

Application of potential field-based algorithms in unmanned aerial vehicle search missions

Lanča, Luka

Doctoral thesis / Doktorski rad

2025

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka, - FACULTY OF ENGINEERING / Sveučilište u Rijeci, - Tehnički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:190:634661>

Rights / Prava: [Imenovanje 4.0 međunarodna](#) / [Attribution 4.0 International](#)

Download date / Datum preuzimanja: **2026-03-20**



Repository / Repozitorij:

[Repository of the University of Rijeka, Faculty of Engineering / Repozitorij Tehničkog fakulteta u Rijeci](#)



UNIVERSITY OF RIJEKA
FACULTY OF ENGINEERING

Luka Lanča

**APPLICATION OF POTENTIAL
FIELD-BASED ALGORITHMS IN
UNMANNED AERIAL VEHICLE SEARCH
MISSIONS**

DOCTORAL THESIS

Rijeka, 2025

UNIVERSITY OF RIJEKA
FACULTY OF ENGINEERING

Luka Lanča

**APPLICATION OF POTENTIAL
FIELD-BASED ALGORITHMS IN
UNMANNED AERIAL VEHICLE SEARCH
MISSIONS**

DOCTORAL THESIS

Supervisor: Assoc. Prof. Stefan Ivić, PhD

Rijeka, 2025

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET

Luka Lanča

**PRIMJENA ALGORITAMA ZASNOVANIH
NA POTENCIJALNOM POLJU U
MISIJAMA PRETRAŽIVANJA
BESPILOTNIM LETJELICAMA**

DOKTORSKI RAD

Rijeka, 2025.

Doctoral thesis supervisor: Assoc. Prof. Stefan Ivić, PhD

The doctoral thesis was defended on _____ at the University of Rijeka, Faculty of Engineering, Croatia, in front of the following Evaluation Committee:

1. Prof. Siniša Družeta, PhD, University of Rijeka, Faculty of Engineering
2. Assoc. Prof. Bojan Crnković, PhD, University of Rijeka, Faculty of Mathematics
3. Assoc. Prof. Matko Orsag, PhD, University of Zagreb, Faculty of Electrical Engineering and Computing

ABSTRACT

Unmanned aerial vehicles (UAVs) provide a versatile and efficient solution for search operations, combining mobility, adaptability, and the ability to cover large areas. This work presents a methodology for autonomous UAV-based search in complex land and maritime environments, integrating probabilistic modeling, ergodic search principles, and model predictive control (MPC). Search missions are guided by a probabilistic model representing uncertainty in target locations, which adapts according to the UAVs' achieved search effort. In maritime scenarios, the model incorporates sea surface dynamics and target drift, allowing the search to adapt to environmental changes.

The search is governed by the potential field-based ergodic control method. It ensures systematic and efficient area coverage suitable for solving a probabilistically formulated search problem. For applications in hilly terrain, MPC is utilized to generate smooth, collision-free trajectories that maintain the desired search height and balance area coverage with computer vision target detection. The approach is validated through numerical simulations over varied terrain and real-world experiments under challenging conditions, demonstrating robustness to uncertainties in target location, UAV control, and localization errors. Static target search results demonstrate close correspondence between estimated search performance and actual detections, confirming the accuracy of the probabilistic model. In dynamic maritime searches, the target drift and its uncertainties are considered with the advection and diffusion of the probability distribution, respectively. The results indicate that the probability field evolves consistently with target movement, improving performance compared to traditional methods.

Overall, the methodology achieves the desired area coverage while adapting to uncertainties in both target location and UAV operation. The combination of probabilistic modeling, ergodic search, and terrain-adaptive motion control enables systematic, uncertainty-aware multi-UAV search operations. This thesis provides a validated framework for autonomous search missions in both terrestrial and maritime environments, offering a significant improvement in operational effectiveness for real-world search and rescue applications.

Keywords: Search and rescue, Multi-UAV control, Ergodic control, Potential field methods, Probabilistic search model, Computer vision detection

PROŠIRENI SAŽETAK

Bespilotne letjelice pružaju svestrano i učinkovito rješenje za operacije traganja, kombinirajući mobilnost, prilagodljivost i sposobnost pokrivanja velikih područja. Ovaj rad predstavlja metodologiju za autonomno traganje temeljeno na bespilotnim letjelicama u složenim kopnenim i pomorskim okruženjima. Metodologija integrira probabilističko modeliranje, ergodička načela pretraživanja i modelsko prediktivno upravljanje. Misije traganja vođene su probabilističkim modelom koji obuhvaća nesigurnost u lokacijama meta, a koji se razvija u skladu s postignutim naporom traganja bespilotnih letjelica. U pomorskim scenarijima, model uzima u obzir dinamiku morske površine i pomicanje meta, omogućujući misiji traganja da se prilagodi promjenama u okolišu.

Traganje se odvija prema metodi ergodičke kontrole temeljene na potencijalnom polju. Ona osigurava sustavno i učinkovito pokrivanje područja i pogodna je za rješavanje probabilistički formuliranog problema traganja. Za primjene na brdovitom terenu, modelsko prediktivno upravljanje se koristi za generiranje glatkih putanja bez sudara koje održavaju željenu visinu leta s ciljem postizanja ravnoteže između pokrivenosti područja i učinkovitosti detekcije meta računalnim vidom. Pristup je validiran numeričkim simulacijama na raznolikom terenu i eksperimentima u stvarnim uvjetima, demonstrirajući otpornost na nesigurnosti u lokaciji meta, upravljanju bespilotnim letjelicama i pogreškama lokalizacije. Rezultati potrage za stacionarnim metama pokazuju blisku podudarnost između procijenjene učinkovitosti traganja i stvarnih detekcija, potvrđujući točnost probabilističkog modela. U pomorskim traganjima, dinamika meta i nesigurnosti u gibanju opisuju s advekcijom i difuzijom raspodjele vjerojatnosti. Rezultati pokazuju da se polje vjerojatnosti dosljedno razvija s kretanjem meta, poboljšavajući performanse u usporedbi s tradicionalnim metodama.

U konačnici, metodologija postiže željenu pokrivenost područja, dok se istovremeno prilagođava nesigurnostima i u lokaciji meta i u radu bespilotne letjelice. Kombinacija probabilističkog modeliranja, ergodičkog pretraživanja i terenski prilagodljive kontrole gibanja omogućuje sustavne operacije traganja s više bespilotnih letjelica, uzimajući u obzir nesigurnosti. Ovaj doktorski rad pruža provjereni sustav za autonomne misije traganja u kopnenim i pomorskim

okruženjima, nudeći značajno poboljšanje operativne učinkovitosti za stvarne primjene traganja i spašavanja.

Ključne riječi: Traganje i spašavanje, Upravljanje više bespilotnih letjelica, Ergodičko upravljanje, Metode potencijalnog polja, Probabilistički model pretraživanja, Detekcija računalnim vidom

CONTENTS

1	Introduction	1
1.1	Hypothesis and Research Goals	2
1.2	Scientific Contribution	3
1.3	Thesis Structure	3
2	Literature overview	6
2.1	UAVs in search and rescue	6
2.2	Path planning and area coverage	7
2.3	Multi-agent systems and collision avoidance	8
2.4	Computer vision detection	9
2.5	Altitude and velocity control	10
2.6	Ergodic control	11
2.7	Dynamic environment search strategies	13
3	UAV search technology	14
3.1	Classification of UAVs	14
3.2	Control and state parameters	16
3.3	Onboard sensors	18
3.4	Experimental UAV system overview	20
4	Modeling UAV motion, computer vision sensing, and detection	22
4.1	Motion model	22
4.2	Theory of search	25
4.3	UAV sensor modeling	27
4.4	Computer vision detection	29
4.5	Combined sensing effect in static target search	31
4.6	Search task and search evaluation	32

5	Ergodic control	34
5.1	Heat Equation Driven Area Coverage – HEDAC	34
5.2	HEDAC implementation	36
5.3	Spectral Multiscale Coverage – SMC	37
6	Yaw constraints and collision avoidance procedure	40
6.1	Yaw control constraints	40
6.2	Collision avoidance optimization problem formulation	41
6.3	Decomposing and solving the optimization problem	45
7	Uneven terrain exploration	47
7.1	Problem formulation – UAV control over uneven terrain	47
7.2	Digital elevation model and elevation function	48
7.3	Trial trajectories and control functions	49
7.4	Optimization problem formulation	51
7.5	MPC optimization procedure	54
7.6	Terrain collision avoidance	56
8	Terrain search simulations	58
8.1	Simulated search scenarios overview	58
8.2	Sensing characteristics	60
8.3	Plastic world	62
8.4	Mount Vesuvius	65
8.5	Star dunes	69
8.6	Robustness analysis	72
9	Terrain search experiments	75
9.1	Experimental validation of UAV motion control	75
9.2	Experimental validation of UAV search methodology	81
10	Sea surface exploration	89
10.1	Problem formulation – Search in maritime environments	89
10.2	Dynamic target probability distribution	90
10.3	Numerical implementation	93

11 Sea surface search simulations	97
11.1 Modeling motion and detection of dynamic targets	97
11.2 Synthetic case – Cavity flow	98
11.3 Realistic search scenario – Unije Channel search	102
11.4 Complex search scenario – MH370 search	106
12 Sea surface search experiments	108
12.1 Surface flow reconstruction	108
12.2 Custom sea targets and detection model	112
12.3 Experimental search mission	113
13 Limitations and discussion	119
14 Conclusion	124
Bibliography	126
List of Figures	138
List of Tables	143
Curriculum Vitae	144
List of Publications	145

1 INTRODUCTION

Search and Rescue (SAR) operations are essential for locating lost, missing, or injured individuals, with the primary goal of ensuring their quick recovery while minimizing physical harm, emotional distress, exposure to environmental hazards, and ultimately saving lives. These missions often take place in complex and challenging environments such as dense forests, mountainous terrain, remote wilderness regions, collapsed urban structures, and coastal or offshore marine areas. The search process is often further complicated due to limited visibility, harsh weather conditions, and dynamic, unpredictable circumstances. The urgency of SAR missions is emphasized by the fact that any delay in locating victims greatly increases the risk of severe outcomes such as injury, dehydration, hypothermia, or death. Traditional manned search methods are often slow, require a lot of resources, and are sometimes hazardous for the responders themselves. Ground-based searches are limited by terrain and visibility, resulting in slow progress across large areas. Aerial searches offer wider coverage and faster results but are costly due to reliance on helicopters or specialized aircraft. Furthermore, responders are exposed to significant risks when operating in disaster-affected environments such as wildfires, earthquakes, or floods.

In recent years, the integration of Unmanned Aerial Vehicles (UAVs) into SAR operations has emerged as a valuable technological solution. UAVs can very quickly survey large and difficult to reach areas at a relatively low operational cost. They also improve mission safety by keeping responders out of high risk environments as they can be remotely operated. These factors make them a cost-effective supplement or alternative to traditional search teams. Their ability to execute predefined or adaptive trajectories, carry a range of onboard sensors such as RGB, thermal, or multispectral cameras for search tasks, along with additional sensors for collision avoidance, makes them highly effective in time-critical situations that require enhanced situational awareness. However, they require skilled operators and high levels of concentration for decision-making during the search process.

The motivation behind this research is to enhance UAV autonomy, allowing them to conduct search missions with minimal operator intervention. The goal is to further increase efficiency,

safety, and effectiveness of SAR operations. By optimizing and automating the search process, it becomes possible to reduce search times, increase detection probability, and lower the operational risks to human personnel. The need for faster victim location in challenging and hazardous environments further highlights the importance of advancing this technology.

1.1 Hypothesis and Research Goals

This thesis aims to develop a robust and practical framework for UAV-based search that integrates autonomous navigation based on ergodic control and computer vision detection algorithms, ultimately aiming to improve outcomes in critical life-saving missions.

In robotics, ergodic methods are widely used to explore unknown environments. They allow the system to both gather new information and combine previously known with newly collected data to focus exploration on high-priority areas. They have proven to be a reliable and efficient method for guiding inspection based on a prior probability (or information) distribution, prioritizing high-probability regions while ensuring coverage across the entire distribution.

Target detection in UAV search missions has traditionally been performed by a human operator monitoring the UAV camera feed and manually identifying targets. In recent years, advances in machine learning and computer vision have enabled automatic target detection on collected images or videos, providing increasingly reliable performance. This allows the system to process the visual data, recognize potential targets, and determine whether they are present within the explored area.

Building on these developments, the research hypothesis is formulated as: *effective autonomous multi-UAV search missions in complex natural terrain and maritime environments can be achieved using ergodic control based on target distribution belief and probabilistic model acknowledging detection sensor performance.*

To confirm the hypothesis, the research is guided by these objectives:

- Constrain the control algorithm to comply with the specific UAVs' technical parameters and physical limitations.
- Extend the two-dimensional ergodic control method to account for uneven terrain through UAV altitude and velocity control.

- Define a probabilistic sensing model based on computer vision detector performance at varying distances from the object.
- Investigate the dynamics of the target probability density field governed by advection and diffusion processes to enable search activities at sea.
- Develop an interface linking the control algorithm with physical UAV platforms.
- Conduct numerical and experimental validation of the proposed UAV control and search methodology.

1.2 Scientific Contribution

The main scientific contributions of this work are threefold. First, an exact probabilistic model is proposed to update the target probability distribution based on detection sensor performance, incorporating camera sensor specifications and computer vision detection model metrics. Second, a velocity and altitude control framework is designed that accounts for UAV dynamics and mission constraints, enabling the application of two-dimensional potential-based control methods in complex terrains. Third, a dynamic probability distribution model is introduced to represent temporally evolving target distributions driven by sea surface layer velocity, enabling effective search for drifting targets.

The complete framework is tested in numerical simulations conducted in both synthetically generated and natural terrains, as well as in maritime environments. The motion control and robustness to uncertainty are validated in experimental flights conducted over complex natural terrain using physical UAV platforms. The search methodology is validated in carefully designed experiments employing custom search targets in terrestrial and dynamic maritime environments.

1.3 Thesis Structure

This thesis is organized into fourteen chapters with corresponding subchapters. The introductory chapter presents the hypothesis and research objectives, highlights the scientific contributions, and outlines the overall thesis structure.

The second chapter provides a literature overview covering UAV applications in search and rescue, path planning and coverage methods, multi-agent systems and collision avoidance, velocity and altitude control methods, computer vision detection, ergodic search methods, and search strategies in dynamic environments.

The third chapter provides an overview of modern UAV platforms, including classification, coordinate systems, key control parameters, and commonly used sensors. It also describes the UAV equipment employed in the real-world tests.

The fourth chapter introduces the UAV motion and perception models. It characterizes the control parameters governing UAV trajectories, gives an overview of the theory of search, describes the spatial and probabilistic properties of the sensor, presents the sensing function that incorporates computer vision detector performance, and, finally, introduces the ergodic search task and evaluation metrics.

The fifth chapter presents ergodic control methods. It describes the procedure for calculating control parameters for a two-dimensional coverage problem using these methods, as well as the numerical implementation.

The sixth chapter focuses on validating and refining the solution provided by the ergodic method. It applies system constraints to the calculated control parameters and presents the formulation of the collision avoidance optimization problem.

The seventh chapter presents the velocity and altitude control procedure for uneven terrain exploration. It describes the terrain surface topography model, introduces the concepts of trial trajectories and control functions, and formulates the model predictive control for UAV altitude and speed.

The eighth chapter details the numerical validation of the proposed uneven terrain exploration methodology. It outlines the specifications of the simulated UAV platforms and the parameters for three search scenarios. It analyzes the results in terms of search performance and constraint compliance, and compares them with alternative methods. Additionally, it provides an estimate of the system's robustness to uncertainty in practical deployment scenarios.

The ninth chapter describes the real-world experimental tests conducted over terrestrial terrain. It validates the proposed motion control methodology through flights over complex natural terrain and evaluates executed flight trajectories with respect to simulation results. Furthermore, it details the training and validation of the custom computer vision detector, and demonstrates its application in validating the proposed search methodology.

The tenth chapter presents the dynamics of the target probability distribution model used in maritime search operations. It covers advection-diffusion modeling of sea surface target drift, introduces the uncertainty compensation procedure, and outlines the numerical implementation used to solve the governing equations.

The eleventh chapter presents the results of maritime search simulations using the proposed dynamic probability model. It demonstrates performance across synthetic, coastal, and large-scale ocean scenarios, examining robustness to parameter selection, effect of different relative velocities between the UAVs and the flow field, uncertainty compensation, and long search delay conditions.

The twelfth chapter describes the real-world experimental tests conducted in maritime environment. It outlines the method used to approximate the flow field and validates the search methodology under dynamic target behavior in experimental setting.

The thirteenth chapter discusses the limitations of the proposed methodology, possible areas of improvement and future research possibilities.

The final chapter reflects on the set hypothesis and summarizes core findings in relation to the defined research objectives.

2 LITERATURE OVERVIEW

This chapter presents a review of the existing scientific studies relevant to UAV applications in search and rescue, path planning and area coverage methods, multi-agent systems and collision avoidance, computer vision detection, altitude and velocity control methods, ergodic search, and dynamic environment search strategies.

2.1 UAVs in search and rescue

The integration of UAVs into SAR missions offers promising opportunities to overcome traditional limitations in speed and accessibility. By enabling rapid surveying of hard-to-reach locations and detailed environmental monitoring, UAVs help accelerate victim detection and improve overall mission effectiveness [1].

UAVs can assist ground searchers in wilderness SAR missions by employing search algorithms to find clues and direct the search toward the missing person [2]. Nonetheless, their integration changes traditional search roles and introduces new coordination challenges between UAV operators and ground personnel. An overview of the current use of UAVs in SAR missions is presented in [3]. The work highlights various real-world deployments of UAVs in disaster scenarios, including hurricanes, tsunamis, floods and fires.

The first documented case of a successful rescue in open terrain using a UAV equipped with a camera sensor and an automatic Computer Vision (CV) detector is reported in [4]. The work outlines guidelines for maximizing the probability of target detection in UAV search missions. To improve human detection in UAV-based SAR missions, an automatic person detector was integrated into a UAV smartphone control application to assist in target identification [5].

For effectiveness in certain specialized scenarios, UAVs can be customized and equipped with additional sensors tailored to the mission. An example is provided in [6], which presents the design, construction, and validation of a custom UAV platform equipped with an avalanche beacon for conducting rescue missions in avalanche scenarios. It utilizes pre-programmed custom missions and updates the current mission based on environmental readings.

In contrast to the experimental applications presented, studies have also focused on UAV control and path planning in SAR operations. In order to conduct a search guided by the estimate of target location, [7] proposes a UAV path planning method utilizing straight paths and 90-degree turns to cover the target probability distribution in wilderness SAR missions. The research in [8] proposes a multi-UAV control technique based on domain partitioning and task distribution to effectively survey a given area. Similarly, [9] demonstrates the deployment of multiple UAVs to effectively obtain situational awareness following a disaster. It simulates the performance of the proposed UAV control methods in areas affected by a tsunami.

In the context of maritime SAR missions, [10] proposes an algorithm for the generation of intelligent maritime response plans and analyzes its performance using real scenarios. The research presented in [11] introduces a framework for maritime search missions that integrates automatic onboard detections with advanced object recognition performed at ground stations. This approach enhances operator decision-making and increases the efficiency of manually conducted UAV searches.

2.2 Path planning and area coverage

The UAVs can perform missions with different levels of autonomy and human intervention [12]. Autonomous operation refers to the capability of UAVs to perceive the environment, make decisions, and execute actions without human intervention. To accomplish missions autonomously, UAVs rely on path planning algorithms to navigate through the environment. The paths can be computed in real-time [13, 14] or pre-computed before the mission start [15].

In UAV path planning, heuristic methods inspired by natural processes and human reasoning, as well as machine learning and other soft-computing techniques, are widely used as they can effectively tackle complex multi-objective problems [16]. Commonly used approaches include the Genetic Algorithm (GA) [21, 22], Particle Swarm Optimization (PSO) [19, 20], and machine learning methods [17, 18]. Various implementations of PSO for three-dimensional trajectory generation are compared and experimentally tested in [23]. Similarly, the study in [24] utilizes PSO in combination with skeletonization and B-spline curves for trajectory generation over complex topographies. A comparison of PSO and GA for UAV path planning is presented in [25], with the conclusion that the GA algorithm generally performs better in the tested implementation. The study in [26] formulates the energy-aware path planning problem

as a traveling salesman problem and solves it using GA. The results show a comparison with the greedy method, with the GA-based version consuming 2–5 times less energy for UAV operations.

Another common approach for trajectory planning is the Receding Horizon Control (RHC), also referred to as Model Predictive Control (MPC). The method formulates the path planning or control problem at each time step as a constrained optimization problem, calculates the solution over the finite prediction horizon, and applies only the first part of the computed solution [27, 28].

A typical consideration in path planning is achieving a desired inspection performance, often quantified by area coverage. The study in [15] presents a comparison of different motion planners executing a coverage task from a fixed altitude. It provides comparison between simulation results and real-world experiments conducted with fixed-wing UAVs. The study in [29] presents MPC for exploration of unknown environments or a priori known surfaces with micro aerial vehicles and validates the method in real-world experiments. A practical search application is presented in [7], in which a UAV equipped with a radiation sensor is used to locate a radioactive source.

2.3 Multi-agent systems and collision avoidance

A multi-agent system refers to the employment of multiple interacting agents such as UAVs or robots, working together to achieve a collective goal. In such a system, agents can either operate in groups commanded by a leader agent [30, 31], or perform autonomous actions independently [9, 32]. Multi-agent or swarm-based strategies are commonly applied to coverage problems, since they enable more efficient inspection in contrast to single-agent methods [33]. The operational domain can either be shared among all agents [34, 35] or partitioned into sub-domains, with individual agents operating independently within their assigned areas [8, 36]. The system can adopt either centralized [37] or decentralized control [38], each providing specific advantages and disadvantages. In a centralized system, a central unit governs or coordinates the behavior of all agents, while in a decentralized system each agent determines its actions primarily based on local information, without relying on a central controller. Decentralized systems generally offer greater scalability, as the computational load is distributed among the agents. However, they are more complex and face increased challenges in collision avoidance and in

sharing the information required for mission execution. A hybrid solution is proposed in [34], combining global information exchange with locally computed control actions for each agent.

Surrounding environment and collision avoidance should be considered to ensure safe and successful task execution. Several strategies for collision avoidance have been explored in previous studies. In [39], collision avoidance for fixed wing UAVs, controlled using MPC and an artificial potential field, is achieved using points of repulsion. In order to guide UAVs traveling at a constant velocity away from static and moving obstacles, [40] proposes a method utilizing local guidance vector fields. Even greater attention to collision avoidance is required in multi-agent systems. Collision avoidance method for decentralized multi-agent systems is detailed in [41], proposing an asynchronous three-dimensional trajectory planner capable of collision free route generation in environments containing both static and moving obstacles. In practical implementations, collision avoidance can be supported by environmental sensing technologies such as sonar and Light Detection and Ranging (LiDAR) sensors [9, 42].

2.4 Computer vision detection

When equipped with camera sensors, UAVs provide an effective and cost-efficient means of conducting inspection and surveillance operations from an aerial perspective. With the integration of machine learning-based detection algorithms, these systems can automatically identify and localize objects of interest. This combination not only improves the speed and accuracy of detection but also enhances overall operational efficiency while reducing the need for human operators. Furthermore, UAVs gain the capability to function as autonomous aerial observers, able to support tasks such as border surveillance [43], wildlife monitoring [44], and search and rescue [4, 5].

Considering the number of scientific articles published in the UAV domain, Faster Region-based Convolutional Neural Networks (Faster R-CNN) [45] and You Only Look Once (YOLO) [46] are among the most commonly employed methods for object detection in UAV images [47]. Faster R-CNN uses a two-stage process in which candidate objects are identified by a region proposal network in the first step, followed by a classification step and additional post-processing. This pipeline often achieves high detection accuracy in aerial imagery [48, 49], but its modular design can complicate optimization, as each component requires separate training and parameter adjustment. In contrast, YOLO predicts bounding boxes and classifies

objects simultaneously in a single stage by applying a neural network to the entire image in one pass. Although it can struggle with precise localization, especially for small objects, it can detect objects in an image very quickly. Typically, YOLO exhibits higher localization error in comparison to Faster R-CNN, but it tends to produce fewer false positive predictions in background regions [46]. In general, single-shot methods are more suitable for onboard UAV detection systems since they require less memory and are faster [50].

Although images captured using onboard UAV cameras are valuable for many applications, they can introduce various challenges for automatic detection systems. Some of these challenges arise from high-altitude imaging resulting in small objects [51], wide camera angles producing high object density [52], and camera motion responsible for image degradation such as motion blur [53]. However, detection performance on degraded images can be improved if the model is also trained on such images [54]. Detection performance is further affected by reduced resolution, which corresponds to less detail captured in the image [55]. The amount of detail captured is determined not only by the image resolution but also by the altitude at which the image is taken. Increasing the altitude increases area coverage but reduces the amount of detail in the image. The amount of detail captured by the image is determined by Ground Sampling Distance (GSD), which reports the ground area captured by a single pixel. A smaller GSD indicates that more detail is captured. In area mapping, it is used to achieve a specific level of detail [56], and it directly influences the performance of CV detectors [57]. Similarly, various research studies have examined the effect of altitude on detection model performance, generally concluding that detection accuracy decreases as altitude increases [58–61].

2.5 Altitude and velocity control

Efficient UAV area survey can be performed with a proper balance between area coverage and CV detection model performance. This balance is accomplished by conducting the flight mission at a carefully chosen height above the terrain. Furthermore, maintaining a safe minimum distance from the terrain is crucial to avoid potential collisions with natural and man-made obstacles. Therefore, in the presence of uneven terrain, UAV altitude control becomes a highly significant aspect of conducting an effective area survey.

Altitude control can be achieved by following pre-computed paths, as presented in [23].

The study introduces a PSO path generation method producing paths that comply with obstacle avoidance and altitude constraints and demonstrates their performance in real-world experiments. In [62], the authors propose an alternative method for terrain-following trajectory generation, where neural networks are used to produce two-dimensional paths that satisfy constraints including UAV dynamics and flight altitude limits. The study in [63] employs digital surface models to determine flight altitude when generating low-altitude paths that connect multiple points of interest. The method is experimentally tested through UAV missions in which the aircraft followed a sequence of waypoints generated from the computed three-dimensional trajectory.

Terrain following functionality in unknown terrain can be achieved utilizing sparse point clouds generated through Simultaneous Localization And Mapping (SLAM), as presented in [64]. The method generally achieves real-time performance in simulation, but encounters difficulties in areas with indistinct terrain textures and in steep, cliff-like regions. Another terrain following approach is presented in [65], focusing on low-altitude flight achieved by generating the trajectory based on environmental readings from a laser sensor.

Critical considerations in search missions covering extensive areas are UAV energy expenditure and flight range, particularly for multi-rotor UAVs. Range or endurance maximization can be achieved at a velocity specific for each UAV, based on its characteristics such as mass, surface area, battery capacity, and propeller size [66]. Considering search performance, an increase in velocity results in higher area coverage in unit time, but as velocity increases, compliance to constraints such as collision avoidance becomes increasingly difficult [65]. This highlights the need for velocity management, allowing UAVs to reduce their velocity during maneuvers that cannot be executed at the optimal endurance or range velocity. In [67], velocity management of a UAV swarm was implemented utilizing mid-field game approach, with the goal of UAV energy expenditure minimization. A practical application of endurance-aware multi-UAV planning for coverage of disaster areas is presented in [68].

2.6 Ergodic control

The search for missing targets is generally performed based on a belief of their last whereabouts. That belief can be represented as a probability distribution throughout the search domain, where the high probability is assigned to regions that are the most likely to contain the target. With

the utilization of greedy methods [69] the search can be focused on the high-probability areas, which can be really effective, but only if the belief is correct. In the case of high uncertainty around the target's location, these methods tend to fail, as they focus only on the highest-probability areas while disregarding low-probability regions that may still contain the target. On the other hand, coverage path planning techniques systematically covering the whole domain area, such as the spiral or lawnmower path methods [70], are inefficient in the case of a non-uniform probability distribution. To balance the trade-off between exploration (inspection of the entire probability distribution) and exploitation (increased focus on high-probability areas), ergodic methods can be utilized. Over time, they produce a trajectory density corresponding to a given distribution, in this case the target probability. This results in the complete coverage of the probability distribution, where each region is inspected for an amount of time that is proportional to the probability assigned to it. The research in [71] presents several benefits of ergodic methods and demonstrates their robustness to the configuration of initial conditions and better performance in the presence of distractions when compared to alternative methods. For multi-agent ergodic search control, the most commonly used approaches are: Spectral Multiscale Coverage (SMC), MPC, and Heat Equation Driven Area Coverage (HEDAC).

The study in [72] presents a metric for determining the ergodicity of a given trajectory with respect to a specified distribution, and introduces SMC by utilizing the presented metric in a closed feedback loop for multi-agent system control. Building on this, [73] introduces the multiscale adaptive search algorithm by coupling SMC, providing trajectory generation for the sensor, and applying Neyman–Pearson lemma for decision making. The SMC method was applied to a complex, dynamic search scenario in [74], where it was used to simulate the search missions for the MH370 aircraft, lost in the Indian Ocean in 2014.

By incorporating the ergodicity metric in the optimization problem cost function, MPC can also be utilized for ergodic trajectory generation. The overview of MPC methods is presented in [75], highlighting various applications, recent advancements, and the current state-of-the-art. The study in [34] utilizes MPC for real-time execution of a coverage and target localization task. In the implementation, each agent computes its own actions, while the coverage information is shared across a network. A similar method is presented in [38], incorporating changes to the control policy to achieve full decentralization of the system.

The HEDAC method, introduced in [76], provides a solution to the multi agent coverage problem by generating ergodic trajectories through a feedback loop that guides the search agents

based on the potential field generated by a stationary heat equation. Building on the base method, [35] introduces agent sensing and detection, while [77] presents the Finite Element Method (FEM) implementation for numerically obtaining the solution to the heat equation. The application of the method in diverse contexts serves as a proof of its versatility. For example, it was employed for multi-agent maze exploration [37], path planning for complex structure inspection [78], and multi-agent non-uniform crop spraying [32]. The method has also been applied to real-world robotic manipulators performing tasks such as whole body ergodic exploration [79] and curved surface cleaning [80].

2.7 Dynamic environment search strategies

The search of maritime environments introduces new challenges as a result of dynamic target behavior governed by the underlying velocity field of the ocean. A method for oceanic search utilizing UAVs and dynamically changing target probability maps is presented in [82]. The approach applies a Gaussian mixture model for evolving the probability map and incorporates diffusion for velocity field error compensation. A cooperative UAV search framework for dynamic targets is presented in [83], utilizing grid-based domain discretization where each cell is assigned a probability of target presence. The method also accounts for communication imperfections and data loss. To facilitate communication, in [84], Unmanned Surface Vehicles (USVs) and UAVs form a temporary communication network for maritime exploration. The method performs grid-based domain discretization and executes global path planning based on sea information maps, while local path planning relies on data collected by onboard sensors.

The research in [85] presents the motion-encoded genetic algorithm with multiple parents and utilizes it to address the moving target localization problem employing multiple UAVs. The method is grid-based and models target locations with a probability distribution, updating the belief map through a Bayesian approach as sensing data is collected. In contrast to maritime applications, [86] introduces an MPC-based ergodic, multi-agent search strategy that utilizes dynamic information maps and low-information sensors to locate and track moving targets, focusing on land-based scenarios.

3 UAV SEARCH TECHNOLOGY

This chapter provides an overview of modern UAV systems relevant to the context of this thesis. It covers UAV classification, coordinate systems, and introduces control parameters essential for flight management. It provides a brief overview of commonly used sensors and concludes with a description of the experimental UAVs used in the real-world tests.

3.1 Classification of UAVs

The most common way to classify UAVs is by platform type, which refers to their physical structure and how they generate lift. There are three main categories: fixed-wing, rotary-wing, and hybrid UAVs.

Fixed-wing UAVs are similar to traditional planes and they use static wings that rely on forward motion to produce lift. They are very efficient in long-range and high-speed missions and can operate for relatively long periods, making them ideal for inspecting large areas. However, they are impractical for surveying small areas due to their low maneuverability and inability to hover in place. One notable drawback of fixed-wing UAVs is that they require a runway or a catapult for takeoff and a large area to land. A typical example of a fixed-wing UAV is shown in Figure 3.1.



Figure 3.1: Airmobi Skyeye 2600 fixed-wing UAV [87].

Rotary-wing UAVs utilize rotors, assemblies of rotating blades, to generate downward facing thrust which enables them to fly. Since the motors provide constant motion of the rotor

blades, the UAV does not require continuous movement to fly, and can hover in place. However, that requires the motors to run continuously, which increases energy consumption. They are highly maneuverable and capable of taking off and landing in confined spaces. They can execute more complex trajectories when compared to fixed-wing UAVs but they produce more aerodynamic drag, are less energy efficient, and have lower flight endurance. Rotary-wing UAVs can be further categorized by the number of rotors they have into: single-rotor UAVs, which resemble traditional helicopters, and multi-rotor UAVs (quad-rotor, hexa-rotor, etc.). Figure 3.2 shows examples of single-rotor and multi-rotor UAVs.



Figure 3.2: Examples of rotary-wing UAVs: The Rotor single-wing [88] and DJI Phantom 4 Pro multi-rotor [89] UAVs.

Hybrid UAVs combine the characteristics of fixed-wing and rotary-wing UAVs. The most common hybrid-UAV type is the Vertical Take-Off and Landing (VTOL) fixed-wing UAV, which combines the endurance of the fixed-wing aircraft and VTOL capability of rotary-wing UAVs. They are essentially a more advanced version of fixed-wing UAVs, but they have a more complex structure, more parts, and are generally more expensive. An example of a VTOL fixed-wing UAV used primarily for mapping is shown in Figure 3.3.



Figure 3.3: WingtraOne hybrid VTOL UAV [90].

3.2 Control and state parameters

UAVs can be observed from two different reference frames, or coordinate systems, each serving a specific purpose and providing distinct information and usefulness. The inertial frame corresponds to the global or domain coordinate system. It has a fixed origin and is used for describing UAV's position and velocity. The body-fixed frame corresponds to the UAV's local coordinate system which has its origin at the UAV's center of gravity and it moves with the aircraft. It is used to capture UAV's orientation, forces, and control inputs. Furthermore, an additional reference frame can be assigned to the sensor, which typically describes the sensor's orientation relative to the body-fixed frame.

The attitude of the UAV is defined as the orientation of its local (body-fixed) coordinate system relative to the global (inertial) coordinate system, commonly expressed with Euler angles: pitch, roll, and yaw. Pitch is described as rotation about the lateral axis, roll is rotation about the longitudinal axis, and yaw is rotation about the vertical axis. The local coordinate system and attitude angles for fixed-wing and multi-rotor UAVs are illustrated in Figure 3.4.

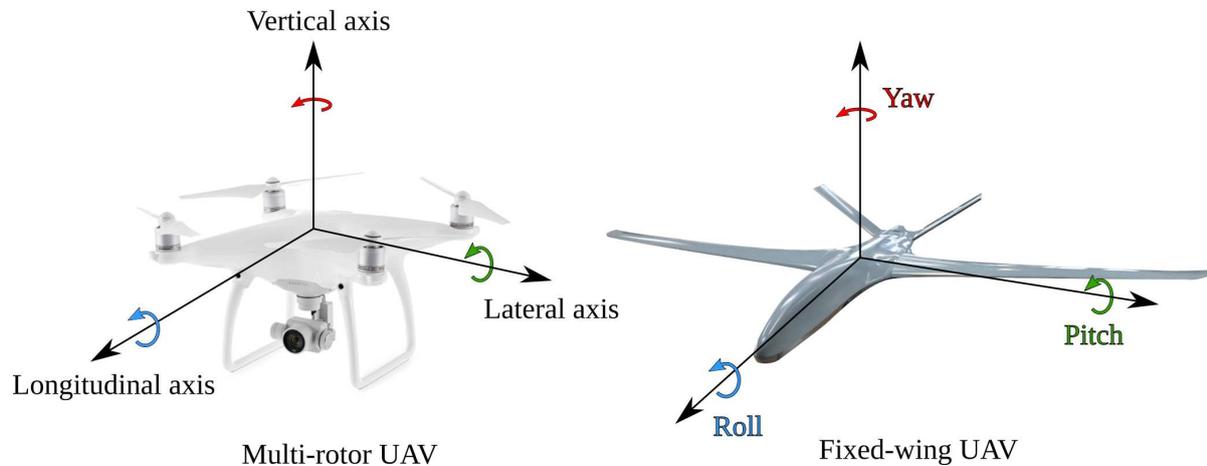


Figure 3.4: Local UAV coordinate system and attitude angles

UAVs are controlled by adjusting thrust levels and the attitude angles. Control of these parameters differs between multi-rotor and fixed-wing UAV types. Multi-rotor UAVs achieve this by varying the rotational speed of their motors (rotors), adjusting thrust individually or collectively to control movement and orientation. The rotors generate downward thrust, so increasing the rotational speed of all motors equally results in a rise in altitude. To move forward or backward, the UAV adjusts its pitch by increasing the speed of the motors at the back or front, respectively. For side to side movement, speed of the side motors is increased. For example,

increasing the speed of the right motors causes the UAV to move left. Standard multi-rotors have an even number of rotors arranged in pairs, with half spinning clockwise and the other half counterclockwise. They are configured in a way that no two adjacent rotors spin in the same direction. To cause yaw rotation, the speed of half the motors rotating in one direction is increased, while the speed of the other half is decreased. This creates a torque imbalance, causing the aircraft to rotate about its vertical axis. Due to the rotor configuration, the overall thrust direction and intensity remain unchanged. Figure 3.5 illustrates a multi-rotor performing pitch, roll, and yaw adjustments.

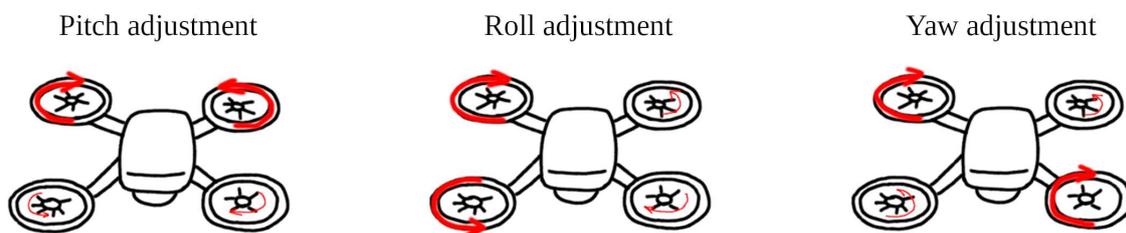


Figure 3.5: Multi-rotor motor power distributions during pitch, roll, and yaw changes. Red arrows indicate motor rotation direction and rotational speed (thicker arrows = higher speed).

High-level control, referring to pilot or auto-pilot level commands, of a multirotor UAV is typically achieved by manually adjusting thrust, pitch, roll, and yaw using a remote controller. When multi-rotor control is handled by a computer, generally, it can be configured in either attitude mode, where thrust and attitude angles (pitch, roll, yaw) are directly adjusted, or velocity mode, where the UAV is commanded to follow specified velocities in the longitudinal, lateral, and vertical directions, along with an angular velocity for yaw.

The motors of fixed-wing UAVs generally provide backward thrust, propelling the aircraft forward. By adjusting the motor power, different thrust levels are produced, resulting in varying forward velocities. Attitude changes in fixed-wing UAVs are achieved by modifying lift and drag forces at specific control surfaces. These forces are adjusted using the elevator, rudder, and ailerons, which are shown in Figure 3.6. To increase or decrease altitude, the aircraft's pitch is adjusted using the elevator, which changes the lift forces at the tail. This alters the wings' angle of attack, resulting in changes to the overall lift force. To increase or decrease altitude, the pitch of the aircraft is adjusted with the elevator changing the lift forces at the aircraft tail, which changes the wings' angle of attack resulting in changes to the aircraft lift force. The

ailerons create a difference in lift between the wings, causing the aircraft to roll. The yaw angle is changed by adjusting the side force generated on the tail by regulating the rudder.

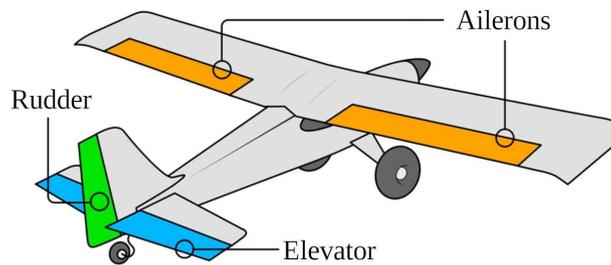


Figure 3.6: Fixed-wing aircraft attitude control components [91].

High level control of fixed-wing UAVs is achieved by adjusting thrust and controlling the elevator, rudder, and ailerons via joystick inputs or the autopilot.

3.3 Onboard sensors

UAVs carry a variety of sensors which they use to collect data while operating. They can be split into two groups considering their function into navigation and perception sensors.

By combining information gathered from all the navigation sensors, the UAV flight controller can perform state estimation, determining the UAV's attitude, position, velocities, and angular velocities. The state estimate is essential for stabilization, navigation, path planning, and performing autonomous missions. The most essential navigation sensors include the Inertial Measurement Unit (IMU), Global Positioning System (GPS) receiver, magnetometer, and barometer. The IMU is a combination of accelerometers, which measure linear acceleration, and gyroscopes, which measure angular velocity. These measurements are used to estimate UAV's attitude and motion. The GPS receiver is used to determine the UAV's global position (latitude, longitude, and altitude). It typically has a low update rate, when compared to the IMU, and may be unusable indoors. Standard GPS has an accuracy of a couple of meters while Real-Time Kinematic (RTK) GPS is accurate to a few centimeters. The magnetometer acts like a digital compass, which measures the Earth's magnetic field to determine the UAV's global heading (yaw angle) relative to north. It can be sensitive to electromagnetic interference coming from onboard electronics or external sources. The barometer is used in a barometric altimeter to measure air pressure, which is used to calculate the altitude relative to a local reference, usually

the UAV's takeoff point. It provides a reliable estimate of relative altitude, but its accuracy can be affected by pressure changes due to weather or nearby airflow.

Perception sensors are used to gather information about the surrounding environment. They enable obstacle detection and avoidance, navigation, terrain mapping, and object detection. Common UAV perception sensors include: standard Red-Green-Blue (RGB) cameras, infrared (thermal) cameras, multispectral and hyperspectral cameras, Light Detection and Ranging (LiDAR) sensors, laser rangefinders and ultrasonic sensors. Figure 3.7 illustrates several perception sensors that can be mounted on UAVs, along with their data outputs.

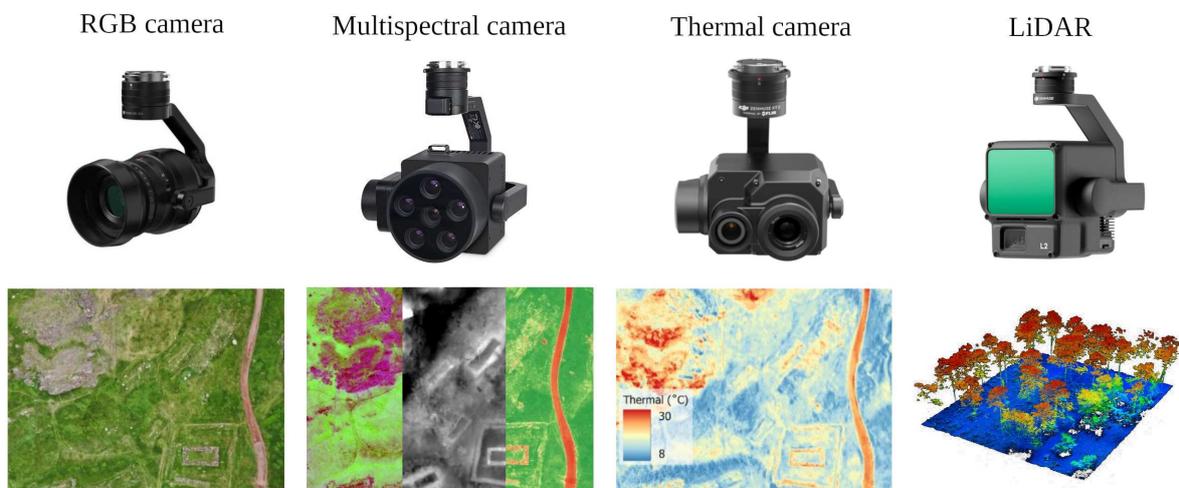


Figure 3.7: Range of UAV perception sensors [92–95] and examples of their data outputs [96, 97].

