Real-Time UAV Teleoperation and Collision Avoidance using Potential Fields in Underground Mines

Dane M. Slattery

A dissertation submitted to the Faculty of Engineering and the Built Environment, University of the Witwatersrand, Johannesburg, in fulfilment of the requirements for the degree of Master of Science in Engineering.

Johannesburg, June 2020

Declaration

I declare that this dissertation is my own, unaided work, except where otherwise acknowledged. It is being submitted for the degree of Master of Science in Engineering to the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination to any other university.

Signed this 19th day of June 2021

Dane Slattery

Dane M. Slattery

Abstract

An Unmanned Aerial Vehicle is equipped for remote operation in an underground mining environment. Strict spacial constraints require active collision avoidance to preserve the Unmanned Aerial Vehicle. Previous literature identifies the Artificial Potential Field algorithm as a candidate for local collision avoidance. The artificial field models obstacles as a repulsive force acting on the vehicle, while the pilot provides an attractive force. Thus, the net effect of these artificial forces is to avoid obstacles and follow pilot commands. However, computing the requisite gains that will prevent collisions while providing pilot authority is previously performed through trial and error. This study aims to develop a tuning methodology to determine these gains given some knowledge about the Unmanned Aerial Vehicle, environment and sensing stack.

To test the methodology, as well as perform a physical validation of the previously-simulated success of the Artificial Potential Field algorithm, a Unmanned Aerial Vehicle is equipped with LiDAR and onboard computing. The Unmanned Aerial Vehicle is then flown amongst simple obstacle configurations to confirm the interactions between the Artificial Potential Field algorithm, the tuning methodology and the obstacle. The Unmanned Aerial Vehicle avoided a 0.4 m wide box in the direct piloted path with a minimum proximity of 0.77 m, and safely passes through a 2.2 m wide gap in a wall of obstacles with a minimum proximity of 1.1 m. The Unmanned Aerial Vehicle is also flown for 10 m in a mock mine tunnel 3 m wide without collision, but with oscillatory motion. By reducing the gains through the tuning methodology, the oscillations are largely damped.

These results suggest that the Artificial Potential Field algorithm is a suitable candidate for collision avoidance, confirming the simulated results. The tuning methodology is also verified on a physical platform, with the caveat being the inverse relationship between collision risk and pilot authority.

To my mother and father who have always fostered and supported my curiosity.

Acknowledgements

I would like to thank Prof. Anton van Wyk and Dr Steven Dinger who constantly pushed me and my ideas to their limits in the pursuit of good science and engineering. Besides their work as supervisors, I also appreciate the many hours of meandering conversation regarding other facets of life.

Without the staff and postgraduate body of the WITS Electrical and Information Engineering faculty my sanity would have long since flown. It was a privilege to be among so many great minds and helping hands.

My thanks are extended to the Sibanye Stillwater Digital Mining Laboratory. The multidisciplinary research group never shied away from my deeper questions into the world of mining engineering. Their funding programme and focused teams allow me to contribute this research to real-world problems.

To my friend and colleague Rowan Reeve who happily provided a brain to pick throughout the research.

Finally, a deep thanks are extended to Jeremy Green, whose support, friendship and ideas were the foundation of this research.

Contents

List of Figures

List of Tables

List of Symbols

The principal symbols used in this thesis are summarised below, and the first equation in which each symbol appears is given. The units, if any, are shown in square brackets.

k	Positive attractive gain,	*Equation (3.1)*
q	Current robot position [m],	*Equation (3.1)*
q_g	Goal robot position [m],	*Equation (3.1)*
$\Vert \cdot \Vert$	The l^2-norm of a vector, denoting Euclidean distance [m],	*Equation (3.3)*
η	Positive repulsive gain,	*Equation (3.3)*
ρ_0	Repulsive distance threshold [m],	*Equation (3.3)*
$\vec{\rho}$	Distance vector between the robot and the obstacle surface [m],	*Equation (3.3)*
n_0	Number of obstacles,	*Equation (3.5)*
$\vec{\rho}_i$	Distance vector between the robot and the i_{th} obstacle [m],	*Equation (3.5)*
$\vec{v_i^t}$	Relative velocity of the i_{th} obstacle [m/s],	*Equation (3.6)*
λ_i	The velocity scaling factor for the i_{th} sensor,	*Equation (3.7)*
v_{max}	The maximum translational velocity of the robot [m/s],	*Equation (3.7)*
T_{sensor}	Period between rangefinder samples [s],	*Equation (4.1)*

$r_{velocity}$	A safety buffer to account for motion between rangefinder samples [m], *Equation (4.1)*
$r_{vehicle}$	Distance from the center of the platform to the furthest tip of a blade [m], *Equation (4.2)*
r_{wall}	A safety buffer to account for the wall effect [m], *Equation (4.2)*
θ_{res}	Rangefinder angular resolution, *Equation (4.3)*
$\vec{u}$	A control vector for teleoperation in the range $[-1 \rightarrow 1]$, *Equation (4.5)*

Nomenclature

2D	Two-dimensional
3D	Three-dimensional
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
MAV	Micro Aerial Vehicle
GPS	Global Positioning System
GNSS	Global Navigation Satellite System
GLONASS	GLObal NAvigation Satellite System
GNRON	Goal Non-Reachable with Obstacle Nearby
APF	Artificial Potential Field
AFF	Artificial Force Field
VFF	Virtual Force Field
VFH	Virtual Force Histogram
GPF	Generalised Potential Field
SLAM	Simultaneous Localisation and Mapping
ROS	Robot Operating System
HITL	Human-In-The-Loop
GCS	Ground Control Station
EKF	Extended Kalman Filter
IMU	Inertial Measurement Unit

FC Flight Controller

RPAS Remote Piloted Aerial System

MAE Mean Absolute Error

COG Centre Of Gravity

6DOF Six Degrees of Freedom

MPP Minimum Potential Path

LiDAR Light Detection And Ranging

Chapter 1

Introduction

The broader context of this research is presented in this chapter, leading into a motivation for aerial vehicles in underground mines. With use cases established, the teleoperation and collision avoidance requirements are introduced. The specific research objectives are then presented, narrowing the scope to focus on the artificial potential field approach for underground mines.

The primary objectives of the research, discussed in this chapter, are a validation of collision avoidance on a real platform in a mine-like environment. The challenges facing a collision avoidance system on an underground aerial vehicle are: a lack of global position information or a consistent map, zero-collision tolerance and limited computational and sensing ability. By applying the proposed parametrisation the Artificial Potential Field (APF) algorithm is shown to succesfully avoid collisions using only onboard computation and sensing.

The research shows that a Unmanned Aerial Vehicle (UAV) with a tip-to-tip dimension of 450 mm can be safely teleoperated over a WiFi network through a gap of only 2.2 m, acceptably avoid obstacles 40 cm as small as wide in the direct flight path, and operate collision free in a 3 m wide mine-like tunnel.

This chapter also provides a brief structure of the dissertation as a guide for the reader.

1.1 Context

Mining is a fundamental industry to support human development in the modern age, but despite vast improvements in risk management and safety procedures, underground mining still faces a number of fatalities each year. It is an inherently dangerous environment; with the Department of Mineral Resources (DMR) of South Africa reporting 81 fatalities for the year of 2018, and 4483 occupational-disease related cases [1].

With a goal of zero-harm, there is a need to remove the miner from dangerous mine conditions, and an opportunity for technology to do so. Some key examples of this in underground mining are: the ability to detect and measure harmful gas levels; mechanised mining, teleoperating machines in dangerous areas; digital shift management and access control systems. The Sibanye-Stillwater Digital Mining Laboratory (DigiMine) is a post-graduate research group focusing on these and many other technologies within the South African context, and are also the sponsors of this research.

Corke *et. al* identifies the importance of technology in the future of mines (not only for safety, but for productivity and profitability), and provide a comprehensive study of mining technology as at 2008 [2]. A decade later, and the predictions of the paper have held true, especially for robotics, which has seen greater adoption thanks to improvements in computation, sensing, actuation and communication technologies.

Robots are well suited to the mines due to their robustness against temperature, air pressure and fatigue. Their primary strength is that they can be controlled remotely (or act autonomously), keeping miners out of harms way as well as allowing exploration into highly constrained areas. Despite their suitability, many engineering challenges face underground robotics, namely communication (especially in rock-fall situations), localisation and mapping in unknown areas, manoeuvring rugged terrain and avoiding collisions arising from the highly constrained environment.

This research focuses on teleoperating an aerial vehicle in an underground mining environment. Aerial vehicles are suitable for flying in vertical spaces (where a ground robot would be unsuited) but are sensitive to collisions which would likely destroy the vehicle. This necessitates that collision avoidance to be built in to the Unmanned Aerial Vehicle (UAV) system.

The research forms a part of the greater UAV Project at the DigiMine Research Group within the University of the Witwatersrand, which aims to utilise UAV technology in mines for mapping, search-and-rescue, infrastructure verification and gas detection to name a few. The project is mandated by the Mine Health and Safety Council.

1.2 Research Objectives

The primary aim of this research is to validate the APF concept on a physical platform. While much is understood about APFs behaviour from simulation, there are many unknowns in implementation, especially in unknown, constrained environments such as mining tunnels. In particular, the objectives of this research are:

1. **To provide and validate (through experiments) a methodology for choosing potential field parameters.**
2. **To determine how the level of constraint in an environment, in particular the tunnel width, affects the performance of the APF. This will help determine the APF algorithm's suitability for the constrained mining environment.**

This research is a step towards realising a fully autonomous UAVs to improve mine safety and enable remote sensing. Without collision avoidance, an UAV underground would experience unrecoverable damage, even in the hands of an experienced pilot.

The delimitations of this study are such that true autonomy is not achieved, since a human is assumed to always be in the loop. Also not explored in this research is the behaviour with signal loss, vehicle failure or obstacle identification. The intrinsic safety of UAVs for mining environments, as well as the impact of other factors in a mine such as dust, moisture and temperature, are not considered. Additionally, the research only considers collisions in the horizontal plane, restricting the problem to a 2D one.

As a side note, there are many terms that could be used to describe a UAV, such as Remote Piloted Aerial System (RPAS), drone, quadcopter, Micro Aerial Vehicle (MAV), and all are applicable in this context[3, 4, 5]. However, for clarity, the term UAV will be used throughout the research, and refers to a remotely controlled aerial vehicle that obtains thrust from a combination of electric motors and propellers. The UAVs under this study typically weigh less than 3 kg, and have a tip-to-tip dimension of less than 1 m.

1.3 Research Contributions

The major contributions of this research is the experimental validation of the APF algorithm for underground aerial robotics as a teleoperation tool. The outcomes also include a novel method for parameter selection of APF algorithms. Additionally, this research addresses the many challenges of a real flying vehicle, discussing options for the selection of hardware components that realise the APF algorithm. The APF algorithm is modified to account for the lack of UAV position and velocity state information.

It is shown that the APF is appropriate and is able to fly collision free through a gap of 2.2 m, and in a tunnel of width 3 m on a platform 0.45 m wide. The performance is studied in terms of vehicle safety (proximity), pilot compliance and oscillation in later chapters. The platform is shown in Figure 1.1.

Additionally, the proposed methodology is widely tested in a simple simulated environment to observe the effects of sensor scan rate, angular resolution, obstacle width and the velocity compensation method. It is found that higher scan rates result in better performance for the APF in all regards, while the parametrisation method is able to sufficiently compensate for the decreased sample rate down to, but not including, 1 Hz. The angular resolution of the sensor was shown to have little effect while being sufficiently compensated for in the tuning methodology. However, it is ill-advised to choose a sensor angular resolution that results in an incomplete detection zone around the UAV. The obstacle width parameter is shown to directly influence the collision avoidance sensitivity, with smaller obstacles producing more sensitive (and therefore safer) fields while reducing pilot authority. Finally, the linear velocity compensation proposed by Krogh [6] is found to perform best, but does not provide sufficient damping to completely eliminate oscillations.

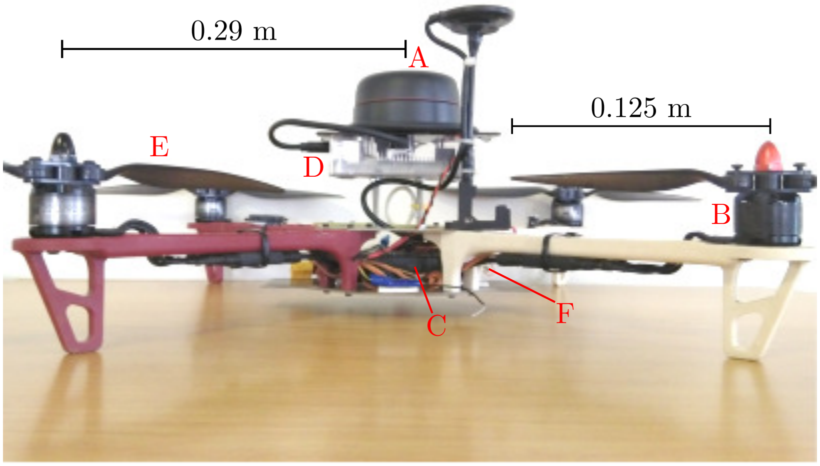


Figure 1.1: The UAV hardware stack with (A) RPLiDAR A2M8, (B) motors, (C) Flight Controller (FC), (D) Raspberry Pi 3B+, (E) 25 cm propellers and (F) downwards facing rangefinder. The image also shows dimensions of the center-to-tip dimension (29 cm) and propellers(12.5 cm) used to calculate the safety buffer size.

1.4 Layout

The remainder of the document is structured as follows:

- Chapter 2 provides a background on robotics with a focus on mining robotics, leading into a discussion on the use of UAVs in mines. The working principle of quadcopters, a specific class of UAV with four motors, is then provided in order to familiarise the reader. The chapter concludes with a description of collision avoidance in terms of algorithms and the hardware required to realise them.

- Chapter 3 discusses the history APFs, introducing the theory of operation as well as the primary drawbacks of the approach. This is followed by an in depth review of state-of-the-art literature on APF technology and it's use in collision avoidance and UAVs.

