

Nature Inspired Guidance in Cluttered Environments for Multiple Vehicles in Real Time

Zeynep Bilgin

Vollständiger Abdruck der von der TUM School of Engineering and Design der
Technischen Universität München zur Erlangung des akademischen Grades einer

Doktorin der Ingenieurwissenschaften (Dr.-Ing.)

genehmigten Dissertation.

Vorsitz:

Prof. Dr.-Ing. Volker Gümmer

Prüfende der Dissertation:

1. Prof. Dr. phil. Ilkay Yavrucuk
2. Prof. Murat Bronz, Ph.D.
3. Prof. Dr.-Ing. Markus Ryll

Die Dissertation wurde am 02.10.2024 bei der Technischen Universität München
eingereicht und durch die TUM School of Engineering and Design am 19.05.2025
angenommen.

Abstract

This dissertation presents a nature-inspired guidance algorithm that addresses the challenges of real-time guidance for multiple vehicles and its integration into the concept of urban air mobility (UAM).

Generating safe trajectories in real-time for multiple vehicles while avoiding obstacles and preventing collisions in cluttered dynamic settings is a challenging problem. Techniques offering rapid solutions for dynamic trajectory generation hold significant value in addressing these challenges. Motivated to address these challenges, this dissertation proposes a guidance algorithm based on potential flow theory. The proposed guidance algorithm, Potential Guidance Flow, offers a vector-based method for guiding multiple aerial vehicles through cluttered environments in real-time. Taking advantage of the properties of the harmonic potential fields, the proposed guidance field does not suffer local minima limitations and guarantees obstacle and collision avoidance.

Potential Guidance Flow's primary strength is its ability to simultaneously guide multiple aerial vehicles through cluttered dynamic environments in real time. Instead of generating trajectories to be followed by the vehicles, the Potential Guidance Flow algorithm relies on calculating desired velocity vectors that guide the vehicles to their respective destinations with obstacle and collision avoidance. Since the guidance vector can be calculated at any position, the proposed method is inherently robust against disturbances like wind gusts. Mimicking the fluid flow around submerged bodies, the resultant paths are smooth and easy to follow even by vehicles with turn radius or minimum speed constraints.

This dissertation also investigates Urban Traffic Management systems. A guidance algorithm integrated with traffic regulations not only improves safety but also increases airspace efficiency. With this motivation, this thesis proposes additional features to assist the integration of the guidance method with the urban air traffic management systems. To this end, corridor models and urban population models are developed and included in the guidance equations.

The research demonstrates the Potential Guidance Flow method's effectiveness, robustness, adaptability, and versatility through a combination of theoretical analysis, simulation studies, and real-world testing. While acknowledging limitations, the dissertation identifies areas for future research, such as guidance in unknown environments, computational efficiency improvements, and integration of vehicle dynamics.

In essence, this thesis thoroughly explores the Potential Guidance Flow algorithm's development and application, highlighting its significant contributions to aerial guidance and urban air mobility. With its efficacy, adaptability, and potential for enhancing safety and efficiency, the Potential Guidance Flow method is a promising solution for autonomous aerial vehicle guidance and urban air traffic management.

Zusammenfassung

Diese Dissertation stellt einen naturinspirierten Guidancealgorithmus vor, der die Herausforderungen der Echtzeitsteuerung für mehrere Luftfahrzeuge sowie deren Integration in das Konzept der Urban Air Mobility (UAM) bewältigt. Die Entwicklung sicherer Trajektorien in Echtzeit, bei der Hindernisse umflogen und Kollisionen in dicht besiedelten und dynamischen Umgebungen vermieden werden müssen, ist eine komplexe Aufgabe. Verfahren, die schnelle Lösungen für die dynamische Trajektorienplanung bieten, sind daher von großem Wert, um diesen Herausforderungen zu begegnen. Vor diesem Hintergrund wird in dieser Arbeit ein Guidancealgorithmus auf Basis der Potentialströmungstheorie vorgestellt. Der sogenannte Potential Guidance Flow“ bietet eine vektorbasierte Methode, um mehrere Luftfahrzeuge in Echtzeit sicher durch komplexe und beengte Umgebungen zu navigieren. Durch die Nutzung der Eigenschaften harmonischer Potentialfelder treten keine Probleme mit lokalen Minima auf, und die Vermeidung von Hindernissen sowie Kollisionen ist garantiert.

Die größte Stärke des Potential Guidance Flow“ liegt in seiner Fähigkeit, mehrere Luftfahrzeuge gleichzeitig und in Echtzeit durch dynamische und herausfordernde Umgebungen zu führen. Anstatt ganze Trajektorien vorab zu generieren, berechnet der Algorithmus kontinuierlich Geschwindigkeitsvektoren, die die Fahrzeuge auf direktem Weg zu ihren Zielen leiten und dabei Hindernissen und Kollisionen ausweichen. Da der Leitvektor an jeder beliebigen Position bestimmt werden kann, erweist sich die Methode als äußerst robust gegenüber äußeren Einflüssen wie beispielsweise Windböen. Ähnlich der Flüssigkeitsströmung um Körper zeichnen sich die resultierenden Pfade durch eine sanfte Krümmung aus, die selbst für Fahrzeuge mit eingeschränkter Manövrierfähigkeit, wie etwa einem Mindestwenderadius oder einer Mindestgeschwindigkeit, leicht zu folgen sind.

Darüber hinaus untersucht die Dissertation auch städtische Verkehrsleitsysteme. Ein Guidancealgorithmus, der bestehende Verkehrsregeln berücksichtigt, erhöht nicht nur die Sicherheit, sondern verbessert auch die Effizienz der Nutzung des verfügbaren Luftraums. Vor diesem Hintergrund werden in der Arbeit zusätzliche Funktionen vorgeschlagen, die die Integration des Leitsystems in städtische Luftverkehrsmanagementsysteme erleichtern. Dazu zählen die Entwicklung von Korridormodellen und urbanen Bevölkerungsmodellen, die in die Leitegleichungen integriert werden. Die in einer Kombination aus theoretischer Analyse, Simulationsstudien und praktischen Tests gewonnenen Forschungsergebnisse zeigen die Effektivität, Robustheit, Flexibilität und Vielseitigkeit der Potential Guidance Flow“-Methode. Trotz einiger Einschränkungen gibt es vielversprechende Bereiche für zukünftige Forschung, wie beispielsweise die Anwendung des Verfahrens in unbekannten Umgebungen, die Verbesserung der Recheneffizienz sowie die Integration fahrzeugspezifischer dynamischer Einschränkungen. Insgesamt bietet die Arbeit

Zusammenfassung

eine umfassende Untersuchung der Entwicklung und Anwendung des Potential Guidance Flow“-Algorithmus und hebt dessen Bedeutung für die Flugführung im Rahmen von Urban Air Mobility hervor. Mit seiner Effizienz, Anpassungsfähigkeit und seinem Potential zur Verbesserung von Sicherheit und Effizienz stellt der Potential Guidance Flow Algorithmus eine vielversprechende Lösung für die autonome Steuerung von Luftfahrzeugen und das urbane Luftverkehrsmanagement dar.

Contents

Abstract	iii
Zusammenfassung	v
Contents	vii
1 Introduction	1
1.1 Motivation	2
1.2 Literature Review	3
1.2.1 Urban Air Traffic Management Efforts	3
1.2.2 Autonomous Guidance Efforts	6
1.2.2.1 Reactive Methods	7
1.2.2.2 Optimization-Based Methods	7
1.2.2.3 Artificial Potential Field Methods	8
1.3 Contributions	10
1.4 Organization	12
2 Harmonic Potential Field Based Guidance	15
2.1 Preliminaries	15
2.1.1 Harmonic Potential Fields	15
2.1.2 Potential Flow Theory	16
2.1.3 Panel Methods	18
2.2 Potential Guidance Flow (PGFlow) Algorithm	20
2.2.1 Guidance Method Development	21
2.2.1.1 Defining a Goal Position	21
2.2.1.2 Collision Avoidance	22
2.2.1.3 Calculating the Flow Velocity	23
2.2.2 Extension to 3D Problems	24
2.3 Enhancements for Safety	25
2.3.1 Defining a Safety Perimeter Around Obstacles	25
2.4 Enhancements for Agile Flight	27
2.4.1 Guidance with Path Propagation	27
2.4.2 Optimal Path Selection	28
2.4.3 Guidance with Simple Position Propagation	29
2.5 Guidance Algorithm	30

CONTENTS

3	Simulation Results	33
3.1	Simulation Setup	33
3.2	Computational Load Study	34
3.2.1	Element Size Sensitivity Study	35
3.2.2	Computation Work Load Sensitivity Study	36
3.3	Single Vehicle Guidance Scenarios	38
3.3.1	Baseline scenario	38
3.3.2	Local Minima Related Tests	39
3.4	Multiple Vehicle Guidance Scenarios	41
3.5	Guidance Scenarios with Agile Flight Enhancements	43
3.5.1	Path Propagation	44
3.5.2	Position Propagation	46
3.6	Conclusions	47
4	Experimental Setup	51
4.1	Indoor Test Facility	51
4.1.1	Tello Platform	52
4.1.2	Explorer Platform	52
4.1.3	Paparazzi Autopilot System	53
4.2	Automatic Flight Controller Integration	54
4.2.1	Automatic Flight Controller	55
4.2.2	Automatic Flight Controller and Guidance Algorithm Integration	56
5	Experimental Validation on VTOL Platforms	59
5.1	Static Goal Positions	59
5.1.1	Single Vehicle	59
5.1.2	Multiple Vehicles	61
5.2	Dynamic Target Tracking	66
5.3	Agile Flight Enhancements	70
5.4	3D Guidance Experiments	74
5.5	Conclusions	77
6	Robustness under Disturbances	79
6.1	Indoor Experiments under Wind Disturbance	79
6.1.1	Experimental Setup	80
6.1.2	The Correction Term for Disturbance Rejection	81
6.1.3	Experiments with a Single Vehicle	82
6.1.4	Experiments with Multiple Vehicles	87
6.2	Outdoor Experiments with Wind Disturbance and Sensor Errors	88
6.2.1	Outdoor Test Facility	88
6.2.2	Quad-Rotor Platform	88
6.2.3	Experiments with a Single Vehicle	90
6.2.4	Experiments with Multiple Vehicles	91
6.3	Conclusions	92

7	Guidance of Fixed-Wing Platform	93
7.1	Experimental Setup	93
7.1.1	Fixed-Wing Platform	93
7.1.2	Automatic Flight Controller Integration	94
7.2	Experiment Results	94
7.3	Conclusions	98
8	Urban Air Mobility Applications	99
8.1	Population Avoidance Application	100
8.1.1	Motivation	100
8.1.2	Modeling Population	100
8.1.3	Experiment Results	101
8.2	Emergency Landing Application	102
8.3	Air Traffic Control Rules Integration	104
8.3.1	Motivation	104
8.3.2	Corridor Models	105
8.3.3	Tactical Deconfliction	107
8.4	Tactical Deconfliction of two Medical Drones	108
8.4.1	Motivation	108
8.4.2	Experiment Results and Discussions	109
8.5	Tactical Deconfliction of an Air Taxi and an Intruder	111
8.5.1	Experiment Results and Discussions	111
8.6	Conclusions	116
9	Conclusions	119
9.1	Summary of Findings and Contributions	119
9.1.1	Impact as a Guidance Method	119
9.1.2	Impact on Urban Air Mobility Applications	122
9.2	Limitations and Future Research	123
9.3	Concluding Remarks	125
	Bibliography	127

1 Introduction

*Natura viam invenit.*¹

Picture a river, flowing freely around rocks and trees, its waters undeterred by obstacles. Now, envision gently releasing a leaf into this stream. The water gracefully carries the leaf along, navigating the path it had created around the obstacles. Like the river carrying the leaf, a guidance algorithm can carry vehicles around obstacles mimicking fluid flow. This thesis work proposes a nature-inspired guidance algorithm called Potential Guidance Flow (PGFlow) for real-time guidance of multiple aerial vehicles in cluttered environments.

Potential Guidance Flow, a guidance method based on vector fields, employs potential flow theory to generate a guidance vector field. Potential flow theory is used for defining the flow of an inviscid and irrotational ideal fluid. Such a flow can be described by a potential function and is often used for defining simple flow elements in the fluid mechanics domain. These elementary flow elements can also be combined to model the fluid flow around an arbitrarily shaped submerged body in the ideal fluid. By definition, the fluid cannot enter into solid objects and the resultant potential field has no local minima. These inherent properties make potential flow theory an ideal candidate for path planning and guidance applications. The guidance vector field generated using potential flow theory not only ensures collision-free guidance around arbitrarily shaped obstacles but also mimics fluid flow to generate smooth and easy-to-follow trajectories. Moreover, the proposed guidance algorithm produces minimal computational burden and is capable of producing real-time guidance vectors for multiple vehicles simultaneously.

The main contribution of this thesis is the development of the PGFlow Algorithm. Although inherent collision avoidance and no-local-minima properties of potential flow make it a suitable candidate for path planning, further modifications in the potential field are necessary to utilize it for the guidance of aerial vehicles. In this thesis, a potential flow field around obstacles is modeled to fit the needs of guidance applications. For this purpose additional flow elements are introduced to serve as goal positions, to avoid collision between vehicles, and to define a safety perimeter around the obstacles. Furthermore, additional features are included in the guidance method to better integrate the method into the existing air traffic control system. An important part of guidance method development is assessing the effectiveness and feasibility of the proposed algorithm. Therefore, this study also includes an extensive simulation and hardware experiment campaign. Hardware experiments are conducted both at indoor and outdoor test facilities on different platforms.

In this chapter, first, the motivation of this study is detailed in Section 1.1. Then, in

¹Nature finds a path.

1 Introduction

Section 1.2 an extensive literature review is presented on path planning and guidance algorithms as well as urban air traffic management. Next, in Section 1.3 the contributions of the thesis are outlined. Finally, the organization of the thesis can be found in Section 1.4.

1.1 Motivation

Advances in technology have accelerated the growth in Unmanned Aerial Vehicle (UAV) operations. The Association for Unmanned Vehicle Systems International (AUVSI) reports that over 4500 Unmanned Aerial Vehicle platforms are in operation, distributed among more than 1350 companies globally as of 2023 [1]. Using Unmanned Aerial System (UAS) for executing tasks that were previously performed by humans is not only safer but also more economical as the expenses associated with manpower is one of the largest cost components in the aviation industry [2]. UAS ended up having a wide range of applications including agriculture [3], surveillance [4], search and rescue [5, 6], mapping [7], inspection [8] and more. The growth encouraged by the advances in automation and electric storage is not limited to above listed areas of UAS applications, it has also initiated the development of the urban aviation industry. Using electric-powered aerial vehicles for the transport of people and goods over short distances in urban areas is a promotion solution to both ground traffic congestion and carbon emissions. According to European Union Aviation Safety Agency (EASA), the European urban air mobility (UAM) market is projected to reach approximately €4.2 billion by 2030 [9]. However, the introduction of this inventive mobility solution comes with its own set of complexities. Creating a sustainable transportation policy and service can be challenging due to its inherently ambiguous boundaries and the involvement of multiple stakeholders with conflicting values[10]. Similarly, in urban air mobility too, achieving safe, sustainable, and efficient deployment requires collaboration among individuals and organizations with diverse perspectives[11]. More specifically, the Urban Air Mobility problem involves various aerial vehicles, differing in designs, sizes, maneuverability, automation levels, and communication capabilities, competing for the same limited urban airspace, which was previously unregulated. This increased demand urges industry, cities, and governments to devise new air traffic management schemes[12]. Considering the expected volume of traffic and the limited airspace, effective traffic management schemes are a must for using the urban airspace capacity efficiently. Integrating fully autonomous vehicles capable of generating collision-free paths not only increases the capacity of urban airspace but also reduces the efforts needed for urban air traffic management[13]. Therefore, the problem of navigation and coordination of multiple autonomous vehicles in urban environments becomes an important issue to be addressed.

The guidance problem in urban airspace involves a large number of vehicles operating in the same limited space. Furthermore, operations can take place in lower than usual altitudes and complex architectural structures need to be taken into account. Consequently, autonomous navigation of aerial vehicles in urban airspace becomes a complex problem involving multiple agents in a cluttered environment. The path planning problem typi-

cally presents as mathematically extensive and computationally demanding, particularly in dynamic environments such as urban areas. Calculating safe and reliable trajectories that lead to the desired destination while avoiding complex architectural structures and dynamic obstacles can quickly turn into a computationally intensive problem, often challenging to resolve even when performed offline. Hence, techniques that offer rapid real-time solutions for dynamic trajectory generation for multiple vehicles, while ensuring obstacle avoidance and collision prevention, hold significant value[14].

This thesis is motivated by both the challenges in the guidance of multiple vehicles in real-time and the integration of the proposed guidance algorithm into Urban Air Mobility (UAM) concept. In order to address these problems a nature-inspired guidance algorithm is proposed. The proposed guidance method generates a guidance vector field with no local minima limitation. Bearing little computational load, the proposed algorithm can guide multiple vehicles in cluttered environments through collision-free paths in real-time. Inspired by the aerodynamics domain, the guidance algorithm developed in this thesis mimics fluid flow and generates easy-to-follow smooth paths for aerial vehicles. Moreover, the proposed algorithm is versatile and can be tailored to fit urban air traffic regulations.

1.2 Literature Review

This thesis introduces a guidance algorithm inspired by nature to tackle the complexities of real-time guidance for multiple vehicles and its integration into the UAM concept. Hence, the study revolves around two primary objectives: the development of a guidance algorithm for UAVs and the incorporation of this algorithm into urban air traffic regulations. The literature survey for this thesis work is similarly divided into two parts. The first part inspects the existing traffic management methods and regulations in UAM. The second part focuses on the state of the art of guidance and path-planning methods.

1.2.1 Urban Air Traffic Management Efforts

The rapid progress in drone technology requires major changes in how airspace is managed, moving away from traditional aircraft towards possibly hundreds of thousands of interconnected drones offering various services, such as urban operations. Existing urban air space management initiatives argue that new air space classes should be introduced and managed by Air Traffic Control (ATC). An increased structure is necessary, especially in densely populated areas and a less structured airspace can be offered as technology levels increase[15, 16, 17].

The NASA UAM Coordination and Assessment Team (UCAT) formulated the UAM Maturity Level (UML) scale to categorize the significant stages anticipated in the development of a UAM transportation system[18]. UML-1 is the pre-operational stage where only operational demonstrations take place in limited environments. UML-2 is marked by specific characteristics: low traffic density, operations of low complexity, and dependence on assistive automation. UML-3 is characterized by relatively low average traffic

1 Introduction

density, operations of medium complexity, and reliance on comprehensive safety automation. This stage is seen as transitional, where technologies necessary for supporting more advanced and scalable operations are gradually introduced. UML-4 represents a stage of UAM characterized by moderate traffic levels and operational complexity, relying on collaborative automation. Anticipated regulatory changes are expected to enhance the reliability and scalability of UAM transportation, enabling practical use in various US metropolitan areas and making it accessible to a significant portion of the public for travel between high-density locations. UML-5 entails high-density and intricate operations, depending on highly-integrated, automated networks. In contrast to UML-4, the system features an exponential increase in aircraft and ports within a specific metropolitan area. Finally, UML-6 signifies the completion of the UAM build-out, with further growth correlating with population increases. However, due to the long-term horizon and evolving technology, the exact characteristics of this mature UAM system remain uncertain[18].

On the European front, in their report titled "European ATM Master Plan: Roadmap for the safe integration of drones into all classes of airspace" SESAR Joint Undertaking[19] outlines the development and implementation of U-space services, aimed at ensuring the safe and efficient integration of drone operations within airspace shared with manned aviation. U-space serves as a framework accommodating diverse drone missions and users across all airspace categories and environments[20]. It consists of four phases: U1 establishes foundational services with a focus on safety and identification. U2 builds upon U1 and expands operations into controlled airspace and interfaces with Air Traffic Management (ATM) systems. U3 expands further on U2, introducing new technologies and capabilities to enable more complex drone missions, particularly in dense urban areas. Finally, U4 achieves full operational capability of U-Space integrated seamlessly with ATM/ATC systems through high levels of automation[19].

The urban airspace can be structured by the introduction of layers, lanes, tubes, or corridors[21, 22, 23, 24]. In the report "UAM Concept of Operations"[21] UAM Corridors are defined as designated airspace areas intended to accommodate safe UAM operations. Initially, UAM corridors are subject to existing regulatory frameworks (e.g., Visual Flight Rules, Instrument Flight Rules) to satisfy their operational needs. As UAM operations increase, corridors may not require traditional ATC services and could operate independently when advantageous. [21] further suggests, that UAM corridors initially supporting point-to-point operations may evolve into more complex networks connecting multiple points, such as vertiports [21]. Other than segregation by corridors, Le Tallec et. al. propose separating aircraft by layers at unregulated lower altitudes[23]. The layer approach is proposed for organizing air traffic below 500 feet above ground level. The airspace is structured into five layers to ensure safety within the airspace. Eastbound and Westbound layers are designated for aerial vehicles flying in eastbound and westbound directions respectively. A 50 feet Thick Buffer Zone separates Eastbound and Westbound layers. There are also two additional buffer zones each 50 feet thick to separate the traffic from Manned Traffic and the ground[23].

Alternatively, air traffic can also be organized by sky lanes[22]. Sky-lane is a conceptual framework for a sky-highway system, in essence expanding the traditional road network

into the skies. It proposes a layered structure where vehicles travel in designated lanes, similar to the way cars travel on roads. Each layer consists of one or two-way lanes aligned parallel to each other. The Sky-lane approach can be divided into variants A1, A2, and A3 according to different intersection and interchange designs. A1 merges intersecting strips in the same layer, A2 stacks crossed strips without sharing layers, and A3 alternates lane directions in neighboring layers[22].

Sky-tubes is an airspace design concept similar to sky-lanes where vehicles travel within tubes instead of lanes[22]. Different from sky-lanes, there are no reference lines to follow. Vehicles can move freely within the tube, as long as they correspond to the traffic direction. There are two design variants proposed for the intersection of sky tubs. For the first variant, a traffic control system is required at every intersection. Traffic control is centralized to ensure separation among vehicles. In the second version, traffic management is accomplished by allotting transition way-points and time frames for vehicles as they cross the boundaries of an intersection[22].

Structured traffic management strategies offer benefits by utilizing established traffic control systems, thereby enabling the swift integration of urban air mobility. On the other hand, as the volume of the operations grows enforcing a structured ATM may prevent meeting the demand for the airspace. In both the United States and Europe, the primary operational constraint on airspace capacity is the workload of air traffic controllers[25, 26, 27]. Study shows that less structured airspace not only has increased capacity but also safety [28, 13, 29]. Hence, some studies propose separation by technology level and offer less structured airspace to vehicles with high autonomy level [30, 31, 32].

In their report titled "Revising the Airspace Model for the Safe Integration of Small Unmanned Aircraft Systems", Amazon[30] proposes less structured airspace that separates vehicles by technology level. In this proposed model, airspace below 500 feet is structured into distinct zones. Low-Speed Localized Traffic Area is the air zone below 200 feet, reserved for non-transit operations Sense and Avoid (SAA) technology. Moreover, the airspace over heavily populated areas in this zone is inaccessible to lesser-equipped vehicles. The High-Speed Transit Space is located between 200 feet and 400 feet. It is designated for well-equipped vehicles meeting relevant performance standards and rules. There is a 100 feet No Fly Zone between 400 feet to 500 feet to separate UAS traffic from Manned Traffic. The model aims to enhance safety by determining airspace access based on vehicle capability[30].

Similarly, in their report titled "DLR Blueprint Concept for Urban Airspace Integration", German Aerospace Center (DLR) proposes segmenting the airspace into cells with varying requirements allocating the airspace to users based on their mission objectives and performance parameters such as navigation, communication, and the capability of being detected and of detecting other airspace users. Airspace users are then modeled by an ellipsoid based on their performance. As the performance of the aircraft increases the assigned ellipsoid becomes smaller. Airspace allocation is based on density, resulting in more freedom of operation at low density and more regulated access at high density[31]. To summarize, studies agree that a less structured airspace has more capacity than a structured and controlled airspace. However, the transition from structured airspace to

1 Introduction

unstructured airspace will have to be gradual. Moreover, for a successful transition, the vehicle technology level has to be increased so that aerial vehicles are able to detect and avoid obstacles and other agents in the airspace autonomously. Therefore, designing fully autonomous vehicles that can generate collision-free and robust paths in dynamic environments is crucial to increasing the efficiency of urban air mobility. However, until vehicles reach this level of autonomy the urban air space has to be managed in a more structured manner. Given this perspective, a guidance algorithm that seamlessly aligns with the more constrained concept of operations in the initial integration stages, while also exhibiting the capacity to autonomously generate collision-free trajectories for when operations transition into less regulated traffic management schemes, would be immensely valuable and visionary.

1.2.2 Autonomous Guidance Efforts

UAM has captured considerable attention from aircraft manufacturers, air navigation service providers, and governments in recent years. Ensuring the avoidance of conflicts among urban aircraft is essential for UAM traffic safety, which is vital for enabling large-scale UAM operations. Recent studies addressed path planning problems in urban environments [33, 34, 35, 36].

In their paper Ahamed et. al. [33] introduce a multilevel framework to address the guidance problem of UAS for on-demand delivery in urban airspace. A prediction-based zone partitioning algorithm is proposed to reduce high-density traffic encounters. Numerical potential fields are considered for global path planning to ensure safe clearance from obstacles and traffic. Similarly, Huang et. al. [36] also proposes an Artificial Potential Field (APF) based algorithm for UAV trajectory planning in complex urban environments. The proposed method is compared with existing heuristic methods through simulations in terms of efficiency and trajectory smoothness. Zhang et. al. [34] introduces a fusion scheme for autonomous drone path planning in urban airspace. The proposed method utilizes a 3D voxel-based path planning model to generate collision-free paths taking static obstacles into account. Additionally, a Markov Decision Process (MDP)-based method is utilized to prevent mid-air collisions in real-time. Dai. et. al. [35] also addresses the issue of conflict-free path planning for UAM operations. They develop a four-dimensional airspace management concept and a Conflict-Free A* (CFA*) algorithm to plan paths for static and dynamic obstacle avoidance.

Although UAM concept attracts the attention of many researchers it still can be considered a relatively new area of study and consequently, path planning applications in this area are limited. On the other hand, guidance for UAV is well-studied in the literature. Therefore, to have a better insight into path planning schemes literature review is conducted irrespective of operation environment.

Classical methods for autonomous vehicle path planning can be categorized into three sub-groups: reactive methods, optimization problem-based methods, and artificial potential field-based methods.

1.2.2.1 Reactive Methods

Reactive methods in path planning based their decisions solely on current sensor information and immediate surroundings. Not explicitly building a map or considering long-term goals reactive methods are suitable for dynamic environments. These methods search safe avoidance maneuvers by considering the velocity and/or acceleration of the obstacles [37, 38, 39, 40, 41, 42, 43]. Fox et. al. [37] propose the dynamic window approach for reactive collision avoidance in mobile robots. The proposed method searches for control commands directly in velocity space, incorporating robot dynamics. The search space is restricted to the dynamic window—velocities reachable within a short time interval. The approach maximizes an objective function, combining progress toward the goal, forward velocity, and distance to the next obstacle. Similarly, Fiorini and Shiller [38] propose a method for robot motion planning based on avoidance maneuvers in velocity space to evade static and moving obstacles. Feasibility of the maneuvers is ensured by intersecting avoidance velocities with admissible velocities based on acceleration constraints. Petti et. al. [39] demonstrate the application of the Partial Motion Planning method to a car-like system navigating a dynamic environment. In their work Chan et. al. [40] address the challenge of navigation of under-actuated autonomous robots in environments with moving obstacles. Chakravarthy and Ghose [41] address the guidance problem on dynamic environments cluttered with arbitrarily shaped obstacles. They propose a Collision Cone approach where collision cones are generated for the vehicle in the immediate surroundings to avoid obstacles [41, 42, 43].

1.2.2.2 Optimization-Based Methods

The second approach in classical path planning is formulating the problem as a large optimization problem with constraints. In this approach the guidance problem needs to be represented by a mathematical model and a cost function to be minimized has to be defined. Depending on the proposed cost function the method can optimize time, distance, energy consumption, number of maneuvers, etc. Furthermore, to fit the needs of the problem better some constraints may be introduced into the cost function. Possible cost functions include the dynamics of the vehicle and performance limitations. Air traffic regulations and other environmental restrictions can also be represented in the equations as constraints. Schulman et. al. [44] proposes an optimization-based approach for robotic motion planning. The proposed algorithm can be applied to robot arms, full-body robots, and humanoid robots. The algorithm incorporates a continuous-time safety constraint formulation and utilizes sequential convex optimization with penalty coefficients adjusted in an outer loop to penalize collisions effectively. In UAV trajectory planning domain Liu et. al. [45] introduce a search-based planning technique aimed at computing dynamically feasible trajectories for a quad-rotor in cluttered environments. The proposed path-planning method focuses on discovering minimum-time trajectories by surveying the map using a series of short-duration motion primitives. These primitives are derived from solving an optimal control problem, creating a finite

1 Introduction

lattice within the state space that can be explored via a graph-search algorithm. Solving Linear Quadratic Minimum Time problems dynamically feasible trajectories optimal within the discretized space are generated. Tordesillas and How [46] present a trajectory planner for UAVs named MADER. In the proposed planner every interval of the trajectories is represented by an outer polyhedral and collision-free trajectories are generated by solving an optimization problem. Committed trajectories of other agents are also included in the optimization problem as constraints to guarantee collision avoidance. Resultant flight distances are shorter than that of centralized approaches, and shorter total times than synchronous decentralized approaches are achieved. Following MADER, Tordesillas et. al. [47] propose FASTER (Fast and Safe Trajectory Planner) for fast path planning in unknown environments. FASTER achieves high-speed trajectories by allowing the local planner to optimize in both known and unknown spaces. Safety is maintained by ensuring there's always a secure backup trajectory in the known space. Researchers have also employed optimization-based approaches in path planning problems in UAM[48, 49, 50, 51]. Teshnizi et. al. [48] propose a method for centralized path planning of multiple aircraft following low-flying rules. The problem is formulated as a unified nonlinear constraint optimization task, minimizing time and control while considering maximum velocity and power constraints. A simplified mathematical model is used for representing the aircraft dynamics. Obstacles either stationary or moving at constant speeds are represented as spheres. In their article Chen et. al. [49] present an enhanced approach to UAV path planning. By introducing an additional control force, the APF method is transformed into a constrained optimization problem. The functional optimization method is then employed to convert this problem into an optimal control problem. The path planning problem is subsequently solved using this optimal control method yielding shorter and smoother paths compared to traditional APF methods. Ge and Topcu [50] develop a hierarchical motion planning system for on-demand passenger-carrying air vehicles, considering flight efficiency, safety, and passenger comfort. First, city maps are transformed into grid worlds estimating transfer times among cells with consideration of no-fly zones and air traffic congestion. A* search is conducted for time-optimal path planning. Trajectory tracking and energy minimization are ensured by model predictive control. Predeep and Wei[51] address the path planning problem for multi-rotor Electric Vertical Take-Off and Landing (eVTOL) aircraft. A fixed-final-time multi-phase optimal control problem formulation focusing on energy consumption as the performance index is proposed. The formulation enables vehicles to meet assigned arrival times and achieve energy-efficient trajectories.

1.2.2.3 Artificial Potential Field Methods

The third approach in UAV guidance efforts is employing artificial potential functions. In this approach, an artificial potential field consisting of attractive and repulsive potential functions is generated. Repulsive potential functions are used for preventing collisions and attractive potential leads the vehicle towards goal position[52]. In other words, The APF is a tool to establish functional relationships between the vehicle and surround-

ing objects[53]. The potential field approach for path planning finds application across a range of fields, spanning from wheeled ground vehicles to unmanned aerial vehicles [54, 55, 56, 49, 57, 58]. Its simplicity and ease of implementation represent the main advantages of this approach. Moreover, the effectiveness of potential field-based methods can be further enhanced by combining them with other classical or heuristic approaches [59, 60, 61, 62]. Nazarahari et. al. [60] present a hybrid approach for path planning of multiple mobile robots using an APF with an enhanced Genetic Algorithm. In the proposed method, APF finds feasible paths between the start and goal positions in a grid map and an enhanced Genetic Algorithm (EGA) is utilized to find the optimal path between the start and goal positions. Tsourveloudis et. al. [61] combine an electrostatic potential field (EPF) path planner with a two-layered fuzzy logic inference engine. First, the environment is mapped into a resistor network to generate an electrostatic potential field then collision avoidance with dynamic obstacles while following the trajectory is performed by layers of the fuzzy logic inference engine. Hickling et. al. [62] address the UAV path planning problem by utilizing Deep Reinforcement Learning (DRL) scheme. The training of the agent is integrated with an APF to enhance training efficiency and obstacle avoidance performance.

The artificial potential field method suffers from two major drawbacks. The first concern is that the vehicle might become trapped in a local minimum in the potential field, while the second is that obstacles near the goal position may prevent the vehicle from reaching its destination due to their repulsive force[63, 64]. One possible approach to overcome these drawbacks is utilizing harmonic potential fields, a special category of potential fields. Harmonic potential fields are defined as potential fields formed by a potential function that satisfies Laplace's equation. One of the advantageous properties of harmonic potential fields is that they do not have local minima by definition [65, 66]. For any initial point and goal position within a connected domain, there exists a unique path between them which corresponds to the maximum gradient path of the harmonic function originating from the initial point and terminating at the goal point[67]. Harmonic potential fields used for generating smooth and well-behaved robot trajectory in static and dynamic environments for various platforms[68, 69, 70, 67]. In his work, Masoud [68] considers path planning at the servo-level of a mobile robot and utilizes harmonic potential fields for generating provably correct, constrained, well-behaved trajectory and control signal for a rigid, nonholonomic robot in a stationary, cluttered environment. Daily and Bevely [70] propose a method to represent complex-shaped obstacles in harmonic potential fields for path planning for a high-speed ground vehicle. In their article, Garrido et. al. [67] use the Finite Elements Method to represent complex-shaped obstacles and generate a harmonic potential field for guidance in a known environment. Harmonic potential fields are often encountered in physical phenomena in nature such as electromagnetism or fluid mechanics. The concept of mimicking the behavior of gases and fluids has a wide range of applications. Pimenta et. al. [71] concentrate on developing feedback control laws for swarms of robots, utilizing models derived from fluid dynamics. Applying an incompressible fluid model a pattern generation task is solved. More specifically, decentralized controllers are created to force the robots in a swarm to mimic the behavior of fluid particles. Similarly, in their work, Song et. al. developed

1 Introduction

a multi-vehicle flocking and guidance control scheme that gives fluid properties to the vehicle swarm of autonomous underwater vehicles[72]. In potential flow-based guidance models, analytical solutions can be obtained through the combination of elementary potentials like uniform flow, sinks, and sources. Nevertheless, these solutions are limited to generating flow fields around relatively simple geometries, such as circles and ellipses[73, 74]. In their consecutive works Xiao et. al. [75] and Honglun et. al. [58] use this principle to develop a method for planning the three-dimensional path for low-flying UAV in complex terrain. In order to solve the fluid equations the terrains transformed into cylindrical, conical, and spherical obstacles and their combinations. In the context of guidance especially for urban air mobility obstacles can have complex shapes and an analytically calculated fluid potential field, using only circular representations, may become insufficient in cluttered environments. Alternatively, extending a global path planning approach using harmonic potential fields to obstacles of arbitrary shape can be achieved by employing a numerical method borrowed from the aerodynamics domain, known as the panel method [74, 76]. Kim and Khosla [66] use the panel method to represent arbitrarily shaped obstacles and generate a harmonic potential over the whole space for obstacle avoidance of a bar-shaped mobile robot and a three-degree-of-freedom planar redundant manipulator. The authors show that the use of harmonic functions completely eliminates local minima. Similarly, Zhang and Valavanis [77] propose collision-free path planning through a potential panel method in an obstacle-filled 3-D environment. In his book titled "Autonomous Robots: Modeling, Path Planning, and Control", Fahimi [69] details how to utilize the panel method and harmonic potential field generated by potential fluid flow around complex shaped obstacles as a path planning tool. Based on the method proposed in Fahimi's book, Velagic et. al. [78] propose a path planner which is based on the enhanced robust panel method for the mobile robot motion in partially unknown static and dynamic environments. In their publication, Cruz et al. [79] tackle the task of developing an obstacle avoidance system for fixed-wing aircraft. The authors generate a potential field algorithm using the panel method. The paper relies on hardware-in-the-loop simulations to demonstrate 3D obstacle avoidance capabilities. Uzo et. al. [14] utilize panel method-based guidance for collaborative target tracking for swarming Micro Air Vehicles (MAV)s in the urban environment. Authors point out that MAVs have limited computational power; thus, an algorithm that can generate smooth collision-free paths for multiple vehicles in one step without too much computational requirement is valuable making panel method-based harmonic potential field approach an ideal candidate for guidance of multiple aerial vehicles in cluttered urban environments.

1.3 Contributions

This thesis work proposes a nature-inspired guidance algorithm called PGFlow for real-time guidance of multiple aerial vehicles in cluttered environments. The thesis aims to address both the challenges in the guidance of multiple vehicles in real-time and the

integration of the proposed guidance algorithm into UAM concept. The main contributions of this thesis are as follows:

- Inspired by the fluid flow around obstacles a vector field-based guidance method named PGFlow is proposed. The proposed guidance algorithm can generate guidance vectors for multiple aerial vehicles in a cluttered environment in real-time. The potential field of the fluid flow is calculated using the panel method, a numerical tool borrowed from the aerodynamics domain. Additional flow elements are included in conventional panel method equations to satisfy the needs of the guidance problem. To represent goal positions point sink elements are introduced and to prevent collision with other vehicles a point source element is assigned to each vehicle.
- An additional safety feature named, *safety source element* is introduced into guidance vector field calculations. Although trajectories tangent to obstacle boundaries are valid solutions for fluid dynamics, it is not a feasible solution for path planning of aerial vehicles. To generate safer paths that are not tangent to obstacles an additional source element that travels with the vehicle itself is introduced. This novel *safety source element* can generate trajectories as far away from obstacles as desired while still guaranteeing convergence to the goal position. The main advantage of employing the *safety source element* is its flexibility. The safety perimeter defined by the *safety source element* can be adjusted even online.
- The proposed guidance algorithm is integrated with an automatic flight controller. The output of the modified potential flow equations is flow velocity at a given position. This flow velocity is scaled and bounded taking arena size and vehicle dynamics into account to obtain a guidance vector. The guidance vector is then fed to the vehicle as the desired velocity vector. The desired velocity is tracked using Incremental Nonlinear Dynamic Inversion (INDI) controller. For fixed-wing applications, an alternative heading tracking approach is employed where the guidance vector is normalized and fed to the controller as the desired heading.
- To facilitate agile flight in cluttered environments, the proposed guidance algorithm is enhanced with a path projection and optimal path selection algorithm. Using the PGFlow method, alternative paths are projected into the future. Among the alternative paths, the path with the least curvature is selected and the corresponding guidance vector is fed to the vehicle for more agile flight.
- As an alternative to the path projection method, the position projection method is developed. In position projection method future position of the vehicle is predicted using the current velocity of the vehicle. Then, if any collisions are expected, an avoidance maneuver is initiated. With this method, smoother paths can be achieved with minimal computational load.
- Target tracking feature is introduced by assigning a dynamic point sink element to a moving target.

1 Introduction

- Existing 2D potential flow equations are altered to extend the guidance method to 3D while maintaining its low computational load property and real time solution capability.
- The proposed guidance algorithm is modified to fit the needs of urban air mobility applications and urban air traffic regulations.
- Crowd avoidance feature is integrated into the guidance algorithm. Vehicles are enabled to avoid crowded regions within the city while traveling to their destination. With this feature, the safety of operations in urban airspace is aimed to be improved. Furthermore, population avoidance also reduces visual pollution and can enhance positive public perception of urban air mobility services.
- An emergency landing mode is included in the guidance algorithm to further improve safety. In case of a failure, the defective vehicle is guided to the nearest emergency landing pad while other vehicles in the vicinity avoid the faulty vehicle.
- To better integrate the guidance algorithm with existing air traffic regulations, air traffic corridor models in accordance with the existing concept of operations are developed to enable a vehicle to follow a preassigned path with minimal deviation while preserving the collision avoidance property.
- A new vortex element is introduced into flow velocity calculations to guarantee vehicles alter their course following the current air traffic management rules. That is, when two vehicles heading in opposite directions encounter, they both alter their course to the right.
- Effectiveness and feasibility of the proposed guidance algorithm are evaluated in a simulation environment in Python.
- Performance of the proposed guidance method is verified through an extensive hardware test campaign. The experiments are conducted in The Toulouse Occitanie Drone Flight Area located in France through an international collaboration effort between the Technical University of Munich (TUM) in Germany and Ecole Nationale de l'Aviation Civile (ENAC) in France.
- Robustness under disturbances is assessed under wind disturbances and sensor errors through hardware experiments conducted under windy conditions and with sensor errors.

1.4 Organization

This thesis is structured to provide a comprehensive understanding of the research conducted, starting with Chapter 1: Introduction. In Section 1.1 in this chapter, the motivation behind the research is elaborated, outlining the key factors driving the development of the proposed guidance method. Additionally, a thorough literature review is presented

in Section 1.2, surveying existing works in both UAV guidance and path planning and UAM and Urban Air Traffic Management to establish context and significance of the proposed guidance algorithm. The chapter further discusses the contributions made by this thesis work in Section 1.3. Finally, in Section 1.4 the organization of the thesis is presented.

Following Chapter 1: Introduction, Chapter 2: Harmonic Potential Field Based Guidance discusses the methodological framework employed in the study. It begins with Section 2.1 where the foundational concepts and theories essential for grasping the research methodology are presented. Subsequently, in Section 2.2, the Potential Guidance Flow (PGFlow) Algorithm is explained, offering a detailed insight into its development, safety enhancements, and extensions for agile and 3D flight.

Moving forward, Chapter 3: Simulation Results provides a presentation of the simulations conducted as part of the evaluation of the proposed guidance algorithm. It starts with a description of the simulation setup in Section 3.1. The chapter then proceeds with a computational load study given in Section 3.2, which investigates the computational demands associated with the proposed methodology through sensitivity analyses. The chapter continues to present the results and analyses derived from simulations involving both single and multiple vehicles to set the baseline performance of the proposed method in Section 3.3 and Section 3.4. Simulation results for agile flight enhancements are delivered in Section 3.5.

This thesis involves an extensive experiment campaign to test the effectiveness of the proposed method. In the following chapters experimental results are presented. Starting with Chapter 4: Experimental Setup where the setup utilized in hardware experiments is presented focusing on the practical components and capacities of indoor and outdoor testing facilities. This chapter lays the groundwork for subsequent experimental validations and evaluations discussed in later chapters.

Chapter 5: Experimental Validation on Hovering Platforms presents the experimental validations conducted on hovering platforms. Section 5.1 experiments with multiple quad-rotors are conducted. In these initial experiments, each quad-rotor has its own static goal position and they try to reach their respective destinations while avoiding obstacles and collision with other vehicles in the arena. In Section 5.2, instead of static goal positions, a dynamic target is introduced. Vehicles are tasked with tracking the target while avoiding obstacles and collisions with each other. Further hardware experiments are conducted for agile flight enhancements and 3D guidance in Section 5.3 and Section 5.4 respectively. The chapter concludes with insights drawn from the experimental results.

Chapter 6: Robustness under Disturbances assesses the robustness of the proposed algorithm under wind disturbances and sensor errors, through indoor and outdoor experiments. The chapter discusses the experimental setups and presents conclusions drawn from the evaluation.

Chapter 7: Guidance of Fixed-Wing Platforms consider the fixed-wing application of the proposed guidance algorithm. The fixed-wing platform is challenging as it has a minimum speed and minimum turn radius constraint. The fixed-wing platform is introduced in Section 7.1. In Section 7.1.2, automatic flight controller integration for a fixed-wing

1 Introduction

platform is detailed. The chapter presents experimental results and delivers conclusions derived from fixed-wing experiments.

Chapter 8: Urban Air Mobility Applications examines the integration of air traffic control regulations into the proposed algorithm. Population avoidance and emergency landing scenarios are explored in Section 8.1 and Section 8.2. Integration of air traffic management rules is detailed in Section 8.3. To better integrate the guidance algorithm with existing air traffic regulations, air traffic corridor models in accordance with existing concepts of operations are developed in Section 8.3.2. Tactical deconfliction scenarios are presented in Section 8.5.1 and Section 8.4.

Finally, Chapter 9: Conclusions summarizes the key findings, contributions, and implications of the thesis work.

Additionally, the thesis includes a Bibliography and an Appendix providing supplementary information.

2 Harmonic Potential Field Based Guidance

In this thesis, a nature inspired guidance algorithm called PGFlow is developed. PGFlow is a vector field based guidance method that uses potential flow theory to calculate the guidance vectors. Resultant vector field not only provides collision free guidance around complex shaped obstacles but also mimics fluid flow to yield smooth and well behaved trajectories.

In this chapter the development of the PGFlow method is discussed in detail. The chapter is organized as follows; first in Section 2.1 preliminaries for potential flow theory is reestablished and the numerical tool called panel method that is used for potential field calculations is inspected. Section 2.2 discusses how potential flow theory can be applied in guidance problem and development of PGFlow algorithm. PGFlow algorithm is developed for 2D applications. Section 2.2.2 explains the extension of the developed 2D method into 3D problems. Next, safety enhancements and enhancements for agile flight are discussed in Section 2.3 and Section 2.4 respectively. Finally, Section 2.5 details how all components of the guidance algorithm come together.

2.1 Preliminaries

The guidance method proposed in this thesis is a potential field based method. Proposed PGFlow method utilizes harmonic potential fields, a sub category of potential fields, for guidance. Harmonic potential fields have multiple properties that are advantageous for guidance applications. Furthermore, harmonic potential fields are also used in fluid mechanics and aerodynamics domains to describe the flow of an incompressible, irrotational, ideal fluid [76, 74].