Standard RGB cameras capture images or video in the visible spectrum. Their video feed is typically transmitted to the pilot as the primary source of visual information for operating the aircraft, often at reduced quality due to limited bandwidth. The captured data can be used for visual inspection, mapping, and object detection. Thermal cameras are used to capture the infrared radiation, producing images that show the temperature distribution of the observed area. They are commonly used in night operations, SAR, and fire detection. Multispectral cameras capture images containing a few discrete spectral bands which are selected to target specific features. They are often used in environmental monitoring and agriculture to estimate crop health. Hyperspectral cameras are similar to multispectral cameras, but they capture many more bands which are narrower and closely spaced. This results in more complex data, which is generally used in scientific research or for detailed material analysis. LiDAR sensors emit a large amount of laser pulses which are used to measure distance between the sensor and the surfaces they

hit. The resulting data forms detailed point clouds that can be used for terrain mapping and obstacle detection. Due to the high volume of data generated, LiDAR systems typically require a powerful onboard computer for real-time processing. Laser rangefinders operate on the same principle as LiDAR systems, but they provide only a single-point measurement. They offer accurate distance measurements and are often used to determine a UAV's height above the terrain. Ultrasonic sensors utilize high-frequency sound waves to measure distance of nearby surfaces. They have low computational requirements and can operate in real-time, but have a limited range of a couple of meters. They are typically used for indoor obstacle avoidance, low-altitude altitude hold and assisting with takeoff and landing.

3.4 Experimental UAV system overview

The UAVs used in the experimental part of this thesis are commercially available models manufactured by DJI. Specifically, the models used are the DJI Matrice 210 V2 and the Mavic 2 Enterprise Dual, as shown in Figure 3.8.



DJI Matrice 210 v2 with
Zenmuse X5S Camera



DJI Mavic 2 Enterprise Dual

Figure 3.8: Experimental UAVs used in real-world testing [92, 98, 99].

The Matrice 210 v2 UAV is paired with the DJI Zenmuse X5S RGB camera, which captures images at a resolution of 5280×2970 pixels with a 16:9 aspect ratio. The Mavic 2 Enterprise Dual features an integrated RGB and thermal camera, but only the RGB camera is used in this research. It captures images at a resolution of 4056×3040 pixels with a 4:3 aspect ratio. Table 3.1 summarizes the technical characteristics of the experimental UAVs, based on manufacturer documentation. It should be noted that the maximum flight times specified by the manufacturer (Table 3.1) are often calculated considering ideal conditions and likely exclude

any payload. In realistic operating environments, the actual flight times can be significantly lower.

Table 3.1: Overview of technical specifications for the UAVs used in field experiments

Specification	DJI Matrice 210 V2	Mavic 2 Enterprise Dual
Weight (without payload)	4.8 kg	899 g
Dimensions	883 × 886 × 398 mm	322 × 242 × 84 mm
Max flight time	34 min	31 min
Max speed (P-mode)	17 m/s	14 m/s
Max ascent speed	5 m/s	5 m/s
Max descent speed	3 m/s	3 m/s
Max angular velocity (P-mode)	120°/s	100°/s
Obstacle sensing	Forward, backward, downward sensors	Forward, downward, backward
Payload capability	Supports multiple payloads	Integrated dual camera (RGB + Thermal)

To enable autonomous control managed by a Ground Control Station (GCS) computer, a communication system was implemented to handle data exchange between the PC and the UAVs. For any type of flight, each UAV must be operated with a remote controller and a smart mobile device running the official "DJI Pilot" or a custom DJI-based application. Communication between the UAV and the remote controller is established via radio signals, while the remote controller is connected to the tablet through a wired link.

To allow PC-based autonomous control, a custom Android application was developed using the DJI Software Development Kit (SDK). This application is connected to the GCS, where commands are computed and then forwarded to the UAV. Likewise, state updates from the UAV are captured and transmitted back to the PC, enabling real-time bidirectional communication. The overall communication scheme is illustrated in Figure 3.9.

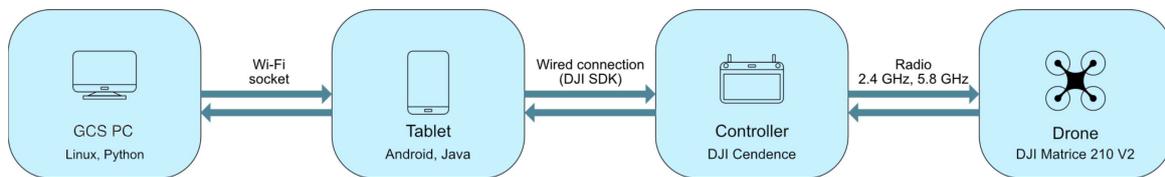


Figure 3.9: Overview of the communication scheme between the PC and the UAVs [100].

4 MODELING UAV MOTION, COMPUTER VISION SENSING, AND DETECTION

This chapter presents the components describing the UAV control parameters, along with the detection and probabilistic model for the search. The UAV motion is first characterized through a motion model, where the kinematic relationships and control parameters governing its trajectory are defined. This is followed by a summary of search theory and an introduction to the undetected target probability, which forms the basis of the search. Next, the modeling of a realistic UAV sensor is presented, describing its spatial and detection characteristics as well as its probabilistic nature. The integration of the computer vision detection model is then presented, with its detection performance quantified through relevant metrics, incorporated into the sensing function. Following this, the collaborative sensing effect of all search agents in the static target search scenario is outlined. Lastly, a measure for evaluating search success is defined, providing a quantitative assessment of the search effectiveness over time.

4.1 Motion model

To conduct the search, multiple UAVs operate within a predefined domain. Each UAV performing the search is referred to as a search agent. The total number of search agents is denoted by n , and each agent is identified by a unique index i . They move in a three-dimensional domain $\Omega_{3D} \subset \mathbb{R}^3$, and their motion is controlled by regulating the following variables in time t :

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

The velocity intensity ρ expresses the proportion of the maximum absolute velocity that is currently being utilized. Since fixed-wing UAVs, in contrast to multi-rotor UAVs, require a

certain amount of horizontal velocity to generate lift and avoid stalling, the minimum horizontal velocity $v_{s,min}$ is introduced. It effectively constraints ρ to the minimum value of $\rho_{min} = v_{s,min}/v_{s,max}$, where $v_{s,max}$ denotes the maximum horizontal velocity of the UAV. The minimum and maximum horizontal velocities must satisfy the conditions: $v_{s,min} \geq 0$ and $v_{s,max} > v_{s,min}$. In the case of multi-rotor UAVs, which can hover in place, $v_{s,min} = 0$, and therefore $\rho_{min} = 0$.

The incline parameter φ represents the angle between the horizontal plane and the velocity vector which is tangential to the UAV's trajectory. It should be noted that the aircraft pitch differs from the incline angle φ which represents the slope of the resulting trajectory. The pitch denotes the angle between the horizontal plane and the longitudinal axis of the aircraft. Additionally, possible lateral motion resulting from multi-rotor roll adjustments is not considered.

To accurately capture the clear differences in velocity characteristics UAVs display during horizontal flight, ascent, or descent, the limit velocity function $v(\varphi)$ is introduced. This function denotes the highest absolute velocity achievable by the UAV at a given incline. It offers a clear description of the UAV's velocity characteristics and can be easily experimentally determined for a given aircraft. To reduce complexity, $v(\varphi)$ is approximated with an (asymmetric) ellipse defined by three characteristic velocities: maximum ascending velocity $v_{z,max} \equiv v(\varphi_{max})$, maximum horizontal velocity $v_{s,max} \equiv v(0)$, and maximum descending velocity $v_{z,min} \equiv v(\varphi_{min})$. As presented in Figure 4.1, the horizontal and vertical velocity components are indicated by the green and red vectors, respectively, while the total velocity is represented by the orange vector. The total velocity vector originates from the aircraft's center of mass and the feasible region for its tip is represented with area shaded in light blue color, while the velocity limits associated with various inclines are illustrated with the red line.

Depending on the mission objectives and operating conditions, it may be more beneficial to use the optimal range or endurance velocity, considering UAV energy constraints [66]. Additionally, since CV detector performance depends on the flight velocity at which UAV-mounted camera captures images [101], the UAV velocity can be set up to achieve the best balance between area coverage and detection performance. In such cases, the parameters $v_{s,max}$, $v_{z,max}$, and $v_{z,min}$ can be defined based on the velocities that maximize endurance, range, or detection rather than the UAV's performance limits.

Additional constraints are introduced to realistically model the dynamic behavior of various UAV types. For fixed-wing UAVs, the range of φ values is constrained with φ_{min} and φ_{max} in order to prevent aircraft stall, which typically happens around $15^\circ \approx 0.26$ rad. Assuming

that the angle of attack (the angle between the wing chord line and the oncoming airflow) is closely aligned with the aircraft incline, φ limits within $[-0.25, 0.25]$ can be imposed. Since the minimum absolute velocity is constrained by ρ_{min} , as φ deviates from 0, it is possible to slightly breach the prescribed minimum horizontal velocity $v_{s,min}$. However, this is considered insignificant due to the realistic φ limits which ensure that the velocity does not drop more than 4% below $v_{s,min}$ and can be compensated by more conservative value of ρ_{min} .

In contrast, multi-rotor UAVs are capable of performing purely vertical motion, without movement in the horizontal plane. Therefore, φ limits are imposed with $\varphi_{min} = -\pi/2$ corresponding to vertical descent and $\varphi_{max} = \pi/2$ representing vertical ascent. Additionally, multi-rotor UAVs can fly backward, which potentially adds additional motion and search flexibility. However, this is not considered due to practical reasons and better awareness of pilots supervising the autonomous flight.

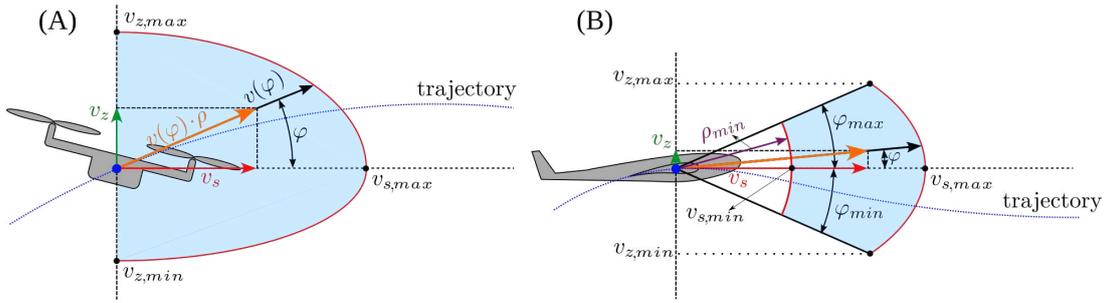


Figure 4.1: Velocity components and constraints for multi-rotor (A) and fixed-wing (B) UAVs.

Based on this framework, the horizontal and vertical UAV velocity components are formulated as functions of control parameters:

$$v_{s,i}(\rho_i, \varphi_i) = \rho_i \cdot v_i(\varphi_i) \cdot \cos(\varphi_i), \quad (4.1)$$

$$v_{z,i}(\rho_i, \varphi_i) = \rho_i \cdot v_i(\varphi_i) \cdot \sin(\varphi_i). \quad (4.2)$$

The UAV trajectory $\mathbf{Y}_i(t) = [x_i(t), y_i(t), z_i(t)] \subset \Omega_{3D}$ is obtained by solving the equations of motion while accounting for the different components of the realized UAV velocity. Accordingly, the relations can be written as:

$$\frac{dx_i}{dt} = v_{s,i}(\rho_i, \varphi_i) \cdot \cos \theta_i,$$

$$\frac{dy_i}{dt} = v_{s,i}(\rho_i, \varphi_i) \cdot \sin \theta_i,$$

$$\frac{dz_i}{dt} = v_{z,i}(\rho_i, \varphi_i),$$

where θ is the heading angle governed by the yaw angular velocity

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

The implementation of the ω constraints within the control algorithm is detailed later in Section 6.1.

4.2 Theory of search

In the case of continuous observation of a single target, let Γdt represent the probability of detecting the target in a short time interval dt . Here, Γ represents the instantaneous detection probability (of the target). According to Koopman [102], the probability of detecting the target under unchanging conditions, when the target is observed for the time t , is given by

$$p(t) = 1 - e^{-\Gamma t}, \quad (4.3)$$

where $\Gamma \geq 0$.

The probability that the observed target is not detected during observation time t is given by $q(t) = 1 - p(t)$. Before any observations occur, equation (4.3) implies that the target is certainly undetected, since $p(0) = 0$, and therefore $q(0) = 1$. As outlined in [102], q varies with time according to the differential equation

$$\frac{dq}{dt} = -\Gamma q. \quad (4.4)$$

In a search scenario, the location of the target is unknown and the search is conducted based on the belief about the target's location. Let \mathbf{x} represent an arbitrary point in the two-dimensional search domain Ω_{2D} . As given by [35], this belief is represented by the undetected target probability density field $m(\mathbf{x}, t)$, which describes the spatial probability density of undetected target occurrence over time t . At time $t = 0$, prior to any observations, the undetected

target probability density is denoted by $m_0(\mathbf{x})$. Since the target is initially undetected, m_0 represents the spatial probability density of the target's presence at any location \mathbf{x} . With the assumption that the target is within the search domain at $t = 0$, the initial target probability distribution accumulated over the entire domain area Ω_{2D} must equal 1. Therefore, m_0 is required to satisfy the condition

$$\int_{\Omega_{2D}} m_0(\mathbf{x}) \, d\mathbf{x} = 1. \quad (4.5)$$

At time t , the undetected target probability density m is obtained by combining the initial spatial probability density of target presence with the probability of the target remaining undetected. Since a potential target may be located at any position \mathbf{x} , the sensing performance varies across the domain, making q location dependent. Accordingly, the undetected target distribution is modeled as

$$m(\mathbf{x}, t) = m_0(\mathbf{x}) \cdot q(\mathbf{x}, t).$$

Differentiating this expression with respect to time and substituting equation (4.4) yields the governing equation for the temporal evolution of m , given by

$$\frac{dm(\mathbf{x}, t)}{dt} = -\Gamma m(\mathbf{x}, t). \quad (4.6)$$

The probability of an undetected stationary target at \mathbf{x} after an observation time t is given by the analytical solution of equation (4.6), expressed as

$$m(\mathbf{x}, t) = m_0(\mathbf{x}) \cdot e^{-\Gamma t}. \quad (4.7)$$

If Γ varies with time, the probability of the target at \mathbf{x} remaining undetected after an observation time t , according to equation (4.7), is given by

$$m(\mathbf{x}, t) = m_0(\mathbf{x}) \cdot e^{-\int_0^t \Gamma(\tau) \, d\tau}. \quad (4.8)$$

In a UAV-based search, observations are performed by an onboard sensor. The UAV moves over a two-dimensional domain, carrying the sensor along with it. Each point \mathbf{x} in the domain can be represented in the sensor's coordinate system as a vector $\mathbf{R}(\mathbf{x}, t)$, which varies over time as the UAV moves. If Γ is considered a function of \mathbf{R} , the probability that the target at \mathbf{x} remains

undetected over an observation period t is determined from equation (4.8) as

$$m(\mathbf{x}, t) = m_0(\mathbf{x}) \cdot e^{-\int_0^t \Gamma(\mathbf{R}(\mathbf{x}, \tau)) d\tau}. \quad (4.9)$$

Although the equation (4.9) describes the continuous observation, or sensing performance, the model can also be applied to discrete sensing with the assumption that Γ does not vary significantly during the sensing interval Δt_s , corresponding to observation time. In the case of discrete sensing using a camera sensor, images or frames are captured at fixed intervals. Even video, which appears continuous to the human eye, is fundamentally discretized into multiple successive frames. To simulate continuous sensing performance, each captured frame or image is assumed to be observed over the duration of the sensing interval Δt_s , corresponding to the time between successive frames. By increasing the number of frames captured within the same time interval, the instantaneous detection probability function is evaluated more frequently over shorter periods, while preserving the same overall sensing effect i.e. the total detection probability.

4.3 UAV sensor modeling

While UAVs operate in three-dimensional space, their main objective is to explore the terrain or sea surface. Although the terrain is not perfectly flat, the inspection of the terrain surface can reasonably be treated as a two-dimensional search problem, since the vertical variations are small compared to the horizontal extent. Consequently, the search problem is reduced to a two-dimensional horizontal domain $\Omega_{2D} \subset \mathbb{R}^2$. Let $\mathbf{x} \in \Omega_{2D}$ represent an arbitrary point in the two-dimensional domain. The surface topography is described by a elevation function $z_T : \Omega_{2D} \rightarrow \mathbb{R}$, $\mathbf{x} \mapsto z_T(\mathbf{x})$. In the case of land-based operations, the terrain function can be obtained through various sources, such as Digital Elevation Models (DEMs), terrain reconstruction from images captured at different perspectives (photogrammetry), or LiDAR point clouds. For maritime search missions, the sea surface is flat, and consequently, $z_T = 0$.

Inspection is conducted using a sensor mounted onboard a moving UAV. Although they are generally not identical, for this application it is correct to assume that the origin of the sensor's coordinate frame coincides with the UAV position, as defined by \mathbf{Y} , representing the origin of the UAV coordinate frame. The sensor is considered to be mounted on a gimbal that compensates for the UAV's pitch and roll, ensuring that the sensor direction remains aligned with the

vertical (nadir) axis. In the case of a camera sensor, this implies taking orthophoto images. Consequently, the sensor maintains a fixed pitch of -90° relative to the global coordinate frame. The sensor yaw is fixed relative to the UAV body frame, with a value of 0° , meaning that the sensor heading aligns with the UAV heading θ .

The sensor captures information encompassed inside its Field Of View (FOV) $\Omega_{\text{FOV}} \subset \mathbb{R}^3$. Two sensor models are considered: a generic sensor represented by a conical FOV and a real camera sensor modeled by a pyramidal FOV. The conical FOV, employed in several simulated search scenarios, is characterized only by its apex angle γ_c , defining the sensor's angular coverage. The pyramidal FOV represents a camera sensor capturing rectangular images. It is defined by two angles, γ_1 and γ_2 , which correspond to the angles between lateral and longitudinal faces of the FOV pyramid relative to the UAV's heading, respectively. The γ angles for both the conical and pyramidal FOVs are illustrated in Figure 4.2. Accounting for terrain elevation variability and flight altitude, the data collected by the sensor is provided for the terrain surface enclosed within the FOV. The information recorded by the sensor are images of the terrain surface in the case of a camera, but may also comprise any type of surface information acquired by an arbitrary sensor.

The apex of the FOV coincides with the origin of the sensor's coordinate frame, corresponding to \mathbf{Y} . Within the sensor's coordinate system, points are represented by the vector \mathbf{R} . Instantaneous detection probability is zero for all points outside the sensor's FOV or that are visually obstructed, as determined via ray tracing. Evaluating the sensing effect of the sensor located at \mathbf{Y} on an arbitrary point $\mathbf{x} \in \Omega_{2D}$ requires a transformation into the local coordinate system

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

involving a three-dimensional translation and rotation in the horizontal plane based on the UAV's heading angle θ , with \mathbf{x} projected onto the terrain surface using $z_T(\mathbf{x})$. Figure 4.2 illustrates the global and local UAV coordinate frames along with an example of the vector \mathbf{R} .

In a practical scenario, the instantaneous detection probability Γ of a target at \mathbf{x} depends on the Euclidean distance between the sensor and the observed point, given by $\|\mathbf{R}_i\|$. The effect of Γ is limited to the area observed by the sensor through the sensing function ψ , which is

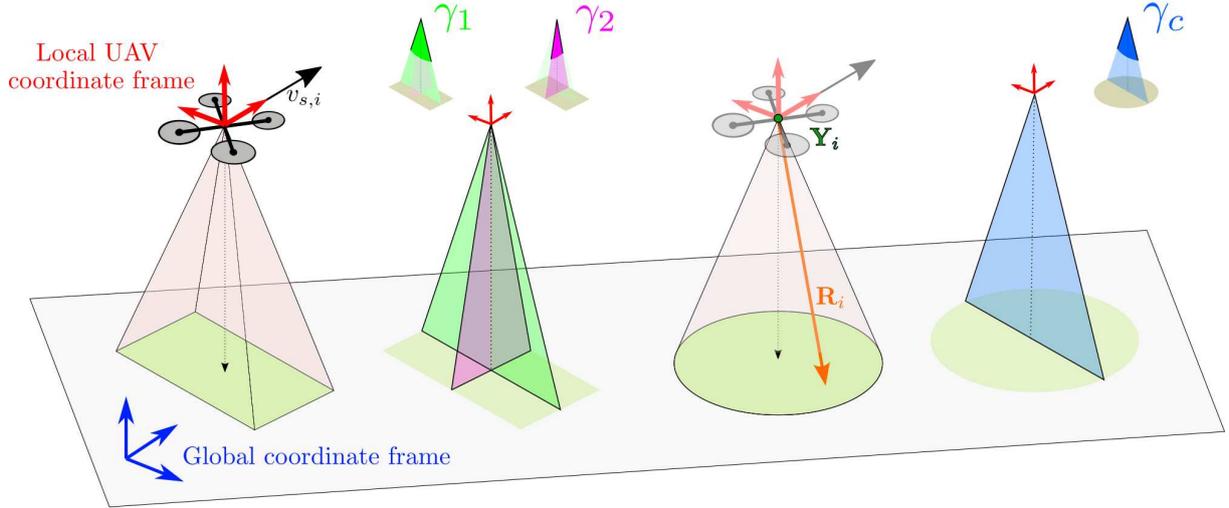


Figure 4.2: Representation of the pyramidal and conical FOV angles (γ), along with the global and local UAV coordinate frames and an example of vector \mathbf{R} .

expressed as

$$\psi_i(\mathbf{R}_i) = \begin{cases} \Gamma_i(\|\mathbf{R}_i\|) & \text{if } \mathbf{R}_i \in \Omega_{\text{FOV},i} \\ 0 & \text{otherwise.} \end{cases} \quad (4.10)$$

A UAV employing a generic ψ function is illustrated in Figure 4.3. Although the function operates over the two-dimensional domain Ω_{2D} , its effect is shown on the terrain surface to illustrate how the detection rate changes with distance. The terrain outside the FOV and the terrain occluded from view (i.e., without a clear line of sight to the sensor) are not affected.

4.4 Computer vision detection

To link the described sensing model with CV detection models used in practice, relevant performance metrics of the detection model are embedded into the sensing framework. In this context, recall μ is used, representing the proportion of correctly detected objects relative to the total number of objects present in an image. Since recall measures the effectiveness of the detection model in identifying all objects instances, it is treated as the probability of detecting an object, such as a missing person.

The CV detection model is usually computed for a set of still images or extracted video frames, while the defined instantaneous detection probability Γ expresses sensing performance

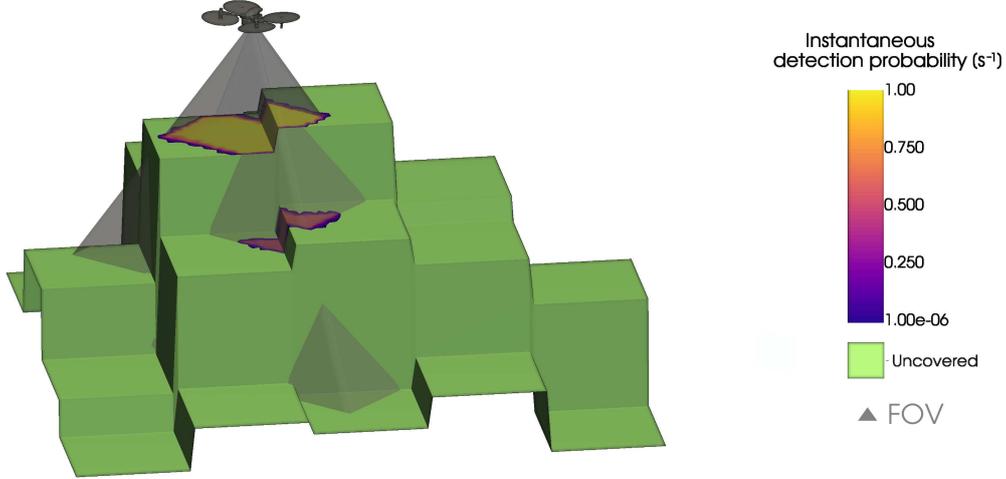


Figure 4.3: Visualization of sensing function ψ .

over time. To establish a connection between μ and Γ , the parameter t_{scene} is introduced, representing the duration over which the scene is considered unchanged, or the time required for significant change to occur. The detection probability that corresponds to the recall value needs to be accumulated at the sensed point over the duration of t_{scene} through the application of Γ . In the static target search scenario, the scene change is assumed to be the primarily caused by camera movement rather than object motion within the observed scene. Consequently, t_{scene} is approximated as the average duration an arbitrary point remains within the camera's FOV during a single flyover. To calculate t_{scene} , it is assumed that the UAV operates over a flat surface at a constant height corresponding to the goal search height h_{goal} , while maintaining an average horizontal velocity $v_{s,avg}$. It can be expressed as

$$t_{scene,i} = \frac{2 \cdot h_{goal,i} \cdot \tan \frac{\gamma_i}{2}}{v_{s,avg,i}}. \quad (4.11)$$

where

$$\gamma_i = \begin{cases} \gamma_{2,i}, & \text{for a pyramidal FOV,} \\ \gamma_{c,i}, & \text{for a conical FOV.} \end{cases}$$

With the assumption that the instantaneous detection probability does not vary significantly over short time intervals, and given that recall varies with altitude [58, 60, 61], recall μ can be

defined from (4.3) as

$$\mu(\|\mathbf{R}\|) = 1 - e^{-\Gamma(\|\mathbf{R}\|) \cdot t_{scene}}. \quad (4.12)$$

Based on the known recall values, which depend on the object's distance from the sensor, the instantaneous detection probability function is determined from (4.12) as

$$\Gamma(\|\mathbf{R}\|) = -\frac{\ln(1 - \mu(\|\mathbf{R}\|))}{t_{scene}}. \quad (4.13)$$

In cases where scene changes result from both camera motion and dynamic elements within the frame, t_{scene} can be adjusted to capture the combined effect of both camera and object motion.

4.5 Combined sensing effect in static target search

To effectively search the area, information about the most probable location of the search target is required. This information is represented by the undetected target probability density function m , as presented in Section 4.2. The function $m(\mathbf{x}, t)$ defines the probability that a target exists at position \mathbf{x} and remains undetected at time t . Its computation differs depending on whether a static or dynamic probability model is used. In the static model, m evolves only due to the effects of sensing and is not influenced by other external factors. The formulation in this section is presented for the static case, while m will be extended to include external dynamics later in Chapter 10.

In order to determine the search performance achieved during the mission, the sensing effect is applied to the domain. For the definition of the sensing process, the UAV trajectories $\mathbf{Y}_i(t)$ are assumed to be known. The combined sensing effect of all UAVs until time t , for the static target case, is assembled as a convolution of sensing ψ along UAV flight paths. This accumulation is referred to as coverage density, and defined as:

$$c(\mathbf{x}, t) = \int_0^t \sum_{i=1}^n \psi_i(\mathbf{R}_i) d\tau.$$

Note that \mathbf{R} implicitly depends on the UAV position $\mathbf{Y}(t)$ and orientation $\theta_i(t)$, for every domain point \mathbf{x} . The probability density of undetected stationary target presence at any point \mathbf{x} and time

t is described with undetected target probability density field

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

Here, $m_0(\mathbf{x})$ represents the probability that a stationary target exists at \mathbf{x} , and $e^{-c(\mathbf{x}, t)}$ represents the probability that a target at \mathbf{x} has not been detected during time t .

The undetected target probability density m at any given time combines the assumed initial probability density of the target with the probability of detection resulting from the search conducted up to that point. This constitutes an exact probabilistic model for the search of a stationary target.

4.6 Search task and search evaluation

The ergodic search system is configured to dynamically regulate $\omega(t)$ over time in order to directly minimize undetected target probability density m across the search domain. Since the search time window is unspecified, in other words, the search continues until the target is found, the objective corresponds to the asymptotic definition of ergodicity. Accordingly, an ergodic search task can be defined as minimization of the spatial probability of undetected target occurrence

$$\lim_{t \rightarrow \infty} \int_{\Omega_{2D}} m(\mathbf{x}, t) d\mathbf{x} = 0, \quad (4.14)$$

where the sensing effort asymptotically approaches the initial undetected target distribution over time.

To assess the overall search performance, the function m is integrated over the entire domain to compute the survey accomplishment metric

$$\eta(t) = 1 - \int_{\Omega_{2D}} m(\mathbf{x}, t) d\mathbf{x}. \quad (4.15)$$

The ergodic task, equivalent to (4.14), can also be described as maximization of the survey accomplishment metric with η in time, asymptotically achieving

$$\lim_{t \rightarrow \infty} \eta(t) = 1.$$

It should be noted that, although the search cannot continue indefinitely, its duration can not be predetermined. Due to this uncertainty, the dynamic exploration of the probability distribution cannot be formulated as a conventional optimization problem and is instead treated as an ergodic search task.

5 ERGODIC CONTROL

In the context of this thesis, the ergodic methods are employed as a key component in the UAV control feedback loop. The ergodic method determines how the two-dimensional search space Ω_{2D} (a horizontal plane) should be explored by computing the UAVs' heading direction. Since the UAVs cannot instantaneously change their orientation due to motion dynamics, the desired heading is converted into yaw angular velocity $\omega_i(t)$. The obtained $\omega_i(t)$ serves as the UAVs' control parameter, determining the appropriate change in UAV orientation.

5.1 Heat Equation Driven Area Coverage – HEDAC

HEDAC is an ergodic search method that employs the heat equation to guide agents toward an imaginary heat source. As the agents move along the simulated temperature gradient toward regions of higher heat, their sensing effectively cools heat source. Essentially, the HEDAC method aims to minimize the source term in the stationary heat equation.

It was initially introduced in [76], and applied to multi-agent spraying task in [32], where it was originally formulated to drive the exploration by using the non-negative difference between the goal density ϕ and the achieved coverage density c ,

$$\max(\phi(\mathbf{x}) - c(\mathbf{x}, t), 0),$$

as the source term in the heat equation. However, this formulation is not appropriate for the direct minimization of m .

The approach was later extended in [35] through the integration of a probabilistic search model, where the source term was directly defined with the undetected target probability density $m(\mathbf{x}, t)$. This formulation supports direct ergodic exploration of the underlying probability distribution, leading to near-optimal search behavior. It is employed throughout the thesis and directly corresponds to the ergodic search task defined in (4.14).

Following the approach in [35], the potential field $u(\mathbf{x}, t)$ is used to direct the UAVs toward regions of the highest undetected target probability. In the context of the heat equation, u represents temperature. However, to avoid possible confusion with physical phenomena unrelated to the present study and to align with other methods, it will be referred to as potential throughout this work. It is obtained by computing the solution to the partial differential equation

$$\alpha \cdot \Delta u(\mathbf{x}, t) = \beta \cdot u(\mathbf{x}, t) - m(\mathbf{x}, t), \quad (5.1)$$

under the imposed Neumann zero-flux boundary condition

$$\frac{\partial u}{\partial \mathbf{n}} = 0, \quad (5.2)$$

with \mathbf{n} defined as the outward normal vector on the domain boundary $\partial\Omega_{2D}$. The parameters $\alpha > 0$ and $\beta > 0$ are used to influence the search behavior. The parameter α , interpreted as a conduction coefficient, controls the smoothness of the probability field and therefore determines whether the search focuses on local areas or explores more globally. In contrast, β mainly serves to ensure numerical stability and has little effect on the search behavior.

The heading direction that the UAV needs to follow from any point \mathbf{x} to pursue the ergodic task is defined by the vector field

$$\mathbf{u}(\mathbf{x}, t) = \frac{\nabla u(\mathbf{x}, t)}{\|\nabla u(\mathbf{x}, t)\|}. \quad (5.3)$$

Following the gradient of the potential ∇u , the UAVs, or search agents in general, are guided towards the higher values of the underlying probability density m . The change in heading over the interval Δt is determined by the angle between the current direction vector $\mathbf{v}_i(t)$ and the vector $\mathbf{u}(\mathbf{y}_i, t)$. To change the heading direction, the UAV performs a yaw rotation with the angular velocity

$$\omega_i^E = \frac{1}{\Delta t} \arccos \left(\frac{\mathbf{v}_i(t) * \mathbf{u}(\mathbf{y}_i, t)}{\|\mathbf{v}_i(t)\|} \right), \quad (5.4)$$

where $*$ denotes the dot product, and \mathbf{y}_i the current position of the UAV.

5.2 HEDAC implementation

When the method was first introduced and in several later works [32, 35, 76], the HEDAC partial differential equation (5.1) was solved using the finite difference method. In [77], the Finite Element Method (FEM) was introduced instead, bringing three main improvements. The first benefit is that FEM makes it possible to solve (5.1) on non-rectangular domains. This removes the need to embed irregular domains within larger rectangular grids, which was previously necessary and resulted in unnecessary computation. The second advantage is that FEM enables strict enforcement of domain boundaries. In earlier implementations, obstacles and regions outside the area of interest were assigned a search importance of zero by setting $m = 0$. These regions were mostly avoided by the UAVs due to no search benefit, but there was no hard constraint preventing the UAV from passing through them to reach other parts of the domain. With FEM, the inner domain boundaries can be explicitly defined, allowing the implementation of constraints that ensure UAVs remain within the domain and do not enter obstacle regions. Finally, the third benefit is FEM's built-in calculation of the potential and its gradient at any point in the domain, eliminating the need for additional interpolation or gradient computation methods.

Due to its advantages, FEM is chosen to solve for potential u in the search control procedures within this study. FEM is used to numerically approximate the solution of a given PDE on the computational domain. In order to solve the PDE (5.1) using FEM, it is multiplied by a smooth test function $w \in H^1(\Omega_{2D})$ and integrated over Ω_{2D} . If the Neumann boundary condition defined in equation (5.2) is applied, the weak formulation of the PDE is obtained and is given by equation

$$\int_{\Omega_{2D}} \alpha \nabla u(\mathbf{x}, t) \cdot \nabla w(\mathbf{x}) + \beta u(\mathbf{x}, t) w(\mathbf{x}) - m(\mathbf{x}, t) w(\mathbf{x}) d\Omega_{2D} = 0.$$

The continuous domain space Ω_{2D} is then discretized into N non-overlapping triangular elements $\{T_1, \dots, T_N\}$. Each element is defined in a local, simplified coordinate system and is mapped to the domain using an affine transformation, preserving the straight element edges. Over each node of the element T , the scalar field of the unknown variable u and the test functions are approximated by

$$u_h = \sum_{j=1}^{n_{nodes}} N_j u_j \quad (5.5)$$

where N_j are the Lagrange polynomial interpolation functions and u_j are the nodal values. In this implementation the second order polynomial degree is used, along with triangular elements discretized with 6 nodes, three vertices and three mid-edge nodes.

To approximate the solution over the whole domain, the system of linear equations is defined with

$$\mathbf{K}\mathbf{u} = \mathbf{F}.$$

Because the PDE coefficients do not vary in time, the system matrix \mathbf{K} is constant for all time steps. Consequently, its inverse can be computed once during algorithm initialization, and the solution at each time step can be reduced to a matrix-vector multiplication given by

$$\mathbf{u} = \mathbf{K}^{-1}\mathbf{F}.$$

However, matrix inversion is computationally inefficient. Efficiency can be improved by applying matrix factorization, for example, LU decomposition ($\mathbf{K} = \mathbf{L}\mathbf{U}$). Solving the resulting triangular systems proceeds with a forward substitution for \mathbf{n} :

$$\mathbf{L}\mathbf{n} = \mathbf{F},$$

and a backward substitution for \mathbf{u} :

$$\mathbf{U}\mathbf{u} = \mathbf{n},$$

Given that \mathbf{K} is sparse, the computation can be further accelerated by using specialized frameworks for sparse linear algebra.

5.3 Spectral Multiscale Coverage – SMC

The SMC method, introduced in [72], computes ergodic trajectories by minimizing the difference between the achieved coverage density $c(\mathbf{x}, t)$ and the goal density $\phi(\mathbf{x})$, using the weighted Sobolev norm

$$\|c(\mathbf{x}, t) - \phi(\mathbf{x})\|_{H^s}^2 = \sum_{\mathbf{k} \in \mathbb{Z}^{*n_s}} \Lambda(\mathbf{k}) |c_{\mathbf{k}}(t) - \phi_{\mathbf{k}}|^2$$

where s is the Sobolev index, n_s is the spatial dimension, \mathbf{k} is the wave-number vector, and $\mathbb{Z}^{*n_s} = [0, 1, 2, \dots]^{n_s}$. The terms $c_{\mathbf{k}}(t)$ and $\phi_{\mathbf{k}}$ are the Fourier coefficients of $c(\mathbf{x}, t)$ and $\phi(\mathbf{x})$,

respectively. The term $\Lambda(\mathbf{k})$, which weights the contribution of small-scale and large-scale frequency modes, is given by

$$\Lambda(\mathbf{k}) = \frac{1}{(1 + \|\mathbf{k}\|^2)^s}.$$

In the original formulation of the SMC method [72], for Sobolev space norm of negative index, exponent s is defined as

$$s = -\frac{n_s + 1}{2},$$

which yields $s = -3/2$ for a two-dimensional domain.

To calculate the potential field u , Fourier basis functions are defined for a rectangular two-dimensional domain $[0, L_1] \times [0, L_2]$ as

$$f_{\mathbf{k}}(\mathbf{x}) = \frac{1}{h_{\mathbf{k}}} \cos\left(\frac{k_1 \pi x_1}{L_1}\right) \cos\left(\frac{k_2 \pi x_2}{L_2}\right), \quad (5.6)$$

where the wave-number vector is defined as $\mathbf{k} = [k_1, k_2]^T$, and

$$h_{\mathbf{k}} = \int_0^{L_1} \int_0^{L_2} \cos^2\left(\frac{k_1 \pi x_1}{L_1}\right) \cos^2\left(\frac{k_2 \pi x_2}{L_2}\right) dx_1 dx_2.$$

The cosine basis functions satisfy the condition

$$\frac{\partial f_{\mathbf{k}}}{\partial \mathbf{n}} = 0,$$

on the domain boundary $\partial\Omega_{2D}$. The potential field u is then calculated by

$$u_{SMC}(\mathbf{x}) = \sum_{\mathbf{k}} \Lambda(\mathbf{k}) (c_{\mathbf{k}}(t) - \phi_{\mathbf{k}}) f_{\mathbf{k}}(\mathbf{x}),$$

and the corresponding gradient of u is calculated by

$$\nabla u_{SMC}(\mathbf{x}) = \sum_{\mathbf{k}} \Lambda(\mathbf{k}) (c_{\mathbf{k}}(t) - \phi_{\mathbf{k}}) \nabla f_{\mathbf{k}}(\mathbf{x}).$$

The UAVs' ω^E values are then calculated analogously to the HEDAC method, using equation (5.3) to obtain the normalized gradient of the potential field and equation (5.4) to compute the yaw angular velocity.

A variation of the SMC method, called mSMC [74], generally demonstrates improved performance over the original and differs only in the choice of Sobolev index, adopting $s = -1/2$

for the two-dimensional case. However, in contrast to SMC, whenever mSMC is used in this thesis, it is applied in the context of solving the ergodic search task defined in (4.14) by minimizing

$$\|m(\mathbf{x}, t)\|_{H^s}^2 = \sum_{\mathbf{k} \in \mathbb{Z}^{n_s}} \Lambda(\mathbf{k}) |m_{\mathbf{k}}(t)|^2$$

where $m_{\mathbf{k}}(t)$ are the Fourier coefficients of $m(\mathbf{x}, t)$. Consequently, the gradient of the potential field is calculated by

$$\nabla u_{mSMC}(\mathbf{x}) = \sum_{\mathbf{k}} \Lambda(\mathbf{k}) (m_{\mathbf{k}}(t)) \nabla f_{\mathbf{k}}(\mathbf{x}).$$

Since the Fourier basis functions (5.6) are inherently defined on rectangular domains, a limitation of both the SMC and mSMC methods is that they can only be applied to rectangular domains with no internal obstacles.

6 YAW CONSTRAINTS AND COLLISION AVOIDANCE PROCEDURE

Before being applied to the UAVs, the yaw angular velocities ω^E , computed using the ergodic method via equation (5.4), are validated against UAV motion constraints and checked for potential collisions. Collision avoidance involves verifying and adjusting the ω^E values to guarantee a collision-free trajectory. The procedure builds upon the technique presented in [77], incorporating certain modifications. Despite the ability to adjust UAV altitude, collision avoidance between UAVs and with domain boundaries is strictly enforced within the two-dimensional search space, meaning flyovers are excluded from consideration and not allowed.

6.1 Yaw control constraints

A limit angular velocity ω_{lim} is defined for each UAV, representing its physical constraint. As a result, the condition $|\omega| \leq \omega_{lim}$ must be met. Furthermore, the value of ω is linked to horizontal velocity and path curvature constraints. The horizontal velocity is computed by evaluating multiple candidate velocities along a given path segment, as detailed in Chapter 7. To ensure the UAV can follow the computed path at any selected velocity, a constraint on the UAV's path curvature is imposed based on the minimum turning radius R_{min} . This curvature restriction is commonly referred to as the Dubins constraint. Consequently, the maximum ω is required when the UAV navigates a turn with radius R_{min} at its maximum horizontal velocity. To satisfy all the previously defined ω constraints, maximal yaw angular velocity ω_{max} is defined with

$$\omega_{max} = \min \left(\omega_{lim}, \frac{v_{s,max}}{R_{min}} \right),$$

and is enforced through the constraint

$$|\omega| \leq \omega_{max}. \quad (6.1)$$

6.2 Collision avoidance optimization problem formulation

The collision avoidance strategy is based on the principle that, at any given moment, a UAV must have at least one collision-free circular trajectory with radius R_{min} available. The adopted procedure from [77] evaluates the circular escape trajectory only at the final position after a fixed time interval Δt , since that position is known for a given ω and a constant moving velocity. The control approach in this thesis considers variable UAV velocity, and the UAV's future position cannot be determined at this stage, as the horizontal velocity is unknown and is calculated later in the control procedure. To address this, the collision avoidance strategy from [77] was adapted by defining an enlarged collision-free region that guarantees that all potential future positions – achieved using horizontal velocities within the operational range $[0, v_{s,max}]$ – have at least one circular escape trajectory free.

Based on the curvature constraint defined in (6.1), two extreme turning maneuvers can be identified: a maximum-rate right turn with $\omega = \omega_{max}$ and a maximum-rate left turn with $\omega = -\omega_{max}$. These maneuvers correspond to circular paths of radius R_{min} , centered at points \mathbf{f}^+ and \mathbf{f}^- , respectively (depicted as blue circles in Figure 6.1). The centers of the escape maneuver paths can be computed as

$$\mathbf{f}_i^m(t) = \begin{bmatrix} x_i(t) \\ y_i(t) \end{bmatrix} + R_{min,i} \begin{bmatrix} m \cdot \sin \theta_i(t) \\ -m \cdot \cos \theta_i(t) \end{bmatrix},$$

where $m = \pm 1$ indicates the left or the right escape route, with $m = 1$ corresponding to \mathbf{f}^+ and $m = -1$ to \mathbf{f}^- . These circular escape paths, or their arc segments, act as emergency turning options and form the foundation of the collision avoidance strategy in the two-dimensional domain Ω_{2D} .