- Chapter 4 then follows on from Chapter 3, identifying the specific gap in literature that is addressed by this research. Here the specific research questions, as well as the approach used to answer them, are given. Chapter 4 then describes the parametrisation method used to tune the APF algorithm. Chapter 4 also discusses key modifications to the APF algorithm to allow underground use.

- Chapter 5 discusses the simulation environment in which a sensitivity analysis is performed. Here, key APF parameters are tested at their limits in order to assess their impact on performance. This set of experiments also serves as a first-pass test of the parametrisation method, providing insight into the real-world performance of the APF algorithm.

- Chapter 6 regards the hardware and software design that is used for validating experimental results. It describes the configuration of the LiDAR scanner used for sensing obstacles around the vehicle, as well as the Robot Operating System (ROS) environment and custom software written to enable collision-avoidance, teleoperation, data logging and representation.

- Chapter 7 presents the results of experiments, and discusses the outcomes of each experiment from the simulation and the physical testing. It presents the conclusions for the experimental validation performed on a physical platform.

- Chapter 8 is then the conclusion of the document as a whole, presenting briefly the results of the research. It also provides a review of the methodology and lists future prospects for the research, as well as for the DigiMine UAV project as a whole.

Chapter 2

Background

Narrowing the research context further from Chapter 1, two sources are presented that discuss the requirements and limitations of an underground UAV. Green [7] presents a semi-autonomous UAV as the result of a full requirement elicitation process that included industry and leading research. Within the DigiMine research group, Edwards [3] presents a vision-based mapping system used in the WITS Mock Mine. The vision-based approach unfortunately requires a lot of processing power, and is performed offboard which introduces delay and range limitations.
The chapter then presents the basic working principle of the quadcopter class of UAV. The minimum instrumentation required for a pilot friendly UAV is detailed, with a discussion on levels of autonomy and abstraction. An overview of obstacle avoidance techniques and sensor technology is then detailed.

2.1 Introduction

The local context of this research is given in two theses which provide the formative ideas and discuss the broader problems facing automated UAVs in mining. These ideas are summarised below.

Green [8] describes a requirement elicitation technique in their thesis, and applies it to an underground UAV to scan and inspect ore passes. They note that traditional ground-based mining robotics do not have the vertical reach or agility to manoeuvre these mining environments. Consider for example the CAESAR robot in Figure 2.1, which cannot climb a slope with inclination more than 50° [9].

Figure 2.1: The CAESAR tracked-robot, which provides thermal imaging and CO, CO_2, H_2S, methane and oxygen sensors for disaster scenarios, amongst its many other sensory and communication technologies.

Green [8] identifies the great risk to human life inside a mine as a major driving factor for mine, health and safety research and development. A number of requirements were identified for a robot that can either perform search-and-rescue or rock-pass scan operations, some of which are listed below:

- Quick deployment.
- Man-packable (able to be carried by a man).
- Intrinsically safe/Flame-Proof.
- Semi-autonomous, such that there will always be a human in the loop, but collisions should be avoided.
- Low cost.
- The ability to map and explore a tunnel or rock-pass.

With these in mind, Green [8] proposes a technology demonstrator with a Platform Preservation system, consisting of 10 ultrasonic sensors in an array to detect and avoid walls in the environment. This would be mounted to a UAV, along with a depth sensor for mapping and visualisation. A key requirement for the UAV system is an operator interface, which should provide live sensory information, and allow for the user to easily input a new trajectory.

Through testing and the requirement elicitation process, Green [8] notes a number of key factors to be considered in an underground mining environment, many of which were also observed in [10, 11]. These factors are:

- Ventilation air-flow causes drift.
- Dust ingress.
- WiFi/Control signal integrity.
- Limited Flight time.

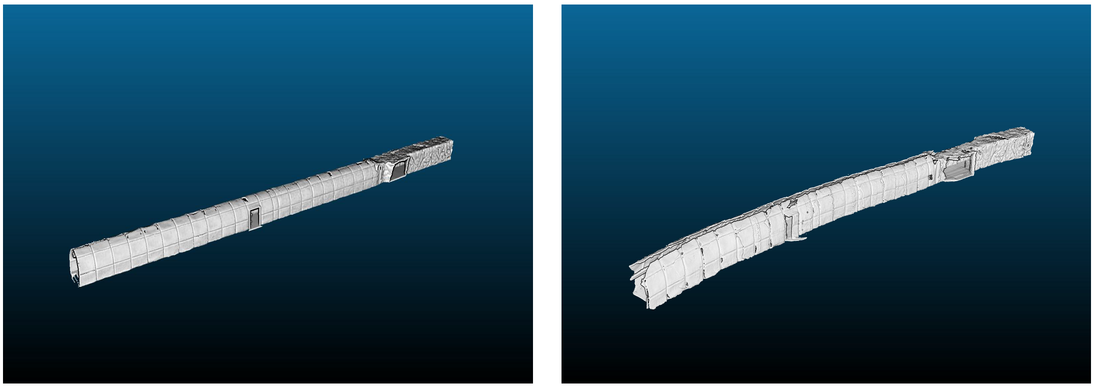
There are many more applications for UAVs inside a mine other than rock-pass inspection, such as mapping and ventilation monitoring but the requirements and limitations listed above are ubiquitous to the mining environment in general.

The thesis of Edwards presents a vision-based quadcopter mapping system to explore and map mine tunnels at a low cost [3]. The research is the first output of the DigiMine UAV project, and provides many lessons to be learned, primarily the existence of noise in the positioning system (which was based entirely on the front-facing camera), and the interference between communication systems, which both operate at 2.4GHz.

The main problem with this interference is that the offboard controller for this quadcopter had to be implemented on an external laptop, which required the vehicle to remain tethered, introducing delay and limiting the operating range. Edwards identifies a number of remedial actions:

- The development of a custom quadcopter to better withstand the dusty conditions and payload requirements.
- Moving low-level control on-board (online processing).
- Improving the communication protocols to prevent interference.

Edwards shows that it is possible to obtain a detailed and accurate map of a mining tunnel using off-the-shelf depth sensors, which fulfils one of many requirements detailed by Green in [8]. Figures 2.2b and 2.2a show how the off-the-shelf sensor point cloud compares to a survey-grade point cloud. However, the choice of a depth-camera increases the payload requirements of the UAV.



(a) (b)

Figure 2.2: Point clouds of the WITS Mock Mine tunnel, (a) using a survey-quality sensor and (b) using a RGBD Scanner.

Figure 2.2a shows how accurate meshing and point cloud alignment is achieved by a survey-quality Light Detection And Ranging (LiDAR) with markers providing exact position information. Figure 2.2a shows that some scale and direction information is lost using an RGBD scanner on a UAV because of positioning and tracking error [3].

The above projects are particularly relevant to South Africa, but UAVs in mining are certainly a global trend. Above ground, companies such as senseFly, microdrones, airobotics, DroneDeploy and RocketMine offer aerial inventory tracking, surveying, traffic, blast, and water management for open-pit mines. Underground, however, new applications are revealing themselves, spawning companies such as TerraDrone, Emesent, ClickMox and Safesite Exploration. The existence of a market for underground UAVs is an additional motivator for the furthering of the field.

2.2 Quadcopters

In this section the basic operational principles of a quadcopter are defined for some insight into how the low-level controllers function on a flight controller, as well as some factors influenced by a mining environment.

We first define a UAV as any aircraft that is tele-operated (as opposed to traditional aircraft which have the pilot on-board). Quadcopters form a class of UAV that generate their lift purely from their 4 fixed-pitch rotors mounted on the ends of symmetrical arms [12]. They have seen vast adoption in the industrial, research, commercial and hobbyist sectors due to their growing ease-of-use, lower operational expenses and extensive configuration options when compared to other aerial vehicles. Quadcopter dynamics are elegant and well understood, but the same principles for modelling and control can be extended to other UAVs.

Before the basic operational principles of quadcopters are discussed, it is necessary to define a frame of reference.

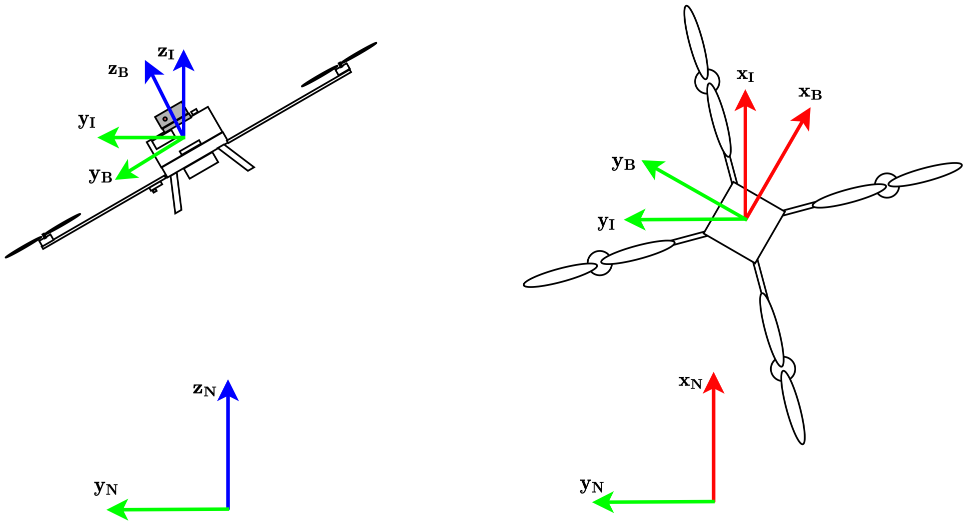
2.2.1 Frames of Reference

A set of right-handed co-ordinate systems is defined:

- Let the map, $\mathbf{N} = \{\mathbf{x_N}, \mathbf{y_N}, \mathbf{z_N}\}$ be the world[1] fixed-frame with the $\mathbf{x}$-axis pointing forward, $\mathbf{y}$-axis pointing left and $\mathbf{z}$-axis pointing up. Rotation about the $\mathbf{z_N}$-axis (yaw) is represented by Ψ, rotation about the $\mathbf{y_N}$-axis (pitch) is represented by θ and rotation about the $\mathbf{x_N}$-axis (roll) is represented by ϕ as depicted in Figure 2.3[2].
- An inertial frame, $\mathbf{I} = \{\mathbf{x_I}, \mathbf{y_I}, \mathbf{z_I}\}$ is defined with it's origin at the Centre Of Gravity (COG) of the quadcopter and with its axis aligned with $\mathbf{N}$.
- The body frame, $\mathbf{B} = \{\mathbf{x_B}, \mathbf{y_B}, \mathbf{z_B}\}$ also has it's origin at the COG, but the axis is *always* aligned along the quadcopters symmetry such that $\mathbf{x_B}$ is always pointing in the forward direction of the quadcopter, $\mathbf{y_B}$ is always pointing left, and $\mathbf{z_B}$ is always pointing up from the centroid of the frame. Figure 2.3 shows how $\mathbf{B}$ (in red) and $\mathbf{I}$ are linked to the quadcopter frame.

[1]Naturally, the world is considered to be flat and infinite.

[2]All rotations are described in terms of roll, pitch and yaw, and should be referred back to this definition.



(a) (b)

Figure 2.3: A side view (a) and top view (b) of the frames of reference **B** and **I** attached to the quadcopter with roll (rotation about the x-axis) (a) and pitch (rotation about the y-axis) (b). The map frame (**N**) origin is located where the UAV started it's motion.

Figure 2.3 shows how each reference frame for the body (**B**), inertial (**I**) and map (**N**) frames are situated relative to each other. The body frame can be seen to rotate about the inertial frame as the UAV rolls, pitches, and yaws. Both the body and inertial frame translate together relative to the map frame of reference in **x**, **y**, and **z** directions.

2.2.2 Working Principle

There are two primary configurations of a quadcopters frame orientation for the forward direction as shown in Figure 2.4. Both the 'X' and '+' orientations provide symmetry along the forward axis, which is important for the stability of the vehicle. The analysis for an 'X' configuration is given, but the operational principles of the two are similar.

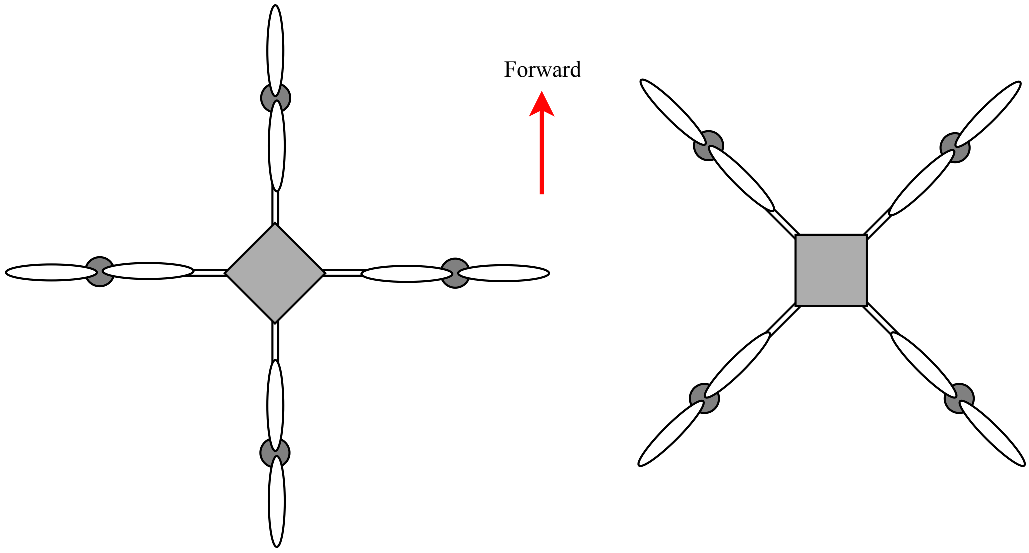


Figure 2.4: The '+' and 'X' frame orientations of a quadcopter, where the red arrow indicates the forward direction for each orientation.

By turning electric rotor s with angular velocity ω_s, aerodynamic lift force from the propeller generates a thrust, T_s, applied perpendicular to the rotor.

Then, varying ω_{1-4} varies the thrust of that rotor, generating a torque about the COG of the vehicle, which is the primary actuation for the system. This induces a rolling, pitching, yawing or thrusting action which enables the quadcopter to move with Six Degrees of Freedom (6DOF) in the body frame.