This section gives preliminary information on harmonic potential fields and their applications in fluid mechanics domain. Moreover, a numerical tool named panel method is also discussed in this section.

2.1.1 Harmonic Potential Fields

Let ϕ be a twice continuously differentiable function in $\mathbb{R}^n$ and let U be an open subset of $\mathbb{R}^n$. Then, $\phi : U \rightarrow \mathbb{R}$ is called a harmonic function if ϕ satisfies Laplace's Equation (Eqn. 2.1) on U [80]:

$$\nabla^2 \phi = 0 \tag{2.1}$$

The word "harmonic" was used to describe the motion of a point on a vibrating string. Harmonic motion on a string can be described with sines and cosines which are sometimes referred as "harmonics". By early 1900s, the term "harmonics" was applied to any

solution of Laplace's Equation [80].

There are several properties of harmonic functions that make them suitable for guidance applications. Following [80, 65, 66], some key properties and their proofs are listed below:

Property 1: Superposition

Harmonic functions satisfy the principle of superposition because of the linearity of the Laplace equation [80, 65, 66].

Let ϕ_1 and ϕ_2 be harmonic functions in $\mathbb{R}^n$. Then their linear combination, $\phi_1 + \phi_2$ is also a harmonic function and satisfies Laplace Equation.

Property 2: Mean Value Property

Let $\phi(x, y) \in \mathbb{R}^2$ be a potential function which is harmonic inside a circle with radius r , centered at (x_0, y_0) . Then, mean value property suggests that;

$$\phi(x_0, y_0) = \frac{1}{2\pi} \oint \phi(x_0 + r \cos \theta, y_0 + r \sin \theta) d\theta$$

That is, value of a harmonic function at the center of a circle is equal to the average of its values along the circle boundary [65, 66].

Property 3: The Maximum Principle

The maximum of a non-constant harmonic function occurs on the boundary [65, 66].

These properties make harmonic potential fields a promising candidate for path planning and guidance applications. The superposition property enables combining multiple potential fields to achieve enhanced path planning performance. The mean value property enables fast and easy calculation of the potential value at a given position. Finally, thanks to the maximum principle, the harmonic potential field does not suffer from local minima constraints.

2.1.2 Potential Flow Theory

Potential flow theory is commonly used in fluid dynamics to describe the flow of an incompressible, irrotational, ideal fluid [76, 74]. The flow is characterized by a scalar field called the velocity potential and governed by Laplace's Equation given in Equation 2.1. Since, flow velocity potential ϕ satisfies the Laplace's Equation, it is specified as a harmonic potential function and satisfies the aforementioned properties of harmonic functions.

It is possible to model the flow around a submerged body in an ideal fluid using potential flow theory. Let $\phi \in \mathbb{R}^2$ be a flow velocity potential function that satisfies Equation 2.1. As the fluid cannot cross the boundaries of the submerged body, the normal component

of the velocity on the surface of the body in the fluid should be zero as given in Equation 2.2 [74].

$$\nabla\phi \cdot \mathbf{n} = 0 \quad (2.2)$$

Here, $\mathbf{n}$ is the normal vector on the surface. Moreover, the disturbance created by the motion decays as it gets further away from the submerged body as given in Equation 2.3 [74].

$$\lim_{r \rightarrow \infty} (\nabla\phi - \mathbf{v}) = 0 \quad (2.3)$$

with $\mathbf{r} = (x, y)$ and $\mathbf{v}$ being the relative velocity between undisturbed fluid and the body. Solution to Equation 2.1 can be obtained by distributing elementary solutions along the surface of the submerged body, i.e. problem boundaries [74]. Elementary solutions, also known as singularity elements, inherently satisfy the boundary condition given in Equation 2.3. To obtain the general solution, appropriate singularity elements that satisfy the Equation 2.2 are placed on the boundaries and integrated over the surface of the submerged body.

In this thesis, following elementary solutions are employed for modelling the proposed guidance vector field:

Point Source/Sink in 2D

Following [81], let $\mathcal{D}$ be a region in $\mathbb{R}^2$ containing an ideal fluid and $\mathbf{P} = (x, y)$ be an arbitrary point in $\mathcal{D}$. Let $\mathbf{V} = (u, v)$ denote the instantaneous velocity of the fluid at any arbitrary point $\mathbf{P} = (x, y) \in \mathcal{D}$. Then, the velocity potential of a point source element placed at point $\mathbf{P}_0 = (x_0, y_0)$ in Cartesian coordinates at the point $\mathbf{P} = (x, y)$ is given in Equation 2.4 [74].

$$\phi = \frac{\sigma}{2\pi} \ln \sqrt{(x - x_0)^2 + (y - y_0)^2} \quad (2.4)$$

Here, $\sigma > 0$ is the source strength. If σ is taken negative then the elementary flow element becomes a point sink. Flow velocity, $\mathbf{V} = (u, v)$, at any point $\mathbf{P} = (x, y)$ can be obtained by differentiating Equation 2.4. Corresponding velocity components u and v are given in Equations 2.5 and 2.6 [74].

$$u = \frac{\partial\phi}{\partial x} = \frac{\sigma}{2\pi} \frac{x - x_0}{(x - x_0)^2 + (y - y_0)^2} \quad (2.5)$$

$$v = \frac{\partial\phi}{\partial y} = \frac{\sigma}{2\pi} \frac{y - y_0}{(x - x_0)^2 + (y - y_0)^2} \quad (2.6)$$

Point Vortex in 2D

Similarly, consider the arbitrary point $\mathbf{P} = (x, y) \in \mathcal{D}$ and the corresponding instantaneous velocity $\mathbf{V} = (u, v)$. Then, the velocity potential of a point vortex element placed at point $\mathbf{P}_0 = (x_0, y_0)$ in Cartesian coordinates at the point $\mathbf{P} = (x, y)$ is given in Equation 2.7 [74].

$$\phi = \frac{-\Gamma}{2\pi} \arctan \frac{y - y_0}{x - x_0} \quad (2.7)$$

2 Harmonic Potential Field Based Guidance

Here, $\Gamma > 0$ is the vortex strength. If Γ is taken negative then the rotation direction of the vortex is reversed. Flow velocity, $\mathbf{V} = (u, v)$, at any point $\mathbf{P} = (x, y)$ can be obtained by differentiating Equation 2.7. Corresponding velocity components are given in Equations 2.8 and 2.9 [74].

$$u = \frac{\partial \phi}{\partial x} = \frac{\Gamma}{2\pi} \frac{y - y_0}{(x - x_0)^2 + (y - y_0)^2} \quad (2.8)$$

$$v = \frac{\partial \phi}{\partial y} = \frac{-\Gamma}{2\pi} \frac{x - x_0}{(x - x_0)^2 + (y - y_0)^2} \quad (2.9)$$

2.1.3 Panel Methods

To model the flow around a submerged body in an ideal fluid, Laplace's Equation (Equation 2.1) needs to be solved. The general solution can be obtained by placing appropriate singularity elements that satisfy the Equation 2.2 on the object surface and integrating them along the boundary [74]. Analytical solutions are available for simple geometries such as circles and ellipses[76]. For more complex shapes, the fluid flow can be modelled using a numerical method named *Panel Method*.

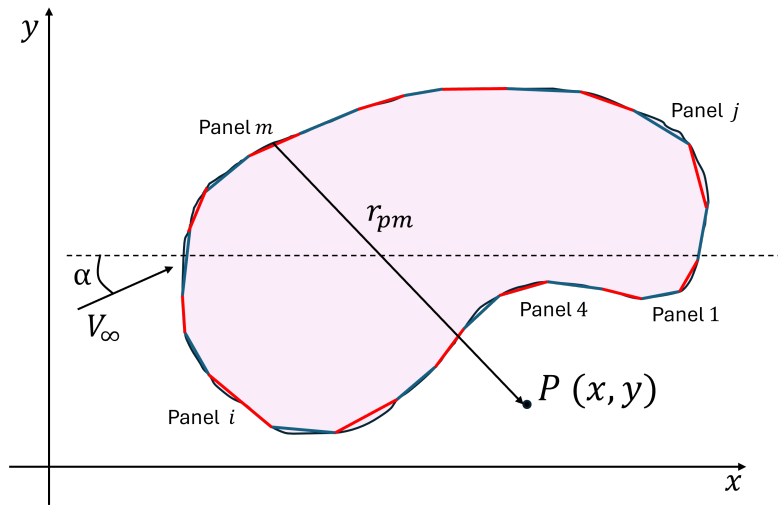


Figure 2.1: Panel distribution over the surface of a body of arbitrary shape submerged in an ideal fluid.

Panel method relies on superposition property of the harmonic functions. The surface of the submerged body is divided into discrete elements called 'panels' (hence, the name 'panel' method) and each panel is assigned a potential flow element (singularity element) with unknown strength. Superposing singularity elements, the flow velocity field around the object can be calculated [74, 76].

A graphical representation of panel method is given in Figure 2.1 where an arbitrarily shaped object submerged in an ideal fluid is shown. The surface of the object is divided

into a number of panels indicated with alternating blue and red line segments on the surface. Panels are ordered clockwise starting from trailing edge of the object. For this study, prescribed flow elements on panels are selected as 2D point vortices. On Figure 2.1, $\mathbf{P} = (x, y)$ denotes an arbitrary point in the fluid. Combining Equations 2.8 and 2.9 [74] the instantaneous flow velocity at point $\mathbf{P} = (x, y)$ induced by the Panel j located at $\mathbf{P}_j = (x_j, y_j)$ can be written as in Equation 2.10 [81, 74].

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{\gamma_j}{2\pi r_{pj}^2} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} x_j - x \\ y_j - y \end{bmatrix} \quad (2.10)$$

where γ_j is the unknown vorticity strength, (x_j, y_j) denote the coordinates of the point vortex and $r_{pj} = \sqrt{(x - x_j)^2 + (y - y_j)^2}$ is the Euclidean distance between point vortex element and the point $\mathbf{P} = (x, y)$. The velocity calculation formula in Equation 2.10 can also be used for calculating the velocity induced by panels on the obstacle surface itself.

In Figure 2.2 sign convention for panel orientation angles and surface normals are

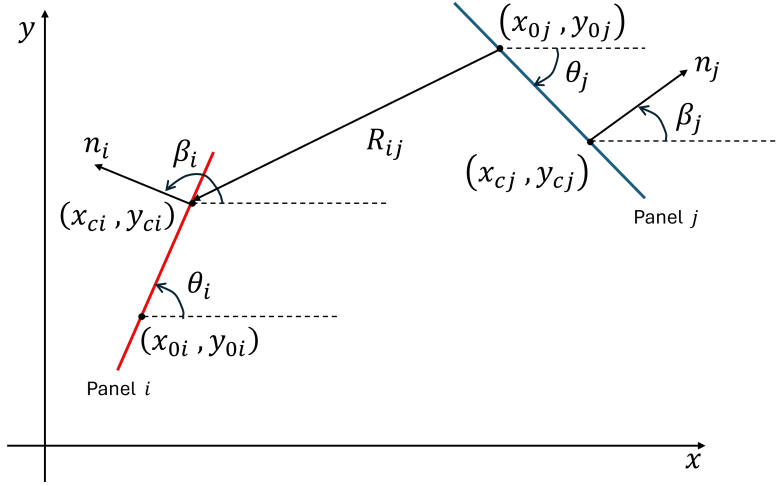


Figure 2.2: Sign convention for panel orientation angles and surface normals.

given. The prescribed singularity elements (2d point vortex for this work) are placed at $\mathbf{P}_{0i} = (x_{0i}, y_{0ci})$ and $\mathbf{P}_{0j} = (x_{0j}, y_{0j})$ for Panel i and Panel j respectively as shown in Figure 2.2. Assuming panels with length l , points $\mathbf{P}_{0i}$ and $\mathbf{P}_{0j}$ are placed at $l/4$ distance from panel's starting edge. Collocation points are located at $\mathbf{P}_{ci} = (x_{ci}, y_{ci})$ and $\mathbf{P}_{cj} = (x_{cj}, y_{cj})$ for Panel i and Panel j respectively as shown in Figure 2.2. Collocation points are placed at a distance of $l/4$ from panel's end. Therefore, there is a $l/2$ distance between collocation point and the singularity element of the same panel. 2D point vortex has its singularity at its center. Therefore, the velocity induced by a point vortex on itself cannot be calculated. By separating the singularity element and the collocation point by a distance of $l/2$, the velocity induced by the panel on itself can be calculated.

Velocity induced by a vortex element placed on j th panel at $\mathbf{P}_{0j} = (x_{0j}, y_{0j})$ on i th panel at $\mathbf{P}_{ci} = (x_{ci}, y_{ci})$ can be calculated as $\mathbf{V}_{ij} = (u, v)_{ij}$ using the relation given in

Equation 2.11, [74].

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{\gamma_j}{2\pi r_{ij}^2} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} x_{0j} - x_{ci} \\ y_{0j} - y_{ci} \end{bmatrix} \quad (2.11)$$

To model the flow around the submerged body, a boundary condition that prevents the flow penetrating obstacle surface has to be considered. That is, the normal velocity on the obstacle surface has to be zero. Enforcing this boundary condition, the unknown vorticity strengths on the panels on the obstacle surface can be calculated by equating the normal velocity induced at each panel on the object surface to zero.

Applying the boundary condition for all the panels on the surface, the unknown vorticity strength problem can be expressed as a system of linear equations given in Equation 2.12, [74].

$$K_{ij}\gamma_j = RHS_i \quad (2.12)$$

Here, K is a coefficient matrix and γ_j is the unknown vortex strength on panel j . The vector RHS consists of normal component of velocity induced by other known potential flow elements in $\mathcal{D}$.

$$K_{ij}\gamma_j = (u, v)_{ij} \cdot \vec{n}_i \quad (2.13)$$

The coefficient matrix and the vortex strength together represent the normal component of the velocity induced by j th panel on i th panel as given in Equation 2.13. Note that, the coefficient matrix K is a known matrix and composed of only the geometrical properties of panels such as their respective orientations and positions. Therefore, the coefficient matrix K is constant as long as the object's shape stays the same.

In aerodynamics, RHS vector usually contains the normal component of free stream velocity $\mathbf{V}_\infty = (u_\infty, v_\infty) \cdot \vec{n}_i$ as in Equation 2.14, [74, 76]. Here, the angle β_i is the orientation of panel surface normal as shown in Figure 2.2 and free stream velocity $\mathbf{V}_\infty = (u_\infty, v_\infty)$ is shown in Figure 2.1.

$$RHS_i = -u_\infty \cos \beta_i - v_\infty \sin \beta_i \quad (2.14)$$

After solving Equation 2.12 for unknown vortex strengths γ_j , flow velocity at any point $\mathbf{P} = (x, y)$ in $\mathcal{D}$ can be calculated using Equation 2.10 for every panel on obstacle surface and superposing the resultant velocities [74].

Panel methods are well studied in fluid mechanics and aerodynamics domain. Reader may refer to [76, 74] for detailed derivation of panel method.

2.2 Potential Guidance Flow (PGFlow) Algorithm

PGFlow algorithm mimics fluid flow in cluttered environments to generate smooth and well behaved trajectories. The PGFlow method relies on potential flow theory to generate a flow field around arbitrarily shaped obstacles and uses the potential flow field as guidance vector field to guide vehicles around the obstacles. To model the flow field around the arbitrarily shaped obstacles panel method is utilized. This section details the modifications in the existing panel method to make the method more suitable for

guidance applications.

2.2.1 Guidance Method Development

Potential flow around any arbitrarily shaped object can be modelled using Panel Methods as described in Section 2.1.3. Panel method equations given in Equation 2.12 consists of three elements: The coefficient matrix, K_{ij} , the unknown vorticity strengths, γ_j and the right hand side vector, RHS_i . The coefficient matrix K_{ij} carries the shape and orientation information of the obstacles. It is constant as long as the obstacles do not change their shape or their orientation. In aerodynamics, traditionally, the right hand side vector, RHS_i consist of the normal components of the free stream velocity [76, 74]. When the Equation 2.12 is solved with RHS given as in Equation 2.14, the resultant flow

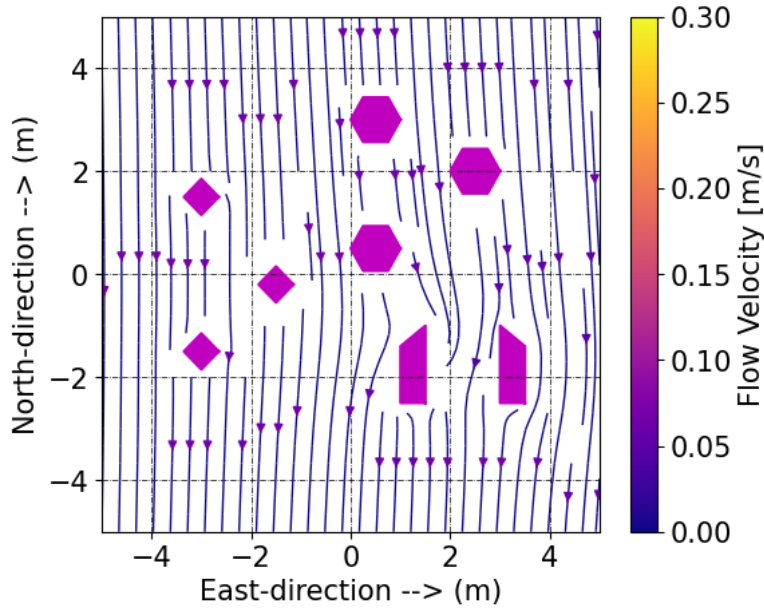


Figure 2.3: Streamlines generated by Panel Method solution.

field flows in a given direction while avoiding the obstacles on its way. Such a solution is presented in Figure 2.3. In this figure the obstacles are plotted with magenta. The axes are in m and flow velocity is in m/s as specified on the color bar. Although this solution is a strong basis for a guidance application, it needs to be further modified to better fit the needs of a guidance application.

2.2.1.1 Defining a Goal Position

The solution of the Equation 2.12 with RHS given as in Equation 2.14 results in a flow traveling in a specified direction while avoiding obstacles as presented in Figure 2.3. Although this solution inherently provides obstacle avoidance property, it lacks a specific

2 Harmonic Potential Field Based Guidance

goal position. In a guidance application it is crucial to guide vehicles to their desired destinations. Hence, only travelling in a certain direction while avoiding obstacles is not a sufficient solution for a guidance problem. To overcome this issue, first, a point sink element is introduced to represent a goal position to the flow field.

Let $\mathbf{P} = (x, y) \in \mathcal{D}$ be an arbitrary point and $\mathbf{P}_g = (x_g, y_g) \in \mathcal{D}$ be the known goal position. A point sink element with known strength, σ_g is placed at the goal position. The velocity induced at point $\mathbf{P} = (x, y)$ by the 2D sink element at $\mathbf{P}_g = (x_g, y_g)$ can be calculated as given in Equation 2.15, [74].

$$\begin{bmatrix} u_{sink} \\ v_{sink} \end{bmatrix} = \frac{-\sigma_g}{2\pi r_g^2} \begin{bmatrix} x - x_g \\ y - y_g \end{bmatrix} \quad (2.15)$$

Here, u_{sink} and v_{sink} are the components of the velocity induced by the point sink element. σ_g is the known sink strength and r_g is the Euclidean distance between the point of interest and the sink position.

The velocity induced by the point sink element are included to RHS vector as in Equation 2.16.

$$RHS_i = -u_\infty \cos \beta_i - v_\infty \sin \beta_i - u_{sink} \cos \beta_i - v_{sink} \sin \beta_i \quad (2.16)$$

The sink element alone is enough to guide the vehicle to the goal position. Therefore, when there is a sink element in the flow field the free stream velocity can be omitted as given in Equation 2.17.

$$RHS_i = -u_{sink} \cos \beta_i - v_{sink} \sin \beta_i \quad (2.17)$$

When Equation 2.12 is solved with RHS given as in Equation 2.17, the resultant flow field still satisfies the Laplace's Equation and, hence, is a harmonic potential field. By the maximum principle of harmonic potential fields, there are no local minima in this resultant vector field and the point sink element acts as the global minima. Therefore, the flow velocity vectors at any point on this potential field avoid obstacles and lead towards the sink position. PGFlow algorithm makes use of this property to have a guidance vector field with obstacle avoidance and convergence to the goal position.

2.2.1.2 Collision Avoidance

The guidance vector field based on the potential flow field has built in obstacle avoidance property, that is, all the vectors generated by the guidance field avoid obstacles. The obstacle avoidance property originates from the boundary condition of zero normal velocity at panel collocation points on obstacle surfaces imposed on panel method equations. When there are sufficient number of panels defined on the obstacle boundary, resultant potential field satisfies the obstacle avoidance property. However, avoiding obstacles alone is not sufficient when there are multiple vehicles are involved in a guidance problem. Collisions between vehicles also have to be avoided for a safe flight. To prevent collisions between vehicles within the potential guidance field, each vehicle is modeled as a point source element.

2.2 Potential Guidance Flow (PGFlow) Algorithm

Let $\mathbf{P}_n = (x_n, y_n) \in \mathcal{D}$ be the position of the n^{th} vehicle in a fleet with N vehicles. Then, the velocity induced by a point source element at $\mathbf{P}_n = (x_n, y_n)$ with known strength σ_n on any point $\mathbf{P} = (x, y) \in \mathcal{D}$ can be calculated using the formula given in Equation 2.18, [74].

$$\begin{bmatrix} u_{source}^n \\ v_{source}^n \end{bmatrix} = \frac{-\sigma_n}{2\pi r_n^2} \begin{bmatrix} x - x_n \\ y - y_n \end{bmatrix} \quad (2.18)$$

Here, u_{source}^n and v_{source}^n are the components of the velocity $V_{source}^n = (u_{source}^n, v_{source}^n)$ at point $\mathbf{P} = (x, y)$, induced by the point source element placed on the n^{th} vehicle in the fleet. σ_n is the known source strength and r_n is the Euclidean distance between the source element and the point of interest.

To incorporate the collision avoidance property to the potential field, the velocity induced by the source elements placed on each vehicle need to superposed and included in RHS vector as in Equation 2.19.

$$\begin{aligned} RHS_i = & -u_\infty \cos \beta_i - v_\infty \sin \beta_i - \sum_{n=1}^N u_{source}^n \cos \beta_i - \sum_{n=1}^N v_{source}^n \sin \beta_i \\ & - u_{sink} \cos \beta_i - v_{sink} \sin \beta_i \end{aligned} \quad (2.19)$$

Similar to Equation 2.17 it is possible to omit the free stream velocity here as well to obtain Equation 2.20.

$$RHS_i = - \sum_{n=1}^N u_{source}^n \cos \beta_i - \sum_{n=1}^N v_{source}^n \sin \beta_i - u_{sink} \cos \beta_i - v_{sink} \sin \beta_i \quad (2.20)$$

Note that, the velocity induced by source elements u_{source}^n and v_{source}^n depend on the known source strength σ_n . Source strength is a design parameter and is a property of the vehicle. That is, the user has the flexibility to assign sources with differing strengths to each vehicle to better manage the interactions between vehicles.

2.2.1.3 Calculating the Flow Velocity

Flow velocity calculation is the final step of generating the flow field. Before flow velocity at any point can be calculated, strengths of vortices placed on the obstacle surface has to be determined. Unknown vortex strengths are calculated by solving the Equation 2.12 with RHS vector as given in Equation 2.19, to take both goal position and other vehicles in the arena into account. Then, the fluid flow velocity $\mathbf{V} = (u, v)$ at any point $\mathbf{P} = (x, y) \in \mathcal{D}$ can be calculated using the relations given in Equation 2.21 and Equation 2.22.

$$u = \frac{-\sigma_g}{2\pi} \frac{x - x_g}{r_g^2} + \sum_{n=1}^N \frac{\sigma_n}{2\pi} \frac{x - x_n}{r_n^2} + \sum_{s=1}^S \sum_{k=1}^K \frac{\gamma_{sk}}{2\pi} \frac{x - x_{sk}}{r_{sk}^2} + u_\infty \quad (2.21)$$

$$v = \frac{-\sigma_g}{2\pi} \frac{y - y_g}{r_g^2} + \sum_{n=1}^N \frac{\sigma_n}{2\pi} \frac{y - y_n}{r_n^2} + \sum_{s=1}^S \sum_{k=1}^K \frac{-\gamma_{sk}}{2\pi} \frac{y - y_{sk}}{r_{sk}^2} + v_\infty \quad (2.22)$$

2 Harmonic Potential Field Based Guidance

Here, σ_g is the sink strength at the goal position, $\mathbf{P}_g = (x_g, y_g)$. The Euclidean distance between the goal position and the point of interest is r_g . The total number of aerial vehicles in the fleet is N and the source strength associated with the n th vehicle in the fleet is denoted by σ_n . Position of the n th vehicle is $\mathbf{P}_n = (x_n, y_n)$ and r_n denotes the Euclidean distance between the n th vehicle and the point of interest. S is the number of obstacles in the flight arena. K is the number of panels on a single obstacle surface. Notice that, this notation does not take interactions between obstacles into account. Position of a vortex element, i.e. the collocation point, on the s th obstacle and the k th panel is denoted by $\mathbf{P}_{sk} = (x_{sk}, y_{sk})$ and associated vortex strength is γ_{sk} . The Euclidean distance between the point vortex and the point of interest is r_{sk} . Finally, u_∞ and v_∞ are the components of the free stream velocity, V_∞ .

Using Equation 2.21 and Equation 2.22 flow velocity at any point in the flight arena can be calculated. The calculated flow velocity carries the properties of the potential field. More specifically, it cannot be normal to the obstacle surfaces at collocation points. That is, calculated flow velocity will avoid collision with obstacles given sufficiently small panels. Furthermore, since there is no local minima in this flow field any vector on this field will lead towards the global minima located at the goal position where a point sink element is placed. Last but not least, resultant vectors will also avoid collision with other vehicles in the arena that are represented by point source elements.

2.2.2 Extension to 3D Problems

This section introduces an extension of the existing 2-dimensional PGFlow method to 3-dimensions. PGFlow algorithm depends heavily upon the panel method solution used in potential flow theory. Potential flow equations and panel method both have 3-dimensional counterparts. In order to use the 3-dimensional panel method solution, 3-dimensional obstacle models are required. When 3-dimensional obstacle models are used, the element number for a single obstacle increases more than 40 times. This increase in the element number brings immense computational load to the algorithm, jeopardizing the real-time solution capability. An alternative strategy is considered to preserve the real-time guidance capability of the PGFlow method. Instead of generating 3-dimensional obstacle models, only the 2-dimensional cross-sections of obstacles at relevant altitudes are considered. The guidance vector field is calculated using these 2-dimensional obstacle cross-sections at a given altitude. Meanwhile, destinations and vehicles are represented as 3-dimensional point sinks and sources.

The combined effect of 2-dimensional obstacles and 3-dimensional sink and sources governs the vehicle's motion. Therefore, flow velocity calculations should be adjusted as in Equation 2.23, Equation 2.24, and Equation 2.25.

$$u = \frac{-\sigma_g}{2\pi} \frac{x - x_g}{r_g^2} + \sum_{n=1}^N \frac{\sigma_n}{2\pi} \frac{x - x_n}{r_n^2} + \sum_{s=1}^S \sum_{k=1}^K \frac{\gamma_{sk}}{2\pi} \frac{x - x_{sk}}{r_{sk}^2} \quad (2.23)$$

$$v = \frac{-\sigma_g}{2\pi} \frac{y - y_g}{r_g^2} + \sum_{n=1}^N \frac{\sigma_n}{2\pi} \frac{y - y_n}{r_n^2} + \sum_{s=1}^S \sum_{k=1}^K \frac{-\gamma_{sk}}{2\pi} \frac{y - y_{sk}}{r_{sk}^2} \quad (2.24)$$

$$w = \frac{-\sigma_g}{4\pi} \frac{z - z_g}{r_g^3} + \sum_{n=1}^N \frac{\sigma_n}{4\pi} \frac{z - z_n}{r_n^3} \quad (2.25)$$

The movement of the vehicle within the 2-dimensional plane is dictated by the collective influence of 2-dimensional obstacle cross-sections and relevant components of the 3-dimensional sources and sinks, expressed in Equation 2.23 and Equation 2.24. The altitude of the vehicle is exclusively controlled by the 3-dimensional source and sink elements, as denoted in Equation 2.25.

The cross-sections of the obstacles should be dynamically updated as vehicles change their altitude to maintain collision-free flight. This 3-D extension method allows vehicles to ascend or descend toward their intended positions without the risk of colliding with obstacles or other vehicles within the same airspace. Importantly, it maintains the computational efficiency of the 2-dimensional panel method solution, ensuring minimal computational cost.

2.3 Enhancements for Safety

2.3.1 Defining a Safety Perimeter Around Obstacles

The obstacle avoidance property in the proposed guidance algorithm relies on the zero normal velocity boundary condition on panel method equations. On the other hand, there is no constraint on the tangent velocity on the obstacle surface. Although flow velocity vectors that are tangent to obstacle boundaries are valid solutions of panel method equations, for guidance of vehicles with finite volume; guidance vectors tangent to obstacle boundary are unsafe. One common solution to this problem is expanding the obstacle boundary with the amount equal to or greater than vehicle size and working with inflated obstacles in a so called configuration space. This way, even if the guidance vector is tangent to the obstacle surface, the vehicle following this vector will not collide with the obstacle. This approach is also employed in this thesis work.

Additionally, to further improve the safety and to increase the safety perimeter around the obstacles a Safety Source Element (SSE) is introduced [82]. The SSE is a point source element that travels on the vehicle itself. The SSE is integrated in RHS vector as in Equation 2.26.

$$\begin{aligned} RHS_i = & -u_\infty \cos \beta_i - v_\infty \sin \beta_i - \sum_{n=1}^N u_{source}^n \cos \beta_i - \sum_{n=1}^N v_{source}^n \sin \beta_i \\ & - u_{sink} \cos \beta_i - v_{sink} \sin \beta_i - u_{safety} \cos \beta_i - v_{safety} \sin \beta_i \end{aligned} \quad (2.26)$$

When this additional source element is included in RHS vector for unknown vortex strength calculations, the resultant vortex strengths on the obstacles in the vicinity

of the SSE are amplified to satisfy the zero normal velocity condition on the obstacle surface. That is, obstacles in the vicinity of the vehicle have stronger repulsion. To benefit from this additional repulsive effect, the SSE has to be only used in unknown vortex strength calculations and not included in velocity vector calculations. This way, resultant amplified vortex elements can prevent the tangent solutions.

The effect of the SSE is demonstrated in Figure 2.4. The resultant paths plotted in

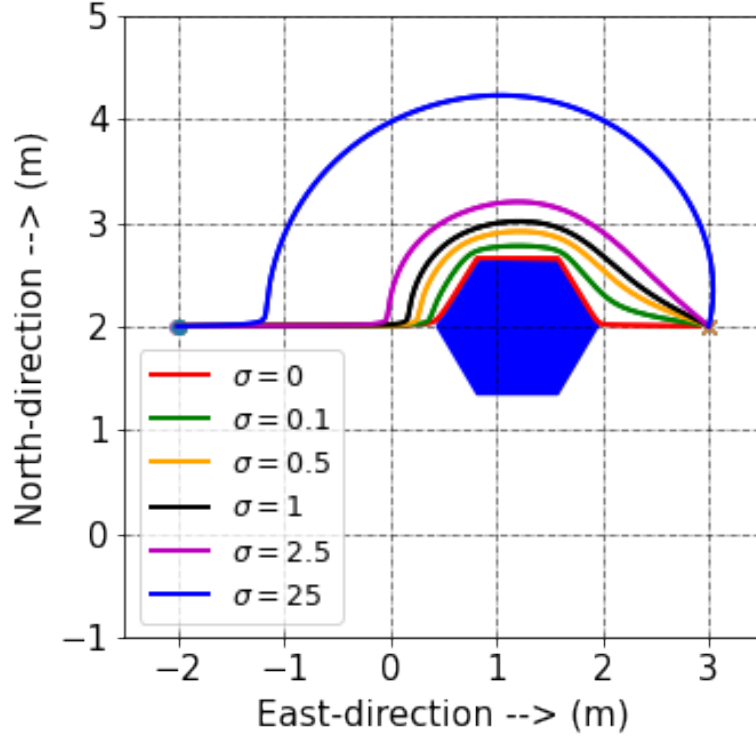


Figure 2.4: Paths generated with safety source elements with strengths ranging from $\sigma = 0$ to $\sigma = 25$ [82].

Figure 2.4 are generated with SSE with strengths ranging from $\sigma = 0$ to $\sigma = 25$. The red path, generated with safety source strength $\sigma = 0$ is the nominal solution without any safety factor. When there is no SSE, the resultant path is tangent to the obstacle surface, as expected. When the strength of the SSE is increased, the safety perimeter around the obstacle also grows. For this scenario, the sink strength at the goal position is set to $\sigma_g = 5$. Notice that, even with a SSE with strength $\sigma = 25$ which is 5 times stronger than the attractive strength of the goal position, the vehicle still converges to the goal position. That is, as the strength of the SSE is increased, a trajectory further away from the obstacle surface is generated while still converging to the goal position. To achieve a similar effect by adjusting the expansion radius of obstacles, the coefficient matrix given in Equation 2.12 has to be recalculated and inverted again for each new safety perimeter change; resulting in a large computational burden. On the other hand, the SSE only appears in calculations as an additional source element; thus,

its computational burden is negligible and it can be adjusted and recalculated online. Furthermore, SSE is a source element attached to the vehicle; that is, it is a property of the vehicle itself. Therefore, each vehicle can have its own SSE; hence, all vehicles can have different safety perimeters around obstacles depending on their performance specifications. For example, vehicles with higher maneuverability can travel in closer proximity to obstacles; hence, can be assigned smaller SSE. Whereas, less agile vehicles need more space to maneuver; hence, they can be assigned larger SSE. This functionality would allow the simultaneous operation of multiple vehicles with different capacities within the same framework, ensuring both safety and efficiency. This feature holds particular significance in urban air mobility applications, where various vehicles with different capabilities are expected to share the same airspace.

2.4 Enhancements for Agile Flight

Resultant trajectories of the PGFlow are inherently smooth as they mimic the fluid flow around submerged bodies. However, vehicle dynamics and maneuverability are not considered in desired velocity calculations. Furthermore, the guidance vector generated by the PGFlow algorithm is not optimized. The only constraint on the guidance vector is that velocities normal to the obstacle surface are zero. Therefore, the resultant path can still have turns with smaller radii that force the vehicle to slow down to stay within flight envelope limits.

Two enhancements are proposed in this thesis work to enable faster flight using PGFlow method. The first approach is propagating the alternative trajectories and selecting the smallest curvature path among the alternatives. The second alternative is projecting the current position on a straight line in the vehicle's current travel direction to detect obstacles ahead. If a collision with an obstacle is foreseen, the collision avoidance maneuver is initiated. Both approaches aim to obtain even smoother paths, thus increasing the vehicle velocity while protecting the flight envelope limits.

2.4.1 Guidance with Path Propagation

The streamline connecting the start and goal positions is unique for given flow elements in a flight arena. Therefore, the resultant trajectory cannot be optimized without modifying the flow elements. To have smoother alternatives to the existing streamline, a uniform flow element perpendicular to the line formed by the current position and goal position is added to flow velocity calculations.

Let $\mathbf{U} = (u_u, v_u)$ be the velocity of the uniform flow. The flow velocity components u_u and v_u are obtained as given in Equations 2.27 and 2.28.

$$u_u = \|\mathbf{U}\| \cos \theta \quad (2.27)$$

$$v_u = \|\mathbf{U}\| \sin \theta \quad (2.28)$$

Here, $\|\mathbf{U}\|$ is the magnitude of the uniform flow, and the angle θ is the angle of attack of the uniform flow perpendicular to the vehicle's travel direction. For a vehicle positioned

2 Harmonic Potential Field Based Guidance

at any point $\mathbf{P} = (x, y) \in \mathcal{D}$ with a desired destination located at $\mathbf{P}_g = (x_g, y_g)$ the angle θ can be calculated as in Equation 2.29.

$$\theta = \arctan \frac{y_g - y}{x_g - x} \pm \frac{\pi}{2} \quad (2.29)$$

As it can be inferred from Equation 2.29, the direction of the uniform flow can be altered as long as it is perpendicular to the vehicle's travel direction. This new uniform flow element is introduced in panel method calculations as shown in Equation 2.30

$$\begin{aligned} RHS_i = & -u_\infty \cos \beta_i - v_\infty \sin \beta_i - \sum_{n=1}^N u_{source}^n \cos \beta_i - \sum_{n=1}^N v_{source}^n \sin \beta_i \\ & - u_{sink} \cos \beta_i - v_{sink} \sin \beta_i - u_{safety} \cos \beta_i - v_{safety} \sin \beta_i \\ & - \|\mathbf{U}\| \cos \theta \cos \beta_i - \|\mathbf{U}\| \sin \theta \sin \beta_i \end{aligned} \quad (2.30)$$

After solving the Equation 2.12 with the RHS vector given in Equation 2.30, the resultant flow velocity can be calculated as given in Equation 2.31 and Equation 2.32.

$$u = \frac{-\sigma_g}{2\pi} \frac{x - x_g}{r_g^2} + \sum_{n=1}^N \frac{\sigma_n}{2\pi} \frac{x - x_n}{r_n^2} + \sum_{s=1}^S \sum_{k=1}^K \frac{\gamma_{sk}}{2\pi} \frac{x - x_{sk}}{r_{sk}^2} + u_\infty + \|\mathbf{U}\| \cos \theta \quad (2.31)$$

$$v = \frac{-\sigma_g}{2\pi} \frac{y - y_g}{r_g^2} + \sum_{n=1}^N \frac{\sigma_n}{2\pi} \frac{y - y_n}{r_n^2} + \sum_{s=1}^S \sum_{k=1}^K \frac{-\gamma_{sk}}{2\pi} \frac{y - y_{sk}}{r_{sk}^2} + v_\infty + \|\mathbf{U}\| \sin \theta \quad (2.32)$$

The uniform flow's magnitude and direction can be changed to obtain alternative paths. Among the alternative propagated paths, the path with a smaller curvature is more desirable. The Potential Guidance Flow algorithm does not generate paths for vehicles to follow; instead, it generates a guidance vector that can be used as reference velocity. That is, the output of the PGFlow algorithm is not a trajectory but an instantaneous guidance vector. However, a path or a path segment is required for curvature comparison. Consequently, to have a path curvature comparison, vehicle trajectories have to be propagated using the proposed uniform flow addition in flow velocity calculations.

In the path propagation step, vehicle dynamics can also be considered. A linear model of the vehicle described in Equation 2.33 can be considered for path propagation.

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k \quad (2.33)$$

Here, the state vector $\mathbf{x} \in \mathbb{R}^{6 \times 1}$ consists of vehicle position and velocity. Control vector $\mathbf{u} \in \mathbb{R}^{3 \times 1}$ is the desired velocity vector. Matrix $\mathbf{A}$ is the state matrix, and matrix $\mathbf{B}$ is the control matrix.

2.4.2 Optimal Path Selection

For optimal path selection, the curvature of the path is considered as the selection criteria. As curvature decreases, the path gets smoother. Smooth paths are easier to

track, and since the turn radius is smaller, it allows for faster traveling speeds.

Alternative propagated paths are plane curves that can be represented by Cartesian parametric equations $x = x(t)$ and $y = y(t)$. Then, the curvature κ of the propagated path can be calculated using Equation 2.34 [83].

$$\kappa = \frac{x'y'' - y'x''}{(x'^2 + y'^2)^{3/2}} \quad (2.34)$$

Here, κ is the curvature of the propagated path. x' and y' are the first-order derivatives of the path with respect to time. Similarly, x'' and y'' denote the second-order derivatives of the path with respect to time.

Calculated curvatures are compared to select the smoothest path, that is, the path with the least curvature. Even though an optimal path in terms of smoothness is selected at this point, the working principle of the proposed guidance algorithm is not trajectory tracking. Rather, it is tracking a guidance velocity vector. Consequently, the uniform flow strength $\|\mathbf{U}\|$ associated with the selected path is inserted into Equation 2.31 and Equation 2.32 to calculate the desired velocity to be fed to the vehicle. Path propagation and optimal path selection procedure are then repeated at the new position.

2.4.3 Guidance with Simple Position Propagation

The path propagation method described in Section 2.4.1 requires calculating the path segment for propagation. Propagating a shorter path segment does not yield the desired increase in vehicle speed, as the sharp turns cannot be predicted early enough. On the other hand, propagating a significantly long path segment is computationally costly. When computational power is a limiting factor, an approach with a lower computational burden is more desirable. The position propagation method described in this section is the simplified version of the path propagation method.

When resultant trajectories are inspected in numerous simulations and experiments, it has been observed that sharper turns usually occur during collision avoidance maneuvers. Therefore, predicting collisions and starting the avoidance maneuver earlier would minimize the sharp turns in the resultant trajectory. For collision prediction, a line segment in the direction of the desired velocity vector at a given time is considered. This line segment represents the trajectory to be followed by the vehicle if it continues to travel with its current velocity for a set amount of time. In order to avoid any additional computational load, this approach does not involve optimal path selection. Instead, if any collisions along the line segment are predicted, a uniform flow element perpendicular to the travel direction is introduced.

Let $\mathbf{U} = (u_u, v_u)$ be the velocity of the uniform flow. The flow velocity components are $u_u = \|\mathbf{U}\| \cos \theta$ and $v_u = \|\mathbf{U}\| \sin \theta$ with $\|\mathbf{U}\|$ being the magnitude of the uniform flow and the angle θ being the angle of attack of the uniform flow. For a vehicle positioned at any point $\mathbf{P} = (x, y) \in \mathcal{D}$ with a desired destination located at $\mathbf{P}_g = (x_g, y_g)$ the angle θ can be calculated as in Equation 2.35.

$$\theta = \arctan \frac{y_g - y}{x_g - x} + \text{sign} \phi \frac{\pi}{2} \quad (2.35)$$

2 Harmonic Potential Field Based Guidance

where *sign* function is as given in Equation 2.36.

$$\text{sign } x = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{if } x = 0 \\ -1, & \text{if } x < 0 \end{cases} \quad (2.36)$$

The direction of the uniform flow is determined geometrically by taking the collision line

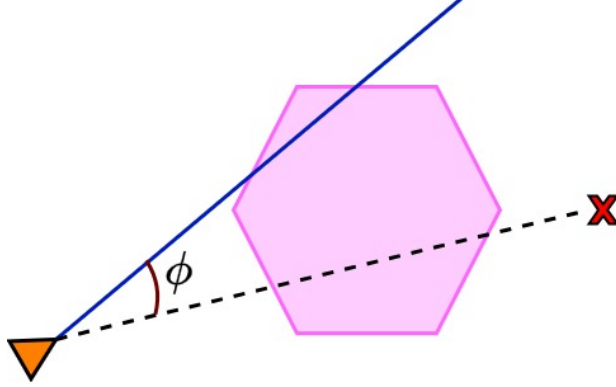


Figure 2.5: The angle between the propagated line segment and the line connecting the vehicle's current position to the goal position.

and goal position into account and represented by the angle ϕ in Equation 2.35. ϕ is the angle between the propagated line segment and the line connecting the vehicle's current position to the goal position as plotted in Figure 2.5. In this figure, the pink hexagon represents an obstacle with an arbitrary shape. The orange triangle is the vehicle, and the red cross is the goal location. The dashed line is the line connecting the vehicle to the goal position, and the solid blue line is the propagated line. For the scenario plotted in Figure 2.5, the angle ϕ is negative. Using the sign of ϕ when determining the normal direction of the uniform flow ensures that uniform flow is always from the propagated line toward the line connecting the vehicle and the goal. Note that picking this direction does not ensure the shortest way to avoid the obstacle.

2.5 Guidance Algorithm

In this section details how all components of the proposed guidance algorithm come together.

Potential Guidance Flow algorithm is based on panel method solution. Panel method equations provide a flow field solution with obstacle avoidance property. Existing panel equations are further modified to fit the guidance problem as detailed in Section 2.2.1. Modified equations result in a vector field that has both obstacle avoidance and collision avoidance properties. Moreover, all the vectors on the resultant vector field lead to the preassigned destination position. Although the modified equations provide a desirable vector field, a guidance algorithm is needed to utilize the vector field for the guidance

of aerial vehicles.

A block diagram for PGFlow algorithm is presented in Figure 2.6. The input for the

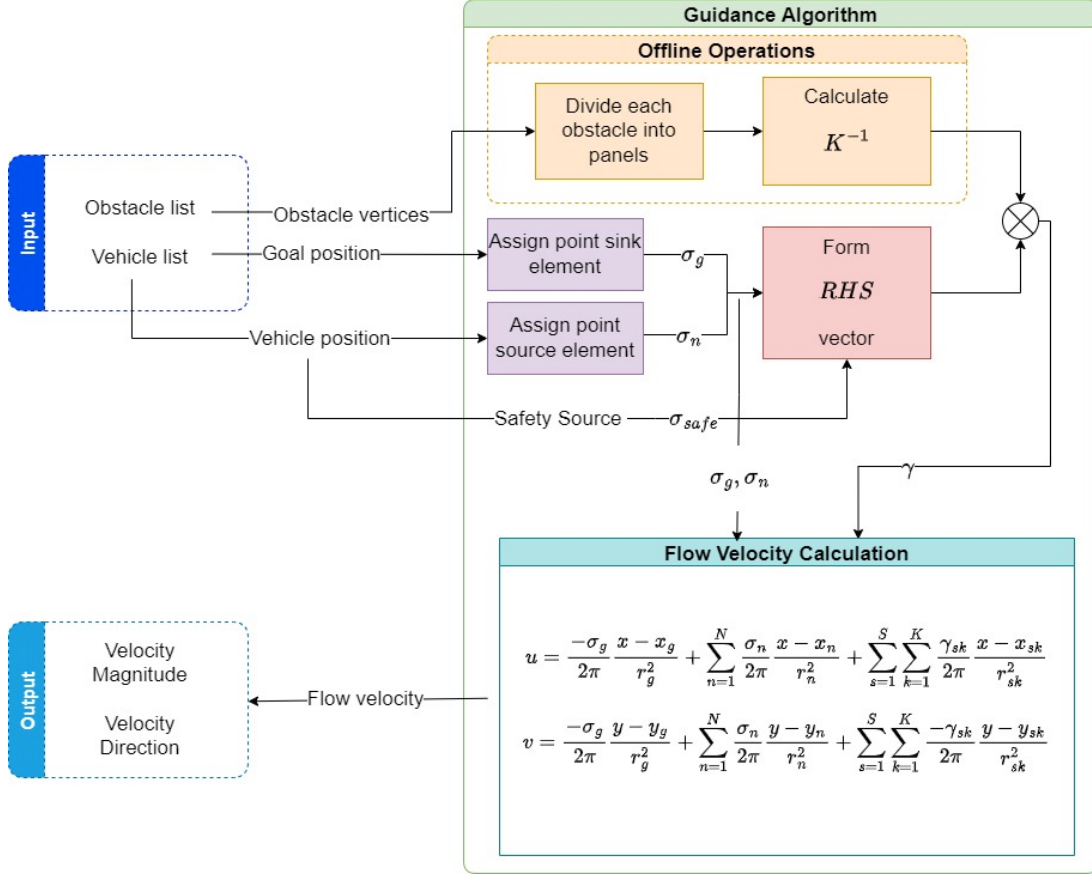


Figure 2.6: Block diagram for Potential Guidance Flow algorithm in 2D.

