and (d) acceleration values are corresponding to the control parameters and their values are inside prescribed limits. (e) During its flight, the aircraft attempts to closely track the target altitude. However, due to its operational constraints and relatively high goal altitude of 300 m, the resulting trajectory exhibits a strong dampening effect on terrain elevation irregularities, making it much more fluid, allowing the aircraft to effortlessly navigate over terrain obstacles.

100 unique targets were scattered across the search domain, each target partially colored with two colors. A selection of the targets used in the experiment is displayed in Fig. 12. The spatial distribution of the targets followed a predefined initial undetected target probability distribution m_0. The search area was divided into three concentric zones with uniform probability density within each zone. Zone 1 ($r_{z1} < 150$ m) contained 20 targets, Zone 2 ($150 < r_{z2} < 300$ m) contained 30 targets, and Zone 3 ($300 < r_{z3} < 450$ m) contained the remaining 50 targets. The probability density inside each zone was calculated as the number of targets in the zone divided by the zone's area, and the resulting distribution was normalized over the entire domain.

A custom YOLOv8 detection model was trained using aerial photos of the targets taken at different altitudes. The data set consisted of 1166 images containing 27 600 object instances in total. The model was trained to detect a single object class (target) for 500 epochs, starting from a pretrained model *yolov8x.pt*, while the 80% of images were used for training and 20% for validation during training. To evaluate the model's performance, a separate validation dataset was used, consisting of 674 images and 6535 object instances. On this dataset, the model achieved a mean

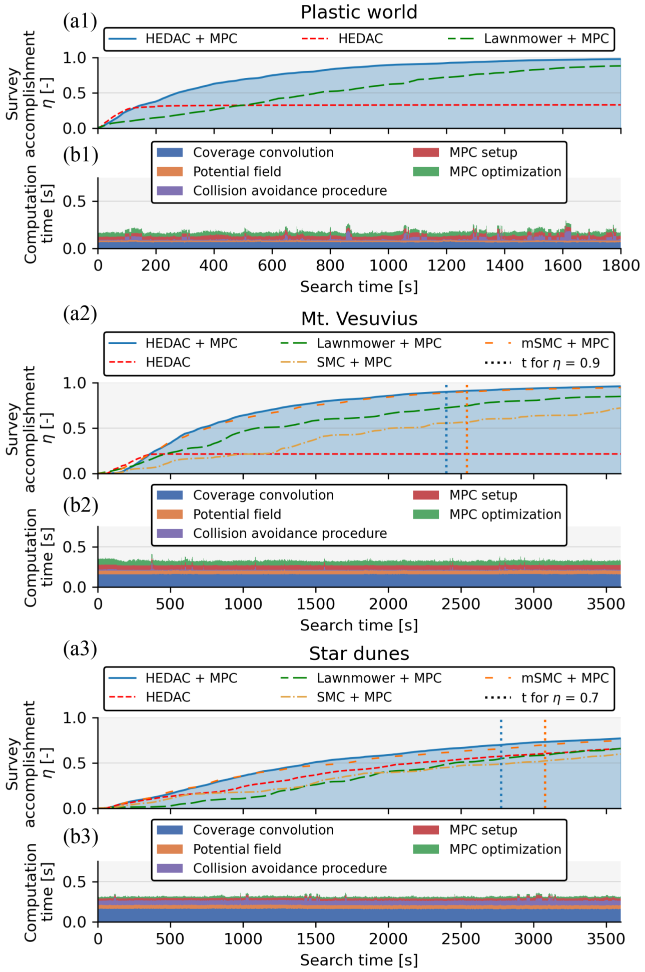


Fig. 11. Survey accomplishment and computation time for executed tests. After concluding the search, (a1) Plastic world case achieved 98%, (a2) Mount Vesuvius 96%, and (a3) Star dunes 77% survey accomplishment. The proposed method was compared with several others: HEDAC without MPC, as well as the lawnmower, SMC, and mSMC methods with MPC. With respect to survey accomplishment, the alternative methods demonstrated lower performance, although the mSMC method appears almost equivalent to HEDAC. To more clearly illustrate the performance difference, a vertical dotted line was added to highlight how much faster the proposed method reaches a given value of compared to mSMC. (b1-b3) Computation times are shown only for the HEDAC MPC method, as the other methods are more efficient in this aspect. However, the computation time for each time step remains well below one second, indicating the suitability of the proposed multi-UAV search control algorithm for real-time applications.