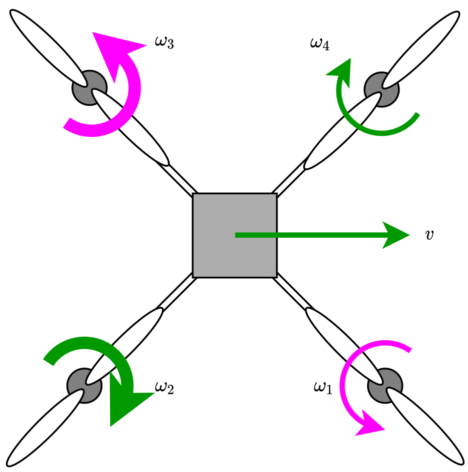


Figure 2.5: A mechanical model of a quadcopter frame showing the angular velocity of the rotors and net translational vector of the body.

As an example, if we reduce T_1 and T_2 relative to T_3 and T_4, the quadcopter will pitch forward and the total thrust-vector of the quadcopter will now have some component in the x-direction, resulting in forward translation. This is illustrated in Figure 2.5, where the thicker lines on the left-most motors' angular velocity arrows indicate an increased motor speed. Faster motors result in more thrust and therefore a net torque, which causes the UAV to pitch. This finally results in a translation as the total UAV thrust vector now contains a horizontal component, indicated by the vector v.

With only 4 inputs and 6DOF quadcopters are under-actuated, inherently unstable and require dynamical modelling and control to produce stability. While the system model and control is beyond the scope of this research, it is important to understand what is meant by stability. We define the stable zero-equilibrium state as one where the quadcopter is level i.e roll and pitch are zero, and yaw-rate is zero. Stability outside of the zero-equilibrium state should then be such that the operator can command attitude setpoints for roll, pitch and yaw-rate in order to translate and rotate the quadcopter.

In order to fully realise a stability controller, an Inertial Measurement Unit (IMU) is typically placed on the quadcopter to measure the attitude angles and rates as well as a barometer for the altitude. A modern FC performs computations at high rates to control the system dynamics. These devices implement what is known as attitude ($\dot{\eta}$) estimation and control, and a common approach uses two nested PID controllers for roll, roll-rate, pitch, pitch-rate, yaw-rate and thrust control. This level of attitude stabilisation is what [13] describes as a pilot-friendly quadcopter, without which it would be impossible for a human to pilot.

GPS-enabled quadcopters then use some separate position control to perform position, velocity and trajectory tracking. In GPS-denied environments it is expected that a quadcopter will drift due to external factors in the environment, sensor error, construction tolerances and non-linearities in the model. The pilot will need to compensate for this drift to avoid collisions, but the lower-level control at least abstracts the unstable vehicle dynamics.

With this abstraction in place (in the form of mature, well-tested flight controller hardware and software, which exists in many forms), it is possible to consider further levels of autonomy as described in Table 2.1.

Table 2.1: Levels of Autonomy. As the level increases, the required pilot involvement decreases, but the required on-board and logical control level increases.

Level	**Type**	**Description**
1	Attitude	The quadcopter autopilot maintains a stable attitude. The pilot controls all other functionality.
2	Pilot Assistance	The quadcopter is able to sense obstacles and alert the pilot.
3	Collision Avoidance	The quadcopter actively avoids collisions, overriding pilot commands.
4	Path Following	The pilot creates a pre-determined path for the quadcopter, which it is able to execute.
5	Path Planning	The system can plan and follow it's own paths, providing self-correction to it's internal path planner.

Table 2.1 shows that as Levels of Autonomy for any system increase, human-involvement decreases, while actuation and decision making load on some intelligent system increases. In the case of the UAV, it is useful to understand the targeted level when dealing with literature, project planning and outcomes.

This research targets a level 3 autonomous system, in which the quadcopter will avoid collisions in lieu of pilot error, under the assumption that level-1 and 2 autonomy are in place. For a mine environment this is absolutely critical, but also provides a stepping stone towards more advanced levels of autonomy. This cascaded-control approach is typical of many control systems and provides a holistic view of the direction for increasingly intelligent machines.

By understanding the working principle of a quadcopter and relying on an attitude controller for roll, pitch and yaw, we abstract the complex rotational dynamics of the vehicle to a translational problem only.

This is a useful abstraction because we can apply Cartesian thinking to the collision avoidance problem, but it is important to be aware of the abstraction and its limitations:

- Translation in x- and y-axes will result in some roll and pitch angle respectively. Any relative range-readings will be affected by this angle according to trigonometry.
- The vehicle will have both angular and translational inertia.
- The vehicle will only respond to actuation as fast as the underlying loop rate allows.
- The translational dynamics are limited by the attitude controllers tuning parameters (e.g. a sluggish roll PID tuning will result in sluggish translation).

The next section will discuss some other factors to consider for a quadcopter in a mining environment.

2.2.3 Mining

Underground mining environments present unique opportunities for a UAV. Compared to crawling-robots, UAVs are more manoeuvrable as they are not limited by terrain, and can move in 6DOF. Because UAVs can fly, places that are inaccessible to other ground-based robots may be reachable by UAVs. Like ground-based vehicles, UAVs can also carry a payload, such as a LiDAR scanner or gas-detection instrumentation.

UAVs are also a potential tool in a shift towards digitisation of mines, allowing remote sensing throughout a large area.

There are some limitations for UAVs. Mining underground releases pockets of gases, and in some cases these are explosive. This necessitates policies on possible ignition sources, such as electronics. This is known as intrinsic safety, which limits energy available to an electronic device and forces designs to consider all possible sources of ignition such as electrical arcing and component temperatures.

Designing an intrinsically safe UAV is not considered as part of this research output, since certification around intrinsic safety is lengthy and depends on a static design[14]. A research prototype will be dynamic as design decisions change and new sensors are added and removed.

Underground mines are dusty and wet, which could certainly pose a threat to the quadcopter control circuitry and any moving parts. For this reason, an underground UAV will need to consider Ingress Protection for sensitive components, but for the same reason above, is not a focus of this research.

Vehicles underground operate in highly-constrained environments, increasing risk to personnel, the vehicles themselves and infrastructure. UAVs are smaller than human-operated mechanised mining equipment such as bolters, but flying in tight spaces such as stopes and tunnels mean they are prone to collisions (A typical tunnel will be between 2 m and 5 m).

To avoid collisions, a skilled pilot would need to navigate tunnels while also avoiding walls and other hanging objects. In low-light conditions, or in the worst case, around corners, this is impossible and some level of autonomy is required on-board the UAV. Fortunately, the constrained environment underground allows the use of short range (<30m) sensors to detect obstacles and localise the UAV in the space, which fits the description for, and necessitates at least level 3 autonomy.

2.3 Obstacle Avoidance Overview

Developing fully autonomous vehicles for GPS-denied environments is a popular topic for research and roboticists. There are many approaches for ground, aerial and aquatic vehicles targeting different levels of autonomy. This research however, operates within level 3 of the hierarchy detailed in Table 2.1, in which sensing of the environment is actuated on automatically by onboard systems.

Collision avoidance has two approaches: global and local. Global methods rely on *a priori* information about the environment and the obstacles in it, usually in the form of a map. From this, a global algorithm can create optimal trajectories between two arbitrary points in the entire search space.

Considering not only the entire environment search space, but also the vehicles' velocity space is a complex task; it is computationally expensive and is typically pre-processed. The robot then navigates along this pre-computed, collision-free path. A* and Voronoi graph approaches are popular choices for path planning, but the true difficulty in global methods is the localisation and path following behaviours of the robot [15, 16, 17].

Local collision avoidance methods rely on real-time sensor information to generate control inputs that avoid obstacles. These methods are applicable to unknown environments. Some local methods such as the front-facing doppler radar in [18] and the Nearness-Diagram [19] are purely reactive and have no memory of the environment or any predictive steps. The dynamic-window approach by Fox *et. al* considers the robot dynamics and operates in a small window of the full velocity space of the vehicle, providing some predictive control [20]. Bareiss *et. al* extend this to also considers the users desired control input, and apply it to a UAV [21].

Other UAV collision avoidance approaches that have been tested in simulation include an ultrasonic-based fuzzy logic controller [22], a open sectors method using LiDAR [23] and vision-based methods [24, 25]. Lastly, the APF approach models obstacles as negative charges which repel the "negative" vehicle. This is the approach considered in this research with details given in Chapter 3.

2.3.1 SLAM

Discussion on GPS-denied robotics would be incomplete without mention of Simultaneous Localisation and Mapping (SLAM), which aims to localise an agent in an unknown environment, while also generating a map of said environment [26]. SLAM algorithms are important for robots aiming for more autonomy than level 3, since a reusable map is required for path planning and exploration. An understanding of how SLAM works is not required in this research. It can be treated as a black box, with inputs being sensor data, and the outputs being a local map of the environment and the position of the agent in the map.

2.3.2 Sensors

In order for collisions to be avoided, they must first be detected by a sensor package onboard the UAV. Researchers have considered many options, but the most common sensors are non-contact rangefinders that have a low mass due to the limited payload capabilities of a UAV. The popular sensor technologies are briefly discussed below.

A sonar sensor depends on the time-of-flight of a pulsed sound wave in order to calculate distance. The wave is transmitted by an ultrasonic actuator, reflects off the obstacle, and is detected by an ultrasonic actuator. The beam width of an ultrasonic sensor ranges from 11°-62°, while their sensing rate (<10 Hz) and range (<10 m) are limited by the speed of sound. Despite their less than 12 g mass and 20 mW average power consumption (which makes them ideal for a UAV) they have been shown to have poor directionality due to the wide beam angle and frequent misreadings, especially when the angle of incidence on the object is high. However, these downsides haven't prevented their use as a collision avoidance sensor [27, 28, 29].

LiDAR functions in a similar fashion to sonar, except instead of sound, it pulses laser light. Since the speed of light is much greater than that of sound, a LiDAR can perform range measurements much faster (>4000 Hz). Since the beam angle of a single LiDAR is small <1°, the laser is often spun to provide range measurements in a plane, which reduces the effective scan-rate as a function of the rotation rate. The range (between 30 cm and 200 m), sample rate, mass (≈1 kg) and power requirements (≈2 W) of rotating LiDAR make them suitable for UAV collision avoidance and real-time SLAM [30].

The last common sensor, as used by Edwards [3] is the visual and visual-with-depth approach. Cameras are lightweight and cheap but produce a large amount of data, with an RGB camera requiring an order of magnitude more bandwidth than a LiDAR at the same frame rate. This increased data rate requires a fast processor which in turn increases power, battery and thrust requirements. With a reduced framerate however, they have found success in [24, 25]. By considering multiple consecutive frames or corresponding images from multiple cameras, depth can be extracted from the environment to measure obstacle proximity. Unfortunately, cameras present weakness in low-light situations making them unsuited for mining environments, although an iteration on LiDAR technology called RGB-D uses a matrix of lasers to directly measure depth, which is then mapped onto an RGB frame. These devices are promising for future robotics, but their mass and power requirements are still too large for UAV consideration.

2.4 Conclusion

A review of the working principle for a UAV is given, leading into a basic overview of how a pilot-friendly controlled UAV is able to abstract the complex dynamics into a translational problem. The challenges facing a UAV in a mining environment are presented, with specific focus on the spacial constraint that necessitates collision avoidance. Strategies for actively preventing collisions are given, where it is shown that global methods are not appropriate in the GPS-denied underground environment. Finally, three core rangefinding technologies namely sonar, LiDAR and camera are detailed listing their features and applicability to the research.

The chapter to follow discusses in depth the history of the APF algorithm, detailing it's origins, formulation and current state-of-the-art adaptations to robotics.

Chapter 3

Artificial Potential Fields

In this chapter the mathematical foundation for the APF collision avoidance algorithm is given, displaying the elegance of modelling the environment and an agent interacting via contrived forces. It is shown how this idea is used to compute collision-free paths through a known environment as a global planner, or to act in real-time as a low-level controller reacting to local changes. While versatile and computationally efficient, under some conditions there are 3 major problems: Goal Non-Reachable with Obstacle Nearby (GNRON); local minima; and oscillation, which are elucidated in this chapter. Following that is a review of the literature, from major historical contributions to more recent applications to UAVs. The holes in literature, particularly those that this dissertation aims to fill, are then connected together in the conclusion.

3.1 Review of Artificial Potential Fields

3.1.1 Modelling and Formulation

The APF algorithm is first described by Khatib [31] as a solution for real-time obstacle avoidance for manipulators and mobile robots. The philosophy of this approach is given by Khatib [31] as follows.

The manipulator (or mobile robot) moves in a field of forces. The position to be reached is an attractive pole for the end-effector, and obstacles are repulsive surfaces for the manipulator parts.

The strength of the algorithm is in the abstraction of desired position and obstacles into attractive and repulsive forces respectively. This allows the manipulator to treat the observed environment as a field of forces, the mathematics of which are well understood in terms of vector calculus. An attractive potential is formulated that will be large when the end-effector is far away from the goal, and 0 when the effector is at the goal [31]. The quadratic attractive potential is given by:

$$U_{Att} = \frac{1}{2}k(q - q_g)^2 \tag{3.1}$$

where

$$\begin{aligned} k &= \text{Positive attractive gain} \\ q &= \text{Current robot position [m]} \\ q_g &= \text{Goal robot position [m]} \end{aligned}$$

To calculate the resultant attractive force, the negative gradient of the potential is taken. [1]

$$F_{Att} = -\nabla U_{Att} = k(q - q_g) \tag{3.2}$$

The repulsive force produced by an obstacle is chosen to tend to infinity as the robot approaches the obstacles surface, much like two negative charges as described by Coulombs Law. The vector between the obstacle surface and the robot ($\vec{\rho}$) is calculated by $\vec{\rho} = q - q_{obs}$. Khatib [31] specifies that the potential function should be non-negative, continuous and differentiable within the region of influence (and 0 otherwise), choosing the following:

$$U_O = \begin{cases} \frac{1}{2}\eta(\frac{1}{\vec{\rho}} - \frac{1}{\rho_0})^2, & \text{if } \|\vec{\rho}\| \leq \rho_0 \\ 0, & \text{if } \|\vec{\rho}\| > \rho_0 \end{cases} \tag{3.3}$$

where

$$\begin{aligned} \|\cdot\| &= \text{The } l^2\text{-norm of a vector, denoting Euclidean distance [m]} \\ \eta &= \text{Positive repulsive gain} \\ \rho_0 &= \text{Repulsive distance threshold [m]} \\ \vec{\rho} &= \text{Distance vector between the robot and the obstacle surface} \\ &\quad\ \text{[m]} \end{aligned}$$

Once again, the negative gradient of the obstacle potential is taken to compute the force on the manipulator, which is seen to decay exponentially with distance:

$$F_0 = -\nabla U_0 = \begin{cases} \eta(\frac{1}{\|\vec{\rho}\|} - \frac{1}{\rho_0})\frac{1}{\|\vec{\rho}\|^2}\nabla\vec{\rho}, & \text{if } \|\vec{\rho}\| \leq \rho_0; \\ 0, & \text{if } \|\vec{\rho}\| > \rho_0; \end{cases} \tag{3.4}$$

[1] The $\frac{1}{2}$ factor from Equation 3.1 exists only to cancel with the 2 that comes out of the derivative.