Potential Guidance Flow algorithm is the vehicle list and the obstacle list. The obstacle list contains the position of obstacle vertices. The vehicle list contains information on vehicle positions, desired destinations as well as the magnitude of sink and source strengths to be assigned to respective positions.

The first step in the algorithm is dividing the obstacle boundary into small line segments called panels and assigning vortex elements with unknown strengths to each panel. To assign the vortex strengths that provide obstacle avoidance, the equation 2.12 needs to be solved for unknown vortex strengths. The coefficient matrix K in Equation 2.12 only depends on obstacle shape and orientation and is a constant as long as obstacles do not change their shape. Therefore, to reduce the computational load, calculation and inversion of coefficient matrix K is completed offline. Simultaneously, point source and point sink elements with known strength are assigned to vehicles' initial position and desired goal positions respectively to obtain the RHS vector in Equation 2.12. Here, the SSE is also included in the RHS vector. After K^{-1} and RHS are calculated, the Equation 2.12

2 Harmonic Potential Field Based Guidance

is to be solved for unknown vortex strengths, γ_j . Once the vortex strengths on the obstacle boundaries are obtained, the flow velocity at the current vehicle position can be calculated by Equation 2.21 and Equation 2.22. For 3D applications, Equation 2.23, Equation 2.24, and Equation 2.25 is used for flow velocity calculations. Note that, the SSE is not included in flow velocity calculation.

3 Simulation Results

This chapter presents a thorough analysis of the developed guidance algorithm, PGFlow, within a simulated environment. Simulations serve as a crucial intermediate step between guidance algorithm development and hardware experiments by providing a controlled and safe environment for testing. Additionally, they enable the exploration of scenarios that may be impractical to replicate in physical experiments. For example, physically building different flight arenas and obstacles for each experiment would be infeasible. In this chapter, the effectiveness of the PGFlow algorithm is examined across various simulated scenarios.

This chapter is organized as follows: Section 3.1 introduces the simulation setup used in this thesis. Section 3.2 presents a computational load and element size study. Then, in Section 3.3, simulations with the single vehicle are presented. The aim of Section 3.3 is to demonstrate the inherent obstacle avoidance and goal converge properties of the proposed method. Next, in Section 3.4, simulation results containing multiple vehicles are presented to demonstrate the collision avoidance capability of the proposed method. Section 3.5 focuses on simulations conducted using the proposed path propagation and position propagation methods. Finally, in Section 3.6, conclusions are derived from simulation results.

3.1 Simulation Setup

This thesis considers two different simulation environments. First, a simple simulation environment in python is used to evaluate the resultant trajectories. This initial simulation environment does not take vehicle dynamics or sensors into account. The simple simulation environment in python is advantageous as it provides an environment for rapid method development. The simple simulation environment also enables user to test large scale scenarios. The second simulation environment is *Gym-Pybullet-Drones* environment described in [84]. The physics engine provides realistic collisions and aerodynamic effects. The proposed simulation environment is shown in Figure 3.1.

The Pybullet-simulation environment includes vehicle dynamics alongside a simplified rotor dynamics model. The rotor model represents rotor force, F_r , and rotor torque, Q_r , as proportional to propeller rotational speed, ω , as given in Equation 3.1.

$$\begin{aligned} F_r &= k_f \omega^2 \\ Q_r &= k_q \omega^2 \end{aligned} \tag{3.1}$$

Where k_f and k_q are the rotor force coefficient and rotor torque coefficient used in the dynamics model. The model does not consider high incidence angle conditions.

3 Simulation Results

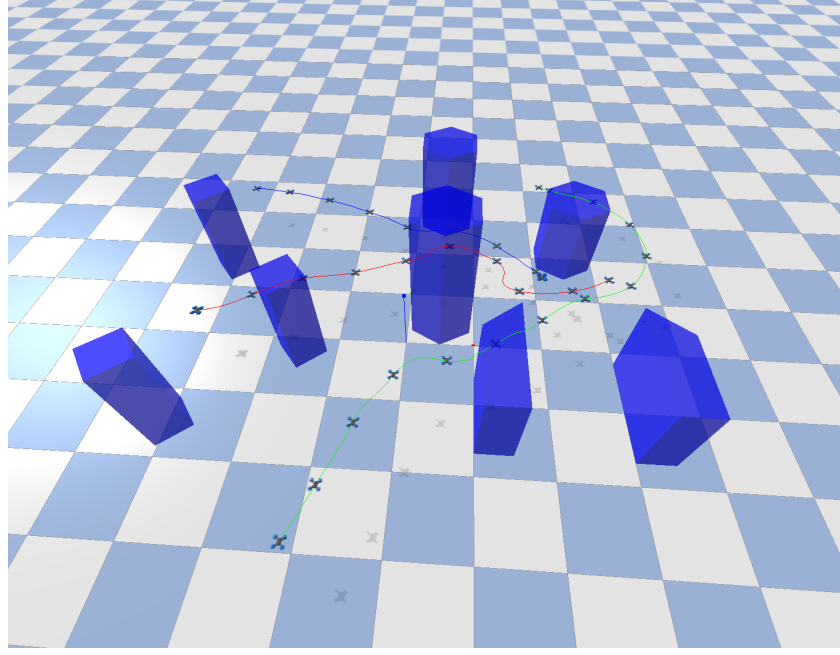


Figure 3.1: Simulation environment used during development and testing of the PGFlow algorithm.

Full knowledge of vehicle position and flight arena is assumed in dynamic simulations. After the flow velocity is calculated and scaled, the simulation uses it as reference velocity. The reference velocity is updated at 24Hz in dynamic simulations. A 2-D cross-section of an example arena is presented in Figure 3.2.

In Figure 3.2, the geometric shapes plotted in magenta represent the 2-D cross-sections of the obstacles in the flight arena. As a safety measure, the boundaries of obstacles are expanded by an amount equal to the radius of the vehicle. Shaded areas around the obstacle boundaries are the expanded safety regions. These expanded safety boundaries are considered in the flow velocity calculations. Horizontal and vertical lines indicate indoor arena limits, and the x-axis and y-axis are both in meters.

3.2 Computational Load Study

One of the main advantages of the PGFlow algorithm is its low computational load. The computational load of a guidance algorithm is a determining factor in whether it can run in real-time or not. Furthermore, low computational load is also advantageous when the algorithm is running on onboard processors. As aerial vehicles get smaller, the computational power they can carry is also reduced; hence, the computational load of a guidance algorithm can be a limiting factor when working with micro aerial vehicles. First, this section inspects the element size sensitivity on the method's performance. After determining an element size suitable for guidance purposes, a computational workload

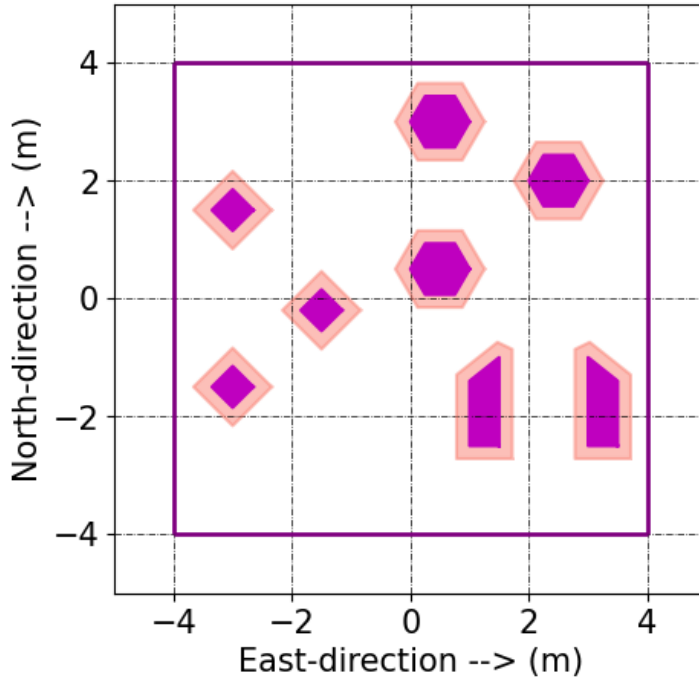


Figure 3.2: 2-D cross-section of the arena.

study is conducted on the number of obstacles and the number of vehicles. For both element size study and computational work load study vehicle dynamics and aerodynamic effects are neglected.

3.2.1 Element Size Sensitivity Study

The panel method is a numerical tool used for approximating potential flow around arbitrarily shaped obstacles by dividing the obstacle boundary into discrete elements. As the obstacle shape becomes more complex, naturally, the number of discrete elements that represent that shape also increases. Both for irregular obstacle shapes and polygons, the more elements are used, the more accurately the obstacle can be represented. However, increasing the number of elements also brings a larger computational load. Unfortunately, there is no one-size-fits-all solution for deciding the number of elements for any specific shape. It depends on the performance requirements of the problem at hand. For the guidance problem addressed in this thesis, being able to run the analysis in real time is as important as accurately representing the obstacles.

An element size study is presented in Figure 3.3. In this figure, magenta shapes are the obstacles in an arena. The red line is the resultant trajectory of the vehicle in this scenario.

In Figure 3.3a, there are a total of 40 panels on 8 obstacles; that is, each edge of the obstacles is represented by a single vortex element. Notice that obstacle shapes cannot be

3 Simulation Results

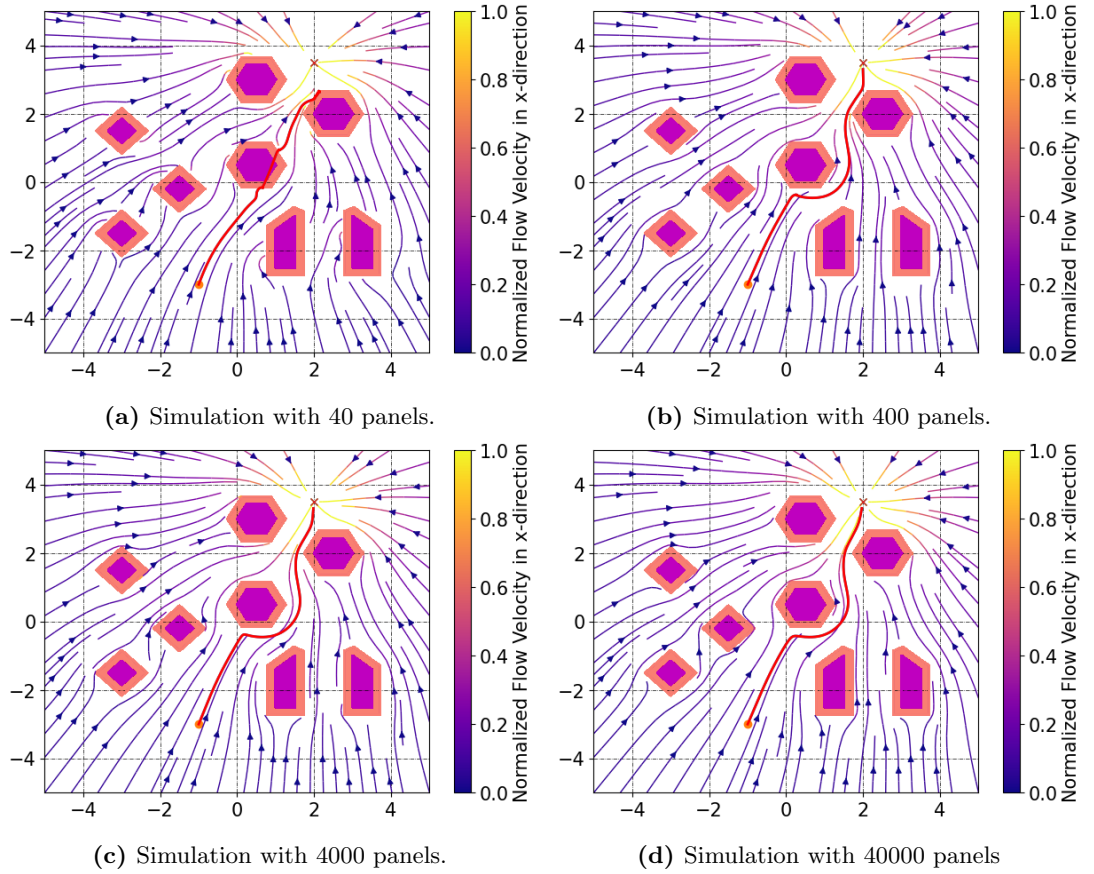


Figure 3.3: Element size sensitivity study.

represented when an insufficient number of elements are used, and the vehicle will collide with the obstacles. In Figure 3.3b element number is increased from 40 to 400. When the number of elements is increased, obstacles are better represented, and streamlines start avoiding them. In Figure 3.3c and Figure 3.3d, the same simulation scenario is repeated with 4000 panels and 40000 panels, respectively. Notice that, after a certain number of elements, resultant streamlines do not change significantly. That means adding more elements in calculations will not improve the obstacle avoidance performance but will still contribute to the computational burden of the algorithm. Therefore, an element size that can provide both real-time operations and sufficiently good obstacle representation should be selected. In this thesis work, the element size is selected as 0.1 m, which corresponds to 4000 elements for the scenario depicted in Figure 3.3.

3.2.2 Computation Work Load Sensitivity Study

Two elements in the Potential Guidance Flow algorithm contribute to the algorithm's computational load: the number of vehicles and the number of obstacles.

The communication speed for the experimental set-up used in this thesis is 10Hz; hence,

3.2 Computational Load Study

the guidance algorithm has to run at least 10Hz to be considered suitable for a real-time application. A series of simulations are run to measure how computational time changes with increasing variables; in this case, the number of obstacles in the arena and the number of vehicles in the simulation.

The computational workload study results are presented in Figure 3.4 and Figure 3.5. Here, in Figure 3.4a, the computation times are recorded for an increasing number of

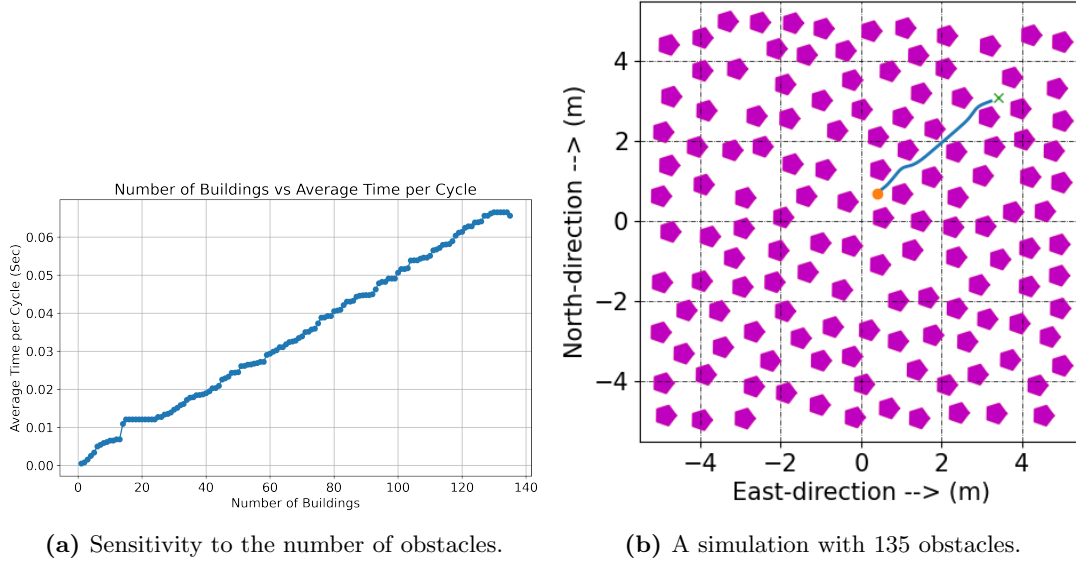


Figure 3.4: Computation load sensitivity study.

obstacles in the arena. For these simulations, a generic pentagon obstacle with a set size is placed at random locations, and the computation time for generating the guidance vector for a single vehicle with a random start and goal position is measured. The recorded computational time data suggests that the computational load of the algorithm increases linearly with an increasing number of obstacles. In Figure 3.4b, an example scenario with 135 buildings is plotted.

Similarly, in Figure 3.5a, the computation time for an increasing number of vehicles is plotted. There is only a single pentagon building at the center of the arena for these simulations, and vehicles with random start and goal positions are added to the simulation. The computational time for generating guidance vectors for the vehicles is measured. Recorded computation time data suggests that the computational load of the algorithm increases quadratically with an increasing number of vehicles. In Figure 3.5b, an example scenario with 25 vehicles is plotted.

System specifications for measuring computation times are given in Table 3.1.

3 Simulation Results

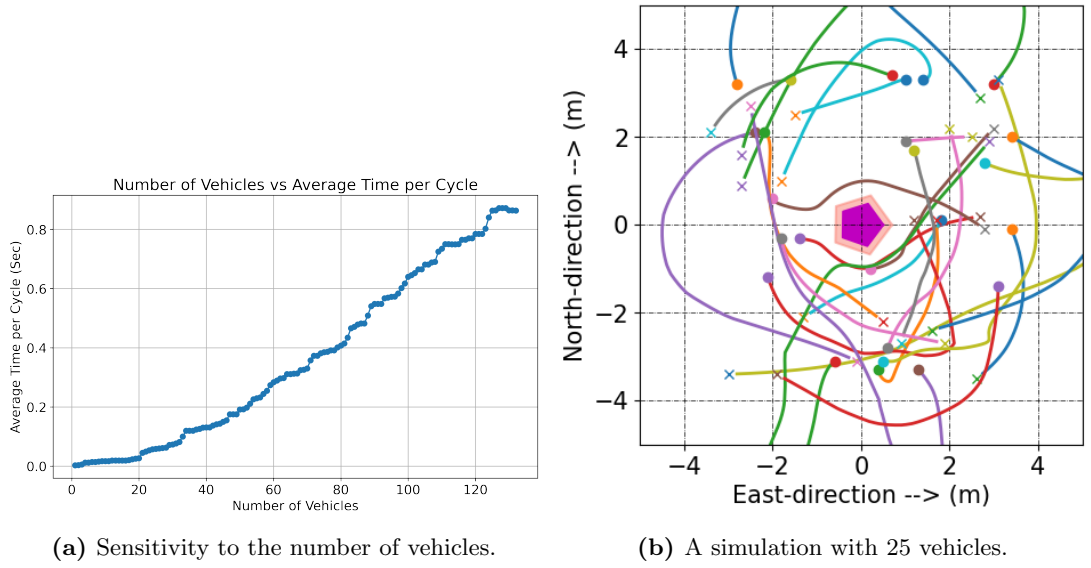


Figure 3.5: Computation load sensitivity study.

Table 3.1: System Specifications

Component	Specification
Processor	12th Gen Intel(R) Core(TM) i7-12800H
Clock Speed	4.7GHz
RAM	32GB
Environment	Jupyter
Python Version	3.10.12

3.3 Single Vehicle Guidance Scenarios

This section presents two different simulation scenarios to demonstrate the guidance capabilities of the PGFlow algorithm. The first scenario is the baseline scenario, where a single vehicle is tasked with traveling around various obstacles to a goal position. In the baseline scenario, the obstacle avoidance and convergence to goal position properties of the proposed guidance method are demonstrated. In the second scenario, a more challenging obstacle configuration is inspected to test if the vehicle can get stuck in a local minima caused by a U-shaped obstacle.

3.3.1 Baseline scenario

An example scenario for a single vehicle is presented in Figure 8.8. In this figure, magenta shapes are the 2D cross-section of obstacles. In Figure 3.6a, the simulation is run in an arena with obstacles with various convex geometries. The obstacles depicted in this example are selected so that they are easy to manufacture and can also be used in hardware experiments later. In Figure 3.6b, an arena with only concave obstacles is

presented to demonstrate the ability of the proposed algorithm to generate collision-free solutions for any arbitrarily shaped obstacle. Figure 8.8 marks the start position with a red circle and the goal position with a red cross. The red line between the start and the goal position is the resultant trajectory followed by the vehicle. The velocity vector field generated by PGFlow solution is also plotted as streamlines on the arena map. Notice that each streamline leads to the desired goal point while avoiding obstacles. This means that starting from any arbitrary position in the arena, the velocity vector calculated by PGFlow algorithm leads to the goal position with obstacle avoidance. This property makes PGFlow algorithm robust against disturbances. Even if a vehicle deviates from its initially intended path due to some disturbance, a new velocity vector that avoids obstacles and leads to the goal position can be calculated at the new position.

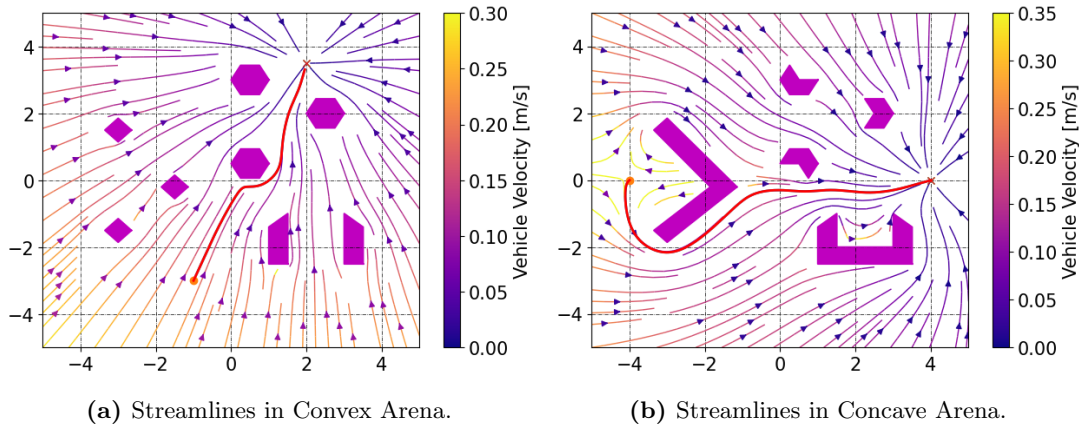


Figure 3.6: Streamlines and resultant paths of aerial vehicles in different arenas.

Although the streamlines generated by the vector field are plotted in Figure 8.8, in practice, neither streamlines in the flight arena nor the entire vehicle trajectory are calculated. Instead, PGFlow algorithm relies on calculating an instantaneous desired velocity vector at the vehicle's current position. Then, the desired velocity vector is fed to the automatic flight controller as described in Section 4.2. This property eliminates the need for a trajectory-following algorithm, and collision-free guidance can be satisfied with a simple velocity-tracking controller. Therefore, the red paths plotted in Fig. 8.8 are the resultant paths traveled by the vehicle by following the instantaneous desired velocity vectors. Consequently, PGFlow algorithm should not be confused with search algorithms that select an optimal path among multiple candidates.

3.3.2 Local Minima Related Tests

The vector field generated using potential flow theory is a harmonic potential field. One of the properties of harmonic potential fields is that there are no local minima, and convergence to global minima is guaranteed [65]. PGFlow uses potential flow equations to generate a guidance vector field to take advantage of this property. Potential flow solution results in a stagnation point on the obstacle surface facing upcoming flow [76].

3 Simulation Results

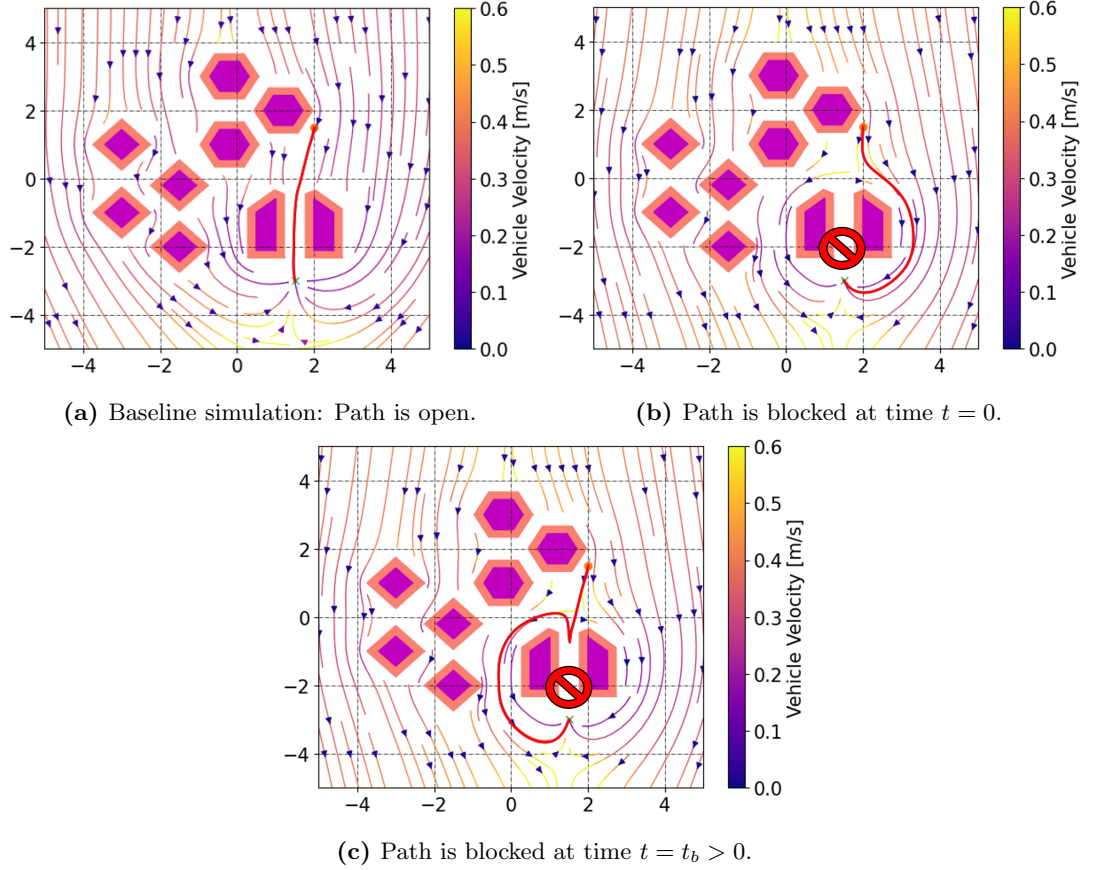


Figure 3.7: Streamlines generated by PGFlow for a passage blocked by a virtual obstacle.

This stagnation point should not be confused with local minima; by definition, the stagnation point in the potential flow field is an unstable saddle point [66].

This section demonstrates the global convergence property of the proposed PGFlow algorithm through an example simulation scenario. In this scenario, a U-shaped obstacle is placed between the vehicle and its goal, which is to evaluate whether the vehicle gets trapped inside this obstacle. Simulation results for this example scenario are presented in Figure 3.7.

There is no U-shaped obstacle for the baseline simulation presented in Figure 3.7a. Instead, the start and goal positions are placed at the opposite ends of a narrow passage. As the resultant path plotted in Figure 3.7a indicates, the vehicle successfully travels through the passage and arrives at the destination. Notice that the vehicle also does not oscillate between two obstacles. In Figure 3.7b, the second simulation is presented where the passage's exit is blocked. Then, the exit is blocked, and the corridor becomes a U-shaped obstacle. Having a priori information on the positions of the obstacles, the vehicle does not even enter the blocked corridor and travels around the U-shaped obstacle. In the final scenario, the passage is blocked after the vehicle enters the narrow

passage. The resultant path is plotted in Figure 3.7c. In this case, the vehicle travels as in the baseline simulation until the passage is blocked. As soon as the corridor's exit closes, the vehicle changes direction and heads out of the U-shaped obstacle. Finally, it reaches the goal position through an alternative path around the obstacles.

This simulation demonstrates obstacles can not cause local minima within the vector field generated by PGFlow. As a matter of fact, U-shaped obstacles are no different than the concave obstacles presented in Figure 3.6b. Furthermore, PGFlow algorithm does not try to calculate the entire path; instead, a desired velocity vector is calculated at each time step. This approach enables the proposed guidance method to calculate an alternative route when there is a change in the environment, such as the one presented in Figure 3.7c.

3.4 Multiple Vehicle Guidance Scenarios

When more than one aerial vehicle is operating in the same arena simultaneously, collision between vehicles should also be prevented in addition to obstacle avoidance. Collisions between vehicles are avoided by modeling each vehicle as a point source element in potential field calculations as described in Section 2.2.1. This way, vehicles repel other vehicles in the arena, assuming complete knowledge of the positions of all aerial vehicles in the fleet. The point source element and the vehicle are attached and move together. Consequently, the guidance vector field needs to be updated with the vehicle's movement.

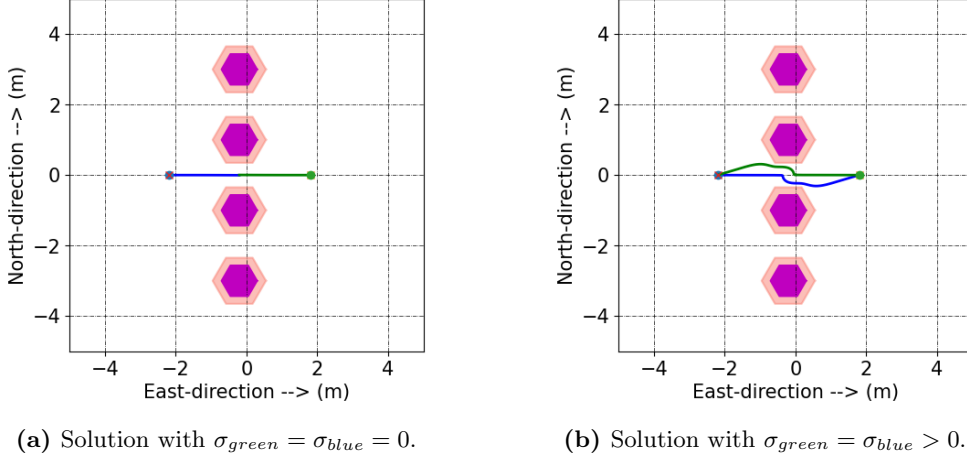


Figure 3.8: Simulations with varying source strengths on vehicles.

3 Simulation Results

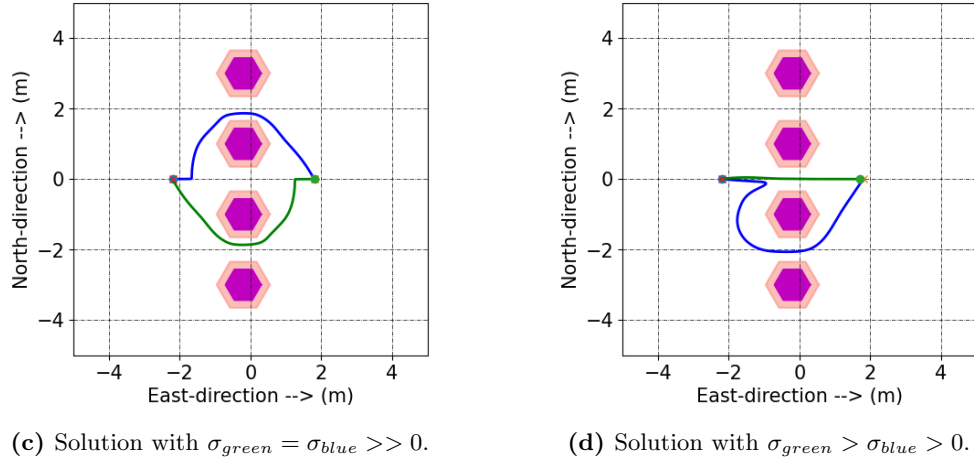


Figure 3.8: Simulations with varying source strengths on vehicles (Continued).

The collision avoidance capability of vehicles is demonstrated with an example simulation scenario given in Figure 3.8 and Figure 3.8. In this scenario, two aerial vehicles are placed opposite sides of the flight arena. The start position of one vehicle is the goal position of the other vehicle and vice versa. This way, without collision avoidance property vehicles are guaranteed to collide with each other. Initially, no source element is assigned to either vehicle. As a result, they collide mid-way, as presented in Figure 3.8a. For the second simulation, both vehicles are assigned a point source with equal source strength, $\sigma_{green} = \sigma_{blue} > 0$. In this case, vehicles avoid collision while trying to fit into the narrow passage in the center. Resultant trajectories are plotted in Figure 3.8b. In the simulations, vehicles are modeled as point masses. However, in reality, vehicles have a finite volume. Therefore, the solution presented in Figure 3.8b may not be always feasible. Alternatively, a point source element with greater strength can be assigned to both vehicles. In this case, vehicles take a detour and avoid each other from a safer distance, as plotted in Figure 3.8c. Although both vehicles travel through safer paths, pushing both vehicles away from the center line could be unnecessary. Fortunately, it is possible to assign uneven source strength to vehicles. The final case presented in Figure 3.8d is an example of such a scenario where point source elements with uneven strengths, $\sigma_{green} > \sigma_{blue} > 0$ are assigned to vehicles. In this configuration, the Green Vehicle is given priority over the Blue Vehicle. Therefore, the Blue Vehicle takes a detour while the Green Vehicle uses the shorter path through the center line. This uneven source strength assignment is beneficial when vehicles with different capabilities operate in the same flight arena. For example, this property can be utilized in urban air traffic management efforts where less maneuverable vehicles or vehicles with less endurance can be given priority over vehicles with greater endurance and maneuverability. An alternative use case for this property is assigning a medical vehicle responding to an emergency a higher priority. This way, the emergency responder would not need to alter its way to avoid traffic.

To better demonstrate the obstacle and collision avoidance capabilities of the pro-

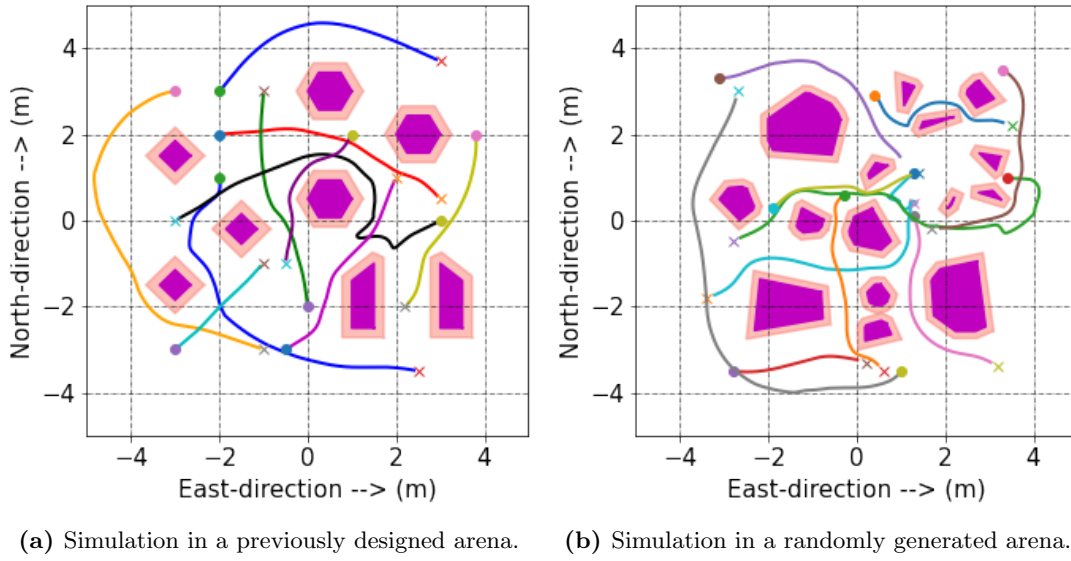


Figure 3.9: Randomized simulations with multiple aerial vehicles. All vehicles are assigned a random source strength, $\sigma > 0$.

posed PGFlow algorithm, two additional simulations with 10 vehicles are presented in Figure 3.9. In these simulations, 10 vehicles with randomized start and goal positions operate simultaneously. Vehicle dynamics are neglected in these simulations. The minimum distance between random start and goal positions is set to 1 m to eliminate trivial scenarios. In Figure 3.9a, a previously designed arena is used for the simulation. The arena is also randomly generated for the second simulation, as shown in Figure 3.9b. The randomly generated arena consists of 15 randomly placed polygonal obstacles with a random number of vertices. Each vehicle is assigned a point source element with random source strength in both scenarios. Vehicles with greater source strength have higher priority and vice versa. High-priority vehicles inhabit inner regions of the arena and travel between obstacles through shorter paths. On the contrary, vehicles with lower source strength have less priority and have to fly longer paths around the obstacles. Both simulations suggest that PGFlow algorithm can successfully solve guidance problems for multiple aerial vehicles simultaneously, even in cluttered environments.

3.5 Guidance Scenarios with Agile Flight Enhancements

The path propagation method proposed in Section 2.4 is tested with simulations. First simulations are conducted with the path propagation and the optimal path selection algorithm, then with the simple position propagation method. Both approaches are compared with baseline method in terms of maximum curvature and average curvature.

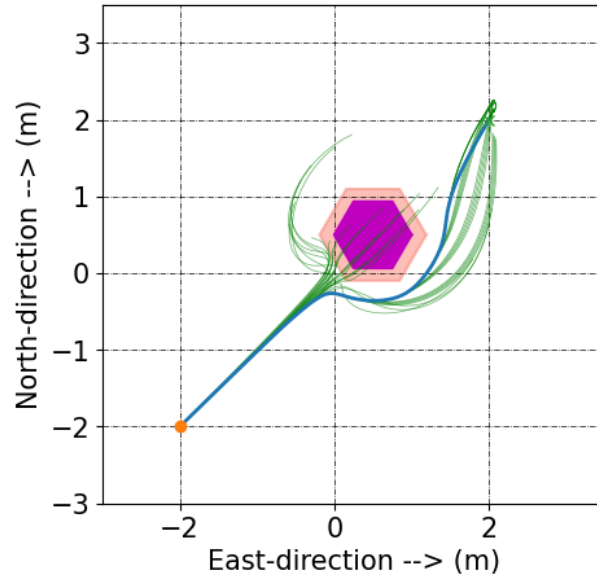


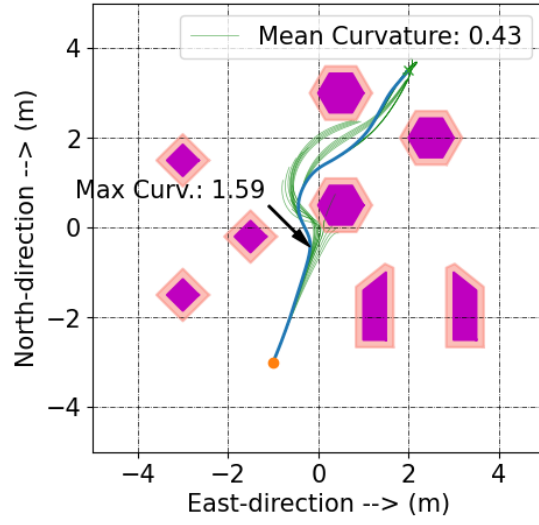
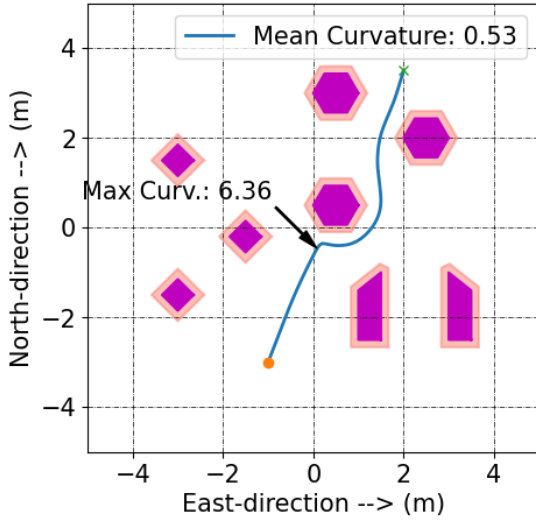
Figure 3.10: Propagated and selected path for example scenario.

3.5.1 Path Propagation

Figure 3.10 plots an example scenario with the proposed path propagation scheme. Here, the blue path is the ultimately selected path with the smallest curvature, and green paths are propagated alternative paths. In this example, paths are propagated for 2.5 seconds into the future, and propagation is done every 0.5 seconds. At each propagation step, 5 alternative paths are produced. The alternative paths take vehicle dynamics into account. Among the alternative paths, the ones causing a collision with the obstacle are discarded, and the path with the smallest curvature is selected among the safe paths.

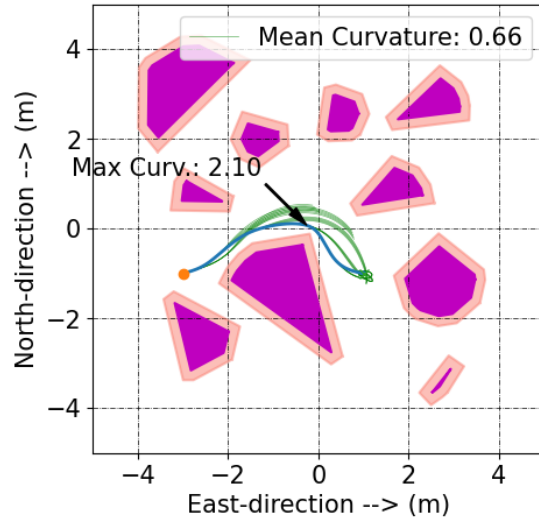
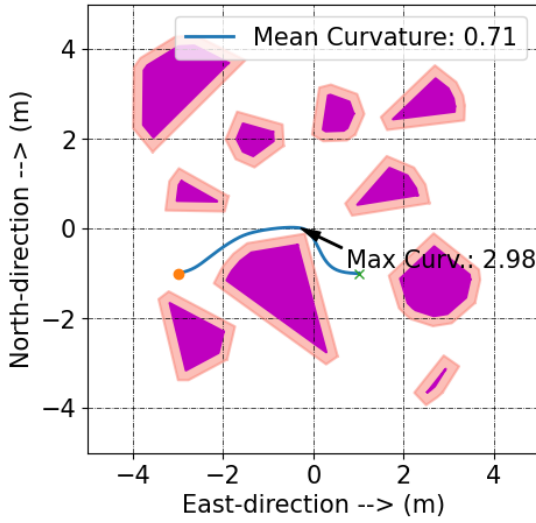
Figure 3.11 presents simulation results for a more complex scenario with multiple obstacles. Figure 3.11 compares the resultant path and its curvature with and without path propagation. The results for the nominal case without the proposed path propagation method are presented in Figure 3.11a. For this case, the mean curvature of the resultant trajectory is calculated to be 0.53. The maximum curvature occurs at the location marked with a black arrow in Figure 3.11a. The maximum curvature is calculated as 6.36 for the resultant trajectory that does not use the path propagation method. Figure 3.11b presents the resultant trajectory obtained from the path propagation method. The mean curvature is calculated as 0.43 for this trajectory. Maximum curvature is calculated to be 1.59 and occurs at a similar location.

Figure 3.12 presents results for another simulated test scenario. This time, a random arena with 10 obstacles is used for the simulation. The resultant trajectories and curvatures with and without the path propagation method are compared. The results for the nominal case without the path propagation method are illustrated in Figure 3.12a. For this case, the mean curvature of the resultant trajectory is calculated to be 0.71. The



(a) Resultant trajectory and its curvature without path propagation. (b) Resultant trajectory and its curvature with path propagation.

Figure 3.11: Simulation Scenario 1: Comparison of resultant trajectories and curvatures with and without path propagation.



(a) Resultant trajectory and its curvature without path propagation. (b) Resultant trajectory and its curvature with path propagation.

Figure 3.12: Simulation Scenario 1: Comparison of resultant trajectories and curvatures with and without path propagation.

maximum curvature occurs at the location marked with a black arrow in Figure 3.12a, with a value of 2.98. On the other hand, Figure 3.12b presents the resultant trajectory

3 Simulation Results

obtained from the path propagation method. The mean curvature for this trajectory is calculated as 0.66, with a maximum curvature of 2.10 occurring at a similar location. These simulation results underline the effect of the proposed path propagation method. The proposed method not only generates a trajectory with a smaller average curvature, but the maximum curvature is also significantly lower.