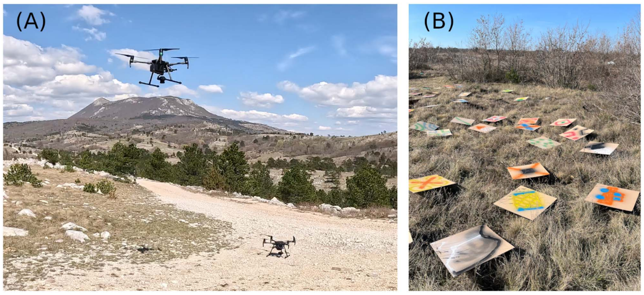


Fig. 12. (a) Matrice 210v2 UAVs at the experiment location, and (b) colored cardboard targets that were used in the experiment.

average precision of 0.68, a precision of 0.82, and a recall of 0.58 when evaluated with the default confidence threshold of 0.001. To obtain the model's recall values across the UAVs' flight altitude range the same dataset was partitioned into altitude intervals of 10 m, in $[30, 90]$ m. For each interval, the recall was computed and attributed to the intervals' corresponding midpoint altitude. The resulting recall values are shown in Fig. 6. For each UAV used in the real-world missions, the average horizontal velocity $v_{s,\mathrm{avg}}$ was experimentally determined, yielding values of 7.449 m/s for UAV M210 and 7.515 m/s for the UAV Mavic 2. Using (5) and (7), the corresponding sensing functions were computed and are presented in Fig. 6.

Although the algorithm's computation time per step is under one second, $\Delta t = 3$ was used in the field experiments to accommodate system constraints, including image focusing, image capture, and reporting the capture back to the algorithm.

Trajectory analysis of the first mission, executed with two M210 UAVs, is displayed in Fig. 13. It includes the search targets and complete flight trajectories for both UAVs, along with the analysis of control parameters, velocity, acceleration, and altitude for one trajectory. Comprehensive results of the conducted field experiments, including flight logs, captured imagery, the detection model, and flight animations, are accessible via the Open Science Framework repository: https://osf.io/t947u/.

A metric $\lambda(t)$, representing the target detection rate, is introduced to validate the proposed probabilistic search approach. It is defined as the ratio of detected targets to the total number of targets, where only the first detection of each target is considered and all subsequent detections of the same target are excluded. The main goal of the experiment is to assess whether the survey accomplishment metric η correlates with λ, which would confirm that η is a meaningful and reliable indicator of search effectiveness.