A clearance distance δ_i is defined to ensure a safe gap between the i -th UAV search agent and any static obstacle or a moving UAV. Accordingly, clearance circles C^+ and C^- , each with radius $R_i + \delta_i$ are introduced. They are centered at \mathbf{f}^+ and \mathbf{f}^- , respectively. In Figure 6.1, the clearance circles corresponding to the current UAV positions, depicted as green arrows, are shown as yellow shaded circles, whereas those representing possible future positions at the next control time step – depending on the horizontal velocity values used – are depicted as gray circles. Since the horizontal velocity of a given UAV is not predetermined, clearance must be guaranteed for all supported velocities within the range $[0, v_{s,max}]$. To achieve this,

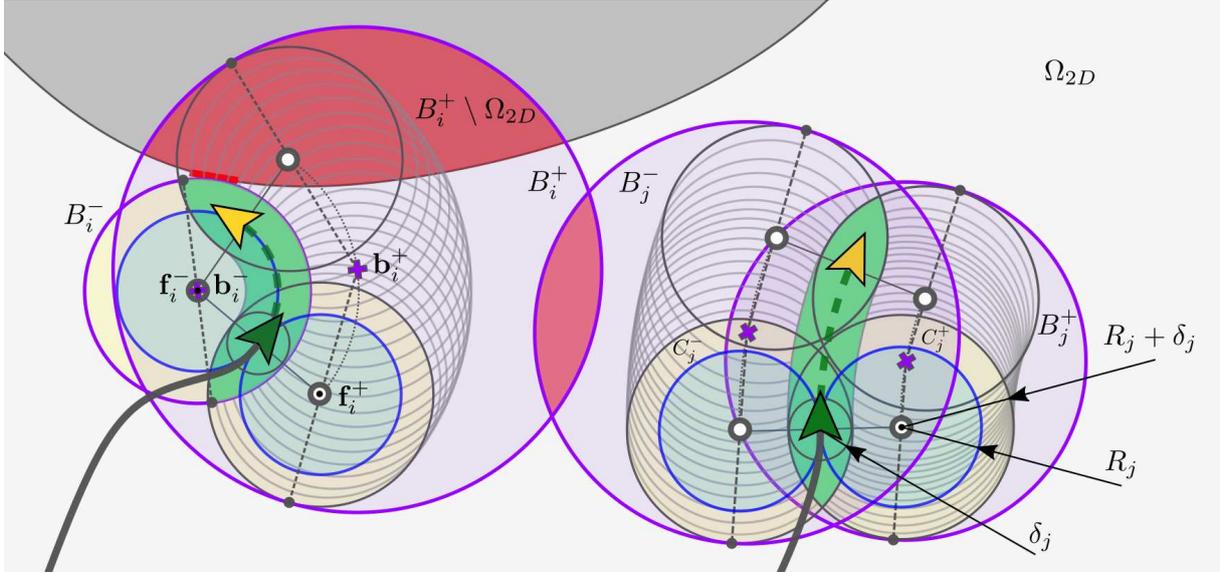


Figure 6.1: UAVs executing the collision avoidance maneuver.

bounding circles B^+ and B^- (depicted as purple circles in Figure 6.1), centered at \mathbf{b}^+ and \mathbf{b}^- , are constructed. These bounding regions encompass all possible clearance circles that may occur due to variations in speed, and thus ensure that an escape trajectory is always available regardless of the selected velocity. In the special case where $\omega = \omega_{max}$ or $\omega = -\omega_{max}$ is applied indefinitely, the UAV maintains a steady circular path around \mathbf{f}^+ and \mathbf{f}^- , respectively. Under such conditions, the corresponding clearance and bounding circles overlap, and their centers are identical (as can be seen from the bounding circle B_i^- in Figure 6.1). To ensure feasibility of the maneuver, at least one of the bounding circles, B^+ or B^- , must be free of collisions.

It is important to note that the collision avoidance method presented below makes use of the full range of the allowed values for the yaw angular velocity $\omega \in [-\omega_{max}, \omega_{max}]$. Within this range, there are typically multiple sub-intervals that provide collision-free movement during the control step Δt . However, the goal is not only to ensure feasibility over a single control step, but also to maintain continuous, safe motion that uses a minimal amount of space and preserves the ability to consider the full range of alternative directions in future steps.

To ensure collision-free motion it is necessary to verify that a collision-free escape circle will be available at the end of the current control interval Δt , regardless of the horizontal velocity used. Since the horizontal velocity is unknown at this stage, bounding circles representing sets of escape circles are considered for both ω_{max} and $-\omega_{max}$, covering all UAV positions reachable at the end of the current control interval using ω_i and velocities within the range $[0, v_{s,max,i}]$. Therefore, it is necessary to verify whether the yaw angular velocities ω_i result in at least one

collision-free bounding circle for each agent. This is completed by evaluating the collision area generated when applying yaw angular velocities ω_i from the UAVs' current positions and orientations over the control interval Δt . The vector containing a set of yaw angular velocities for all UAVs is denoted by $\vec{\omega} = (\omega_1, \dots, \omega_n)$.

To ensure collision-free motion for all UAVs, it is necessary to verify whether the set $\vec{\omega}$ contains ω values that result in at least one collision-free bounding circle for each UAV. The bounding circle centers, if $\omega_i \neq 0$, are computed with

$$\mathbf{b}_i^m(\omega_i, t) = \mathbf{f}^m(t) + \left(\left| \frac{v_{s,\max,i}}{\omega_i} \right| - m \cdot R_{\min,i} \right) \cdot \begin{bmatrix} \sin\left(\frac{\omega_i \cdot \Delta t}{2}\right) \cos \theta_i(t) - \left(1 - \cos\left(\frac{\omega_i \cdot \Delta t}{2}\right)\right) \sin \theta_i(t) \cdot \text{sign}(\omega_i) \\ \sin\left(\frac{\omega_i \cdot \Delta t}{2}\right) \sin \theta_i(t) + \left(1 - \cos\left(\frac{\omega_i \cdot \Delta t}{2}\right)\right) \cos \theta_i(t) \cdot \text{sign}(\omega_i) \end{bmatrix}$$

where $m = \pm 1$ indicates the left or the right bounding circle, with $m = 1$ corresponding to \mathbf{b}^+ and \mathbf{f}^+ , and $m = -1$ to \mathbf{b}^- and \mathbf{f}^- . The term $\text{sign}(\omega_i)$ indicates the turning direction associated with ω_i , taking the value $+1$ for counter-clockwise (left) turns and -1 for clockwise (right) turns. If $\omega_i = 0$, the bounding circle centers are computed with

$$\mathbf{b}_i^m(t) = \mathbf{f}^m(t) + \frac{v_{s,\max,i} \cdot \Delta t}{2} \begin{bmatrix} \cos \theta_i(t) \\ \sin \theta_i(t) \end{bmatrix}.$$

The radius of the bounding circle is equal to

$$R_{\min,i} + \delta_i + \|\mathbf{b}_i^m(t) - \mathbf{f}^m(t)\|.$$

For each agent, there are two possible escape route sets (to the left or the right, corresponding to B^- or B^+ bounding circles, respectively) leading to 2^n possible combinations of maneuver sets. All possible combinations of escape route sets can be systematically represented using

$$\mathbb{B}_{j,i}(\vec{\omega}) = \begin{cases} B_i^-, & \text{if } \mathbf{q}_i(j) = 0 \\ B_i^+, & \text{if } \mathbf{q}_i(j) = 1, \end{cases}$$

where $j \in 0, \dots, 2^n - 1$, and the binary vector $\mathbf{q}(j)$ of length n represents a unique combination of escape route sets for all agents.

Any combination is potentially collision-free and $\mathbb{B}_{j,i}$ indicates which of the two bounding

circles, B^+ or B^- , associated with the escape route set, is selected for each agent i . Each escape route configuration can be checked for safety by computing its associated collision area as

$$A_j(\vec{\omega}) = \sum_{i=1}^{n-1} \sum_{k=i+1}^n \|\mathbb{B}_{j,i}(\vec{\omega}) \cap \mathbb{B}_{j,k}(\vec{\omega})\| + \sum_{i=1}^n \|\mathbb{B}_{j,i}(\vec{\omega}) \setminus \Omega_{2D}\| \quad (6.2)$$

where the collision area between UAVs is computed by summing intersections of their bounding circles in the first term, and the domain collision area is computed by summing intersections between bounding circles and the domain boundary in the second term. To determine whether at least one combination is feasible, a minimum intersection area is defined as

$$A_{min}(\vec{\omega}) = \min(A_0(\vec{\omega}), A_1(\vec{\omega}), \dots, A_{2^n-1}(\vec{\omega})).$$

An optimization problem can now be formulated as follows:

$$\begin{aligned} \underset{\vec{\omega}}{\text{minimize}} \quad & J(\vec{\omega}) = \sum_i (\omega_i^E - \omega_i)^2 \\ \text{subject to} \quad & A_{min}(\vec{\omega}) = 0, \\ & |\omega_i| \leq \omega_i^{max}. \end{aligned} \quad (6.3)$$

The ω values computed using the ergodic method, via equation (5.4), guide effective inspection of the area corresponding to m . Accordingly, for each UAV, a feasible yaw angular velocity is selected to remain as close as possible to this value, denoted by ω_i^E . This is accomplished by minimizing the sum of squared differences between the trial angular velocities and their ergodic counterparts, which defines the objective function J . Although the solution obtained using the ergodic method is optimal in the absence of constraints, enforcing collision avoidance by setting the total collision area A_{min} to zero imposes further requirements. This may necessitate adjustments to the computed ω_i^E values to prevent collisions with obstacles and other UAVs. At least one feasible solution to (6.3) is always guaranteed because a set containing limit turning velocities $\pm\omega_{max}$, which generate bounding circles free of collisions ($A_{min} = 0$), was confirmed in the previous control step. As a result, the solution to (6.3) ensures collision-free UAV motion in the Ω_{2D} domain.

6.3 Decomposing and solving the optimization problem

With each additional agent, the number of possible combinations in (6.2) increases exponentially. This presents a significant drawback, as the equation is repeatedly evaluated as a constraint within the optimization process. This is handled by decomposing the optimization problem (6.3) into separate sub-problems based on potential collisions and UAV interactions, which can be anticipated from the distances between the UAVs. Therefore, individual UAVs or groups that do not interact with others are optimized independently, resulting in significant computational efficiency gains.

The optimization problem (6.3) is decomposed by forming groups within the UAV swarm. Interaction groups are formed by evaluating all pairs of UAVs, where each UAV in a pair is indexed by i_1 and i_2 , and checking whether they satisfy a distance-based condition

$$\|(x_{i_1}, y_{i_1}) - (x_{i_2}, y_{i_2})\| \leq 2R_{min,i_1} + \delta_{i_1} + v_{s,max,i_1} \cdot \Delta t + 2R_{min,i_2} + \delta_{i_2} + v_{s,max,i_2} \cdot \Delta t. \quad (6.4)$$

When the condition in (6.4) is met, agents i_1 and i_2 are placed in the same group, isolated from other groups. This allows the optimization problem (6.3) to be solved independently for each group, significantly reducing the number of optimization variables and leading to a considerable speedup of the collision-avoidance procedure.

Each group is optimized independently and in sequence using a modified Cyclic Coordinate Search (CCS) algorithm, as used in [77]. The modifications include direct integration of constraints within the optimization procedure and the replacement of the traditional Line Search in CCS with the Golden Section Search for one-dimensional optimization. The Golden Section Search was selected because its bracketing strategy guarantees that feasible solutions, particularly those at the boundaries of the variable $\vec{\omega}_i$, are preserved during optimization.

Golden Section Search traditionally addresses bounded, unconstrained one-dimensional problems by comparing objective values at different trial points. When applied to the one-dimensional coordinate search in CCS for solving the proposed problem (6.3), the collision area constraint must be incorporated. As a result, the operator used to compare trial solutions

has been modified to consider both objective metrics and the constraint, as expressed by

$$f(\vec{\omega}_{k_1}) < f(\vec{\omega}_{k_2}) = \begin{cases} \text{if } A_{total}(\vec{\omega}_{k_1}) = A_{total}(\vec{\omega}_{k_2}) : \\ \quad J(\vec{\omega}_{k_1}) < J(\vec{\omega}_{k_2}) \\ \text{otherwise:} \\ \quad A_{total}(\vec{\omega}_{k_1}) < A_{total}(\vec{\omega}_{k_2}) \end{cases} ,$$

where the expression $f(\vec{\omega}_{k_1}) < f(\vec{\omega}_{k_2})$ is used to determine if the trial solution $\vec{\omega}_{k_1}$ outperforms $\vec{\omega}_{k_2}$. It is performed for a minimum of 20 iterations or until a feasible solution satisfying $A_{total} = 0$ is found.

7 UNEVEN TERRAIN EXPLORATION

When conducting searches over uneven terrain, the UAV's altitude must be controlled to maintain a safe distance from the ground while staying low enough to ensure effective target detection. This chapter presents how these objectives are achieved. First, it outlines the problem, highlighting the key considerations and the proposed approach. Next, it describes how the terrain is perceived. Finally, it presents the formulation of the MPC, which enables both altitude and velocity control.

7.1 Problem formulation – UAV control over uneven terrain

The three-dimensional trajectory of the UAV is determined by the combined influence of its horizontal and vertical motion. The yaw angular velocity $\omega(t)$, presented in Chapter 5, combined with a forward velocity defines a two-dimensional trajectory suitable for exploring the search domain Ω_{2D} . While this is sufficient for flight over flat terrain at a constant altitude, exploring uneven terrain requires dynamic altitude adjustment. To ensure a smooth and efficient trajectory over obstacles and terrain alterations/variations, both the horizontal and vertical velocities must be varied. Altitude control allows safe horizontal movement by keeping the UAVs above the specified minimum height h_{min} above the terrain, while also aiming to maintain the goal height h_{goal} that ensures good area coverage and detection performance. Velocity control ensures the UAVs can follow the planned three-dimensional path by taking into account both horizontal and vertical movement, as well as the UAVs' motion constraints. This is achieved by the MPC procedure described in this chapter, which is responsible for computing the regimes of $\rho(t)$ and $\varphi(t)$, and for performing altitude and velocity control necessary for exploring uneven terrain.

The distinction between UAV height and altitude must be clarified to ensure precise understanding of flight parameters. Throughout this thesis, the UAV's height refers to its vertical distance above the terrain directly beneath it, while altitude denotes the vertical distance above mean sea level. Additionally, terrain elevation is defined as the height of the ground surface relative to mean sea level.

the TanDEM-X mission [104] funded through a Public Private Partnership involving Airbus Defence and Space and the German Aerospace Centre (DLR).

7.3 Trial trajectories and control functions

The horizontal control strategy from Chapter 5 is used to generate a two-dimensional path projected to the horizontal plane, as illustrated in Figure 7.2, or later in Figure 7.3, panel A). The corresponding terrain elevation profile is then extracted for this path using z_T . To meet mission objectives while satisfying dynamic and environmental constraints, the two-dimensional path is supplemented with ρ and φ parameters, computed via MPC over the prediction window $[t, t + \tau_{max}]$, to generate an optimal three-dimensional flight path. Here, τ represents relative time from the current time t , and τ_{max} is the duration of the MPC prediction time horizon.

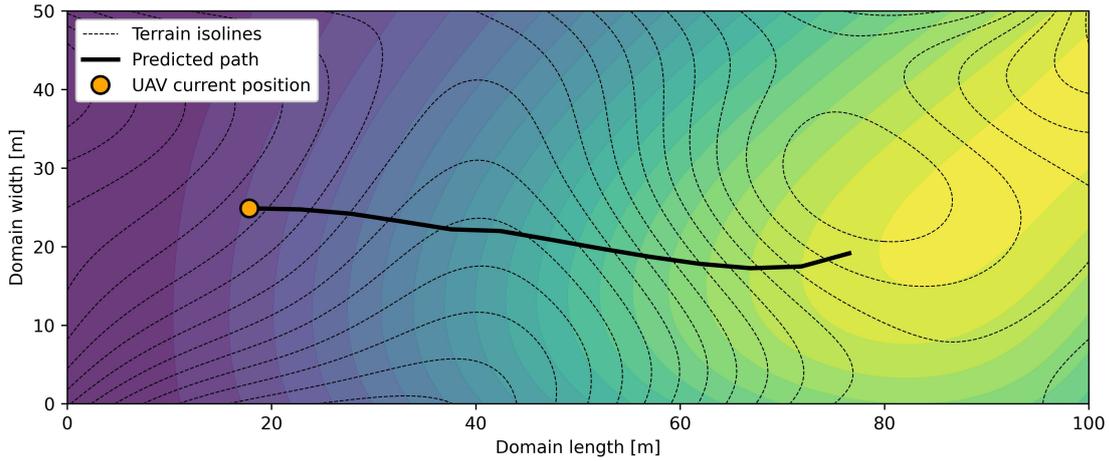


Figure 7.2: Visual representation of the predicted two-dimensional path generated along the potential field gradient.

Initially, a predicted horizontal path is introduced under the assumption of maximum velocity ($\rho = 1$) and no vertical movement ($\varphi = 0$). Its coordinates and derived variables are denoted with $\bar{\tau}$. The following equations define this predicted path:

$$\frac{d\bar{x}_i}{d\tau} = v_{s,max,i} \cdot \cos \bar{\theta}_i,$$

$$\frac{d\bar{y}_i}{d\tau} = v_{s,max,i} \cdot \sin \bar{\theta}_i,$$

$$\frac{d\bar{\theta}_i}{d\tau} = \bar{\omega}_i.$$

Assuming a constant potential field u over the time interval $[0, \tau_{max}]$, obtained at time t , the UAVs' trial yaw angular velocity is calculated using

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

where $*$ denotes the dot product, and $\bar{\mathbf{v}}_i$ represents the heading direction vector. The computed $\bar{\omega}_i(\tau)$ is validated to satisfy constraint (6.1) and ensure collision avoidance over the time interval $[0, \Delta t]$ with procedure described in Chapter 6.

Based on the predicted horizontal path, the trial trajectory is introduced to describe the UAV's arbitrary three-dimensional flight regime within the predicted time window. The predicted path length function $\bar{s}(\tau)$ is defined based on

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

resulting in $\bar{s}_i = v_{s,max,i} \cdot \tau$, which allows the trial trajectory to be parametrized as $\bar{x}_i(s_i)$ and $\bar{y}_i(s_i)$. Based on $\tilde{\rho}$ and $\tilde{\varphi}$, the trial trajectory length function \tilde{s} is defined by

$$\frac{d\tilde{s}_i}{d\tau} = v_{s,i}(\tilde{\rho}_i(\tau), \tilde{\varphi}_i(\tau)), \quad (7.1)$$

By combining the length-based parametrization of \bar{x} and \bar{y} with equation (7.1), the horizontal components of the trial trajectory over the interval $[0, \tau_{max}]$ are expressed as

$$\tilde{x}_i(\tau) = \bar{x}_i(\tilde{s}_i(\tau)),$$

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

Typically, the UAV trial trajectory does not cover the entire predicted path length over $[0, \tau_{max}]$ due to potential reductions in velocity intensity ($\rho(\tau) < 1$) or flight in climbing or descending modes ($\varphi(\tau) \neq 0$). Using the parametrization $\bar{\omega}_i(s)$, the trial yaw angular velocity is defined as

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

which guarantees that the trial trajectory follows the predicted path and maintains its curvature, while allowing variations in trajectory length due to the influence of $\tilde{\rho}$ and $\tilde{\varphi}$. Employing the

trial regimes $\tilde{\rho}$ and $\tilde{\varphi}$, the vertical component of the trial trajectory is expressed by

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

The trail velocity intensity $\tilde{\rho}(\tau)$ and trail incline angle $\tilde{\varphi}(\tau)$ are optimized within the MPC framework, while the trial yaw angular velocity $\tilde{\omega}(\tau)$ is adjusted with trial control functions $\tilde{\rho}(\tau)$ and $\tilde{\varphi}(\tau)$ to ensure that the resulting trajectory preserves the original curvature.

7.4 Optimization problem formulation

An optimal trial flight regime for the time interval $[t, t + \tau_{max}]$ is required to compute the necessary horizontal and vertical velocity controls, managed by ρ and φ . To set up the optimization problem, the trial control functions $\tilde{\rho}$ and $\tilde{\varphi}$ are parameterized using quadratic polynomial interpolation. The interpolation is based on three time points: $\tau_0 = 0$, $\tau_1 = \tau_{max}/2$, and $\tau_2 = \tau_{max}$. Since τ_0 corresponds to time t , the values of the trial control functions at τ_0 are determined by the current UAV state

$$\begin{aligned}\tilde{\rho}(\tau_0) &= \rho(t), \\ \tilde{\varphi}(\tau_0) &= \varphi(t),\end{aligned}$$

which is known. The optimization vector $\mathbf{W} \in \mathbb{R}^4$ is then defined using values of $\tilde{\rho}$ and $\tilde{\varphi}$ at τ_1 and τ_2 as

$$\mathbf{W}_i \equiv [\tilde{\rho}_i(\tau_1), \tilde{\rho}_i(\tau_2), \tilde{\varphi}_i(\tau_1), \tilde{\varphi}_i(\tau_2)]^T. \quad (7.2)$$

The trial control functions, although defined over the relative time variable τ , are also expressed as functions of the optimization vector \mathbf{W}_i . For clarity and simplicity, a notation

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

is used, meaning that any trial function (\cdot) is simultaneously a function of the optimization vector \mathbf{W}_i and relative time τ .

Two primary objectives are established for the flight regime: to maximize the UAV's velocity and to maintain its height as close as possible to the goal search height h_{goal} . Increasing the UAV's velocity improves area coverage efficiency and consequently reduces the total inspection

time. This is achieved by incorporating the first minimization objective

$$o_{v,i}(\mathbf{W}_i) = 1 - \frac{1}{\tau_{max}} \int_0^{\tau_{max}} \tilde{\rho}_i|_{\mathbf{W}_i}(\tau) d\tau.$$

The second objective focuses on maintaining the desired goal search height, which helps achieve an effective balance between area coverage and sensing performance. It is formulated by introducing the terrain elevation function

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

The second objective is now defined as

$$o_{h,i}(\mathbf{W}_i) = \frac{1}{h_{goal,i} \cdot \tau_{max}} \int_0^{\tau_{max}} \left| \tilde{z}_i|_{\mathbf{W}_i}(\tau) - \tilde{z}_{T,i}|_{\mathbf{W}_i}(\tau) - h_{goal,i} \right| d\tau,$$

and quantifies how closely the trial height function $\tilde{z} - \tilde{z}_T$ aligns with the target height h_{goal} . Since both objectives are normalized they can be summed directly without the need for weight factors, as they are considered equally important. The optimization goal is now formulated as the minimization of the objective function

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

Constraints are imposed to ensure adherence to physical and operational limits during flight. These trial trajectory constraints are formulated as inequality conditions $c(\mathbf{W}) \leq 0$. To ensure that the UAV maintains a safe flight height, a minimum height constraint

$$\tilde{z}_i|_{\mathbf{W}_i}(\tau) - \tilde{z}_{T,i}|_{\mathbf{W}_i}(\tau) \geq h_{min,i}. \quad (7.3)$$

is imposed, preventing the height from falling below the specified limit h_{min} . This condition must hold throughout the full τ interval, which is verified by

$$c_{h,i}(\mathbf{W}_i) = \frac{1}{h_{min,i} \cdot \tau_{max}} \int_0^{\tau_{max}} \max \left\{ h_{min,i} - \tilde{z}_i|_{\mathbf{W}_i}(\tau) + \tilde{z}_{T,i}|_{\mathbf{W}_i}(\tau), 0 \right\} d\tau.$$

To comply with the UAVs' velocity specifications, the following constraints must be met, as given by

$$v_{s,min,i} \leq \tilde{v}_{s,i} \lfloor_{\mathbf{W}_i}(\tau) \leq v_{s,max,i}, \quad (7.4)$$

$$v_{z,min,i} \leq \tilde{v}_{z,i} \lfloor_{\mathbf{W}_i}(\tau) \leq v_{z,max,i}, \quad (7.5)$$

where $\tilde{v}_{s,i} \lfloor_{\mathbf{W}_i}(\tau)$ is the horizontal trial velocity function and $\tilde{v}_{z,i} \lfloor_{\mathbf{W}_i}(\tau)$ is the vertical trial velocity function, derived from (4.1) and (4.2). Horizontal velocity limits are checked using

$$c_{vs,min,i}(\mathbf{W}_i) = \frac{1}{v_{s,min,i} \cdot \tau_{max}} \int_0^{\tau_{max}} \max \left\{ v_{s,min,i} - \tilde{v}_{s,i} \lfloor_{\mathbf{W}_i}(\tau), 0 \right\} d\tau,$$

$$c_{vs,max,i}(\mathbf{W}_i) = \frac{1}{v_{s,max,i} \cdot \tau_{max}} \int_0^{\tau_{max}} \max \left\{ \tilde{v}_{s,i} \lfloor_{\mathbf{W}_i}(\tau) - v_{s,max,i}, 0 \right\} d\tau,$$

and similarly, vertical velocity limits are evaluated using

$$c_{vz,min,i}(\mathbf{W}_i) = \frac{1}{v_{z,min,i} \cdot \tau_{max}} \int_0^{\tau_{max}} \max \left\{ v_{z,min,i} - \tilde{v}_{z,i} \lfloor_{\mathbf{W}_i}(\tau), 0 \right\} d\tau,$$

$$c_{vz,max,i}(\mathbf{W}_i) = \frac{1}{v_{z,max,i} \cdot \tau_{max}} \int_0^{\tau_{max}} \max \left\{ \tilde{v}_{z,i} \lfloor_{\mathbf{W}_i}(\tau) - v_{z,max,i}, 0 \right\} d\tau.$$

Considering the UAVs' dynamic limitations, acceleration within trial trajectories is also constrained to account for thrust capacity, pitch control, and inertia. Horizontal and vertical accelerations are restricted by basic limits

$$a_{s,min,i} \leq \tilde{a}_{s,i} \lfloor_{\mathbf{W}_i}(\tau) \leq a_{s,max,i}, \quad (7.6)$$

$$a_{z,min,i} \leq \tilde{a}_{z,i} \lfloor_{\mathbf{W}_i}(\tau) \leq a_{z,max,i}, \quad (7.7)$$

where the trial acceleration functions, denoted by $\tilde{a}_{s,i} \lfloor_{\mathbf{W}_i}(\tau)$ and $\tilde{a}_{z,i} \lfloor_{\mathbf{W}_i}(\tau)$, are derived through numerical differentiation of the respective trial velocity functions $\tilde{v}_{s,i} \lfloor_{\mathbf{W}_i}(\tau)$ and $\tilde{v}_{z,i} \lfloor_{\mathbf{W}_i}(\tau)$. The constraints on horizontal acceleration and deceleration are enforced through

$$c_{as,min,i}(\mathbf{W}_i) = \frac{1}{a_{s,min,i} \cdot \tau_{max}} \int_0^{\tau_{max}} \max \left\{ a_{s,min,i} - \tilde{a}_{s,i} \lfloor_{\mathbf{W}_i}(\tau), 0 \right\} d\tau,$$

$$c_{as,max,i}(\mathbf{W}_i) = \frac{1}{a_{s,max,i} \cdot \tau_{max}} \int_0^{\tau_{max}} \max \left\{ \tilde{a}_{s,i} \lfloor_{\mathbf{W}_i}(\tau) - a_{s,max,i}, 0 \right\} d\tau,$$

and in a similar fashion, the vertical acceleration constraints are enforced with

$$c_{az,min,i}(\mathbf{W}_i) = \frac{1}{a_{z,min,i} \cdot \tau_{max}} \int_0^{\tau_{max}} \max \left\{ a_{z,min,i} - \tilde{a}_{z,i}[\mathbf{W}_i](\tau), 0 \right\} d\tau,$$

$$c_{az,max,i}(\mathbf{W}_i) = \frac{1}{a_{z,max,i} \cdot \tau_{max}} \int_0^{\tau_{max}} \max \left\{ \tilde{a}_{z,i}[\mathbf{W}_i](\tau) - a_{z,max,i}, 0 \right\} d\tau.$$

For simplicity, the vector notation is introduced to combine all define constraints:

$$\mathbf{c}_i(\mathbf{W}_i) = [c_{h,i}(\mathbf{W}_i), \quad c_{vs,min,i}(\mathbf{W}_i), c_{vs,max,i}(\mathbf{W}_i), \\ c_{vz,min,i}(\mathbf{W}_i), \quad c_{vz,max,i}(\mathbf{W}_i), c_{as,min,i}(\mathbf{W}_i), \\ c_{as,max,i}(\mathbf{W}_i), \quad c_{az,min,i}(\mathbf{W}_i), c_{az,max,i}(\mathbf{W}_i)]^T.$$

Finally, the MPC optimization problem for each agent i can then be formulated as follows:

$$\begin{aligned} & \underset{\mathbf{W}_i}{\text{minimize}} \quad o_i(\mathbf{W}_i) = o_{v,i}(\mathbf{W}_i) + o_{h,i}(\mathbf{W}_i) \\ & \text{subject to} \quad \mathbf{c}_i(\mathbf{W}_i) \leq \mathbf{0}, \end{aligned}$$

where \leq is applied element-wise to all constraints in the vector \mathbf{c} , while $\mathbf{0}$ is the null vector.

7.5 MPC optimization procedure

The optimization routine begins with an initial value, \mathbf{W}_0 , of the optimization vector \mathbf{W} , which is defined by equation (7.2). The initial value is chosen from several possible initialization options:

$$\begin{aligned} \mathbf{W}_{0a} &= [1, 1, 0, 0]^T, \\ \mathbf{W}_{0b} &= [1, 1, 1, 1]^T, \\ \mathbf{W}_{0c} &= [0.5, 0.5, 1, 1]^T. \end{aligned}$$

Candidates from a to c are assessed in order, and the first one yielding a feasible outcome is designated as \mathbf{W}_0 . The vector \mathbf{W}_{0a} is designed to promote maximum forward velocity along the horizontal search direction with no change in altitude, while \mathbf{W}_{0b} and \mathbf{W}_{0c} are configured to increase altitude with 2 distinct velocity intensities. If no feasible solution is found within the initialization options, the optimization step is bypassed, and the UAV proceeds with the collision avoidance procedure, described in Section 7.6.

Figure 7.3 illustrates the optimization procedure along with a set of candidate initial vectors. Starting from the UAV's current position, a predicted path is generated by following the gradient of the potential field u and simulating the maximum attainable horizontal motion over n_{pts} prediction time steps, as illustrated in panel A) of Figure 7.3. Along the predicted path, the terrain elevation z_T is sampled and used to construct two reference curves: a minimum altitude ($z_T + h_{min}$) and a goal altitude ($z_T + h_{goal}$) curve. The optimization stage, visualized in panel B) of Figure 7.3, begins with the first viable candidate among the initial vectors \mathbf{W}_{0a} , \mathbf{W}_{0b} , \mathbf{W}_{0c} . The objective is to determine an optimal control regime over n_{pts} steps that balances maximizing velocity with maintaining the goal height, while adhering to velocity, acceleration, and minimum height constraints. Once the optimal regime is identified, the UAV executes it for a duration of Δt , resulting in an updated position from which the entire process is repeated.

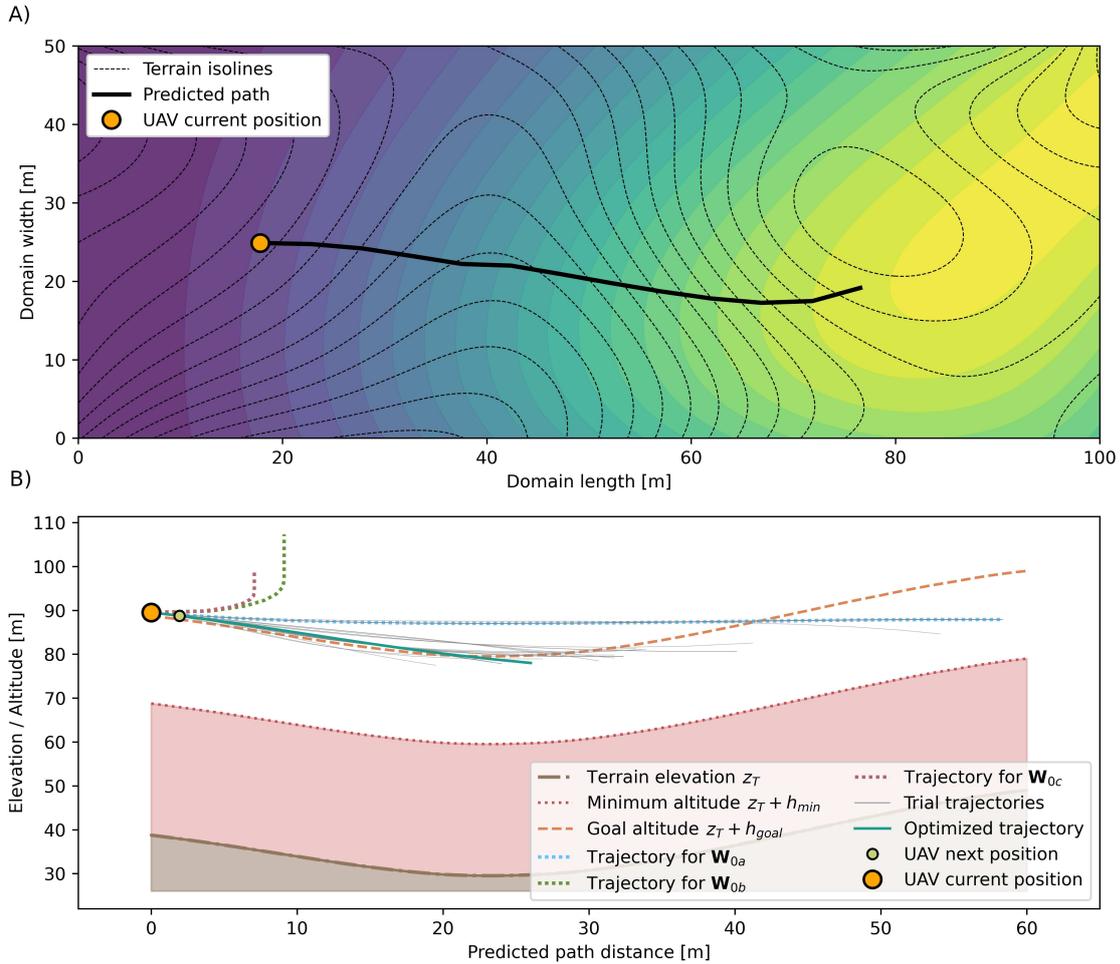


Figure 7.3: Graphical representation of the predicted path (A) alongside the corresponding MPC optimization process (B).

Optimization is carried out using a modified version of the GPS-MADS algorithm [105],

referred to as Multi-Scale Grid Search (MSGs), implemented in the *Indago* Python module [106]. The optimization process is configured to run for up to 30 iterations, with stopping criteria defined by a maximum of 10 stalled iterations and a target fitness threshold of 10^{-3} . The resulting optimal vector, \mathbf{W}_{opt} , provides the candidate control parameters governing velocity intensity, incline, and yaw angular velocity denoted as

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

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

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

respectively.

7.6 Terrain collision avoidance

The flight control parameters $\rho_{opt,i}$, $\varphi_{opt,i}$, and $\omega_{opt,i}$ computed through MPC optimization do not inherently guarantee a circular, collision-free trajectory that can be immediately executed as a collision avoidance escape maneuver. While collisions with other UAVs and the domain boundary are addressed within the two-dimensional trajectory in Ω_{2D} , as described in Chapter 5, terrain collisions must also be accounted for in this context. A terrain collision is defined as a violation of the minimum height constraint h_{min} , which must be satisfied throughout the entire collision avoidance escape maneuver. Given the high computational cost, embedding this constraint in the MPC optimization makes the algorithm unsuitable for real-time UAV control. Therefore, the flight control parameters from MPC are reviewed and corrected as needed before execution.

The escape maneuver is a controlled UAV response designed to fulfill the following flight criteria:

- The escape maneuver may be initiated at any time from arbitrary UAV positions, states, and control parameters. In this context, the UAV is considered either at time $t + \Delta t$ with optimized control inputs or, if necessary, at the present time t with the control parameters currently in use.

- The horizontal projection of the escape path is a circular arc (or a complete circle) with a radius $\pm R_{min,i}$, achieved by adjusting $\omega_{esc,i}$ to match one of the escape circles defined in Chapter 6.
- The UAV engages in the most aggressive deceleration it can perform i.e. $\rho_{em,i} \rightarrow \rho_{min,i}$.
- The UAV climbs at the steepest rate permitted by its dynamic limits, i.e. $\varphi_{em,i} \rightarrow \varphi_{max,i}$.
- The feasibility of the escape route is evaluated based on previously defined constraints: minimum height (7.3), horizontal velocity (7.4), vertical velocity (7.5), horizontal acceleration (7.6), and vertical acceleration (7.7).
- The escape maneuver is evaluated until either zero horizontal velocity is reached or the entire escape circle is completed.

Provided that a feasible escape maneuver can be performed using the optimal control parameters at time $t + \Delta t$, the optimal control parameters

$$\rho_i(t + \Delta t) = \rho_{opt,i},$$

$$\varphi_i(t + \Delta t) = \varphi_{opt,i},$$

$$\omega_i(t + \Delta t) = \omega_{opt,i},$$

are utilized by the UAV. When no valid escape maneuver exists at time $t + \Delta t$, implying that the current control parameters are invalid, the UAV executes the escape maneuver with alternate control inputs

$$\rho_i(t + \Delta t) = \rho_{em,i}(t + \Delta t),$$

$$\varphi_i(t + \Delta t) = \varphi_{em,i}(t + \Delta t),$$

$$\omega_i(t + \Delta t) = \omega_{em,i}(t + \Delta t),$$

which are determined for $t + \Delta t$ based on the escape maneuver verified in the previous computation step, simulating the regime starting from time t .

8 TERRAIN SEARCH SIMULATIONS

This section begins with an overview of three simulated search scenarios used for validating the proposed control methodology, followed by the introduction of the CV detection model and its integration into the sensing framework. Each scenario is then described in detail, including visualizations of the terrain, the UAV trajectories achieved, an evaluation of search performance, and comparisons with other methods. Finally, a robustness analysis is conducted by introducing uncertainty into the simulations to approximate realistic real-world operating conditions.

8.1 Simulated search scenarios overview

Each simulated search mission is performed with a specific UAV configuration, composed of units selected from three available types: UAV A, UAV B, and UAV C. The characteristics of each UAV type are presented in Table 8.1.

Table 8.1: Motion, vision/sensing and control UAV parameters used in simulations [107].

UAV parameters	UAV A	UAV B	UAV C	Units
Type	Multi-rotor	Multi-rotor	Fixed-Wing	–
Min turning radius R_{min}	25	25	100	m
Min clearance distance δ	7	7	60	m
Min search height h_{min}	30	30	100	m
Goal search height h_{goal}	50	100	300	m
Max horizontal velocity $v_{s,max}$	10	10	15	m/s
Min horizontal velocity $v_{s,min}$	0	0	5	m/s
Max ascending velocity $v_{z,max}$	5	5	1.2	m/s
Max descending velocity $v_{z,min}$	-3	-3	-1.2	m/s
Max horizontal acceleration $a_{s,max}$	2	2	2	m/s ²
Min horizontal acceleration $a_{s,min}$	-3.6	-3.6	-2	m/s ²
Max vertical acceleration $a_{z,max}$	2.8	2.8	1	m/s ²
Min vertical acceleration $a_{z,min}$	-2	-2	-1	m/s ²
Min incline ϕ_{min}	-90	-90	13.5	°
Max incline ϕ_{max}	90	90	13.5	°
Camera FOV γ_1	62.8	33.94	23	°
Camera FOV γ_2	37.9	19.48	13.06	°
Zoom factor Z	1×	2×	3×	–
Sensing function Γ	Γ_A	Γ_B	Γ_C	–
Prediction time steps n_{pts}	25	25	30	time steps

UAVs A and B are modeled after the DJI Matrice 210 v2 multi-rotor aircraft. Their velocity characteristics are derived from technical specifications, with the horizontal velocity set close to the optimal battery endurance speed for a large UAV, according to [66]. The acceleration values were obtained through real-world manual flight tests. The minimum search height was set to 30 m to ensure a safe clearance from potential natural or man-made objects at the terrain. Although the two UAVs are essentially identical, they operate at different goal search heights (h_{goal}), which results in different sensing functions (defined in the following section) and FOV. Both utilize the same camera sensor, with UAV A operating without zoom ($1\times$) and UAV B employing a $2\times$ zoom. They are configured to cover the same ground area from their respective altitudes.

UAV C represents a fixed-wing aircraft and is therefore more constrained. Due to its lower maneuverability, it performs the search at a goal height of 300 m, with a minimum search height set to 100 m. Its specifications are approximated based on typical operational characteristics and constraints. Its horizontal velocity is higher, while its other velocities and accelerations are lower than those of the multi-rotor aircraft. Its sensor covers an area equivalent to that of the multi-rotor aircraft and is designed to maintain comparable sensing performance by adjusting the sensing function for the higher search height.

To ensure feasible flight over complex terrain, the MPC horizon for each UAV is tailored to its specifications, flight parameters, and the terrain complexity. The duration of the prediction horizon is defined by the number of predicted time steps n_{pts} , resulting in a total prediction time window of $\tau_{\text{max}} = n_{\text{pts}} \cdot \Delta t$.

Three test cases were designed with varying area sizes and terrain complexity. The UAV fleet configurations and associated parameters for each case are summarized in Table 8.2.

Table 8.2: Overview of simulated test cases and their parameters [107]

Test case parameters	Plastic world	Mt. Vesuvius	Star dunes	Units
Domain size	0.72	7.44	7.5	km ²
Number of mesh nodes	8380	21825	21946	-
Number of mesh elements	16300	43098	43340	-
Elevation difference	421	608	221.4	m
Alpha α	1000	2300	2900	-
Beta β	0.1	1	1	-
Time step Δt	1	2	2	s
Search duration	30	60	60	min
Number of UAVs A	3	3	0	-
Number of UAVs B	0	2	0	-
Number of UAVs C	0	0	2	-

Multiple methods were employed to compute trajectories for each case, and their performance was evaluated using the survey accomplishment metric η . The proposed HEDAC + MPC method was compared against: HEDAC without the MPC framework (with UAVs flying at fixed altitude), the lawnmower method with MPC, SMC with MPC, and mSMC with MPC. All methods shared the same settings and initial UAV configurations at the start, except for the lawnmower method, where UAVs were approximately positioned to ensure near-optimal search performance. Additionally, for the lawnmower method, the MPC acceleration constraints were relaxed to enable feasible trajectories, as it does not support the collision avoidance method presented in the earlier chapter. The mSMC and SMC methods were not applied in certain cases because they do not support inter-domain obstacles or no-fly zones. Finally, the search altitude for the HEDAC method without MPC was set as the maximum terrain elevation plus the goal search height.

The simulations were computed on a system equipped with a CPU having a base clock speed of 3.7 GHz and a maximum boost frequency of 5.4 GHz. All computations were performed using a single thread. Visualization of flight data and analytical plots were created utilizing *Matplotlib* [108] and *PyVista* [109].

8.2 Sensing characteristics

Parameters such as the zoom factor Z and the respective FOV for each zoom level were defined for every camera sensor combined with a UAV, as shown in Table 8.1. Corresponding detection rate functions were constructed for all utilized camera configurations based on detection metrics provided by the YOLOv4 model in [61]. Although developed for animal detection, the model reports performance metrics across various altitudes, making it possible to link detection accuracy to image capture height.

Recall values from [61] were used and reduced by approximately 30% to account for differences in the apparent size of objects, specifically between humans and large mammals in images taken at the same height, as well as other operational factors that may reduce detection performance, such as image degradation caused by motion blur, focus errors, or haze [55].