3.5.2 Position Propagation

Figure 3.13 plots an example scenario with the position propagation scheme, an alternative to computationally heavy path propagation method. Here, the blue path is the

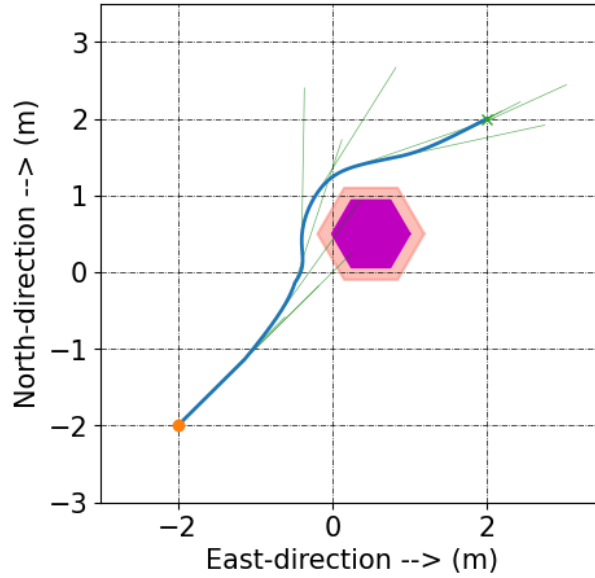
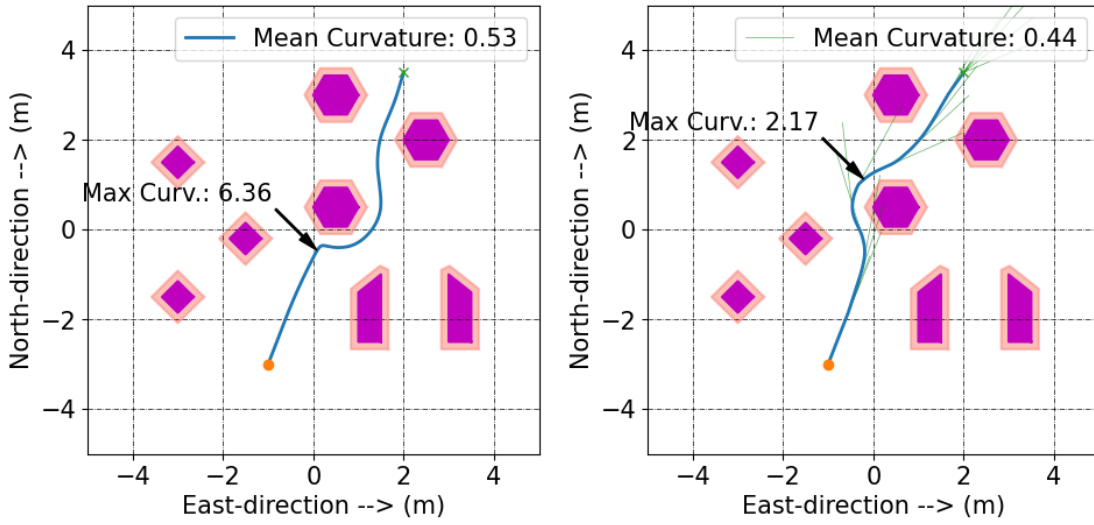


Figure 3.13: Resultant trajectory obtained from position propagation method.

resultant trajectory, and the green lines are propagated lines. In this example, paths are propagated for 3 seconds into the future, and propagation is done every 0.5 seconds. When a collision is detected, the angles ϕ and θ described in Equation 2.35 are calculated, and the uniform velocity is added to flow velocity calculations as given in Equation 2.31 and Equation 2.32.

The simulation scenarios presented for the path propagation method are repeated using the position propagation method for comparison. Figure 3.14 presents results for the first simulation scenario. Figure 3.14 compares the resultant path and its curvature with and without position propagation. The results for the nominal case without the proposed path propagation method are presented in Figure 3.14a. Figure 3.14b presents the resultant trajectory obtained from the position propagation method. The mean curvature is calculated as 0.44 for this trajectory. The mean curvature is very close to the mean curvature from the path propagation method. Both are smaller than the nominal case.



(a) Resultant trajectory and its curvature without position propagation. (b) Resultant trajectory and its curvature with position propagation.

Figure 3.14: Simulation Scenario 1: Comparison of resultant trajectories and curvatures with and without position propagation.

Maximum curvature is calculated to be 2,17. The location of the maximum curvature is different from the nominal results and path propagation results.

Figure 3.15 presents the results of a simulated test scenario with the random arena with 10 obstacles. The resultant trajectories and curvatures are compared between cases with and without the position propagation method. The results for the nominal case without the position propagation method are illustrated in Figure 3.15a. Figure 3.15b presents the resultant trajectory obtained from the position propagation method. The mean curvature for this trajectory is 0.66, with a maximum curvature of 2.25 occurring at a similar location with path propagation and the nominal cases. The mean curvature is the same as the path propagation method, and the maximum curvature is very close. Simulation results indicate that the position propagation method reduces the curvature of the path almost as well as the path propagation method with significantly lower computational cost.

3.6 Conclusions

This chapter analyzes the PGFlow guidance algorithm within a controlled environment via simulations. Conducting simulations is an essential step in algorithm development and preparation for real-world experiments. The simulation environment offers a safe and adjustable platform to evaluate performance across diverse scenarios.

Taking advantage of the flexibility of the simulation environment, an element size study and a computational load study are conducted. Element size study suggests that the

3 Simulation Results

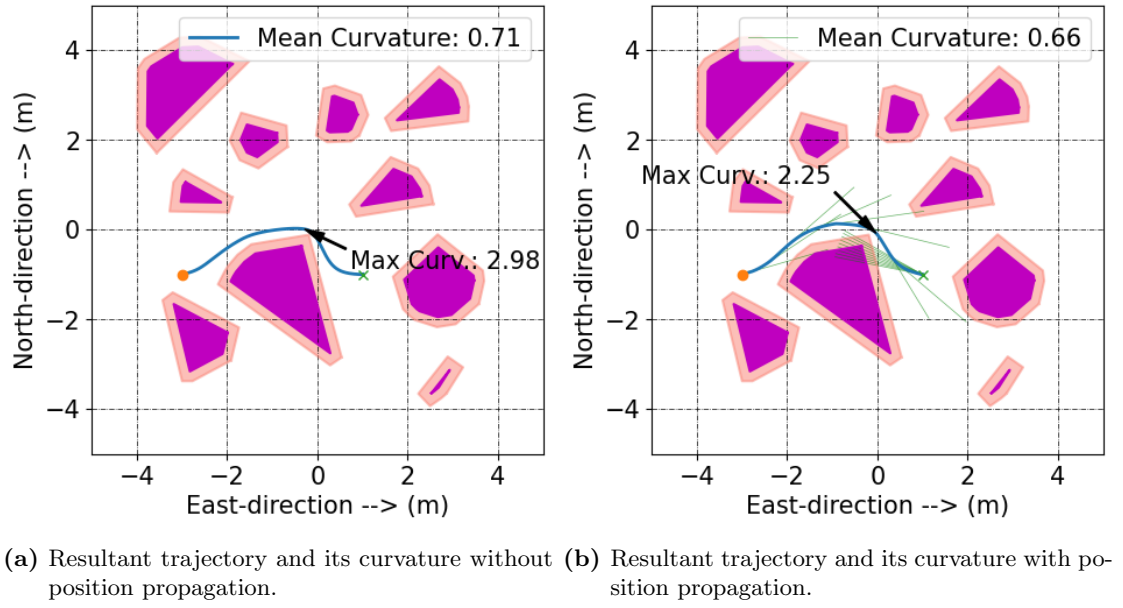


Figure 3.15: Simulation Scenario 1: Comparison of resultant trajectories and curvatures with and without position propagation.

PGFlow algorithm is sensitive to element size. An insufficient number of elements poorly represent the obstacles and hinder the obstacle avoidance property of the proposed guidance method. The computational load of a guidance algorithm plays a critical role in real-time applications and onboard processing, particularly relevant for micro aerial vehicles with limited computational resources. Vehicle and obstacle numbers are selected as the most significant parameters for the computational load study. The computational load study suggests that the computation time of the PGFlow algorithm increases linearly with an increasing number of obstacles and quadratically with an increasing number of vehicles. Results indicate that the algorithm can easily run in real-time on the given test platform with more than 140 obstacles and 35 vehicles. It is possible to improve the computational efficiency of the guidance algorithm further by running some of the computations in parallel. However, this is outside the scope of this study.

Simulation results with a single vehicle demonstrate the algorithm's obstacle avoidance and goal convergence properties. The proposed guidance method is robust against disturbances and can adapt to dynamic environments. Furthermore, the algorithm's capability to navigate around arbitrarily shaped obstacles without getting trapped in local minima is also demonstrated with simulations. Moreover, the algorithm's collision avoidance capabilities are demonstrated in simulations with multiple vehicles. Vehicles can effectively navigate in cluttered environments while avoiding collisions with each other. Additionally, the concept of assigning uneven source strengths to vehicles is introduced, providing a mechanism for prioritizing certain vehicles over others, a valuable feature for applications such as urban air traffic management or emergency response.

This chapter also presents simulation conducted using the agile flight enhancements

for the PGFlow method. The two approaches, path propagation and simple position propagation improve the smoothness of the trajectories. The proposed path propagation method calculates alternative trajectories and selects the optimal path based on minimum curvature criteria. This method not only generates smoother paths, but also considers vehicle dynamics for more accurate predictions. Simulation results demonstrate the effectiveness of this approach in reducing path curvature. On the other hand, the simple position propagation method offers a computationally efficient alternative by predicting collisions using projected line segments and initiating avoidance maneuvers. Despite its simplicity, this approach achieves notable reductions in path curvature, comparable to the path propagation approach, with significantly lower computational load. To conclude, the simulation results presented in this chapter demonstrate the effectiveness and versatility of the PGFlow algorithm in different scenarios, laying a solid foundation for further hardware experiments.

4 Experimental Setup

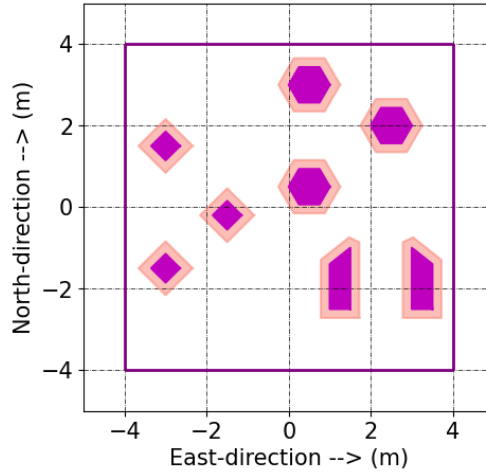
Evaluating guidance methodologies through empirical experimentation is a necessary step in validating the proposed algorithm’s feasibility. One of the main contributions of this thesis is the experimental demonstration of the proposed method’s performance in various settings on different platforms. Hardware experiments presented in this thesis are conducted in Toulouse, France, through an international collaboration effort between Technical University of Munich (TUM) and Ecole Nationale de l’Aviation Civile (ENAC). For all the experiments, ENAC’s existing infrastructure for autonomous systems is used. This chapter explains the experimental setup and experiment facilities in detail. The indoor flight test facility is introduced in Section 4.1. Along with the test facility, the test equipment and flying platforms used in the indoor facility are also explained. Section 4.2 details the automatic flight controller and its integration to guidance algorithm.

4.1 Indoor Test Facility

Indoor experiments are carried out at The Toulouse Occitanie Drone Flight Arena in Toulouse, France, as depicted in Figure 4.1a.



(a) The Toulouse Occitanie Drone Flight Arena located in Toulouse, France.



(b) 2-D cross-section of the arena.

Figure 4.1: Indoor flight facility.

The flight area, equipped with high-precision localization and measurement instruments, stands out as one of the most technologically advanced indoor flight zones in

4 Experimental Setup

Europe [85]. The indoor flight arena is $8\text{ m} \times 8\text{ m} \times 8\text{ m}$ large. Obstacles of various shapes and heights are placed within the flight area to replicate a scaled urban environment as shown in Figure 4.1a. The quad-rotor's motion is tracked with a 16-camera Optitrack system with sub-millimeter resolution in 320 Hz. Moreover, the Optitrack system can also automatically locate the obstacles in the arena.

The indoor flight facility can accommodate two different kinds of quad-rotor platforms.

4.1.1 Tello Platform

The first platform supported by the Toulouse Occitanie Drone Flight Arena is the DJI-Tello quad-rotor shown in Figure 4.2. DJI-Tello quad-rotor is an off-the-shelf flying



Figure 4.2: DJI-Tello quad-rotor.

platform that weights 87 g and can reach max speed of 8 m/s. Tello drones are equipped with 2.4 GHz 802.11n Wi-Fi and can be controlled through a ground station. The DJI-Tello quad-rotors directly interpret translation velocity commands and internally try to track the provided reference velocity by comparing it to their onboard velocity estimation derived from a downward-looking optic-flow camera. The desired velocity calculated with PGFlow algorithm is being updated at a frequency of 20 Hz as input. The velocity control is further improved by the motion capture system with 16-Optitrack cameras. Initially, the velocity error is computed between the reference velocity acquired from the PGFlow and the vehicle velocity obtained from the Optitrack camera system. Subsequently, this error is transformed into the body frame of the DJI-Tello and transmitted as a reference to be tracked through the DJI-Tello's API.

4.1.2 Explorer Platform

Another quad-rotor platform supported by the indoor flight facility is the Explorer platform illustrated in Figure 4.3. With the battery included, it has a mass of 400 g and can generate a maximum thrust of 40 N. Powered by a 3-cell battery operating at 11.1 Volt

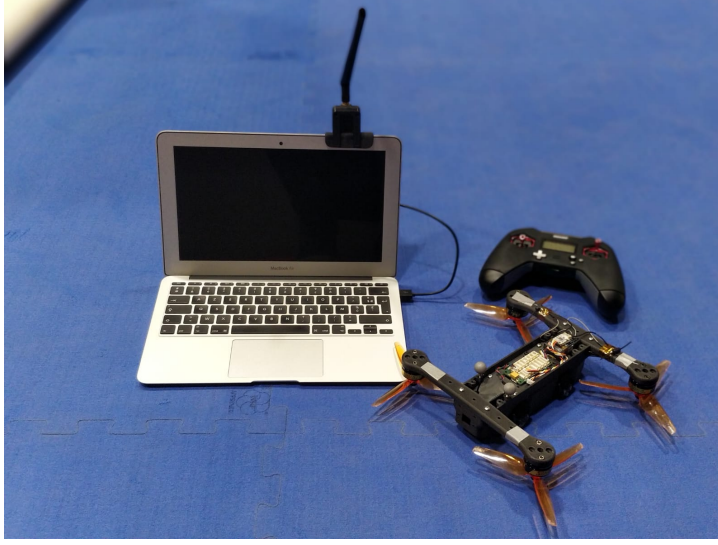


Figure 4.3: Ground control station and quad-rotor Explorer 1.

with a capacity of 2.700 mAh, it offers approximately 14 minutes of flight time. Table 4.1 presents detailed specifications of Explorer 1. Note that this platform also has an outdoor counterpart, explained in detail in Section 6.2.1. Similar to the Tello platform, the velocity control for the indoor Explorer is also achieved through the Optitrack system. The Optitrack system senses the vehicle's position and attitude, which are then relayed to the control computer. Subsequently, the control computer computes the desired velocities and transmits this information to the vehicle, along with the vehicle's current position, via wireless communication. The communication scheme for the indoor test environment is sketched in Figure 4.4.

4.1.3 Paparazzi Autopilot System

Throughout the flight test campaign, the *Paparazzi* Autopilot system is utilized [86]. This system was initiated in 2003 as an open-source project and has been adopted by numerous research groups, academics, and hobbyists ever since. As one of the earliest open-source autopilot systems globally, *Paparazzi* includes all three segments: ground, airborne, and the communication link between them.

Paparazzi also features a complete flight plan language, enabling users to define various trajectories using existing commands such as circle, line, hippodrome, figure-eight, survey, and more. Moreover, users can call and execute any function written in C language from the flight plan. This flexibility offers a wide range of application possibilities, such as triggering navigation procedures via sensor outputs. In addition, *Paparazzi*'s integrated ground control station allows for the control of flight plan execution, waypoint adjustment, or modification of aircraft parameters during flight.

4 Experimental Setup

Specification	Units	Paparazzi Quadrotor	
		Indoor	Outdoor
Mass	[g]	400	440
Maximum Flight Speed	[m/s]	4	14
Flight Duration	[min]	14	12
Localization		Optitrack	GPS
Autopilot Software		Paparazzi V.6[86]	
Autopilot Hardware		Tawaki V1.1	
Controller		INDI[87]	
Communication		XBee 2.4 GHz	
Motors		T-Motor F40 2400 Kv	
Propellers		T-Motor F5146	
Battery		11.1V 2.7 Ah LiPo	

Table 4.1: Specifications of the Explorer platform, including the effect of additional hardware parts and software limitations.

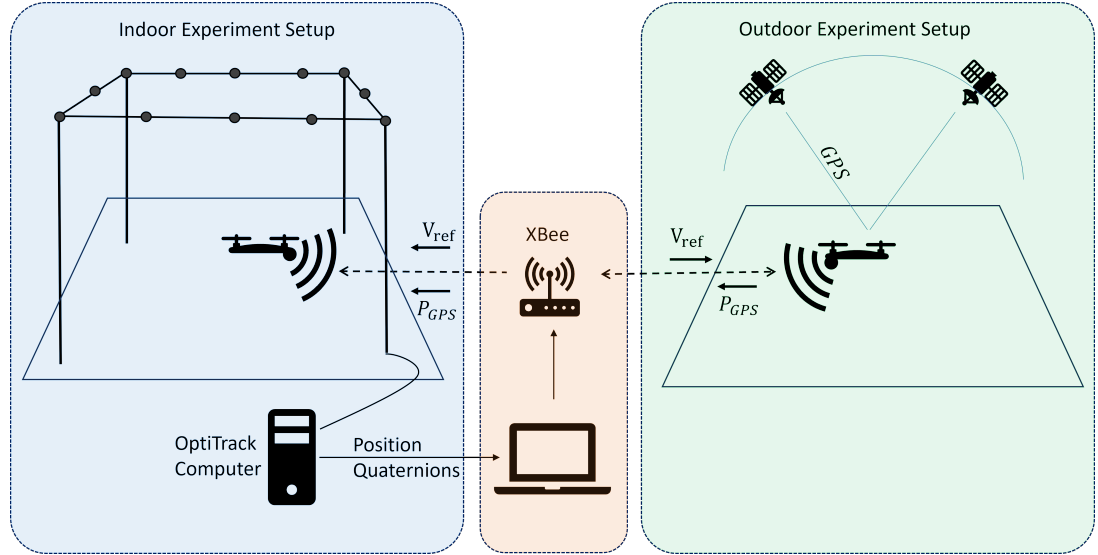


Figure 4.4: Communication Scheme for Indoor and Outdoor Experiment Setup.

4.2 Automatic Flight Controller Integration

In this section, PGFlow integration with automatic flight controller is detailed. First, the flight controller used in this study is inspected. Then, the integration of the automatic flight controller and the guidance algorithm is presented.

4.2.1 Automatic Flight Controller

The flight controller used in this study is based on the control of the angular accelerations of the vehicle in an *incremental* way. The control method used in this thesis is based on the work of Smeur et. al.[87]. The basic working principle of the proposed controller is presented here, reader may refer to [87] for a more detailed derivation of the controller law.

Consider the angular acceleration of the vehicle, neglecting the gyroscopic moments, it

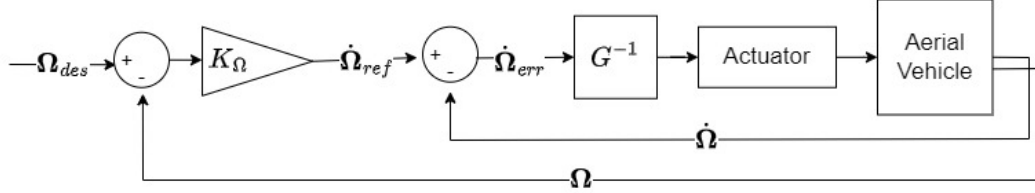


Figure 4.5: Block diagram for Incremental Nonlinear Dynamic Inversion (INDI) controller.

is a function of the angular rate Ω , airspeed v , and the input vector u as presented in Equation 4.1, [87].

$$I\dot{\Omega} = f(\Omega, v) + g(u) \quad (4.1)$$

Here I is the moment of inertia matrix, Ω is the angular rate vector and v is the airspeed vector, and u is the input vector. f and g are functions. Given a desired angular acceleration vector, the functions f and g need to be known accurately, in order to invert Equation 4.1 for the input u . Instead, a first-order Taylor series expansion of the functions f and g in Equation 4.1 is considered as in Equation 4.2.

$$\begin{aligned} I\dot{\Omega} = I\dot{\Omega}_0 + \frac{\partial}{\partial \Omega} f(\Omega, v_0)|_{\Omega=\Omega_0}(\Omega - \Omega_0) + \frac{\partial}{\partial v} f(\Omega_0, v)|_{v=v_0}(v - v_0) \\ + \frac{\partial}{\partial u} g(u)|_{u=u_0}(u - u_0) \end{aligned} \quad (4.2)$$

Here, the angular acceleration $\dot{\Omega}$, is expressed in terms of the current angular acceleration Ω , the partial derivatives of the functions f and g and the change in state and inputs. If a small time interval only long enough for the actuators to adopt their new values is considered, the change in Ω and v and their derivatives can be neglected. Moreover, it is possible to assume the partial derivative of g with respect to u is a static matrix for some part of the flight envelope. Then, dividing the Equation 4.2 by I , the angular acceleration can be approximated as in Equation 4.3, [87].

$$\dot{\Omega} = \dot{\Omega}_0 + G(u - u_0) \quad (4.3)$$

Here G is the control effectiveness matrix, which contains the effectiveness of each actuator on each axis. The values in the G matrix are determined by a least squares fit using changes in input and angular acceleration obtained from real flight data.

Finally, the control law can be obtained by inverting Equation 4.3.

$$u = u_0 + G^{-1}(\dot{\Omega} - \dot{\Omega}_0) \quad (4.4)$$

4 Experimental Setup

Here, the desired angular acceleration, $\dot{\Omega}$, becomes the virtual control ν . A Proportional-Derivative controller provides the virtual control; that is, the desired angular acceleration can be designed to control the attitude.

A block diagram for tracking a desired velocity input using the INDI controller is presented in Figure 4.5. A more detailed explanation of controller design can be found in [87].

4.2.2 Automatic Flight Controller and Guidance Algorithm Integration

A block diagram for PGFlow algorithm integrated with automatic flight controller is presented in Figure 4.6. The input of the automatic flight controller integrated system

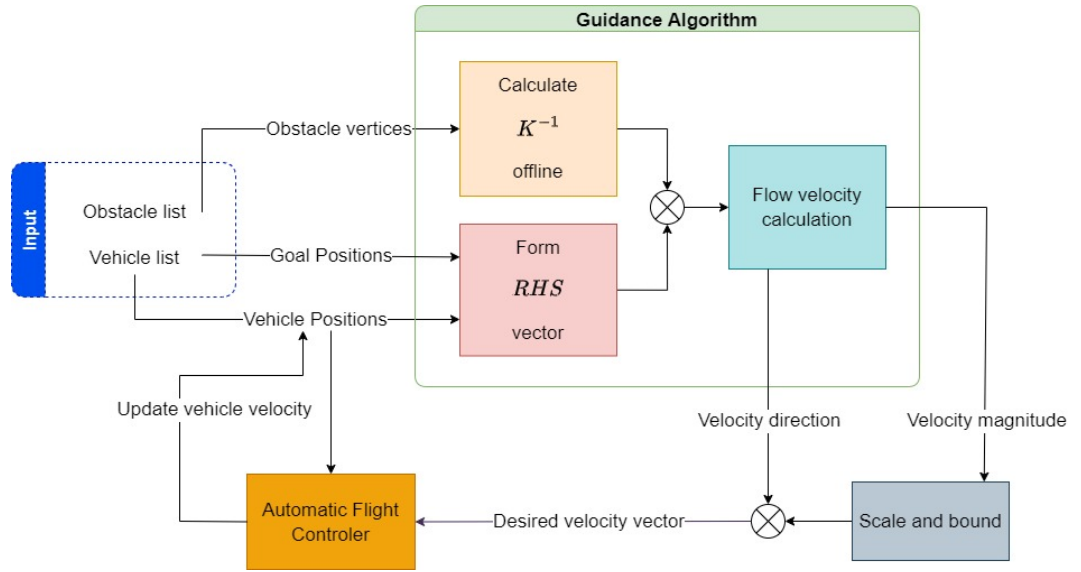


Figure 4.6: Block diagram for Potential Guidance Flow integrated automatic flight controller.

is the obstacle list and the vehicle list. At the initialization, obstacle vertices obtained from the obstacle list are used for the calculation of the inverse of the coefficient matrix K^{-1} in the Equation 2.12. The vehicle list provides the initial positions and preassigned goal positions for each vehicle. Each goal position is assigned a point sink element and each vehicle is assigned a point source element to repel other vehicles and a *safety source* element. Known sink and source elements are used for forming the RHS vector as given in Equation 2.26. After obtaining the inverse of the coefficient matrix K^{-1} and the RHS vector, Equation 2.12 can be solved for the unknown vortex strengths, γ . After γ are found, Equation 2.21 and Equation 2.22 are used for flow velocity calculation. The calculated flow velocity vector is separated into magnitude and unit direction vector components. The velocity magnitude is calculated by taking the Euclidean norm of the flow velocity vector as given in Equation 4.5.

$$\|\mathbf{V}\| = \sqrt{u^2 + v^2} \quad (4.5)$$

4.2 Automatic Flight Controller Integration

Where, $\mathbf{V} = (u, v)$ denotes the non-zero instantaneous flow velocity at any point of interest. Velocity magnitude has to be scaled down in proportion to arena size and can be bounded to take vehicle limits into account. Using the flow velocity magnitude $\|\mathbf{V}\|$, the normalized flow velocity vector can be written as $\bar{\mathbf{V}} = \mathbf{V}/\|\mathbf{V}\|$.

The desired velocity to be fed into the automatic flight controller can be calculated as in Equation 4.6.

$$\mathbf{V}_{des} = c\|\mathbf{V}\|\bar{\mathbf{V}} \quad (4.6)$$

Here, c is a scaling constant that depends on arena size and maximum allowable speed for vehicles. $\mathbf{V}_{des}$ obtained in Equation 4.6 is then used as input to the automatic flight controller as described in the block diagram presented in Figure 4.5. Here, the velocity error is multiplied by a gain before it is fed to the INDI controller as reference acceleration $\dot{\boldsymbol{\Omega}}_{ref}$, which acts as the virtual control input.

5 Experimental Validation on VTOL Platforms

Validation in real-world settings that genuinely confirms the effectiveness and feasibility of new algorithms and approaches. This chapter inspects the experiments performed on Vertical Take-Off and Landing (VTOL) platforms, particularly quad-rotors, to evaluate the feasibility of the PGFlow algorithm. Through an extensive experiment campaign, the chapter aims to verify the algorithm's performance in both static goal positioning and dynamic target tracking scenarios.

First, Section 5.1 discusses experiments focused on static goal positioning. In these experiments, quad-rotors navigate indoor environments to reach predefined goal positions while maneuvering around obstacles. Both single and multiple vehicle scenarios are explored, demonstrating the PGFlow algorithm's adaptability and scalability. Furthermore, in Section 5.2, dynamic target tracking experiments, tasking quad-rotors with dynamically following moving targets are inspected. Section 5.3 presents the hardware experiments conducted using the agile flight enhancements. Finally, in Section 5.4 experiments in 3 dimensional space with multiple vehicles are presented. Conclusions drawn from the experimental findings are delivered in Section 5.5 where the PGFlow algorithm's effectiveness and feasibility is discussed.

5.1 Static Goal Positions

This section inspects the obstacle and collision avoidance properties of the proposed PGFlow algorithm. Initial tests are conducted in a controlled environment with simple scenarios to set a baseline for upcoming experiments in this thesis work. In this section, vehicles traveling to set goal positions are considered. The static goal position is modeled as a point sink element in PGFlow equations, and the desired velocity vector leading to this pre-specified goal position is calculated in real-time for all the vehicles in the flight arena simultaneously. Experiments are first conducted for a single quad-rotor and then expanded to multiple quad-rotors.

5.1.1 Single Vehicle

This section presents two different experiment scenarios conducted with a single aerial vehicle in the indoor flight facility. Both scenarios aim to demonstrate the obstacle avoidance capability of the proposed PGFlow algorithm.

The first experiment scenario revisits the simulation scenario presented in Section 3.3, Figure 3.7. In this scenario, a virtual obstacle blocks a narrow passage in an attempt

to trap the vehicle inside the corridor. Figure 5.1 presents the results of the first exper-

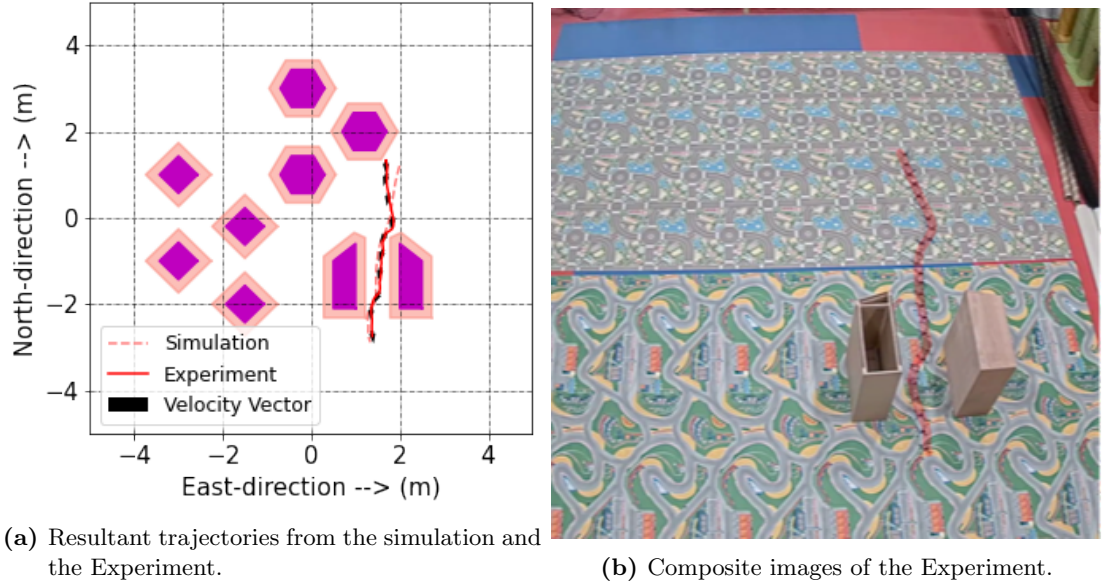
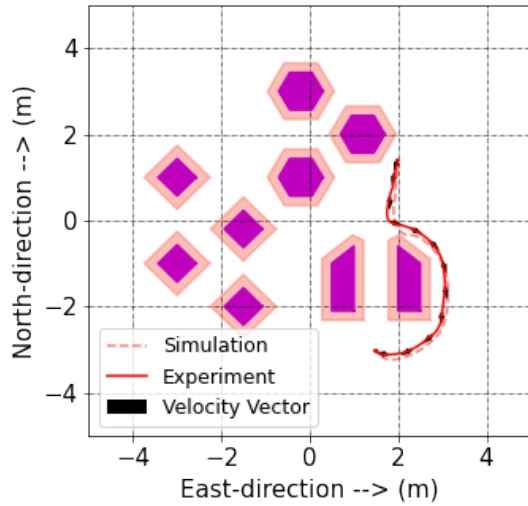


Figure 5.1: Experiment with a single vehicle through an open passage.

iment scenario. For the first case, the passage connecting the start and goal positions is not blocked. In Figure 5.1a, the resultant trajectory from both the simulation and the hardware experiment are plotted together. The dashed line represents the simulation result, while the solid line is the trajectory followed by the aerial vehicle in the Experiment. In addition to the resultant trajectory, desired velocity vectors commanded to the vehicle are overlaid on the actual trajectory. In this experiment, the vehicle was placed in the start position manually. As manual placement was not precise, there is a slight difference between the simulated trajectory and the actual path followed in the Experiment. Even though the starting position was not precise, the guidance vector field generated by PGFlow algorithm carries the vehicle to the desired destination. In Figure 5.1b, composite images generated with camera snapshots of the hardware experiment are presented. The second case in this Experiment with a virtually blocked path is given in Figure 5.2. Similarly, in Figure 5.2a, the resultant trajectory from both the simulation and the hardware experiment are plotted together. The composite images generated with camera snapshots of the hardware experiment are presented 5.2b. In this case, too, there is a strong correlation between simulated results and the hardware experiment. The vehicle successfully avoids obstacles and arrives at its desired destination in both cases.

In the second scenario with a single vehicle, the vehicle travels between 5 different locations sequentially for 9 minutes. Figure 5.3 presents the experiment results for this scenario. In Figure 5.3a, a single lap is plotted, and destinations are marked with stars and named with letters. The vehicle starts from position E and travels to positions A-B-C-D in this order before returning to position E again. Although the vehicle travels



(a) Resultant trajectories from the simulation and the Experiment.



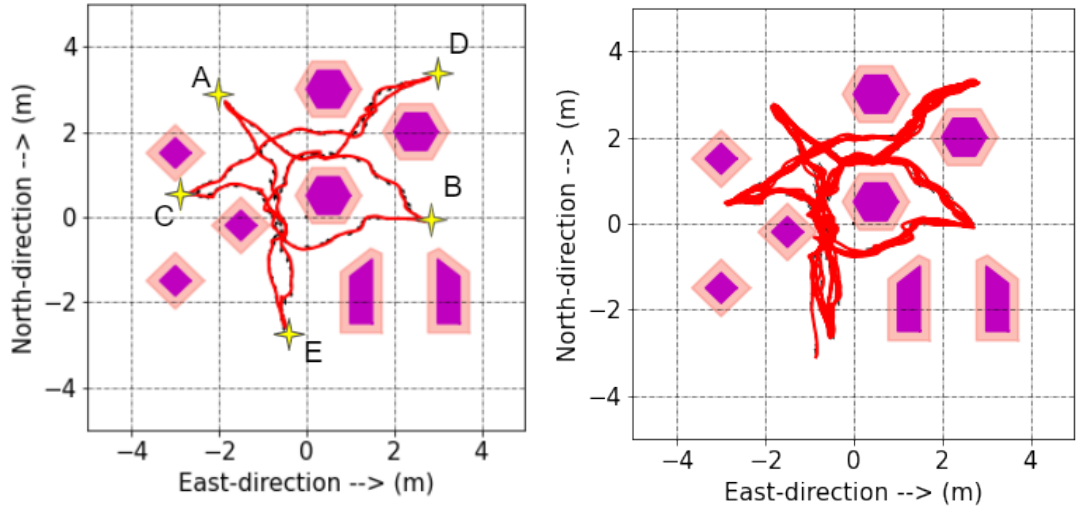
(b) Composite images of the Experiment.

Figure 5.2: Experiment with a single vehicle through a blocked passage.

between 5 different destinations, only a single point sink element is active at all times. When the vehicle reaches its current destination, the point sink element located at that position is deactivated, and the sink element at the next goal position is activated. Goal locations are strategically chosen and ordered to force the aerial vehicle into narrow corridors between obstacles. This selection aims to effectively demonstrate the obstacle avoidance capability of the PGFlow algorithm. In Figure 5.3b, the trajectory resulting from a 9-minute flight of the vehicle is illustrated. It is evident in Figure 5.3b that the vehicle consistently steers clear of obstacles, ensuring collision avoidance, and reliably reaches the intended goal position. In Figure 5.3c, composite images captured during the Experiment are presented.

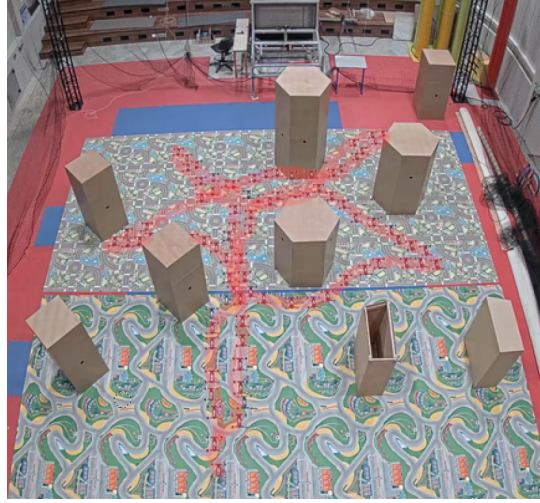
5.1.2 Multiple Vehicles

One notable strength of the PGFlow algorithm is its capacity to simultaneously generate collision-free paths for multiple vehicles in real-time. This section presents two scenarios involving multiple vehicles in the same flight arena. For both scenarios, aerial vehicles have their own starting position and destination. All vehicles are assigned a point source element with constant strength to prevent collisions between vehicles. The point source element is included in calculations of vortex strengths on obstacle surfaces. This way, vehicles cannot repulse each other into obstacles. This approach provides both obstacle avoidance and collision avoidance. Resultant vehicle trajectories for the first example scenarios are plotted in Figure 5.4. Figure 5.4a shows a 2-D cross-section of the indoor flight arena. The resultant trajectories from the simulation for three aerial vehicles are plotted alongside the actual paths traveled by the vehicles during the hard-



(a) Single trajectory and goal positions.

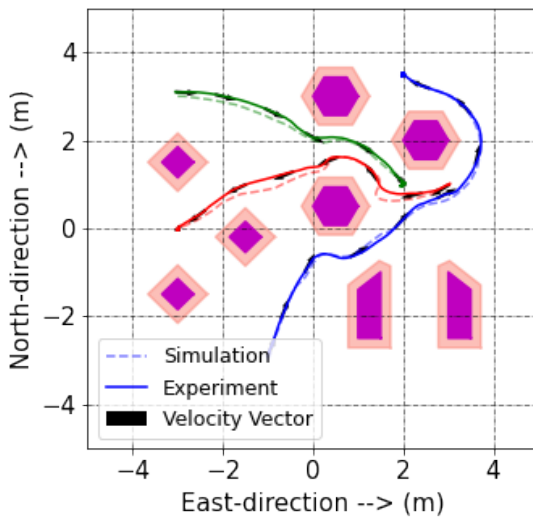
(b) Trajectories for 9 minute flight.



(c) Captured images of the Experiment.

Figure 5.3: Experiment 1. The vehicle is tasked with circling between positions marked with E-A-B-C-D-E for 9 minutes while avoiding obstacles.

ware experiment in Figure 5.4a. The dashed lines represent the resultant trajectories generated in the simulation, while the solid lines are the resultant trajectories of the vehicles in the Experiment. Additionally, black arrows plotted along the trajectories indicate the desired velocity vectors commanded to the vehicles at specific time steps. The difference between the simulated trajectories and the hardware experiment trajectories is primarily due to the initial imprecise positioning of the vehicles at experiments. In this experiment, vehicles were placed manually in their starting positions. In hardware experiments involving multiple aerial vehicles, when a vehicle is placed at a different start position from the simulation, it not only follows a different streamline but also,



(a) The resultant trajectories.



(b) Captured images of the Experiment.

Figure 5.4: Multiple Vehicle Experiment 1. Three aerial vehicles travel to different destinations while avoiding collision with obstacles and each other.

due to the repulsive forces between the vehicles, causes other vehicles to generate varied paths even if they have started from the exact location as in the simulation. Nevertheless, results of the Experiment presented in Figure 5.4 indicate that irrespective of their starting positions, each vehicle successfully reaches its destination without any collisions with obstacles or other vehicles. Composite images generated with camera snapshots for experiment 1 are shown in Figure 5.4b. Figure 5.5 illustrates the desired velocity input

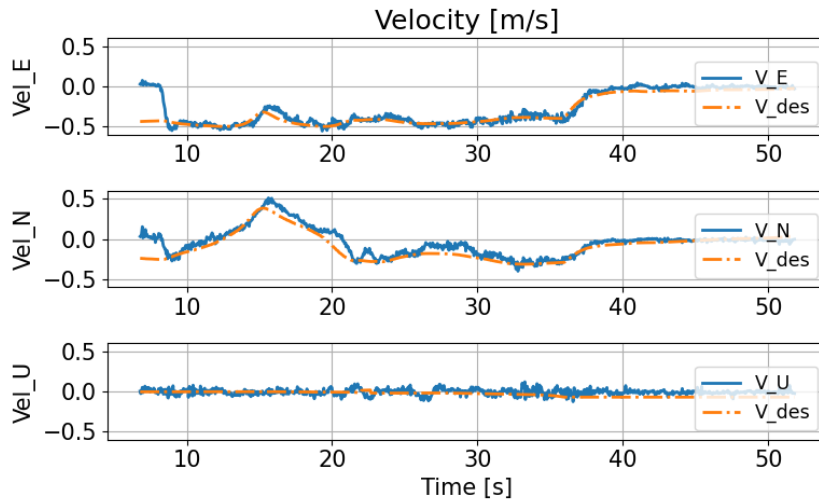


Figure 5.5: Multiple Vehicle Experiment 1. The desired velocity and measured velocity of the Red Vehicle.

5 Experimental Validation on VTOL Platforms

and the measured velocity of the Red Vehicle of the Multiple Vehicle Experiment 1. The orange dashed lines are the desired velocity input in East-North-Up directions, and the blue solid lines are the actual measured velocities. In this Experiment, the desired velocity is constrained to 0.5 m/s to prevent the vehicle dynamics from dominating the resultant trajectories.

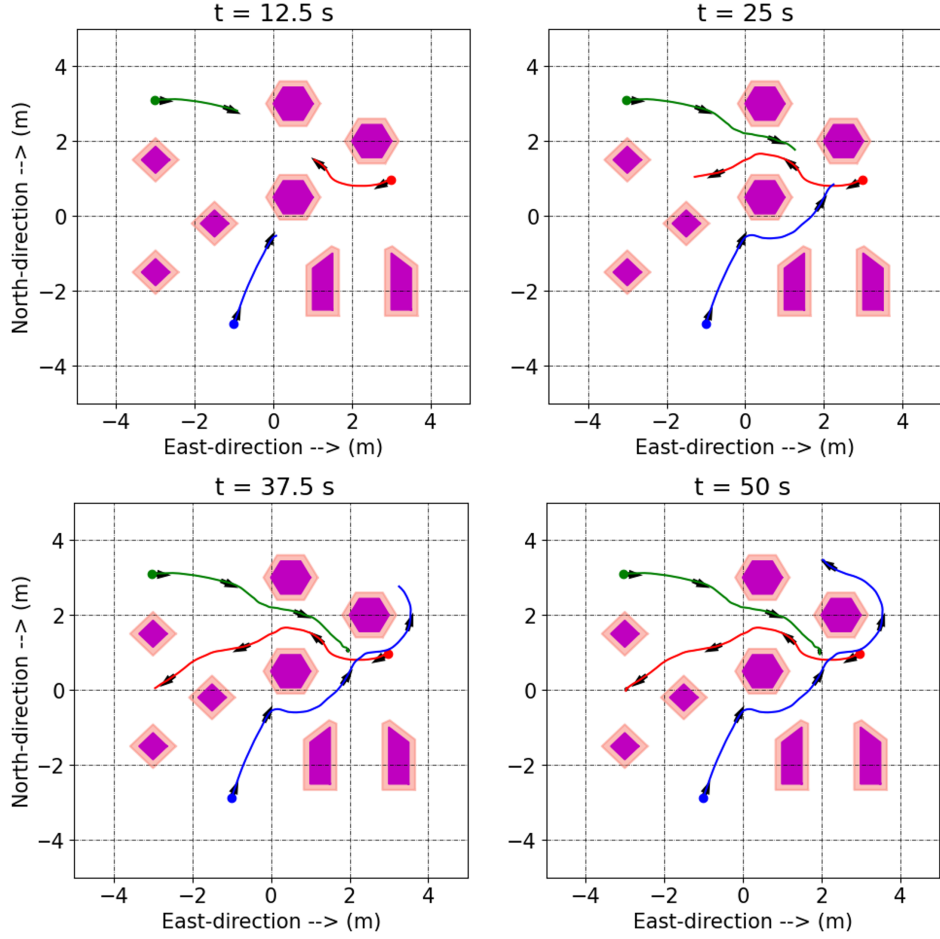
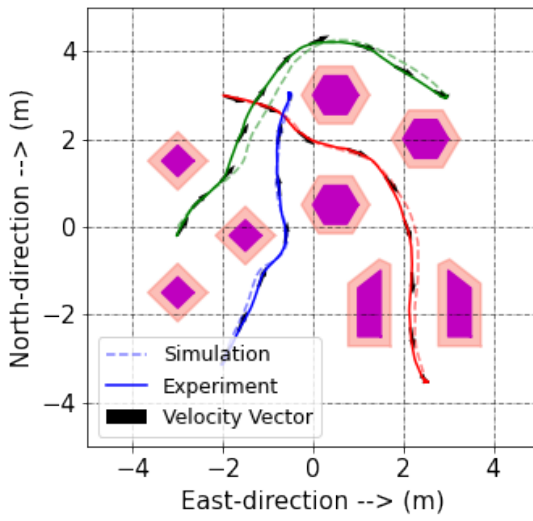


Figure 5.6: Multiple Vehicle Experiment 1. Sequential images of trajectories followed by the vehicles.



(a) Simulation and experiment trajectories.



(b) Captured images of Experiment.

Figure 5.7: Multiple Vehicle Experiment 2. Three aerial vehicles travel to different destinations simultaneously while avoiding collision.

Figure 5.6 provides sequential plots of the resulting trajectories for the three vehicles in experiment 1. Snapshots of the resultant trajectories are taken at $t = 12.5$ s, $t = 25$ s, $t = 37.5$ s, and $t = 50$ s. Furthermore, the desired velocity vectors fed to the aerial vehicles are also plotted in Figure 5.6. These sequential plots indicate there are no collisions between vehicles.