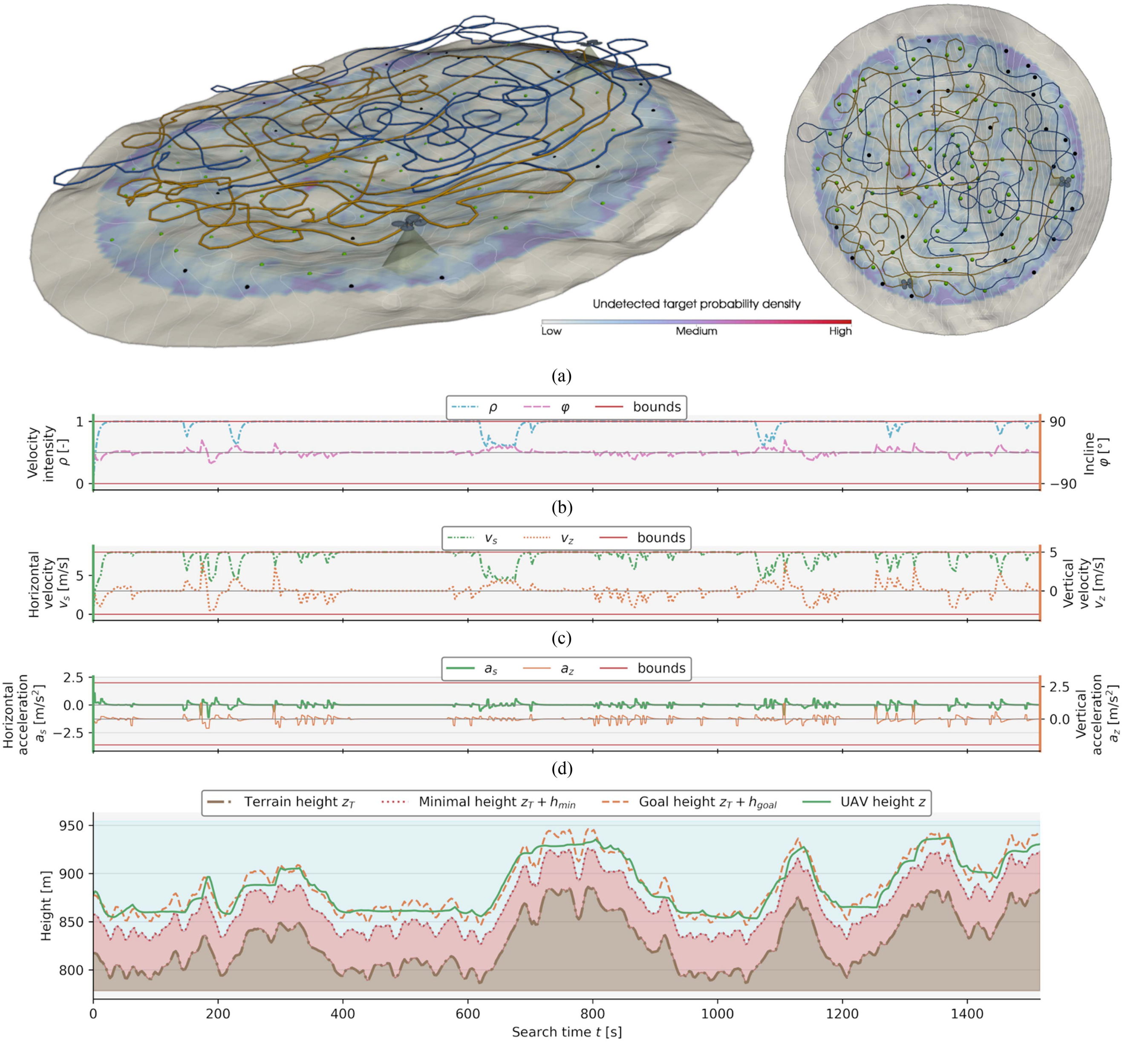
A notable correlation between η and λ was observed in both missions conducted during the Učka experiment, as illustrated in Fig. 14. These results suggest that η could serve as a practical metric for search coordinators to monitor progress and make informed operational decisions in real-world search missions.

IX. Possible Drawbacks

The proposed control can produce inadequate results if set-up inappropriately. One possible cause could be a lack of proper integration between the minimal turning radius, minimal altitude, and the camera sensor, particularly its FOV. This error could result in an endless circular motion around the area of interest as shown in Fig. 15. To prevent this issue, it is essential for the camera sensors' lateral scope to be greater than twice the minimal turning radius.

The drawback of proposed search control is that it is not truly three-dimensional. It explores a 3-D surface (the terrain), which is effectively two-dimensional, as the vertical dimension is subordinate to the horizontal plane. This means it cannot explore areas beneath trees and may encounter issues with steeply sloped terrains.

The ability of navigating particular terrain with a designated UAV setup, considering clearance and minimal altitude



(a)

(b)

(c)

(d)

(e)

Fig. 13. Visualization and analysis of the Uka experiment (Mission 1). (a) The 3-D scene displays the search area, along with the undetected target probability density field, and the UAVs' trajectories at the end of the search mission. The undetected and detected targets are shown as black and green dots, respectively. The blue UAV trajectory is further analyzed in following graphs which demonstrate that (b) the control variables, (c) velocities, and (d) accelerations remain within the prescribed limits. (e) The expected terrain-following behavior, confirming compliance with the minimum altitude constraint.

values, relies on the maximum terrain incline rather than directly checking horizontal clearances from the terrain during computation. Implementing horizontal clearance checks within the MPC and collision avoidance procedures would require evaluating terrain clearance at every point along the predicted path and the escape routes, which is computationally expensive and limits real-time control. Instead, the algorithm ensures horizontal clearance across the entire domain during the initialization process, using the specified minimum altitude. This approach eliminates computational overhead during MPC and collision avoidance since clearance satisfaction is guaranteed in advance.