The acquired recall–height data was extended by introducing an additional height point which assumed zero recall, implying that detection is impossible. According to the research in [110], the detection threshold was established as the minimum number of pixels a person

must occupy in an image, which was set to 25 pixels. This defines the smallest size at which humans can reliably recognize objects in images, and because CV models are trained using human annotations, they typically fail to learn to detect objects below this size. The height at which recall reaches zero was estimated by assuming that an average person occupies an area of about 0.5 by 0.25 meters when observed from above. Using this information, the height corresponding to an individual covering roughly 25 pixels in a 4K image with an aspect ratio of 16:9 was calculated. Based on the FOV and parameters for UAV A, utilizing no magnification ($1\times$ zoom), detection is determined to be impossible when the image is taken from heights greater than 222 meters. Because altitude control is implemented, the zero-recall point lies outside the intended sensing height range and is not particularly relevant. Despite this, it is specified to enable a fair comparison with a method that does not employ altitude control but uses the same sensing principle. Recall values from [61] and estimated recall values for humans, both in relation to height, along with the estimated zero-recall point are shown in Figure 8.1, panel (A).

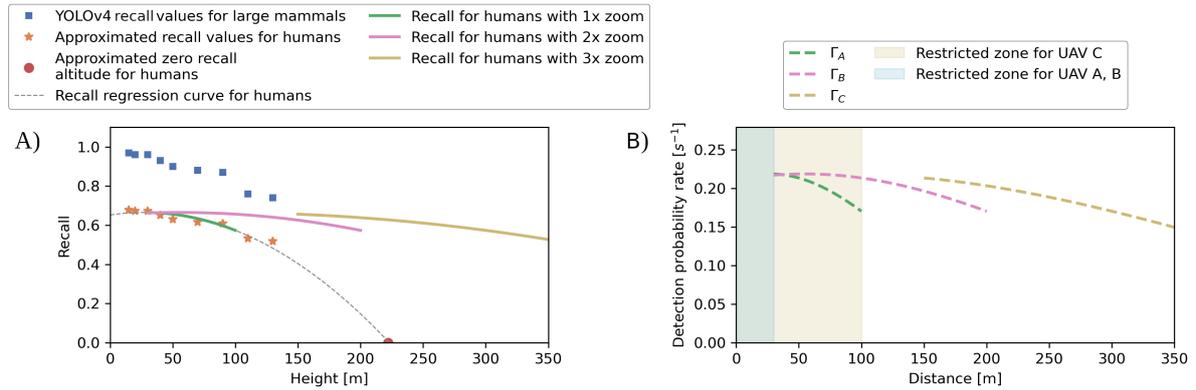


Figure 8.1: Dependence of recall on image capture height (A) and detection rate functions for UAVs used in the simulations (B).

Equation (4.13) was applied to determine the detection rate function, in which the recall function $\mu(\|\mathbf{R}\|)$ was produced by quadratically interpolating the measured recall-height points. It was assumed that $\|\mathbf{R}\|$ for the area within the captured image closely corresponds to the UAV's height. The value of t_{scene} was computed using equation (4.11), under the assumption that the average horizontal velocity $v_{s,avg}$ is 70% of the maximum horizontal velocity $v_{s,max}$.

To ensure equivalent detection performance when transitioning from UAV A to UAVs B and C, the Ground Sampling Distance (GSD) of the captured imagery must remain constant. GSD defines the real-world distance represented by a single pixel and it can be used to determine

the level of detail in the image. When using the same camera sensor at higher flight heights, optical zoom is employed to maintain the original GSD, ensuring the same coverage area and detection capability. Optical zoom, in contrast to digital zoom, magnifies the scene through lens optics without degrading resolution, effectively replicating the effect of lowering altitude by narrowing the FOV and enhancing image detail. For a zoom factor Z , the recall function is modified to $\mu(\|\mathbf{R}\|/Z)$, and the FOV angles are recalculated with

$$\gamma_Z = 2 \cdot \arctan\left(\frac{\tan \frac{\gamma}{2}}{Z}\right),$$

where γ denotes the FOV angle with zoom factor $1\times$, while γ_Z corresponds to the FOV angle at zoom Z . The detection rate of all UAV types, together with the interpolated recall-height functions for human targets across different optical zoom settings, are shown in Figure 8.1.

In each test case, the control interval Δt is equal to the sensing interval Δt_s . The undetected target probability is updated every Δt seconds by applying the sensing effect using ψ . The function ψ , defined in equation (4.10), is essentially the Γ function constrained to the visible area of the sensor. With each application of sensing, the captured snapshot of the area is considered observed for a duration of Δt_s seconds.

8.3 Plastic world

The first test case, named "Plastic world", consists of an artificially generated domain that contains key features of natural terrain modeled with simplified shapes. It is designed to provide a controlled environment with enough complexity to evaluate the algorithm's robustness. The initial probability of undetected targets is primarily concentrated over the depression and summit features. Access between these features is restricted by a no-fly zone positioned between them. In Figure 8.2, the terrain incline of the domain surface is shown together with the initial target probability distribution.

The domain is explored using 3 identical multi-rotor UAVs of type A, operating at a goal height of 50 m and a minimum height of 30 m. A detailed analysis of a trajectory flown by one of the UAVs from 600 s to 1600 s is presented in Figure 8.3. Panel (A) shows a three-dimensional representation of the terrain and all UAV trajectories, with the analyzed trajectory highlighted in red. Panels (B), (C), and (D) present the control parameters, velocities, and accelerations

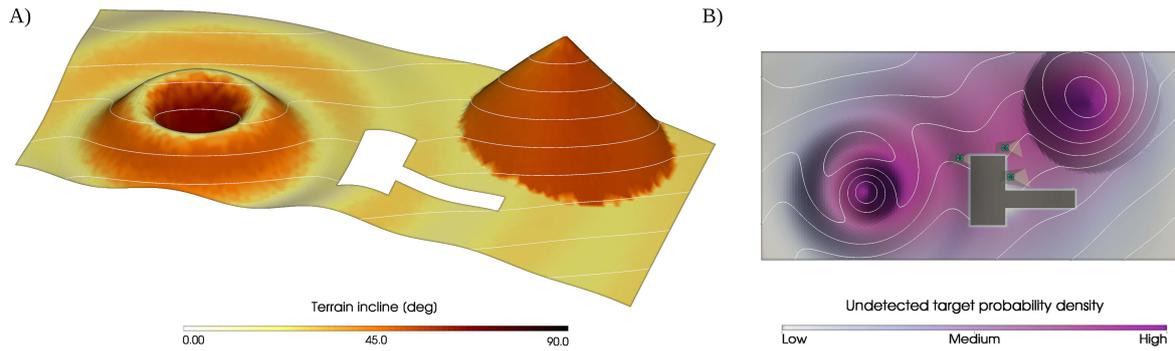


Figure 8.2: Illustration of terrain incline (A) and initial target probability density (B) for the Plastic world case.

for the same time window, respectively. Panel (E) depicts the terrain elevation and the UAV trajectory, with the corresponding goal and minimum flight height plotted in reference to the terrain elevation profile.

As shown in Figure 8.3, the control method frequently varies the control parameters and the velocities are adjusted accordingly. The velocities and accelerations are constantly within the operating limits, which indicates that the constraints are respected. As the terrain elevation profile becomes more complex, the complexity of the goal trajectory increases accordingly, since it is defined as an offset from the terrain. However, the UAV's executed trajectory starts to slightly diverge from the goal trajectory as complexity increases to create a smoother path, facilitating flight speed and efficiency (see Figure 8.3, panel (E), around 1200 s, 1300 s, and 1500 s).

The search mission simulation was conducted using three methods: HEDAC and lawnmower methods with MPC, and HEDAC without MPC (with UAVs flying at a constant altitude). The HEDAC + MPC framework achieved the best performance, accomplishing 98% survey accomplishment by the end of the 30-minute search mission. The lawnmower + MPC method ranked second, while the plain HEDAC method performed significantly worse. It stalled around 100 seconds into the search because the inspection height was too high for the equipped sensor, highlighting the importance of altitude control in search missions over hilly terrain. Survey accomplishment metrics for all tested methods are presented in panel (A) of Figure 8.4, alongside computation times for each procedure of the HEDAC + MPC framework over the time steps in panel (B). The total computation time per time step throughout the search remained more than 50% below the control interval $\Delta t = 1$ s, indicating the feasibility of real-time UAV control.

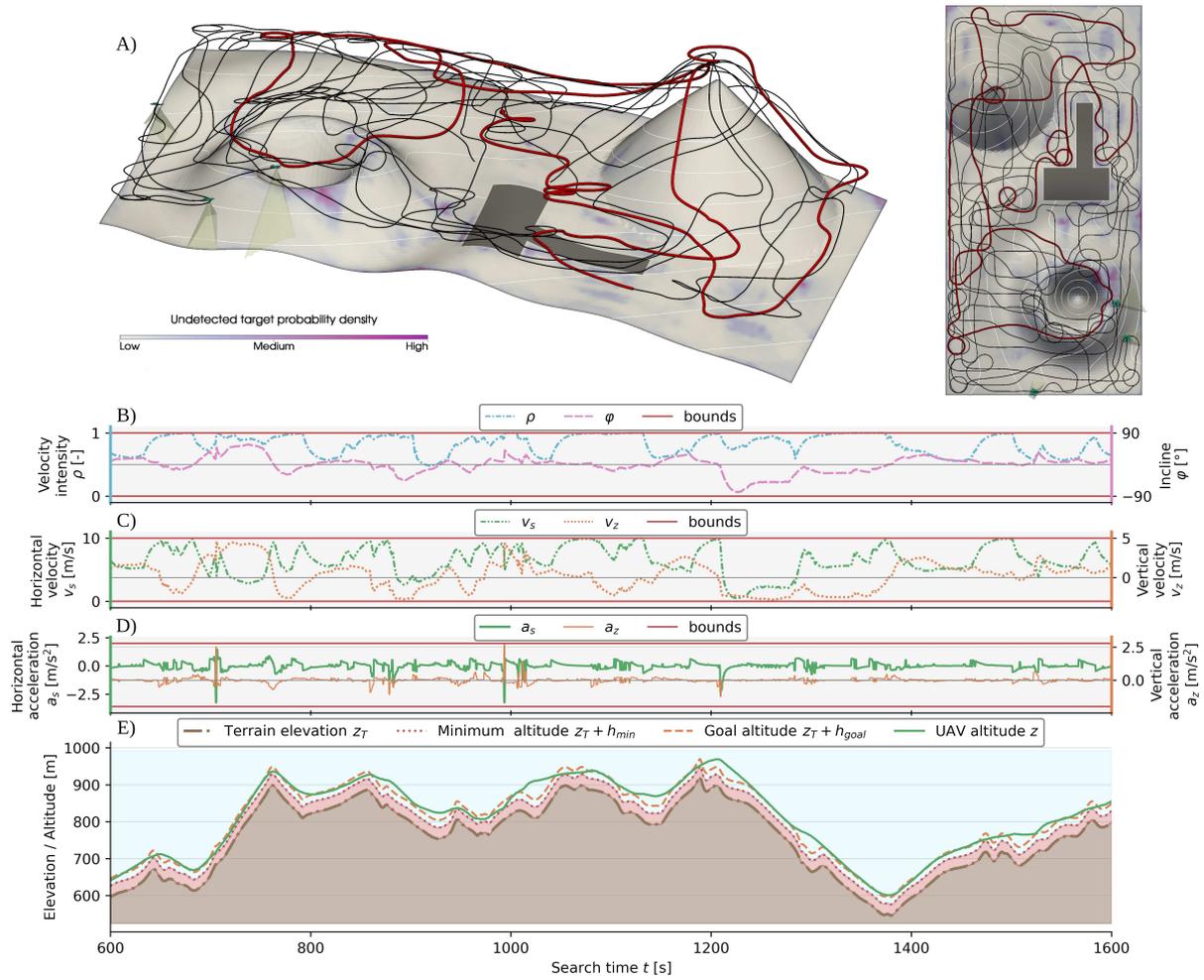


Figure 8.3: Analysis of a 1000-second trajectory for UAV A during exploration of the Plastic World case.

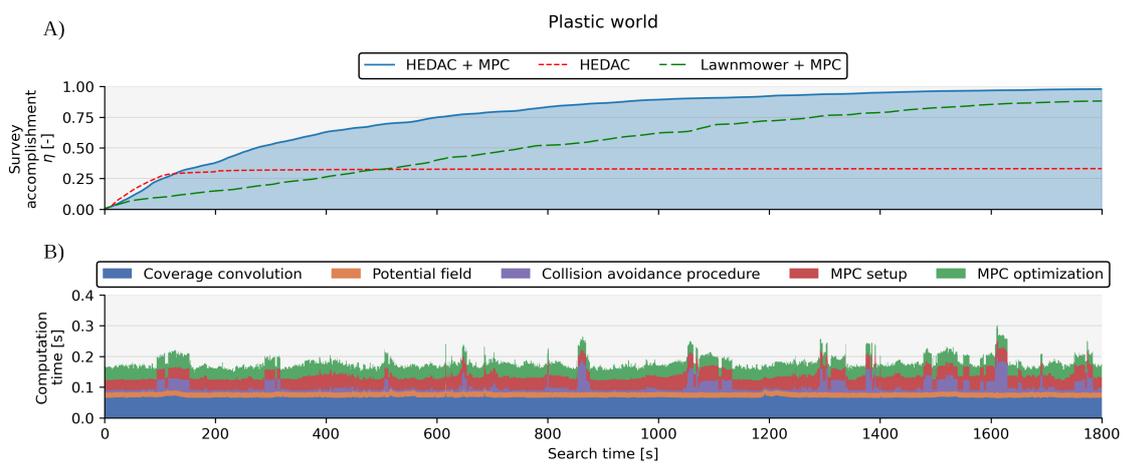


Figure 8.4: Survey accomplishment (A) and computation times (B) for the Plastic world case.

8.4 Mount Vesuvius

This scenario uses real terrain data from Mount Vesuvius located near Naples, Italy. It covers the area of the summit and the volcanic crater. Due to its popularity as a tourist site, it offers a realistic test environment for evaluating UAV-based missing person search methods. Figure 8.5 shows the terrain representation with incline and the initial probability distribution.

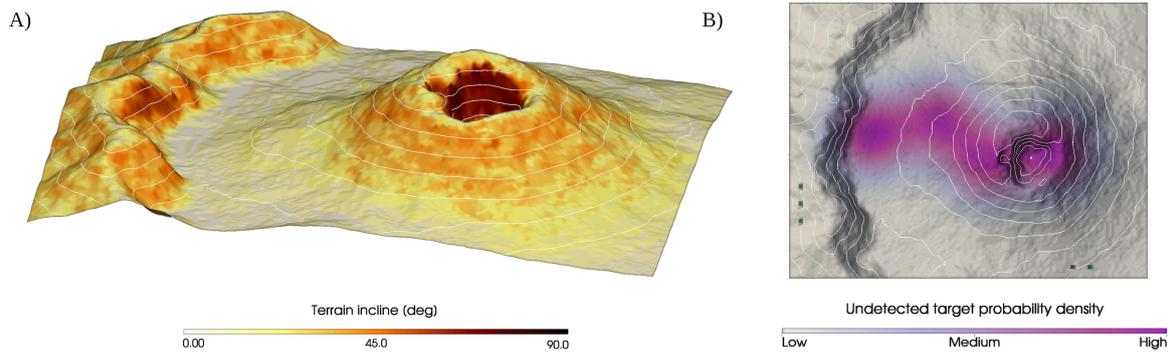


Figure 8.5: Mount Vesuvius terrain slope (A) and undetected target probability at $t = 0$ s (B).

The search mission is conducted using 5 multi-rotor UAVs, three of type A and two type B. While their flight characteristics are identical, they utilize different sensor configurations and operate at different goal heights. Nevertheless, their sensors cover the same ground area at the designated heights and maintain identical detection capabilities, as their FOV and detection rate functions are adjusted accordingly.

The flight trajectories of all UAVs at 1400 s into the search, computed using the coupled HEDAC MPC framework, are shown in Figure 8.6, panel (A). The trajectory of a type B UAV between 100 s and 1100 s, operating at a goal height of 100 m, is highlighted in red and analyzed in detail in panels (B–E), which present the control parameters, velocities, altitudes, and UAV height relative to the underlying terrain, respectively. The UAV generally maximizes its velocity intensity ρ while respecting velocity and acceleration constraints, and adjusts the incline parameter φ to follow the goal height. When encountering consecutive terrain elevation peaks, it prioritizes maintaining its current velocity over strict adherence to the goal height. This results in a smoother trajectory, with a slight altitude reduction at the peak to maintain good detection performance in the following valley, before preparing to climb the next peak, visible in panel (E) around 200 s and 550 s.

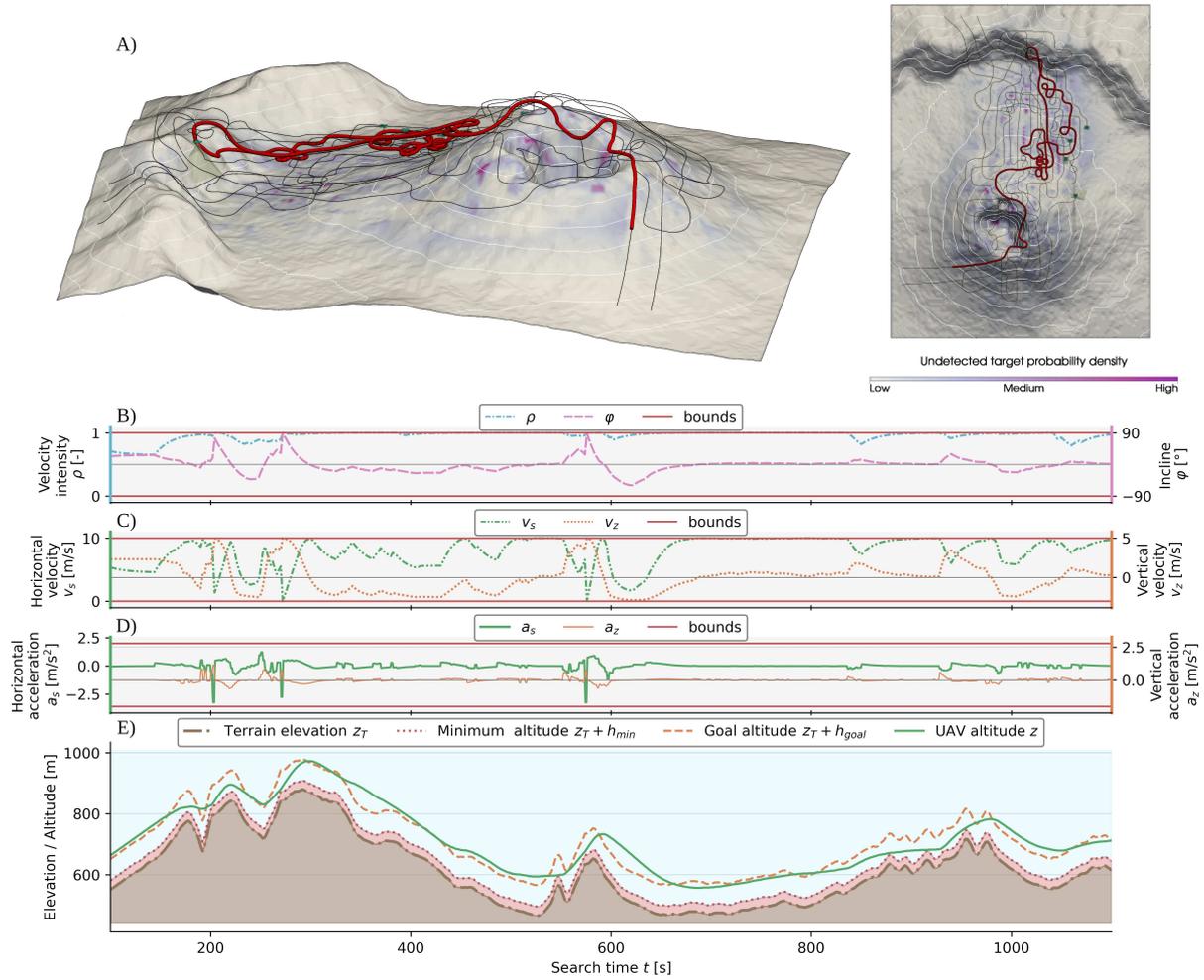


Figure 8.6: UAV trajectories at 1400 s of the Mount Vesuvius survey (A), with detailed analysis of the red-highlighted trajectory shown in panels (B–E) [107]

The area survey was performed using multiple methods, and the computed trajectories of all UAVs at the end of the 60-minute search are shown in Figure 8.7. Their performance was evaluated using η , and the results are shown in Figure 8.8, panel (A). The best performance was achieved with the HEDAC + MPC method, reaching $\eta = 96\%$ by the end of the search. The mSMC method delivered a very similar overall performance, and although the $\eta(t)$ curves for both methods appear nearly identical, a notable difference is observed when comparing the time required to reach $\eta = 90\%$, indicated by the vertical lines in the plot. The SMC and lawnmower methods achieved lower overall performance, while the plain HEDAC method performed the worst, stalling as in the Plastic World case. The computation time for the HEDAC + MPC method consistently remained under 0.5 s per iteration, well below the control interval of $\Delta t = 2$ s, further demonstrating the method’s feasibility for real-time UAV control.

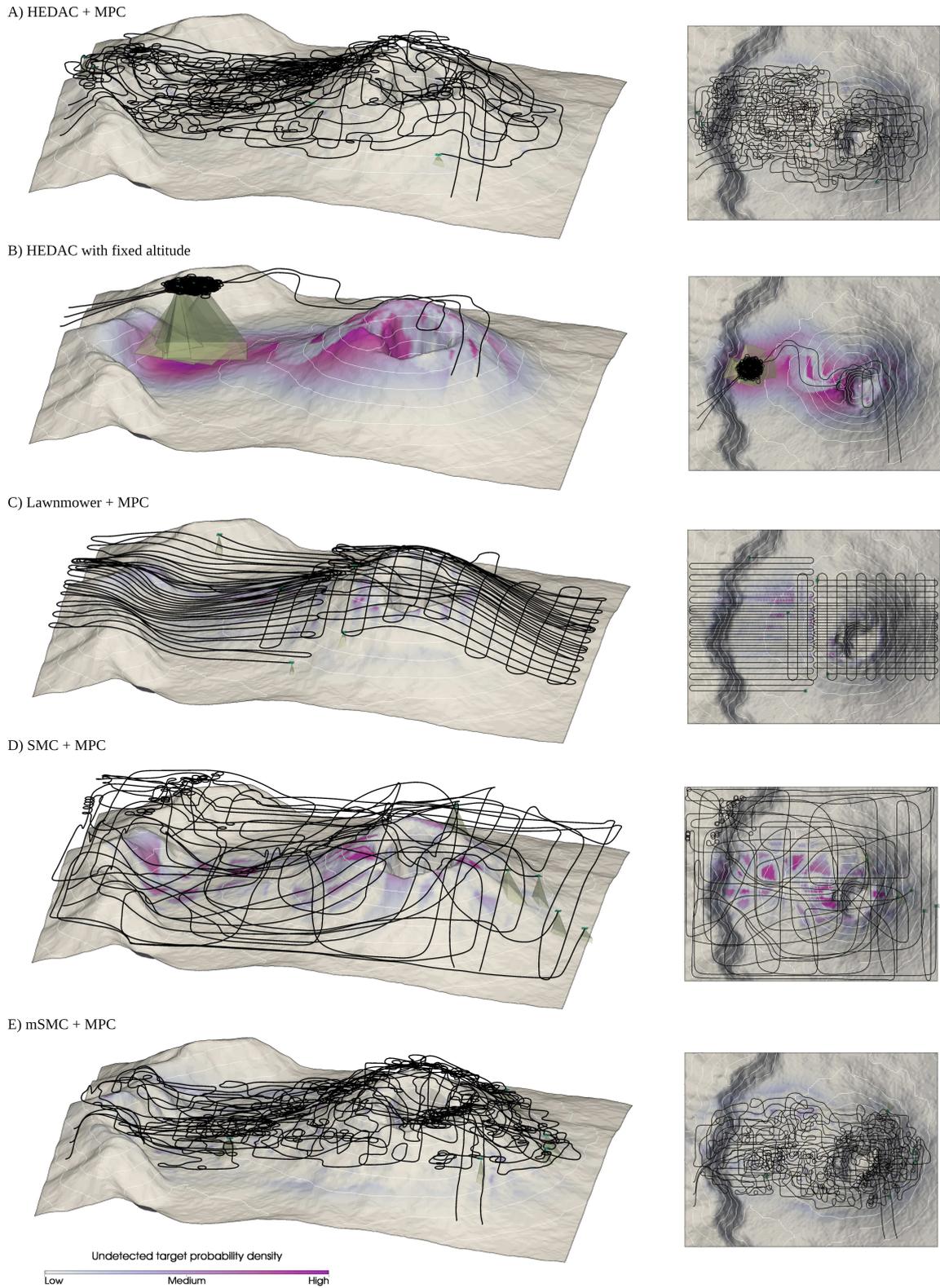


Figure 8.7: UAV trajectories for the Mount Vesuvius survey case across all compared methods.

An additional variant of this case was created by introducing a no-fly zone within the volcano crater area to demonstrate the capability of the HEDAC + MPC method to handle inter-domain obstacles and perform collision avoidance. The UAV configuration remained identical

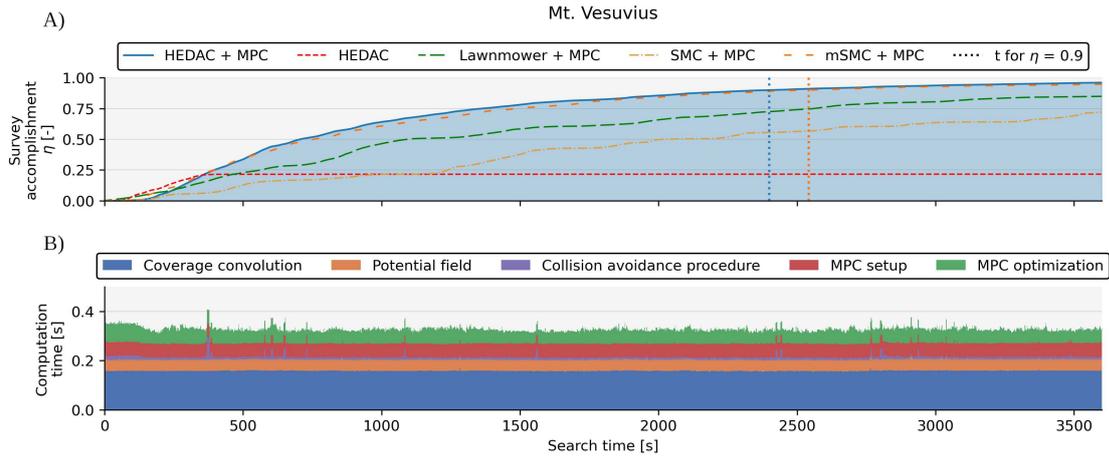


Figure 8.8: Mount Vesuvius case η across compared methods (A) and computation time per step for the HEDAC + MPC method (B).

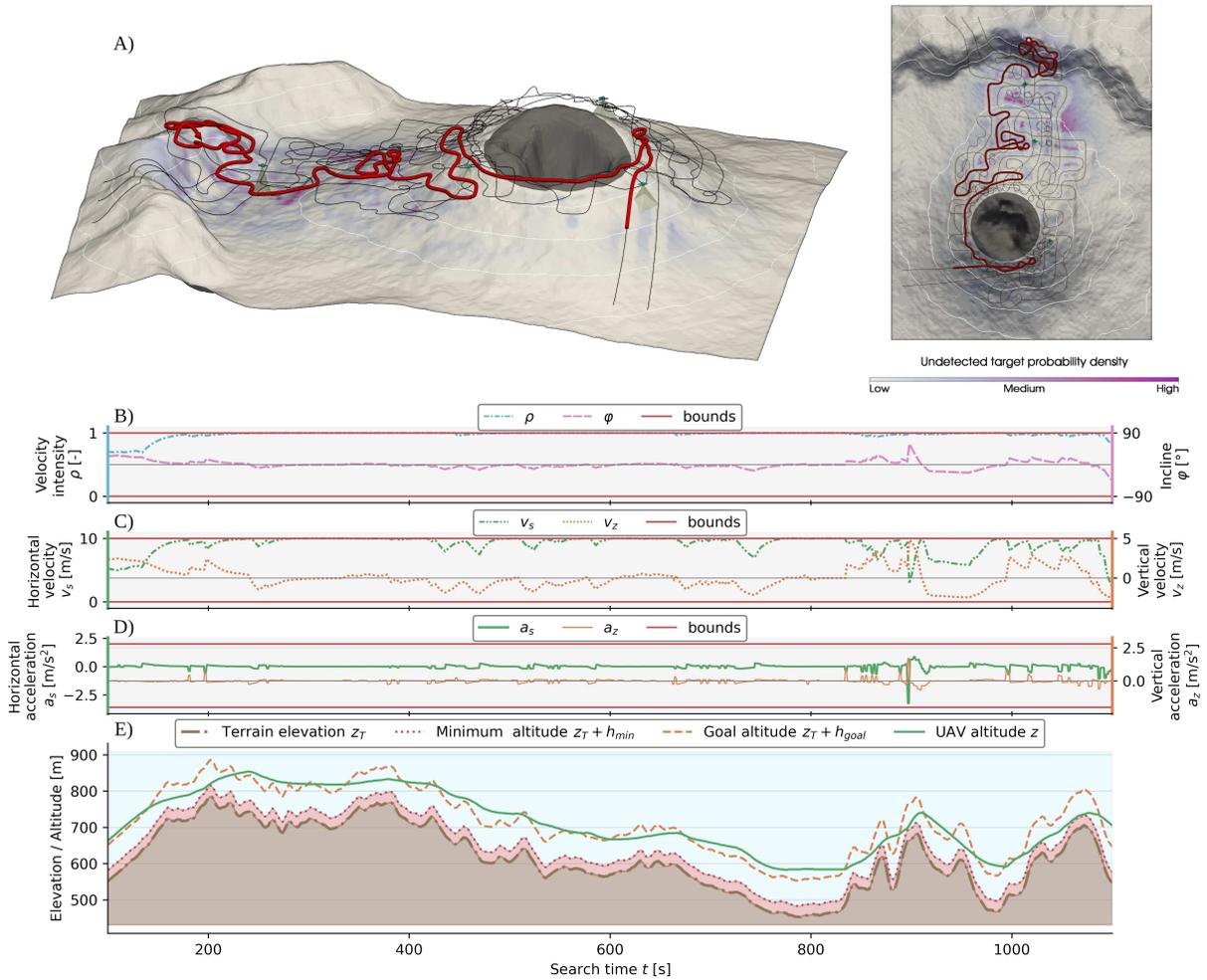


Figure 8.9: UAV trajectories during the Mount Vesuvius survey with a no-fly zone at 1400 s (A), including a detailed analysis of the red-highlighted trajectory shown in panels (B–E)

to that of the original case, with the only modification being the presence of the restricted area. The scenario analogous to the original case shown in Figure 8.6 is presented for the case including the no-fly zone in Figure 8.9. The trajectory highlighted in red corresponds to the same UAV, and behavioral differences are apparent when comparing the two figures. As seen in panel (A) of Figure 8.9, all UAVs successfully respect and avoid the no-fly zone. Although the analyzed trajectory differs, it exhibits similar dynamic characteristics to those in the original case, as demonstrated in panels (B-E).

8.5 Star dunes

This scenario features a specific desert terrain in Algeria. Due to environmental influences, the sand dunes develop star-like shapes with several arms branching out from the central peak, hence the name *Star Dunes*. The terrain does not have a large elevation difference between its highest and lowest points, and the slopes are moderate, as illustrated in Figure 8.10, panel (A). Therefore, this case is used to demonstrate a search operation using fixed-wing UAVs. The area is similar in size to the Mount Vesuvius case. However, since a different UAV type that can cover a larger area in the same amount of time is employed, only two UAVs are deployed instead of five. Both UAVs operate at a target altitude of 300 m, with sensor footprint and detection performance matched to the previous test cases by adjusting the detection rate function, sensor zoom level, and corresponding FOV angles. The search simulation is conducted for 60 min, and the initial undetected target probability is shown in Figure 8.10, panel (B).

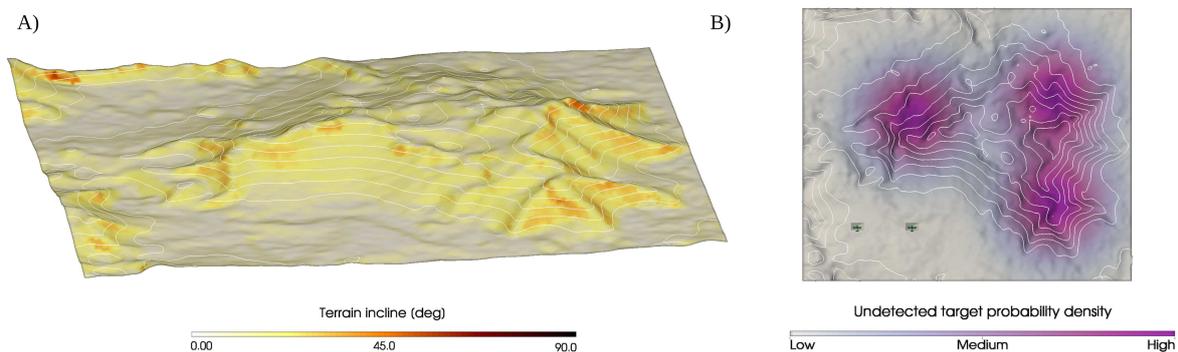


Figure 8.10: Star dunes terrain with incline representation (A) and initial undetected target probability density (B).

The UAV trajectories at $t = 1300$ s, computed with the HEDAC + MPC framework, are shown in Figure 8.11, panel (A). As in the previous cases, the red-highlighted trajectory is

analyzed over a 1000 s period, in panels (B-E) of the same figure. As displayed in panel (B), the UAV maintains the highest possible velocity intensity ρ throughout the observed time period, while adjusting the incline angle φ to distribute the velocity between horizontal and vertical components, as shown in panel (C). The velocity changes are applied gradually, as seen in the acceleration plot in panel (D), reflecting the lower maneuverability characteristics of fixed-wing UAVs. Given the UAVs' dynamic characteristics and the variability of the terrain elevation, the selected UAV would not be able to maintain the target search height precisely at all times. However, the optimization within the MPC framework smooths the trajectory, allowing the UAV to traverse the terrain while utilizing the maximal ρ , adhering to the minimum height constraint, and maintaining effective sensing performance by avoiding large deviations from the target altitude.

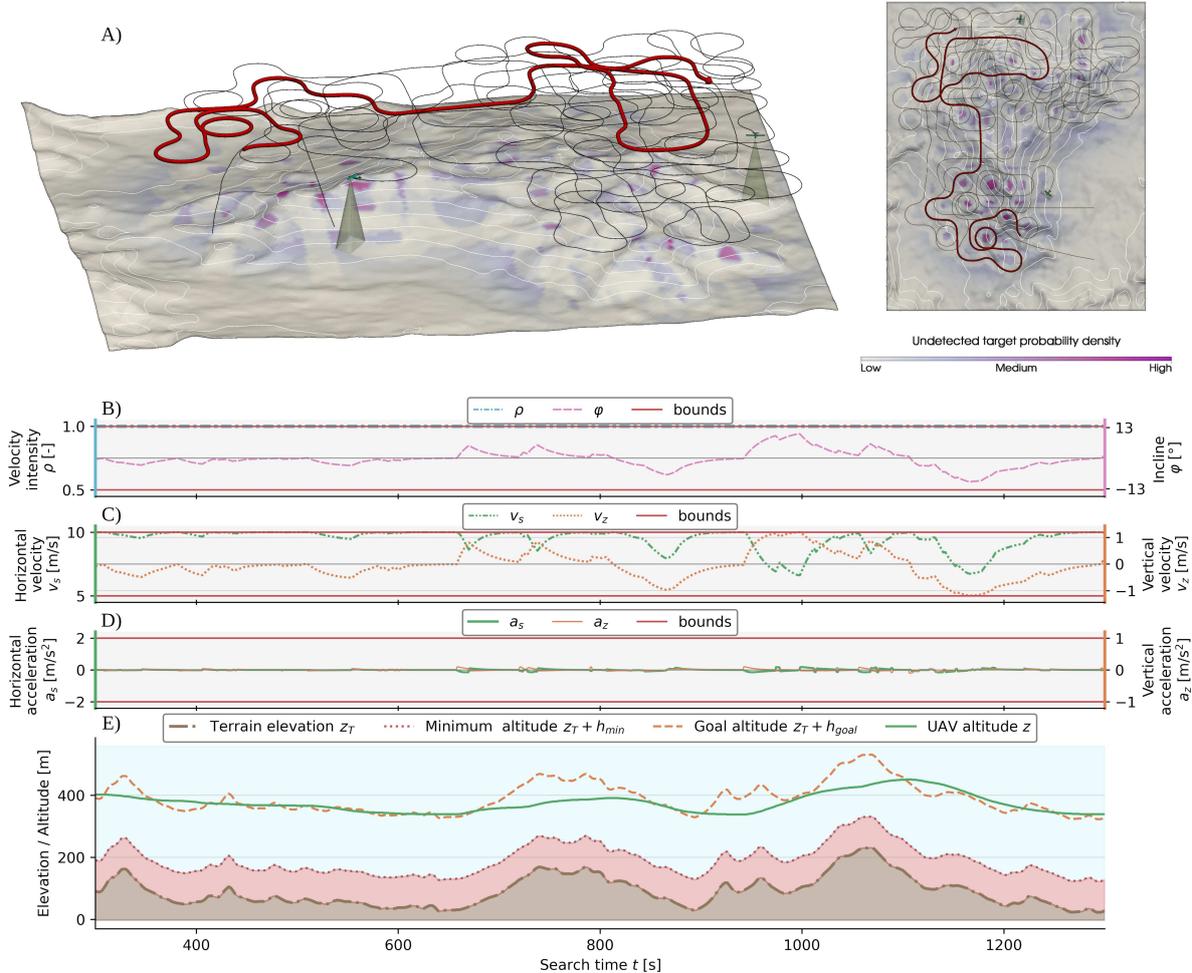


Figure 8.11: Star dunes survey UAV trajectories at $t = 1300$ s (A), along with analysis of flight parameters over the duration of the red-highlighted trajectory (B-E).

The search simulation was conducted using multiple methods, and the results showing $\eta(t)$ are displayed in Figure 8.12, panel (A). The HEDAC + MPC method again achieved the best results, while the mSMC method performed comparably. Since the $\eta(t)$ curves of the two best-performing methods are relatively close, vertical dotted lines in the corresponding colors are plotted to clearly indicate the time difference required for both methods to reach $\eta = 70\%$. All other methods showed a notable decrease in performance, as reflected by their $\eta(t)$ curves. Because the terrain elevation difference is not very large, the search using the HEDAC method at a constant altitude was able to perform sensing effectively and did not stall, as in the previous two cases.

The computation times for each time step of the HEDAC + MPC method are displayed in Figure 8.12, panel (B). It can be seen that the use of fixed-wing UAVs increases the time required for performing the collision avoidance procedure compared to the multi-rotor UAVs from the previous test cases, since they cannot reduce their horizontal velocity to zero and must plan the entire circular escape route. However, this does not pose significant issues because, given the lower maneuverability of fixed-wing UAVs, they do not require such frequent commands, and the control time step can be increased if necessary. Considering all of this, with a control time step of $\Delta t = 2$ s, the computation times never exceed 0.4 s, demonstrating that the feasibility of real-time UAV control is not compromised.

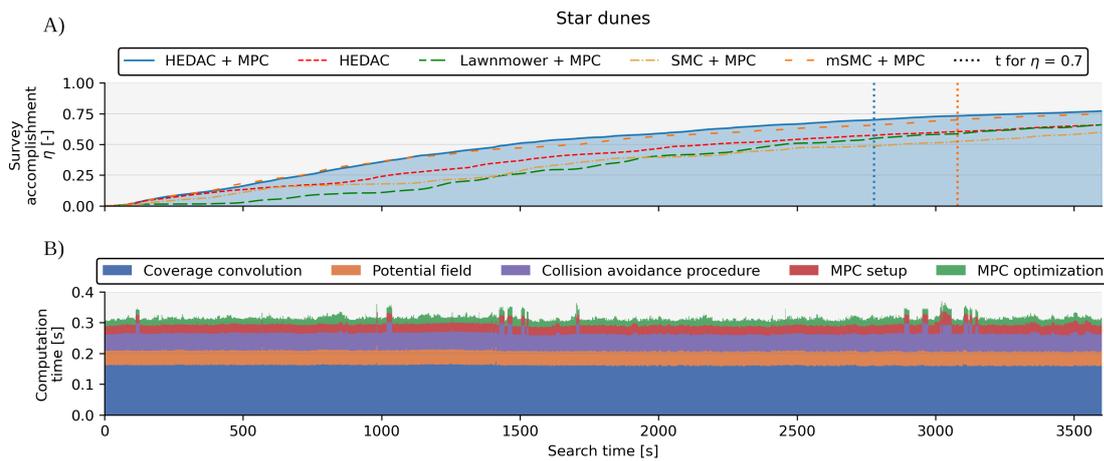


Figure 8.12: Star dunes case, η across methods (A) and HEDAC + MPC computation time per step (B).

8.6 Robustness analysis

In practical applications, there are numerous uncertainties that must be accounted for, such as potential flight control and localization errors. Flight control errors refer to deviations between the intended flight trajectory and the flight path actually executed by the UAV. They may occur due to external factors like wind disturbances and/or inaccuracies caused by the flight controller's response. On the other hand, localization errors involve discrepancies between the UAV's reported state (e.g., position, orientation, and velocity) and its actual state. Sensors, like GPS, barometers, or compasses are typically the source of those inaccuracies which lead to incorrect assessments of the UAV's position and heading.

To estimate the performance of the proposed method in a real-world scenario, the previously mentioned uncertainties are incorporated into the simulation by introducing error to the relevant variables. The errors are included in the corresponding control and location variables as additional random values, sampled from a Gaussian distribution with zero mean value, at each time step. The error range for each variable is represented by its standard deviation, as shown in Table 8.3. The minimum and maximum error values for each variable are capped at three times the standard deviation values.

Table 8.3: Variables' uncertainty error

Error source	Parameter	Standard deviation	Units
Flight control	v_s	0.1	m/s
	v_z	0.05	m/s
	ω	1	°/s
Localization	x	0.5	m
	y	0.5	m
	z	1	m
	θ	1	°

A total of 50 simulations of Mount Vesuvius UAV search simulations with included uncertainties are computed. The results are statistically evaluated and compared with the reference solution (presented in Section 8.4) in Figure 8.13. Graphs (A-1) and (B-1) showcase 1000 s of flight time for one UAV within one randomly selected simulated error test case, while the histograms (A-2) and (B-2) are generated using data from all simulated error cases, for all UAVs across the entire simulation duration. It can be concluded that the UAVs violated the velocity constraints for less than 10% of the simulation time, with the error consistently remaining below 0.5 m/s. The altitude constraint was violated in less than 0.1% of the time, with the deviation consistently staying under 5 m. Graphs (C-1) and (C-2) are also computed using data from all simulated error cases, for UAVs throughout the simulation duration. They display the distance

of the UAVs' paths to those from the ideal reference simulation, showing the absolute separation between the UAVs at each time step, highlighting the method's sensitivity to error and its impact on overall path accuracy. However, this has no significant effect on search success, as evident from the survey accomplishment graphs (D-1) and (D-2). Graph (D-1) was generated using all simulated error cases over the entire duration of the simulation, while the histogram (D-2) focused only on the values recorded after the search was completed.