The trajectories of the vehicles in the second experiment scenario are plotted in Fig-

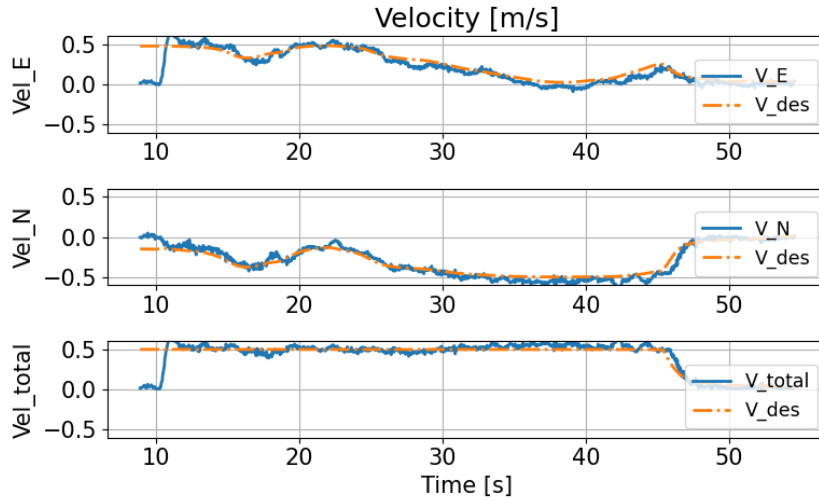


Figure 5.8: Multiple Vehicle Experiment 2. The desired velocity and measured velocity of the Red Vehicle.

Figure 5.7. Figure 5.7a illustrates a 2-D cross-section of the indoor flight arena. The simulation's resultant trajectories for three aerial vehicles are compared with the actual paths they followed during the hardware experiment in Figure 5.7a. The dashed lines indicate the simulated resultant trajectories, and the solid lines represent the actual paths of the vehicles during the Experiment. Furthermore, black arrows plotted along the trajectories are the desired velocity vectors commanded to the vehicles. The results of the Experiment presented in Figure 5.7 indicate PGFlow algorithm successfully guides each vehicle to its destination without any collisions with obstacles or other vehicles. Composite images generated with camera snapshots for the experiment 1 is shown in Figure 5.7b.

Figure 5.8 shows the desired velocity input and the measured velocity of the Red Vehicle of the Multiple Vehicle Experiment 2. Velocities are presented in the East-North-Up coordinate system. The orange dashed lines are the desired velocity input, and the blue solid lines are the actual measured velocities. Similar to experiment 1, in this Experiment, too, the desired velocity is constrained to 0.5 m/s to prevent the vehicle dynamics from dominating the resultant trajectories. Figure 5.9 presents sequential plots of the resulting trajectories for the three vehicles in experiment 2. Snapshots of the resultant trajectories are taken at $t = 12.5$ s, $t = 25$ s, $t = 37.5$ s, and $t = 50$ s. Furthermore, the desired velocity vectors fed to the aerial vehicles are plotted in Figure 5.6. Even though the resultant paths seem to cross each other in Figure 5.7, the sequential plots given in Figure 5.9 indicate no collisions between vehicles.

Both results from Experiment 1 and Experiment 2 demonstrate the collision avoidance property of the proposed PGFlow algorithm. Moreover, by including the repulsive point source component in obstacle vortex strength calculations, obstacle avoidance is also satisfied together with collision avoidance.

5.2 Dynamic Target Tracking

After demonstrating the collision avoidance property of the proposed PGFlow algorithm with multiple vehicles with static goal positions, a more challenging task is considered. In this section, each aerial vehicle tracks the same dynamic target. Tracking a moving target turns the guidance problem into a dynamic one where the goal position (the dynamic target) needs to be updated as it moves in the arena for vehicles to follow.

For the following experiments, the moving target's position is known by all the vehicles at all times. All the vehicles are tasked with following the target while avoiding collision. In this sense, the target can also be considered as a leader to be followed by all the agents in the arena. A human operator changes the position of the target manually. Vehicles can be released from any location within the indoor arena and follow the target (leader) wherever they are. In this thesis, four different experiment scenarios are considered. Sequential vehicle trajectories for Target Tracking Experiment 1 are plotted in Figure 5.10. In this figure, the start position of vehicles is marked with a circle, and the position of the target (leader) is marked with a star. Arrows on the vehicle trajectories indicate the desired velocity fed to the vehicles at given positions. Snapshots

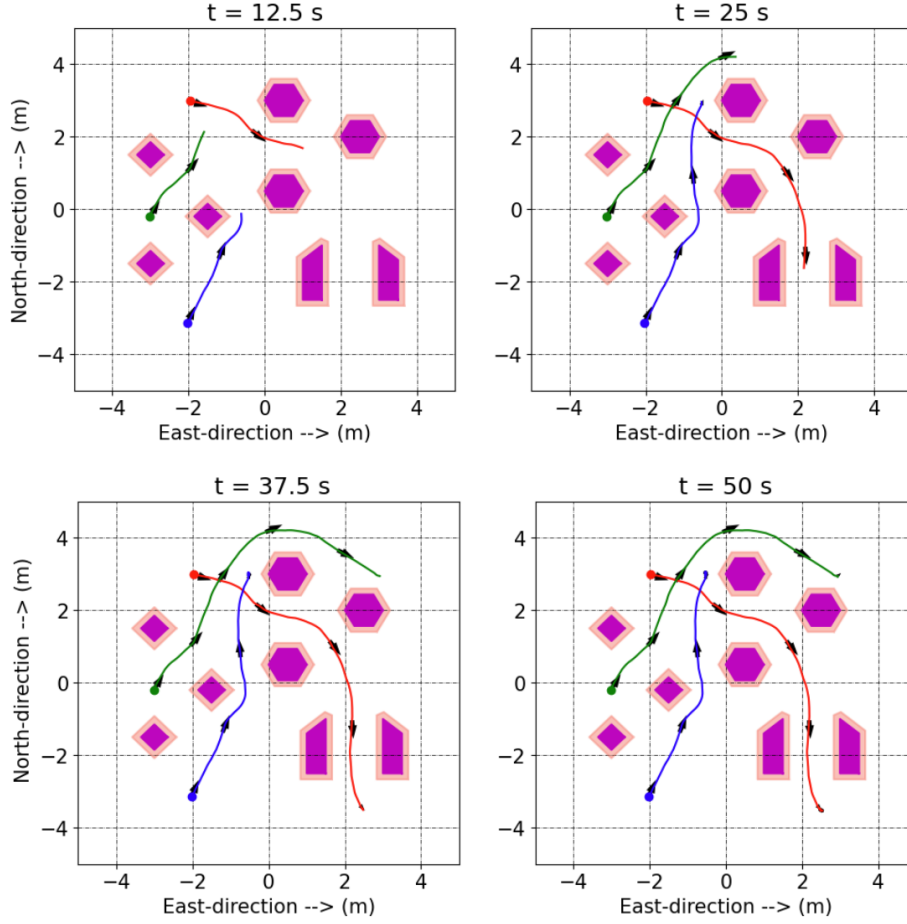


Figure 5.9: Multiple Vehicle Experiment 2. Sequential images of trajectories followed by the vehicles.

of the resultant trajectories of vehicles and the target (leader) are taken at $t = 17.5$ s, $t = 35$ s, $t = 52.5$ s, and $t = 70$ s. In this scenario, three vehicles start from the north side of the indoor flight arena, and the leader (target) starts approximately 2 m away from the vehicles towards the center. The leader moves in a J-shaped trajectory between the buildings and ends its movement in the passage between the trapezoid obstacles. As they follow the leader (target), vehicles avoid obstacles and collision with each other. For example, at $t = 35$ s, the Green Vehicle changes its direction to avoid collision with the Blue Vehicle. Similarly, the Red Vehicle too alters its path to avoid collisions. However, in the end, all three vehicles converge to the target position using different paths. Sequential vehicle trajectories for Target Tracking Experiment 2 are plotted in Figure 5.11. Snapshots of the resultant trajectories of vehicles and the target (leader) are taken at $t = 20$ s, $t = 40$ s, $t = 60$ s, and $t = 80$ s. In this scenario, three vehicles start from the north side of the indoor flight arena. Their start positions are marked with circles in Figure 5.11. The leader (target) starts from the center of three hexagons. The position

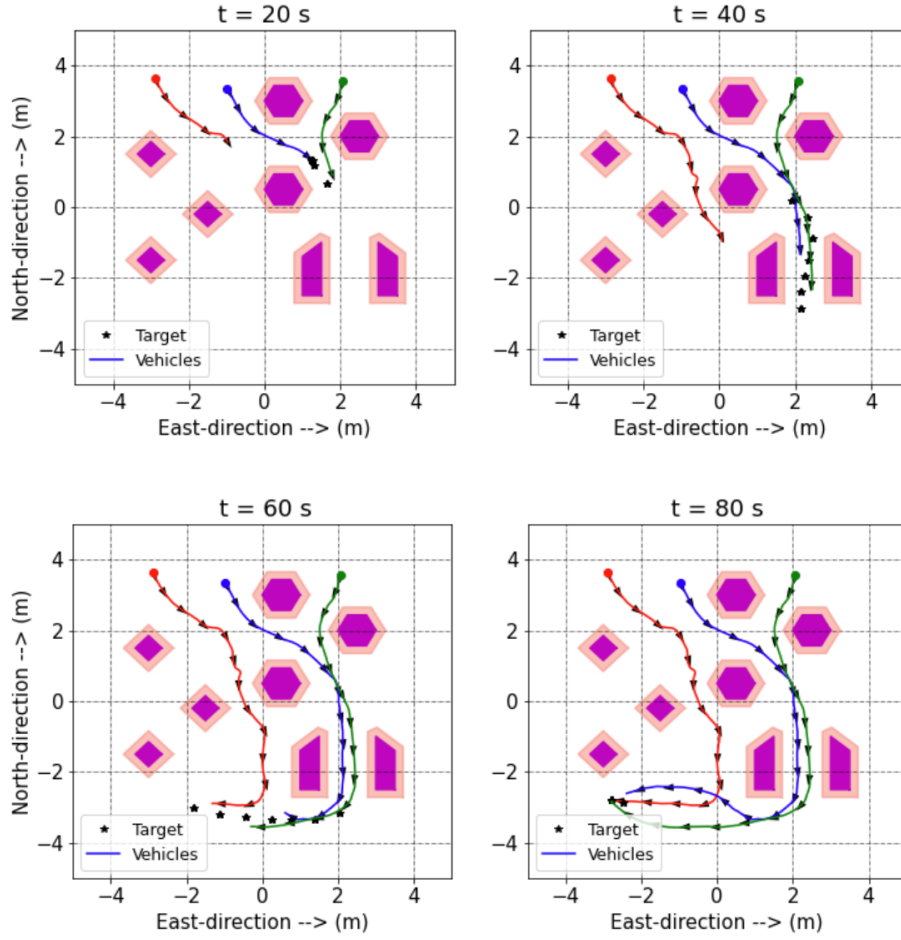


Figure 5.11: Sequential vehicle trajectories for Target Tracking Experiment 2.

the leader (target) with a constant speed of 0.4 m/s. In this Experiment, three vehicles start from north of the arena, and their start positions are marked with circles. The point source elements assigned to vehicles prevent collisions when two vehicles end up in the same corridor. One of the strengths of the proposed PGFlow algorithm is its low computational load. This property enables the proposed method to handle dynamic problems such as target tracking. Since calculating only a single guidance vector for each vehicle is enough for both collision avoidance and goal convergence, the proposed PGFlow algorithm can adapt to dynamically changing environments. For example, at $t = 50$ s, the leader (target), which is moving at a faster speed than the vehicles, changes direction. At this instant, the Blue vehicle, having entered the passage between trapezoid obstacles, is in a disadvantaged position. However, as the leader (target) moves, the guidance vectors are recalculated, and the Blue Vehicle exits the narrow passage and heads towards the leader (target) again. The fourth case considered in this section is plotted in Figure 5.13. It is a more complicated scenario where, instead of a simple

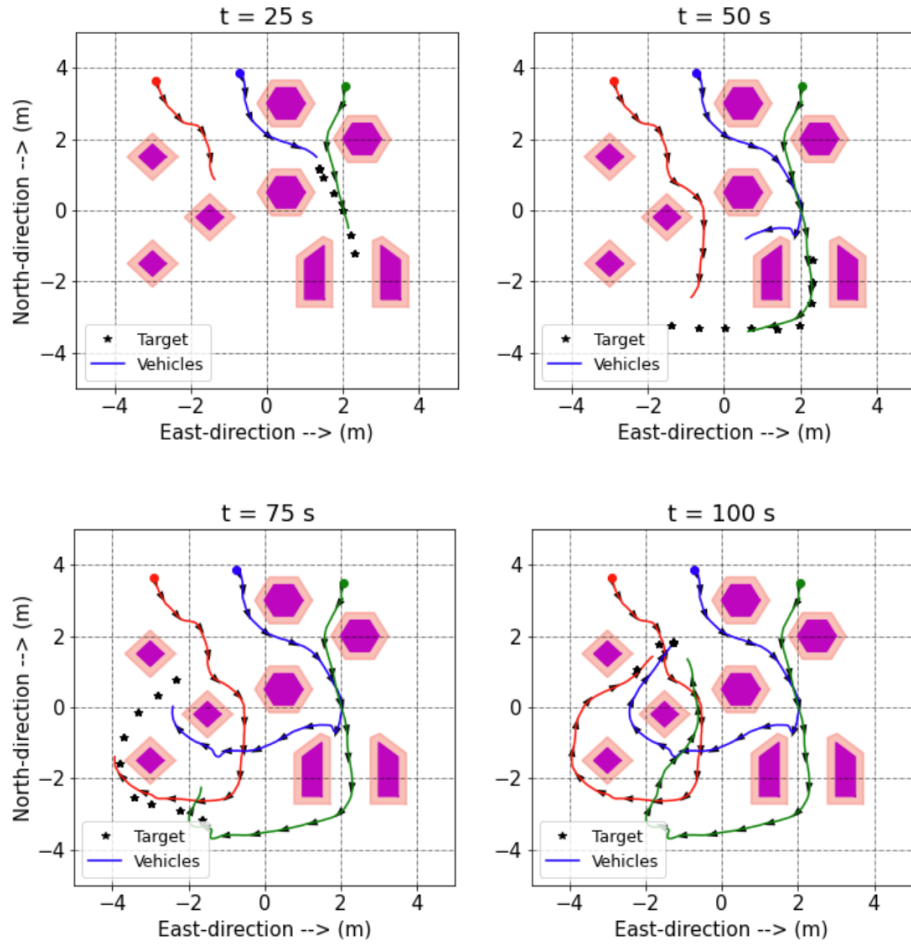


Figure 5.12: Sequential vehicle trajectories for Target Tracking Experiment 3.

trajectory, the leader (target) travels between the obstacles for 2 minutes. Furthermore, unlike the previous experiments, vehicles start from different sides of the map for this scenario. Even in this complex scenario, vehicles successfully track the dynamic target while avoiding collision with obstacles and each other.

Finally, a video link for another complex experiment scenario with three vehicles and a dynamic target is presented in the following link [88]. In this additional Experiment, too, the target tries to avoid vehicles by randomly moving around in the arena. Three vehicles managed to corner the target more than once throughout the Experiment with their collaborative effort while avoiding collision with obstacles and each other.

5.3 Agile Flight Enhancements

This section presents hardware experiments conducted using the agile flight enhancements described in Section 2.4. Both path propagation and position propagation meth-

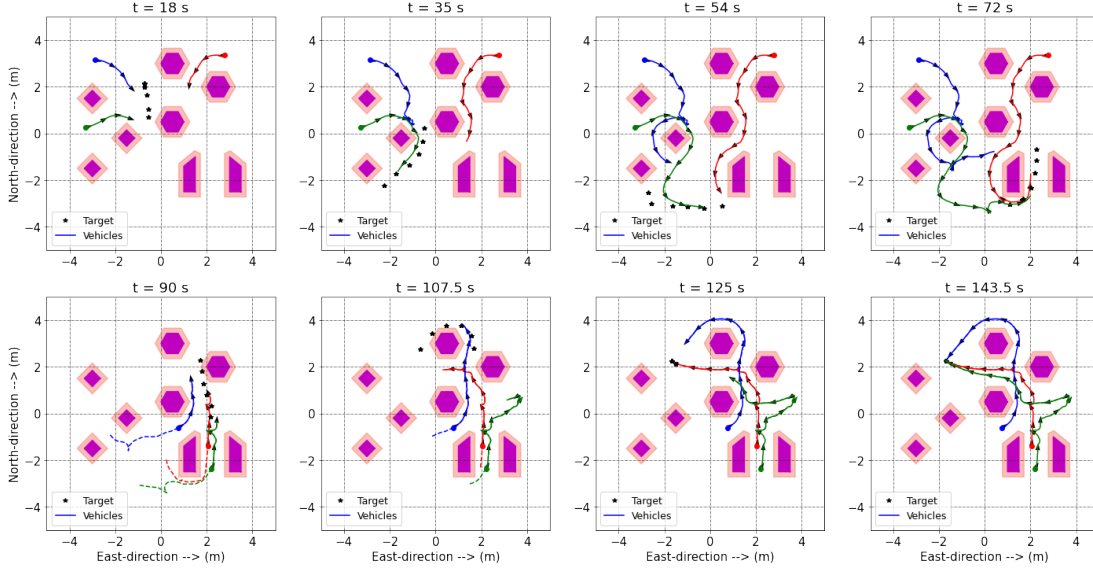


Figure 5.13: Sequential vehicle trajectories for Target Tracking Experiment 4

ods are tested with a single vehicle following a manually controlled dynamic target. The results are evaluated in terms of highest speed achieved by the vehicles without having collision with obstacles. In Figure 5.14, target tracking scenario using the path propagation method for a single vehicle is presented. Here, 8 sequential images from the experiment are presented. The black arrow plotted on the resultant trajectory is the desired velocity vector fed to the automatic flight controller. In this experiment scenario, the vehicle is able to travel four times faster than previous experiments with a maximum speed of 1.6 m/s as opposed to 0.4 m/s thanks to the path propagation method.

A similar dynamic target tracking scenario is considered for the hardware experiment for the position propagation method. In Figure 5.15, presents the resultant trajectories for the position propagation experiment. The resultant trajectories are plotted with 10 second intervals for 60 seconds. The black star is the position of the target logged at 2 second intervals. Furthermore, the resultant velocity profile for target tracking with the position propagation method is presented in Figure 5.16.

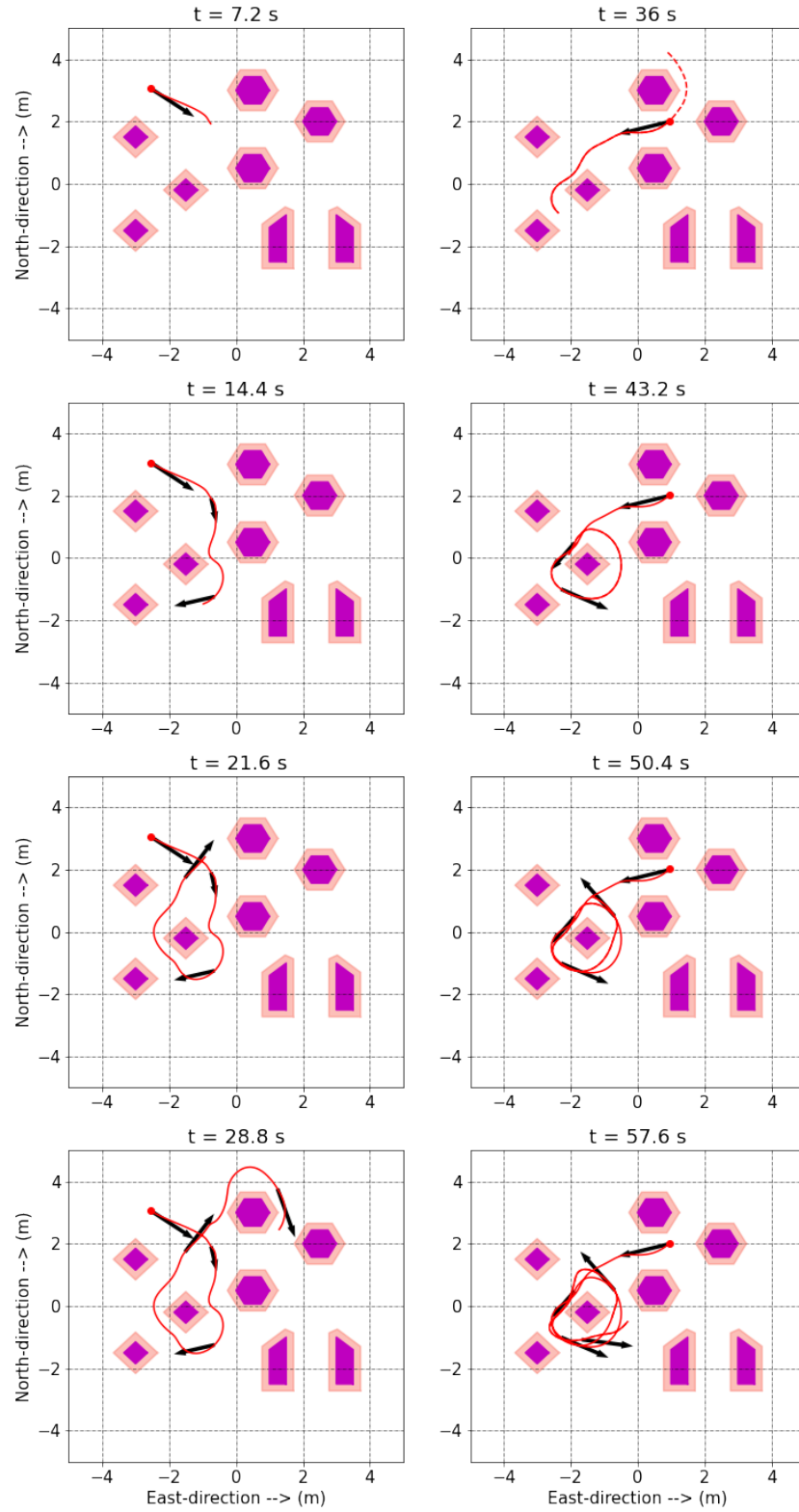


Figure 5.14: Sequential resultant vehicle trajectories for dynamic target tracking with optimal path propagation method.

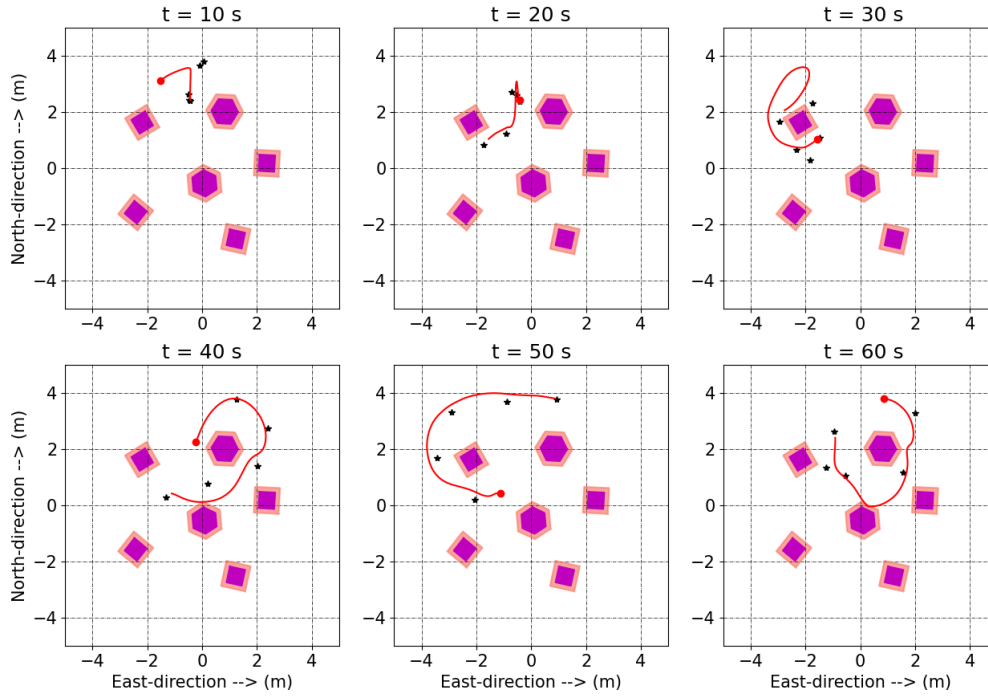


Figure 5.15: Sequential resultant vehicle trajectories for dynamic target tracking with position propagation method.

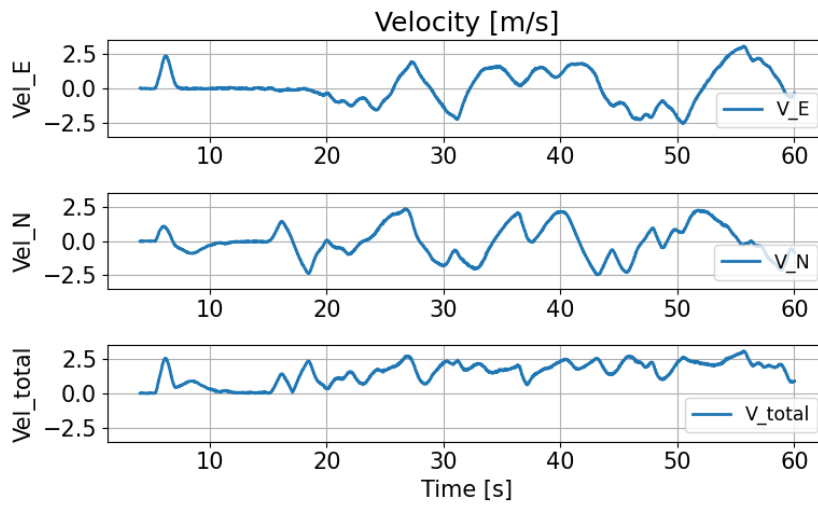


Figure 5.16: Velocity profile for target tracking with position propagation method.

Velocity recordings suggest that the vehicle is able to travel significantly faster than previous experiments with a maximum speed of 2.5 m/s. Even though the curvatures of resultant trajectories obtained with the path and position propagation methods are similar in the simulations, the position propagation method has one notable advantage in real flights. It has a significantly lower computational load. Consequently, the position can be projected into a longer horizon, enabling early avoidance maneuvers and faster flight in real-time.

5.4 3D Guidance Experiments

Further indoor flight tests are conducted to assess the effectiveness of the 3D guidance method proposed in Section 2.2.2. All flight tests in this section use identical obstacle configurations and flight missions.

For the experiments, flight arena setup, and missions described in Figure 5.17 are used.

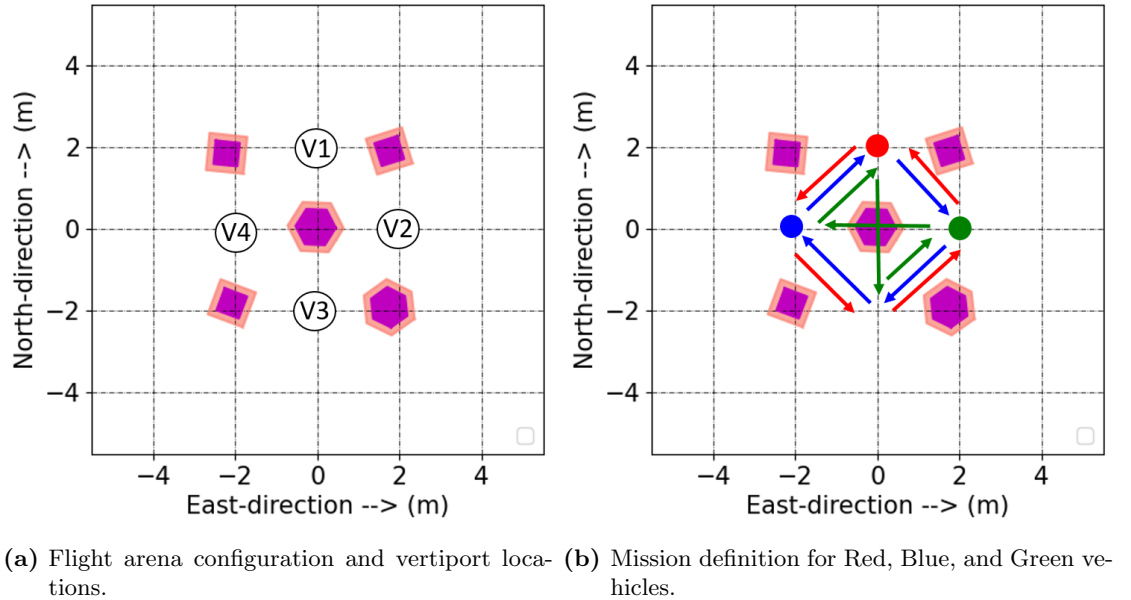


Figure 5.17: Flight arena setup and vehicle tasks.

This configuration represents scaled urban airspace with buildings as obstacles and consecutive goal positions as vertiports within the city. Figure 5.17a displays the top view of the scaled urban area where four vertiports are located between five buildings. Vertiport locations within the city are marked with circles with vertiport numbers. The magenta shapes represent the 2-D cross-sections of buildings within the scaled urban arena. While the PGFlow method solution is capable of generating a guidance vector field around objects of arbitrary shapes, square and hexagonal obstacles are used in the experiments for ease of manufacturing. To ensure safety, each building is expanded by the radius of the vehicle. Expanded safety regions are depicted in lighter colors around

each building.

Throughout the flight test campaign, each vehicle has its own mission. Vehicle missions are visualized in Figure 5.17b. The starting positions of the vehicles are marked with Red, Blue, and Green circles. Vertiport visiting orders for each vehicle are specified with arrows that are the same color as the circles. Once a vehicle visits all vertiports, it returns to its starting position and repeats the task until the test concludes. The blue vehicle commences from 'V4' and proceeds clockwise between consecutive vertiports. Similarly, the red vehicle starts from 'V1' and moves counterclockwise. Lastly, the green vehicle follows the sequence V2-V4-V1-V3-V2, as illustrated in Figure 5.17b. These tasks are designed to demonstrate the collision avoidance capabilities of the modified PGFlow method, as vehicles traveling in opposite directions along the same route are inevitably bound to encounter each other and must execute avoidance maneuvers.

Figure 5.18 presents the experiment results for the 3-D PGFlow method. Similar to the

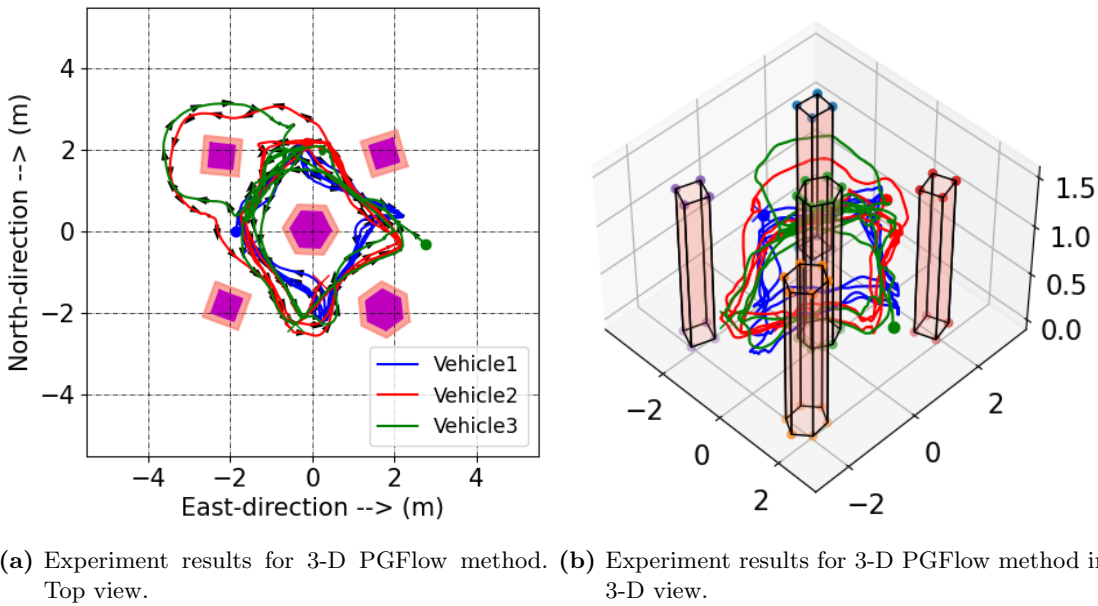
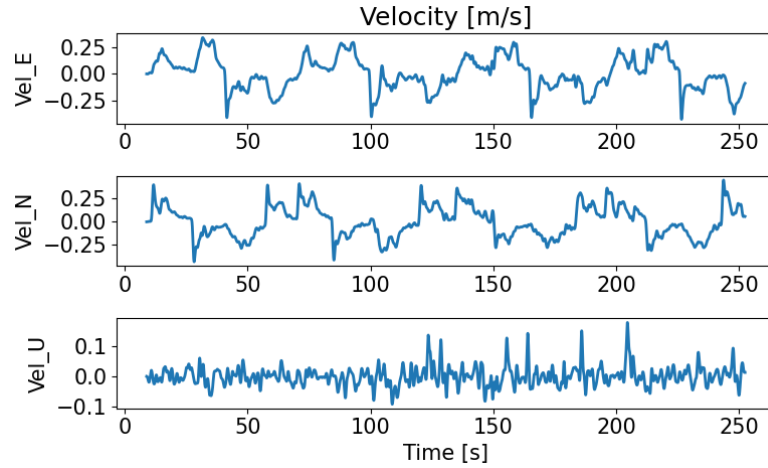


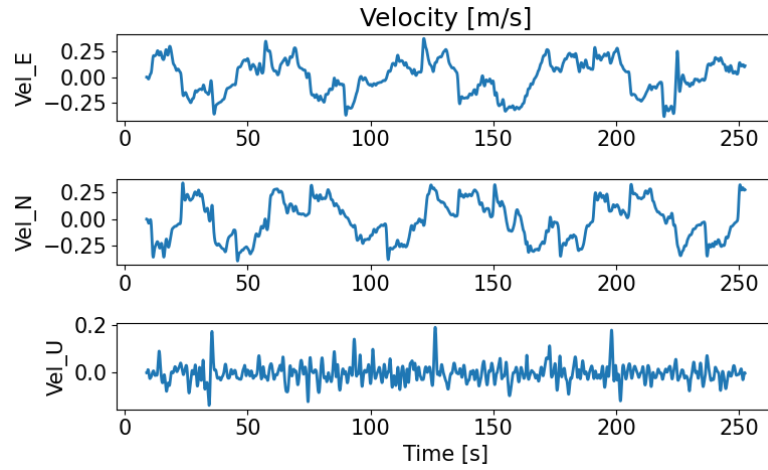
Figure 5.18: Flight test results for nominal case plotted in 2-D and 3-D views.

2-D solution, the 3-D solution also does not provide the trajectories to be followed to the vehicles. Instead, at each time step, PGFlow algorithm calculates a desired velocity vector that guides the vehicles toward their destination while guaranteeing obstacle and collision avoidance. Desired velocity vectors are plotted as black arrows on the resultant trajectories in Figure 5.18a. Notice that when a corridor between buildings becomes congested, vehicles take detours around buildings to evade collisions with one another. 3-D view of the experiment is presented in Figure 5.18b.

Potential flow equations suggest that the flow velocity is faster near sources and sinks, which is not ideal for urban air mobility applications due to safety concerns. Accelerating when approaching a vertiport or other vehicles increases the risk of collision. A dynamic sink strength approach is considered to achieve a more desirable velocity profile, and the



(a) Velocity change for Vehicle 1.



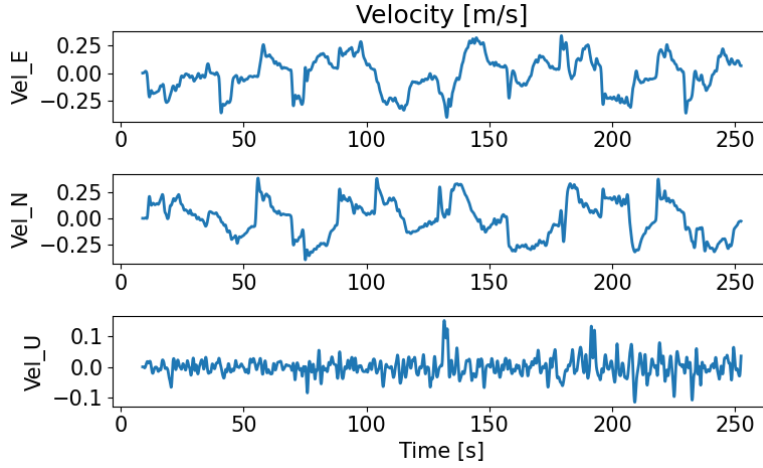
(b) Velocity change for Vehicle 2.

Figure 5.19: Velocity change for vehicles with respect to time. Vehicles slow down when approaching vertiports and when they encounter another vehicle.

sink strength is modeled as a function of distance. As the vehicle moves away from the sink, the sink strength grows, and as it approaches its destination, the sink strength is reduced. This allows the vehicle to travel at higher speeds when it is further away from its goal and slow down as it nears the final destination.

Similarly, the source strengths assigned to other vehicles are adjusted based on the changing sink strength. This ensures that instead of suddenly accelerating to avoid collisions, vehicles slow down, come to a stop, and then change direction if necessary. Velocity profiles for each vehicle in the experiment are illustrated in Figure 5.19.

Experiment results suggest that modifications in the PGFlow algorithm successfully enable collision-free guidance in 3-dimensional space without sacrificing the real-time computation property. Furthermore, the dynamic sink strength function provides a



(c) Velocity change for Vehicle 3.

Figure 5.19: Velocity change for vehicles with respect to time. Vehicles slow down when approaching vertiports and when they encounter another vehicle (Continued).

more desirable velocity profile, especially during the vertiport approach phase. Similarly, dynamic source strength helps vehicles to perform avoidance maneuvers in a more controlled way.

5.5 Conclusions

This chapter focuses on the experimental validation of the PGFlow algorithm on hovering platforms, particularly quad-rotors, within real-world settings. The effectiveness and feasibility of the algorithm in both static goal positioning and dynamic target tracking scenarios are assessed.

Initially, experiments were conducted in the most controlled environment with a single vehicle and static goal positions. The scalability of the proposed algorithm is demonstrated through experiments with multiple vehicles. Moreover, the algorithm's ability to handle real-world uncertainties, such as imprecise initial vehicle positioning, without compromising performance was highlighted. Experiments with static goal positions highlight the PGFlow algorithm's inherent obstacle avoidance and collision avoidance properties.

Transitioning to dynamic target tracking, more challenging scenarios were considered where quad-rotors had to track a moving target while navigating a cluttered environment. Through an extensive experiment campaign, the PGFlow algorithm's capability to dynamically adjust the guidance vector field, ensuring both target tracking and collision avoidance simultaneously, is observed. The low computational load of the PGFlow algorithm is demonstrated through successful real-time adaptation to changing circumstances.

Additionally, dynamic target tracking experiments were repeated with enhancements for

5 Experimental Validation on VTOL Platforms

agile flight. This thesis introduces two methods for achieving faster, collision-free flight: the path propagation and position propagation methods. The proposed path propagation method demonstrated real-time performance, enabling collision-free guidance at speeds four times faster than the baseline approach. Similarly, the position propagation method, with its lower computational cost, achieved even faster real-time flight. Both methods were successfully implemented in hardware experiments, consistently providing collision-free, high-speed flight.

Last but not least, indoor flight tests were conducted using the 3D guidance method. To extend the 2D guidance algorithm to 3D, rather than directly incorporating 3D building models into the calculations, 2D cross-sections of buildings are stacked. Although this approach helps maintain a low computational load and enables real-time guidance capability, several challenges are encountered along the way. Firstly, the quad-rotors' onboard automatic flight controller performed poorly, causing oscillations in the altitude. Secondly, while the approach of stacking 2D cross-sections of buildings offered obstacle avoidance at a relatively low computational cost, it also presented limitations for maneuvers navigating near the top of the buildings. Despite these challenges, the approach successfully enabled real-time 3D navigation and collision avoidance, demonstrating the effectiveness of the method in guiding multiple vehicles in 3 dimensional space.

Results presented in this section underline the collision avoidance property of the PGFlow algorithm, both in static and dynamic scenarios. Furthermore, the performance of agile flight and 3d guidance enhancements are evaluated in terms of computational time and collision free guidance effectiveness. Both enhancements successfully run in real time and provided effective guidance. In conclusion, the experimental validation on hovering platforms reaffirmed the effectiveness and feasibility of the PGFlow algorithm in aerial robotic applications.

6 Robustness under Disturbances

The robustness of autonomous systems, particularly under environmental disturbances, is an important aspect of ensuring safety and reliability. Disturbances to the guidance system can occur in many ways, such as wind gusts, sensor inaccuracies, and controller tracking errors. Wind gusts can introduce unpredictable forces, when strong enough these forces can cause the vehicle to stray from its intended path. Sensor errors, such as inaccurate Global Positioning System (GPS) data or faulty Inertial Navigation System (INS) readings, can lead to incorrect position or velocity estimates, causing the system to misjudge its current state. Controller tracking errors, where the control commands don't perfectly match the vehicle's dynamic response, may result in deviations from the planned trajectory. What all of these disturbances have in common is that they cause the aerial vehicle to end up at a position that was not intended. It is normal to have disturbances in real life when working with real systems. What is important is to counter these unexpected disturbances and continue providing collision-free guidance that leads the vehicle toward the desired destination.

This chapter assesses the robustness of the proposed PGFlow algorithm under wind disturbances large enough that cannot be countered by the automatic flight controller. Wind disturbances, especially in urban airspace cluttered with tall architectural structures, pose a significant challenge for aerial vehicles.

The aim of the robustness study is twofold: first, to investigate the resilience of the PGFlow algorithm under wind disturbances, and second, to enhance its robustness by introducing a correction term in velocity calculations. In order to comprehensively assess the algorithm's robustness, hardware experiments were conducted both indoors and outdoors.

Indoor flights are conducted in the Toulouse Occitanie Drone Flight Arena in France. Wind disturbances are generated using a modular wind generator, allowing for controlled experimentation. Outdoor experiments take place in a local Radio Controlled (RC) airfield runway, a more dynamic and realistic environment for testing.

In the subsequent sections, indoor and outdoor experiments are detailed. First, the experiment setup and flying platform used in the experiments are reviewed. Then, the results obtained from the experiments are presented. Finally, a summary of our findings and insights gained from the experiments are given in the conclusions section.

6.1 Indoor Experiments under Wind Disturbance

This section details the experiments conducted in the indoor flight arena. The indoor arena presents a controlled environment with minimal sensor errors to establish basics. Equipped with a modular wind generator, tests under wind disturbance can also be

conducted. After reviewing the experimental setup and the aerial platform used in the tests, experimental results are presented.

6.1.1 Experimental Setup

The Toulouse Occitanie Drone Flight Arena at ENAC in France is used for indoor flight tests. The indoor flight arena offers a $8\text{ m} \times 8\text{ m} \times 8\text{ m}$ flight volume for experiments. The indoor flight arena is shown in Figure 6.1. The indoor test facility is explained more in detail in Section 4.1. A 16-camera Optitrack system tracks the quad-rotor’s motion



Figure 6.1: The Toulouse Occitanie Indoor Drone Flight Arena.

in real-time with sub-millimeter precision—an aspect substituted by significantly less accurate GPS when conducting outdoor flights.

The flying platform used in these experiments is shown in Figure 6.2. With the battery included, the Explorer quad-rotor has a mass of 535 g and can generate a maximum thrust of 40 N. The 3-cell battery provides approximately 15 minutes of flight time. The indoor quad-rotor, Explorer 1, utilizes the *Paparazzi Autopilot System* [86]. Paparazzi Autopilot System includes a ground segment operating on a personal computer, an airborne segment running onboard the quad-rotor, and an XBee communication link connecting them. The test vehicles are equipped with the latest autopilot hardware from the *Paparazzi* project, known as *Tawaki v1.0*. The *Paparazzi Autopilot System* explained in detail in Section 4.1.

To evaluate the PGFlow algorithm’s robustness against unknown wind gusts, the modular wind generator shown in Figure 6.3 is employed. *WindShape* serves as a modular wind generator situated within ENAC’s indoor flight facility. Spatial wind control is facilitated through *wind pixels*, which are individual counter-rotating small electric fans. These fans have a significantly lower moment of inertia compared to traditional wind

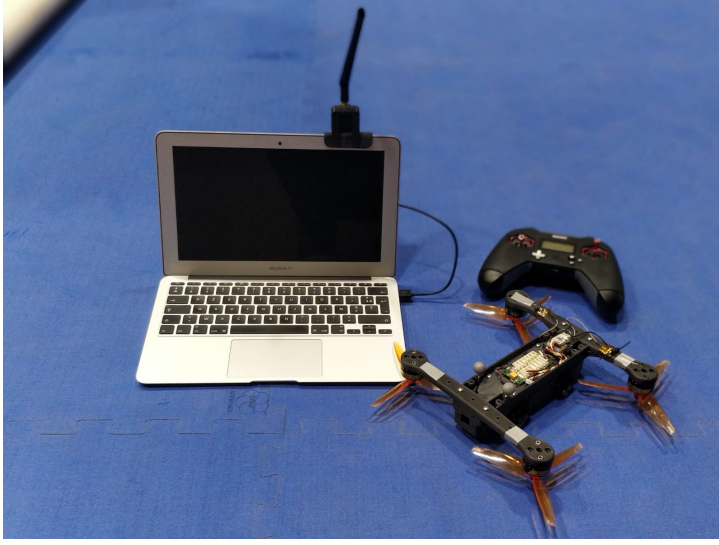


Figure 6.2: Ground control station and quad-rotor Explorer 1.



Figure 6.3: WindShape, wind generator located inside ENAC's indoor flight facility.

tunnel fans. As a consequence, they are more responsive to changes in rotation rate, which allows for rapid variation in wind speed. However, throughout the experiments presented in this thesis, the spatial wind speed was kept constant.

6.1.2 The Correction Term for Disturbance Rejection

Under the effects of wind disturbance, the aerial vehicle inevitably deviates from its intended path. PGFlow algorithm can generate a guidance vector at any position on

the arena map that avoids obstacles and leads to the destination. Therefore, even if the vehicle deviates from its original path at the new position, a new guidance vector that leads to the goal position can be calculated. This property is beneficial in rejecting small disturbances in the trajectory; however, a stronger disturbance rejection may be necessary for strong wind gusts. In order to compensate for the deviation caused by wind gusts, a correction term is introduced.