Here, $\nabla\vec{\rho}$ denotes the partial derivative of the vector, and provides directional information. Where there are many obstacles the total repulsive force is computed by the vector sum of the repulsion from each obstacle:

$$F_{rep} = \begin{cases} \sum_{i=0}^{n_0} \eta(\frac{1}{\|\vec{\rho}_i\|} - \frac{1}{\rho_0}) \frac{1}{\|\vec{\rho}_i\|^2} \nabla\vec{\rho}_i & \text{if } \|\vec{\rho}_i\| \leq \rho_0; \\ 0, & \text{if } \|\vec{\rho}_i\| > \rho_0; \end{cases} \tag{3.5}$$

where

n_0 = Number of obstacles

$\vec{\rho}_i$ = Distance vector between the robot and the i_{th} obstacle [m]

The total APF is then the sum of the attractive force in Equation 3.2 and repulsive forces in Equation 3.5, $F_t = F_{GAtt} + F_{rep}$ [31]. Since these forces are artificial, and not actually acting on the manipulator, it is important to note that this is used as the control input to the manipulator, that then should be translated to motion by actuators, and is limited by the capabilities of the actuators or mobile robot.

3.1.2 Use in Robotics

It is noted that in Khatibs [31] work obstacles are modelled by geometrical primitives such as ellipsoids and points which are fed to the algorithm from some higher level controller. With modern sensing such as LiDAR the obstacle surfaces can be measured and directly used by the algorithm. Each point sampled on the obstacle surface from the robot-mounted sensor can be used to measure the distance vector ($\vec{\rho}_i$). This makes the algorithm suitable then for not only manipulators, but also mobile robotics as a reactive real-time collision avoidance algorithm, responding to live distance measurements in order to compute the repulsive force. This is also known as online and local collision avoidance.

In a fully known environment in which a robot is able to localise, the APF algorithm has seen use as a global (also offline, *a-priori*) path planner, computing a collision-free trajectory for the robot to move from a starting position to a goal. By discretising the environment with known obstacle positions and sizes, the entire space can be mapped by the potential function. Then, given some goal position, a path can be found using a heuristic function.

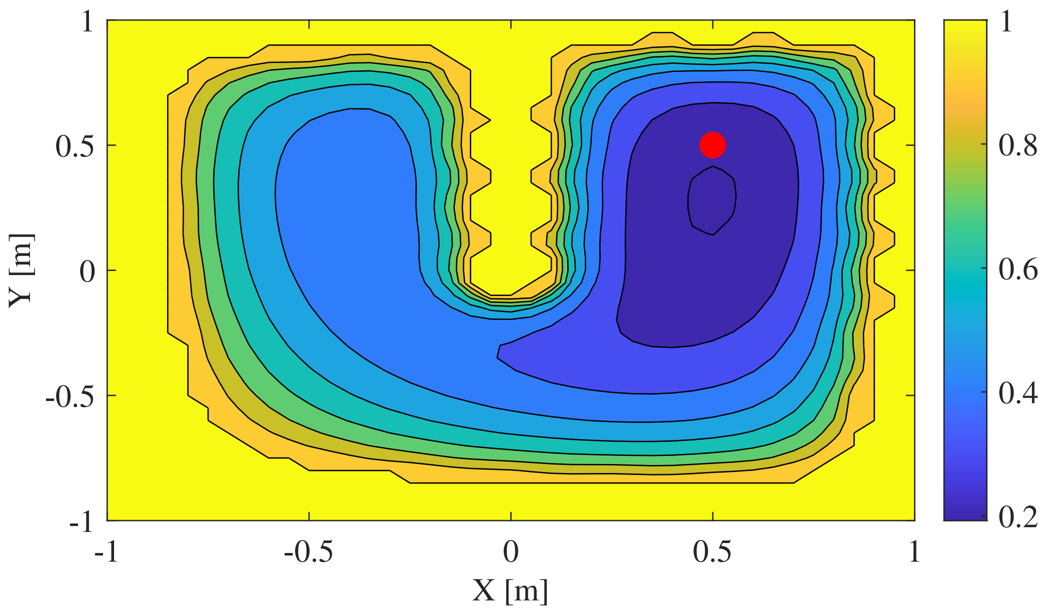


Figure 3.1: A contour plot of the potential field computed around a set of known obstacles with a goal position, represented by a red dot at (0.5, 0.5). The colorbar denotes potential, and is capped at 1.

An example environment with multiple obstacles is shown in Figure 3.1, where the contours represent equipotential regions. Starting anywhere in this environment, a path to the obstacle can be computed through, for example, gradient descent. Note in this example that the region of minimum potential is not actually at the goal, one of three key problems faced by the APF algorithm, discussed in more detail in the next section.

3.1.3 The Goal Non-Reachable with Obstacle Nearby Problem

It is well known, even by Khatib [31], that the APF approach exhibits minima problems. Figure 3.2 shows the problem on a macro scale: with an obstacle positioned at (0, 0), and the goal at (0.5, 0), the potential is computed by the sum of repulsive and attractive potentials at every point in the environment. Starting at any position in the field (except inside the obstacle), the artificial force on the robot will attract it to the global minima, which should be ideally be at the goal position. However, the global minima is at (0.48, 0) instead, which describes the Goal Non-Reachable with Obstacle Nearby (GNRON) problem for potential fields (attributed to Khatib [31]) since the robot will not reach equilibrium at the goal.

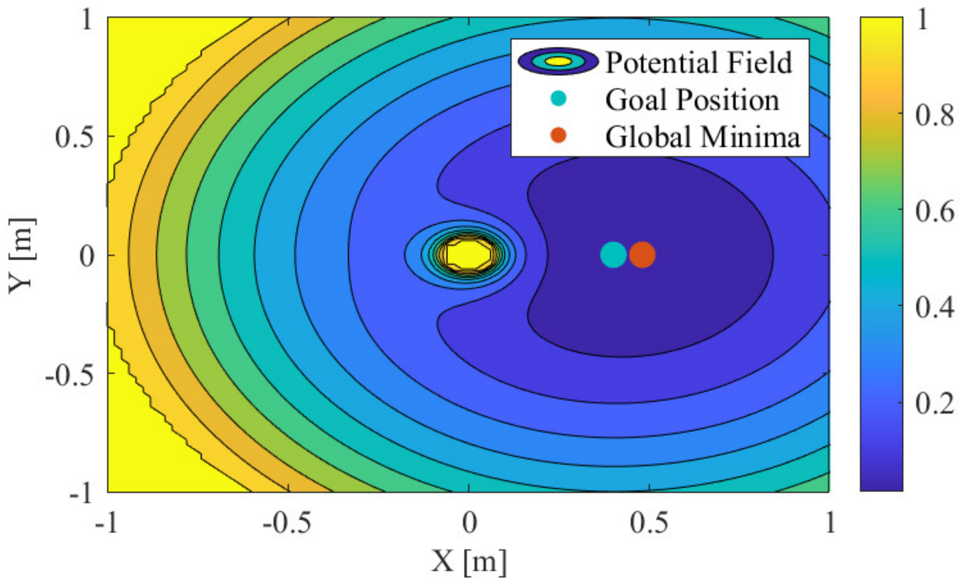


Figure 3.2: The APF computed around a single obstacle at (0, 0), with a goal at (0.4, 0).

Figure 3.2 highlights the GNRON problem, showing the Global Minimum Potential of this APF (red dot) being at a different position than the target position (blue dot).

The target position is too close to an obstacle for the attractive force to overcome the repulsive force of that obstacle. In the local collision avoidance context however, this becomes a nonissue, since the goal is either non-existent (as is the case in this research) or being constantly updated by a higher level navigation system which can get the vehicle out of minima traps.

3.1.4 Local Minima Traps

Another flaw of the APF algorithm is the existence of local minima or trap-situations [32]. Consider Figure 3.3, which shows two distinct minima for the artificial potential around two obstacles. If the robot were in either one of the minima, with the goal being the other minima, it could possibly be trapped there if the attractive potential is too low to overcome the "hump" between the obstacles. In the global path planning case, this is solved by Lee and Park [33] by creating an artificial obstacle at the local minima, however for the local case, the problem leads to the inability to pass between closely spaced obstacles.

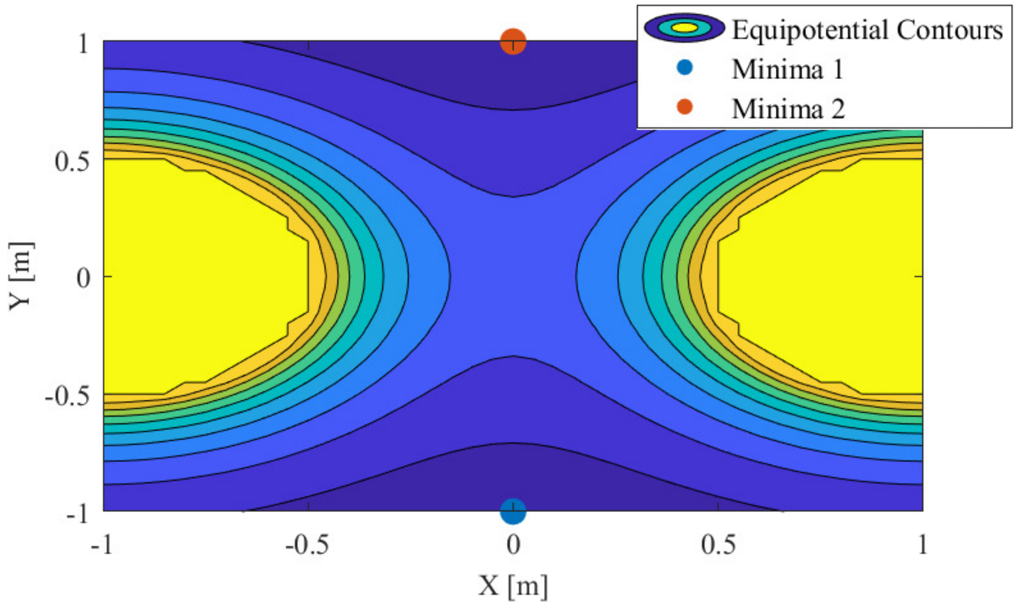


Figure 3.3: The APF computed around a pair of obstacles at position (1, 0) and (-1, 0). Equipotential regions of the field are shown by contour lines, where the field intensity is capped at 1. Two local minima are shown.

In Figure 3.3, while the potential between the obstacles (at (0, 0)) is relatively low, the gradient leading up to that midpoint is the barrier to entry. For an agent acting in this field to move from one minima to the other, it must navigate through a steep potential gradient (in the saddle). If the agent has insufficient attractive force to overcome the gradient, it is said to be trapped in a local minima.

3.1.5 Inherent Stability and Oscillation

The APF algorithm has distinct popularity due to it's stability in obstacle-dense environments. Stability in this sense is that the minima of the potential field are regions that are spatially central, and are therefore their maximal distance from all obstacles within the repulsive threshold, and therefore at the lowest risk of collision. However, this field cancellation effect is not always desirable.

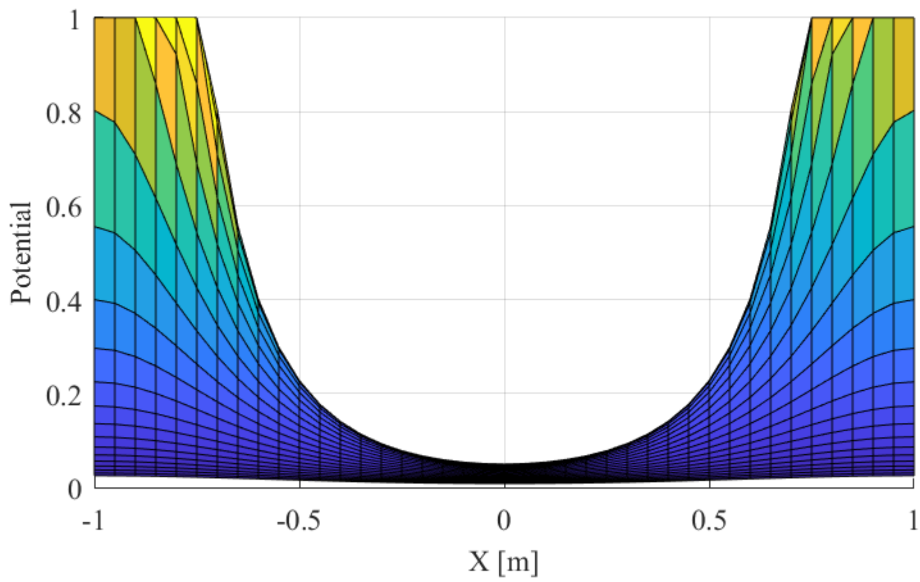


Figure 3.4: The valley shape along the x-axis between two obstacles, showing the minimum potential as the midpoint between them.

Consider two obstacles which create a potential valley between them as in Figure 3.4. A robot acting along the x-axis only, would be “pushed” into the midpoint between the obstacles, similar to a marble in a bowl, but in the same fashion would exhibit oscillatory motion before settling due to its friction. The problem is well known [31, 34, 32] and becomes particularly problematic for fast moving robots, narrow passages and/or obstacle perturbations. This will have to be addressed in order for the UAV system to be useful underground, where there are particularly narrow passages.