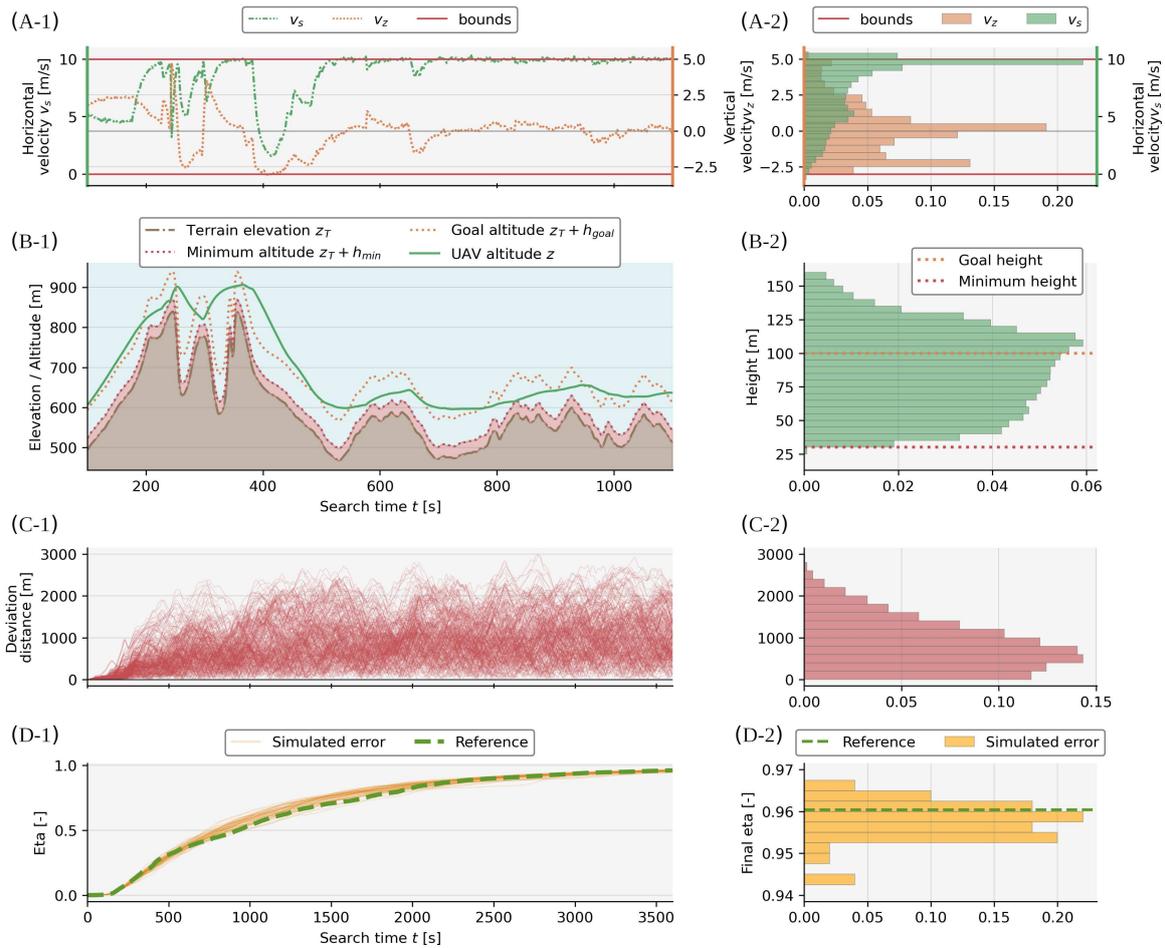


Figure 8.13: Statistical analysis of method robustness demonstrated on the Mount Vesuvius test case.

As expected, in the simulated error test cases, the UAVs occasionally break the altitude and velocity constraints. However, this did not significantly impact their flight performance since they corrected their velocity and altitude in subsequent time steps, converging back to the specified operational range. The UAVs exhibited dynamic fluctuations in their paths relative to the corresponding UAVs in the referent simulation, demonstrating the chaotic nature of the HEDAC method and its sensitivity to disturbances. Yet, this divergence did not adversely

affect the survey accomplishment metric, indicating the robustness of the method concerning uncertainty.

9 TERRAIN SEARCH EXPERIMENTS

This chapter presents the experimental validation of the proposed HEDAC + MPC control framework over terrestrial environments, chosen both for achieving the best performance in prior numerical tests and for its capability to operate within irregularly shaped search domains. First, the motion control strategy is evaluated to verify its ability to satisfy flight constraints and generate feasible trajectories over complex terrain. Following that, an additional experiment incorporating the UAV camera sensor, CV detection model, and search targets is conducted to validate the search methodology in a real-world scenario.

9.1 Experimental validation of UAV motion control

The first real-world flight tests were conducted to evaluate the HEDAC + MPC motion control in a complex terrain environment. The experiment was carried out in the Raša River valley in Istria County, Croatia ($45^{\circ}09'N$, $14^{\circ}03'E$), depicted in Figure 9.1. This location was selected because the flat riverbed area is bordered by steep, rugged slopes, making it a challenging environment for UAV motion control and navigation.



Figure 9.1: Illustration of the Raša River valley.

The three-dimensional representation of the terrain at the experimental site, including incline information, is shown in Figure 9.2, panel (A).

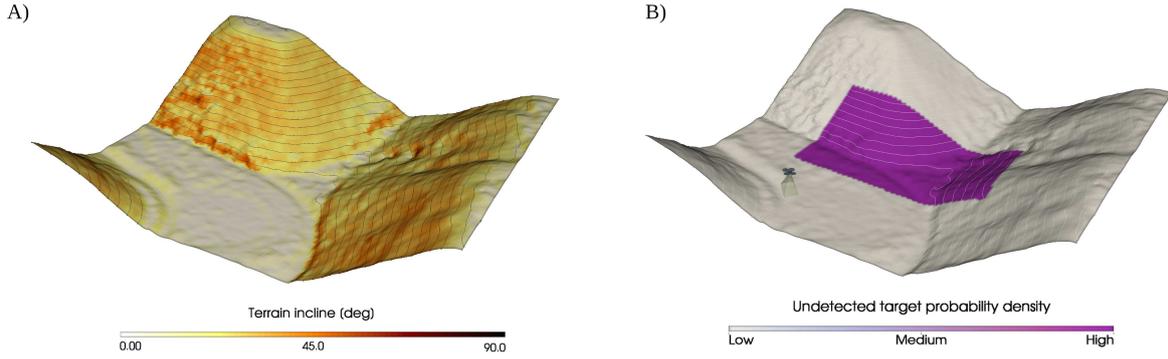


Figure 9.2: Motion control validation case: domain incline (A) and area of interest encoded in the undetected-target probability density field (B).

The domain is enclosed by an approximately rectangular polygon with an area of about 1.17 km^2 . The mission aims to systematically fly over a nearly rectangular (top-down view) area of interest of approximately 0.21 km^2 , maintaining a uniform trajectory density until the UAV’s battery is depleted. The area of interest is represented by a polygon with a uniform undetected target probability density, as shown in Figure 9.2, panel (B). The motion control is based on the undetected target probability field, and the algorithm models the sensing process even though no actual inspection is conducted. The UAV’s sensing characteristics, including camera FOV angles, zoom factor, and sensing function, match those of UAV A (Table 8.1) used in the simulated test cases *Mount Vesuvius* and *Plastic World*.

A total of four single-UAV flight missions were conducted using the DJI Matrice 210 v2 aircraft. The control interval was set to $\Delta t = 3 \text{ s}$, while the HEDAC parameters were set to $\alpha = 5000$ and $\beta = 0.1$. All UAV motion characteristics were configured to match those of UAV A (Table 8.1), except for the maximum horizontal velocity, prediction time steps, and minimum and goal heights. For safety reasons, the maximum horizontal velocity was reduced to $v_{s,max} = 5 \text{ m/s}$, while the minimum and goal heights were increased to $h_{min} = 35 \text{ m}$ and $h_{goal} = 55 \text{ m}$. For additional analysis, two different values of the MPC prediction horizon τ_{max} were tested: 15 s and 30 s, corresponding to prediction step counts n_{pts} of 5 and 10, respectively. For both values of τ_{max} , simulation results were computed and compared with the executed real-world flights. All comparisons were performed over a flight duration of 20 minutes, as the actual durations of the real-world flights slightly exceeded this interval.

A representative flight mission with a prediction horizon of $\tau_{max} = 30 \text{ s}$ is shown in Figure 9.3. Panel (A) presents the UAV trajectory and the terrain from two views, while panels

(B–E) show analysis graphs of the trajectory over the full flight duration. As seen in panel (B), the UAV tends to maintain high velocity intensity ρ when flying horizontally or descending, while ρ decreases during steep ascents. The velocity and acceleration values remain within the prescribed constraints (panels C and D, respectively), and the absence of sudden, intense jumps in the acceleration curve indicates that the executed flight trajectory is smooth. Although the manufacturer specifications state that the Matrice 210 v2 has a maximum flight time of 34 minutes, the mission lasted only about 22 minutes before the UAV reached a critical battery state and automatically initiated the Return To Home (RTH) procedure. The high energy consumption may be attributed to the ~ 100 m elevation difference within the area of interest and the UAV’s tendency to follow the goal search height, which resulted in frequent ascents and descents, as seen in panel (E).

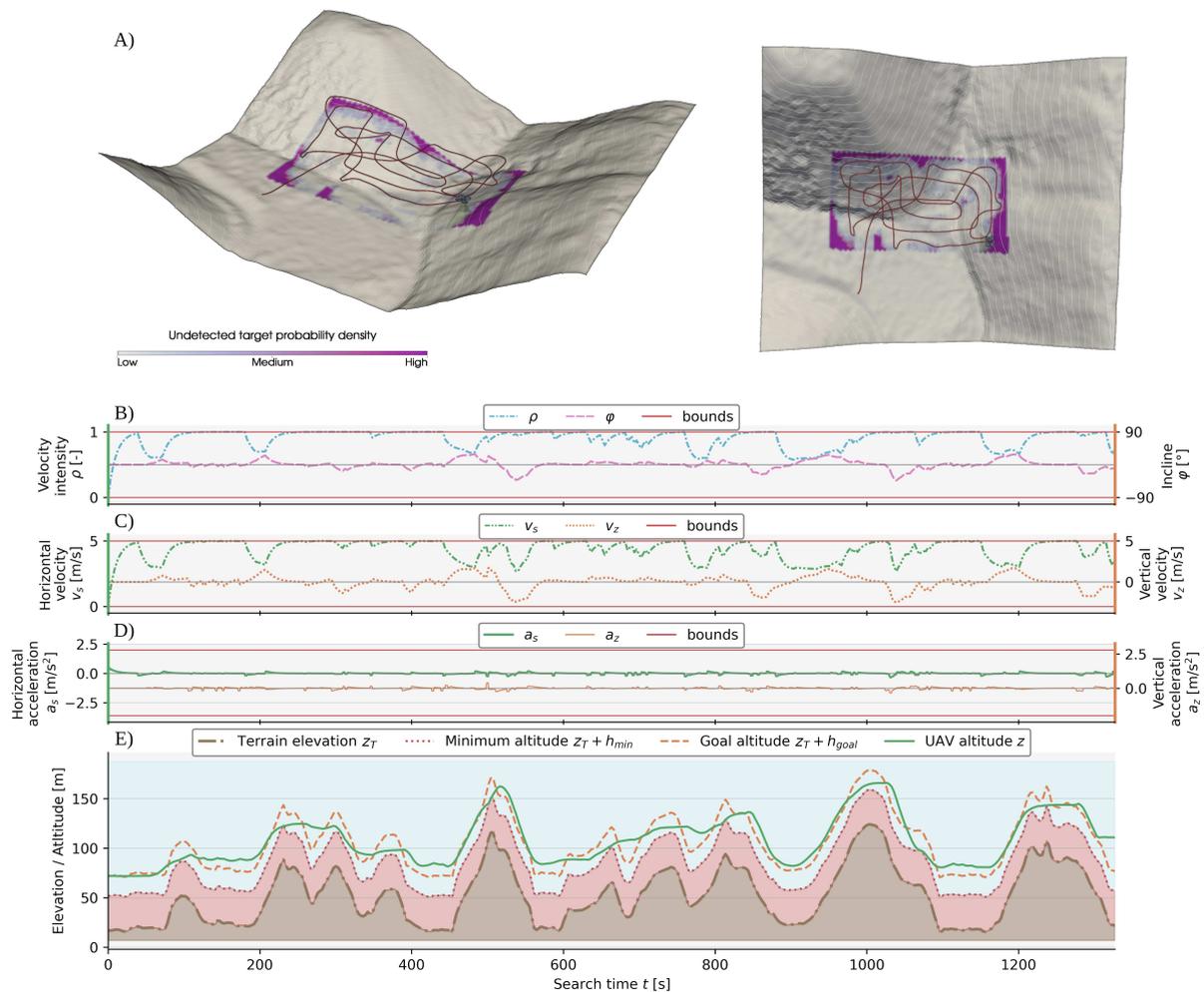


Figure 9.3: UAV trajectory for a single flight mission during motion-control validation with prediction horizon $\tau_{max} = 30$ s (A), and corresponding trajectory analysis (B-E).

As the UAV moves from its position at time t to the position at $t + \Delta t$ during uninterrupted flight, the flight regime is simultaneously computed for the next control interval, from $t + \Delta t$ to $t + 2\Delta t$. To perform this computation, the position at $t + \Delta t$ is required to sample the potential field for determining the new heading direction and calculating the yaw angular velocity, as well as for the MPC procedure and collision avoidance. Since only the current position at time t is known, the future position at $t + \Delta t$ is approximated by simulating movement from the current position using the commanded control parameters $\rho(t)$ and $\varphi(t)$. Since the UAV cannot perfectly execute the commanded flight regime and external factors such as wind may introduce deviations, an error arises between the calculated position and the actual position at $t + \Delta t$. This predicted position error is defined as the Euclidean distance between the planned and achieved positions. Figure 9.4 presents a histogram of these errors computed using data from all executed flight missions, where each bin represents an error range and the height indicates how frequently that error occurred. Note that the UAV position is measured using GPS and a barometric altimeter, both of which are subject to measurement errors. These localization errors are present both as inputs when calculating the next position and in the reported achieved position used to evaluate the predicted position error. As shown in Figure 9.4, the predicted position error is within 1 m for the majority of the time and remains below 3 m in nearly all cases, which is acceptable considering that the UAV can traverse a 15 m trajectory when utilizing the maximum horizontal velocity $v_{s,max}$ over the 3 second control interval Δt . However, in some cases the observed error increases to almost 8 m.

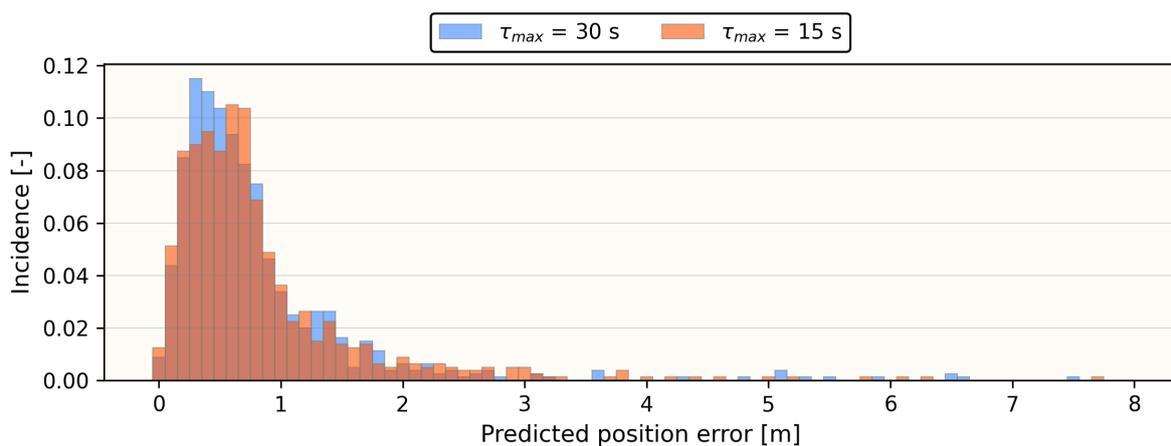


Figure 9.4: Predicted UAV position errors during all real-world flight experiments.

To assess how the system handles both UAV localization errors and predicted position errors,

the trajectories from the executed real-world flights were compared to the corresponding simulated trajectories. The location error is defined as the Euclidean distance between the UAV's position in the real-world flight and the position in the simulation, where the simulated trajectory is considered to have zero location error. Comparisons for both $\tau_{max} = 30$ s and $\tau_{max} = 15$ s with their respective reference simulations are shown in Figure 9.4. It can be observed that flights with a shorter prediction window of $\tau_{max} = 15$ s achieve lower errors, and the two real-world flights are fairly consistent with each other. In contrast, a longer prediction window results in higher errors, and the corresponding simulations are less consistent. These deviations between the simulated and real-world trajectories arise from the combination of UAV localization errors and prediction errors, causing the potential field u to be sampled at different points, which in turn produces variations in the yaw angular velocities ω . However, even though the errors reach relatively high values (considering the area of interest of 0.21 km^2), these errors do not affect the search performance, as shown in Figure 9.6. In all cases, whether simulated or real-world and regardless of the prediction horizon, the $\eta(t)$ curves exhibit similar behavior, indicating the method's robustness to localization errors.

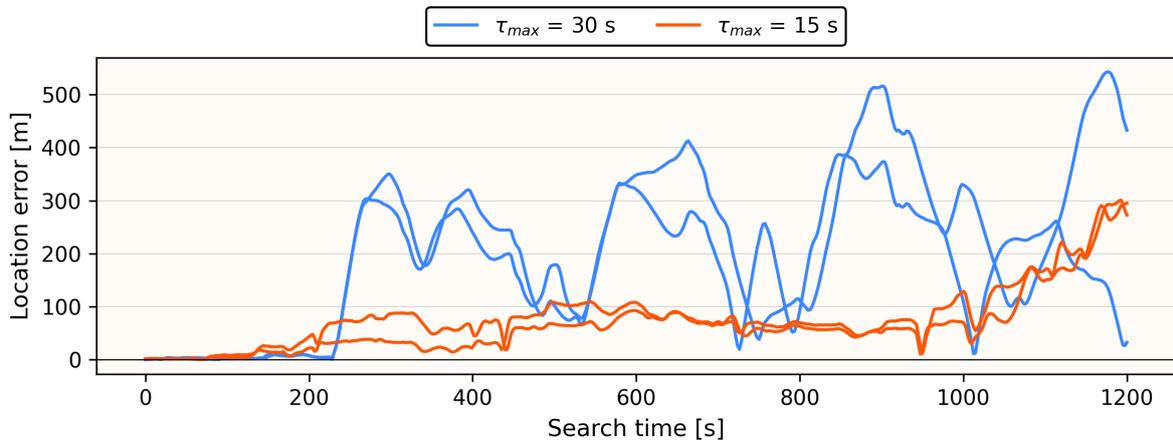


Figure 9.5: Location error between the simulated trajectory (zero error) and the executed real-world trajectory for both prediction horizon lengths of 15 s and 30 s.

Additionally, a detailed analysis of flight height for all executed real-world flights, compared with simulation results, is shown in Figure 9.7. Panel (A) presents the results for $\tau_{max} = 30$ s, while panel (B) presents the results for $\tau_{max} = 15$ s. The shorter prediction horizon achieved better results with respect to adherence to the target flight height, both in simulation and in the real-world tests. Additionally, the total number of instances in which the height constraint was

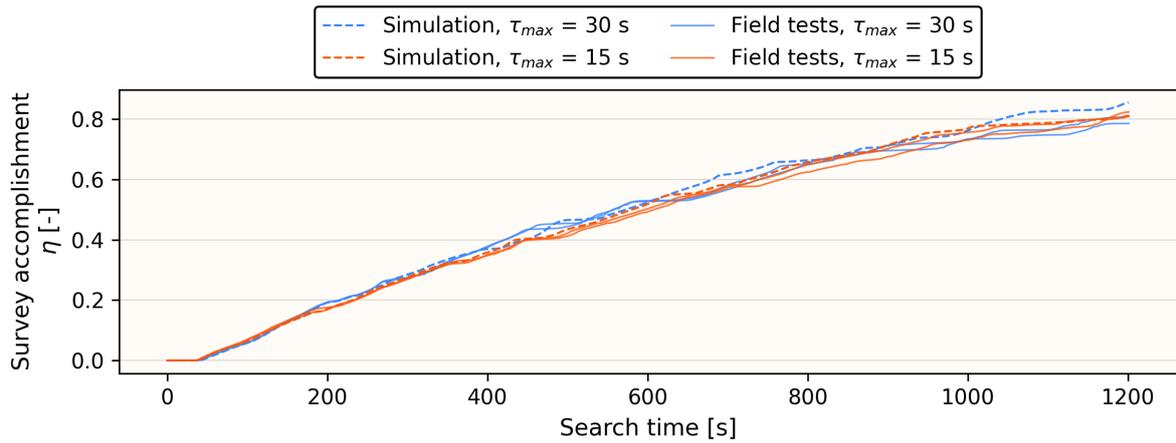


Figure 9.6: Survey accomplishment for all simulated missions and real-world missions, for both prediction horizon lengths of 15 s and 30 s.

breached during the two executed flights was 2, compared to 6 instances for the larger prediction horizon. Nevertheless, these minor breaches of the height constraint do not pose a significant issue for real-world operation, as they remained below 2 m.

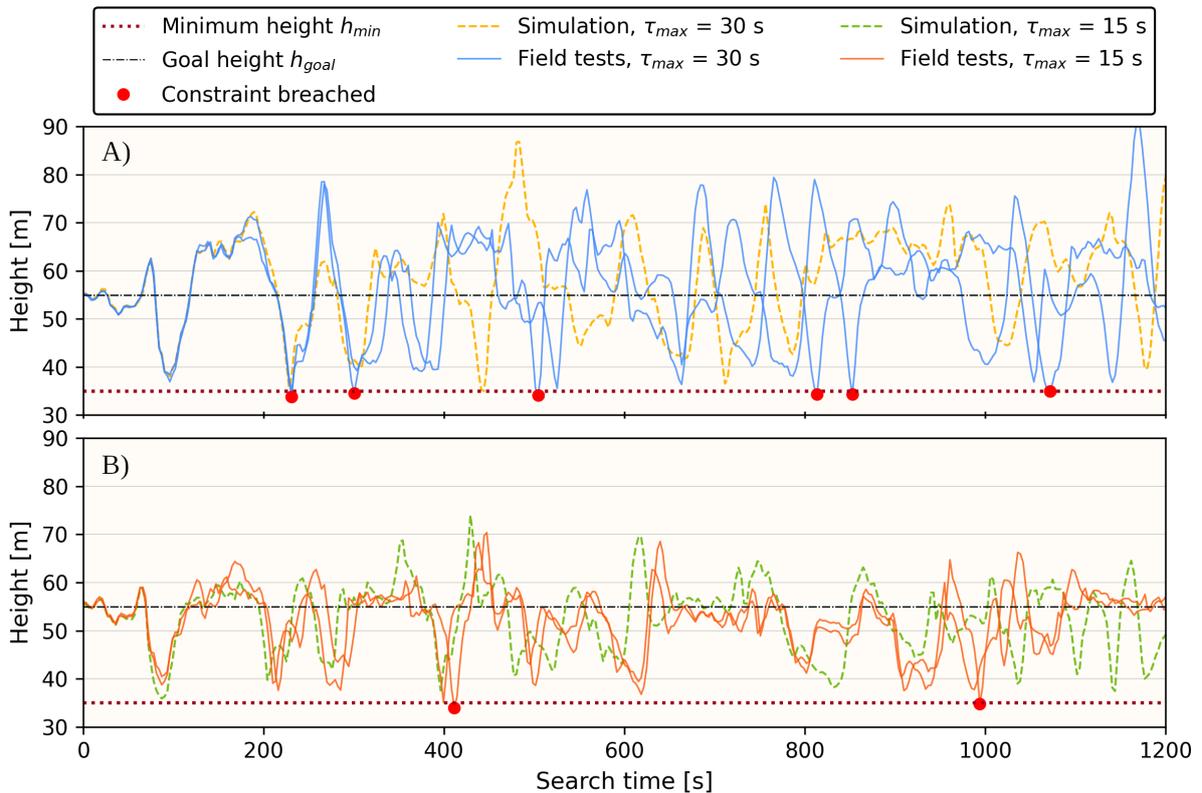


Figure 9.7: Flight height analysis and comparison with simulation results for $\tau_{max} = 30$ s (A) and $\tau_{max} = 15$ s (B).

The shorter prediction horizon achieved slightly better results in several of the conducted

analyses, but reducing the prediction time indefinitely is not a viable option. An excessively short prediction horizon, relative to the aircraft's velocity and acceleration limits, may prevent the UAV from traversing complex terrain or climbing steep obstacles. Therefore, the MPC horizon must be determined with consideration of both the aircraft's dynamic characteristics and the terrain complexity. In practice, selecting a longer prediction horizon is preferable for safety, as it still provides good search performance, as shown in Figure 9.6.

9.2 Experimental validation of UAV search methodology

In order to validate the implemented search methodology, it is necessary to assess whether the actual target detection performance aligns with the survey accomplishment metric η . To achieve this, search targets and a corresponding detector must be introduced into the experiment.

A total of 100 experimental targets are made from 0.5×0.5 m cardboard sheets. Each of the 100 targets is uniquely patterned marked with two colors selected from the following palette: white, black, green, orange, red, blue, and yellow. The color combinations were chosen to simulate the variety of clothing and appearances that real humans may have, aiming to achieve detection performance comparable to that for actual humans. The targets are practical because they can be positioned within the area such that their distribution matches the undetected target probability density, and they remain stationary throughout the experiment. Examples of the targets are presented in Figure 9.8.



Figure 9.8: Representative custom cardboard targets used in the experiment.

Target detection is achieved by having the UAVs capture images with the onboard camera. The acquired images are then processed with a CV detection model to identify which targets have been successfully detected. Since no detector existed for the custom targets, a model was specifically trained to enable their detection. The YOLO detection algorithm was chosen because it provides fast detection and is widely used for UAV imagery [58, 111].

For model training and performance analysis at varying image capture heights, a data set was collected consisting of 1840 images, containing a total of 34135 target instances. A subset of 1166 images, containing 27600 object instances, was used to train the model. This subset was split into training and validation sets in an 80-20 ratio. The YOLOv8 architecture was employed, and training was initialized with the extra-large pre-trained model *yolov8x.pt*, previously trained on the COCO dataset. The model was trained for 500 epochs with a batch size of 2, with the original images automatically scaled down to 1280 pixels by the YOLO algorithm.

The remaining images from the data set were grouped into bins corresponding to capture heights in the range [30, 90] m, with each bin spanning 10 m. For each bin, validation was performed to compute the recall metric, and the recall value for the bin was assigned to the mean height of the bin. To obtain recall values for the full operational height range of the aircraft, a quadratic polynomial regression was performed. The discrete recall values for each bin and the corresponding recall regression function μ are shown in Figure 9.9, panel (A).

Two types of UAVs were used in the experiments: DJI Matrice 210v2 and DJI Mavic 2 Enterprise Dual. Their motion, sensing, and control parameters are listed in Table 9.1. For each UAV type, the average velocity $v_{s,avg}$ during autonomous flight over the search domain was experimentally determined. The Matrice achieved an average velocity of 7.515 m/s, while the Mavic achieved 7.449 m/s. Based on the measured $v_{s,avg}$, the goal search altitude h_{goal} , and the camera FOV angle γ_2 , the scene duration t_{scene} was computed according to equation (4.11). The resulting t_{scene} was 5.736 s for the Matrice 210v2 and 8.775 s for the Mavic 2 Enterprise Dual. Finally, the detection rate function Γ was computed for each UAV type using the recall function μ and the scene duration t_{scene} according to equation (4.13). The resulting detection rate functions are illustrated in Figure 9.9, panel (B).

The experiment was conducted on Učka Mountain, Croatia (45° 14' 30'' N, 14° 12' 00'' E), and consisted of two search missions. The first mission employed two Matrice UAVs, while the second mission utilized one Matrice UAV and one Mavic UAV. Both Matrice UAVs were equipped with the DJI Zenmuse X5S RGB camera, whereas the Mavic UAV used its integrated

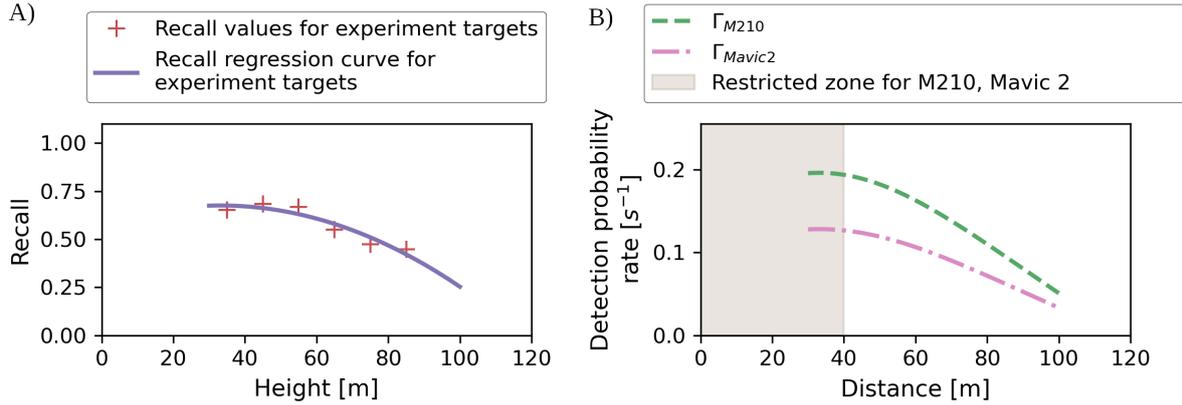


Figure 9.9: Recall values of the cardboard target detection model at discrete heights and the corresponding regression curve (A), along with detection rate functions Γ for the UAVs used in the experiments (B).

Table 9.1: Motion, sensing, and control parameters of the UAV used in the search methodology validation experiment [107].

UAV parameters	DJI Matrice 210 v2	Mavic 2 Enterprise Dual	Units
Type	Multi-rotor	Multi-rotor	-
Min turning radius R_{min}	30	30	m
Min clearance distance δ	50	50	m
Min search altitude h_{min}	40	40	m
Goal search altitude h_{goal}	60	60	m
Max horizontal velocity $v_{s,max}$	8	8	m/s
Min horizontal velocity $v_{s,min}$	0	0	m/s
Max ascending velocity $v_{z,max}$	5	3	m/s
Max descending velocity $v_{z,min}$	-3	-2	m/s
Max horizontal acceleration $a_{s,max}$	2	2	m/s ²
Min horizontal acceleration $a_{s,min}$	-3.6	-3.6	m/s ²
Max vertical acceleration $a_{z,max}$	2.8	2.8	m/s ²
Min vertical acceleration $a_{z,min}$	-2	-2	m/s ²
Min incline ϕ_{min}	-90	-90	°
Max incline ϕ_{max}	90	90	°
Camera FOV γ_1	64.7	72.5	°
Camera FOV γ_2	39.2	57.58	°
Sensing function Γ	Γ_{M210}	Γ_{Mavic2}	-
Prediction time steps n_{pts}	5	5	time steps

RGB camera. More detailed specifications of the cameras, as well as the UAV control implementation, are provided in Section 3.4. Both search missions continued until one of the UAVs reached its critical battery level. The experimental site on Učka mountain, along with two Matrice UAVs, is shown in Figure 9.10.

The control time step Δt was set to 3 s, which corresponds to the the sensing interval Δt_s . This interval accounts for the time required by the camera to generate an image, including performing auto-focus, capturing the image, and storing it in memory. In case a new image capture command is issued before the previous command has completed, the command is ignored and the subsequent image is not recorded. The sensing interval $\Delta t_s = 3$ s allowed successful image



Figure 9.10: Experimental area on Učka Mountain including two Matrice 210 v2 UAVs.

captures in most cases. However, if an image was not acquired for any reason, the corresponding sensing effect was not applied to the domain. The values of the HEDAC parameters were set to $\alpha = 5000$ and $\beta = 0.1$.

The search domain was defined as a circular area with a radius of 550 m, corresponding to an area of 0.95 km^2 . The terrain elevation variation within the domain is approximately 100 m, and the terrain, including incline representation and elevation isolines, is illustrated in Figure 9.11, panel (A). A total of 100 search targets were distributed across three concentric zones. Within each zone, the targets were uniformly scattered, and the initial undetected target probability density m_0 matched the target distribution. 20 targets were placed in Zone 1 ($r_{z1} < 150\text{m}$), 30 targets in Zone 2 ($150 < r_{z2} < 300\text{m}$), and 50 targets in Zone 3 ($300 < r_{z3} < 450\text{m}$), where r_z defines the radial limit of each zone. The probability within each zone was uniform and calculated as the ratio of the number of targets in the zone to the corresponding zone area. The resulting distribution was then normalized to satisfy the condition defined in equation (4.5). The initial undetected target probability distribution and the locations of the search targets, shown as black dots, are illustrated in Figure 9.11, panel (B).

After the images are taken, they are processed with the detection model to determine the detected targets. The model is trained to detect only one class, named *Target*, which represents a binary classification problem of object versus background. The detection model can produce four possible classification outcomes: a True Positive (TP) occurs when the target object is

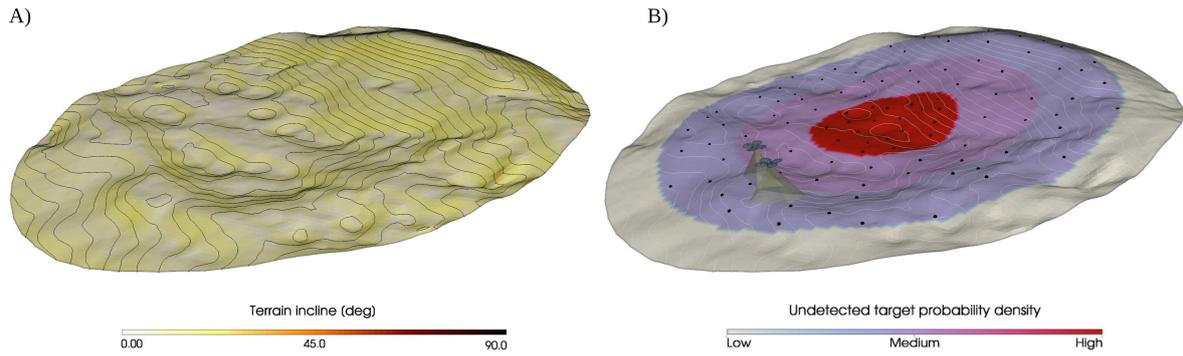


Figure 9.11: Učka experiment terrain (A) and the initial undetected target probability density with target locations shown as black dots (B).

correctly detected, a False Positive (FP) is the result of the model indicating a target when none is present, a False Negative (FN) occurs when there is a target present and the model fails to detect it, and a True Negative (TN) represents the case where no target is present and no detection is indicated by the model. Examples of classification outcomes on the collected images from the experiment are shown in Figure 9.12. True positives and false positives are marked with blue labels produced by the model in panels (A) and (B), respectively. False negative detections are manually marked in red, while true negatives correspond to regions with no annotations.

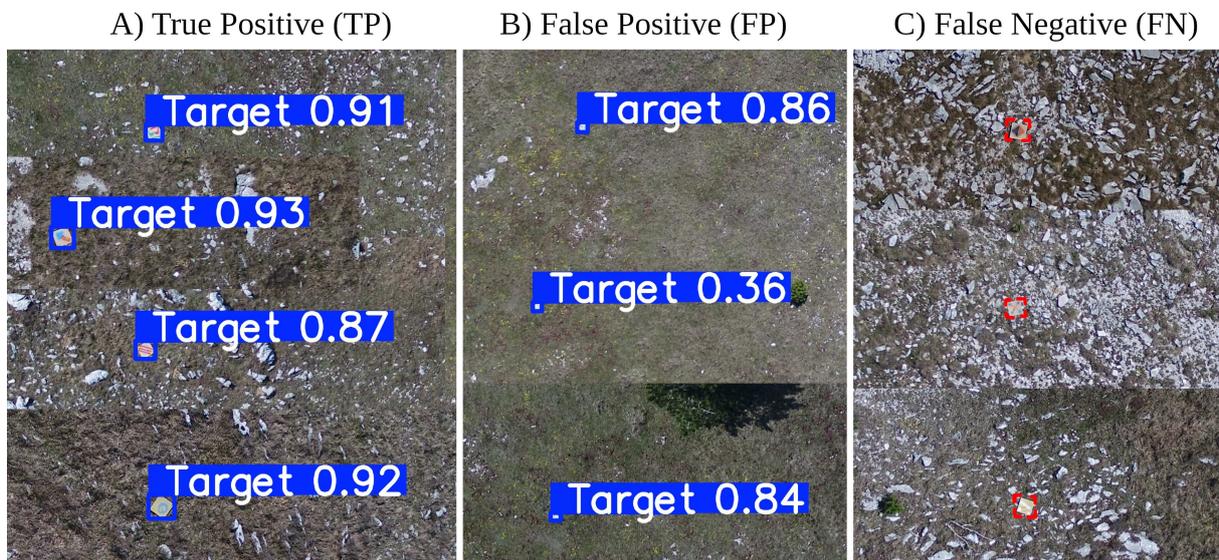


Figure 9.12: Detection model classification outcomes: true positives (A) and false positives (B) annotated by the model in blue, and false negatives (C) manually annotated in red. True negatives are represented by regions with no annotations in the images. Numbers next to the annotations in (A,B) indicate the model's detection confidence.

Flight trajectories during Mission 1, along with a trajectory analysis for one of the Matrice UAVs, are shown in Figure 9.13. The search was conducted for 25.26 minutes, and the targets' detection status and trajectories at the end of the search are displayed in panel (A). The green dots represent the detected targets, while the black dots represent the undetected targets. Flight parameters along the blue trajectory are presented in graphs (B–E). All set constraints for flight parameters (B), velocities (C), accelerations (D), and flight height (E) were respected. The UAV generally maintained the target height, balancing area coverage and sensing performance, while executing a relatively smooth trajectory at high velocity.

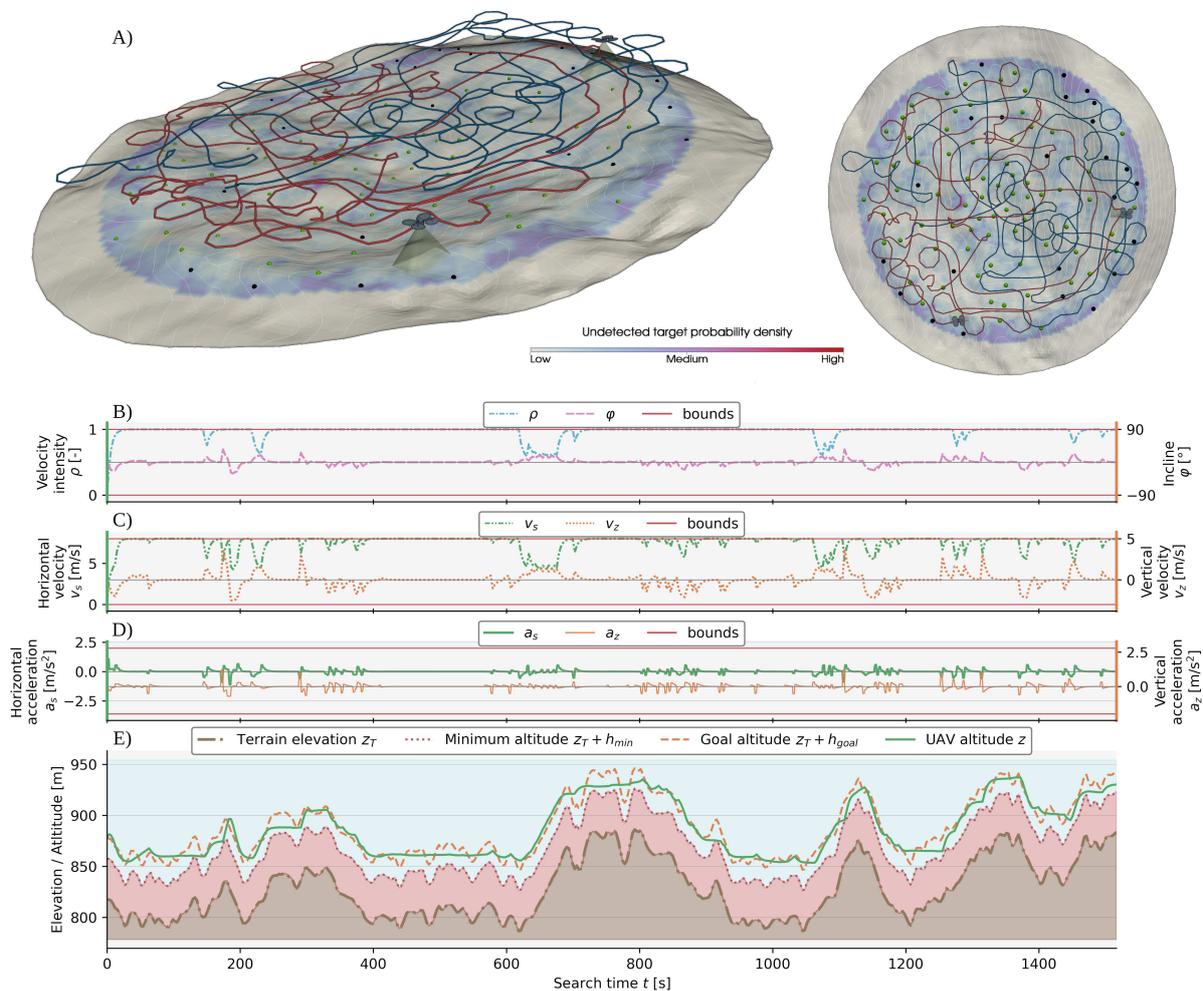


Figure 9.13: Mission 1 trajectories and marked targets (A) and flight parameters graphs (B-E) for the red trajectory executed by the Matrice UAV.

Mission 2 is presented in Figure 9.14. The search was conducted for 19.36 minutes, and the search trajectories at the end of the mission are shown in panel (A), along with the search domain containing the targets, which are colored according to their detection status, with green indicating detected targets and black indicating undetected targets. Some undetected targets

were never observed by the camera, as the undetected target probability density in their regions remained the same as at the start of the search. This is visible in the small region in the upper part of the domain in the top-down view shown in panel (A). Other undetected targets, located within the observed area, were missed due to limitations of the detection model, which failed to identify them. The search was performed with one Matrice UAV and one Mavic UAV. The yellow trajectory in panel (A) and the corresponding flight parameter graphs (B–E) correspond to the Mavic UAV. Similar to the Matrice trajectory analyzed in the previous mission (Figure 9.13), the Mavic UAV respected all given constraints (B–E) and maintained good adherence to the target flight height (E).