To guarantee that horizontal clearance distances from the terrain are satisfied across the entire domain, the worst-case scenario is considered. This occurs when the UAV is at its lowest possible altitude and the smallest possible horizontal clearance is attained near the terrain of maximal incline, as shown in Fig. 16. Therefore, for each UAV, we can calculate the maximal supported terrain incline

$$\kappa_i = \arctan\left(\frac{h_{\min,i}}{\delta_i}\right)$$

which has to be greater than maximal terrain incline $\kappa_{T,\max}$. For all test cases, compatibility between the deployed UAV fleet

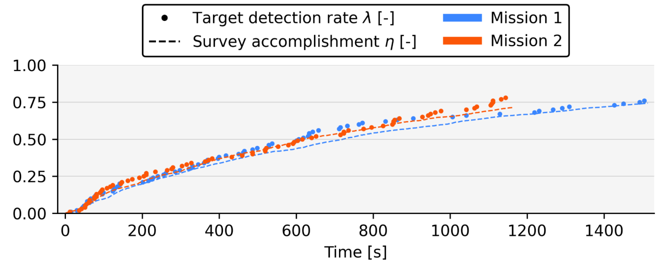


Fig. 14. Correlation between survey accomplishment η and target detection rate λ recorded during the Učka experiment missions. The dashed lines represent the evolution of η, while the dot markers indicate instances of new target detections contributing to λ. Each mission is distinguished by color.

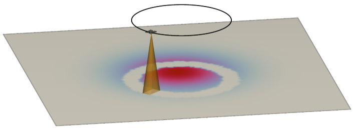


Fig. 15. UAV stuck in a circular motion around the area of interest due to improperly defined parameters. To address this problem, an appropriate combination of minimal turning radius, minimal altitude, and the camera sensor (with adequate FOV) should be employed.

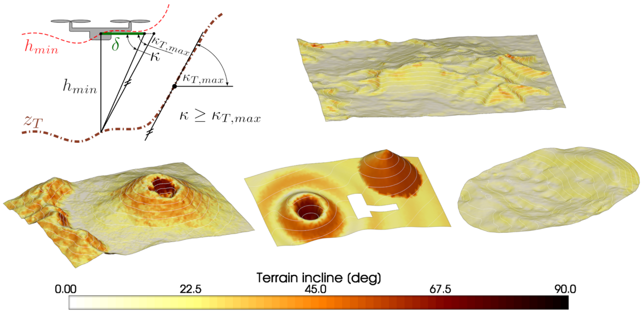


Fig. 16. Sketch illustrating the idea behind UAVs' maximal supported incline calculation and terrain incline visualization for all used test cases.

and the domain was verified by ensuring that the condition $\kappa_i > \kappa_{T,\max}$ was satisfied.

To explore a domain with an incline higher than supported while maintaining the desired minimal clearance, the only option (that does not include terrain modification/smoothing) is to raise the minimal flight altitude. However, for real-world implementation, this problem can be completely resolved by utilizing horizontal distance sensors usually integrated within the modern UAVs.

X. Conclusion

In this article, we presented a centralized multi-UAV control algorithm that conducts complex terrain inspection and manages UAVs' flight in three dimensions. It concurrently addresses two-dimensional coverage problem, utilizing the HEDAC method, and employs MPC for altitude and velocity regulation. Horizontal movement is driven by a potential field that is dynamically adjusted using custom sensory framework featuring a simulated real-time image capture and detection system. Implemented collision avoidance technique ensures adherence to clearance distance restrictions among multiple UAVs and with the boundary, while also satisfying the minimal turning radius constraints. It can be effectively employed with both multirotor and fixed-wing UAVs, depending on inspection requirements and the search domain characteristics.

The proposed method was simulated on three test cases. The first test case consisted of a challenging, synthetically generated terrain that was inspected through the collaborative action of three multirotor UAVs. The second case was an inspection of Mount Vesuvius that was conducted using five multirotor UAVs utilizing two different sensing setups with accompanying flight altitudes. The third case showcased a survey of desert dunes that was performed using two fixed-wing UAVs. Interesting sections of UAVs' trajectories were highlighted and thoroughly analyzed. In all three cases, the chosen parameters yielded satisfactory survey results with minimal computation time, facilitating real-time UAV control. The search success is the overall and final grade of the UAV control. It is evaluated with survey accomplishment η metric in simulations of UAV search in realistic scenarios. It exceeded 77% in the poorest performing tested scenario, which would translate into less than 23% chance of an undetected target, on average, across the entire search domain. The opportunity to increase η exists through improvements in sensing/detection capabilities, UAV maneuverability, and reliable terrain data, along with an extension of flight time. Deploying more UAVs would undoubtedly have the most significant impact. With the scalability of HEDAC algorithm [19] and the results presented here, authors are confident that additional improvements will allow a greater number of UAVs to be utilized in the search.