3.2 Literature Review

The APF approach and it's various problems are the topic of much research in robotic path planning, going through various iterations trying to address the major issues detailed above. The major variations are discussed below, as well as their applicability to the research.

Krogh [6] expanded on the concept by considering the relative velocity of obstacles, adjusting repulsive parameters to compensate for the momentum of the vehicle in the local/real-time context in order to reduce oscillations, and prevent potential collisions due to high speeds . This velocity can be computed using the same sensors used for local collision avoidance, requiring no absolute vehicle velocity to be computed.

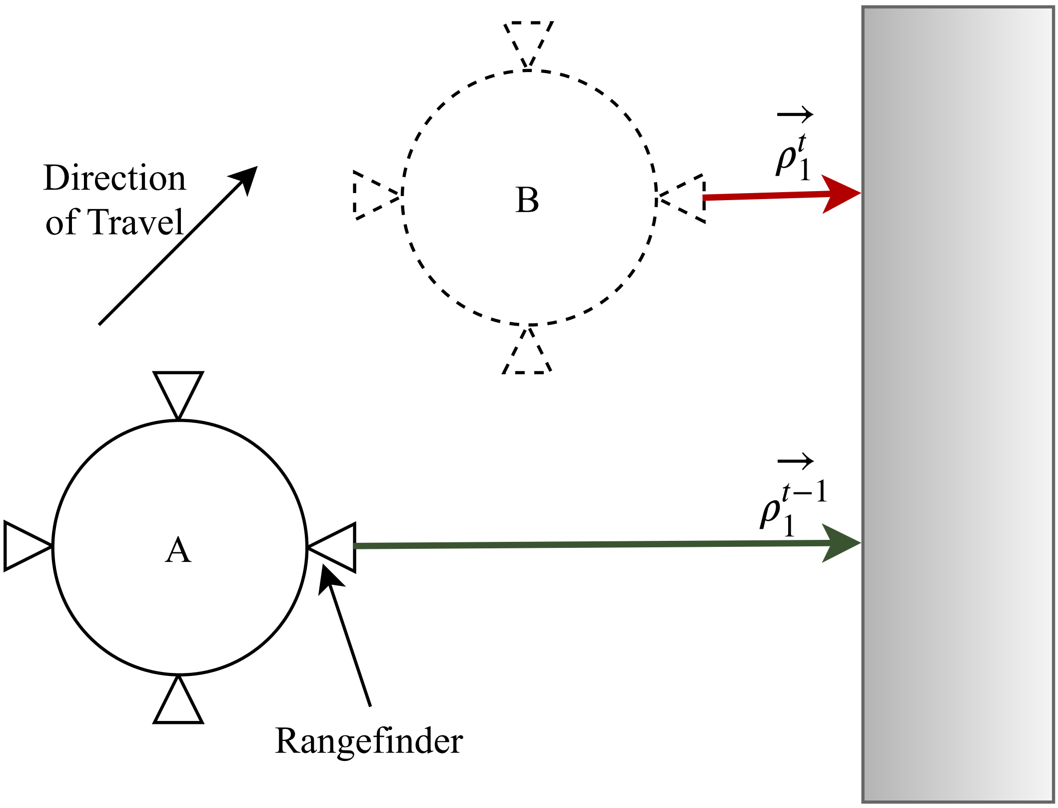


Figure 3.5: A mobile robot moving towards a static obstacle, which takes two range samples from the rangefinder on the right, first at position A, and then at position B.

Consider the mobile robot in Figure 3.5, which measures the vector to the wall at two positions (denoted by $\vec{\rho_i^t}$) as it moves. The relative velocity is then found using the rate of change of the vector (which is linked to the sampling rate of the sensor):

$$\vec{v_i^t} = \frac{\vec{\rho_i^t} - \vec{\rho_i^{t-1}}}{\Delta t} \tag{3.6}$$

In the case shown in Figure 3.5, the vehicle itself is moving both vertically and horizontally, but the relative velocity of that particular rangefinder will record a horizontal velocity only. The velocity scaling factor is defined for every obstacle as a linear function of the relative velocity and the maximum velocity of the robot[6].

$$\lambda_i = \begin{cases} 1 + \frac{|\vec{v_i^t}|}{v_{max}} & \text{if } \|\vec{v_i^t}\| \leq 0; \\ 1 & \text{if } \|\vec{v_i^t}\| > 0; \end{cases} \tag{3.7}$$

where

λ_i = The velocity scaling factor for the i_{th} sensor

v_{max} = The maximum translational velocity of the robot [m/s]

This scaling factor is only relevant when moving towards an obstacle, and has the effect of increasing the repulsive force when included in Equation 3.5.

$$F_{rep} = \begin{cases} \sum_{i=0}^{n_0} \lambda_i \eta (\frac{1}{\|\vec{\rho}_i\|} - \frac{1}{\rho_0}) \frac{1}{\|\vec{\rho}_i\|^2} \nabla \vec{\rho}_i & \text{if } \|\vec{\rho}_i\| \leq \rho_0; \\ 0, & \text{if } \|\vec{\rho}_i\| > \rho_0; \end{cases} \tag{3.8}$$

Thorpe and Krogh [6] then developed the Generalised Potential Field (GPF), which computes and follows critical waypoints using *a priori* environment information, all the while sensing and reacting to local/unknown obstacles along the route using the velocity-compensated APF . This method relies on some previous information to compute the critical waypoints, and suffers from many of the same problems described above, although oscillations are damped by the consideration of relative velocity. This method is able to effectively damp oscillations in tight tunnels, as well as increase robustness of the algorithm at higher velocities when used on a ground vehicle.

Borenstein and Koren made many contributions to Artificial Force Field (AFF) methods (the class of collision avoidance algorithms that use some artificial field), building on the work of Khatib [31] and Krogh[6]. The Virtual Force Field (VFF) method for local path planning and collision avoidance described in [28] uses a certainty grid to deal with the many shortcomings inherent in ultrasonic sensors as discussed in Chapter 2. The area around the robot is divided into a grid, and sensor hits within the grid increase that cells certainty. Cells within a window around the vehicle, with sufficient certainty then generate repulsive forces.

To overcome local minima, a trap-state detection algorithm is used, with a wall-following algorithm to recover from the trap state, as well as the same relative velocity considerations as Krogh [6] for reducing oscillations. However, after a detailed analysis and critique of APF algorithms [32], [34] developed an alternative Virtual Force Histogram (VFH) approach . This method still models the world as a grid, but it then reduces the local obstacles to a one-dimensional polar histogram, in which it chooses the best candidate angle to be the commanded direction. Relevant to this research however are key trajectories that are shown to cause unstable or undesirable behaviour for APFs:

- Closely spaced obstacles,
- Step-changes in a wall, and
- Narrow passages.

The VFH approach is particularly suited to non-holonomic steering vehicles such as ground-based rovers or autonomous vacuums, but have been implemented for an outdoor UAV by van Breda [35]. Van Breda used the A* algorithm with the VFH method for an outdoor UAV to perform collision avoidance. His experiments show the vehicle navigating through obstacle gaps of 10m, which is too unconstrained for a mining environment. The dependence on a Global Navigation Satellite System (GNSS) or odometry system to create a global grid disqualifies both the VFF and VFH approaches for an underground robot.

In more recent times, Sfeir *et al* propose a modified potential field that guarantees that the field potential will be minimum at the goal, regardless of nearby obstacles [36]. This eliminates the GNRON problem, however they note that the local minima problem is still present. They are able to reduce oscillations by adding a rotational force, turning the vehicle away from obstacles as they come into the periphery, which is particularly important for ground based and non-holonomic vehicles. They also note that the proper choice of parameters leads to smoother trajectories, but that the only way to find them is through trial and error.

Cho *et al.* present another form of potential fields that models obstacle position within a Gaussian distribution, creating the repulsive potential field from that. Similar to the VFH, they then compute the lowest-collision-likelihood direction that is most similar to the goal direction. This is also how many gap-finding algorithms function.

Their Obstacle Dependant Gaussian Potential Field (ODGPF) is immune to the local minima and global minima problem, and performs well around sparse obstacles compared to other APF methods, but once again is more applicable to non-holonomic vehicles in its current form.

Asadi and Ozgoli compare their own novel approach using Nesterov's Accelerated Gradient against a momentum-based and gradient descent optimisation, showing reduced oscillations and smoother paths. However, their method is not that much better than a momentum-based method, and is also applicable to path-planning rather than pure collision avoidance.

While a UAV is not strictly holonomic, since it rolls and pitches in order to translate, it can be treated as such in the context of an APF. Recent literature provides evidence that real-time collision avoidance for a UAV is possible using a potential-based method.

Mok *et al.* simulate a guided point-mas flying through a field of obstacles using a Gaussian Mixture Model to represent them (as opposed to primitives), and the Expectation-Maximisation (EM) algorithm to update the model with new data [37]. The EM approach allows for the predictive avoidance of dynamic obstacles, while not being computationally expensive, but this was not validated in their approach, nor was there experimental demonstration of the techniques.

Keeping a human in control of the UAV, while decreasing the level of competence required to operate them is a recurring theme iterated by many researchers. In this method, the attractive vector is defined by an operator input and coupled with some sensory information, the user can avoid local minima, while the APF algorithm provides reactive collision avoidance. In this way, a human performs higher-level navigation, while an on-board system performs lower level control and collision avoidance. The benefit of this is a relaxation of the computing requirements that one might need for a full 3-dimensional SLAM and autonomous navigation approach, which is of course favourable for reducing human input, but greatly increases costs, sensor requirements, network bandwidth, mass and complexity.

A great example of a human-in-the-loop approach is given by Lam et. al [13], which proposes that haptic feedback provides an intuitive operator interface, resisting pilot control inputs based on obstacles sensed in the environment. They use two novel AFFs called Basic Risk Field (BRF), which is based on the theory of the GPF and Parametric Risk Field (PRF), which considers the vehicle diameter, stopping distance, operator reaction time and maximum deceleration.

They also perform a sensitivity analysis of their algorithm in a 2D simulation under the assumption of a linear transform function between the velocity command and the UAV velocity. The obstacle-rich environment introduces the vehicle to the following situations: control vector directly into a wall, control vector parallel to a wall, moving from a wide tunnel into a narrow tunnel with a step-like change, control vector through closely-spaced obstacles. Note the similarity in experiments with [32]. From this sensitivity analysis, a range of values for the repulsive gain is found, noting that smaller values result in a field that is too weak to avoid collisions, while larger fields prevent freedom of movement. The simulated sensor is a 360^o scanner with a 50m range, which provides environment information to both the PRF and BRF, with the PRF showing superior performance in terms of robustness against collisions and oscillation as well as faster robot velocity.

Lam notes an important tradeoff between safety and operability that is controlled through parameter selection. Fields are not only influenced by the choice of parameters, such as gain and maximum repulsion, but also by the environment: how close together passages are between obstacles and the geometry of obstacles.

Moving underground, Kanellakis *et al.* [38] propose using a simple APF for a UAV to fly in a mine shaft collision-free. Using ROS and Gazebo to simulate the dynamics of a hexacopter, as well as a simple ultrasonic array, the APF implementation was able to locally replan the trajectory through a 30m vertical mineshaft with obstacles that intentionally obstruct the initial trajectory. The approach is simple but effective, and while not particularly novel in the development of APFs, demonstrates their versatility as well as suitability for use in mining environments.

In general, literature shows the vast use cases for APF in many spheres of robotics, and with papers being published as recently as 2019 on the topic, shows the relevance of the algorithm. Despite the vast amounts of processing power available compared to the times of first conceptualisation, there is still vast opportunity for real-time algorithms with low computational cost.

3.3 Conclusion

The APF method is designed such that obstacles impose a repulsive force on an agent. From this concept a mathematical description is given, allowing the algorithm to be applied across many situations where collisions must be avoided. While relatively weak as a global path planner due to the GNRON and local minima, it is suitable as a real-time system since obstacles far away do not impact the agent, while nearby obstacles cause a repulsive action. A review of the literature reveals the adoption of APF onto many Unmanned Ground Vehicles (UGVs) with success.

UAVs are identified in the previous chapter as requiring active collision avoidance in order to operate in underground mines, and the APF algorithm is a popular choice in literature. The risk of performing collision avoidance experiments with real UAVs has driven most of the research to be performed purely in simulation. Collisions do not matter in a simulation, and the only cost of choosing poor parameters is having to reset the vehicle. In a real-world scenario, the parameter selection must be reasonable before flight tests in order to prevent irreparable damage. If this can be achieved, real experimentation is warranted as a verification of simulation results.

Chapter 4

Problem Statement and Experiment Design

An analysis of the above literature is given in order to identify the lacuna in knowledge that this research aims to fill. The research is then split into specific questions, detailing how they can be measured through metrics.

The research objectives are:

- A novel tuning methodology for the APF algorithm.
- An experiment testing intrinsic parameters used by the tuning mechanism.

The tuning methodology is detailed in this chapter followed by considerations required for underground teleoperation. Finally, the experiment design is given.

4.1 Research Questions

Literature on collision avoidance discussed in Chapter 2 reveals many local and global approaches to collision avoidance. Focusing on local, real-time contexts, the APF approach stands out as a simple, elegant and fast algorithm. In Chapter 3, the problems of oscillation, GNRON and local minima are detailed, as well as the long history of contributions to try and address these, as well as many simulated applications of the algorithm. Since the APF method will be used for real-time local collision avoidance, oscillations and local minima will have to be addressed.

Many researchers note the importance of parameter selection in the performance of the algorithm, but tend to tune the parameters through trial-and-error, which is time consuming. Iterative tuning also increases the risk of dangerous UAV collisions. Narrow tunnels and gaps between obstacles are noted as a particular challenge for APFs, causing oscillations, and minima traps for poorly tuned fields [32]. Existing literature also neglects the effects of sensor sampling rate, angular resolution and noise on the algorithm. Taking into account these deficiencies, noted from literature, the following research objectives are given:

1. **Provide a methodology for choosing potential field parameters η and ρ_0, rather than through iterative testing. Also provide a sensitivity analysis of these parameters, as well as sample rate and angular resolution (similar to Lam *et. al* [13]) noting each parameters impact on APF performance in terms of controllability, oscillation and proximity.**

2. **To determine how the tunnel and gap width affects the performance of the APF. This will help determine the algorithm's suitability for the constrained mining environment.**

4.1.1 Safe Flight and Metrics

In order to resolve these issues quantitatively, a metric is used to benchmark experiments. A simple but effective metric is to compare the number of collisions of the UAV across each experiment as in [21], but a single collision is enough to cause unrecoverable damage to the test platform.