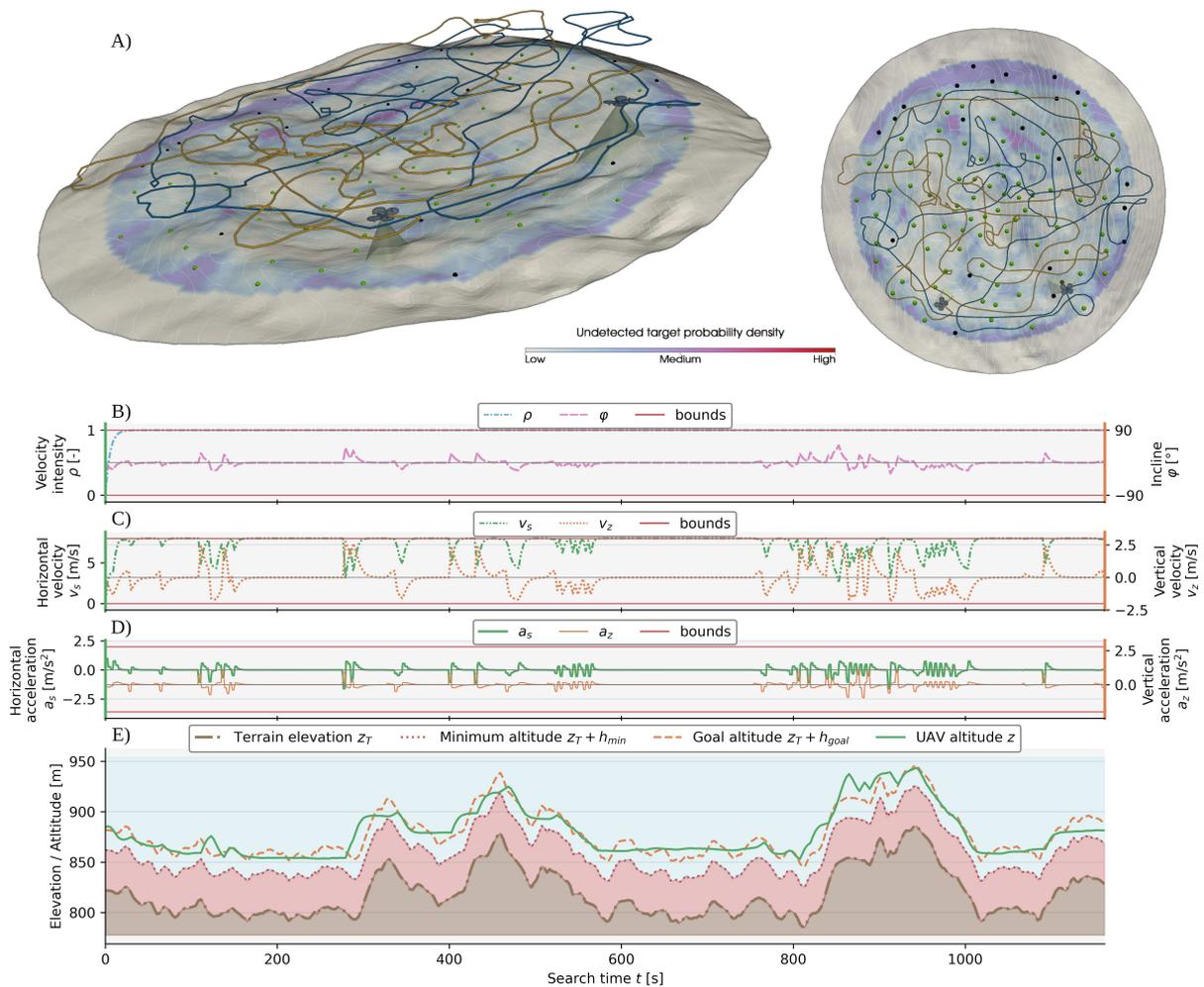


Figure 9.14: Mission 2 trajectories (A) and flight parameters graphs (B-E) for the yellow trajectory executed by the Mavic UAV.

Finally, to validate the search methodology, the survey accomplishment metric η is compared to the achieved target detection rate κ . The target detection rate is defined as the ratio of detected targets to the total number of targets in the experiment. During the search mission,

multiple images may capture the same target, and the detection model may detect the same target multiple times. However, only the first detection of each target is counted toward the total number of detected targets, so repeated detections of the same target do not affect κ . Figure 9.15 shows $\eta(t)$, κ , and their relative error for both search missions. The η parameter is presented as a continuous curve, while κ is plotted discretely as dots. Each dot corresponds to the time when κ increases, caused by the detection of a new target. The mean absolute relative error over the entire search duration is 0.094 for Mission 1 and 0.073 for Mission 2. Under realistic field conditions, the achieved relative error is considered acceptable, which suggests that η can serve as a reliable indicator of search performance.

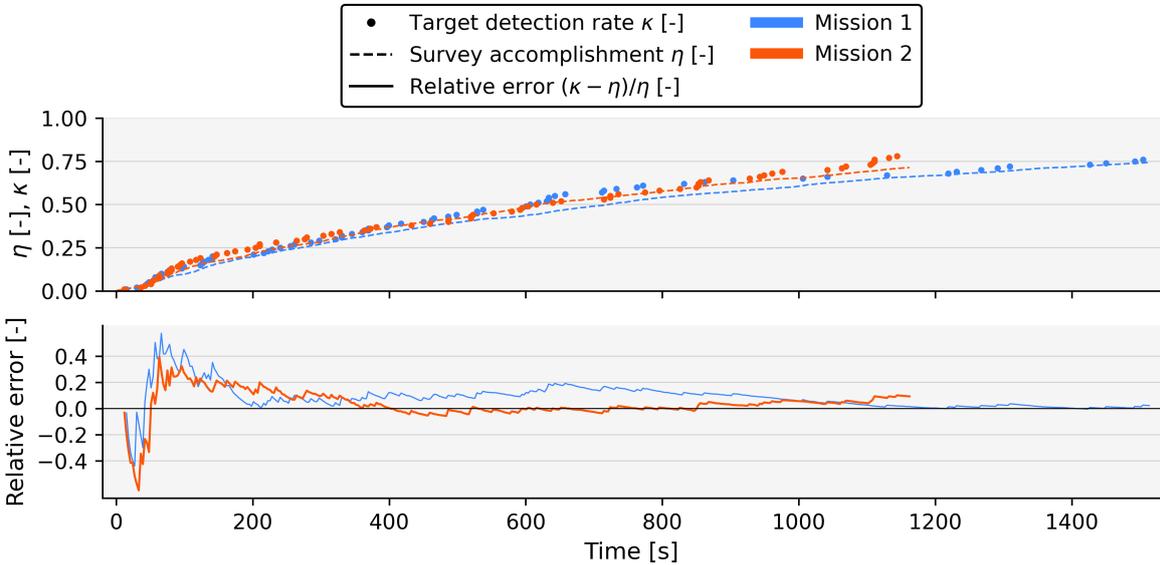


Figure 9.15: Survey accomplishment η , target detection rate κ , and their relative error over the duration of both search missions.

10 SEA SURFACE EXPLORATION

This chapter presents the modifications and enhancements on the static target search methodology in order to adapt the method for the search of a dynamic sea surface environment. First, the problem is defined, detailing the differences between terrain and sea surface exploration. Then, the behavior of drifting sea targets is discussed, and a dynamic probability model using the advection-diffusion equation is introduced, along with uncertainty compensation through diffusion. Finally, the implementation of the dynamic probability model using OpenFOAM is detailed.

10.1 Problem formulation – Search in maritime environments

With the search focus shifting from land to sea, certain aspects of the methodology can be simplified, while others present new challenges that require further development. Since the sea surface is flat, there is no need to account for terrain obstacles, and the search can be executed at a constant flight altitude. Therefore, the MPC optimization detailed in Chapter 7 can be omitted, as the UAV flight is effectively conducted in the Ω_{2D} domain, which is offset from the sea surface to the goal flight height h_{goal} . The velocity intensity and incline parameters are set to constant values of $\rho = 1$ and $\varphi = 0$, respectively. This implies that the UAVs fly at their maximum velocity intensity, maintaining a constant velocity of $v_{s,max}$, and the vertical velocity $v_z = 0$. Since the UAVs maintain a constant velocity, the acceleration constraints are automatically satisfied. A change in velocity occurs only if the UAVs begin the search from a non-moving state, in which case the acceleration constraint is not considered. In practice, this introduces an initial deviation between planned and actual positions. However, since the method tolerates such errors and corrects them over time, as shown in sections 8.6 and 9.1, this simplification does not compromise performance.

Given the flatness of the sea surface, the sensing model (detailed in Section 4.3) can be slightly simplified by omitting the ray-tracing observation check. In the absence of terrain elevation or occluding structures, any point within the field of view is guaranteed to be observable.

The main challenge this search problem presents is dynamic nature of the search targets, whose positions evolve due to environmental factors. Targets at the sea surface drift due to the movement of the surface water layer. This movement is caused by a combination of several natural effects such as sea currents, tides, wind, and waves. In order to determine the way that the targets move, the sea surface flow field must be obtained. It can be estimated through various approaches, such as numerical ocean simulation models, satellite data, high-frequency radar measurements, or surface drifters equipped with GPS sensors. The appropriate method is chosen based on the needed spatial resolution, geographic coverage, and real-time data availability. For the purpose of defining the dynamic target search methodology, the surface flow field, represented with the vector field $\mathbf{w}(\mathbf{x}, t)$, is considered known.

10.2 Dynamic target probability distribution

The movement of surface targets can be simulated by the effects of advection in the velocity field \mathbf{w} and diffusion. Advection represents the transport of a substance or target due to the fluid flow, while diffusion models the spreading of a substance on a molecular level, driven by differences in concentration. In many real-world problems, both processes occur together. For example, a pollutant in a river is advected downstream by the current while it is simultaneously being diffused sideways and vertically. A similar principle applies to a group of people in the sea: they are carried by the currents through advection, while their individual swimming movements and other stochastic impacts can be approximated as uncertainty using a diffusive process.

Advection and diffusion can be modeled using either the Lagrangian and Eulerian approach. The Lagrangian approach focuses on tracking individual particles and observes how they move and evolve over time. Advection is in this case captured naturally by the moving particles effected by the flow field, governed by the Lagrangian motion law

$$\frac{d\mathbf{z}}{dt} = \mathbf{w}(\mathbf{z}, t), \quad (10.1)$$

where the particle location is represented with $\mathbf{z}(t) \in \Omega_{2D}$. Diffusion is modeled as a stochastic process, such as Brownian motion. Brownian motion is represented by a stochastic differential equation

$$d\mathbf{z}(t) = \sigma d\xi(t), \quad (10.2)$$

where σ represents the standard deviation of the Brownian motion and ξ denotes a stochastic Wiener process. Both advection and diffusion can be modeled together by combining (10.1) and (10.2) resulting in

$$d\mathbf{z}(t) = \mathbf{w}(\mathbf{z}, t) dt + \sigma d\xi(t). \quad (10.3)$$

Numerical implementation of the method is executed by simulating a large number of particle trajectories. While this approach is well suited for tracking individual particles, it can struggle to compute smooth field approximations as it requires many particles and can become computationally expensive, especially for large-scale problems.

In contrast, the Eulerian approach monitors how the field variables, for example pollution concentration, change at fixed points within the domain. In this case, the observed variable is the probability of undetected target presence m . The advection, using the example of $m(\mathbf{x}, t)$, is modeled with the term $\mathbf{w} \cdot \nabla m$ in the PDE, while diffusion is represented $D \cdot \nabla^2 m$, where D is the diffusion coefficient. The resulting advection-diffusion PDE is given by

$$\frac{\partial m}{\partial t} = D \cdot \nabla^2 m - \mathbf{w} \cdot \nabla m. \quad (10.4)$$

Since the positions of the targets are uncertain in a search scenario, they need to be represented by a probability density field. This also enables certain regions to be probabilistically prioritized based on the constructed field. Due to the need to model both advection and diffusion of the field, the Eulerian approach was chosen. While the Lagrangian approach can also generate a probability field, by discretizing the domain into a grid and counting the number of particles in each cell, it requires an additional processing step. Furthermore, incorporating diffusion in the Lagrangian framework demands a significantly larger number of particles to provide a statistically representative probability field. The Eulerian model directly computes the field and diffusion can be included with practically no additional computational cost.

Diffusion is used to model the cumulative uncertainty of the system, which may arise from the inaccuracies in the advection flow field, initial probability distribution of undetected targets, UAV localization errors causing sensing inaccuracy, or other sources of uncertainty in the drift of targets. The diffusion coefficient D is estimated using the mean square displacement formula for two-dimensional Brownian motion, given by

$$E^2(t) = 4 \cdot D \cdot t,$$

where the variable E denotes the average displacement of a particle over time t . To align the model with real-world uncertainty, the diffusion coefficient is computed using the estimated positional drift error E_e of targets over time t . That estimate can either be measured directly or computed based on the error in the measured flow field, if that information is available. For the two-dimensional flow, the diffusion coefficient is defined as

$$D = \frac{E_e^2}{4 \cdot t}. \quad (10.5)$$

To conduct an efficient search of the dynamic sea surface area, the drift model for m must be integrated into the sensing process to account for the influence of sea dynamics on the targets, and consequently on m . To achieve this, the sensing term, given by equation (4.6), is incorporated into the advection-diffusion PDE (10.4). Considering the collective sensing effect of all search agents, based on the sensing function defined in equation (4.10), the resulting PDE describing the dynamic behavior of m can be formulated as

$$\frac{\partial m}{\partial t} = D \cdot \nabla^2 m - \mathbf{w} \cdot \nabla m - m \cdot \left(1 - \prod_{i=1}^n (1 - \psi_i(\mathbf{R}_i)) \right). \quad (10.6)$$

In practical implementation, given realistic sea surface layer flow velocities, the advection-diffusion effects can be considered negligible over the duration of a single sensing interval Δt_s .

Since the behavior of m is dynamic, it must be modeled starting from the time the initial probability distribution of the targets is established, at $t = 0$. If the search is delayed, the probability field needs to be evolved up to the search start time and then continuously modeled during the search. The undetected target probability density at $t = 0$, denoted as m_0 , needs to satisfy the condition (4.5).

The exploration of evolving m is managed by the ergodic search control defined in Chapter 5, and the resulting search performance is evaluated using equation (4.15). The search agents are controlled using constant values of ρ and φ , while the ω values are computed using the selected ergodic search method (Section 5.1 or Section 5.3), and later adjusted through the collision avoidance procedure detailed in Chapter 6.

10.3 Numerical implementation

The dynamic behavior of m , described by equation (10.6), is implemented by integrating OpenFOAM with the proposed sensing and ergodic search framework. OpenFOAM is an open-source software suite for Computational Fluid Dynamics (CFD), capable of modeling various physical processes such as heat transfer, fluid flow, and scalar transport.

The evolution of the m field at each time step Δt , begins with the application of the UAVs' sensing effects, followed by advection and diffusion. The advection-diffusion effects are modeled using the OpenFOAM solver *scalarTransportFoam*, which solves the equation

$$\frac{\partial m}{\partial t} + \nabla \cdot (\mathbf{w}m) - \nabla \cdot (D\nabla m) = S_m, \quad (10.7)$$

where S_m is the source of the scalar m . Since the sink/source representing sensing is considered with the sensing function applied externally, the *scalarTransportFoam* sink term $S_m = 0$. Additionally, considering that the flow field \mathbf{w} is incompressible (implicating that $\nabla \cdot \mathbf{w} = 0$), and that the diffusion coefficient D is a constant scalar, the equation (10.7) can be rewritten as

$$\frac{\partial m}{\partial t} = D \cdot \nabla^2 m - \mathbf{w} \cdot \nabla m,$$

which relates to the equation (10.6) without the sensing term

$$m \cdot \left(1 - \prod_{i=1}^n (1 - \psi_i(\mathbf{R}_i)) \right). \quad (10.8)$$

In contrast to the ergodic control framework which employs FEM, OpenFOAM utilizes the Finite Volume Method (FVM). In FVM, numerical discretization is performed by dividing the physical domain into non-intersecting control volumes, also known as cells. Although the proposed problem is two-dimensional (in Ω_{2D}), a three-dimensional mesh with only a single cell in the third dimension is required. To lay the groundwork for implementing the governing equation (10.7) in FVM, it is rewritten in integral form for each cell volume as

$$\int_V \frac{\partial m}{\partial t} dV + \int_{A_V} m \mathbf{w} \cdot \mathbf{n}_V dA_V = \int_{A_V} D \nabla m \cdot \mathbf{n}_V dA_V + \int_V S_m dV,$$

where V is the control volume, A_V is its boundary surface, and \mathbf{n}_V is the outward normal vector on A_V . To solve the equation using FVM, each term must be discretized. This involves

approximating the integrals over control volumes and their boundaries. The time derivative is typically approximated using finite difference schemes such as the Euler scheme. The fluxes through each face of the control volume, due to advection, can be approximated using various interpolation schemes, including upwind interpolation, linear interpolation, quadratic upwind interpolation, and other higher-order methods [112]. Gradients responsible for diffusion, are commonly computed using the central difference theorem or the Gauss's divergence theorem. The source is treated as constant within each cell, which requires the cells to be reasonably smaller than the scope of the sensing function.

Using the discretized forms, the explicit Euler scheme can be applied to advance the solution in time. In this approach, the value of m in each control volume is updated based on the fluxes and sources evaluated at the previous time step. The surface integrals are replaced by a sum of fluxes across all faces of the control volume, where the advective flux for each face f is given by $m_f(\mathbf{w} \cdot \mathbf{n})_f A_f$ and the diffusive flux by $-D_f A_f (\nabla m)_f \cdot \mathbf{n}_f$. Combining these contributions yields the explicit finite volume update equation for each cell P , expressed as

$$m_P^{j^*} = m_P^{j^*-1} - \frac{\Delta t}{|V_P|} \sum_f ((\mathbf{w} \cdot \mathbf{n})_f A_f m_f - D_f A_f (\nabla m)_f \cdot \mathbf{n}_f) + \Delta t S_{m,P},$$

where j^* denotes the advection-diffusion time step, A_f indicates the area of a face f , and \mathbf{n}_f represents its unit normal vector. The summation runs over all faces of the control volume P .

The main advantage of the FVM is that it naturally satisfies the conservation laws governing fluid flow, including mass, momentum, and energy, at both local and global levels [112]. Another advantage is that it can be extended to multi-physics problems, such as solving the fluid dynamics Navier-Stokes equations and the advection-diffusion equation simultaneously to model both fluid flow and scalar transport.

Since the coupled methods are discretized differently, they use different numerical meshes. The FEM method within the HEDAC control framework employs a two-dimensional triangular mesh, whereas the FVM utilizes a three-dimensional hexahedral mesh with a single cell layer in the third dimension. Additionally, FEM stores the data at mesh nodes, while FVM stores the data at cell centers. To avoid the introduction of additional numerical error, the undetected target probability density field m is handled entirely within the FVM mesh. The UAV control framework reads the OpenFOAM scalar field m and directly applies sensing using the term (10.8),

via equation

$$m^j = m^{j^*} \cdot \prod_{i=1}^n (1 - \psi_i(\mathbf{R}_i)),$$

where j^* denotes the advection-diffusion time step before sensing is applied, and j represents the same time step after sensing is applied. At time steps when sensing is not executed, $\psi(\mathbf{R}) = 0$, and therefore $m^j = m^{j^*}$. The m field is then projected to the FEM mesh to compute the potential u , utilizing the nearest neighbor interpolation method. This method is chosen for its high computational efficiency, and since u is recalculated at each time step from the current values of m , error does not accumulate through the simulation time. The detailed procedure for the HEDAC FEM method with dynamic probability field incorporated using OpenFOAM is provided in Algorithm 1.

Algorithm 1 Procedure for ergodic sea exploration using HEDAC

procedure UAV MOTION CONTROL WITH DYNAMIC PROBABILITY DENSITY FIELD

function INITIALIZATION

Initialize general parameters, UAVs, and FEM system

Normalize and set the m field in the finite element system

▷ Equation (4.5)

Save normalized m field to OpenFOAM case for $t = 0$

Initialize OpenFOAM vector field \mathbf{w}

Initialize diffusion coefficient field in OpenFOAM

Set initial time: $t \leftarrow 0$

end function

function PRE-SEARCH ADVECTION-DIFFUSION

Perform scalar transport of m until search start time

▷ Equation (10.4)

$t \leftarrow$ search start time

end function

while search not complete **do**

function COMPUTE TRAJECTORIES

Retrieve scalar field $m(t)$ from OpenFOAM case

for $i = 1$ to n **do**

▷ For all agents

Apply sensing: $m(t) \leftarrow m(t) - \psi_i(\mathbf{R}_i) \cdot m(t)$

end for

Save updated m field with applied sensing to OpenFOAM case

Update m field in the finite element system

Compute potential u

▷ Equation (5.1)

Compute UAVs' yaw angular velocities

▷ Equation (5.4)

Execute collision avoidance procedure

▷ See Chapter 6

Update UAVs' positions

Perform scalar transport of m in OpenFOAM for time step Δt

▷ Equation (10.4)

Advance time: $t \leftarrow t + \Delta t$

end function

end while

end procedure

A similar procedure is used for the mSMC method, with the primary differences being that they do not use FEM or a triangular mesh. Instead, the m field is represented on a structured

rectangular grid. Furthermore, field m is used to compute the UAVs' yaw angular velocities by employing Fourier basis functions, as detailed in Section 5.3.

Additionally, in the method's implementation, UAV motion control is executed with a period of Δt , while the OpenFOAM scalar transport advection-diffusion simulation for the duration of Δt is performed with the integration time step of $\Delta t/10$. Numerical simulations of the advection-diffusion process are performed using *OpenFOAM v2406* [113].

11 SEA SURFACE SEARCH SIMULATIONS

This chapter presents the results of sea search simulations conducted using the proposed control methodology for dynamic target search. The first test is performed on a synthetically generated domain, and the results are compared to a baseline simulation employing the static probability model. This case is also used to examine the effect of different relative velocities between the UAVs and the flow field on search performance, as well as the robustness for search parameter selection. The second case represents a realistic search scenario in a coastal sea region and demonstrates uncertainty compensation using diffusion. The third case applies the method to a large-scale ocean search operation, evaluating its performance under long search delays and complex transient flow conditions.

11.1 Modeling motion and detection of dynamic targets

To evaluate how well the probability distribution reflects search success in a dynamic scenario, simulated search targets are introduced into all test cases. The sensor simultaneously applies its sensing effect to the domain and performs target detection using the same detection rate function. In each test case, 1000 simulated targets are introduced at $t = 0$, and distributed throughout the search domain according to the initial target probability density m_0 . They are advected with the flow field \mathbf{w} according to the Lagrangian motion law given by the equation (10.1), or (10.3) when considering uncertainties.

Each simulated search scenario is evaluated using two metrics: the survey accomplishment metric η , representing the system's estimate of survey completeness, and the target detection rate κ , representing the ratio of detected targets to the total number of targets, providing an accurate assessment of survey completion.

11.2 Synthetic case – Cavity flow

The first test case consists of a synthetically generated scaled domain featuring a cavity lid-driven flow, a common benchmark problem in fluid dynamics. It is very convenient because the geometry and boundary conditions are simple, but the resulting flow is relatively complex. The fact that the probability cannot escape the domain makes it ideal for testing the proposed search methodology with dynamic probability, even over longer durations. The domain is defined by a 1×1 m square centered at $(0.5, 0.5)$ m, containing an internal rectangular obstacle described with two opposite corners located at $(0.7, 0.2)$ and $(0.8, 0.6)$. The flow field is generated using the *simpleFoam*, a steady-state solver for incompressible, turbulent flow included in OpenFOAM. The velocity boundary condition on all surfaces is set to the *no slip* condition, implying that the fluid in contact with the boundary has zero velocity with respect to it. The flow is driven by moving the upper domain boundary at $2 \cdot 10^{-2}$ m/s, with the kinematic viscosity of the fluid set to $1 \cdot 10^{-6}$. The pressure boundary condition is set to *zero gradient* on all the boundaries, and the turbulence is modeled with $k - \omega$ Shear Stress Transport (SST) model. The resulting flow field is shown in Figure 11.1, where the black arrows represent the flow vector field \mathbf{w} and the contour plot represents the velocity magnitude. The average velocity magnitude of the computed flow is $3 \cdot 10^{-4}$ m/s, and the flow remains steady during the search simulation. The simulated targets are distributed across five differently shaped areas within the domain, according to m_0 . In the two upper areas, they are scattered following a normal distribution centered within each respective area, while in the remaining areas the targets are distributed uniformly. The target distribution is shown in Figure 11.1, where the targets are represented as red dots.

The search is conducted for $T = 900$ s using three identical search agents moving at a constant velocity of $v_s = 0.015$ m/s, with a minimum turning radius of $R_{min} = 0.01$ m, and a minimum clearance distance of $\delta = 0.01$. The control and sensing interval is defined as $\Delta t = 0.2$ s, and the HEDAC parameters are set to $\alpha = 5 \cdot 10^{-2}$ and $\beta = 1$. Each agent is equipped with a conical sensor defined by $\gamma_c = 90^\circ$, and the sensing is executed from a constant height of $h_{goal} = 0.015$ m. The detection probability in one flyover directly below the search agent is equal to $\mu_\alpha = 0.65$, and it diminishes laterally relative to the agent's trajectory, following a normal distribution with a standard deviation of 0.015. The resulting detection rate function is

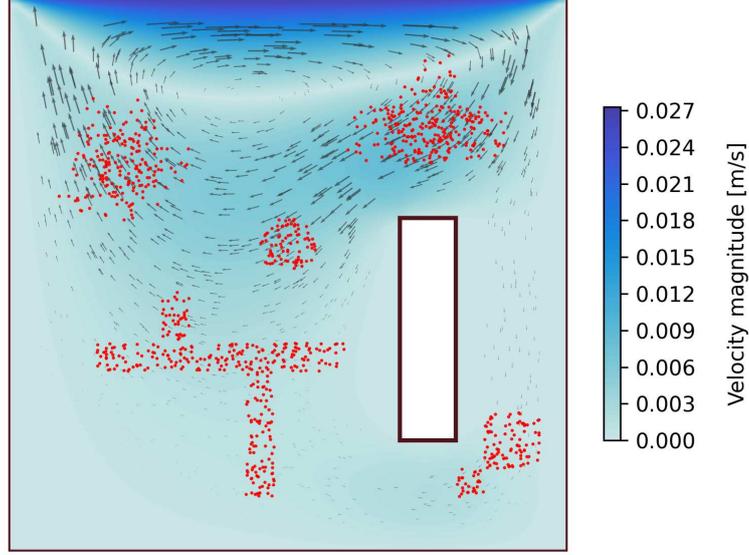


Figure 11.1: Cavity flow field during the simulation and the distribution of simulated targets at $t = 0$.

given by

$$\Gamma = \frac{\ln(1 - \mu_a)}{t_{scene}} \cdot e^{-0.5 \left(\frac{\|\mathbf{R}\|^2 - h_{goal}^2}{0.015^2} \right)}$$

where $t_{scene} = 2$ s according to equation (4.11), as constant velocity implies $v_{s,avg} = v_s$.

The search is conducted using the proposed method, which employs a dynamic probability distribution model, where the probability evolves according to the combined effects of the flow field \mathbf{w} and sensing. The results are compared with a baseline method that utilizes a stationary probability distribution, in which the probability is influenced solely by the effects of sensing. Essentially, the baseline method corresponds to the proposed method with $\mathbf{w} = 0$. Figure 11.2 shows the UAVs' trajectories and the undetected target probability distribution at $t = 450$ s, along with the evolution of η and κ throughout the search for both the proposed and baseline methods. The undetected targets are shown as red dots, while detected targets are represented in green. In contrast to the baseline method, the spatial distribution of undetected targets in the proposed method is in line with the undetected target probability density. This consistency is further confirmed by the relatively good agreement between the η and κ metrics throughout the search. Compared to the baseline, the proposed method achieves approximately 50% better performance in terms of κ . Notably, although the baseline method reports a higher η , it substantially overestimates actual performance since it is unaware of the probability density advection. In contrast, the proposed method not only achieves superior detection performance

but also provides a more reliable estimate of survey completion, as reflected by the alignment of η with κ .

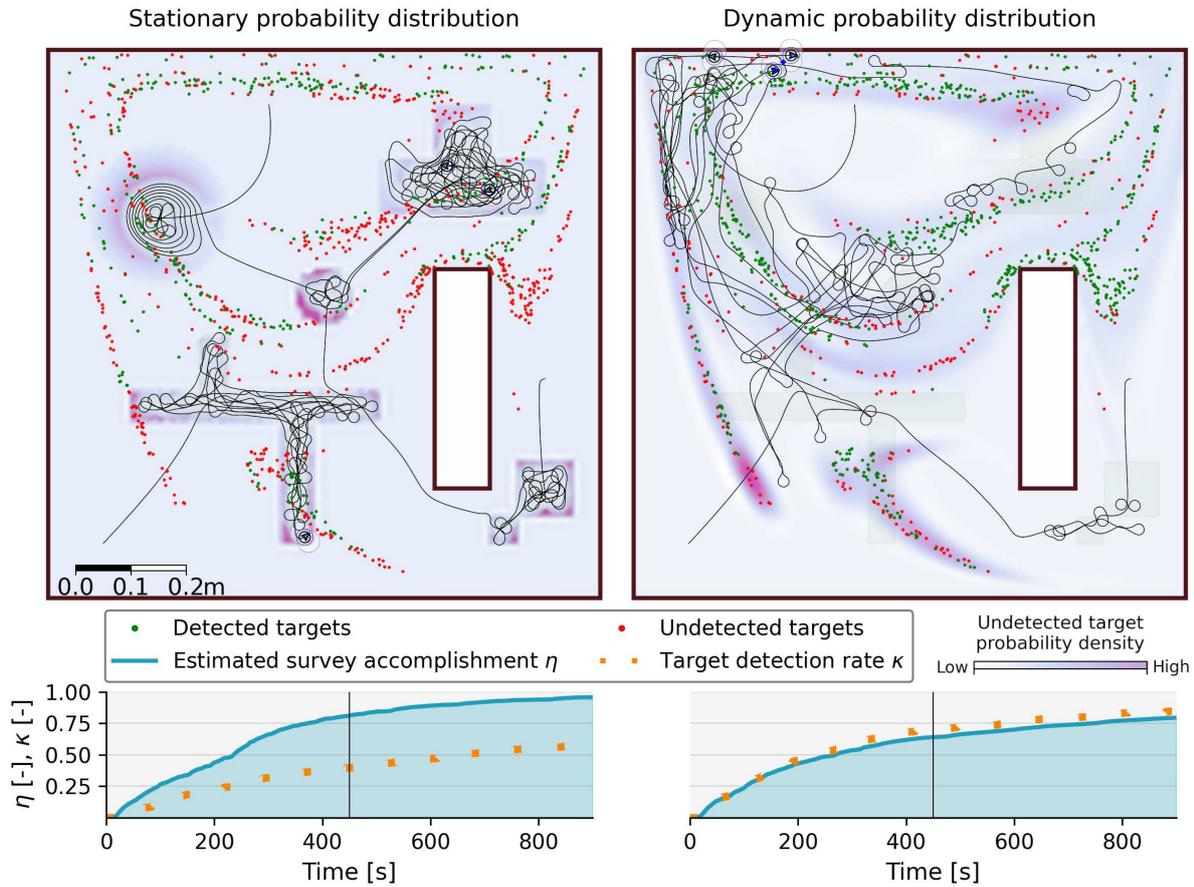


Figure 11.2: Undetected target probability field, simulated targets, and UAV trajectories for both the proposed and baseline methods at $t = 450$, together with η and κ values throughout the search.

To evaluate the influence of search agent speed and flow field velocity on search performance, a new parameter, λ , is introduced. This parameter represents the ratio between the search agents' velocities and the average velocity magnitude of the flow field \mathbf{w} . Large λ values indicate that search agents move considerably faster than the flow field, while small values correspond to cases where the flow field velocity dominates. Specifically, $\lambda = 1$ represents the condition in which the average flow field velocity equals the UAV velocity. Different λ values are obtained by scaling the flow field from the original case, while keeping the velocities of the search agents constant. For reference, λ value in the order of 50 would be representative for a realistic search scenario performed in a coastal sea region. This estimate assumes multi-rotor UAVs operating at 10 m/s and the average velocity of a submesoscale flow around 0.2 m/s [114, 115].

The performance of both the proposed and baseline methods is assessed and compared across λ values in the range $[0.25, 1000]$. Figure 11.3 displays the results at the midpoint of the search ($0.5T$) and at the end of the search (T). At very high λ values, no significant difference is observed between the performance of the baseline and proposed methods, as the influence of the flow field becomes negligible. In this regime, both methods perform similarly well, and η closely corresponds to κ . For $\lambda < 1$, the flow moves faster relative to the search agents, making it effectively impossible to conduct the search in certain cases, depending on the flow characteristics. Detections are largely caused by the circulation effect as targets are transported beneath the search agents and detected. The resulting κ is similar for both methods but the proposed method provides an advantage by accurately estimating κ through the η metric, while the baseline method significantly overestimates it. The primary results of interest correspond to λ values around 50, indicated by a black vertical dashed line in the graphs, which represent realistic operational conditions. The proposed method achieves considerably higher κ , and unlike the baseline method, κ aligns with η , indicating an accurate performance estimate.

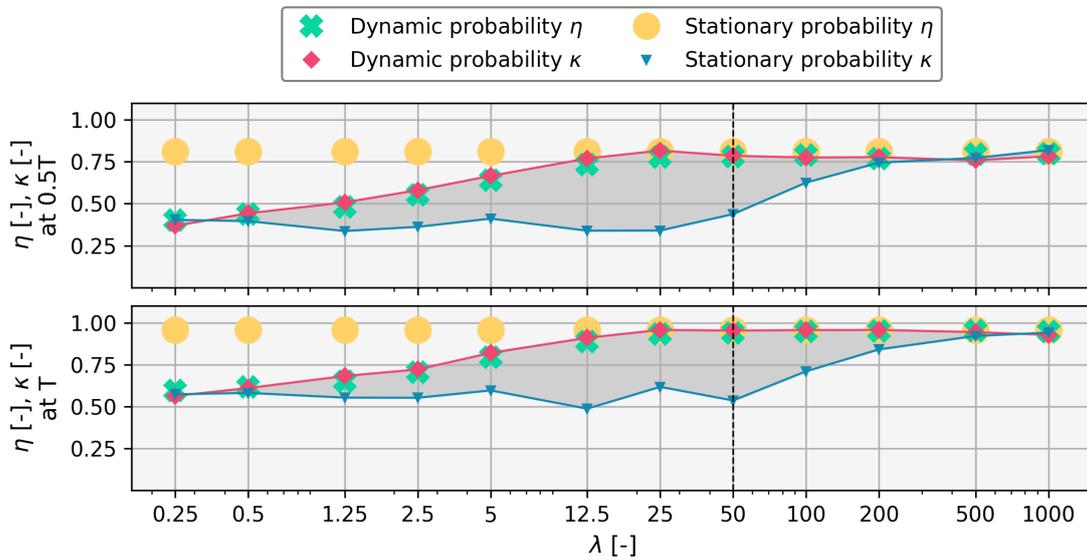


Figure 11.3: Performance comparison of the proposed and baseline methods across different λ values.

Additional performance analysis of the proposed method was conducted by varying the HEDAC parameter α and the number of agents performing the search while keeping all other parameters of the original case unchanged. The results are shown at the midpoint of the search ($0.5T$) and at the end of the search (T) in Figure 11.4. The results demonstrate that the method is

generally robust with respect to the selection of α and achieves slightly better performance with a lower α than the originally selected value. As expected, search performance improves with the addition of more search agents. However, beyond a certain point, adding additional agents yields only marginal gains. Furthermore, at the end of the search (T), across all combinations of α and agent count, the mean discrepancy between κ and η is approximately 3%. This difference can be considered as a statistical error due to the stochastic nature of the initial distribution of targets and detection process.

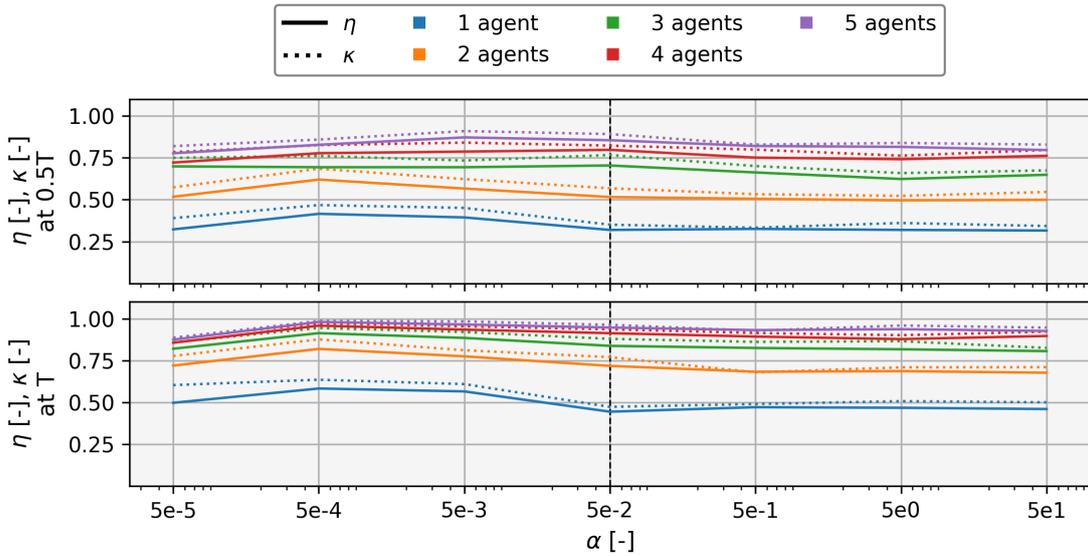


Figure 11.4: Performance analysis of the proposed method across different α values and varying numbers of UAVs conducting the search.

11.3 Realistic search scenario – Unije Channel search

The second test case simulates a survey taking place in the Unije Channel, located between the islands of Unije and Lošinj, Croatia ($44^\circ 37' N, 14^\circ 19' E$). The search domain covers a total of 95.9 km^2 . To reflect realistic operational conditions, the start of the search is delayed by 3 hours to account for the travel time of the response team. The mission is carried out using five identical multi-rotor UAVs over six consecutive search waves. Each wave has a duration of 25 minutes, corresponding to the approximate duration of one battery charge, followed by a 5-minute pause to simulate battery replacement.

The UAVs operate at a constant velocity of $v_s = 10 \text{ m/s}$, with a minimum turning radius of $R_{min} = 100 \text{ m}$ and a minimum clearance distance of $\delta = 50 \text{ m}$. The survey is performed from

a constant height of $h_{goal} = 100$ m using a sensor with a pyramidal FOV defined by $\gamma_1 = 77.3^\circ$ and $\gamma_2 = 48.5^\circ$. Under this sensing configuration, each image covers a rectangular ground area of 160×90 m, corresponding to an aspect ratio of 16:9. Since the search is performed at a constant height over the flat sea surface, the recall (representing detection probability) is defined as a constant value of $\mu_b = 0.75$. The corresponding detection rate function is obtained using equation (4.13), resulting in

$$\Gamma = -\frac{\ln(1 - \mu_b)}{t_{scene}},$$

where $t_{scene} = 9$ s, as defined in equation (4.11), under the assumption of constant velocity, implying $v_{s,avg} = v_s$. The sensing interval Δt_s is set equal to the control interval, $\Delta t = 3$ s, and the HEDAC parameters set to $\alpha = 10^5$ and $\beta = 1$.

To represent the realistic behavior of the sea surface flow, this case models a transient flow field $\mathbf{w}(\mathbf{x}, t)$. The flow is computed using *simpleFoam* OpenFOAM solver, and its transient behavior is introduced following the procedure described in [114]. The velocities of the flow field are within the range $[0, 0.4]$ m/s, which is consistent with the surface layer velocities measured in domains with similar characteristics [114, 115]. Snapshots of the flow field at $t = 0$, at the start of the search ($t = 10800$ s), and at the end of the search ($t = 21300$ s) are shown in Figure 11.5.

At $t = 0$, the undetected target probability density is uniformly distributed within a circular area of radius 2 km, with 1000 targets scattered uniformly inside the circle. To model realistic target drift, uncertainty is incorporated into the advection of the targets by introducing Brownian motion. By extrapolating the measured drift error of buoys on the sea surface reported in [114], the drift error after 3 h (the midpoint of the simulation) is estimated to be approximately 330 m. To reproduce this effect, the target motion is modeled using the equation (10.3), with the standard deviation of the two-dimensional Brownian motion $\sigma = 3.889$ m and an advection time step of 3 s. The positions of the simulated targets at $t = 0$, at the start of the search ($t = 10800$ s), and at the end of the search ($t = 21300$ s) are shown as red dots in Figure 11.5.

To compensate for target drift error, diffusion is introduced into the simulation. The diffusion coefficient, corresponding to a drift error of $E_e = 330$ m over $t = 3$ h, is calculated using equation (10.5), yielding $D = 2.521$. The search simulation is performed both with diffusion (to account for uncertainty) and without diffusion. The computed results for both cases are shown in Figure 11.6. The figure illustrates the search domain with UAV trajectories from the final search wave, target positions and their detection status, the flow field, and the undetected target

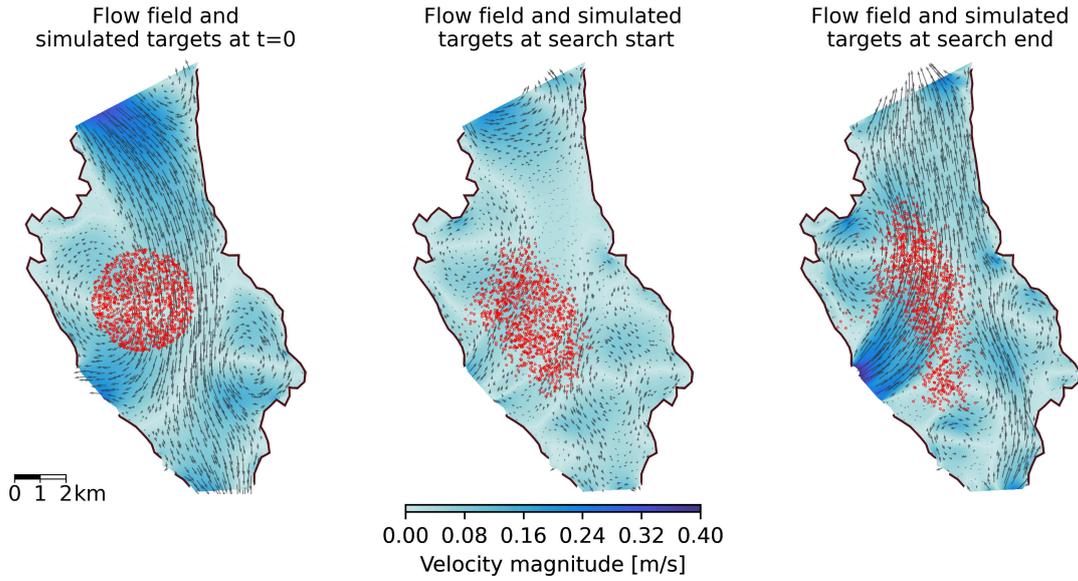


Figure 11.5: Flow field and target positions at $t = 0$, at search start $t = 10800$ s, and at search end $t = 21300$ s.

probability at $t = 21150$ s, near the end of the search. The lower plots present the evolution of η and κ for the compensated and the uncompensated cases over the search duration. Both cases perform similarly in terms of κ during the first search wave, but the performance of the uncompensated method declines over time. This decline is caused by the increased drift error over time and the lack of error compensation. The uncertainty compensated method achieved higher performance detecting 941 targets, while the uncompensated method detected 926 out of the total 1000 targets. An additional advantage of the compensated case is a more accurate survey accomplishment estimate. The compensated method achieved $\eta = 0.944$, closely corresponding to $\kappa = 0.941$, while the uncompensated method achieved $\eta = 0.985$ and $\kappa = 0.926$, overestimating survey performance by 6%. Correct performance estimation is crucial in search missions, as it ensures that the mission continues until a certain level of certainty is reached that the target is not within the search domain, which is represented by η . If the survey competed with $\eta = 0.98$, the use of the uncompensated method would result in premature termination of the search, potentially leaving the target undetected, whereas continuing the survey until the actual η value is reached could result in detection of the target.