Drawbacks of the method have been discussed, with the primary limitation being its inability to navigate domains featuring significant terrain inclines. However, in a practical application this problem can be resolved by employing proximity sensors usually installed in modern UAVs.

The main scientific contributions of the presented UAV search framework are threefold. First, it introduces an accurate probabilistic model for target distribution and detection over uneven terrain. Second, it presents an advanced control framework for multi-UAV search, combining planar ergodic control with MPC-based altitude and velocity adjustments tailored to terrain features. Finally, the framework is validated through both simulated and real-world experiments, confirming the effectiveness of the proposed methodology by evaluating flight kinematics, search performance, and robustness to uncertainties.

We believe that the experimental validation, which confirmed the search accomplishment predicted by the probabilistic model, serves as the ultimate endorsement of proposed methodology and represents a highly promising feature for guiding and managing a real-world search operations.

In terms of future improvements, there are several enhancements that could be incorporated into the method. One possible enhancement involves implementing automatic terrain smoothing when dealing with steeply inclined domains. While implementing absolute distance checks between the UAV and the nearest terrain point sounds like an ideal scenario, it is crucial to acknowledge that this could result in a substantial increase in computation time, potentially making the method impractical for real-time UAV control. Integration of the UAVs' proximity sensor could serve as a convenient way to address the horizontal clearance problem. Another area for improvement is introducing the capability to selectively include or exclude UAVs from the search and manage their return to the starting point. This feature would address scenarios, such as battery replacement, allowing for extended search duration within a single run. In addition, it would be beneficial to investigate the option of distributed computation within the proposed method to assess its advantages, drawbacks, and limitations. Promising practical direction for future research is the integration of an advanced sensor options, such as model that explicitly accounts for variations in pitch and roll, and zoom, enabling more accurate sensing and detection.

References

[1] I. Abraham and T. D. Murphey, "Decentralized ergodic control: Distribution-driven sensing and exploration for multiagent systems," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 2987–2994, Oct. 2018.

[2] A. Ahmadzadeh, A. Jadbabaie, V. Kumar, and G. J. Pappas, "Cooperative coverage using receding horizon control," in *Proc. Eur. Control Conf.*, 2007, pp. 2466–2470.

[3] E. T. Alotaibi, S. S. Alqefari, and A. Koubaa, "LSAR: Multi-UAV collaboration for search and rescue missions," *IEEE Access*, vol. 7, pp. 55817–55832, 2019.

[4] R. D. Arnold, H. Yamaguchi, and T. Tanaka, "Search and rescue with autonomous flying robots through behavior-based cooperative intelligence," *J. Int. Humanitarian Action*, vol. 3, no. 1, pp. 1–18, 2018.

[5] G. M. Atınç, D. M. Stipanović, and P. G. Voulgaris, "A swarm-based approach to dynamic coverage control of multi-agent systems," *Automatica*, vol. 112, 2020, Art. no. 108637.

[6] C. Audet and J. E. Dennis Jr, "Mesh adaptive direct search algorithms for constrained optimization," *SIAM J. Optim.*, vol. 17, no. 1, pp. 188–217, 2006.

[7] L. Bauersfeld and D. Scaramuzza, "Range, endurance, and optimal speed estimates for multicopters," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 2953–2960, Apr. 2022.

[8] J. Bellingham, A. Richards, and J. P. How, "Receding horizon control of autonomous aerial vehicles," in *Proc. Amer. Control Conf.*, 2002, vol. 5, pp. 3741–3746.

[9] C. Bilaloglu, T. Löw, and S. Calinon, "Whole-body ergodic exploration with a manipulator using diffusion," *IEEE Robot. Autom. Lett.*, vol. 8, no. 12, pp. 8581–8587, Dec. 2023.

[10] B. Crnković, S. Ivić, and M. Zovko, "Fast algorithm for centralized multiagent maze exploration," 2023, *arXiv:2310.02121*.

[11] F. Gao, Y. Lin, and S. Shen, "Gradient-based online safe trajectory generation for quadrotor flight in complex environments," in *Proc. IEEE/RSJ Int. Conf. Intell. robots Syst.*, 2017, pp. 3681–3688.