Another metric often used shows the vehicle trajectory in the environment in the presence of obstacles using some third-party position-tracking such as a VICON motion capture system, which is expensive, or GNSS, which doesn't work underground. Instead, the trajectory in the real-world experiments is tracked using a SLAM implementation that is a combination of the Canonical Scan Matcher [39] for odometry and a Rao-Blackwellized particle filter SLAM solution [40]. In the simulated environment, the vehicle position is a known state which can be used to track the trajectory without any SLAM method.

Showing the UAV trajectory provides a tool for observing oscillations and local minima. However, trajectory alone is difficult to compare. In the sensitivity analysis, the flight performance metric for pilot authority and oscillation is calculated as the deviation of the real trajectory from an idealised path through the fixed environment. In the actual experiments, the metric is calculated as the deviation of the total applied force from the pilot intended force.

Proximity to the nearest obstacle provides another metric that is useful for determining if safe flights are achieved under different conditions. A definition for safe flight is to follow. A collision will occur if an obstacle is closer than the vehicle dimensions ($r_{vehicle}$), defined as the distance from the center of the UAV to the tip of the blades, as shown in Figure 4.1. It is critical that a collision avoidance algorithm avoids this situation.

A buffer is necessary to reduce the risk of the wall effect. The wall effect is a phenomenon describing the unstable aerodynamic behaviour of a spinning rotor near a surface, which causes vehicle instability, increasing the risk of collision [41]. The wall effect buffer (r_{wall}) should be at least $\frac{r_{vehicle}}{2}$, as the wall effect is negligible at this distance[41]. While translating the UAV has momentum based on it's velocity and so cannot stop instantly, even if commanded to. A second buffer accommodates for the distance that could be covered between sensor samples (sensor sample period, T_{sensor}) while at maximum speed. This is a static value, since the vehicle maximum velocity and sample delay are both static values based on the platform and sensor selection.

$$r_{velocity} = T_{sensor} v_{max} \tag{4.1}$$

Safe flight is then achieved when no obstacle is measured in the buffer region, meeting the following condition. Figure 4.1 shows the buffer components relative to a UAV frame.

$$\vec{\rho}_i > r_{buffer} = r_{vehicle} + r_{wall} + r_{velocity} \tag{4.2}$$

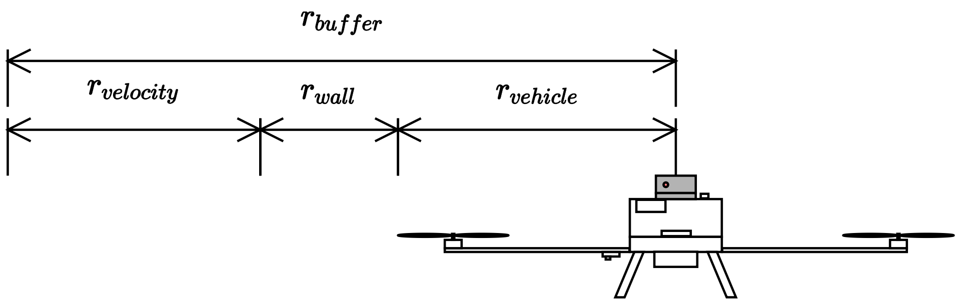


Figure 4.1: The safety buffer distance relative to the UAV.

In Figure 4.1 above, the safety buffer is defined as the region around the UAV in which obstacles can cause unsafe conditions. The safety buffer is composed of the vehicle, wall-effect, and velocity buffers. Obstacles within a circle centered at the vehicle centre with diameter equal to the tip-to-tip dimension will cause irreparable damage to the UAV blades. A wall within the wall-effect buffer may trigger the wall effect, effectively pulling the UAV towards the wall and resulting in a collision. Finally, the velocity buffer is designed to compensate for the distance that could be covered by the UAV at maximum velocity between two sensor samples.

In some research, the safety buffer may be formulated differently, and is used to artificially increase the size of obstacles in the periphery [42, 20, 13]. In the APF context however, it is desirable to have a maximal repulsive force vector from an obstacle on the edge of the safety buffer. This idea is what leads to the following section.

4.2 Parametrisation

This section proposes a method for choosing parameters of the traditional APF. By considering the minimum size of the obstacle to be avoided and the sensor angular resolution, it is possible to calculate the repulsive gain η, which is arguably the most important parameter to tune, since setting it too low could cause collisions, while setting it too high leads to the attractive forces being negated, and possibly unstable oscillations, again leading to collisions. Quantifying this range of values is left up to the sensitivity analysis in Chapter 5.

Consider a single obstacle with a flat side of length w which is sensed by radial rangefinders from a distance (d) as in Figure 4.2. We then ask the question "what is the minimum repulsive gain that will guarantee 100% repulsion at this distance?". To answer this question, first note that the rangefinders with some angular resolution θ_{res}[1] will measure

$$n_0 = \lfloor \frac{2arctan(\frac{w}{2d})}{\theta_{res}} \rfloor \tag{4.3}$$

points on the obstacle . The resultant force vector F_{rep} can then be computed for all points ($\vec{\rho_i}$) by Equation 3.5 which is rearranged to obtain the following form:

$$\eta = \frac{F_{rep}}{\sum_{i=0}^{n_0}(\frac{1}{|\vec{\rho_i}|} - \frac{1}{\rho_0})\frac{1}{|\vec{\rho_i}|^2}} \tag{4.4}$$

Choosing the maximum magnitude for F_{rep} to be 1 (corresponding to 100% repulsion), the value for η can be determined that will guarantee maximal repulsion at distance d for an obstacle size w. The value of d should be equal to the safety buffer (r_{buffer}) defined in the section above.

It is important to note the approximation of all obstacle surfaces as flat, where one can certainly expect to encounter many different surfaces. Obstacles which are perceived as smaller than w are likely to result in unsafe conditions, however, the parametrisation technique does provide a method of obtaining sensible parameters, which is an improvement on the risky trial-and-error approach.

The next important parameter to characterise is the repulsive distance threshold, ρ_0. The effect of this variable is twofold, first as the maximum distance from which a force can be applied, and second as a reciprocal term in the repulsive field, causing the field intensity to be zero at the threshold, which ensures smooth and continuous behaviour. The greater this distance threshold, the smaller it's effect on the field, since obstacles further away provide exponentially lessening force, but also because the $\frac{1}{\rho_0}$ term becomes smaller. The choice of this parameter is then controlled by the maximum useful range of the sensor, but also the desired steady-state distance from obstacles given no input (obstacles will continuously apply a force while within the threshold). In corridors and obstacle-dense environments, it is desirable to have force effects from as many obstacles as possible in order to achieve spatial centering as well as prevent oscillations. For this reason, ρ_0 should be equal to the maximum useful range of the rangefinder in an obstacle-dense environment.

[1] For a LiDAR sensor, this angular resolution is a fixed value decided by the OEM. In the case of an ultrasonic or other single-point rangefinder, array as in Du and Liu, the angular resolution depends on how they are arranged on the platform [22].

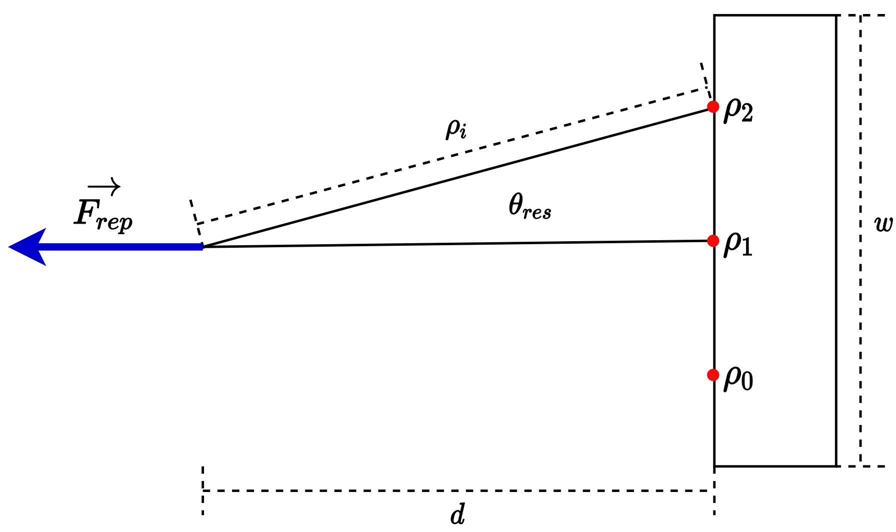


Figure 4.2: A single static obstacle with a fixed width at a known distance from a set of sensors

In Figure 4.2 a LiDAR sensor at a distance d from a wall of width w detects three points ($\rho_{1,2,3}$) on the wall. The angle between the rays connecting each point and the LiDAR is θ_{res}, a property of the LiDAR scanner. Using this geometry, and the formulation for the APF algorithm, it is possible to calculate the repulsive gain η that would provide the desired magnitude of the repulsive vector F_{rep} to keep the wall within a safe distance.

4.3 Adapting APF to Underground Teleoperation

Some modifications to the APF algorithm are required to enable teleoperation of a UAV underground. In Chapter 3, the field intensity is computed for every point a discretised environment, determining the potential across the 2D environment. This is computationally intensive, and unnecessary since the force of each obstacle on the vehicle at it's current position can be computed directly using Equation 3.8. The GPF proposed by Krogh [6] is preferred to the standard APF since it damps oscillations, and enables more robust collision avoidance at higher velocities, or in the presence of external disturbances such as wind.

The attractive force is redefined to allow direct human input as opposed to the goal position. Instead of being derived by the gradient of potential, it is given simply as a function of the human (or higher level controller) desired direction.

$$F_{Att} = \vec{u} \tag{4.5}$$

where

$$\vec{u} = \text{A control vector for teleoperation in the range } [-1 \to 1]$$

The control input is still a sum of the attractive and repulsive force vectors, each mapped into the range $[-1 \to 1]$. These control vectors dictate the desired translation vector, but since the UAV is non-holonomic, the translations are to be transformed into roll and pitch rotations. Yaw rate and height are controlled separately by the pilot.

4.3.1 Human Teleoperation

In hobbyist UAV flight, pilots interact with the flight controller using a radio transmitter, demanding roll, pitch, yaw and thrust set-points. The pilot receives limited telemetry such as battery life and current attitude, which is insufficient for laboratory setups. Adapting hobbyist hardware and technology is important to reduce cost and development time. Researchers typically rely on a WiFi connection to relay far more telemetry back to a computer known as a Ground Control Station (GCS), storing and displaying information such position, attitude, sensor health and images. The pilot then uses a human input device such as a joystick connected to the GCS to command attitude. In this research, the pilot input is handled by the APF algorithm running on the companion computer first before being passed to the flight controller, as in Figure 4.3b.

Latency is a concern for a teleoperated device, with signal transmission delays in the hundreds of ms possible for a noisy wireless network. For this reason, it is necessary to compute essential collision avoidance components on-board the UAV. This also solves the potential problem of a loss-of-signal wherein the companion computer can process collision free trajectories without human input or reliance on an external computer.

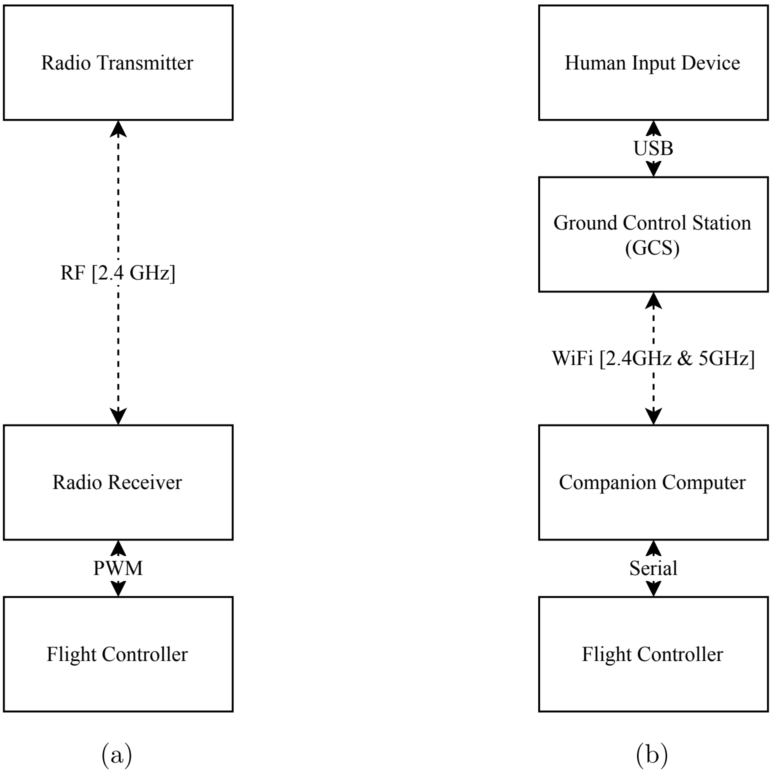


Figure 4.3: Differences between a) the hobbyist communication stack and b) a common research communication architecture.

Figures 4.3a and 4.3b above show two common communication stacks. Hobbyist pilots do not require the level of control and telemetry that is available on the research platform, and therefore need only a transmitter and receiver to command a UAV. A research platform requires more stages to enable autonomy, so a WiFi based wireless network is sensible. On the GCS side, a human input device allows pilot control over the network, which is interpreted by software on the companion computer before being relayed to the FC over a serial bus.