The search domain was discretized using a FEM mesh with 18289 triangular elements and 9357 nodes, as well as a FVM mesh with 61340 hexahedral cells and 124628 points. The case was computed on a PC equipped with a 6-core 2.6 GHz CPU, 16 GB of RAM, and an SSD. The mean, median, and maximum time step computation times are reported in Table 11.1. As

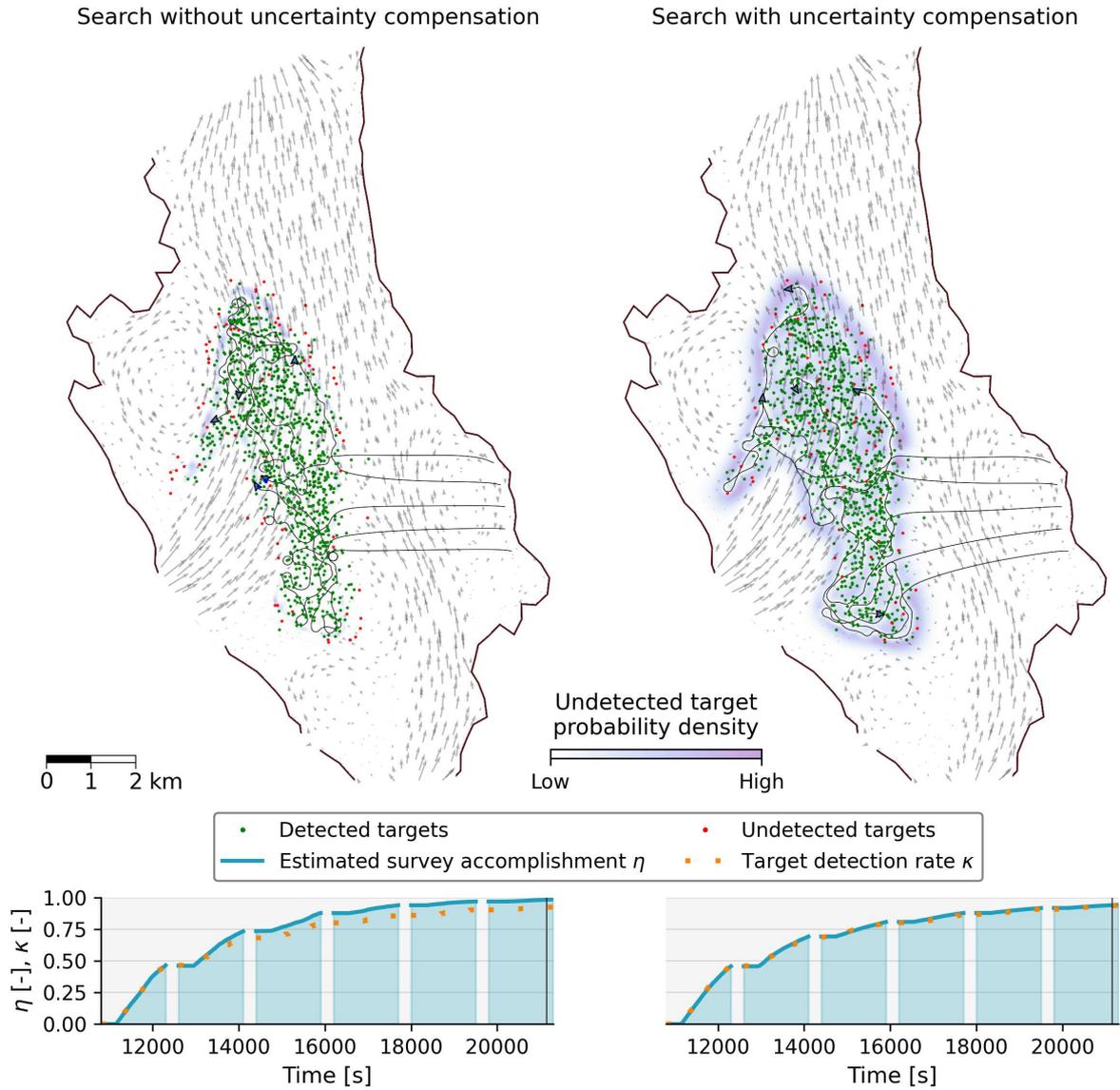


Figure 11.6: Unije Channel search at $t = 21300$, showing UAV trajectories from the final search wave, target locations and detection status for both the uncompensated and compensated cases, along with the evolution of η and κ over time.

evident from the table, the majority of the computation time is allocated to coverage convolution, which involves solving the OpenFOAM advection-diffusion simulation. Additionally, the maximum computation time does not exceed the control time step Δt , indicating the feasibility of real-time UAV control.

Table 11.1: Computation time for the Unije Channel search scenario [116].

Computation time	Max	Mean	Median	Units
Coverage convolution	2.0101	1.8141	1.8133	s
Potential field	0.0817	0.0399	0.0404	s
Collision avoidance procedure	0.8688	0.0151	0.0	s
Total	2.6805	1.8691	1.8573	s

11.4 Complex search scenario – MH370 search

To represent a complex scenario, a search simulation is conducted for the MH370 aircraft that disappeared in the Indian Ocean in March 2014, replicating the *splash area A* scenario from [74]. The complexity of this case arises from the sheer size of the domain, spanning 1000×900 km, combined with a long search delay of 20 days within a complex transient flow field.

The aircraft went missing on March 8, marking the starting point for the diffusion-advection of m_0 from the initial splash area. Within this area, 1000 targets are introduced and advected with the flow, while uncertainty is modeled analogously to the *Unije Channel search* scenario. The search began on March 28 and consisted of 3-hour missions conducted on each of the following five days. The search is performed with search agents flying at constant velocity of $v_s = 105$ m/s, employing $R_{min} = 100$ m and $\delta = 100$ m. They perform sensing from a constant height of $h_{goal} = 1500$ m, utilizing a conical FOV defined with $\gamma_c = 90^\circ$, which corresponds to a 1.5 km circular sensing radius as in the original case [74]. The probability of detection in a single flyover is set to $\mu_c = 0.75$, and the sensing function is given by

$$\Gamma = -\frac{\ln(1 - \mu_c)}{t_{scene}},$$

where $t_{scene} = 28.57$ s, calculated using $v_{s,avg} = v_s$ and equation (4.11). The control interval is set to $\Delta t = 10$ s, while sensing is performed more frequently, at intervals of $\Delta t_s = 2$ s. The HEDAC parameters are set to $\alpha = 10^8$ and $\beta = 1$.

The search simulation was performed using the HEDAC and mSMC methods, and the computed results are compared. Figure 11.7 shows the UAV trajectories, targets' positions and their detection status at the end of the first day search mission (9:00, March 28, 2014) computed with HEDAC, along with the underlying flow field and the evolution of κ and η throughout the search. The methods achieved comparable results at the search end, with mSMC yielding $\eta = 0.739$ and $\kappa = 0.693$, while HEDAC achieved slightly better performance with $\eta = 0.748$ and $\kappa = 0.718$. Both methods slightly overestimated the search performance, but the error is

considered tolerable given the long advection times and the complexity of the flow field.

This test case demonstrates that the method can handle prolonged search simulations in highly variable flow conditions, delivering fairly accurate performance metrics even under realistic large-scale operational conditions.

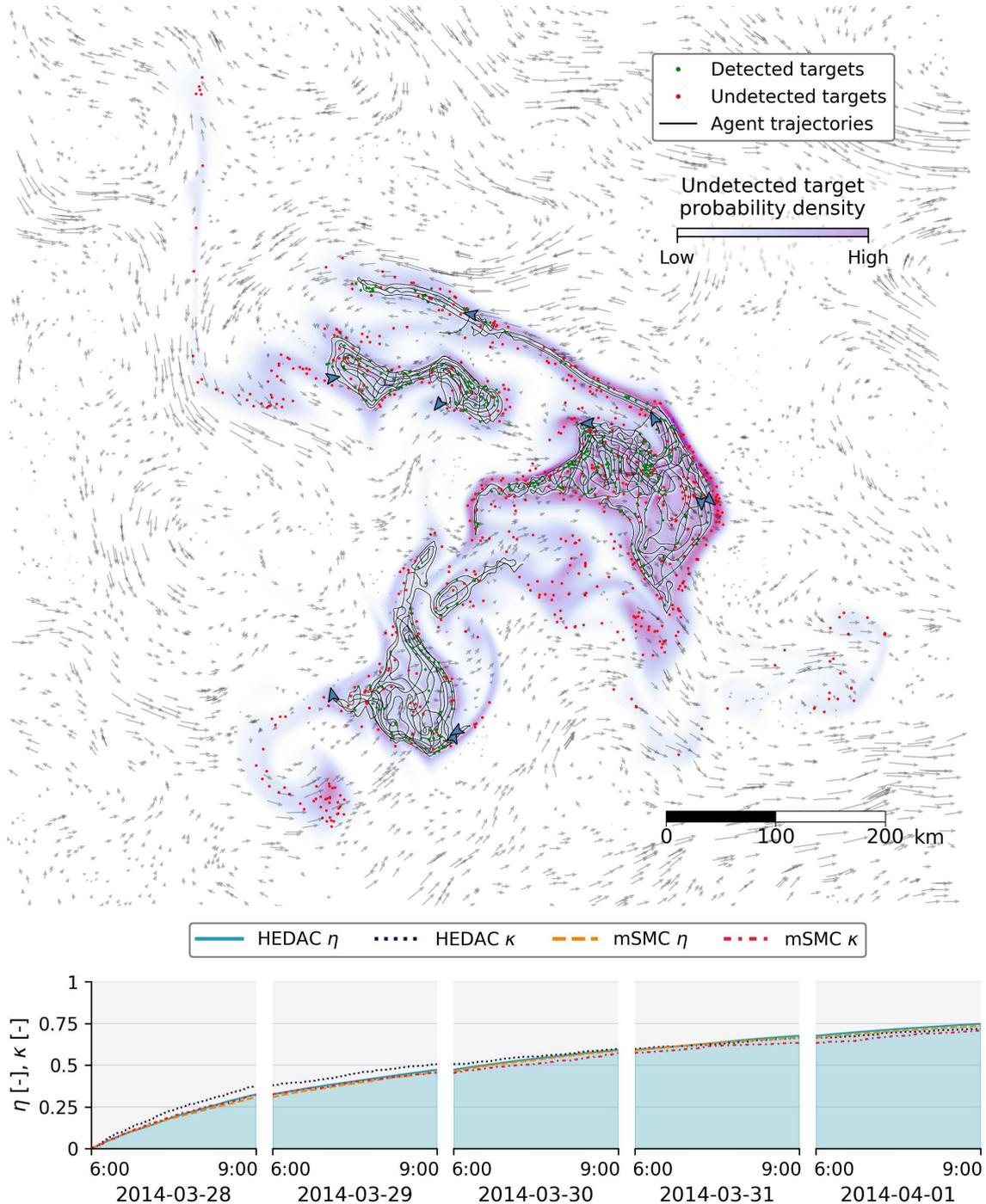


Figure 11.7: UAV trajectories, targets' positions and their detection status at the end of the first search mission in the MH370 simulation executed with HEDAC, along with the underlying flow field and the evolution of κ and η over all five search missions [116].

12 SEA SURFACE SEARCH EXPERIMENTS

This chapter presents the experimental validation of the sea surface exploration methodology. It begins by detailing the methodology for obtaining the sea surface flow field and the equipment used for this purpose. Following this, the custom search targets and the machine vision detection model used are described. The chapter concludes with a description of the experiment, including the search area, UAV setup, procedure, and results.

12.1 Surface flow reconstruction

In order to perform the search, the flow field \mathbf{w} representing the flow of the sea surface layer needs to be obtained. There are various ways of obtaining sea surface flow, including High-Frequency (HF) radars, satellite measurements, numerical ocean circulation models, and GPS-equipped floating buoys, commonly known as drifters. Each method offers specific benefits and has its constraints. For example HF radars provide a dense grid of surface flow measurements, but are expensive and their coverage is limited to specific coastal areas determined by their location. Satellite measurements provide wide area coverage, but the temporal resolution and real-time data availability is limited. Numerical models can simulate flows over large domains, but require numerous input parameters and relatively long computation time. Drifters provide reliable surface flow point measurements with relatively high frequency in real-time, but their spatial coverage is limited by deployment logistics and they face limitations in independently approximating the flow over large areas.

For this application, real-time data availability and area coverage are crucial. Therefore, drifters are chosen, since they provide measurements immediately after deployment and their coverage is flexible, as they can be deployed in any location. In this research, custom-made drifters were used, consisting of a floating buoy equipped with an Alltek Marine Electronics Corp TB-560 tracking beacon. They provide GPS location and velocity data at 10 s intervals through radio communication. Figure 12.1 shows a drifter deployed in the sea.

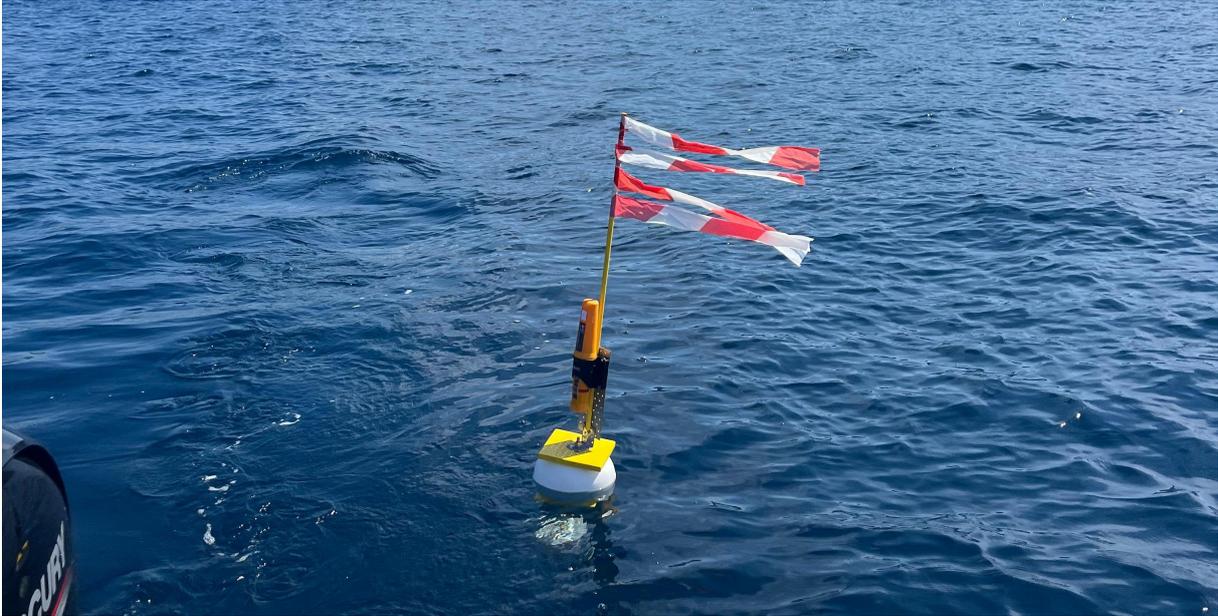


Figure 12.1: Deployed drifter used in the experiment for obtaining surface flow velocity measurements.

Since drifters provide velocity values at coarse scattered points, a continuous velocity field \mathbf{w} must be approximated over the entire domain. This is achieved by fitting a surrogate model that combines optimization and CFD to compute a simulated two-dimensional velocity field matching the measured values at corresponding points [114]. The procedure is described in this Section. The advantage of this approach, compared to the interpolation, is that it preserves the physics of a realistic flow, and the flow field optimization can be completed in a relatively short amount of time, since the simulated flow field is two-dimensional.

Modeling the sea flow is challenging because the flow can vary significantly across depth layers and tidal dynamics can lead to significant outflow or inflow of water within the domain. Therefore, an isolated two-dimensional simulation of the surface layer flow cannot realistically reproduce the actual state, since it is constrained by mass-conservation laws. To address this issue, the resulting surface layer flow field is obtained by combining the outputs of two separate two-dimensional flow simulations. The first flow simulation, referred to as bounded flow \mathbf{w}_b , includes the coastline with a no-slip boundary condition and the open sea with an inlet-outlet boundary condition. The second simulation, referred to as open flow \mathbf{w}_o , is defined over a circular domain enclosing the bounded flow domain, with inlet-outlet condition applied along the entire domain boundary. The resulting fused flow field, representing the surface layer flow,

is obtained by combining the bounded and open flow simulations

$$\mathbf{w}_f = \mathbf{w}_b + \mathbf{w}_o. \quad (12.1)$$

The boundary conditions for the bounded and open flow simulations are encoded in the optimization vector \mathbf{h} , which contains the pressure and tangential velocity values at control points defined along the inlet-outlet boundaries. The complete boundary profiles are then obtained by interpolating these values. The size of the optimization vector is twice the number of control points. In this case, the bounded flow contains three control points, while the open flow contains four control points.

Using the specified boundary conditions, the bounded and open flow simulations are computed with the *simpleFoam* solver implemented in OpenFOAM, employing the $k - \omega$ SST turbulence model. The fused flow \mathbf{w}_f is then obtained using equation (12.1). The bounded, open, and fused flow fields are shown in Figure 12.2, along with the defined boundary condition control points.

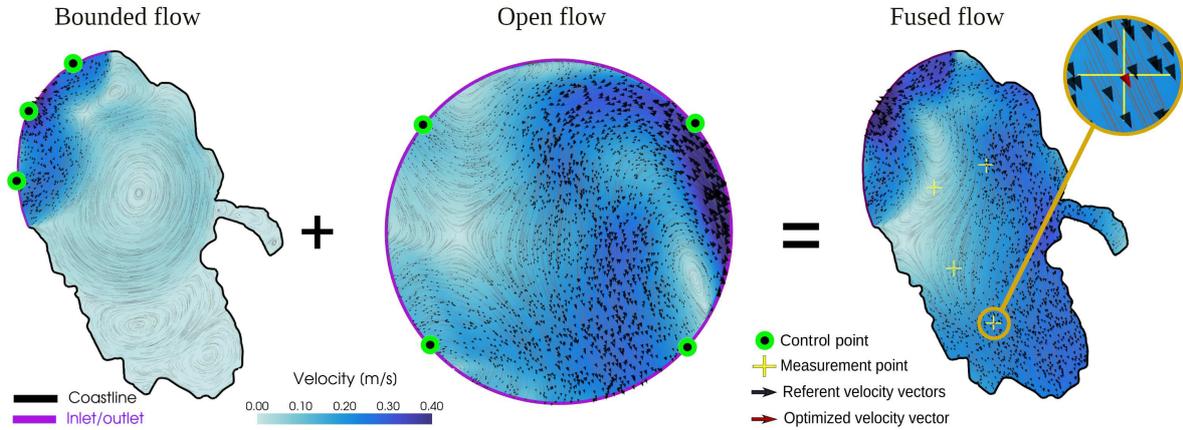


Figure 12.2: Visual representation of the bounded, open, and fused flow fields, along with the positions of the boundary control points and the drifter measurement locations. The figure also displays the measured reference velocity vectors and the optimized velocity vectors within the fused flow.

The optimization goal is to minimize the error between the point measurements obtained from the drifters and the corresponding values of the fused flow field. The error is defined as

$$\varepsilon_d(\mathbf{h}) = \frac{1}{n_d} \sum_{i=1}^{n_d} (\mathbf{w}_{r,i} - \mathbf{w}_{s,i}(\mathbf{h}))^2,$$

where n_d is the number of drifters, \mathbf{w}_r is the vector containing the reference velocity values obtained from the drifters, and \mathbf{w}_s is the vector of simulated velocities obtained from \mathbf{w}_f at the positions corresponding to the drifter locations.

To ensure numerical stability of the computed flow simulations, optimization constraints are imposed on the simulation residuals. The pressure residuals are limited to a maximum of $1 \cdot 10^{-3}$, while the velocity components and turbulence variables are constrained to remain below $1 \cdot 10^{-4}$.

The lower and upper bounds of the optimization vector are denoted by \mathbf{h}_l and \mathbf{h}_u , respectively. The limits of the tangential velocity optimization variables are set from -0.5 to 0.5 m/s, and the pressure bounds are set from -0.05 to 0.05 m^2/s^2 , based on measurements provided in [115, 117].

An optimization problem is now formulated as follows:

$$\begin{aligned} \underset{\mathbf{h}}{\text{minimize}} \quad & \varepsilon_d(\mathbf{h}) = \frac{1}{n_d} \sum_{i=1}^{n_d} (\mathbf{w}_{r,i} - \mathbf{w}_{s,i}(\mathbf{h}))^2 \\ \text{subject to} \quad & \mathbf{h}_l \leq \mathbf{h} \leq \mathbf{h}_u. \end{aligned} \tag{12.2}$$

The solution to the optimization problem (12.2) is obtained through a PSO algorithm, as it has been proven effective for this class of problems [118]. The PSO implementation used is provided by the *Indago* optimization module for Python [106]. The optimization is performed for 10 minutes or until the defined error satisfies the condition $\varepsilon_d \leq 10^{-4}$, representing the mean squared difference between measured and computed velocities at drifter locations in m/s.

To reproduce the transient behavior of the flow field, a new flow field is obtained through surrogate model fitting every 10 minutes, which is generally sufficient given the realistic dynamics of the coastal flow. Within each 10-minute period, the flow is considered constant.

To compensate for the error of the approximated flow field, the diffusion coefficient is calculated using equation (10.5), where E_e is calculated as the mean distance error for all drifters over a 10-minute interval on the constant flow field. It is defined as the difference between the actual drifter positions and the simulated positions obtained by advecting the drifters using equation (10.1) on the computed flow field.

12.2 Custom sea targets and detection model

In order to conduct the search experiment, four custom identical floating targets were created. They were constructed from 0.5×0.5 m wooden planks painted yellow. Each target featured marking tape to enhance target visibility for other participants in maritime traffic, attached to a 1 m metal rod mounted at the target center. Figure 12.3 shows an example of a custom target deployed in the sea.

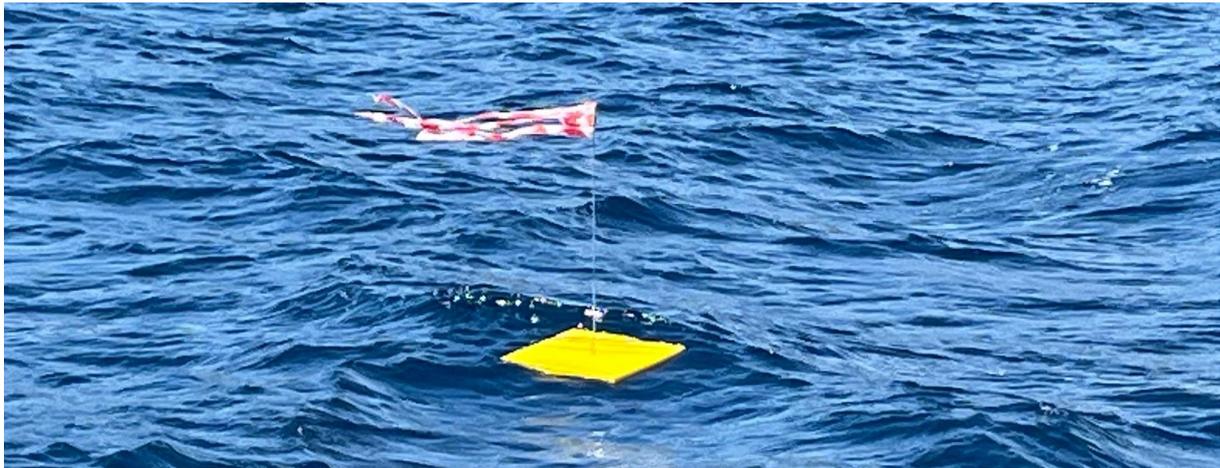


Figure 12.3: Example of a floating target used in the experiment.

To train the sea target detection model and evaluate its performance, a data set of 522 aerial images captured at altitudes ranging from 60 to 100 m were collected. The images contained instances of sea targets, drifters, and boats, with a total of 447 target instances, 45 drifter instances, and 132 boat instances. The data set was uniformly split into training, validation, and test sets in an 80 : 10 : 10 ratio, taking into account both the number of images and the distribution of target instances. For object detection, the YOLOv8 algorithm was employed. The training was initialized with the large pre-trained YOLO model *yolo8l.pt*, pre-trained on the COCO data set. It continued for 100 epochs, with a batch size of 4 and an image size of 640 pixels. It was trained to detect the three mentioned classes.

The performance of the model was then analyzed on the test data set. The model achieved the mean average precision of 0.723 with the intersection-over-union (IoU) threshold of 0.5. For the default detection confidence threshold of 0.001, the model achieves the precision value of 0.861, and the recall of 0.643 across all classes. Since the focus of the experiment is on sea target detection, only the performance metrics corresponding to this class are considered when

constructing the sensing function. The relevant metric is recall, yielding $\mu_{st} = 0.68$ for the target class, considering the default detection confidence threshold of 0.001. Example detections of sea targets are shown in Figure 12.4, with the corresponding detection confidence values displayed next to the class labels.



Figure 12.4: Example detections of sea targets, with detection confidence indicated.

12.3 Experimental search mission

The search experiment took place on 04 June 2025, in Valun Bay, located on the western coast of Cres Island, Croatia ($44^{\circ} 55' N$, $14^{\circ} 22' E$). The team conducting the experiment was divided into two groups: a sea unit onboard a vessel responsible for target and drifter deployment, and a land unit responsible for processing drifter data, approximating the flow field, and performing the UAV search. The land unit was positioned at the UAV base station, located on the central part of the eastern coastline of Valun Bay at an elevation of 85 m, in order to ensure a good overview of the search domain and reliable signal coverage for receiving drifter data and controlling the UAV. Figure 12.5 shows the UAV base station and the surrounding view of Valun Bay.



Figure 12.5: UAV base station overlooking the Valun Bay search domain.

To conduct the search mission using the current state of the flow field, a custom experimental search framework was developed to simultaneously approximate the flow field using real-time drifter measurements and conduct the search based on the computed flow. In the experimental search framework, drifter measurements are obtained via a laptop connected to a dedicated radio receiver module. Since the flow field approximation requires significant computational resources, the drifter data is processed on-site and then forwarded to a remote workstation, where the flow field and drift error are computed. The resulting flow field is transferred to the UAV Ground Control Station (GCS), where advection-diffusion of m is performed and UAV control is executed using HEDAC ergodic guidance. Data sharing between the drifter data acquisition system, the remote workstation, and the GCS is facilitated through Dropbox file synchronization. The complete framework is illustrated in Figure 12.6.

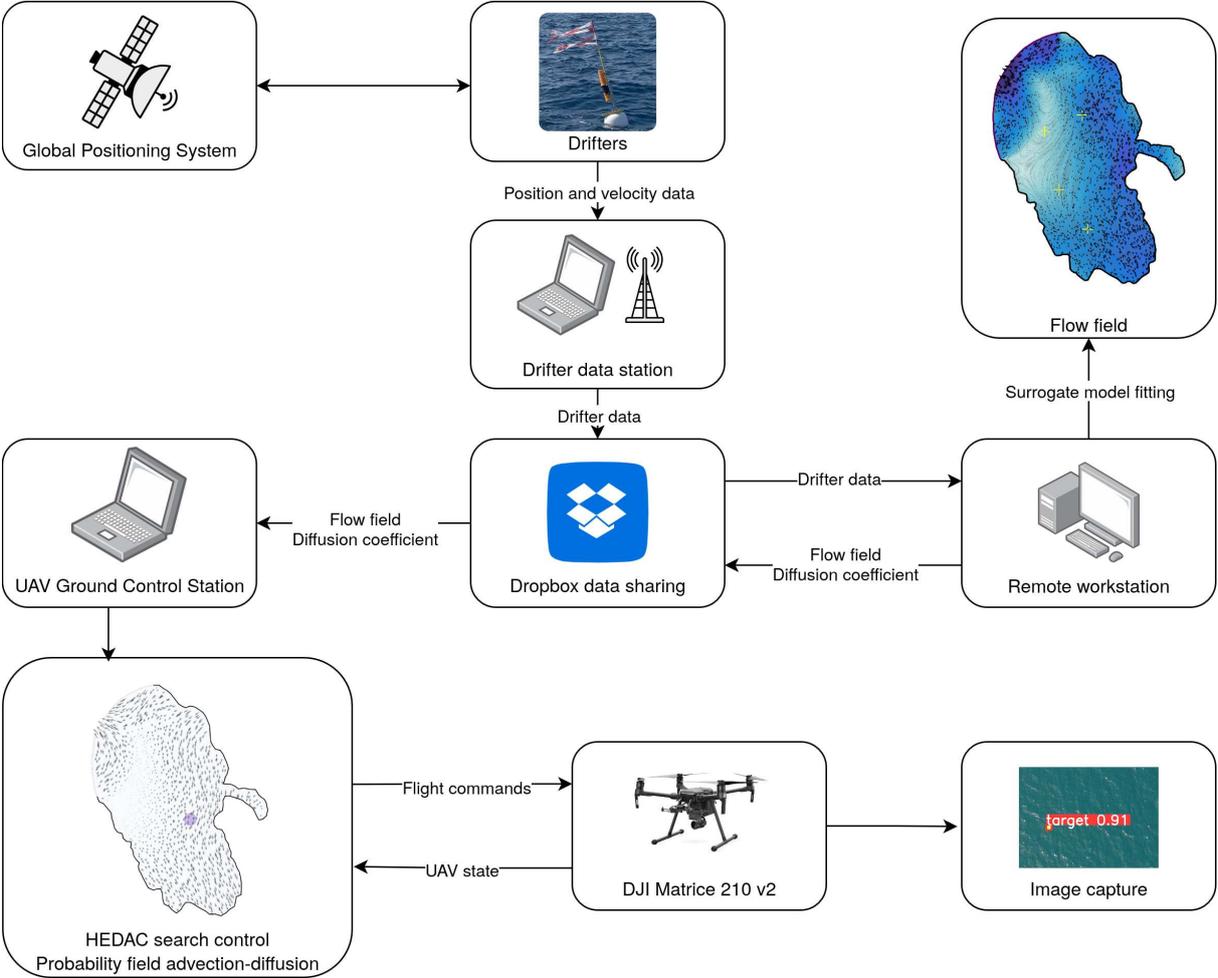


Figure 12.6: Diagram of the experimental dynamic target search framework.

The complete computational domain representing the bay area covers 55.8 km². The flow field was approximated using nine drifters distributed throughout the domain. Four drifters

were evenly scattered around the bay, while the remaining five were placed near or at the target deployment site to achieve a more accurate approximation of the flow field in that region.

The search was conducted using the DJI Matrice 210v2 UAV coupled with the DJI Zenmuse X5S camera, presented in Section 3.4. To prevent overexposure from intense sunlight and water reflections during the experiment, the camera was equipped with an ND16 neutral density filter. The UAV utilized a constant velocity of $v_s = 8$ m/s, minimum turning radius of $R_{min} = 30$ m and a minimum clearance distance of $\delta = 50$ m. The search was conducted from a constant height of $h_{goal} = 75$ m using a pyramidal FOV sensor defined by $\gamma_1 = 64.7^\circ$ and $\gamma_2 = 39.2^\circ$. Utilizing this configuration, each captured image covered roughly 95×53.4 m² of sea surface area. Both the sensing interval Δt_s and the control interval Δt were set to 3 s and the HEDAC parameters were set to $\alpha = 5000$ and $\beta = 0.1$.

In this mission, for each captured image, the probability of undetected target presence within the observed area was discretely reduced according to the recall of the detection model, as

$$m^j(\mathbf{x}) = \begin{cases} m^{j*}(\mathbf{x}) \cdot (1 - \mu_{st}) & \text{if } \mathbf{x} \in \Omega_{FOV} \\ m^{j*}(\mathbf{x}) & \text{otherwise.} \end{cases}$$

The target deployment area was defined as a circular region with a radius of 300 m, located approximately 1.4 km west of the UAV base station. Four targets were deployed in a plus-shaped pattern, with each target positioned approximately 120 m from the center. At 10:15, corresponding to $t = 0$, the targets were successfully deployed in the search area. Figure 12.7 illustrates the deployment of a sea target.



Figure 12.7: Sea unit deploying the experimental target.

The undetected target probability density at $t = 0$ was distributed uniformly within the circular target deployment area. The search domain at $t = 0$, including the approximated flow field, target locations, and the undetected target probability density, is shown in Figure 12.8.

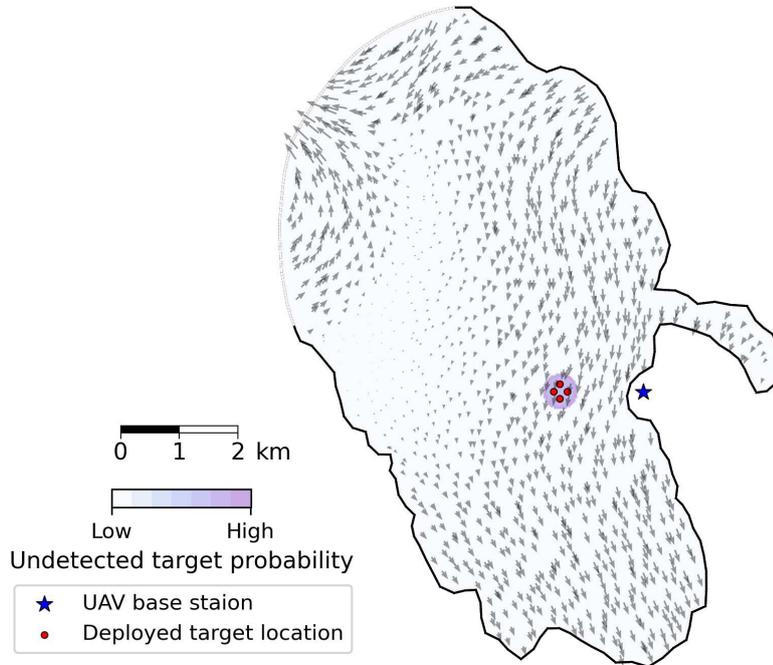


Figure 12.8: The search domain containing the approximated flow field, target locations, and the undetected target probability distribution immediately after target deployment ($t = 0$).

The search mission was delayed 30 minutes from the time of target deployment, and started at 10:45. The undetected target probability was advected and diffused to account for the delay. The targets were not equipped with GPS trackers, so their positions were estimated using Lagrangian particle advection given by equation (10.1). Figure 12.9 displays the undetected target probability, estimated trajectories of the targets during the 30-minute delay and their estimated positions at the start of the search.

The search was initialized by manually flying the UAV to the start position, after which the search was conducted autonomously. It continued until the UAV reached a critical battery depletion level, at approximately 11:45. After the search concluded, the images were processed using the detection model to identify detected targets. The position of each detected target was

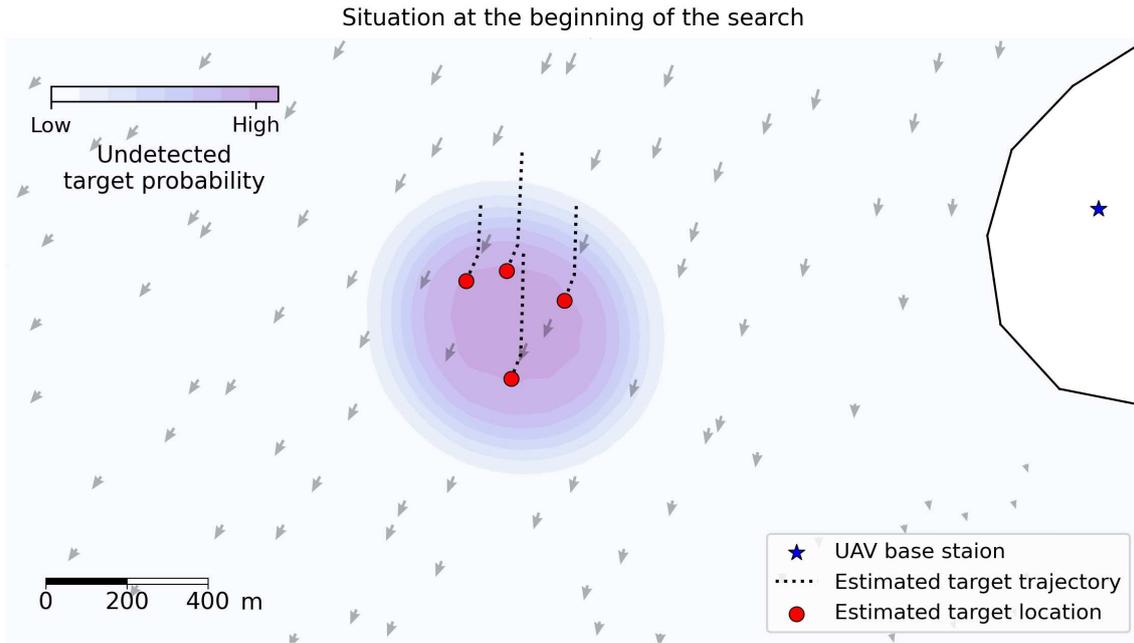


Figure 12.9: Undetected target probability and estimated target positions at the start of the search, including estimated target trajectories during the 30-minute delay.

determined based on the UAV state at the time of image capture and the location of the detected target label within the image. The situation at the end of the search is shown in Figure 12.10, illustrating the undetected target probability, UAV trajectory, target detection locations, estimated target positions, and estimated target trajectories starting from $t = 0$. It is important to note that the detections and their corresponding positions were recorded at various times during the search and therefore do not necessarily correspond to the estimated target positions, which are presented for the search end state. Based on the few rightmost target detection locations, it is assumed that they correspond to the rightmost target, even though the estimated trajectory of that target does not fully align with the detections. This indicates that the flow field approximation exhibits a degree of inaccuracy. Nevertheless, the target was detected multiple times in that area, as the probability field was diffused to account for errors in the flow field approximation.

Since the targets were not equipped with GPS trackers, detections could not be reliably associated with specific targets, and therefore an accurate assessment of the target detection rate could not be obtained for comparison with the estimated survey accomplishment. Moreover, even if such an assessment is possible, the sample size of four targets would not provide strong

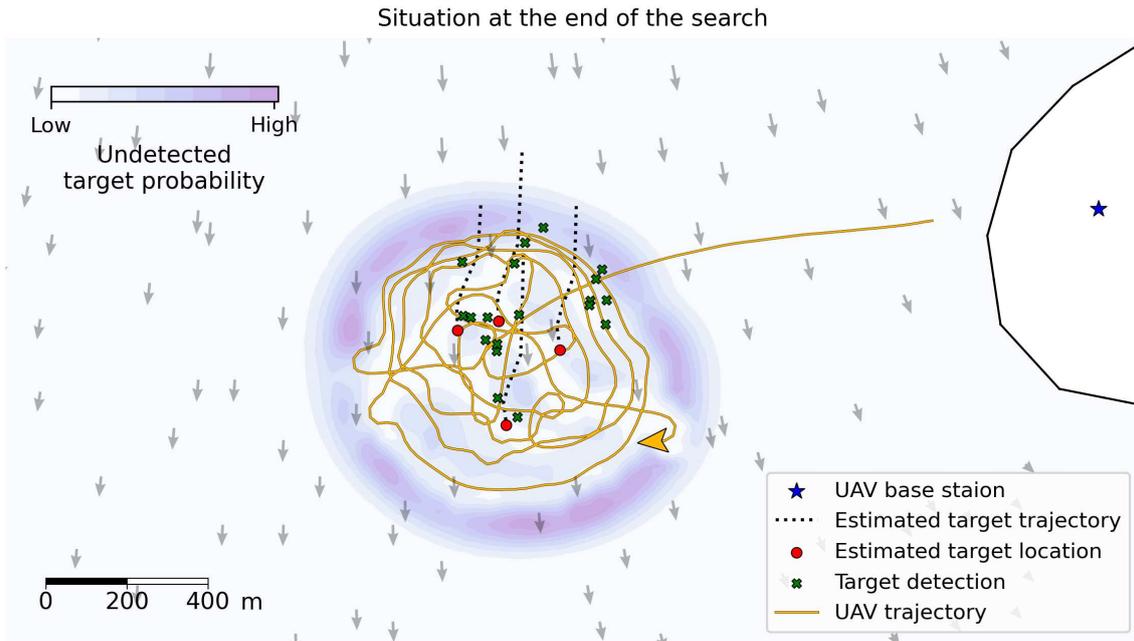


Figure 12.10: Undetected target probability, UAV trajectory, detected target locations, and estimated target positions and trajectories at the end of the search.

statistical significance. Nevertheless, by analyzing the estimated target trajectories and the locations of detected targets, assuming each detection corresponds to the target with the closest trajectory, it can be concluded that all targets were detected.

Given the results, the experiment can be considered a successful validation of the method, as the probability field closely followed the target distribution, and a target was detected outside its estimated trajectory, providing a valid confirmation of the uncertainty compensation methodology.

13 LIMITATIONS AND DISCUSSION

If the UAV and search control parameters are poorly configured, the UAV can get stuck in an indefinite circular motion, as shown in Figure 13.1. This usually happens when a relatively large turning radius is combined with a sensor that has an excessively narrow FOV, or when the search is conducted from insufficient height. To prevent this, the UAV should be configured such that, at the operational search height, the lateral coverage of the sensor relative to the UAV's heading is at least twice the minimum turning radius.

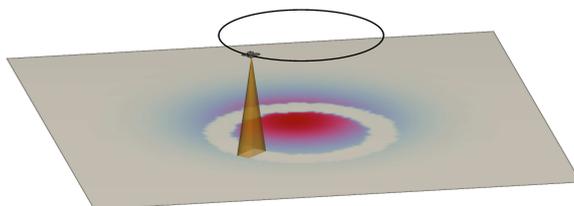


Figure 13.1: Improper parameter configuration causes the UAV to remain in constant circular motion [107].

Another limitation of the method is that the minimum clearance constraint is explicitly enforced only between multiple UAVs or between the UAVs and the domain boundaries. It is not directly checked against the terrain within the area because doing so would be too computationally demanding and would hinder real-time control.

Instead, terrain and structure clearance is ensured by considering the worst-case scenario. By taking into account the desired horizontal clearance δ and the minimum height constraint h_{min} , the maximum terrain incline that the UAV can safely handle is calculated as

$$\alpha_T = \arctan\left(\frac{h_{min}}{\delta}\right).$$

If the UAV satisfies the condition $\alpha_T \geq \beta_T$, where β_T is the steepest terrain slope in the domain, the minimum clearance is guaranteed throughout the domain. Figure 13.2 depicts the UAV approaching the area of maximal terrain incline, with the relevant parameters marked.

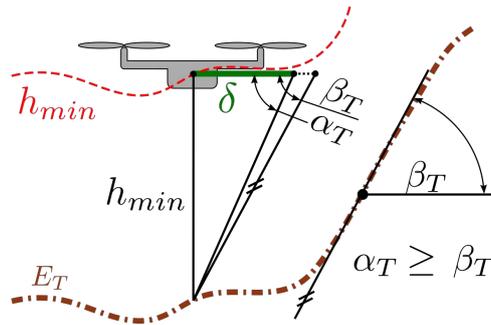


Figure 13.2: Illustration of the UAV approaching the area of maximum terrain slope, with key parameters indicated [107].

The only way to increase the UAV's maximum supported incline is by raising its minimum flight altitude. However, this can only be done to a limited extent, as increasing altitude reduces detection performance. Furthermore, due to the way minimum clearance from the terrain is ensured, the method cannot handle near-vertical or vertical slopes. To address this, future research could consider overlaying a smoothed terrain model on the real terrain. The UAV could then conduct searches based on the smoothed surface, maintaining clearance while allowing operation over steep inclines. In practice, this limitation is not critical for modern UAV platforms, which often include horizontal distance sensors and can maintain the required horizontal separation with the UAV's low-level control rather than with the search control.

Another limitation of the method is that, although it operates in three-dimensional space, it effectively explores only the two-dimensional terrain surface. A key limitation is that it cannot differentiate between vertical levels, such as flying above or below treetops.

In the effort to test the search framework in a realistic scenario including human targets, two experiments with volunteers were conducted on Učka Mountain. The first experiment was carried out to collect a data set used to train the detection model and validate its performance at various search heights. Shortly after, a second experiment was conducted with the goal of validating the search framework. The experimental search area was divided into three zones with uniform probability, where the probability within each zone was determined as the ratio between the number of people in the zone and the zone area. To encourage participants to remain in their designated zones, thereby maintaining the intended probability distribution, volunteers were tasked with finding hidden markers within the area. Each of the three zones contained 50 markers, with each marker serving as a ticket for a chance to win a prize in a tombola held after the experiment. An example of a marker, the zones with indicated marker locations, and the volunteers who participated in the experiment are shown in Figure 13.3.