Let $\mathbf{P} = (x, y)$ be the position of the aerial vehicle in the flight arena. The velocity of the vehicle at this position is denoted by $\mathbf{V} = (u, v)$. Solving PGFlow equations, a desired velocity vector at this position is also calculated as $\mathbf{V}_{des} = (u_{des}, v_{des})$. The INDI controller described in Section 4.2 considers the velocity error $\mathbf{V}_e = \mathbf{V}_{des} - \mathbf{V}$ as input. In addition to the velocity error component, the proposed correction term also considers the position error defined in Equation 6.1.

$$\mathbf{P}_e = \mathbf{P}_{des} - \mathbf{P} \quad (6.1)$$

Here, $\mathbf{P}_{des}$ is the desired position. The desired position is calculated by propagating the current position, $\mathbf{P}$ with desired velocity, $\mathbf{V}_{des}$ for a single time step, Δt as given in Equation 6.2.

$$\mathbf{P}_{des} = \mathbf{P} + \mathbf{V}_{des}\Delta t \quad (6.2)$$

Consequently, when the correction term is used together with the existing velocity tracker, it acts like a Proportional-Derivative (PD) controller.

The effectiveness of the correction term is tested with a single vehicle in an empty arena under wind disturbances. Results of this experiment are presented in Figure 8.1. In this figure, the blue box represents the *WindShape*, and the light blue region is the wake of the *WindShape*. For this experiment, the vehicle needs to travel in a straight line from the arena's East side to the West side. The experiment is repeated with four different wind speeds: 0 m/s, 4 m/s, 7 m/s, and 10 m/s. The wind speed is measured at the exit of the *WindShape* with a hot wire. The solid lines in Figure 8.1 are the resultant trajectories obtained when the PGFlow algorithm is used without the correction term. The dashed lines are the resultant trajectories when the correction term is used. The plot clearly demonstrates the enhancement brought by the correction term. Particularly at higher wind speeds, the correction term provides significant disturbance rejection. The correction term is utilized in all the subsequent experiments.

6.1.3 Experiments with a Single Vehicle

Two basic cases are considered to assess the effectiveness of PGFlow under wind disturbance. For the first case, an obstacle is positioned downstream of the wake of the *WindShape* to inspect whether the wind disturbance could potentially cause a collision with the obstacle. The experiment with a single obstacle is first conducted without wind and then repeated at four different wind speeds: 2 m/s, 3 m/s, 4 m/s, and 7 m/s. The resultant paths are plotted in Figure 8.3. In all the experiments, the vehicle successfully navigates around the obstacle—even when the wind speed reaches up to 7 m/s. However, when the wind speed is as high as 7 m/s, the vehicle overshoots immediately upon exiting

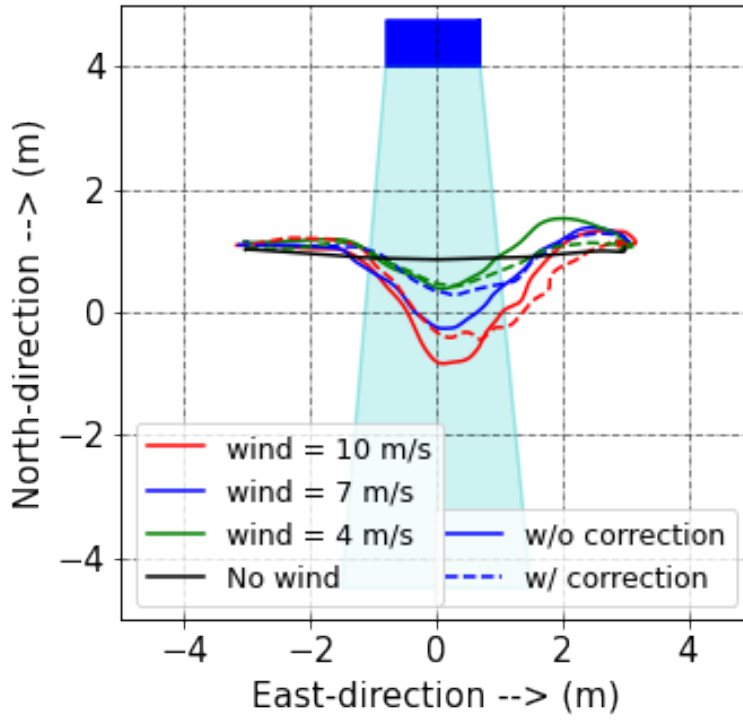


Figure 6.4: The effect of the correction term.

the WindShape wake. This overshoot behavior is to be expected for two main reasons. First, there is an abrupt change in wind speed at the exit of the wind generator wake, and second, the velocity correction solely relies on instantaneous velocity error rather than projecting the desired trajectory several steps ahead.

In the next case, another obstacle is placed right next to the wind generator cone to assess whether this overshoot movement could potentially cause a collision with the obstacle. In the initial tests, due to the effects of the wind, the vehicle avoided the obstacle from the south side. Although this is also a valid solution, the effect of overshooting at the exit of the WindShape's wake could not be assessed. The starting position is shifted 1 m north of the initial starting position to force the vehicle between two strategically placed obstacles. The experiment with the new starting position is repeated at four different wind speeds: 2 m/s, 3 m/s, 4 m/s, and 7 m/s. The resultant paths are plotted in Figure 6.6. In this scenario, too, PGFlow enhanced with the correction term demonstrates satisfactory disturbance rejection while maintaining the collision avoidance property.

After establishing the effectiveness of the proposed algorithm under wind disturbances through basic case studies, more complex scenarios are considered. Two additional experiments are conducted with a single vehicle under wind disturbance in a cluttered arena.

The results of the Experiment 1 are presented in Figure 6.7. In this scenario, the aerial vehicle starts from the north of the flight arena, and its destination is the southeast of the

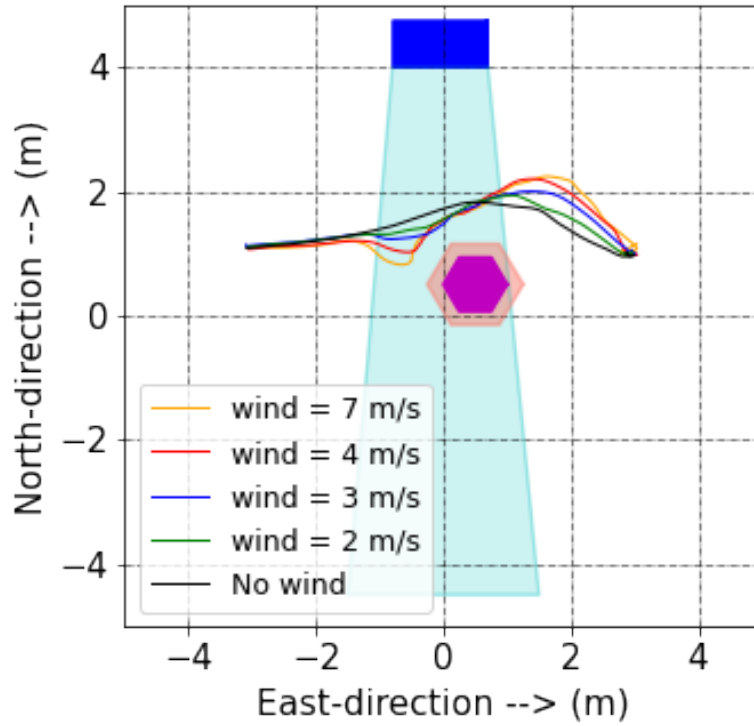


Figure 6.5: Basic Case 1: Arena with a single obstacle.

arena. Consequently, the vehicle has to travel downstream in the wind cone generated by WindShape. The experiment is first conducted without wind and then repeated at three different wind speeds: 2 m/s, 3 m/s, and 4 m/s. The resultant trajectories from experiments conducted at different wind speeds are plotted together in Figure 6.7a. Despite the evident wind disturbance, the vehicle reaches its destination with minimal deviation from the original trajectory in all tests. In Figure 6.7b, sequential image captures from the experiment with wind speed of 4 m/s are presented.

In the second experiment scenario, an aerial vehicle is tasked with traveling from west to northeast. This start and goal position configuration forces the vehicle to travel across the wake of the WindShape. The results of the Experiment 2 are presented in Figure 6.8. The experiment is first conducted without wind and then repeated at three different wind speeds: 2 m/s, 3 m/s, and 4 m/s. The resultant trajectories from experiments conducted at different wind speeds are plotted together in Figure 6.8a. Particularly at high wind speeds, there is a visible deviation from the resultant trajectory from the test without wind. Nonetheless, the vehicle arrives at the desired destination through a collision-free path. In Figure 6.8b, sequential image captures from the experiment with wind speed of 4 m/s are presented.

In addition to the experiments conducted under wind disturbances, an alternative experiment with a different type of disturbance is conducted. This experiment considers a more sudden disturbance than the effect of the WindShape gusts. The results of the

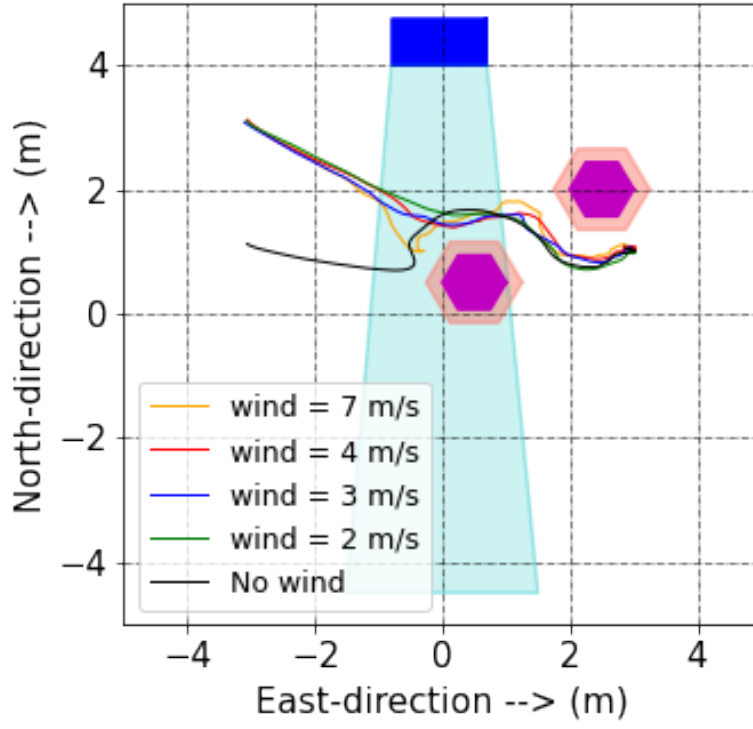
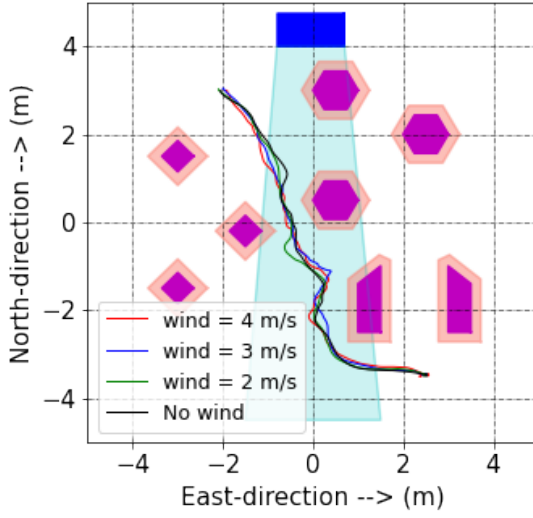
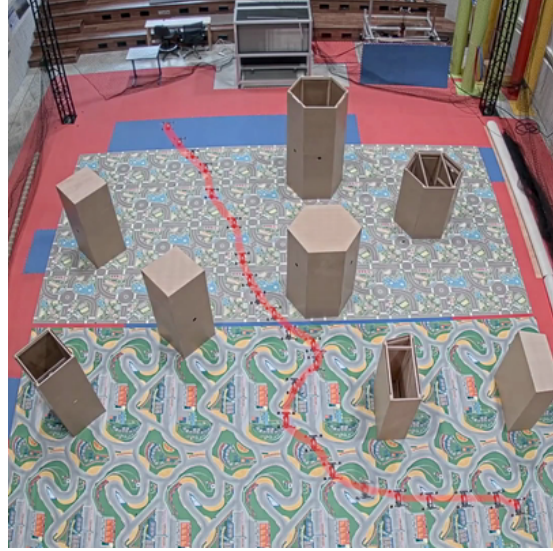


Figure 6.6: Basic Case 2: Arena with two obstacles.

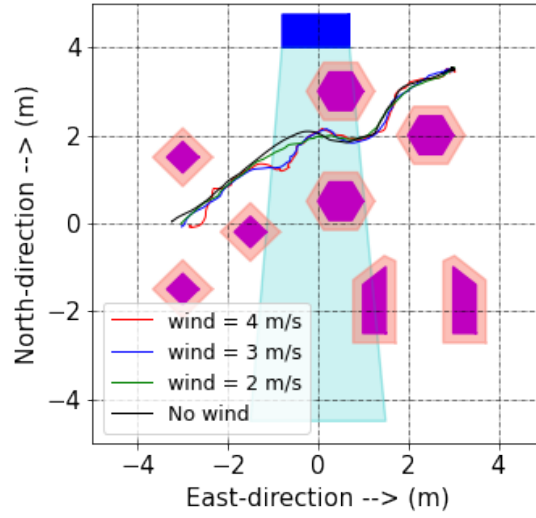


(a) Trajectories for Experiment 1.

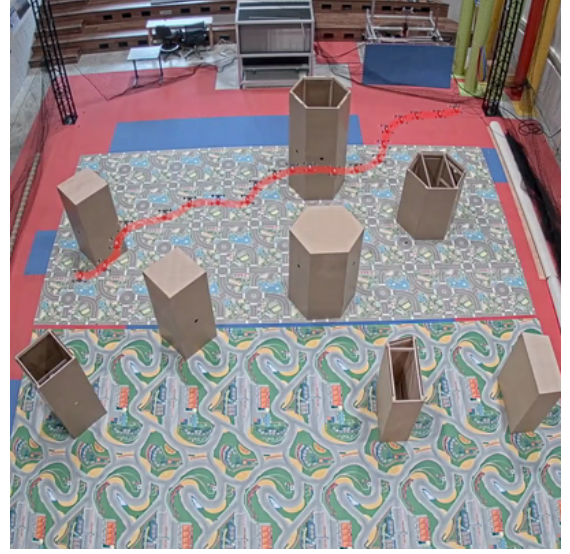


(b) Images captured from experiment with wind speed, $V_w = 4$ m/s.

Figure 6.7: Experiment 1: Travelling downstream in wind cone.

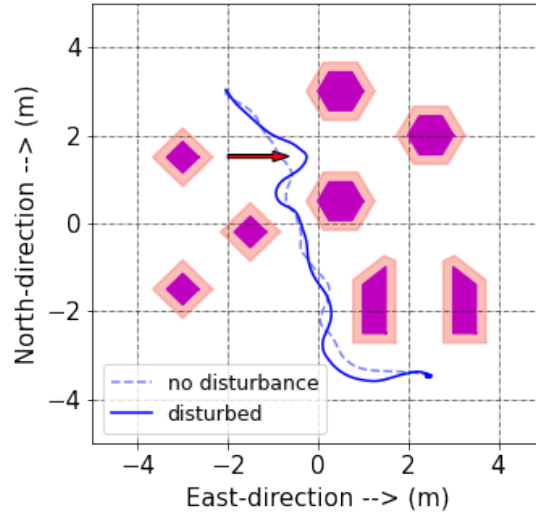


(a) Trajectories for Experiment 2.

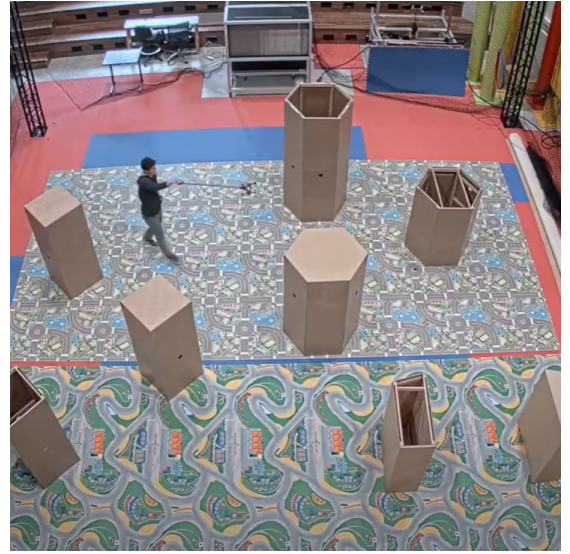


(b) Images captured from experiment with wind speed, $V_w = 4$ m/s.

Figure 6.8: Experiment 2: Travelling across wind cone.



(a) trajectory under sudden disturbance.



(b) Image captured from the experiment with sudden disturbance.

Figure 6.9: The experiment with sudden disturbance.

additional experiment are presented in Figure 6.9. Similar to Experiment 1, in this experiment, the single aerial vehicle needs to travel from northwest to southeast in the indoor flight arena. At the location marked with the red arrow in Figure 6.9a, the quad-

rotor experiences a sudden push towards the east. Thanks to the correction term, the vehicle compensates for this disturbance. The collision avoidance property is maintained even when the vehicle is pushed towards obstacles.

6.1.4 Experiments with Multiple Vehicles

This section tests the PGFlow algorithm's capacity to simultaneously generate collision-free paths for multiple vehicles under wind disturbances. The test setup has only a single quad-rotor available for the tests under wind disturbance. Fortunately, Paparazzi's dynamic simulation tool enables users to include a virtual vehicle in the experiments. Consequently, experiments for multiple vehicles are conducted with an actual vehicle and a virtual vehicle. The experiment results with the real and virtual vehicles are presented

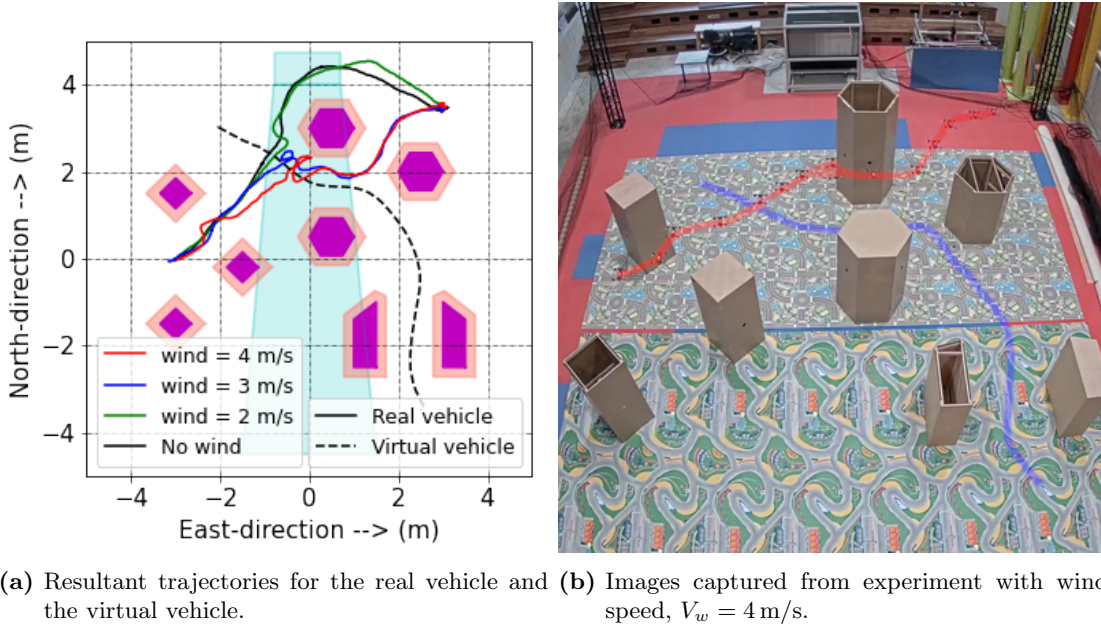


Figure 6.10: Guidance for a real vehicle and a virtual vehicle.

in Figure 6.10. In this experiment, the virtual vehicle is programmed to travel from the northeast to the southwest, as shown in Figure 6.10a with a dashed line. The virtual vehicle created with the Paparazzi's dynamic simulation tool shares the same dynamics with the Explorer quad-rotor described in Section 6.1.1. Furthermore, in this experiment, the virtual vehicle too was guided via PGFlow method in real time simultaneously with the real vehicle. To prevent collisions, vehicles are modeled as point source elements in calculations. Both vehicles communicate their positions to the control computer. The computer calculates the desired velocity vectors for both vehicles. The velocity input for the real vehicle is communicated back to the real vehicle. The velocity command for the virtual vehicle is inputted into the Paparazzi dynamic simulator, where the vehicle position is updated. The only difference between the real vehicle and the virtual vehicle

is that the virtual vehicle is not affected by the wind disturbance present in the real arena.

In the experiment scenario presented in Figure 6.10, the mission of the real vehicle is to travel from west to east. The experiment is conducted under different three wind speeds: 2 m/s, 3 m/s, and 4 m/s. Figure 6.10a presents resultant trajectories from experiments at different wind speeds. Notice that for the experiments with wind speed 2 m/s and 3 m/s, the real vehicle travels upstream in the wind cone to avoid collision with the virtual vehicle. When the wind speed is increased to 4 m/s, the real vehicle can no longer travel upstream; nevertheless, it travels through the wind cone while avoiding obstacles. In all three wind speeds, there is no collision with the virtual vehicle or the obstacles. In Figure 6.10b, sequential image captures from the experiment with wind speed of 4 m/s are presented.

6.2 Outdoor Experiments with Wind Disturbance and Sensor Errors

This section details the experiments conducted in the outdoor flight test facility. The indoor flight tests present a controlled environment with minimal sensor errors to establish basics. However, to better evaluate the method's robustness, experiments are conducted outdoors in a more chaotic environment. The outdoor experiments involve natural wind and more significant localization errors and present an opportunity to thoroughly assess the robustness of the proposed PGFlow method under disturbances.

6.2.1 Outdoor Test Facility

PGFlow algorithm can be scaled easily for larger problems. Hence, extending the method to outdoor experiments is possible where a much larger area is used compared to indoor tests. Outdoor tests are conducted to better evaluate performance under sensor uncertainties and wind disturbances. Outdoor runway experiments take place at a local RC airfield runway in Muret, France, measuring $124\text{ m} \times 12.5\text{ m}$ in size, as shown in Figure 6.11. ENAC enjoys privileged access to this airfield, enabling tests within a volume of a 500 m radius and up to 120 m height (which may be extended up to 450 m under certain circumstances). In the outdoor setup, the absence of an Optitrack system requires sensing the vehicle position and attitude via GPS. The vehicle communicates its position to the control computer through a wireless connection. Subsequently, the control computer computes the desired velocities and transmits this information to the vehicle. The communication scheme for the outdoor test environment is sketched in Figure 4.4.

6.2.2 Quad-Rotor Platform

Quad-rotors used for the outdoor flights are similar to the Explorer setup introduced in Section 4.1. Outdoor quad-rotors, presented in Figure 6.12, are accompanied by a ground control station (an ordinary laptop), safety-pilot transmitters, and an XBee radio modem

6.2 Outdoor Experiments with Wind Disturbance and Sensor Errors



Figure 6.11: Outdoor flight field located in Muret, France.

used for telemetry and downlink communication. The quad-rotors used in outdoor flights

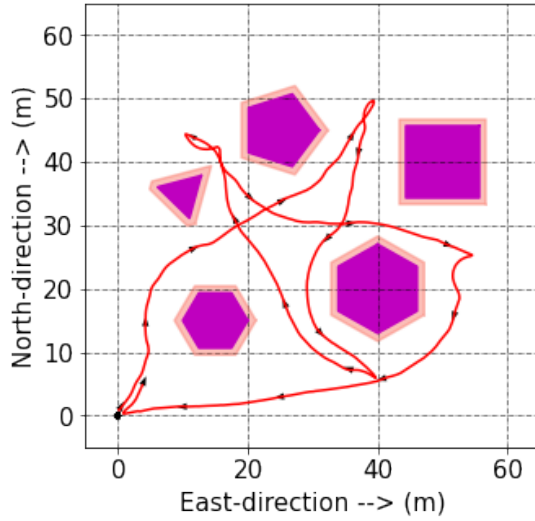


Figure 6.12: Ground control station and the outdoor quad-rotor.

feature the same avionics and propulsion system used in the indoor Explorer platform. Since there is no Optitrack system on the outdoor runway, the drones are additionally equipped with a Ublox-M8 GPS receiver, adding 40 g of additional mass. This GPS receiver provides position information at 5 Hz with an accuracy of approximately 1.5 m. Detailed specifications are presented in Table 4.1 together with the indoor version of the platform.

6.2.3 Experiments with a Single Vehicle

Outdoor experiments are initially conducted with a single vehicle. There are no real obstacles in the outdoor scenarios in the test area. To demonstrate the obstacle avoidance capability of the proposed method under wind disturbances and sensor error in outdoor conditions, no-fly zones with various shapes are defined within the outdoor test facility. The outdoor flight arena with the virtual no-fly zones is presented in Figure 6.13a. A



(a) The resultant trajectory.



(b) Satellite image captured from the experiment with no-fly zones.

Figure 6.13: Outdoor experiment with a single vehicle.

satellite image captured from the outdoor test facility is presented in Figure 6.13a. No-fly zones are plotted as magenta-colored polygons on the satellite image in Figure 6.13b. The shaded areas around the no-fly zones are the buffer zones for increased safety. The safety perimeter in the outdoor experiments is increased to 1 m (as opposed to 20 cm safety perimeter in the indoor arena) to compensate for the greater sensor errors such as GPS accuracy or heading information from the magnetometer.

The resultant trajectories of the outdoor experiment with a single vehicle are presented in Figure 6.13. In this experiment, the vehicle's mission is to start from the origin on the map and travel to four goal positions sequentially before returning to the origin again. The black arrows plotted on the resultant trajectory in Figure 6.13a are the desired velocity vectors calculated by the PGFlow algorithm at given positions. The arrows also indicate the travel direction of the quad-rotor during the experiment. The recorded trajectory is also plotted on the satellite image given in Figure 6.13b.

The outdoor experiment with a single vehicle was conducted on a relatively windy day with a wind speed of approximately 4 m/s. The resultant trajectory suggests that the proposed guidance algorithm successfully performed its obstacle avoidance duty even under wind disturbances and more significant sensor errors.

6.2.4 Experiments with Multiple Vehicles

Similar to indoor experiments, only a single quad-rotor was available in outdoor experiments. Therefore, to conduct experiments with multiple agents a virtual vehicle had to be used in the experiments. The virtual vehicle has the same dynamics and outdoor sensor errors as the real vehicle. The only difference between the real and the virtual vehicle is that the virtual vehicle cannot experience wind gusts in the outdoor airspace. Results from two experiments with the real and the virtual vehicle are presented in Figure 6.14. For both experiments, vehicles travel between three consequent

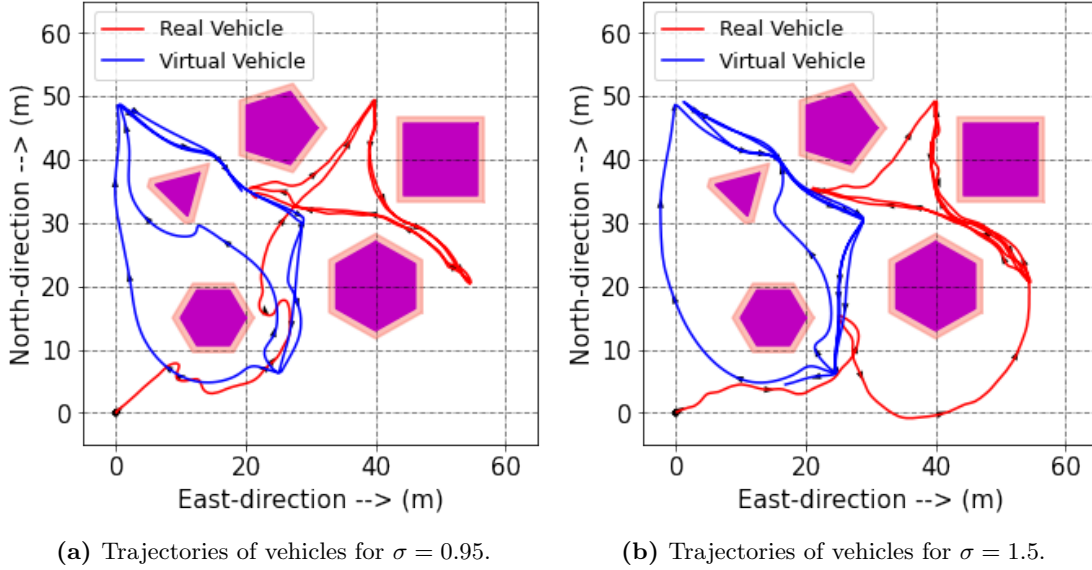


Figure 6.14: Outdoor experiment with a real and a virtual vehicle.

goal positions while avoiding no-fly zones and collision with the other vehicle. In Figure 6.14, the trajectory plotted in blue is the resultant trajectory of the virtual vehicle, and the red trajectory belongs to the real vehicle. The black arrows plotted on top of both trajectories are the calculated desired velocity vectors at given positions. Collisions between vehicles are prevented by modeling each vehicle as a point source element in the PGFlow calculations. Source strength is a design parameter that can be adjusted to control airspace. In Figure 6.14a, the resultant trajectories of an experiment with source strengths $\sigma_{real} = \sigma_{virtual} = 0.95$ are given. This can be considered a relatively low source strength as the sink strengths for both vehicles in this experiment are set to $\sigma_{real} = \sigma_{virtual} = -5.0$. In this scenario, both vehicles enter narrow passages between no-fly zones. The avoidance maneuver does not start until vehicles get close enough to each other. The results of the second experiment with a real and a virtual vehicle are plotted in Figure 6.14b. In this experiment, vehicles have the same missions as in the previous case. However, this time, the source strengths of the vehicle are increased to $\sigma_{real} = \sigma_{virtual} = 1.5$. This increase in the source strength resulted in different airspace usage in the second experiment. In this instance, rather than navigating through nar-

row passages between no-fly zones, the vehicles take detours around virtual obstacles and maintain a longer distance from each other.

6.3 Conclusions

This section investigates the robustness of the PGFlow method under environmental disturbances through hardware experiments conducted in indoor and outdoor test facilities. PGFlow is a vector field-based guidance method. That is, vehicles are guided with a guidance vector calculated at the vehicle position. This guidance vector leads the vehicle toward the goal position while avoiding collisions. Even in cases where the vehicle deviates from its original path due to disturbances, a new guidance vector that guides the vehicle towards the global minimum can be calculated at the new position. Therefore, PGFlow method is inherently robust against disturbances.

In this thesis, an additional correction term is introduced in order to reduce the deviation from the original path under strong disturbances. The correction term feeds the position error to the controller together with the velocity error like a PD controller.

Indoor experiments indicate that the PGFlow algorithm exhibits robustness against wind gust disturbances. Despite the disturbance caused by wind, vehicles can successfully navigate around obstacles and reach their destinations. The effect of the proposed correction term is also demonstrated under wind disturbances. The correction term is shown to enhance disturbance rejection and reduce deviation from the original path.

Outdoor experiments present a greater challenge compared to indoor tests, as they involve not only wind disturbances but also greater sensor errors. Consequently, larger safety perimeters are defined around no-fly zones. These experiments were conducted in an open area with virtual obstacles. Thus, the effect of wind around real obstacles could not be observed in the outdoor experiments.

In summary, both indoor and outdoor flight tests indicate that the PGFlow algorithm offers effective disturbance rejection and is a promising tool for guidance in cluttered airspace.

7 Guidance of Fixed-Wing Platform

This chapter inspects the performance of the proposed PGFlow method for platforms with minimum speed and turn radius constraints. The most common example of such a platform is a fixed-wing aircraft. Unlike hovering platforms, which do not suffer from minimum speed limitations, fixed-wing aircraft would stall at low speeds. Moreover, fixed-wing platforms can also not move backward or sideways like quad-rotors. Consequently, creating trajectories that can be followed by a fixed-wing aircraft is a significant challenge.

This chapter demonstrates the guidance capability of the PGFlow algorithm for a fixed-wing platform through high-fidelity simulations and outdoor flight tests.

7.1 Experimental Setup

7.1.1 Fixed-Wing Platform

An overview of the flight test system used for the fixed-wing platform is presented in Figure 7.1. A Unix/Linux-based laptop serves as the Ground Control Station, responsi-



Figure 7.1: Complete flight test system for fixed wing platform.

ble for controlling the autonomous mission and transmitting the reference velocity input calculated by the PGFlow algorithm during flight. An X-Bee radio modem, affixed to the side of the computer screen as shown in Figure 7.1, is employed for telemetry between the aircraft and the Ground Control Station. Data link communication proved adequate when the flight tests were conducted within a 500 m radius. An RC transmitter is utilized to ensure the safe recovery of the vehicle in the event of loss of control in autonomous mode or any unforeseen circumstances. Detailed specifications of the fixed-wing UAV and the flight test system are presented in Table 7.1.

Specification		Units
Wing Span	1.2	[m]
Surface Area	0.28	[m ²]
Mass	0.75	[kg]
Battery Capacity	30	[Wh]
Flight speed	12	[m/s]
Flight time	60	[min]
Components		
Motor	T-Motor 2208/18 - Kv 1100	
Autopilot	Paparazzi Chimera v1.0 ¹	
GPS	U-Blox M8N	
Companion board	Raspberry Pi Zero v 1.3	

Table 7.1: Airframe Specifications

]

7.1.2 Automatic Flight Controller Integration

In addition to the minimum speed limit, the fixed-wing platform cannot move backward or sideways like the quad-rotor platforms. These constraints call for an alternative automatic flight controller integration. A block diagram for automatic flight controller adjustment for fixed wing platform is presented in Figure 7.2. As always, the output of the PGFlow algorithm is the desired velocity vector. The automatic flight controller computes the velocity error as the difference between the measured vehicle velocity and the desired velocity. In their work, Marina et al. [89] propose a method for converting velocity error into heading error for guiding a fixed-wing aircraft in a vector field. Therefore, the velocity error is converted to a heading error using the method outlined in [89]. Subsequently, the heading error is transformed into a roll angle error, constrained by a sustained turn limitation of the aerial vehicle. Finally, the desired roll angle is fed to the vehicle as input. The reader may refer to [89] for a detailed explanation of the automatic flight controller design.

7.2 Experiment Results

Experiments are conducted at the outdoor test facility detailed in Section 6.2.1. Since obstacles cannot be placed in the outdoor flight arena, no-fly zones with various shapes are defined on the map. This chapter inspects two experiment scenarios with different no-fly zone configurations. The first configuration is plotted in Figure 7.3. The first scenario is simulated through the Paparazzi Autopilot system. The dynamic simulation includes vehicle dynamics and sensor errors. The mission of the fixed-wing UAV is to travel between the points A-B-C-D-E marked with stars in Figure 7.3. Before the mis-

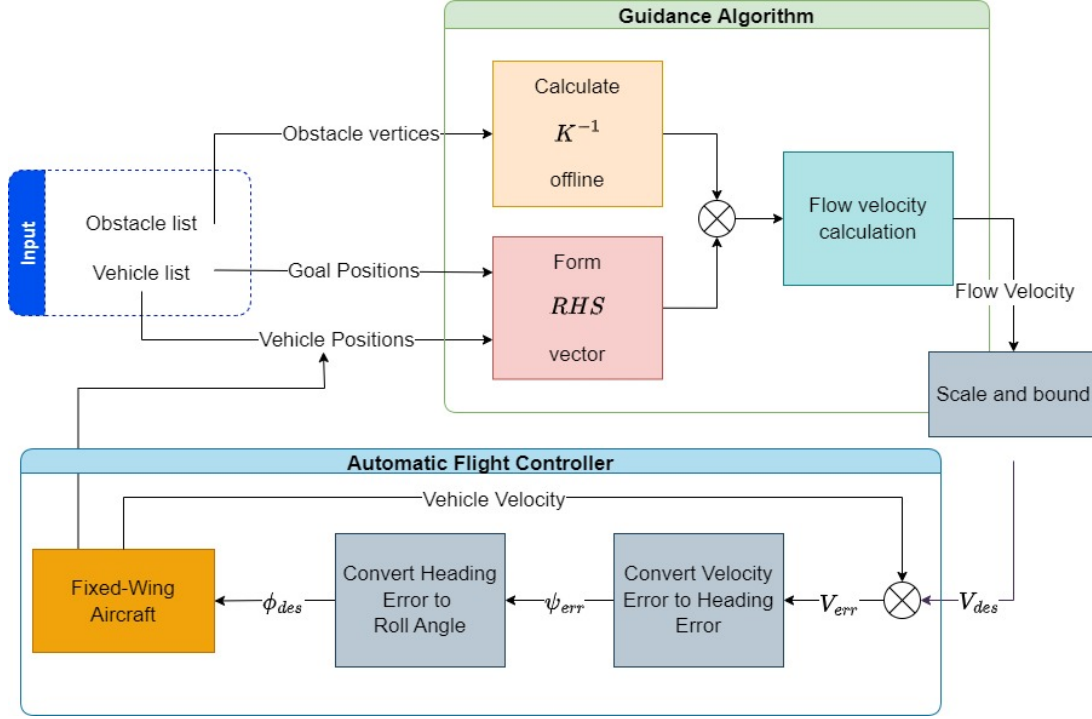


Figure 7.2: Block diagram for automatic flight controller adjustment for fixed wing platform.

sion, the fixed-wing UAV loiters around a standby position just outside the flight mission arena, as indicated by the start arrow in Figure 7.3. In Figure 7.4, the simulation results are plotted on the satellite image of the outdoor flight facility at Muret, France. A red diamond denotes the standby position in the lower right corner of Figure 7.4, and the loiter path is outlined with a green circle around the standby point. Additionally, in Figure 7.4, no-fly zones are visible as red polygons. Notice that a trapezoidal no-fly zone is defined on the point marked with Home in Figure 7.4. In fact, this zone is where the test crew stands during the experiments. The no-fly zone is defined for this area to ensure the safety of the researchers. When the start command is given, the fixed-wing UAV starts traveling between points A-B-C-D-E sequentially without violating the no-fly zones. Upon reaching point E, the vehicle repeats the sequence until it is instructed to return to the standby position again. The arrows placed on the resultant trajectory in Figure 7.3 represent the instantaneous flow velocity calculated by the PGFlow method. The difference between this instantaneous flow velocity and the measured vehicle velocity is used in heading error calculation in the automatic flight controller. Throughout the simulated scenarios, the ground speed of the vehicle was maintained at 13 m/s, while the maximum bank angle was constrained to 30 degrees.

Figure 7.5 illustrates the resultant path logged during an actual flight test for scenario 1 in Muret, France. In this experiment, the fixed-wing UAV is tasked with traveling between points A-B-C-D-E and repeating the task until it is called to the standby position again. While the resultant trajectory followed by the fixed-wing UAV in real flight

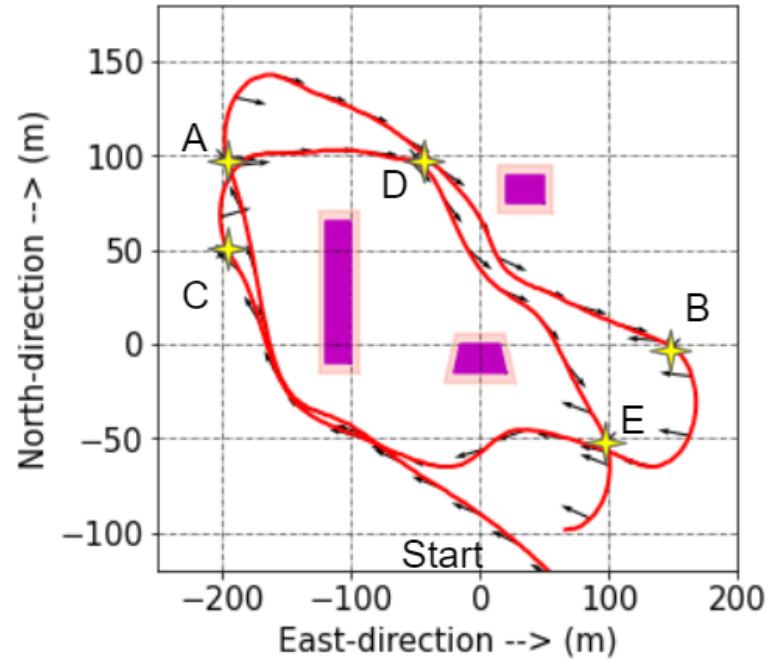


Figure 7.3: Simulation for Scenario 1: Desired velocity vectors and the resultant trajectory.

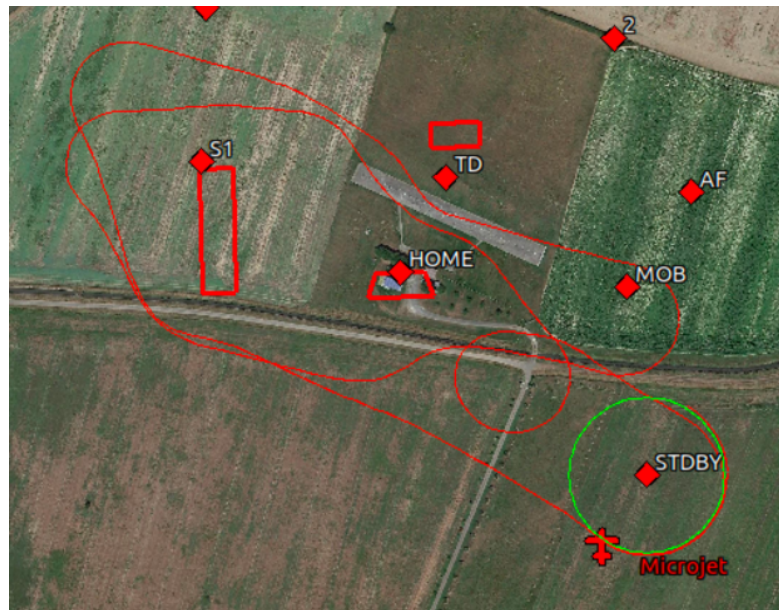


Figure 7.4: Simulation for Scenario 1: Resultant trajectories on the flight facility at Muret, France.

deviates slightly from the simulated case, the vehicle successfully completes the mission

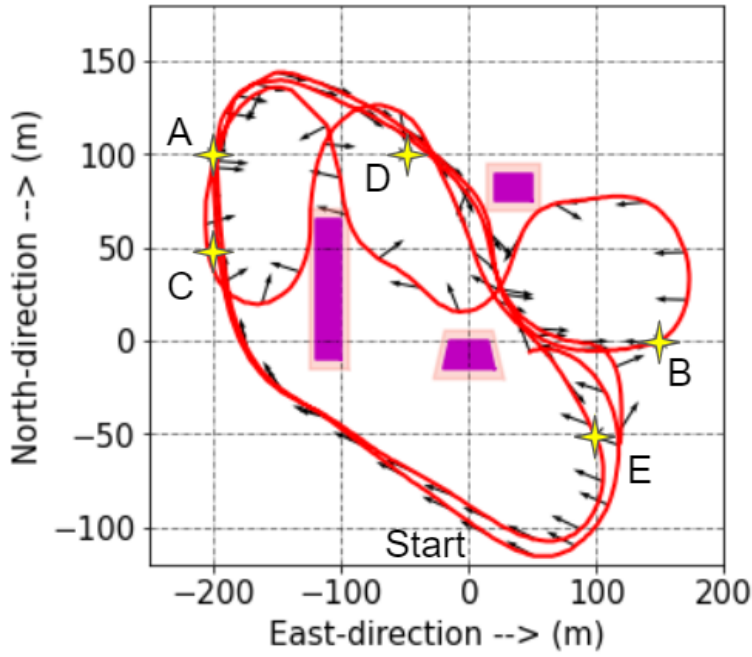


Figure 7.5: Real flight experiment for Scenario 1: Desired velocity vectors and the resultant trajectories.

multiple times without violating the no-fly zones. One of the advantages of the proposed PGFlow method is that it can easily be scaled to larger problems. There is no need for adjustments when working with smaller or larger obstacles so long as the obstacle shape is defined with a sufficient number of elements. In this regard, the second experiment scenario is conducted in a larger airspace, as shown in Figure 7.6. Additionally, unlike the first scenario, the second arena includes arbitrarily shaped no-fly zones rather than just rectangles. For the second experiment scenario, the fixed-wing UAV loiters around the standby position outside the experiment area. The mission of the fixed-wing UAV is sequentially traveling between points A-B-C-D-E marked with stars in Figure 7.6 and then returning to point A to repeat the task.

In experiment scenario 1, a relatively smaller arena was considered. When navigating within such a confined space, the minimum turn radius of the fixed-wing aircraft becomes a limiting factor. Conversely, in experiment scenario 2, a much larger area is available, providing ample space for maneuvers at the goal locations. To increase the difficulty of this scenario, no-fly zones are strategically positioned to create narrow corridors between the goal positions. During real flight tests, the ground speed of the vehicle was maintained at 15 m/s, and the maximum bank angle was restricted to 50 degrees.