Another concern when teleoperating an UAV is line-of-sight. Hobbyist pilots typically fly with the vehicle in full line-of-sight, but for a vehicle teleoperated underground, this may not always be possible. This is where the WiFi network shines, since LiDAR and camera information can be transmitted to the GCS to be used as visual feedback for the pilot when the vehicle is not visible. Although not tested in this research, a fully remote setup is enabled by the visual and sensor feedback, such that the pilot and GCS can be situated above ground.

4.4 Experiment Design

4.4.1 Sensitivity Analysis

The first set of experiments are designed to test intrinsic parameters used in the parametrisation. The inputs to the parametrisation are the vehicle maximum velocity, vehicle center-to-tip dimension, sensor sample rate, sensor angular resolution, sensor maximum range and minimum obstacle width. For a given UAV, the maximum velocity and center-to-tip dimension are fixed and it assumed that the sensor maximum range is fixed.

To test the parametrisation method and the effect of sensor sample rate, angular resolution, velocity compensation and the minimum obstacle size, a static simulated environment is to be used. Each of the above variables are independent, varying between extremes to observe: firstly under which conditions the parametrisation fails, and secondly the broader effects of varying the parameters. The dependent variables are then proximity, pilot authority and oscillations which are observed both quantitatively and qualitatively.

The hypotheses are presented below for each variable:

- It is hypothesised that a decreased sample rate leads to an increased oscillatory behaviour when using the APF algorithm.
- It is hypothesised that for very low sample rates, the parametrisation will not create a sufficiently sensitive field to avoid collisions.
- It is hypothesised that a smaller angular resolution will result in a less oscillatory response.
- It is hypothesised a lower minimum obstacle width will result in higher pilot authority and lower proximity measurements.
- It is hypothesised that the GPF (APF with linear velocity compensation) will outperform the standard APF in terms of proximity, pilot authority and oscillation.

4.4.2 Physical Validation

The second set of experiments are designed to put into practice the results of the sensitivity analysis and to validate the APF algorithm functionality on a real UAV. The testing environment also serves as a stress test to verify the algorithm at its limits. Closely spaced obstacles, step changes along a wall and narrow passages are shown by Borenstein and Koren [34] to negatively impact the APF algorithm performance. The experiments are explicitly chosen to represent the worst case scenarios of collision avoidance. In each of these experiments, the proximity, pilot compliance and oscillatory behaviour are measured and stored in real time, while the trajectory is post-processed using the SLAM algorithm discussed above. Each experiment is performed once in simulation and on a real platform.

Single Obstacle

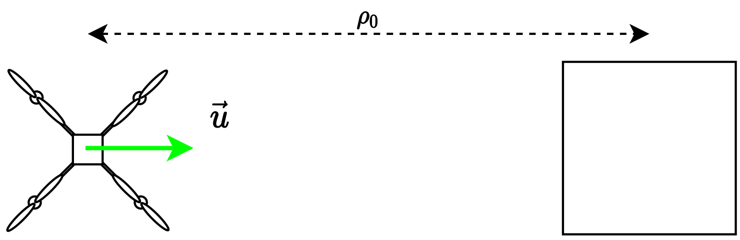


Figure 4.4: Experiment 1 layout where $\vec{u}$ is the piloted direction.

The first experiment tests the validity of the parametrisation method using a single, static obstacle of known dimensions that is obstructing the piloted trajectory, positioned as in Figure 4.4. It is designed to test the computed parameters' ability to avoid an obstacle, while respecting the minimum distance and piloted direction. The UAV will take-off to a height of 1 m from a fixed distance away from the obstacle, and then have a fixed pilot vector directly into the obstacle. It is hypothesised that the UAV will avoid the obstacle, resulting in low pilot compliance.

Wall

In the previous experiment, there are many collision-free candidate trajectories (to the left/right of the obstacle). This experiment then observes the behaviour when the pilot commands a course into a wall at maximum velocity. The layout is the same as that of the first, except that the obstacle should be wide enough such that there are no collision-free options along the piloted direction. It is hypothesised that this will bring the UAV to a stop.

Two Obstacle Barrier

While the local-minima problem of the APF algorithm is not solved by the parametrisation, an experiment has been derived to determine at which gap-size a local minima presents passage. The results of the experiment will quantify the trade-off between controllability and safety in terms of level of constraint and the repulsive gain.

The primary goal of the pilot is to move between two obstacles that are separated by some gap. For gap sizes smaller than twice the safety buffer safe flight is not possible, but the experiment will still be conducted until the path becomes impassable. The barrier/gap will be approached perpendicularly to the two obstacles so as to present the worst case change in gradient at the highest velocity, as in Figure 3.3. It is hypothesised that the smallest passable gap distance will be double the safety buffer distance.

Tunnel

Borenstein and Koren [28] showed through experiments that the basic APF approach is oscillatory in tunnels, and worsens with perturbations in the tunnel. Oscillation is not ideal, since it may increase the effort of the pilot if the pilot tries to compensate for the unwanted motion. However, it is possible that using Krogh's proposed velocity compensation technique may be sufficient to damp these effects. This experiment will test the UAV teleoperation in a mock-mine environment, observing for oscillation, flight safety and controllability. It is hypothesised that the GPF approach with velocity scaling will be sufficient to damp oscillations, while providing safe teleoperated flight.

4.4.3 Delimitations

Chapter 2 presents a UAV motion model based purely on translation in the x- and y-axis. The altitude of the UAV and obstacle height is not explored in this research. The primary reasoning for this is the cost of a LiDAR that provides a full 3-dimensional point-cloud is too high. The chosen LiDAR scanner only provides relative obstacle locations in a horizontal plane around the UAV as in Figure 4.5.

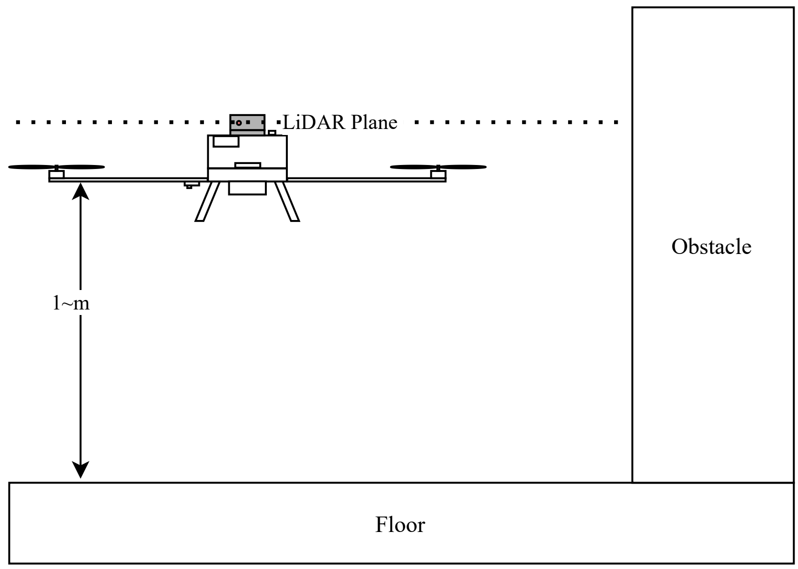


Figure 4.5: A side-view of the experiments highlighting the LiDAR plane relative to the ground.

The UAV is fitted with a downwards facing rangefinder that provides an altitude measurement that is consumed by the FC. For every experiment the UAV will takeoff to an altitude of 1 m which is maintained throughout the experiment. A PID controller onboard the FC holds this altitude, which is chosen to avoid the ground-effect, while minimising the distance to fall in the event of a fault. All obstacles considered a part of the experiment will be 1.4 m tall such that they are always visible by the LiDAR.

4.5 Conclusion

The APF algorithm is identified as a suitable candidate for real time collision avoidance. However, the body of existing literature relies on trial and error to identify sufficient repulsive parameters. With that in mind, a novel tuning methodology is proposed that considers the UAV dimensions and velocity as well as the sensor parameters to generate an approximate repulsive gain that sufficiently repels a nearby obstacle.

The tuning method is developed in this chapter, along with a set of experiments designed to test the limits of the method. Additional experiments are also proposed to validate the APF approach shown in literature on a physical UAV platform in a mine-like environment. The chapter to follow details the simulation environment and also contains the sensitivity analysis for the parametrisation variables.

Chapter 5

Simulation and Sensitivity Analysis

The previous chapter presents a methodology to choose the repulsive gain of an APF collision avoidance algorithm considering the sample rate and angular resolution of the proximity sensor; obstacle width; and minimum distance to an obstacle. This simulation environment allows for the validation of the parametrisation while also investigating the effects of these factors on the APF in terms of controllability, collision proximity and oscillation.
The models used to represent physical systems such as the environment and LiDAR are presented. The UAV system model is treated as a grey box, where a model is determined by data from the real system. An intentionally difficult to navigate environment is used to characterise the effects of the listed factors on the performance of the algorithm. The simulations also provide insight into the range of suitable values under which the parametrisation will ensure collisions are avoided. The results of this chapter are used to inform the decisions around choosing parameters for real systems.

5.1 Modelling

A common approach in UAV research is to develop a comprehensive simulation in order to test prototypes risk-free [43]. This has been done ad nauseam, especially for low-level UAV attitude control, which is evidenced by numerous commercially available flight controllers, capable of stable attitude control [44].

This simulation assumes a level of autonomy above that of the attitude control layer, where the UAV is a user-friendly quadcopter. The assumption means a less-comprehensive simulation of actual vehicle dynamics, which is appropriate since the goal is to better understand the APF algorithm and it's interactions with the environment, rather than attitude stability.

The experiments are run in a `MATLAB` program for a total of 10 seconds, sufficient time to traverse the course below the maximum speed 4 m/s. A custom physics engine computes the changes in velocity and position of the UAV every 5 ms.

5.1.1 Environment Model

Obstacles and free-space are represented in a 2D binary-occupancy grid of 15x15 m with a resolution of 10 cells/m. Each cell in the grid is either occupied (filled in black) or empty (filled in white), and can be manually edited to create obstacles of various shapes and sizes. Figure 5.4 shows the static environment used for the sensitivity analysis.

5.1.2 UAV

The UAV is modelled as a pilot-friendly quadcopter that only *translates* in the horizontal plane, without yaw[1]. Lam *et al.* [13] previously assumed the model in Figure 5.1. A translational velocity model is sensible in this case over a positional or attitude model. A position-based model suits a simulation since the UAV state is known, but it would not translate into the real world where position information is not available without SLAM or GNSS. An attitude model breaks away from the holonomic abstraction presented in Chapter 2.

[1]The UAV is assumed to be in flight at an altitude where the ground effect is negligible and can be neglected from this model.

Lam *et al.* [13] present the following transfer function between the velocity setpoint and the translational horizontal velocity:

$$G(s) = \frac{2}{s+2} \tag{5.1}$$

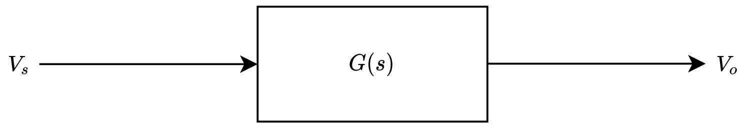


Figure 5.1: Block diagram of the UAV system model where V_s is the velocity setpoint and V_o is the UAV horizontal translational velocity. The complexities of the real system dynamics and control are discussed in Chapter 2.

Since telemetry on the real system records the target- and estimated velocity on the flight controller, the plant dynamics can be estimated more accurately. Figure 5.2 below shows the real step response with a maximum piloted input when flown in a large open area at a constant height above ground level. This data is sampled at 10 Hz, which limits the ability to observe higher-frequency effects. Evident in this data are the effects of feed-forward control, since the output velocity reaches the maximum long before the set-point itself does. The maximum velocity is 4 m/s, while the maximum acceleration/deceleration is estimated to be 8 m/s^2.

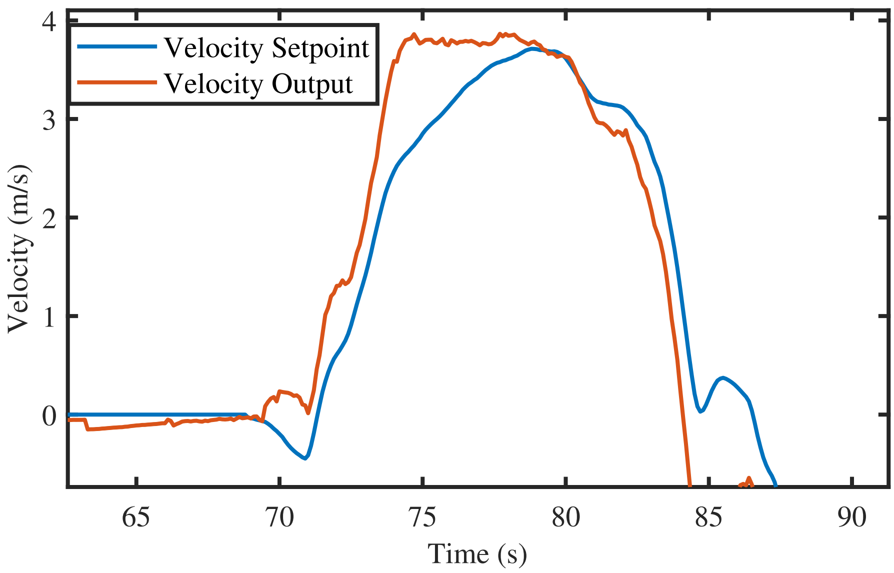
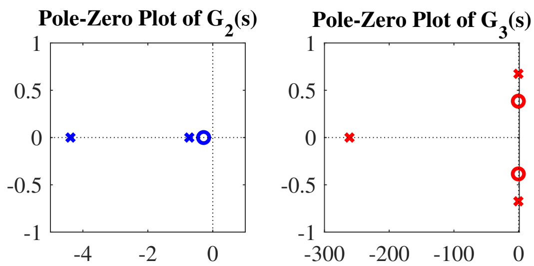


Figure 5.2: The velocity profile of the UAV showing the target and actual velocity in m/s.

Figure 5.2 shows measurements from the onboard logging of the real system revealing minor non-linearities in the y-velocity step response. The velocity setpoint rise-time is limited by the UAV maximum acceleration, leading to a slow rise despite a step input in the pilot stick deflection. Note the effects of feed-forward control, as the actual vehicle velocity reaches the maximum (4 m/s) before the set-point does.