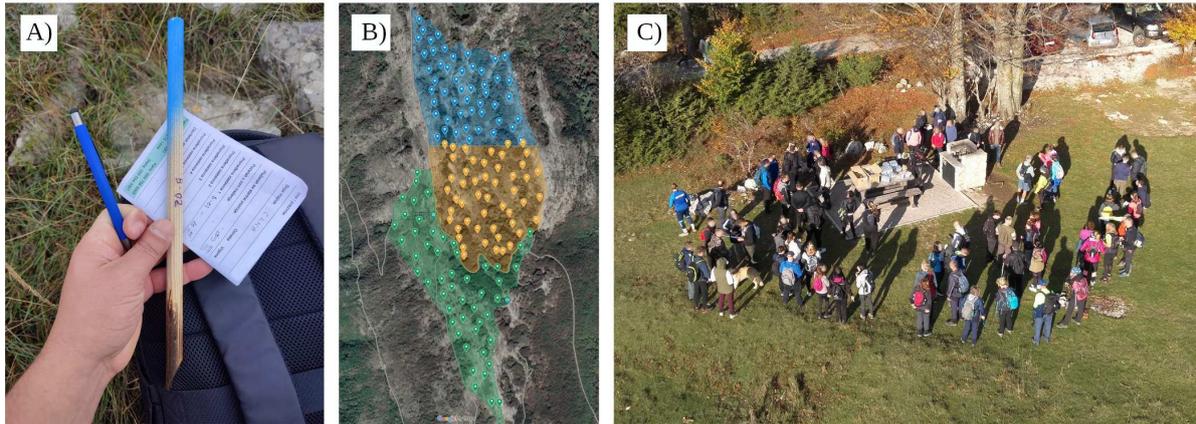


Figure 13.3: Example of a marker (A), zone layout with indicated marker locations (B), and volunteers that participated in the second Učka search experiment (C).

Although the experiment was successfully executed and some valuable conclusions were drawn, as presented in [119], the results were not suitable for validating the proposed framework for several reasons:

- The recall metrics of the YOLO detection did not align with those from the previously collected training database (where people were mainly on roads, and the background environment differed significantly due to seasonal changes between early summer and late autumn).
- The targets (persons), although within the search area, were moving throughout the duration of the search, whereas the proposed framework assumes stationary targets.
- First detection of each individual could not be accurately determined – required for calculating the target detection rate κ – due to similar clothing among participants and some low-resolution images (good enough for detection, but not for identification).

This led to the design of a new experiment with stationary cardboard targets to validate the method. However, the results of the previous experiment provided valuable information for designing this new setup, which was successfully used to validate the proposed method. Furthermore, the experiment with volunteers highlighted a promising direction for future research, indicating that a proper dynamic probability model describing human movement in such scenario is needed to further increase search effectiveness.

Another promising direction for future research is to consider multiple camera sensor orientations, which could improve detection performance and provide a better balance between

detection and area coverage. Additionally, in this study, all objects within the same image were assumed to be at the same distance from the sensor when validating the machine vision detection model, and all targets were treated equally regardless of their position within the image. Future work could account for the target's relative position in the image and the corresponding distance from the sensor, since the proposed probabilistic model already allows for this. This would allow for more accurate evaluation of sensing performance, considering that detection probability may vary if an object is directly beneath the sensor or at an angle.

During the real-world flights over hilly terrain, the system occasionally breached the minimum height constraint, but the violations were minor relative to real-world operational conditions and did not affect overall performance or safety. Additionally, occasional Wi-Fi communication dropouts occurred between the ground station and the android smart device that sends the flight commands to the UAV, highlighting the need for smooth error handling and the implementation of appropriate UAV actions under such conditions for practical deployment. Furthermore, UAV communication cutoffs occurred when the line of sight between the remote controller and the UAV was obstructed by terrain during autonomous missions in hilly areas. This demonstrated the importance of accounting for line-of-sight conditions between the UAV and the ground station. Additionally, a valuable improvement could be achieved by implementing automatic search domain reduction when communication is lost, ensuring that the UAVs avoid revisiting areas with weak signal.

The UAV signal dropouts also occurred during the testing phase of the sea surface search experiment. During flights at an altitude of 75 m above sea level, conducted from the UAV base station situated at an elevation of 85 m, the locations where the UAV began to lose signal were recorded. Figure 13.4 shows the observed signal coverage across the Valun Bay area, with the UAV base station also marked on the image. Similar to land-based flights, the UAV lost signal when the line-of-sight condition was obstructed, as observed in the area south of the base station, where terrain and vegetation still blocked the signal. The flight range remained reliable when line-of-sight was maintained, until the critical range limit was reached.

Another limitation is the complexity of the complete sea surface search framework. The system requires multiple on-site PCs, a central workstation, UAVs, and drifters, all of which must be coordinated during operations. In addition, substantial logistical support is necessary for the deployment of drifters and the transport and setup of UAV base station equipment. This complexity increases the planning effort, operational time, personnel requirements, and



Figure 13.4: Observed UAV signal range during the Valun Bay sea surface search experiment.

financial demands associated with conducting the experiments. Furthermore, it is impossible to provide statistical validation of the dynamic sea target search methodology due to operational constraints, as there is no feasible way to deploy and recover 100 targets from the sea after each experiment.

14 CONCLUSION

UAVs provide a versatile and efficient solution for search operations, combining mobility and adaptability with the ability to operate in complex terrains and dynamic sea environments. Motivated by this capability, the research in this thesis undertook a comprehensive approach, including the development of theoretical foundations, algorithm implementation, and numerical and experimental validation for searching both static and dynamic targets.

The search control governed by the potential field has proven to be both flexible and robust, demonstrating strong performance across distinctly diverse search problems, including hilly terrain and maritime environments. The approach is compatible with the developed enhancements, allowing for seamless integration of the velocity and altitude control achieved via MPC, and the probabilistic model supporting dynamic target search. By extending the search algorithm to account for all environmental and technical conditions, a fully autonomous multi-UAV search procedure was achieved.

Autonomous multi-UAV search control is implemented through a communication bridge between the ground station PC and the UAVs, realized using the DJI's ecosystem. Multi-UAV motion control is extensively validated in simulations over varying terrain, with the MPC framework generating smooth trajectories that closely follow the target search height while maintaining the desired balance between area coverage and detection performance. The motion control consistently produces collision-free flights while adhering to specified velocity, acceleration, and height constraints, and is further validated experimentally over challenging terrain, demonstrating robustness under real-world conditions with multiple sources of uncertainty.

Numerous manual and autonomous multi-rotor UAV flights are conducted to collect a data set of aerial images. These images are used to train multiple detection models based on the YOLO architecture, which are subsequently employed in search missions. The detection models are validated to assess their performance, confirming the influence of image capture height on detection effectiveness. The resulting quantitative performance data are then used to define the detection characteristics for the experimental sensor configurations.

The presented search missions are conducted based on the belief regarding the targets' locations. This belief is represented by a probabilistic model capturing the uncertainty of target positions and adapts according to the achieved search effort, as reflected by the probabilistic sensor model. The static target search methodology is experimentally validated in a hilly environment. The results show considerable agreement between the estimated search performance and the actually detected targets. In the dynamic sea target search methodology, the probability field evolves based on the achieved search effort and additionally accounts for the drift effects caused by the velocity field of the sea surface layer. Advection describes the impact of the velocity field, while diffusion compensates for the uncertainties of target drift. The method is tested numerically, and the computed results show notable agreement between the estimated search performance and the detected targets, validating the approach both with and without introduced uncertainty in target drift. Furthermore, compared to a method that does not account for the dynamic behavior of the sea, the proposed method demonstrates a significant improvement in performance, particularly within the operational range of realistic sea surface velocities and UAV flight speeds. The model was validated through a real-world maritime search experiment, indicating its promising potential to improve SAR operations at sea. The dynamic probabilistic model accounts for the targets' drift and UAV sensing, while simultaneously compensating for errors in the velocity field approximation.

Both the numerical simulations and real-world experiments showcase the benefits of the ergodic search control. Each search mission attains the desired area coverage as represented by the probability distribution. Given sufficient time, it achieves complete coverage of the probability distribution, demonstrating robustness to uncertainties in the targets' location belief. Furthermore, the method's robustness to uncertainty in UAV control and localization errors is demonstrated both numerically and experimentally.

Overall, the presented numerical and experimental results indicate that the potential field-based methods can be utilized to conduct effective autonomous multi-UAV area search missions in both complex natural and maritime environments, thereby confirming the proposed hypothesis.

BIBLIOGRAPHY

- [1] M. Lyu, Y. Zhao, C. Huang, and H. Huang, “Unmanned aerial vehicles for search and rescue: A survey,” *Remote Sensing*, vol. 15, no. 13, p. 3266, 2023.
- [2] M. A. Goodrich *et al.*, “Supporting wilderness search and rescue using a camera-equipped mini uav,” *Journal of Field Robotics*, vol. 25, no. 1-2, pp. 89–110, 2008.
- [3] R. Ashour, S. Aldhaheri, and Y. Abu-Kheil, “Applications of uavs in search and rescue,” in *Unmanned aerial vehicles applications: challenges and trends*, Springer, 2023, pp. 169–200.
- [4] T. Niedzielski, M. Jurecka, B. Miziński, W. Pawul, and T. Motyl, “First successful rescue of a lost person using the human detection system: A case study from beskid niski (se poland),” *Remote Sensing*, vol. 13, no. 23, p. 4903, 2021.
- [5] I. Martinez-Alpiste, G. Golcarenenji, Q. Wang, and J. M. Alcaraz-Calero, “Search and rescue operation using uavs: A case study,” *Expert Systems with Applications*, vol. 178, p. 114937, 2021.
- [6] M. Silvagni, A. Tonoli, E. Zenerino, and M. Chiaberge, “Multipurpose uav for search and rescue operations in mountain avalanche events,” *Geomatics, Natural Hazards and Risk*, vol. 8, no. 1, pp. 18–33, 2017.
- [7] L. Lin and M. A. Goodrich, “Uav intelligent path planning for wilderness search and rescue,” in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2009, pp. 709–714.
- [8] E. T. Alotaibi, S. S. Alqefari, and A. Koubaa, “Lsar: Multi-uav collaboration for search and rescue missions,” *IEEE Access*, vol. 7, pp. 55 817–55 832, 2019.
- [9] R. D. Arnold, H. Yamaguchi, and T. Tanaka, “Search and rescue with autonomous flying robots through behavior-based cooperative intelligence,” *Journal of International Humanitarian Action*, vol. 3, no. 1, pp. 1–18, 2018.

- [10] B. Ai, B. Li, S. Gao, J. Xu, and H. Shang, “An intelligent decision algorithm for the generation of maritime search and rescue emergency response plans,” *IEEE Access*, vol. 7, pp. 155 835–155 850, 2019.
- [11] M. Messmer, B. Kiefer, L. A. Varga, and A. Zell, “Uav-assisted maritime search and rescue: A holistic approach,” *arXiv preprint arXiv:2403.14281*, 2024.
- [12] H.-M. Huang, “Autonomy levels for unmanned systems (alfus) framework volume i: Terminology version 2.0,” 2004.
- [13] Z. Xu, D. Deng, Y. Dong, and K. Shimada, “Dmpc-planner: A real-time uav trajectory planning framework for complex static environments with dynamic obstacles,” in *2022 International Conference on Robotics and Automation (ICRA)*, IEEE, 2022, pp. 250–256.
- [14] P. Yao, H. Wang, and Z. Su, “Real-time path planning of unmanned aerial vehicle for target tracking and obstacle avoidance in complex dynamic environment,” *Aerospace Science and Technology*, vol. 47, pp. 269–279, 2015.
- [15] A. Xu, C. Viriyasuthee, and I. Rekleitis, “Optimal complete terrain coverage using an unmanned aerial vehicle,” in *2011 IEEE International conference on robotics and automation*, IEEE, 2011, pp. 2513–2519.
- [16] S. Ghambari, M. Golabi, L. Jourdan, J. Lepagnet, and L. Idoumghar, “Uav path planning techniques: A survey,” *RAIRO-Operations Research*, vol. 58, no. 4, pp. 2951–2989, 2024.
- [17] J. L. Junell, E.-J. Van Kampen, C. C. de Visser, and Q. P. Chu, “Reinforcement learning applied to a quadrotor guidance law in autonomous flight,” in *AIAA guidance, navigation, and control conference*, 2015, p. 1990.
- [18] A. Loquercio, A. I. Maqueda, C. R. Del-Blanco, and D. Scaramuzza, “Dronet: Learning to fly by driving,” *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1088–1095, 2018.
- [19] Y. Wang, P. Bai, X. Liang, W. Wang, J. Zhang, and Q. Fu, “Reconnaissance mission conducted by uav swarms based on distributed pso path planning algorithms,” *IEEE access*, vol. 7, pp. 105 086–105 099, 2019.

- [20] J. L. Foo, J. Knutzon, J. Oliver, and E. Winer, “Three-dimensional multi-objective path planner for unmanned aerial vehicles using particle swarm optimization,” in *48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2007, p. 1881.
- [21] H. Shorakaei, M. Vahdani, B. Imani, and A. Gholami, “Optimal cooperative path planning of unmanned aerial vehicles by a parallel genetic algorithm,” *Robotica*, vol. 34, no. 4, pp. 823–836, 2016.
- [22] J. D. Silva Arantes, M. d. Silva Arantes, C. F. Motta Toledo, O. T. Júnior, and B. C. Williams, “Heuristic and genetic algorithm approaches for uav path planning under critical situation,” *International Journal on Artificial Intelligence Tools*, vol. 26, no. 01, p. 1 760 008, 2017.
- [23] M. D. Phung and Q. P. Ha, “Safety-enhanced uav path planning with spherical vector-based particle swarm optimization,” *Applied Soft Computing*, vol. 107, p. 107 376, 2021.
- [24] 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 Systems with Applications*, vol. 38, no. 8, pp. 10 036–10 048, 2011.
- [25] V. Roberge, M. Tarbouchi, and G. Labonté, “Comparison of parallel genetic algorithm and particle swarm optimization for real-time uav path planning,” *IEEE Transactions on industrial informatics*, vol. 9, no. 1, pp. 132–141, 2012.
- [26] R. Shivgan and Z. Dong, “Energy-efficient drone coverage path planning using genetic algorithm,” in *2020 IEEE 21st International Conference on High Performance Switching and Routing (HPSR)*, IEEE, 2020, pp. 1–6.
- [27] Y. Kuwata, T. Schouwenaars, A. Richards, and J. How, “Robust constrained receding horizon control for trajectory planning,” in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2005, p. 6079.
- [28] J. Bellingham, A. Richards, and J. P. How, “Receding horizon control of autonomous aerial vehicles,” in *Proceedings of the 2002 American control conference (IEEE Cat. No. CH37301)*, IEEE, vol. 5, 2002, pp. 3741–3746.

- [29] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, “Receding horizon path planning for 3d exploration and surface inspection,” *Autonomous Robots*, vol. 42, no. 2, pp. 291–306, 2018.
- [30] 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, p. 108 637, 2020.
- [31] 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, p. 84, 2023.
- [32] S. Ivić, A. Andrejčuk, and S. Družeta, “Autonomous control for multi-agent non-uniform spraying,” *Applied Soft Computing*, vol. 80, pp. 742–760, 2019.
- [33] D. Gkouletsos, A. Iannelli, M. H. de Badyn, and J. Lygeros, “Decentralized trajectory optimization for multi-agent ergodic exploration,” *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6329–6336, 2021.
- [34] A. Mavrommati, E. Tzorakoleftherakis, I. Abraham, and T. D. Murphey, “Real-time area coverage and target localization using receding-horizon ergodic exploration,” *IEEE Transactions on Robotics*, vol. 34, no. 1, pp. 62–80, 2017.
- [35] S. Ivić, “Motion control for autonomous heterogeneous multiagent area search in uncertain conditions,” *IEEE Transactions on Cybernetics*, vol. 52, no. 5, pp. 3123–3135, 2020.
- [36] Y. Zheng and C. Zhai, “Distributed coverage control of multi-agent systems in uncertain environments using heat transfer equations,” *arXiv preprint arXiv:2204.09289*, 2022.
- [37] B. Crnković, S. Ivić, and M. Zovko, “Fast algorithm for centralized multi-agent maze exploration,” *arXiv preprint arXiv:2310.02121*, 2023.
- [38] I. Abraham and T. D. Murphey, “Decentralized ergodic control: Distribution-driven sensing and exploration for multiagent systems,” *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 2987–2994, 2018.
- [39] T. J. Stastny, G. A. Garcia, and S. S. Keshmiri, “Collision and obstacle avoidance in unmanned aerial systems using morphing potential field navigation and nonlinear model predictive control,” *Journal of dynamic systems, measurement, and control*, vol. 137, no. 1, 2015.

- [40] A. Marchidan and E. Bakolas, "Collision avoidance for an unmanned aerial vehicle in the presence of static and moving obstacles," *Journal of Guidance, Control, and Dynamics*, vol. 43, no. 1, pp. 96–110, 2020.
- [41] J. Tordesillas and J. P. How, "Mader: Trajectory planner in multiagent and dynamic environments," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 463–476, 2021.
- [42] B. Zhou, F. Gao, L. Wang, C. Liu, and S. Shen, "Robust and efficient quadrotor trajectory generation for fast autonomous flight," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3529–3536, 2019.
- [43] M. J. Iqbal, M. M. Iqbal, I. Ahmad, M. O. Alassafi, A. S. Alfakeeh, and A. Alhomoud, "Real-time surveillance using deep learning," *Security and Communication Networks*, vol. 2021, no. 1, p. 6 184 756, 2021.
- [44] S. Lee, Y. Song, and S.-H. Kil, "Feasibility analyses of real-time detection of wildlife using uav-derived thermal and rgb images," *Remote Sensing*, vol. 13, no. 11, p. 2169, 2021.
- [45] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.
- [46] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.
- [47] P. Mittal, R. Singh, and A. Sharma, "Deep learning-based object detection in low-altitude uav datasets: A survey," *Image and Vision computing*, vol. 104, p. 104 046, 2020.
- [48] D.-M. Seo, H.-J. Woo, M.-S. Kim, W.-H. Hong, I.-H. Kim, and S.-C. Baek, "Identification of asbestos slates in buildings based on faster region-based convolutional neural network (faster r-cnn) and drone-based aerial imagery," *Drones*, vol. 6, no. 8, p. 194, 2022.
- [49] X. Lou *et al.*, "Measuring loblolly pine crowns with drone imagery through deep learning," *Journal of Forestry Research*, pp. 1–12, 2022.

- [50] A. Ramachandran and A. K. Sangaiah, "A review on object detection in unmanned aerial vehicle surveillance.," *International Journal of Cognitive Computing in Engineering*, 2021.
- [51] G.-S. Xia *et al.*, "Dota: A large-scale dataset for object detection in aerial images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3974–3983.
- [52] K. Levi and Y. Weiss, "Learning object detection from a small number of examples: The importance of good features," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, IEEE, vol. 2, 2004, pp. II–II.
- [53] X. Yang, F. Sang, T. Wang, X. Pei, H. Wang, and T. Hou, "Research on the influence of camera velocity on image blur and a method to improve object detection precision," in *2021 International Conference on Cyber-Physical Social Intelligence (ICCSI)*, IEEE, 2021, pp. 1–6.
- [54] C. Liu, Y. Tao, J. Liang, K. Li, and Y. Chen, "Object detection based on yolo network," in *2018 IEEE 4th information technology and mechatronics engineering conference (ITOEC)*, IEEE, 2018, pp. 799–803.
- [55] Y. Pei, Y. Huang, Q. Zou, X. Zhang, and S. Wang, "Effects of image degradation and degradation removal to cnn-based image classification," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 4, pp. 1239–1253, 2019.
- [56] M. Farah and A. Alruwaili, "Optimizing ground sampling distance for drone-based gis mapping: A case study in riyadh, saudi arabia," in *2024 9th International Conference on Robotics and Automation Engineering (ICRAE)*, IEEE, 2024, pp. 1–5.
- [57] J. Shermeyer and A. Van Etten, "The effects of super-resolution on object detection performance in satellite imagery," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.
- [58] S. Niu, Z. Nie, G. Li, and W. Zhu, "Multi-altitude corn tassel detection and counting based on uav rgb imagery and deep learning," *Drones*, vol. 8, no. 5, p. 198, 2024.
- [59] L. Qingqing *et al.*, "Towards active vision with uavs in marine search and rescue: Analyzing human detection at variable altitudes," in *2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, IEEE, 2020, pp. 65–70.

- [60] J. Suo, T. Wang, X. Zhang, H. Chen, W. Zhou, and W. Shi, “Hit-uav: A high-altitude infrared thermal dataset for unmanned aerial vehicle-based object detection,” *Scientific Data*, vol. 10, no. 1, p. 227, 2023.
- [61] 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 *2021 IEEE 6th International Conference on Signal and Image Processing (ICSIP)*, IEEE, 2021, pp. 33–39.
- [62] A. Kosari, H. Maghsoudi, A. Lavaei, and R. Ahmadi, “Optimal online trajectory generation for a flying robot for terrain following purposes using neural network,” *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 229, no. 6, pp. 1124–1141, 2015.
- [63] C. D. Melita, D. C. Guastella, L. Cantelli, G. Di Marco, I. Minio, and G. Muscato, “Low-altitude terrain-following flight planning for multirotors,” *Drones*, vol. 4, no. 2, p. 26, 2020.
- [64] Z. Xia, S. Bu, L. Chen, and P. Han, “A fast reconstruction method based on real-time terrain following,” in *2022 China Automation Congress (CAC)*, IEEE, 2022, pp. 1479–1484.
- [65] E. N. Johnson, J. G. Mooney, C. Ong, J. Hartman, and V. Sahasrabudhe, “Flight testing of nap-of-the-earth unmanned helicopter systems,” in *67th American Helicopter Society International Annual Forum 2011*, 2011, pp. 3180–3192.
- [66] L. Bauersfeld and D. Scaramuzza, “Range, endurance, and optimal speed estimates for multicopters,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2953–2960, 2022.
- [67] H. Gao *et al.*, “Energy-efficient velocity control for massive numbers of uavs: A mean field game approach,” *IEEE Transactions on Vehicular Technology*, vol. 71, no. 6, pp. 6266–6278, 2022.
- [68] H. Song *et al.*, “Multi-uav disaster environment coverage planning with limited-endurance,” in *2022 International Conference on Robotics and Automation (ICRA)*, IEEE, 2022, pp. 10 760–10 766.
- [69] L. S. Pitsoulis and M. G. Resende, “Greedy randomized adaptive search procedures,” *Handbook of applied optimization*, pp. 168–183, 2002.

- [70] J. F. Araujo, P. Sujit, and J. B. Sousa, “Multiple uav area decomposition and coverage,” in *2013 IEEE symposium on computational intelligence for security and defense applications (CISDA)*, IEEE, 2013, pp. 30–37.
- [71] L. M. Miller, Y. Silverman, M. A. MacIver, and T. D. Murphey, “Ergodic exploration of distributed information,” *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 36–52, 2015.
- [72] 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.
- [73] A. Hubenko, V. A. Fonoberov, G. Mathew, and I. Mezić, “Multiscale adaptive search,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 4, pp. 1076–1087, 2011.
- [74] S. Ivić, B. Crnković, H. Arbabi, S. Loire, P. Clary, and I. Mezić, “Search strategy in a complex and dynamic environment: The mh370 case,” *Scientific Reports*, vol. 10, no. 1, p. 19 640, 2020.
- [75] M. Schwenzer, M. Ay, T. Bergs, and D. Abel, “Review on model predictive control: An engineering perspective,” *The International Journal of Advanced Manufacturing Technology*, vol. 117, no. 5, pp. 1327–1349, 2021.
- [76] S. Ivić, B. Crnković, and Mezić, “Ergodicity-based cooperative multiagent area coverage via a potential field,” *IEEE Transactions on Cybernetics*, vol. 47, no. 8, pp. 1983–1993, Aug. 2017, ISSN: 2168-2267.
- [77] S. Ivić, A. Sikirica, and B. Crnković, “Constrained multi-agent ergodic area surveying control based on finite element approximation of the potential field,” *Engineering Applications of Artificial Intelligence*, vol. 116, p. 105 441, 2022.
- [78] S. Ivić, B. Crnković, L. Grbčić, and L. Matleković, “Multi-uav trajectory planning for 3d visual inspection of complex structures,” *Automation in Construction*, vol. 147, p. 104 709, 2023.
- [79] C. Bilaloglu, T. Löw, and S. Calinon, “Whole-body ergodic exploration with a manipulator using diffusion,” *IEEE Robotics and Automation Letters*, 2023.

- [80] C. Bilaloglu, T. Löw, and S. Calinon, “Tactile ergodic coverage on curved surfaces,” *IEEE Transactions on Robotics (T-RO)*, vol. 41, pp. 1421–1435, 2025. DOI: 10.1109/TR0.2025.3532513.
- [81] T. Löw, J. Maceiras, and S. Calinon, “Drozbot: Using ergodic control to draw portraits,” *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 728–11 734, 2022.
- [82] Y. Lun, H. Wang, J. Wu, Y. Liu, and Y. Wang, “Target search in dynamic environments with multiple solar-powered uavs,” *IEEE Transactions on Vehicular Technology*, vol. 71, no. 9, pp. 9309–9321, 2022.
- [83] L. Li, X. Zhang, W. Yue, and Z. Liu, “Cooperative search for dynamic targets by multiple uavs with communication data losses,” *ISA transactions*, vol. 114, pp. 230–241, 2021.
- [84] T. Yang, Z. Jiang, R. Sun, N. Cheng, and H. Feng, “Maritime search and rescue based on group mobile computing for unmanned aerial vehicles and unmanned surface vehicles,” *IEEE transactions on industrial informatics*, vol. 16, no. 12, pp. 7700–7708, 2020.
- [85] M. A. Alanezi *et al.*, “Dynamic target search using multi-uavs based on motion-encoded genetic algorithm with multiple parents,” *IEEE Access*, vol. 10, pp. 77 922–77 939, 2022.
- [86] H. Coffin, I. Abraham, G. Sartoretti, T. Dillstrom, and H. Choset, “Multi-agent dynamic ergodic search with low-information sensors,” in *2022 International Conference on Robotics and Automation (ICRA)*, IEEE, 2022, pp. 11 480–11 486.
- [87] Airmobi. “Skyeye 2600 fixed-wing uav platform.” Accessed: 2025-07-30. (n.d.), [Online]. Available: <https://www.airmobi.com/product/skyeye-2600-fixed-wing-uav-platform/> (visited on 07/30/2025).
- [88] I. Technology. “The galaxy uav.” Accessed: 2025-07-30. (2025), [Online]. Available: <https://innoflighttechnology.com/the-galaxy/> (visited on 07/30/2025).
- [89] Aeromotus. “Dji phantom 4 pro+ obsidian edition.” Accessed: 2025-07-30. (2025), [Online]. Available: <https://www.aeromotus.com/product/phantom-4-pro-plus-obsidian/> (visited on 07/30/2025).
- [90] KOREC Group. “Image of wingtraone uav.” Product image. (n.d.), [Online]. Available: <https://www.korecgroup.com/product/wingtraone/> (visited on 07/30/2025).

- [91] AMain Hobbies. “Rc airplanes: Understanding transmitter flight controls.” Image retrieved from webpage. (n.d.), [Online]. Available: <https://www.amainhobbies.com/rc-airplanes-understanding-transmitter-flight-controls/cp1090> (visited on 08/01/2025).
- [92] DJI. “Zenmuse x5s.” Accessed: 2025-07-31. (n.d.), [Online]. Available: <https://www.aviteh.hr/dji-zenmuse-x5s.html> (visited on 07/31/2025).
- [93] Ysense. “Image of aq600 pro 5-bands multispectral camera.” Image taken from the product page, accessed 2025-07-31. (n.d.), [Online]. Available: <https://www.ghostysky.com/product/aq600-pro-5-bands-multispectral-camera/> (visited on 07/31/2025).
- [94] DJI. “Flir zenmuse xt2 thermal camera - 640x512 30hz 25mm.” Accessed: 2025-07-31. (n.d.), [Online]. Available: <https://www.dronenerds.com/products/dji-flir-zenmuse-xt2-thermal-camera-640x512-30hz-25mm-zxt2a25fr-dji> (visited on 07/31/2025).
- [95] —, “Image of zenmuse l2 lidar camera (2-year coverage).” Image taken from the product page, accessed 2025-07-31. (n.d.), [Online]. Available: <https://dronovishop.hr/dji-zenmuse-l2-lidar-2y/> (visited on 07/31/2025).
- [96] J. Hollesen, M. S. Jepsen, and H. Harmsen, “The application of rgb, multispectral, and thermal imagery to document and monitor archaeological sites in the arctic: A case study from south greenland,” *Drones*, vol. 7, no. 2, p. 115, 2023.
- [97] M. Sharon. “Image from “demystifying lidar point cloud data”.” Image taken from the Medium article, accessed 2025-07-31. (2024), [Online]. Available: <https://medium.com/@matt-sharon/lidar-point-cloud-data-guide-62c126101c08> (visited on 07/31/2025).
- [98] M. D. Services. “Image of approved used dji matrice 210 rtk.” Image taken from the product page, accessed 2025-07-31. (n.d.), [Online]. Available: <https://munsterdroneservices.com/product/approved-used-dji-matrice-210-rtk/> (visited on 07/31/2025).
- [99] V. Drones. “Dji mavic 2 enterprise dual thermal drone.” Accessed: 2025-07-30. (2025), [Online]. Available: <https://volatusdrones.com/blogs/posts-without-blog/dji-mavic-2-enterprise-dual-thermal-drone> (visited on 07/30/2025).

- [100] L. Lanča, S. Dumenčić, K. Jakac, and S. Ivić, “Experimental validation and robustness analysis of ergodic uav control,” in *2025 11th International Conference on Automation, Robotics, and Applications (ICARA)*, IEEE, 2025, pp. 257–261.
- [101] L. Lanča, M. Mališa, K. Jakac, and S. Ivić, “Optimal flight speed and height parameters for computer vision detection in uav search,” *Drones*, vol. 9, no. 9, p. 595, 2025.
- [102] B. O. Koopman, “The theory of search. ii. target detection,” *Operations research*, vol. 4, no. 5, pp. 503–531, 1956.
- [103] Copernicus, *Copernicus digital elevation model product handbook*, version 2.1, Campaign ID: GEO.2018-1988-2, European Space Agency, Jun. 25, 2020. [Online]. Available: https://dataspace.copernicus.eu/sites/default/files/media/files/2024-06/geo1988-copernicusdem-spe-002_producthandbook_i5.0.pdf.
- [104] G. Krieger *et al.*, “Tandem-x: A satellite formation for high-resolution sar interferometry,” *IEEE transactions on geoscience and remote sensing*, vol. 45, no. 11, pp. 3317–3341, 2007.
- [105] C. Audet and J. E. Dennis Jr, “Mesh adaptive direct search algorithms for constrained optimization,” *SIAM Journal on optimization*, vol. 17, no. 1, pp. 188–217, 2006.
- [106] S. Ivić and S. Družeta, *Indago: Python 3 module for numerical optimization*, version 0.4.5. [Online]. Available: <https://pypi.org/project/Indago/0.4.5/>.
- [107] L. Lanča, K. Jakac, and S. Ivić, “Model predictive altitude and velocity control in ergodic potential field directed multi-uav search,” *arXiv preprint arXiv:2401.02899*, 2024.
- [108] J. D. Hunter, “Matplotlib: A 2d graphics environment,” *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007. DOI: 10.1109/MCSE.2007.55.
- [109] C. Sullivan and A. Kaszynski, “Pyvista: 3d plotting and mesh analysis through a streamlined interface for the visualization toolkit (vtk),” *Journal of Open Source Software*, vol. 4, no. 37, p. 1450, 2019.
- [110] A. Torralba, “How many pixels make an image?” *Visual neuroscience*, vol. 26, no. 1, pp. 123–131, 2009.

- [111] C. Chen *et al.*, “Yolo-based uav technology: A review of the research and its applications,” *Drones*, vol. 7, no. 3, p. 190, 2023.
- [112] J. H. Ferziger, M. Perić, and R. L. Street, *Computational methods for fluid dynamics*. springer, 2019.
- [113] O. Ltd, *Esi open CFD release openfoam® v2406*, <https://www.openfoam.com/news/main-news/openfoam-v2406>, [Online; Date last accessed 10-March-2025], 2025.
- [114] K. Jakac, L. Lanča, A. Sikirica, and S. Ivić, “Efficient data-driven flow modeling for accurate passive scalar advection in submesoscale domains,” *arXiv preprint arXiv:2503.07452*, 2025.
- [115] S. Cosoli, M. Ličer, M. Vodopivec, and V. Malačič, “Surface circulation in the gulf of trieste (northern adriatic sea) from radar, model, and adcp comparisons,” *Journal of Geophysical Research: Oceans*, vol. 118, no. 11, pp. 6183–6200, 2013.
- [116] L. Lanča, K. Jakac, S. Calinon, and S. Ivić, “Ergodic exploration of dynamic distribution,” *arXiv preprint arXiv:2503.11235*, 2025.
- [117] G. Notarstefano, P.-M. Poulain, and E. Mauri, “Estimation of surface currents in the adriatic sea from sequential infrared satellite images,” *Journal of Atmospheric and Oceanic Technology*, vol. 25, no. 2, pp. 271–285, 2008.
- [118] K. Jakac, L. Lanča, A. Sikirica, and S. Ivić, “Approximation of sea surface velocity field by fitting surrogate two-dimensional flow to scattered measurements,” *arXiv preprint arXiv:2401.12746*, 2024.
- [119] S. Dumenčić, L. Lanča, K. Jakac, and S. Ivić, “Experimental validation of uav search and detection system in real wilderness environment,” *arXiv preprint arXiv:2502.17372*, 2025.

LIST OF FIGURES

3.1	Airmobi Skyeye 2600 fixed-wing UAV [87].	14
3.2	Examples of rotary-wing UAVs: The Rotor single-wing [88] and DJI Phantom 4 Pro multi-rotor [89] UAVs.	15
3.3	WingtraOne hybrid VTOL UAV [90].	15
3.4	Local UAV coordinate system and attitude angles	16
3.5	Multi-rotor motor power distributions during pitch, roll, and yaw changes. Red arrows indicate motor rotation direction and rotational speed (thicker arrows = higher speed).	17
3.6	Fixed-wing aircraft attitude control components [91].	18
3.7	Range of UAV perception sensors [92–95] and examples of their data outputs [96, 97].	19
3.8	Experimental UAVs used in real-world testing [92, 98, 99].	20
3.9	Overview of the communication scheme between the PC and the UAVs [100].	21
4.1	Velocity components and constraints for multi-rotor (A) and fixed-wing (B) UAVs.	24
4.2	Representation of the pyramidal and conical FOV angles (γ), along with the global and local UAV coordinate frames and an example of vector \mathbf{R}	29
4.3	Visualization of sensing function ψ	30
6.1	UAVs executing the collision avoidance maneuver.	42
7.1	Geographical region (left) and the corresponding Digital Elevation Model (right).	48
7.2	Visual representation of the predicted two-dimensional path generated along the potential field gradient.	49
7.3	Graphical representation of the predicted path (A) alongside the corresponding MPC optimization process (B).	55

8.1	Dependence of recall on image capture height (A) and detection rate functions for UAVs used in the simulations (B).	61
8.2	Illustration of terrain incline (A) and initial target probability density (B) for the Plastic world case.	63
8.3	Analysis of a 1000-second trajectory for UAV A during exploration of the Plastic World case.	64
8.4	Survey accomplishment (A) and computation times (B) for the Plastic world case.	64
8.5	Mount Vesuvius terrain slope (A) and undetected target probability at $t = 0$ s (B).	65
8.6	UAV trajectories at 1400 s of the Mount Vesuvius survey (A), with detailed analysis of the red-highlighted trajectory shown in panels (B–E) [107]	66
8.7	UAV trajectories for the Mount Vesuvius survey case across all compared methods.	67
8.8	Mount Vesuvius case η across compared methods (A) and computation time per step for the HEDAC + MPC method (B).	68
8.9	UAV trajectories during the Mount Vesuvius survey with a no-fly zone at 1400 s (A), including a detailed analysis of the red-highlighted trajectory shown in panels (B–E)	68
8.10	Star dunes terrain with incline representation (A) and initial undetected target probability density (B).	69
8.11	Star dunes survey UAV trajectories at $t = 1300$ s (A), along with analysis of flight parameters over the duration of the red-highlighted trajectory (B-E).	70
8.12	Star dunes case, η across methods (A) and HEDAC + MPC computation time per step (B).	71
8.13	Statistical analysis of method robustness demonstrated on the Mount Vesuvius test case.	73
9.1	Illustration of the Raša River valley.	75
9.2	Motion control validation case: domain incline (A) and area of interest encoded in the undetected-target probability density field (B).	76
9.3	UAV trajectory for a single flight mission during motion-control validation with prediction horizon $\tau_{max} = 30$ s (A), and corresponding trajectory analysis (B-E).	77
9.4	Predicted UAV position errors during all real-world flight experiments.	78

9.5	Location error between the simulated trajectory (zero error) and the executed real-world trajectory for both prediction horizon lengths of 15 s and 30 s.	79
9.6	Survey accomplishment for all simulated missions and real-world missions, for both prediction horizon lengths of 15 s and 30 s.	80
9.7	Flight height analysis and comparison with simulation results for $\tau_{max} = 30$ s (A) and $\tau_{max} = 15$ s (B).	80
9.8	Representative custom cardboard targets used in the experiment.	81
9.9	Recall values of the cardboard target detection model at discrete heights and the corresponding regression curve (A), along with detection rate functions Γ for the UAVs used in the experiments (B).	83
9.10	Experimental area on Učka Mountain including two Matrice 210 v2 UAVs.	84
9.11	Učka experiment terrain (A) and the initial undetected target probability density with target locations shown as black dots (B).	85
9.12	Detection model classification outcomes: true positives (A) and false positives (B) annotated by the model in blue, and false negatives (C) manually annotated in red. True negatives are represented by regions with no annotations in the images. Numbers next to the annotations in (A,B) indicate the model's detection confidence.	85
9.13	Mission 1 trajectories and marked targets (A) and flight parameters graphs (B-E) for the red trajectory executed by the Matrice UAV.	86
9.14	Mission 2 trajectories (A) and flight parameters graphs (B-E) for the yellow trajectory executed by the Mavic UAV.	87
9.15	Survey accomplishment η , target detection rate κ , and their relative error over the duration of both search missions.	88
11.1	Cavity flow field during the simulation and the distribution of simulated targets at $t = 0$	99
11.2	Undetected target probability field, simulated targets, and UAV trajectories for both the proposed and baseline methods at $t = 450$, together with η and κ values throughout the search.	100
11.3	Performance comparison of the proposed and baseline methods across different λ values.	101

11.4	Performance analysis of the proposed method across different α values and varying numbers of UAVs conducting the search.	102
11.5	Flow field and target positions at $t = 0$, at search start $t = 10800$ s, and at search end $t = 21300$ s.	104
11.6	Unije Channel search at $t = 21300$, showing UAV trajectories from the final search wave, target locations and detection status for both the uncompensated and compensated cases, along with the evolution of η and κ over time.	105
11.7	UAV trajectories, targets' positions and their detection status at the end of the first search mission in the MH370 simulation executed with HEDAC, along with the underlying flow field and the evolution of κ and η over all five search missions [116].	107
12.1	Deployed drifter used in the experiment for obtaining surface flow velocity measurements.	109
12.2	Visual representation of the bounded, open, and fused flow fields, along with the positions of the boundary control points and the drifter measurement locations. The figure also displays the measured reference velocity vectors and the optimized velocity vectors within the fused flow.	110
12.3	Example of a floating target used in the experiment.	112
12.4	Example detections of sea targets, with detection confidence indicated.	113
12.5	UAV base station overlooking the Valun Bay search domain.	113
12.6	Diagram of the experimental dynamic target search framework.	114
12.7	Sea unit deploying the experimental target.	115
12.8	The search domain containing the approximated flow field, target locations, and the undetected target probability distribution immediately after target deployment ($t = 0$).	116
12.9	Undetected target probability and estimated target positions at the start of the search, including estimated target trajectories during the 30-minute delay.	117
12.10	Undetected target probability, UAV trajectory, detected target locations, and estimated target positions and trajectories at the end of the search.	118
13.1	Improper parameter configuration causes the UAV to remain in constant circular motion [107].	119

13.2	Illustration of the UAV approaching the area of maximum terrain slope, with key parameters indicated [107].	120
13.3	Example of a marker (A), zone layout with indicated marker locations (B), and volunteers that participated in the second Učka search experiment (C).	121
13.4	Observed UAV signal range during the Valun Bay sea surface search experiment.	123

LIST OF TABLES

3.1	Overview of technical specifications for the UAVs used in field experiments . . .	21
8.1	Motion, vision/sensing and control UAV parameters used in simulations [107].	58
8.2	Overview of simulated test cases and their parameters [107]	59
8.3	Variables' uncertainty error	72
9.1	Motion, sensing, and control parameters of the UAV used in the search methodology validation experiment [107].	83
11.1	Computation time for the Unije Channel search scenario [116].	106

CURRICULUM VITAE

Luka Lanča was born on April 1, 1998, in Rijeka, Croatia. He completed his Bachelor's degree in Mechanical Engineering at the Faculty of Engineering, University of Rijeka, in 2019, and his Master's degree in Mechanical Engineering at the same university in 2021. Later that year, he began his doctoral studies in Technical Sciences, focusing on Mechanical Engineering, Fundamental Technical Sciences, and Interdisciplinary Technical Sciences. Since 2021, Mr. Lanča has worked as a research assistant at the Faculty of Engineering, specializing in ergodic and model predictive control, autonomous Unmanned Aerial Vehicle (UAV) systems, machine learning-based detection models, and computational fluid dynamics. As part of the research group working on the Autonomous Oceanic Search and Rescue project, he actively contributes to the development of an autonomous UAV system for both land-based and maritime search and rescue operations. He has authored or co-authored seven scientific publications, including five in top-quartile (Q1) journals according to Web of Science.

LIST OF PUBLICATIONS

Scientific papers in peer-reviewed journals:

1. Lanča, L., Jakac, K., Calinon S., and Ivić, S., 2025. Ergodic Exploration of Dynamic Distribution. *IEEE Robotics and Automation Letters*, 1 - 8, 2377-3766.
2. Lanča, L., Mališa, M., Jakac, K., and Ivić, S. 2025. Optimal Flight Speed and Height Parameters for Computer Vision Detection in UAV Search. *Drones*, 9(9), 595.
3. Jakac, K., Lanča, L., Sikirica, A. and Ivić, S., 2025. Efficient data-driven flow modeling for accurate passive scalar advection in submesoscale domains. *Applied ocean research*, 162, p.104699.
4. Dumenčić, S., Lanča, L., Jakac, K., and Ivić, S. 2025. Experimental Validation of UAV Search and Detection System in Real Wilderness Environment. *Drones*, 9(7), 473.
5. Jakac, K., Lanča, L., Sikirica, A. and Ivić, S., 2024. Approximation of sea surface velocity field by fitting surrogate two-dimensional flow to scattered measurements. *Applied ocean research*, 153, p.104246.

Conference papers:

1. Lanča, L., Dumenčić, S., Jakac, K. and Ivić, S., 2025, February. Experimental Validation and Robustness Analysis of Ergodic UAV Control. In 2025 11th International Conference on Automation, Robotics, and Applications (ICARA) (pp. 257-261). IEEE.
2. Jakac, K., Lanča, L., Sikirica, A. and Ivić, S., 2025, February. Experimental Validation of Intelligent Ad Hoc System for Sea Surface Velocity Approximation. In 2025 11th International Conference on Automation, Robotics, and Applications (ICARA) (pp. 273-279). IEEE.