[12] H. Gao et al., "Energy-efficient velocity control for massive numbers of UAVs: A mean field game approach," *IEEE Trans. Veh. Technol.*, vol. 71, no. 6, pp. 6266–6278, Jun. 2022.

[13] S. Garrido, J. Muñoz, B. López, F. Quevedo, C. A. Monje, and L. Moreno, "Fast marching techniques for teaming UAV's applications in complex terrain," *Drones*, vol. 7, no. 2, 2023, Art. no. 84.

[14] C. Geuzaine and J.-F. Remacle, "Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities," *Int. J. Numer. Methods Eng.*, vol. 79, no. 11, pp. 1309–1331, 2009.

[15] D. Gkouletsos, A. Iannelli, M. H. de Badyn, and J. Lygeros, "Decentralized trajectory optimization for multi-agent ergodic exploration," *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 6329–6336, Oct. 2021.

[16] M. A. Goodrich et al., "Supporting wilderness search and rescue using a camera-equipped mini UAV," *J. Field Robot.*, vol. 25, no. 1/2, pp. 89–110, 2008.

[17] D. Guan and W. Yuan, "A survey of mislabeled training data detection techniques for pattern classification," *IETE Tech. Rev.*, vol. 30, no. 6, pp. 524–530, 2013.

[18] A. Hubenko, V. A. Fonoberov, G. Mathew, and I. Mezic, "Multiscale adaptive search," *IEEE Trans. Systems, Man, Cybern., Part B (Cybern.)*, vol. 41, no. 4, pp. 1076–1087, Aug. 2011.

[19] S. Ivić, "Motion control for autonomous heterogeneous multiagent area search in uncertain conditions," *IEEE Trans. Cybern.*, vol. 52, no. 5, pp. 3123–3135, May 2022.

[20] S. Ivić, A. Andrejčuk, and S. Družeta, "Autonomous control for multiagent non-uniform spraying," *Appl. Soft Comput.*, vol. 80, pp. 742–760, 2019.

[21] S. Ivić, B. Crnković, H. Arbabi, S. Loire, P. Clary, and I. Mezić, "Search strategy in a complex and dynamic environment: The mh370 case," *Sci. Rep.*, vol. 10, no. 1, 2020, Art. no. 19640.

[22] S. Ivić, B. Crnković, L. Grbčić, and L. Matleković, "Multi-UAV trajectory planning for 3D visual inspection of complex structures," *Automat. Construction*, vol. 147, 2023, Art. no. 104709.

[23] S. Ivić and S. Družeta, "Indago: Python 3 module for numerical optimization," version 0.4.5. 2023. Accessed: Jul. 3, 2023. [Online]. Available: https://pypi.org/project/Indago/0.4.5/

[24] S. Ivić, A. Sikirica, and B. Crnković, "Constrained multi-agent ergodic area surveying control based on finite element approximation of the potential field," *Eng. Appl. Artif. Intell.*, vol. 116, 2022, Art. no. 105441.

[25] S. Ivić, B. Crnković, and I. Mezić, "Ergodicity-based cooperative multiagent area coverage via a potential field," *IEEE Trans. Cybern.*, vol. 47, no. 8, pp. 1983–1993, Aug. 2017.

[26] E. N. Johnson, J. G. C. Mooney, J. Ong Hartman, and V. Sahasrabudhe, "Flight testing of nap-of-the-earth unmanned helicopter systems," in *Proc. 67th Amer. Helicopter Soc. Int. Annu. Forum*, 2011, pp. 3180–3192.

[27] B. O. Koopman, "The theory of search. II. target detection," *Operations Res.*, vol. 4, no. 5, pp. 503–531, 1956.

[28] A. Kosari, H. Maghsoudi, A. Lavaei, and R. Ahmadi, "Optimal online trajectory generation for a flying robot for terrain following purposes using neural network," *Proc. Inst. Mech. Engineers, Part G: J. Aerosp. Eng.*, vol. 229, no. 6, pp. 1124–1141, 2015.

[29] Y. Kuwata, T. Schouwenaars, A. Richards, and J. How, "Robust constrained receding horizon control for trajectory planning," in *Proc. AIAA Guidance, Navigat., Control Conf. Exhibit*, 2005, Art. no. 6079.