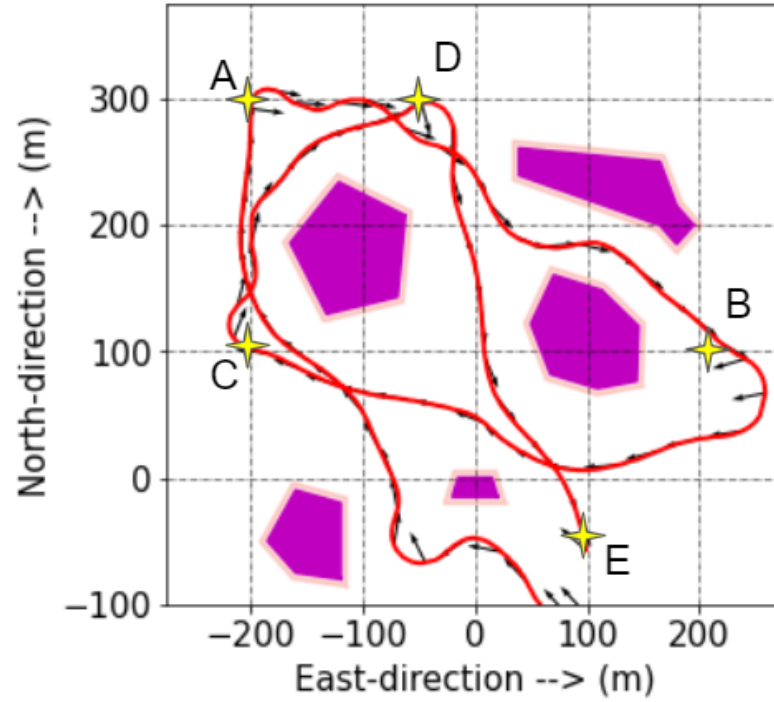


Figure 7.6: Real flight log for Scenario 2.

7.3 Conclusions

Guiding a fixed-wing aircraft presents a challenging task due to the constraints imposed by minimum speed and turn radius requirements. The PGFlow method mimics fluid flow and produces smooth trajectories that are easy for vehicles to follow. By integrating the PGFlow algorithm into the Paparazzi Autopilot system, both simulations and real-flight experiments are conducted, demonstrating the feasibility of the proposed approach on platforms with minimum speed constraints. Experiment results demonstrate the effectiveness of the PGFlow method in navigating fixed-wing aircraft while respecting no-fly zones and maintaining safety margins. The comparison between simulation and real-flight trajectories highlights the method's robustness and adaptability in outdoor environments. Furthermore, the scalability of the PGFlow method is demonstrated through experiments in varied airspace sizes. In conclusion, simulations and flight experiments demonstrate that fixed-wing aircraft can follow the guidance vector generated by the PGFlow method without exceeding the flight envelope limits.

8 Urban Air Mobility Applications

This chapter presents a practical application of PGFlow algorithm for traffic management in urban airspace. To better fit the needs of air traffic rules additional features aimed at enhancing safety and social acceptance of UAM are introduced. Furthermore, concept of flight corridors and tactical deconfliction in urban air mobility are studied. This chapter utilizes the 3-dimensional guidance method introduced in Section 2.2.2 to enhance safety and social acceptance of UAM. Extending the PGFlow algorithm to 3-dimensions enables more comprehensive modeling of the urban airspace. By considering the vertical dimension, vehicles can utilize the entire urban airspace. Consequently, extending the PGFlow algorithm to 3-dimensions results in more dynamic and efficient navigation enabling two additional features: emergency landing and crowd avoidance capabilities. The emergency landing mode provides a contingency plan for unforeseen failures, ensuring the safety of both passengers and bystanders. The crowd avoidance feature dynamically adjusts flight paths to avoid densely populated areas, addressing both safety and visual pollution concerns. The effectiveness of the 3-dimensional PGFlow method and the safety enhancements are assessed through hardware experiments conducted in the indoor flight facility in Toulouse.

Additionally, this chapter inspects the integration of air traffic control rules with the guidance algorithm. Controlling the air traffic by utilizing flight corridors is a popular idea in Urban Air Traffic Management (UTM) [21]. UAM Corridors can be defined as specified airspace zones created to facilitate secure UAM activities. The proposed PGFlow algorithm imitates fluid flow around obstacles to generate a guidance vector field that can be utilized to navigate vehicles through flight corridors. The algorithm is adaptable and can be customized to align with the requirements of existing air traffic regulations. This chapter assesses the effectiveness of the proposed PGFlow method's conflict resolution and corridor following capabilities through a comprehensive flight test campaign. The assessment in this chapter covers scenarios involving the conflict of two medical drones within the same corridor, as well as conflicts between an air taxi and a non-cooperative intruder. Hardware experiments for both scenarios are conducted in a scaled urban environment, incorporating both simple and complex corridor shapes. Various corridor models are proposed, each offering different levels of flexibility in movement within the corridor.

This chapter is organized as follows: In Section 8.1 the crowd avoidance feature is detailed. Importance of social acceptance in any mobility system is underlined and the integration of population model within the guidance algorithm is discussed. Next, Section 8.2 discusses the emergency landing feature. Integration of air traffic control rules with the guidance algorithm is explained in Section 8.3. Two tactical deconfliction scenarios inspired by existing urban air traffic management studies are replicated and solved

with the proposed method in Section 8.4 and Section 8.5.1. Conclusions are delivered in Section 8.6.

8.1 Population Avoidance Application

This section details the population avoidance feature added to the PGFlow method. This feature dynamically adjusts the trajectories to avoid densely populated areas within the city to improve safety and reduce visual pollution.

This section presents the population model in the PGFlow equations after discussing the motivation behind designing a crowd avoidance feature. The effectiveness of the proposed feature is tested through hardware experiments conducted in the indoor flight test facility.

8.1.1 Motivation

Achieving public acceptance is a significant challenge in implementing air mobility for urban transportation, as community resistance is the most probable risk that could undermine the success of any transport system [90]. To mitigate this threat and promote adequate community engagement, it's essential to consider public concerns and feedback when designing a transportation system. Surveys conducted among European Union (EU) citizens reveal that the most prevalent concerns regarding Urban Air Mobility (UAM) operations include safety, noise, visual impact, and security [9, 91, 92, 93, 94, 95]. Addressing these concerns is an essential step in obtaining public acceptance and successful implementation of Urban Air Mobility concepts.

The anticipated usage for urban air mobility services in 2035 suggests a limited number of automated personal aerial transportation missions [90]. Consequently, when addressing safety concerns, it is more appropriate to consider the entire urban population rather than just the users of these services, given that most residents will not utilize aerial transportation but will still be impacted by it [9]. The proposed crowd avoidance feature enhances the safety of aerial transportation for both users and non-users. While simply blocking off areas with crowds may seem practical, it can hinder the efficiency of the air mobility system by restricting routes even when there is no crowd to avoid. Thus, a more dynamic approach is proposed to ensure citizen safety while maintaining the mobility system's efficiency. The proposed automatic crowd avoidance feature integrated into the guidance system allows vehicles to autonomously steer clear of crowded areas while avoiding collisions with each other and buildings.

8.1.2 Modeling Population

To enhance safety without compromising efficiency, implementing a dynamic approach that restricts flying over crowded regions while allowing for flexibility in empty areas can balance the safety of citizens with the efficiency of the air mobility system. For this purpose, the crowds within the city are represented as point source elements with adaptable strengths. These populations can emerge dynamically, increasing their intensity as

the area becomes more congested or disappearing as crowds disperse. Additionally, the source points can be relocated in real time if the population shifts within the city. The desired velocity vector for avoiding both crowds and collisions in the urban environment can be computed using Equations 8.1, 8.2, and 8.3.

$$u = \frac{-\sigma}{2\pi} \frac{x - x_g}{r_g^2} + \sum_{n=1}^N \frac{\sigma_n}{2\pi} \frac{x - x_n}{r_n^2} + \sum_{m=1}^M \frac{\sigma_m}{2\pi} \frac{x - x_m}{r_m^2} + \sum_{s=1}^S \sum_{k=1}^K \frac{\gamma_{sk}}{2\pi} \frac{x - x_{sk}}{r_{sk}^2} \quad (8.1)$$

$$v = \frac{-\sigma}{2\pi} \frac{y - y_g}{r_g^2} + \sum_{n=1}^N \frac{\sigma_n}{2\pi} \frac{y - y_n}{r_n^2} + \sum_{m=1}^M \frac{\sigma_m}{2\pi} \frac{y - y_m}{r_m^2} + \sum_{s=1}^S \sum_{k=1}^K \frac{-\gamma_{sk}}{2\pi} \frac{y - y_{sk}}{r_{sk}^2} \quad (8.2)$$

$$w = \frac{-\sigma_g}{4\pi} \frac{z - z_g}{r_g^3} + \sum_{n=1}^N \frac{\sigma_n}{4\pi} \frac{z - z_n}{r_n^3} + \sum_{m=1}^M \frac{\sigma_m}{4\pi} \frac{z - z_m}{r_m^3} \quad (8.3)$$

In these equations, (x_g, y_g) represents the goal position, while σ denotes the sink strength linked to this goal position. N stands for the number of vehicles present in the arena, with each vehicle having a source with strength σ_n . Similarly, M represents the overall number of restricted zones within the arena, with each zone being assigned a source and strength denoted by σ_m . Additionally, S denotes the number of obstacles, and K signifies the number of panels.

8.1.3 Experiment Results

The effectiveness of the proposed crowd avoidance feature is assessed through hardware experiments. In the crowd avoidance experiment, a population density increase is introduced in the city. Flight over densely populated areas raises concerns about the safety of ground populations in the event of a catastrophic failure during flight. Moreover, even without such failures, flights in overcrowded areas can disrupt peace due to the noise and visual pollution generated by urban air mobility vehicles. Thus, avoiding crowded areas becomes crucial not only for safety but also for promoting social acceptance of such mobility solutions. In this experiment, vehicles receive notifications to steer clear of flying over this section when a particular region in the city becomes crowded. The vehicles are instructed to maintain the same route sequence between vertiports as in the nominal case presented in Section 5.4. Figure 8.1 presents the results of flight tests both in 2-D and 3-D view, where the densely populated area is highlighted with an orange circle. It's evident from Figure 8.1 that the trajectories followed by vehicles differ significantly from those in Figure 5.18, representing the nominal case. In this experiment,

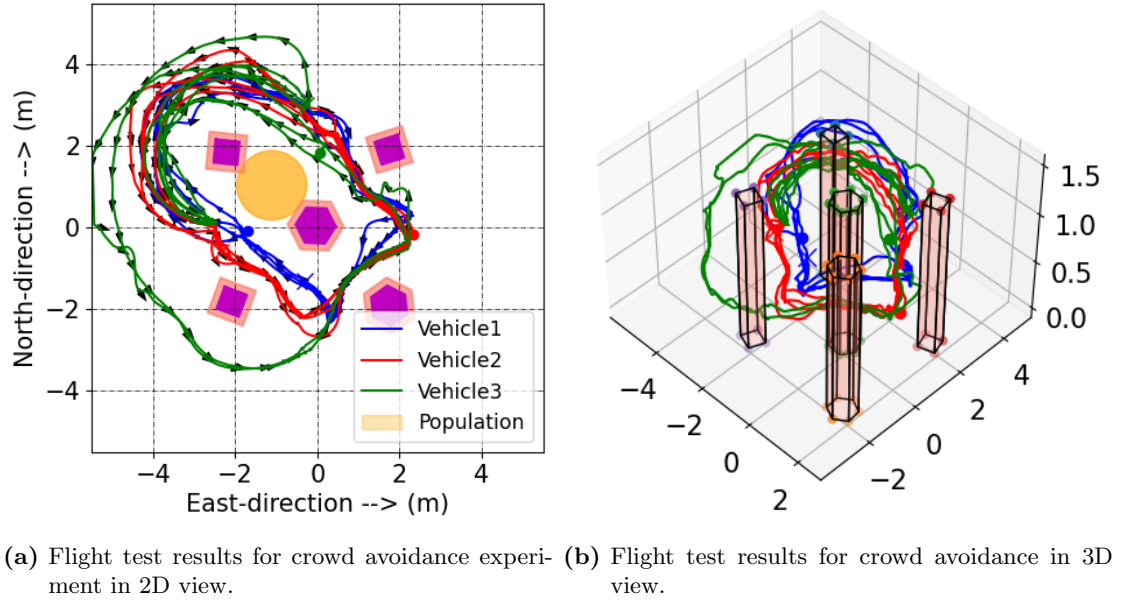


Figure 8.1: Flight test results for the crowd avoidance experiment plotted in 2D and 3D view.

vehicles not only steer clear of crowded streets but also avoid collisions with buildings and each other.

Experiment results suggest that the proposed population model enables vehicles to avoid flying over densely populated regions within the city. Being able to dynamically adjust the size and the location of the population term in the velocity calculations ensures the effective use of the available airspace.

8.2 Emergency Landing Application

An emergency landing feature is introduced to serve as a contingency plan for unforeseen failures and to ensure the safety of both passengers and bystanders. This study defines a failure as a vehicle that is unable to proceed with its mission but can still maneuver and safely land at the nearest emergency landing pad. The emergency landing pads are positioned in close proximity to vertiports and away from the surrounding buildings, as shown in Figure 8.2. There are four emergency landing pads located near vertiports marked with boxes. When a vehicle experiences a failure and needs to perform an emergency maneuver, other vehicles in the airspace can continue their missions uninterrupted and take measures to avoid the failed vehicle, thereby avoiding any potential accidents.

In the hardware experiments, actual failures are not induced on the vehicles. Instead, during the mission, a selected vehicle is informed that it is experiencing a failure and must execute an emergency landing on the nearest emergency landing pad as soon as possible. Vehicles do not have prior knowledge of which vehicle will fail or when the failure will occur. Consequently, when a failure arises, vehicles must autonomously ad-

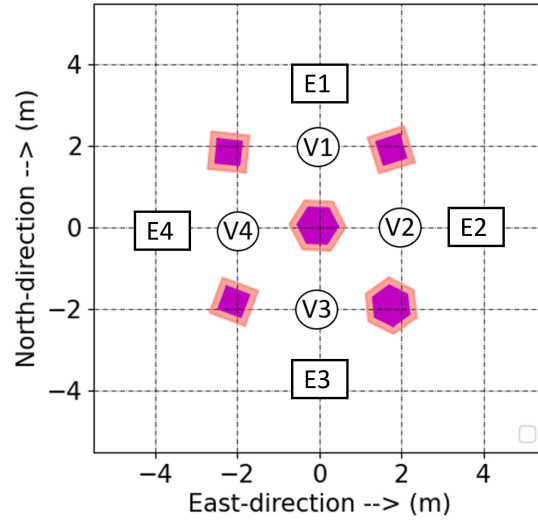
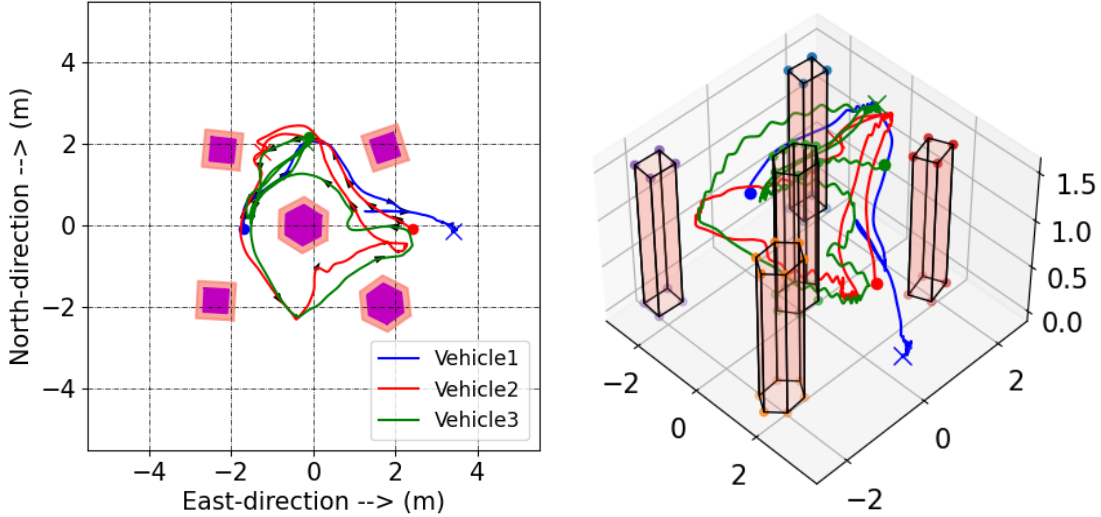


Figure 8.2: Flight arena setup with vertiport locations and emergency landing pads.



(a) Experiment results for the emergency landing case. Top view. (b) Flight test results for the emergency landing experiment in 3-D view.

Figure 8.3: Flight test results for emergency landing experiment plotted in 2-D and 3-D views.

just their path and either execute the emergency landing or evade the failed vehicle and continue with their mission.

The flight test results for the emergency landing experiment are presented in Figure 8.3. In this experiment, 40 seconds after the initiation of the missions, the blue vehicle is notified of a failure and instructed to land immediately. The selected vehicle aborts its mission and identifies the nearest emergency landing pad to execute an emergency

landing maneuver. Meanwhile, the red and green vehicles maneuver to avoid the blue vehicle and continue with their respective missions.

Figure 8.3a presents the resultant trajectories for the emergency landing experiment in the top view of the flight arena. Figure 8.3b presents the flight test results for the emergency landing experiment 3-D view.

Figure 8.4 illustrates the altitude changes of vehicles over time. In this experiment,

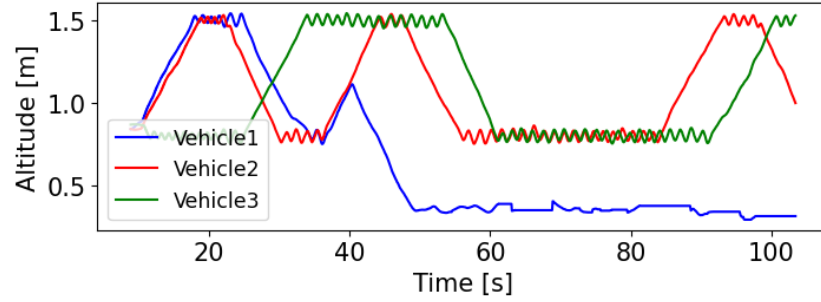


Figure 8.4: Altitude change of vehicles for the emergency landing experiment

each vertiport is situated at a distinct altitude, and the emergency landing pad is positioned at 0.4 m. This altitude is the reference zero detectable by the Opti-Track system in the indoor flight test facility. Each vertiport is modeled by a 3-D sink element at a different altitude, causing vehicles to ascend or descend between destinations. Similarly, the emergency landing pad is also represented by a 3-D sink element, enabling automatic landing maneuvers for vehicles. The oscillations in altitude are due to the altitude hold feature of the onboard automatic flight controller. In Figure 8.4, at $t = 40.0$ seconds, it is apparent that the blue vehicle not only alters its heading toward the emergency landing pad but also initiates its descent.

8.3 Air Traffic Control Rules Integration

8.3.1 Motivation

Transitioning a portion of urban traffic from the ground to the air not only has the potential to decrease traffic congestion and travel times but also offers a significant business opportunity. According to projections, the European urban air mobility (UAM) market is expected to reach around €4.2 billion by 2030 [9]. As a result, the urban air mobility concept attracts increasing attention each day, urging large and small companies to compete against each other to develop the next urban air mobility concept[96]. Nevertheless, the emergence of various aerial vehicles with different designs, sizes, maneuverability, automation levels, and communication capabilities into once unregulated and restricted urban airspace requires designing new air traffic management systems [12].

Existing urban air space management initiatives support and increased structure in densely populated areas [15, 16, 17]. The airspace structure can be reinforced by imple-

menting lanes, tubes, or corridors [21, 22, 23, 24]. This approach offers an advantage by leveraging an existing traffic management system, enabling a swift integration of urban air mobility. Conversely, there are also studies suggesting segregating airspace based on technology level and providing less structured airspace for vehicles with higher levels of autonomy would increase the capacity of the urban air space and reduce the urban air traffic management efforts [30, 31, 32, 28, 13, 29]. However, postponing UAM integration until vehicles attain the required level of autonomy could delay the utilization of urban airspace. One potential solution to this issue is to initially align UAM corridors with existing regulatory frameworks like Visual Flight Rules and Instrument Flight Rules to meet operational requirements. Then, as UAM operations expand, these corridors could progressively operate autonomously, no longer relying on traditional ATC services [21]. From this viewpoint, a guidance algorithm that seamlessly adapts to the initially constrained concept of operations during the integration phase while demonstrating the ability to autonomously generate collision-free trajectories as operations shift towards less regulated traffic management schemes would be incredibly valuable and forward-thinking.

8.3.2 Corridor Models

The corridor concept, extensively utilized in air traffic control, can also be integrated into the urban air mobility framework [21, 22, 23, 24]. The proposed PGFlow algorithm can model corridors in various ways, depending on the desired level of strictness for enforcing corridor boundaries. This section presents three different corridor models, ranging from less strict to most strict.

The least strict corridor model is the *check point model*. This model places control points named *check points* along the corridor. A free stream velocity pointing from the current vehicle position toward the next *check point* guides the aerial vehicle within the corridor. Having no hard constraints on the corridor boundaries or control points, this method provides the most freedom to the aerial vehicle.

Figure 8.5 presents a schematic representation of the *check point model*. In this figure, light blue regions represent the corridor boundaries. The dashed lines around the corridor boundaries indicate that there is no hard constraint enforced in the corridor boundaries. The blue triangle is the aerial vehicle traveling inside the corridor. Green dots represent the consecutive *check points* within the corridor.

The second corridor model is the *way point model*. Similar to the *check point model*, control points are placed along the corridor for this model. The control points are named *way points* and are modeled as point sink elements. Point sink elements are activated sequentially to guide the aerial vehicle from one *way point* to the next. Similar to the *check point model*, there is no hard constraint on the corridor bounds. The strictness of this corridor model can be adjusted by changing the distance between consecutive *way points* and assigned sink strengths.

Figure 8.6 presents a schematic representation of the *way point model*. In this figure, light blue regions represent the corridor boundaries. The dashed lines around the corridor boundaries indicate that there is no hard constraint enforced in the corridor

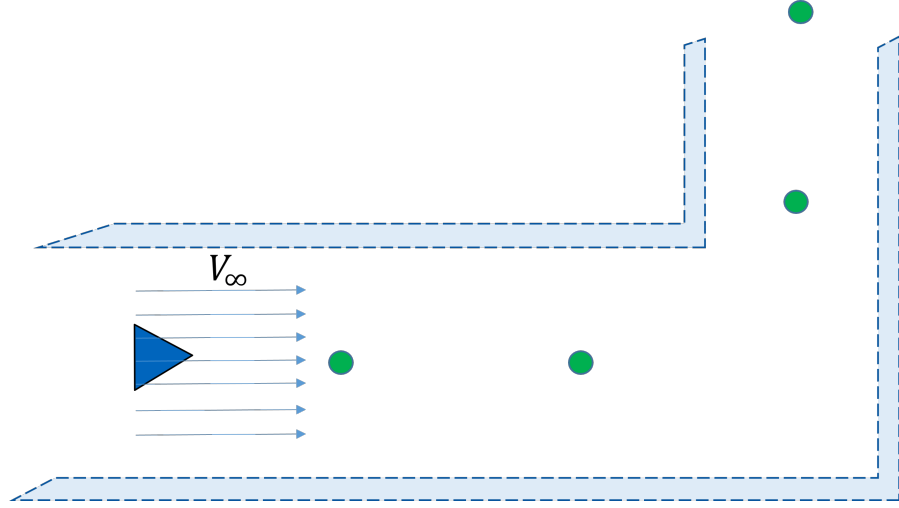


Figure 8.5: Checkpoint corridor model.

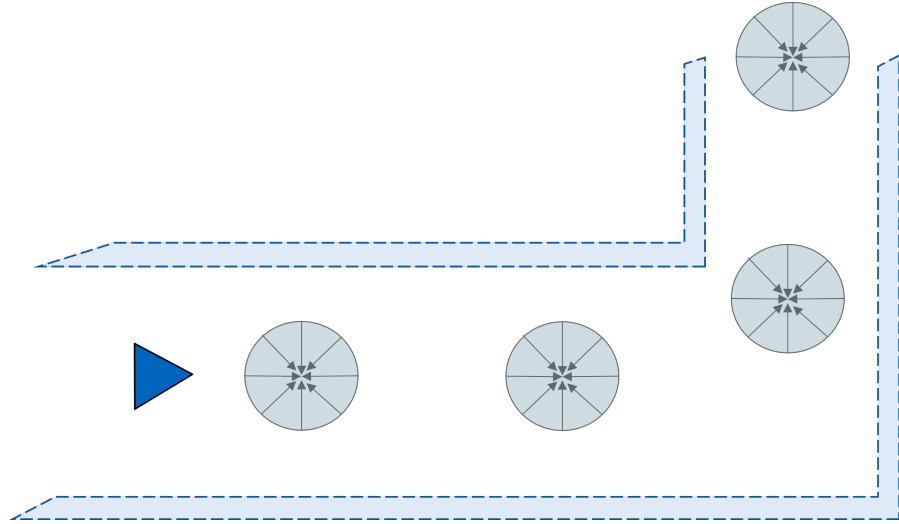


Figure 8.6: Way point corridor model.

boundaries. The blue triangle is the aerial vehicle traveling inside the corridor. Circles with inward arrows are the point sink elements that represent the consecutive *way points* within the corridor.

Last but certainly not least, the corridor boundaries can be represented as obstacles within the corridor model. In this *obstacle model*, a hard constraint is enforced on the corridor boundaries. Consequently, although the vehicle has unrestricted movement within the corridor, it is prevented from crossing the defined corridor boundaries. To compute the flow velocity within the corridor, the corridor boundary is segmented into small line segments, and the flow field within the corridor is determined using Equa-

tion 8.4 and Equation 8.5.

$$u = \frac{-\sigma_g}{2\pi} \frac{x - x_g}{r_g^2} + \sum_{s=1}^S \sum_{k=1}^K \frac{\gamma_{sk}}{2\pi} \frac{x - x_{sk}}{r_{sk}^2} \quad (8.4)$$

$$v = \frac{-\sigma_g}{2\pi} \frac{y - y_g}{r_g^2} + \sum_{s=1}^S \sum_{k=1}^K \frac{-\gamma_{sk}}{2\pi} \frac{y - y_{sk}}{r_{sk}^2} \quad (8.5)$$

Here, S is the number of corridor boundaries that are disconnected. K is the number of point vortex elements placed on the corridor boundary. The position of a vortex element on a given corridor boundary is denoted as (x_{sk}, y_{sk}) , and the associated vortex strength is γ_{sk} . Figure 8.7 presents a schematic representation of the *obstacle model*. In

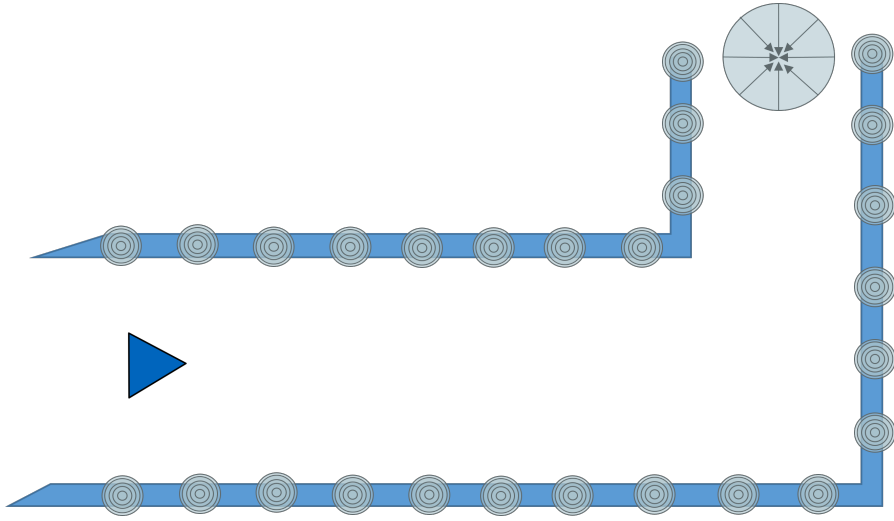


Figure 8.7: Obstacle corridor model.

this figure, the blue regions represent the corridor boundaries. The solid lines around the corridor boundaries indicate a hard constraint enforced on the corridor boundaries. The blue triangle is the aerial vehicle traveling inside the corridor. The circle with inward-facing arrows represents the point sink element at the end goal position. Circles with spirals inside are the point vortex elements placed on the corridor boundary. This approach can be particularly valuable for corridors restricted by no-fly zones.

8.3.3 Tactical Deconfliction

Tactical deconfliction is the process of managing the separation between aircraft to prevent collisions and ensure safe air traffic flow in real-time in ATC. Given its low computational load, the proposed PGFlow algorithm is a promising tool for tactical deconfliction. To prevent real-time collisions between aerial vehicles, each vehicle can be represented as a point source element, and the flow velocities can be calculated using

Equation 2.21 and Equation 2.22 as described in Section 2.2.1. Equation 2.21 and Equation 2.22 do not specify the direction of the avoidance maneuver. Therefore, to be more synced with current ATC regulations, Equation 2.21 and Equation 2.22 are further modified. An additional vortex element assigned to each vehicle is introduced in Equation 8.6 and Equation 8.7.

$$u = \frac{-\sigma_g}{2\pi} \frac{x - x_g}{r_g^2} + \sum_{n=1}^N \frac{\sigma_n}{2\pi} \frac{x - x_n}{r_n^2} + \frac{\gamma_n}{2\pi} \frac{x - x_n}{r_n^2} + \sum_{s=1}^S \sum_{k=1}^K \frac{\gamma_{sk}}{2\pi} \frac{x - x_{sk}}{r_{sk}^2} \quad (8.6)$$

$$v = \frac{-\sigma_g}{2\pi} \frac{y - y_g}{r_g^2} + \sum_{n=1}^N \frac{\sigma_n}{2\pi} \frac{y - y_n}{r_n^2} + \frac{-\gamma_n}{2\pi} \frac{y - y_n}{r_n^2} + \sum_{s=1}^S \sum_{k=1}^K \frac{-\gamma_{sk}}{2\pi} \frac{y - y_{sk}}{r_{sk}^2} \quad (8.7)$$

This vortex element makes vehicles always alter their course to the right in case of a conflict, obeying the so-called *right of way* rule in ATC.

8.4 Tactical Deconfliction of two Medical Drones

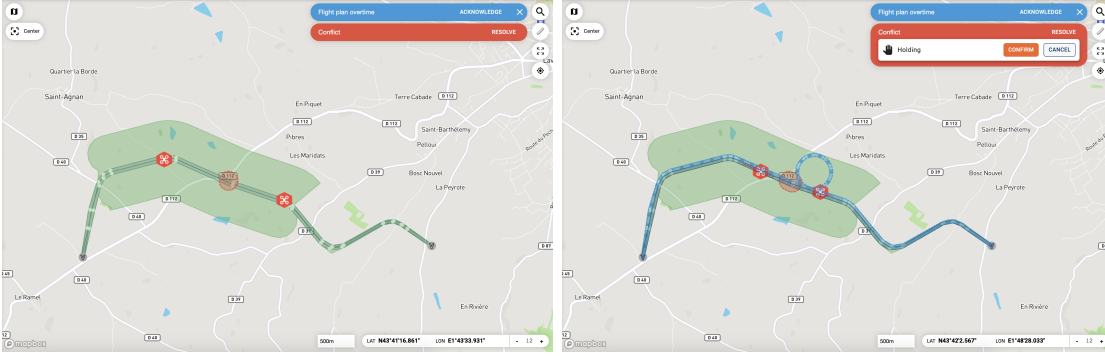
This section presents a realistic example scenario to test the performance of the proposed corridor models. Performance of the proposed tactical deconfliction and corridor models are evaluated through hardware experiments conducted in the indoor flight facility described in Section 4.1. The quad-rotor platform given in Section 4.1.1 is used for the flight tests. Additionally, different communication schemes between the vehicles are also inspected in this section.

8.4.1 Motivation

The motivation for the experiment scenario presented in this section comes from the European Union's Horizon 2020 Tactical Instrumental Deconfliction and In-flight Resolution (TindAIR) project[95]. The scenario occurs near Toulouse in France, where a UAV is transporting a critical emergency kit from a clinic to an accident site. On its return journey, the UAV encounters another drone traveling in the opposite direction, transporting an extension of the emergency kit. Given that both drones are utilizing the same corridor, without intervention, a collision is imminent. This conflict is illustrated in Figure 8.8a.

The tactical deconfliction solution provided by TindAIR involves instructing one of the drone operators to adjust the speed and altitude or follow an elliptical path. Figure 8.8b illustrates how the collision is averted with the TindAIR solution, allowing both drones to proceed safely toward their destinations.

8.4 Tactical Deconfliction of two Medical Drones



(a) Encountering two emergency drones on the same corridor poses a collision risk if no action is taken. (b) The TindAIR solution to this conflict involves instructing drone pilots to adjust their courses to prevent a collision.

Figure 8.8: Motivational scenario from European Union’s TindAIR project[95]: Tactical deconfliction for two medical flights that intersect along the same corridor.

8.4.2 Experiment Results and Discussions

For the first experiment, the motivational scenario generated by the TindAIR project[95] is recreated within the indoor flight arena in Toulouse, France. As the experiments had to be conducted in a scaled environment, only the corridor segment in the immediate vicinity of the conflict is modeled, rather than the entire flight. In Figure 8.9, two med-

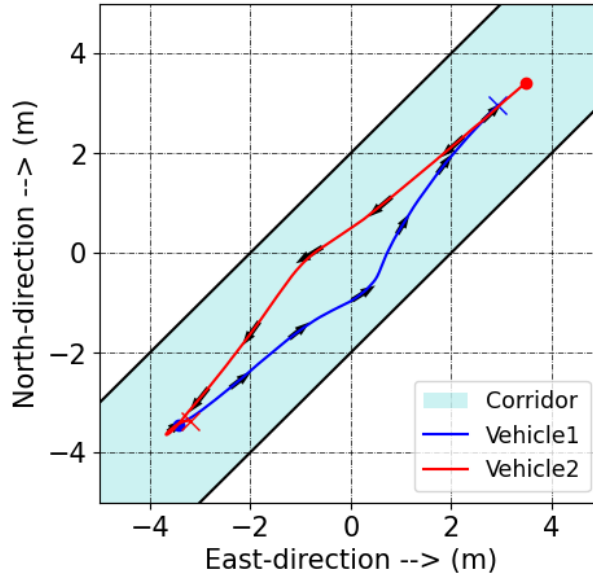


Figure 8.9: Two medical drones heading opposite directions encounter in the same corridor. Limited communication between vehicles.

ical drones heading in opposite directions encounter the same corridor. The Red drone

is responding to an emergency, and the Blue drone is returning to the base. The red drone holds higher priority as it is responding to an emergency, whereas the blue drone is returning to base. The corridor is represented in this scenario with the *obstacle model*. Vehicles are free to maneuver within the corridor but are prohibited from crossing its boundaries. Figure 8.9 considers a full communication case where both vehicles inform each other about their positions at all times. The black arrows plotted on the trajectories are the desired velocity vectors calculated by the PGFlow method. These vectors can be directly fed to the vehicle's automatic flight controller in autonomous flight. Alternatively, PGFlow method can be used as an advisory system in more controlled airspace, and the desired velocity vectors can be communicated to the pilots or the air traffic controller as suggested tactical deconfliction maneuvers.

Figure 8.10 considers a limited communication case where the Red receives no information about the Blue vehicle's position. However, the Blue vehicle still gets updates on the position of the Red vehicle. In this scenario, only the blue medical drone executes a tactical deconfliction maneuver when the two vehicles encounter each other. It is worth

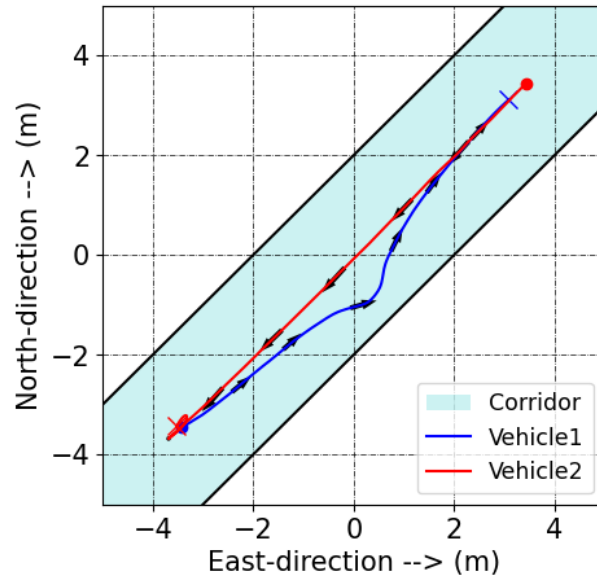


Figure 8.10: Two medical drones heading opposite directions encounter in the same corridor. Full communication between vehicles.

noting that, thanks to the vortex element placed on the vehicles, the Blue vehicle resolves the conflict by steering to the right in both limited and full communication cases.

Figure 8.11 presents the resultant paths of a similar scenario. In this case, too, two medical UAVs encounter in the same corridor. Unlike the previous scenario, the corridor in this scenario features a 90-degree curve, as shown in Figure 8.11. Moreover, for this scenario, rather than using the *obstacle model*, corridor boundaries are modeled using *checkpoint model*. Figure 8.12 presents the results for the full communication case. In this instance, both vehicles maneuver right to avoid each other. Notice that

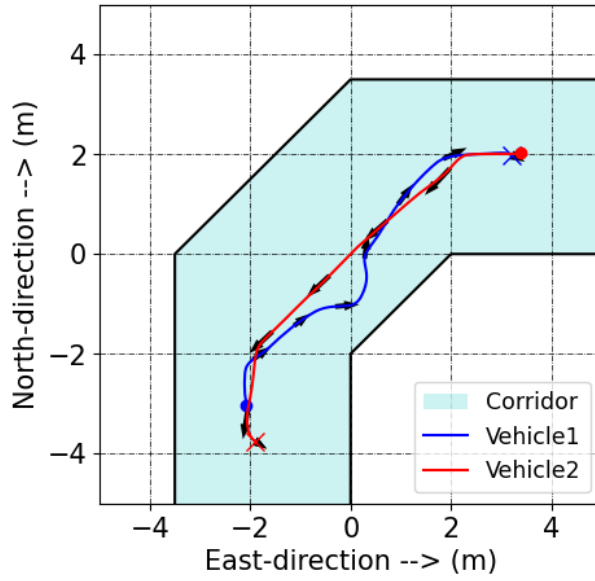


Figure 8.11: Two medical drones heading opposite directions encounter in the same corridor. Limited communication between vehicles.

in Figure 8.12, the red vehicle adjusts its path slightly less than the blue vehicle. This discrepancy arises from the assignment of uneven sources to each vehicle. By allocating stronger or weaker sources, the repulsiveness, and thus the priority in airspace, can be determined. Figure 8.11 presents the resultant trajectories for the limited communication case. In this case, only the blue vehicle changes direction to avoid conflict. During the tactical deconfliction maneuver, the next *check point* in the corridor is still active. Therefore, the blue vehicle does not violate the corridor limits in this test. However, unlike the *obstacle model*, the *check point model* does not guarantee that vehicles remain within the corridor.

8.5 Tactical Deconfliction of an Air Taxi and an Intruder

This section examines a more challenging scenario to effectively examine the distinctions between various corridor models and shapes. In these scenarios, a non-cooperative agent (intruder) intersects with an air taxi as the air taxi maneuvers within a corridor. Performance of the proposed tactical deconfliction and corridor models are evaluated through hardware experiments conducted in the indoor flight facility described in Section 4.1. The quad-rotor platform given in Section 4.1.1 is used for the flight tests.

8.5.1 Experiment Results and Discussions

In the first experiment scenario, a straight corridor is considered to compare different corridor models. In this scenario, the air taxi travels within the corridor and encounters

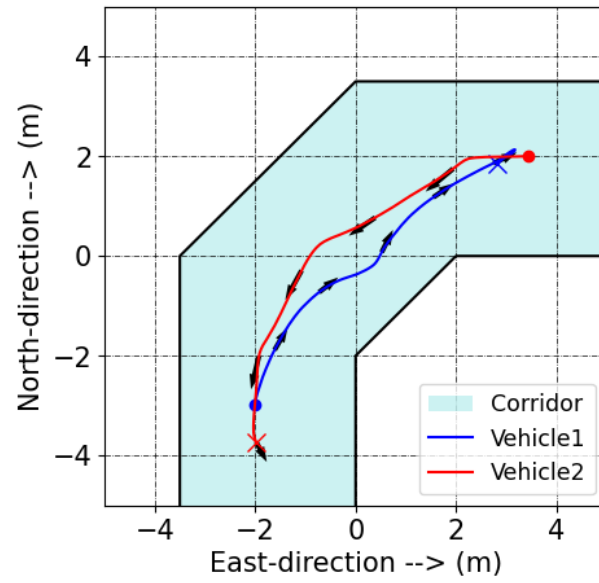


Figure 8.12: Two medical drones heading opposite directions encounter in the same corridor. Full communication between vehicles.

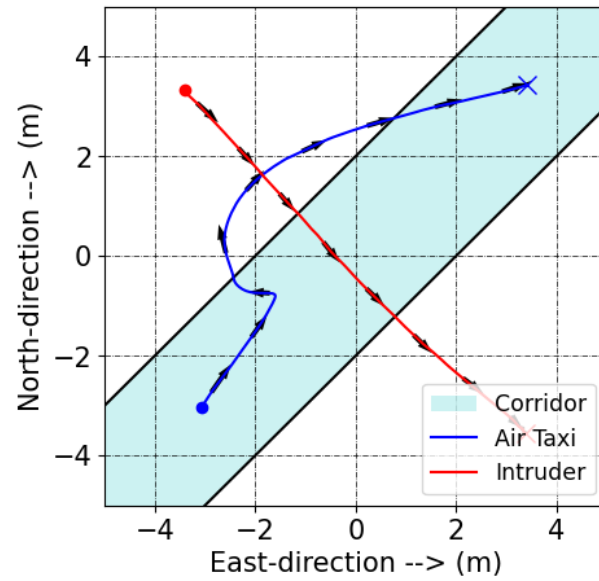


Figure 8.13: A non-cooperative agent and an air taxi encounter. No corridors are modeled for the vehicles.

a non-cooperative agent. The air taxi is tasked with resolving the conflict while adhering to air traffic regulations.

Figure 8.13 presents the resulting trajectories for the nominal case with no specified cor-

ridor boundaries. The intruder is non-cooperative and unable to alter its trajectory. The air taxi maneuvers behind the intruder to resolve the conflict. Since no corridor model is in effect, this resolution breaches corridor boundaries. Figure 8.14 shows the resolution

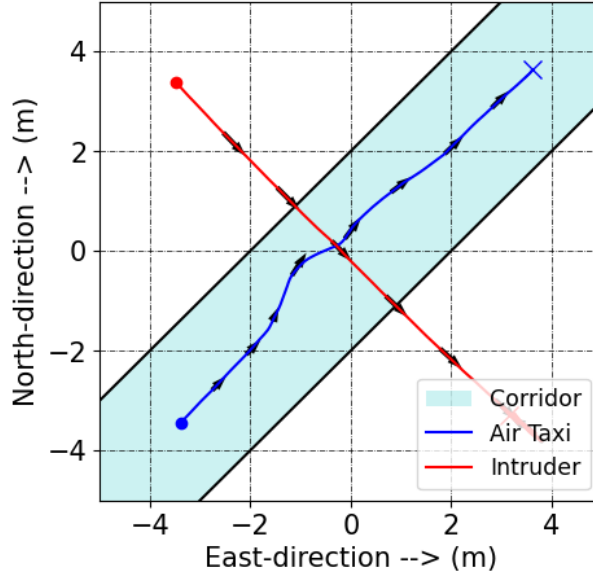


Figure 8.14: The Obstacle Model. The vehicle can move freely within the corridor but cannot breach the corridor limits.

when the *Obstacle Model* is utilized as the corridor model. In this model, the vehicle is permitted to move freely within the corridor but cannot cross its boundaries. When the conflict with the intruder arises, the air taxi adjusts its path to maneuver behind the intruder and evade any potential collision. Throughout this maneuver, the corridor boundaries also apply pressure on the vehicle, ensuring it stays within the designated space. No corridor boundaries were specified for the vehicle for the results shown in Figure 8.15. Instead, utilizing the *Way Point Model*, the vehicle is guided to consecutive waypoints through the corridor. When the air taxi encounters the intruder, it adjusts its trajectory to prevent a collision. Moreover, as the next waypoint also draws the vehicle to itself, the air taxi successfully stays within the corridor. In this model, the strength of the waypoint is a design parameter. The likelihood of breaching the corridor boundary is reduced. If the waypoint is strong, the vehicle may also have difficulty avoiding a collision due to the suction of the waypoint. If the waypoint is weak, the vehicle has the freedom to evade any conflicts. However, there is no assurance that avoidance maneuvers will not breach the corridor boundaries.

The same scenario with the air taxi and the intruder is solved in the next experiment for a differently shaped corridor. One of the strengths of the proposed PGFlow algorithm is its capability to compute paths around obstacles of arbitrary shapes. As shown in Figure 8.16, the L-shaped corridor is modeled using the *Obstacle Model*. Without requiring additional waypoints for guidance, the PGFlow algorithm can effectively guide the vehi-

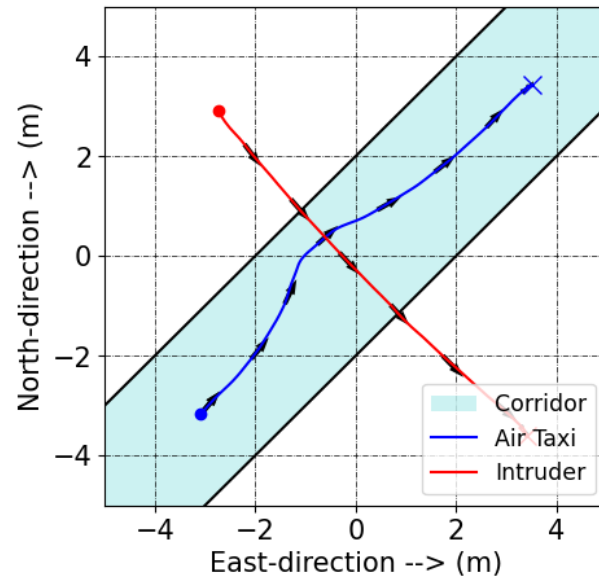


Figure 8.15: The Way Point Model. The vehicle is guided from one waypoint to the next, but there is no assurance that corridor boundaries will be respected.