[30] B. Li, Y. Zhu, Z. Wang, C. Li, Z.-R. Peng, and L. Ge, "Use of multi-rotor unmanned aerial vehicles for radioactive source search," *Remote Sens.*, vol. 10, no. 5, 2018, Art. no. 728.

[31] L. Lin and M. A. Goodrich, "Uav intelligent path planning for wilderness search and rescue," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2009, pp. 709–714.

[32] C. Liu, Y. Tao, J. Liang, K. Li, and Y. Chen, "Object detection based on yolo network," in *Proc. IEEE 4th Inf. Technol. Mechatron. Eng. Conf.*, 2018, pp. 799–803.

[33] Y. Liu and Z. Ren, "A probabilistic measure of multi-robot connectivity and ergodic optimal control," in *Proc. Robot.: Sci. Syst.*, Los Angeles, CA, USA, 2025. [Online]. Available: https://www.roboticsproceedings.org/rss21/p100.html

[34] T. Löw, J. Maceiras, and S. Calinon, "drozBot: Using ergodic control to draw portraits," *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 11728–11734, Oct. 2022.

[35] A. Marchidan and E. Bakolas, "Collision avoidance for an unmanned aerial vehicle in the presence of static and moving obstacles," *J. Guidance, Control, Dyn.*, vol. 43, no. 1, pp. 96–110, 2020.

[36] G. Mathew and I. Mezić, "Metrics for ergodicity and design of ergodic dynamics for multi-agent systems," *Physica D: Nonlinear Phenomena*, vol. 240, no. 4–5, pp. 432–442, 2011.

[37] A. Mavrommati, E. Tzorakoleftherakis, I. Abraham, and T. D. Murphey, "Real-time area coverage and target localization using receding-horizon ergodic exploration," *IEEE Trans. Robot.*, vol. 34, no. 1, pp. 62–80, Feb. 2018.

[38] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.

[39] L. M. Miller, Y. Silverman, M. A. MacIver, and T. D. Murphey, "Ergodic exploration of distributed information," *IEEE Trans. Robot.*, vol. 32, no. 1, pp. 36–52, Feb. 2016.

[40] T. Petso, R. S. Jamisola, D. Mpoeleng, and W. Mmereki, "Individual animal and herd identification using custom YOLO v3 and v4 with images taken from a UAV camera at different altitudes," in *Proc. IEEE 6th Int. Conf. Signal Image Process.*, 2021, pp. 33–39.

[41] M. D. Phung and Q. Phuc Ha, "Safety-enhanced UAV path planning with spherical vector-based particle swarm optimization," *Appl. Soft Comput.*, vol. 107, 2021, Art. no. 107376.

[42] A. Prabhakar et al., “Ergodic specifications for flexible swarm control: From user commands to persistent adaptation,” in *Proc. Robot.: Sci. Syst.*, Corvalis, Oregon, USA, Jul. 2000, doi: 10.15607/RSS.2020.XVI.067.

[43] L. Qingqing et al., “Towards active vision with uavs in marine search and rescue: Analyzing human detection at variable altitudes,” in *Proc. IEEE Int. Symp. Safety, Secur., Rescue Robot.*, 2020, pp. 65–70.

[44] Z. Ren, A. K. Srinivasan, B. Vundurthy, I. Abraham, and H. Choset, “A pareto-optimal local optimization framework for multiobjective ergodic search,” *IEEE Trans. Robot.*, vol. 39, no. 5, pp. 3452–3463, Oct. 2023.

[45] A. Romero, S. Sun, P. Foehn, and D. Scaramuzza, “Model predictive contouring control for time-optimal quadrotor flight,” *IEEE Trans. Robot.*, vol. 38, no. 6, pp. 3340–3356, Dec. 2022.

[46] M. S. H Salem, F. H. K. Zaman, and N. M. Tahir, “Effectiveness of human detection from aerial images taken from different heights,” *TEM J.*, vol. 10, no. 2, 2021, Art. no. 522.

[47] N. Sandrić, “A note on the birkhoff ergodic theorem,” *Results Math.*, vol. 72, no. 1, pp. 715–730, 2017.

[48] J. Schöberl, “Netgen an advancing front 2D/3D-mesh generator based on abstract rules,” *Comput. Visualization Sci.*, vol. 1, no. 1, pp. 41–52, 1997.

[49] J. Schöberl, “C 11 implementation of finite elements in NGSolve,” Institute for analysis and scientific computing, Vienna Univ. Technol., Vienna, Austria, ASC Rep. 30/2014, pp. 1–23, 2014.

[50] M. Silvagni, A. Tonoli, E. Zenerino, and M. Chiaberge, “Multipurpose UAV for search and rescue operations in mountain avalanche events,” *Geomatics, Natural Hazards Risk*, vol. 8, no. 1, pp. 18–33, 2017.

[51] A. Siouras, K. Stergiou, P. Karlsson, and S. Moustakidis, “Hybrid object detection methodology combining altitude-dependent local deep learning models for search and rescue operations,” *J. Control Decis.*, vol. 11, no. 3, pp. 355–365, 2022.

[52] H. Song et al., “Multi-UAV disaster environment coverage planning with limited-endurance,” in *Proc. Int. Conf. Robot. Autom.*, 2022, pp. 10760–10766.

[53] J. Thomas, G. Stastny, A. Garcia, and S. S. Keshmiri, “Collision and obstacle avoidance in unmanned aerial systems using morphing potential field navigation and nonlinear model predictive control,” *J. Dynamic Syst., Meas., Control*, vol. 137, no. 1, 2015, Art. no. 014503.

[54] T.-Y. Sun, C.-L. Huo, S.-J. Tsai, Y.-H. Yu, and C.-C. Liu, “Intelligent flight task algorithm for unmanned aerial vehicle,” *Expert Syst. with Appl.*, vol. 38, no. 8, pp. 10036–10048, 2011.

[55] 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.

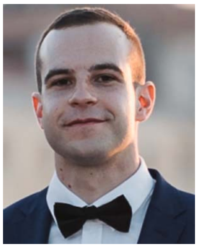
[56] A. Torralba, “How many pixels make an image?,” *Vis. Neurosci.*, vol. 26, no. 1, pp. 123–131, 2009.

[57] A. Xu, C. Viriyasuthee, and I. Rekleitis, “Optimal complete terrain coverage using an unmanned aerial vehicle,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 2513–2519.

[58] Y. Zheng and C. Zhai, “Distributed coverage control of multi-agent systems in uncertain environments using heat transfer equations,” 2022, *arXiv:2204.09289*.

[59] B. Zhou, F. Gao, L. Wang, C. Liu, and S. Shen, “Robust and efficient quadrotor trajectory generation for fast autonomous flight,” *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 3529–3536, Oct. 2019.

[60] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, “Object detection in 20 years: A survey,” *Proc. IEEE*, vol. 111, no. 3, pp. 257–276, Mar. 2023.



Luka Lanča received the M.S. degree in mechanical engineering and the Ph.D. degree in technical sciences from the University of Rijeka, Rijeka, Croatia, in 2021 and 2025, respectively.

He is currently a Postdoctoral Researcher with the Department of Fluid Mechanics and Computational Engineering, Faculty of Engineering, University of Rijeka. His work focuses on the Croatian Science Foundation project *Autonomous UAV system for oceanic search and rescue*. His research interests include ergodic and model predictive control, autonomous UAV systems, machine-learning-based detection models, and computational fluid dynamics.

Karlo Jakac received the M.S. degree in mechanical engineering and the Ph.D. degree in technical sciences from the University of Rijeka in 2021 and 2025, respectively.

Since 2021, he has been with the Faculty of Engineering, University of Rijeka, where he is currently a Postdoctoral Researcher with the Department of Fluid Mechanics and Computational Engineering. He is a member of the Croatian Science Foundation project *Autonomous UAV system for oceanic search and rescue*, contributing to the development and experimental verification of UAV-based methodologies. His research interests include experimental fluid mechanics, computational fluid dynamics, optimization techniques, and autonomous control of UAVs.

Stefan Ivić received the M.S. degree in mechanical engineering and the Ph.D. degree in technical sciences from the University of Rijeka in 2008 and 2014, respectively.

Since 2008, he has been with the Faculty of Engineering, University of Rijeka, where he is currently an Associate Professor with the Department of Fluid Mechanics and Computational Engineering. He is a PI of the Croatian Science Foundation project *Autonomous UAV system for oceanic search and rescue*. His research interests are development of optimization methods and their applications to simulation-based engineering problems, as well as ergodic control and autonomous UAV systems.