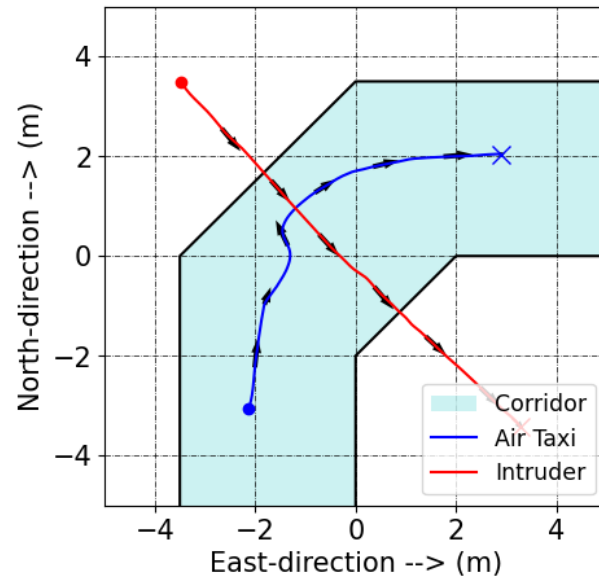


Figure 8.16: The Obstacle Model. L-Shaped Corridor.

cle through this corridor. Moreover, when the vehicle within the corridor encounters an intruder, it automatically alters its course to resolve conflict. PGFlow method ensures both collision avoidance and adherence to the corridor limits during this maneuver. Alternatively, the PGFlow algorithm can also be employed to utilize waypoints that

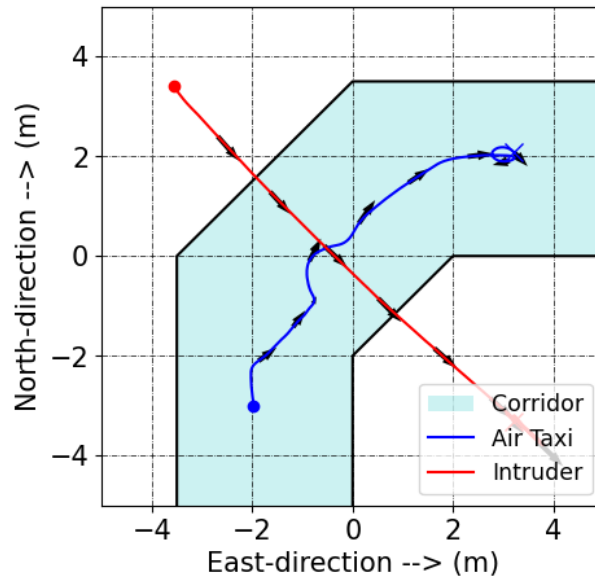


Figure 8.17: The Way Point Model. L-Shaped Corridor.

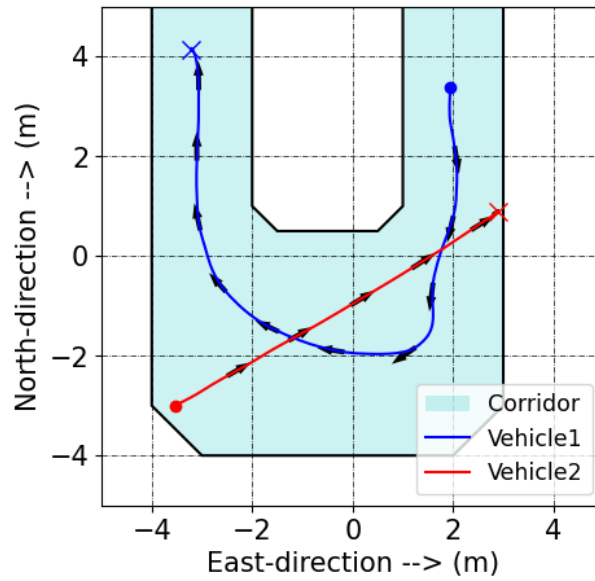


Figure 8.18: The Obstacle Model. U-Shaped Corridor.

navigate a vehicle through maze-like paths. The result for the waypoint model in an L-shaped corridor is presented in Figure 8.17. Once again, with the waypoint model too, the vehicle successfully avoids the intruder and safely reaches its final destination. It is worth noting that in Figure 8.17, the vehicle executes a much sharper turn compared to Figure 8.16. This maneuver is due to the position and strength of the waypoint relative

to the intruder.

Finally, the intruder and the air taxi case are tested in a U-shaped corridor as given in Figure 8.18. In this scenario, the corridor is modeled as an obstacle using PGFlow algorithm. The proposed guidance algorithm does not require waypoints to navigate through complex-shaped corridors. Instead, it can autonomously guide vehicles around objects of arbitrary shapes.

8.6 Conclusions

This chapter introduces additional features designed to improve safety and promote social acceptance of UAM as well as enhancements to PGFlow algorithm to be used as a tactical deconfliction tool for . The feasibility and effectiveness of the proposed air traffic management enhancements in complex airspace scenarios were assessed through a flight test campaign in an indoor test facility. The experimental findings illustrate how well the PGFlow algorithm performs in addressing conflicts and ensuring secure trajectories. The proposed crowd avoidance and emergency landing features both utilize the 3-D guidance property of PGFlow to improve safety in urban airspace and promote social acceptance. The emergency landing feature aims to guide the vehicle to the nearest emergency landing pad in case of an unforeseen failure. During flight tests, the guidance algorithm demonstrated its effectiveness by successfully directing a malfunctioning vehicle to the nearest emergency landing pad while allowing other vehicles to proceed with their missions without interruption. The crowd avoidance feature assigns a dynamic 3-D source element to densely populated regions to prevent vehicles from flying over this area. The experiment results indicate that the proposed guidance approach effectively achieves dynamic crowd avoidance while simultaneously ensuring collision and obstacle avoidance. In the experiments, it was observed that the source strength assigned to the population needs to be fine-tuned to balance the need to push vehicles away from crowded areas adequately while preventing excessive avoidance. Additionally, the combined influence of sources and sinks may result in stagnation points, especially in scenarios where the population is situated in close proximity to the goal positions. These enhancements collectively advance the safety and social acceptability of urban air mobility operations.

Experimental results demonstrate that the proposed guidance algorithm has the potential to serve as an effective tool for the successful implementation of urban air mobility systems in the future.

Furthermore two realistic air traffic control scenarios are considered to evaluate the tactical deconfliction performance of the guidance algorithm: the conflict between two medical drones sharing the same corridor and the conflict involving an air taxi and a non-cooperative intruder. Various corridor models were investigated for each scenario, including the waypoint, checkpoint, and obstacle methods. Each model presents distinct trade-offs and advantages concerning strictness, maneuverability, and tactical deconfliction.

By employing *way point Method* and *checkpoint method*, where the latter is a less strict

form of the former, users can tailor the enforcement of corridor bounds to their preference. Weaker waypoints offer greater maneuverability, allowing potential conflicts to be safely resolved, while stronger waypoints assist vehicles in following the corridor more accurately. However, it is important to note that neither method ensures absolute adherence to the corridor bounds.

In the *Obstacle Method*, corridor limits are modeled as obstacles using the PGFlow algorithm. Modeling corridors as obstacles makes use of the collision avoidance assurance of the PGFlow algorithm. PGFlow, can automatically generate collision-free paths through corridors of arbitrary shapes. This corridor model offers a dual advantage. Firstly, its implementation is straightforward: once the corridors are defined, a collision-free trajectory following the corridor to the destination is guaranteed without the need for additional parameter adjustments. Secondly, vehicles can maneuver freely within the corridor while strictly remaining within the boundaries, ensuring that any potential conflicts can be safely resolved within the corridor's limits, provided it is sufficiently wide for maneuvers. This method proves particularly valuable in situations where the airspace outside the corridor is designated as a no-fly zone, posing significant danger if the corridor boundaries are breached.

In this chapter, the algorithm's effectiveness was also assessed in scenarios involving complete and restricted aerial vehicle communication. In instances of full communication, the proposed guidance technique enables users to fine-tune the relative priorities of each vehicle. This feature ensures that high-priority vehicles automatically receive the right of way while low-priority ones yield accordingly. Conversely, even in situations of limited communication, the proposed guidance algorithm successfully generates safe paths. As long as the intruder's position is available, the PGFlow algorithm can ensure collision-free navigation within the corridor.

The experimental findings indicate that the proposed guidance algorithm effectively navigates vehicles through corridors of diverse shapes, resolving conflicts in real-time, even in scenarios with limited communication.

To conclude, the robust performance of the PGFlow algorithm across various situations ensures safe navigation in compliance with air traffic regulations in urban airspace.

9 Conclusions

This dissertation focuses on the integration of potential flow theory into the field of aerial guidance methodologies, ultimately leading to the creation of the Potential Guidance Flow (PGFlow) Algorithm. This thesis has two main objectives: (1) the development of a nature-inspired guidance algorithm based on potential flow theory in the fluid dynamics domain and (2) the integration of the proposed guidance algorithm to urban air mobility applications to contribute to the urban air traffic control efforts. The following sections elaborate on the essential findings and contributions of this dissertation.

9.1 Summary of Findings and Contributions

This dissertation proposes a nature-inspired guidance algorithm named Potential Guidance Flow (PGFlow) designed for the real-time guidance of multiple aerial vehicles operating in cluttered environments. The development of the PGFlow Algorithm is the primary contribution of this dissertation. Moreover, this dissertation also introduces additional features incorporated into the guidance method to integrate it into existing air traffic control systems. An essential aspect of the guidance method's development involves assessing its effectiveness and feasibility. Therefore, this study consists of an extensive simulation and hardware experiment campaign. Hardware experiments are conducted in both indoor and outdoor test facilities using various platforms to validate the proposed algorithm's performance and robustness.

9.1.1 Impact as a Guidance Method

PGFlow method uses potential flow theory to construct a guidance vector field that can provide collision-free solutions. Potential flow theory defines flow patterns through a potential function and is commonly used to model the flow of an inviscid and irrotational ideal fluid in fluid mechanics. Through potential flow theory, basic flow elements can be combined to simulate fluid flow around arbitrarily shaped objects submerged in an ideal fluid. By definition, fluid cannot penetrate solid objects, and the resultant potential field lacks local minima in potential flow theory. Consequently, the guidance vector field generated using potential flow theory ensures collision-free guidance around obstacles of arbitrary shapes without local minimum constraints. Additionally, the proposed guidance algorithm imposes minimal computational load and is capable of generating real-time guidance vectors for multiple vehicles. While the inherent collision avoidance and absence of local minima properties of potential flow theory make it suitable for path planning, additional modifications to the potential field are required to adapt it for the guidance of aerial vehicles. In this thesis, a potential flow theory is tailored to meet the

9 Conclusions

requirements of guidance applications. To achieve this, supplementary flow elements are introduced to serve as goal positions, facilitate collision avoidance between vehicles, and establish a safety perimeter around obstacles.

The safety perimeter around the obstacles is generated by an additional safety feature named *safety source element*. While trajectories tangent to obstacle boundaries are acceptable in fluid dynamics, they are not feasible for path planning of aerial vehicles. The *safety source element* is an additional source element that moves with the vehicle and can generate trajectories that stay a desired distance away from obstacles while ensuring convergence to the goal position. The proposed *safety source element* gives the user the flexibility to define and adjust the safety perimeter even in real time.

This dissertation also proposes enhancements for the PGFlow method aimed at achieving faster and more agile flight. Two approaches, path propagation, and simple position propagation, are introduced to improve the smoothness of trajectories, resulting in significantly faster flights. The path propagation method calculates alternative trajectories propagated into the future and selects the optimal path based on minimum curvature criteria. This method generates smoother paths, allowing for higher speeds while maintaining safety. It also considers vehicle dynamics for more accurate predictions. Simulation and experimental results demonstrate the effectiveness of this approach in reducing path curvature and enabling faster flight. On the other hand, the simple position propagation method offers a computationally efficient alternative by predicting collisions using projected line segments and initiating avoidance maneuvers. Despite its simplicity, this approach achieves notable reductions in path curvature, comparable to the path propagation approach, with significantly lower computational load. Both methods contribute to enabling faster and more agile flight by minimizing trajectory curvature and maximizing vehicle speed.

Last but not least, the proposed guidance algorithm is extended to 3-dimensional problems. To extend the 2D guidance algorithm to 3D, instead of directly incorporating 3D building models into the calculations, 2D cross-sections of buildings are stacked. This approach helps maintain a low computational load and enables real-time guidance capability.

In Chapter 3, the PGFlow algorithm is analyzed within a controlled environment through simulations. Simulations serve as an adjustable platform to evaluate performance across diverse scenarios. This dissertation conducts two key simulation studies: an element size study and a computational load study. The element size study reveals the sensitivity of the PGFlow algorithm to the element size. Inadequate representation of obstacles due to an insufficient number of elements can compromise the obstacle avoidance capability of the proposed guidance method. The computational load of a guidance algorithm is essential in real-time applications, particularly for micro aerial vehicles with limited computational resources. The computational load study identifies vehicle and obstacle numbers as the most significant parameters affecting computation time.

Simulation results with a single vehicle illustrate the algorithm's obstacle avoidance and goal convergence properties, robustness against disturbances, and adaptability to dynamic environments. Additionally, the algorithm's capability to navigate around arbitrarily shaped obstacles without getting trapped in local minima is demonstrated.

Collision avoidance capabilities are showcased in simulations involving multiple vehicles, with effective navigation in cluttered environments while avoiding collisions.

After simulations, this study moves on to the experimental validation of the PGFlow algorithm. Initially, experiments are conducted in a highly controlled environment with a single hovering platform and static goal positions within an indoor flight facility. Chapter 5 presents the results of the experiments conducted on VTOL platforms. The scalability of the proposed algorithm is demonstrated through experiments with multiple vehicles. The hardware experiments with static goal positions highlight the PGFlow algorithm's inherent obstacle avoidance and collision avoidance properties. Transitioning to dynamic target tracking, more challenging scenarios are considered where quadrotors must track a moving target while navigating a cluttered environment. Through an extensive experiment campaign, the PGFlow algorithm's capability to dynamically adjust the guidance vector field, ensuring both target tracking and collision avoidance simultaneously, is observed. The low computational load of the PGFlow algorithm is demonstrated through successful real-time adaptation to changing circumstances.

Chapter 6 investigates the robustness of the PGFlow method under environmental disturbances through hardware experiments conducted in both indoor and outdoor test facilities. The PGFlow method inherently exhibits robustness against disturbances as a vector field guidance approach. The proposed algorithm guides the vehicles by an instantaneous guidance vector calculated at their current position. Therefore, even if the vehicle deviates from its original path due to disturbances, a new guidance vector can be calculated at the new position, guiding the vehicle toward the global minimum. This thesis introduces an additional correction term to further mitigate deviation from the original path under strong disturbances. This correction term incorporates position error into the controller, similar to a PD (Proportional-Derivative) controller. Indoor experiments demonstrate that despite wind disturbances, vehicles navigate around obstacles and reach their destinations successfully. The effect of the proposed correction term is also evident under wind disturbances. Outdoor experiments pose more significant challenges compared to indoor tests, involving not only wind disturbances but also greater sensor errors. Nonetheless, both indoor and outdoor flight tests illustrate that the PGFlow algorithm offers effective disturbance rejection.

VTOL platforms are effective in establishing basic concepts in the proposed guidance algorithm. Guiding a fixed-wing aircraft is more challenging than guiding quadrotors due to the constraints imposed by minimum speed and turn radius requirements. Chapter 7 inspects the integration of PGFlow algorithm to platforms with minimum speed constraints. In order to operate fixed-wing platforms, the automatic flight controller is modified to convert desired velocity input into heading error and desired roll angle input. Simulations and real-flight experiments are conducted on a fixed-wing platform to demonstrate the feasibility of the proposed approach on platforms with minimum speed constraints. Experiment results suggest the effectiveness of the PGFlow method in navigating fixed-wing aircraft while respecting the no-fly zones and maintaining safety margins. Moreover, the scalability of the PGFlow method is demonstrated through experiments conducted in varied airspace sizes. Both simulations and flight experiments affirm that fixed-wing aircraft can effectively follow the guidance vector generated by

the PGFlow method without exceeding the flight envelope limits.

In conclusion, this thesis presents a comprehensive exploration of the PGFlow method through a combination of theoretical analysis, simulation studies, and experimental validations. The efficacy and versatility of the PGFlow algorithm have been demonstrated across various scenarios and environments. The algorithm showed remarkable performance in collision avoidance, obstacle navigation, target tracking, and disturbance rejection, making it a promising solution for the guidance of multiple aerial vehicles in real time. Moreover, the robustness and adaptability of the algorithm have been confirmed through experiments with different types of vehicles and airspace sizes. The enhancements for faster and more agile flight trajectories and the extension of the algorithm to 3D environments further broaden its applicability while maintaining computational efficiency. Overall, the findings presented in this dissertation contribute to aerial vehicle guidance, offering a robust and adaptable solution with potential implications for a wide range of industries and applications.

9.1.2 Impact on Urban Air Mobility Applications

The Urban Air Mobility (UAM) concept involves aerial vehicles with varying designs, sizes, maneuverability, automation levels, and communication capabilities occupying the same limited urban airspace, previously unregulated. Given the expected volume of traffic and the constrained airspace, developing effective air traffic management schemes is a must. This thesis is motivated by the challenges associated with guiding multiple vehicles in real-time and integrating the proposed guidance algorithm into the UAM concept. The proposed guidance algorithm is adaptable and can be customized to adhere to urban air traffic regulations.

Chapter 3 demonstrates some key features of the proposed guidance algorithm. The capability to guide multiple aerial vehicles simultaneously in a cluttered environment makes PGFlow method an ideal candidate for UAM applications. Furthermore, the concept of assigning uneven source strengths to vehicles provides a mechanism for prioritizing certain vehicles, which is valuable for applications like urban air traffic management or emergency response.

The robustness assessment was done in Chapter 6 also gives valuable insights on possible UAM applications. The wind disturbance in the urban airspace is a significant disturbance that cannot be neglected. Conducting experiments in a scaled urban environment under wind disturbances, PGFlow is shown to be a promising candidate for guiding aerial vehicles in the urban airspace.

Lastly, in Chapter 8, the proposed PGFlow algorithm is adapted to serve as a guidance tool for . First, 3-dimensional problems such as emergency landing and crowd avoidance within the city are addressed. The additional features introduced in this section aim at enhancing safety and promoting social acceptance of UAM. These modifications to the guidance algorithm are validated through hardware experiments conducted in an indoor flight arena.

As demonstrated in flight tests, the emergency landing feature guides malfunctioning vehicles to the nearest emergency landing pad. It effectively redirected a malfunction-

ing vehicle while allowing others to continue their missions uninterrupted. The crowd avoidance feature assigns a dynamic 3-D source element to densely populated regions to prevent vehicles from flying over them, achieving dynamic crowd avoidance while ensuring collision and obstacle avoidance. These enhancements collectively advance the safety and social acceptability of urban air mobility operations, as demonstrated by experimental results, indicating the potential effectiveness of the proposed guidance algorithm in future UAM implementations. Next, the feasibility and effectiveness of the PGFlow algorithm in tactical deconfliction in complex airspace scenarios were evaluated through a flight test campaign in an indoor test facility. This thesis presents two realistic scenarios: conflicts between two medical drones sharing the same corridor and conflicts involving an air taxi and a non-cooperative intruder. Various corridor models were explored for each scenario, including the waypoint, checkpoint, and obstacle methods, each offering distinct trade-offs and advantages in terms of strictness, maneuverability, and tactical deconfliction.

The algorithm's effectiveness was also assessed in scenarios involving complete and restricted aerial vehicle communication. In instances of full communication, the PGFlow algorithm allows users to adjust the relative priorities of each vehicle, ensuring high-priority vehicles receive the right of way. Even in situations of limited communication, the algorithm generates safe paths, ensuring collision-free navigation within the corridor as long as the intruder's position is available.

The experimental findings demonstrate that the proposed guidance algorithm effectively navigates vehicles through corridors of diverse shapes, resolving conflicts in real-time, even in scenarios with limited communication. The robust performance of the PGFlow algorithm ensures safe navigation in compliance with air traffic regulations.

In conclusion, this dissertation has presented a comprehensive investigation into the development and application of the PGFlow guidance algorithm within the context of UAM. Various aspects of the algorithm's capabilities and adaptability have been explored, ranging from its ability to guide multiple aerial vehicles simultaneously in cluttered urban environments to its robustness in handling significant disturbances like wind. Moreover, the adaptation of the PGFlow algorithm for UTM systems has been demonstrated, showcasing its feasibility and effectiveness in complex airspace scenarios through rigorous flight testing. Overall, the findings presented in this thesis highlight the significant contributions of the PGFlow guidance algorithm towards advancing the safety, efficiency, and social acceptability of UAM operations.

9.2 Limitations and Future Research

While the proposed guidance algorithm in this dissertation presents significant advancements in autonomous aerial vehicle guidance and urban air traffic management, several limitations and opportunities for future work require consideration.

First and foremost, the proposed guidance algorithm relies on complete knowledge of the flight arena and the positions of the vehicles within. Although it is not unrealistic to have this information with current mapping tools for urban air mobility applications,

9 Conclusions

this property limits the proposed algorithm from being used in unknown environments. Several adjustments are proposed in the literature to overcome this drawback. For example, Velagic et al. [78] propose integrating the panel method with an occupancy map to navigate within unknown environments with a ground robot. Similarly, Cruz et al. [79] used a range finder to model unknown obstacles near the aerial vehicle and evaluated their approach through hardware in the loop experiments. Future research can be conducted on developing additional methods for guidance in unknown environments and evaluation of these methods on aerial vehicles with flight tests.

Element size and computational workload studies presented in Chapter 3 suggest that the proposed method is sensitive to element size and cannot provide obstacle avoidance unless obstacles are defined with sufficient elements. However, increasing the element number also increases the computational load of the guidance method. In the current experimental setup, this did not pose a limitation; however, for larger-scaled applications, element number and computation methods need to be optimized by running the calculations in parallel or only considering the obstacles and vehicles in a smaller region of interest. Optimizing computational efficiency is out of the scope of this dissertation. The resultant trajectories generated by the PGFlow method are not optimized for parameters like travel time, distance, or energy consumption. PGFlow algorithm instead relies on inherent properties of harmonic potential fields to generate smooth paths that are not subject to local minima limitations. Having said that, it may be a future research avenue to combine the proposed PGFlow algorithm with distance or energy-optimizing guidance methods for enhanced performance. In this respect, a path propagation and curvature optimal path selection method is developed. However, paths can be propagated in only a short time horizon due to computational limitations. Using parallel computing tools can encounter this drawback in the future.

Furthermore, PGFlow algorithm does not take vehicle dynamics into account. Coupling the guidance algorithm with vehicle models is a future research area. Including vehicle dynamics in calculations would enable the generation of trajectories that also consider vehicle limitations.

The 3-D extension of the method also opens new avenues for future research. Instead of stacking 2-D cross-sections to model 3-D flow, actual 3-D models of the obstacles can be used. This research calls for computational optimization of the current guidance code to accommodate the increased number of elements from the 3-D obstacle models.

The corridor and population models introduced in Chapter 8 enable integration of the proposed guidance algorithm with current air traffic control regulations. However, the proposed corridor models also bring new design parameters, such as waypoint strength or positioning, into equations. Although these design parameters enable users to control the airspace fully, tuning many parameters takes away from the ease of implementation. In conclusion, while the proposed guidance algorithm represents a significant advancement in autonomous aerial vehicle guidance and urban air traffic management, several limitations have been identified. However, each limitation presents an opportunity for further research and innovation. Future work could focus on developing methods for guidance in unknown environments, optimizing computational efficiency, integrating vehicle dynamics into calculations, and exploring the potential of 3-D extensions of the

proposed method. Addressing these limitations will enhance the performance and applicability of the proposed algorithm.

9.3 Concluding Remarks

To conclude, this dissertation contributes to the autonomous aerial vehicle guidance and urban air traffic management field, mainly through the development and application of the PGFlow algorithm. Through a meticulous combination of theoretical analysis, simulation studies, and real-world experimentation, the efficacy, versatility, and adaptability of PGFlow method have been demonstrated.

The algorithm's main advantage is its ability to navigate multiple aerial vehicles simultaneously in cluttered urban environments in real-time. Its resilience against significant disturbances like wind and its integration into UTM systems for complex airspace management have been demonstrated with extensive experiment campaigns.

While acknowledging certain limitations, these drawbacks only serve as stepping stones for future research and innovation. Conducting further research on guidance in unknown environments, optimizing computational efficiency, incorporating vehicle dynamics, and exploring 3-D extensions, the potential of PGFlow can be further enhanced.

In essence, this dissertation embodies extensive research in the development of the PGFlow algorithm. It highlights the contributions of the proposed algorithm in aerial guidance and urban air mobility domains through its efficacy, robustness, adaptability, and potential for enhancing safety and efficiency.

Bibliography

- [1] Unmanned systems and robotics database. <https://www.auvsi.org/usrd>, n.d. Accessed: January 18, 2024.
- [2] M. Ball, C. Barnhart, G. Nemhause, and A. Odoni. Air transportation: irregular operations and control. *Handbooks Oper. Res. Manage. Sci.*, 14:1–67, 2007.
- [3] J. del Cerro, C. Cruz Ulloa, A. Barrientos, and J. de León Rivas. Unmanned aerial vehicles in agriculture: A survey. *Agronomy*, 11(2):203, 2021. URL: <https://doi.org/10.3390/agronomy11020203>, doi:10.3390/agronomy11020203.
- [4] X. Li and A. V. Savkin. Networked unmanned aerial vehicles for surveillance and monitoring: A survey. *Future Internet*, 13(7):174, 2021.
- [5] S. W. Cho, H. J. Park, H. Lee, D. H. Shim, and S.-Y. Kim. Coverage path planning for multiple unmanned aerial vehicles in maritime search and rescue operations. *Computers & Industrial Engineering*, 161:107612, 2021.
- [6] V. Ajith and K. Jolly. Unmanned aerial systems in search and rescue applications with their path planning: a review. In *Journal of Physics: Conference Series*, volume 2115, page 012020. IOP Publishing, 2021.
- [7] A. Gupta and X. Fernando. Simultaneous localization and mapping (slam) and data fusion in unmanned aerial vehicles: Recent advances and challenges. *Drones*, 6(4):85, 2022.
- [8] E. Jeong, J. Seo, and J. Wacker. Literature review and technical survey on bridge inspection using unmanned aerial vehicles. *Journal of Performance of Constructed Facilities*, 34(6):04020113, 2020.
- [9] EASA. Study on the societal acceptance of urban air mobility in europe. In *European Union Aviation Safety Agency*, 2021.
- [10] V. Mitchell, T. Ross, A. May, R. Sims, and C. Parker. Empirical investigation of the impact of using co-design methods when generating proposals for sustainable travel solutions. *CoDesign*, 12(4):205–220, 2016. doi:10.1080/15710882.2015.1091894.
- [11] P. Bardwell and S. Marr. Making the most of collaboration: An international survey of public service co-design, demos report 23. Technical report, DEMOS, Price Waterhouse Coopers (PWC) Public Sector Research Centre, London, 2008.

BIBLIOGRAPHY

- [12] R. Hansman and P. Vascik. Constraint identification in on-demand mobility for aviation through an exploratory case study of los angeles. May 2018.
- [13] A. Bauranov and J. Rakas. Designing airspace for urban air mobility: A review of concepts and approaches. *Progress in Aerospace Sciences*, 125:100726, 2021. URL: <https://www.sciencedirect.com/science/article/pii/S0376042121000312>, doi:<https://doi.org/10.1016/j.paerosci.2021.100726>.
- [14] O. Uzol, I. Yavrucuk, and N. S. Uzol. Panel-method-based path planning and collaborative target tracking for swarming micro air vehicles. *Journal of Aircraft*, 47(2):544–550, 2010.
- [15] U. NASA. Air traffic management for low-altitude drones, na a. *SA (NASA), Washington DC, USA*, 2015.
- [16] K. Balakrishnan, J. Polastre, J. Mooberry, R. Golding, and P. Sachs. Blueprint for the sky: The roadmap for the safe integration of autonomous aircraft. *Airbus UTM, San Francisco, CA*, 2018.
- [17] K. Low, L. Gan, and S. Mao. A preliminary study in managing safe and efficient low-altitude unmanned aircraft system operations in a densely built-up urban environment. *Air Traffic Management Research Institute, School of Mechanical and Aerospace Engineering Nanyang Technological University*, 2014.
- [18] K. H. Goodrich and C. R. Theodore. Description of the nasa urban air mobility maturity level (uml) scale. In *AIAA Scitech 2021 forum*, page 1627, 2021.
- [19] S. J. Undertaking. European atm master plan: Roadmap for the safe integration of drones into all classes of airspace. *SESAR Joint Undertaking, Publications office of the European Union, Luxembourg*, 2018.
- [20] S. J. Undertaking. U-space blueprint. *SESAR Joint Undertaking, Publications office of the European Union, Luxembourg*, JUN. 9, 2017.
- [21] S. Bradford. Urban air mobility (uam) concept of operations v1. 0. *FAA, Washington DC, USA*, 2020.
- [22] D.-S. Jang, C. A. Ippolito, S. Sankararaman, and V. Stepanyan. Concepts of airspace structures and system analysis for uas traffic flows for urban areas. In *AIAA Information Systems-AIAA Infotech@ Aerospace*, page 0449. 2017.
- [23] C. Le Tallec, P. Le Blaye, and M. Kasbari. Low level rpas traffic management (llrtm) concept of operation. In *17th AIAA Aviation Technology, Integration, and Operations Conference*, page 3938, 2017.
- [24] FAA. Urban air mobility (uam) concept of operations v1.0,. June 2020.
- [25] R. Hansman and L. Li. Experimental studies of cognitively based air traffic control complexity metrics for future operational concepts. 04 2010.

- [26] B. Hilburn. Cognitive complexity in air traffic control: a literature review. 01 2004.
- [27] A. Majumdar and J. Polak. Estimating capacity of europe’s airspace using a simulation model of air traffic controller workload. *Transportation Research Record*, 1744:30–43, 01 2001. doi:10.3141/1744-05.
- [28] E. Sunil, J. Hoekstra, J. Ellerbroek, F. Bussink, D. Nieuwenhuisen, A. Vidosavljevic, and S. Kern. Metropolis: Relating airspace structure and capacity for extreme traffic densities. In *Proceedings of the 11th USA/Europe Air Traffic Management Research and Development Seminar (ATM2015), Lisbon (Portugal), 23-26 June, 2015*. FAA/Eurocontrol, 2015.
- [29] AMU-LED. D2.1 operational safety analysis and concept. March 2021.
- [30] A. P. Air. Revising the airspace model for the safe integration of small unmanned aircraft systems. *Amazon Prime Air*, 2015.
- [31] D. Geister and B. Korn. Concept for urban airspace integration dlr u-space blueprint. *German Aerospace Center-Institut of Flight Guidance*, 2017.
- [32] B. Lascara, A. Lacher, M. DeGarmo, D. Maroney, R. Niles, and L. Vempati. Urban air mobility airspace integration concepts. *The Mitre Corporation*, 2019.
- [33] S. Ahamed, S. Prakash, J. Shuvrangshu, and G. Debasish. Uas flight path planning using numerical potential fields in dense non-segregated airspace. In *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1010–1019, 2021. doi:10.1109/ICUAS51884.2021.9476791.
- [34] N. Zhang, M. Zhang, and K. H. Low. 3d path planning and real-time collision resolution of multirotor drone operations in complex urban low-altitude airspace. *Transportation Research Part C: Emerging Technologies*, 129:103123, 2021. URL: <https://www.sciencedirect.com/science/article/pii/S0968090X2100142X>, doi:<https://doi.org/10.1016/j.trc.2021.103123>.
- [35] W. Dai, B. Pang, and K. H. Low. Conflict-free four-dimensional path planning for urban air mobility considering airspace occupancy. *Aerospace Science and Technology*, 119:107154, 2021. URL: <https://www.sciencedirect.com/science/article/pii/S1270963821006647>, doi:<https://doi.org/10.1016/j.ast.2021.107154>.
- [36] S. Huang and K. H. Low. A path planning algorithm for smooth trajectories of unmanned aerial vehicles via potential fields. In *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 1677–1684, 2018. doi:10.1109/ICARCV.2018.8581198.
- [37] D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *Robotics and Automation Magazine, IEEE*, 4:23 – 33, 04 1997. doi:10.1109/100.580977.

BIBLIOGRAPHY

- [38] P. Fiorini and Z. Shiller. Motion planning in dynamic environments using velocity obstacles. *The International Journal of Robotics Research*, 17(7):760–772, 1998. doi:10.1177/027836499801700706.
- [39] S. Petti and T. Fraichard. Safe motion planning in dynamic environments. pages 2210 – 2215, 09 2005. doi:10.1109/IR0S.2005.1545549.
- [40] N. Chan, J. J. Kuffner, and M. Zucker. Improved motion planning speed and safety using regions of inevitable collision. 2008. URL: <https://api.semanticscholar.org/CorpusID:16622850>.
- [41] A. Chakravarthy and D. Ghose. Obstacle avoidance in a dynamic environment: A collision cone approach. *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, 28(5):570–574, September 1998. doi:10.1109/3468.709600.
- [42] A. Chakravarthy and D. Ghose. Generalization of the collision cone approach for motion safety in 3-d environments. *Autonomous Robots*, 32(3):243–266, 2012. doi:10.1007/s10514-011-9270-z.
- [43] A. Chakravarthy and D. Ghose. Collision cones for quadric surfaces. *IEEE Transactions on Robotics*, 27(6):1166, December 2011. doi:10.1109/TR0.2011.2159413.
- [44] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel. Motion planning with sequential convex optimization and convex collision checking. *The International Journal of Robotics Research*, 33(9):1251–1270, 2014.
- [45] S. Liu, N. Atanasov, K. Mohta, and V. Kumar. Search-based motion planning for quadrotors using linear quadratic minimum time control. *CoRR*, abs/1709.05401, 2017. URL: <http://arxiv.org/abs/1709.05401>, arXiv:1709.05401.
- [46] J. Tordesillas and J. P. How. MADER: trajectory planner in multi-agent and dynamic environments. *CoRR*, abs/2010.11061, 2020. URL: <https://arxiv.org/abs/2010.11061>, arXiv:2010.11061.
- [47] J. Tordesillas, B. T. Lopez, M. Everett, and J. P. How. Faster: Fast and safe trajectory planner for navigation in unknown environments. *IEEE Transactions on Robotics*, 38(2):922–938, 2022. doi:10.1109/TR0.2021.3100142.
- [48] M. Mirzaei Teshnizi, A. Kosari, S. Goliaei, and S. Shakheshi. Centralized path planning for multi-aircraft in the presence of static and moving obstacles. *International Journal of Engineering*, 33(5):923–933, 2020. doi:10.5829/ije.2020.33.05b.25.
- [49] Y. Chen, G. Luo, Y. Mei, J. Yu, and X. Su. Uav path planning using artificial potential field method updated by optimal control theory. *International Journal of Systems Science*, 47:1407 – 1420, 2016. URL: <https://api.semanticscholar.org/CorpusID:29757767>.

- [50] D. Ge and U. Topcu. Hierarchical path planning for urban on-demand air mobility. In *2019 IEEE Conference on Control Technology and Applications (CCTA)*, pages 894–899, 2019. doi:10.1109/CCTA.2019.8920446.
- [51] P. Pradeep and P. Wei. Energy-efficient arrival with rta constraint for multirotor evtol in urban air mobility. *Journal of Aerospace Information Systems*, 16, 04 2019. doi:10.2514/1.1010710.
- [52] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, volume 2, pages 500–505, 1985. doi:10.1109/ROBOT.1985.1087247.
- [53] H. Li, W. Liu, C. Yang, W. Wang, T. Qie, and C. Xiang. An optimization-based path planning approach for autonomous vehicles using dynefwa-artificial potential field. *IEEE Transactions on Intelligent Vehicles*, pages 1–1, 2021. doi:10.1109/TIV.2021.3123341.
- [54] U. Orozco-Rosas, O. Montiel, and R. Sepúlveda. Mobile robot path planning using membrane evolutionary artificial potential field. *Applied Soft Computing Journal*, pages 236–251, April 2019. doi:10.1016/j.asoc.2019.01.036.
- [55] L. Lifen, S. Ruoxin, L. Shuandao, and W. Jiang. Path planning for uavs based on improved artificial potential field method through changing the repulsive potential function. In *2016 IEEE Chinese Guidance, Navigation and Control Conference (CGNCC)*, pages 2011–2015, 2016. doi:10.1109/CGNCC.2016.7829099.
- [56] Z. Yingkun. Flight path planning of agriculture uav based on improved artificial potential field method. In *2018 Chinese Control And Decision Conference (CCDC)*, pages 1526–1530, 2018. doi:10.1109/CCDC.2018.8407369.
- [57] Y. Liu and Y. Zhao. A virtual-waypoint based artificial potential field method for uav path planning. In *2016 IEEE Chinese Guidance, Navigation and Control Conference (CGNCC)*, pages 949–953, 2016. doi:10.1109/CGNCC.2016.7828913.
- [58] W. Honglun, L. Wentao, P. Yao, X. Liang, and L. Chang. Three-dimensional path planning for unmanned aerial vehicle based on interfered fluid dynamical system. *Chinese Journal of Aeronautics*, 28, 12 2014. doi:10.1016/j.cja.2014.12.031.
- [59] F. Bayat, S. Najafi-Nia, and M. Aliyari. Mobile robots path planning: Electrostatic potential field approach. *Expert Systems with Applications*, 100, 06 2018. doi:10.1016/j.eswa.2018.01.050.
- [60] M. Nazarahari, E. Khanmirza, and S. Doostie. Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm. *Expert Systems with Applications*, 115, 08 2018. doi:10.1016/j.eswa.2018.08.008.
- [61] N. Tsourveloudis, K. Valavanis, and T. Hebert. Autonomous vehicle navigation utilizing electrostatic potential fields and fuzzy logic. *Robotics and Automation, IEEE Transactions on*, 17:490 – 497, 09 2001. doi:10.1109/70.954761.

BIBLIOGRAPHY

- [62] T. Hickling, N. Aouf, and P. Spencer. Robust adversarial attacks detection based on explainable deep reinforcement learning for uav guidance and planning. *IEEE Transactions on Intelligent Vehicles*, 8(10):4381–4394, 2023. doi:10.1109/TIV.2023.3296227.
- [63] W. Li, C. Yang, Y. Jiang, X. Liu, and C.-Y. Su. Motion planning for omnidirectional wheeled mobile robot by potential field method. *Journal of Advanced Transportation*, 2017:1–11, 03 2017. doi:10.1155/2017/4961383.
- [64] S. M. Hosseini Rostami, A. Kumar, J. Wang, and X. Liu. Obstacle avoidance of mobile robots using modified artificial potential field algorithm. *EURASIP Journal on Wireless Communications and Networking*, 2019, 03 2019. doi:10.1186/s13638-019-1396-2.
- [65] C. R. Chester. Techniques in partial differential equations. *McGraw-Hill*, pages 72–73, 1970.
- [66] J.-O. Kim and P. Khosla. Real-time obstacle avoidance using harmonic potential functions. *IEEE Transactions on Robotics and Automation*, 8(3):338–349, 1992. doi:10.1109/70.143352.
- [67] S. Garrido and L. Moreno. Robotic navigation using harmonic functions and finite elements. pages 94–103, 01 2006.
- [68] A. A. Masoud. A harmonic potential field approach for joint planning and control of a rigid, separable nonholonomic, mobile robot. *Robotics and Autonomous Systems*, 61(6):593–615, 2013. URL: <https://www.sciencedirect.com/science/article/pii/S0921889013000432>, doi:<https://doi.org/10.1016/j.robot.2013.02.007>.
- [69] F. Fahimi. Autonomous robots. *Springer*, pages 81–127, 2010.
- [70] R. Daily and D. M. Bevely. Harmonic potential field path planning for high speed vehicles. In *2008 American Control Conference*, pages 4609–4614, 2008. doi:10.1109/ACC.2008.4587222.
- [71] L. C. A. Pimenta, G. A. S. Pereira, N. Michael, R. C. Mesquita, M. M. Bosque, L. Chaimowicz, and V. Kumar. Swarm coordination based on smoothed particle hydrodynamics technique. *IEEE Transactions on Robotics*, 29(2):383, April 2013. doi:10.1109/TR0.2013.2243998.
- [72] Z. Song, D. Lipinski, and K. Mohseni. Multi-vehicle cooperation and nearly fuel-optimal flock guidance in strong background flows. *Ocean Engineering*, 141:388–404, September 2017. doi:10.1016/j.oceaneng.2017.05.024.
- [73] R. I. Lewis. *The surface vorticity method for inviscid ideal fluid flow*, pages 3–98. Cambridge Engine Technology Series. Cambridge University Press, 1991.

- [74] J. Katz and A. Plotkin. *Singularity Elements and Influence Coefficients*, page 230–261. Cambridge Aerospace Series. Cambridge University Press, 2 edition, 2001. doi:10.1017/CB09780511810329.012.
- [75] X. Liang, W. Honglun, D. Li, and C. Liu. Three-dimensional path planning for unmanned aerial vehicles based on fluid flow. pages 1–13, 03 2014. doi:10.1109/AERO.2014.6836520.
- [76] J. D. Anderson. Fundamentals of aerodynamics. *McGraw-Hill*, pages 319–325, 2011.
- [77] Y. Zhang and K. P. Valavanis. A 3-d potential panel method for robot motion planning. *Robotica*, 15:421 – 434, 1997. doi:10.1017/S0263574797000520.
- [78] J. Velagić, L. Vuković, and B. Ibrahimović. Mobile robot motion framework based on enhanced robust panel method. *International Journal of Control, Automation and Systems*, 18(5):1264–1276, 2020. doi:https://doi.org/10.1007/s12555-019-0009-5.
- [79] G. C. S. Cruz and P. M. M. Encarnação. Obstacle avoidance for unmanned aerial vehicles. *Journal of Intelligent & Robotic Systems*, 65:203–217, 2012. doi:10.1007/s10846-011-9587-z.
- [80] S. Axler, P. Bourdon, and R. Wade. *Harmonic Function Theory*. Graduate Texts in Mathematics. Springer New York, 2013. URL: <https://books.google.de/books?id=CYbdBwAAQBAJ>.
- [81] J. C. Maxwell. On physical lines of force. *Philosophical Magazine Series 4*, 21(139):161–175, 1861. doi:10.1080/14786446108643680.
- [82] Z. Bilgin, M. Bronz, and I. Yavrucuk. Experimental evaluation of panel-method-based path planning for evtol in a scaled urban environment. In *Vertical Flight Society Forum 78*, May 10-12 2022.
- [83] E. Kreyszig. *Differential Geometry*. Dover, New York, 1991.
- [84] J. Panerati, H. Zheng, S. Zhou, J. Xu, A. Prorok, and A. Schoellig. Learning to fly—a gym environment with pybullet physics for reinforcement learning of multi-agent quadcopter control, 03 2021.
- [85] ENAC. Drone flight arena toulouse occitanie. Ecole Nationale de l’Aviation Civile, Retrieved 19 October 2021, from <https://www.enac.fr/en/drone-flight-arena-toulouse-occitanie-0>, 2021.
- [86] G. Hattenberger, M. Bronz, and M. Gorraz. Using the paparazzi uav system for scientific research. 08 2014. doi:10.4233/uuid:b38fbd7-e6bd-440d-93be-f7dd1457be60.

BIBLIOGRAPHY

- [87] E. Smeur, G. de Croon, and Q. Chu. Cascaded incremental nonlinear dynamic inversion for mav disturbance rejection. *Control Engineering Practice*, 73:79–90, 2018. URL: <https://www.sciencedirect.com/science/article/pii/S0967066118300030>, doi:<https://doi.org/10.1016/j.conengprac.2018.01.003>.
- [88] Z. Bilgin. Experimental evaluation of potential flow based target tracking for multiple micro aerial vehicles. <https://youtu.be/xcEKARPA9xM>, May 2022. Accessed: 2024-09-17.
- [89] H. G. De Marina, Y. A. Kapitanyuk, M. Bronz, G. Hattenberger, and M. Cao. Guidance algorithm for smooth trajectory tracking of a fixed wing uav flying in wind flows. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 5740–5745. IEEE, 2017.
- [90] ASSURED-UAM. Financing and public acceptance. H2020-EU.3.4. - SOCIETAL CHALLENGES - Smart, Green And Integrated Transport, DOI: 10.3030/101006696, 2021.
- [91] CAMI. Components of public acceptance for aam and uam. In *The Community Air Mobility Initiative*, 2020.
- [92] P. Boucher. ‘you wouldn’t have your granny using them’: Drawing boundaries between acceptable and unacceptable applications of civil drones. *Sci Eng Ethics*, 22, 2016.
- [93] R. L. Finn, D. Wright, L. Jacques, and P. D. Hert. Study on privacy, data protection and ethical risks in civil remotely piloted aircraft systems operations. In *European Commission*, 2014.
- [94] H. Eissfeldt, V. Vogelpohl, M. Stolz, A. Papenfuss, M. Biella, J. Belz, and D. Kügler. The acceptance of civil drones in germany. *CEAS Aeronautical Journal*, 2020.
- [95] TindAIR. Three, two, one... the u-space very large-scale demonstrations started for tindair. Retrieved 15 July, 2023, from. <https://tindair.eu/?p=1299>, 2022.
- [96] RolandBerger. How air taxis and freight drones are revolutionizing the future of transportation. Retrieved 12 November, 2020, from. <https://www.rolandberger.com/en/Insights/Global-Topics/Urban-Air-Mobility/>, 2022.