By innervating the UAV with erratic inputs, a transfer function can be fit using the `System Identification` applet in `MATLAB`. Appendix A contains more information regarding the data used for fitting. Three transfer functions, $G_1(s), G_2(s)$ and $G_3(s)$- where n in $G_n(s)$ indicates the order- are fitted to find an appropriate order to describe the system. The models fit the provided estimation data with an accuracy of 61%, 85% and 90% respectively. The pole-zero plots of $G_2(s)$ and $G_3(s)$ are given in Figure 5.3. All of these functions suggest a faster response than G_s, but $G_1(s)$ follows the set-point too closely, which loses the feed-forward effects where the higher order models do not. Although the actual system is likely many-orders higher, $G_2(s)$ is chosen to reduce complexity. More information about the data used for system identification and identified systems is provided in Appendix A.

$G_2(s)$ has a relatively close pole-zero pair which cancel out, leaving a dominant pole at -4. The time constant is then approximately 0.25 s.



(a) (b)

Figure 5.3: Pole-zero plots of the second and third order systems fitted to real velocity input/output data.

Figure 5.3 shows the pole-zero plots of the system as identified by the `System Identification` applet. The estimated models achieved an accuracy of 85% (5.3a) and 90% 5.3b respectively. The systems show differing left-most poles, with G_3 showing a much faster pole (-300 on the real plane) than G_2 which behaves similarly to the feed-forward characteristics seen in the real system response.

It is assumed that the dynamics are equivalent in both the x and y directions, which was validated with real x-velocity data passed through $G_2(s)$ which achieved an accuracy of 79%. Having the same model for both axes simplifies the problem to a fully holonomic model for which the input velocity is a vector sum of the pilot and repulsive commands. The change in position is then computed every time step as the inputs are passed through the system model. Because the simulation is discretised the model must be as well, which is done with a Zero-Order Hold approximation where the sample rate is 0.005 s.

5.1.3 LiDAR

The LiDAR is modelled as a planar 360° scan around its centre, which is attached to the UAV centre. These rays collide with obstacles in the occupancy-grid, where their distance is computed, otherwise they are returned as NaN. The noise of these distance readings is modelled as a Gaussian distribution with a zero-mean ($\mathcal{N}(0, \sigma^2)$) noise. The scan rate, angular resolution, and noise variance are all variable in this model. Because of the ray-based implementation, the model takes range samples on an objects surface, much like a real LiDAR. The real LiDAR laser sensor is able to scan at 4 kHz which is much faster than the system dynamics, and is modelled as an instantaneous distance measurement.

It is assumed in this model that the LiDAR does not exhibit any veiling around an objects edges [45] and that obstacle surfaces are perfectly reflective, regardless of the angle of incidence. It is also assumed that the effects of roll and pitch angle of the UAV are neglible on distance readings. This only holds for small angles, which would not be true in more agile UAVs.

5.2 Sensitivity Analysis

In Chapter 4, a number of experiments are proposed, testing extrinsic factors regarding the environment. For the purposes of this sensitivity analysis, a custom, static environment will be used, with intrinsic parameters set as in Table 5.1, unless used as the variable under scrutiny. The obstacles are designed to challenge the collision avoidance, while also allowing a fixed control vector (north-east). Figure 5.4 shows the environment starting position at the red star, as well as three notable obstacle configurations:

- **A** - 1.6 m wide doorway leading out of the starting room,
- **B** - 4 m wide tunnel with a single obstacle perturbation, and
- **C** - 0.4 x 0.4 m obstacle.

Ideally, the UAV would be as North-East as possible at the goal position, while also avoiding collisions and unsafe flight conditions. The measurement of trajectory and minimum proximity provide good metrics to quantify this behaviour.

Additionally, deviations away from the local minima in each experiment indicate a suboptimal path [46]. When not traversing a Minimum Potential Path (MPP), the APF computes some repulsive force that is trying to find the local minima. This means that computing the total repulsive force on the UAV for the entire run provides a comparable metric of path optimality and smoothness[2] [44].

The deviations from this minimum-potential path are also computed as the perpendicular distance between the paths as in Figure 5.5, giving an error metric which is computed as the Mean Absolute Error (MAE). The MPP to which each trajectory is compared is calculated using gradient descent to find the trajectory through the environment that minimises repulsive potential with a constant attractive force towards a goal at (14, 14).

[2] A purely-local algorithm cannot be expected to produce optimal, energy efficient paths, especially not with a constant forcing function and only real-time sensor data.

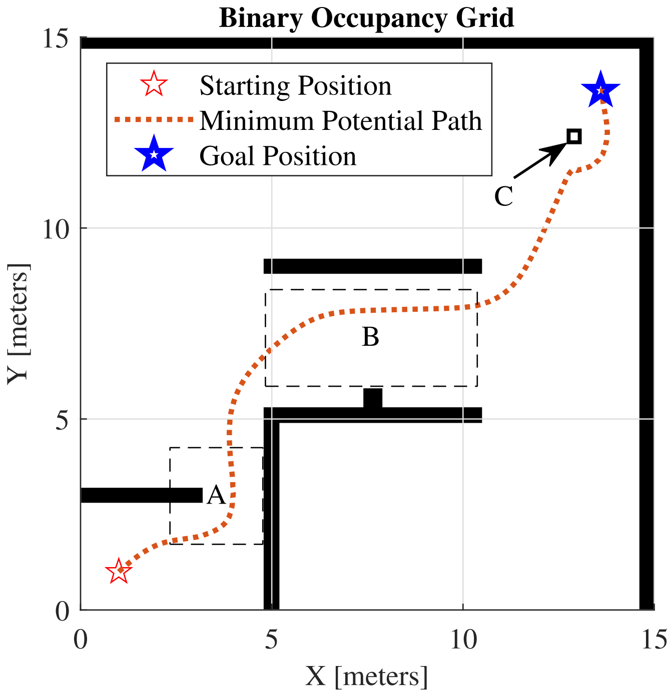


Figure 5.4: The sensitivity analysis binary obstacle grid environment.

Figure 5.4 shows the static environment used for sensitivity analysis with challenging obstacle configurations. The map is navigated by an agent from the red star with a constant north-east vector. Region A is a 1.6 m wide doorway, B is a 4 m wide tunnel with a perturbation and C is a 0.4x0.4 m wide obstacle.

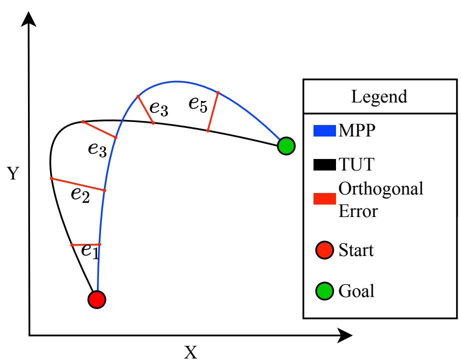


Figure 5.5: How the error between two trajectories is calculated.

Figure 5.5 shows how the MPP is compared to the TUT. At fixed intervals along the MPP, a perpendicular error line is projected until it intersects with the TUT. The euclidean distance of the error line is used to calculate the absolute error, which is averaged to obtain the MAE.

In the simulated experiments, Lam *et. al* used a PID controller tracking a goal location to model pilot input, but this may obscure the effects of some variables (such as oscillation), despite performing more similarly to a human [13]. For this reason, the human input is a constant maximum vector. This is the most simple way in which a pilot would interact with the system, but does not reflect how a higher level controller (be it human or otherwise) would navigate this particular environment [47].

Table 5.1: The default intrinsic and extrinsic simulation parameters used for the sensitivity analysis, based off of data from the real system.

Description	**Default Value**
UAV Radius	0.41 m
UAV Maximum Velocity	4 m/s
UAV Maximum Acceleration	8 m/s^2
LiDAR Scan Rate	10 Hz
LiDAR Angular Resolution	1°
LiDAR Range Noise Variance	0.02 m
Wind Velocity and Direction	0 m/s
Piloted Vector	North-East
Smallest Avoidable Obstacle	0.4 m
Repulsive Threshold Distance	12 m
Velocity Compensation	Krogh

The values in Table 5.1 show variables relevant to the platform dynamics obtained from the system modelling above, such as the maximum velocity and acceleration. The LiDAR data is also informed by the physical platform. LiDAR parameters, and the chosen obstacle and threshold distances are all used to calculate the repulsive gain in the parametrisation methodology given in Chapter 4.

5.2.1 Scan Rate

Previous research does not pay much attention to the effects of the scan rate of rangefinders on APF algorithms. It is important to understand how the choice of sensor may impact real-world performance, which is the aim of this experiment.

The repulsive gain is recomputed in each case based on the given scan rate. Figure 5.6a shows that this higher repulsive gain is not sufficient to ensure no collisions at very low sample rates. At 1 Hz, the scanner is simply not scanning fast enough to respond to obstacles, and collides (as evidenced by the trajectory passing through the wall).

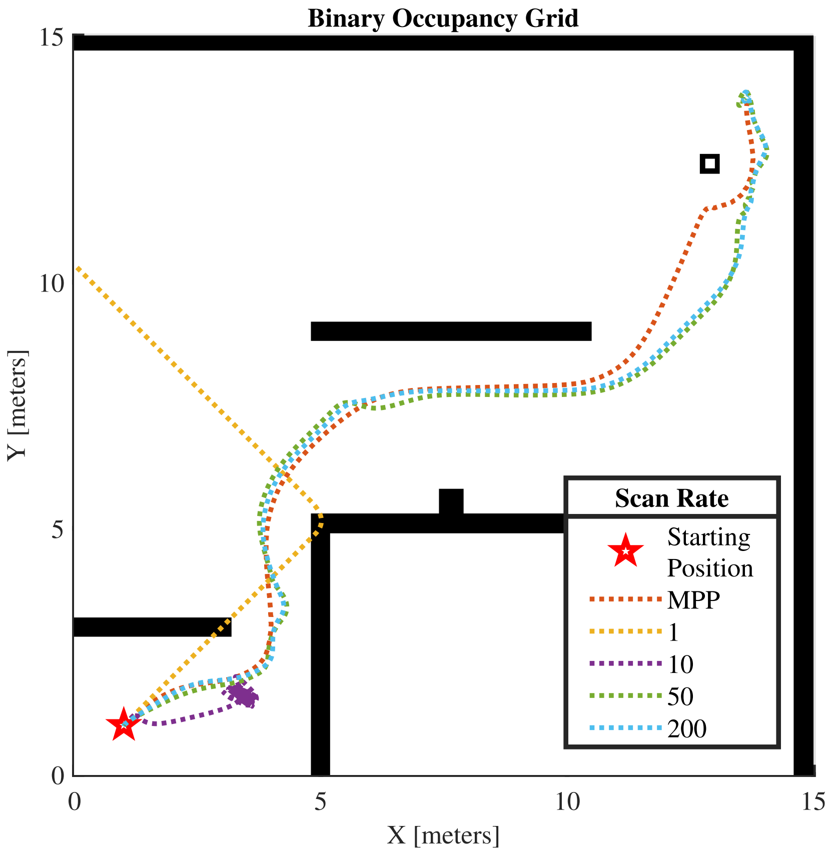
At 10 Hz, the APF algorithm is able to avoid collisions, but the field sensitivity (repulsive gain) is too high to enable passage through region A resulting in the UAV being stuck in a local minima.

The trajectories at the higher sample rates (> 50 Hz) tell a different story, and are able to navigate through the environment smoothly, with little oscillation. They are also the only trajectories that end as far north-east as is possible in this environment. This is because the safety buffer radius decreases sufficiently to allow passage through very restricted regions. The trajectories also deviate less from the MPP with small deviations in regions (< 1 m) A and C.

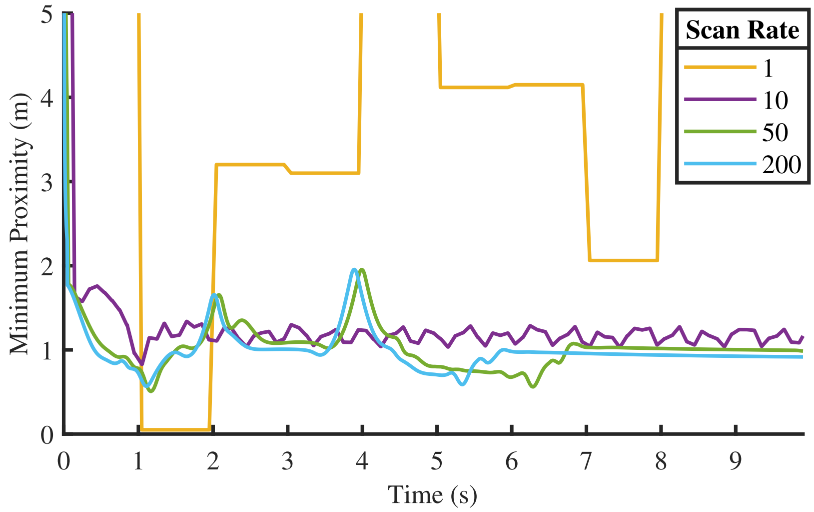
Figure 5.6b shows that as the UAV passes constrained regions with obstacle perturbations or minimally sized obstacles in region A and C, it briefly veers off the MPP. In this state the obstacles are within the safety buffer distance and so the field is not sensitive enough to achieve safe-flight in all situations, even with very high sample rates. The other default parameters are too liberal.

Table 5.2 provides a quantitative view of these observations, with the lower frequencies showing a higher total repulsive force due to increased oscillation amplitude and frequency. The table also reflects changes in the repulsive gain from the proposed parametrisation method. This is useful to compare the "field sensitivity" across different experiments.

In general as the scan rate increases, safer, smoother trajectories are generated, however the improved performance shows diminishing returns. The experiments also reveal that the parametrisation methodology provides reasonable estimates for the repulsive gains.



(a) The trajectories of a UAV with different LiDAR sampling rates, given in Hz.



(b) The closest obstacle to the UAV as recorded by the LiDAR, with different sampling rates, given in Hz.

Figure 5.6: The trajectories (a) and minimal proximities (b) during the scan-rate experiments, displaying how APF performance is improved with a higher scan rate.

Table 5.2: The computed Safety Buffer (r_{buffer}) and Repulsive Gain (η) for different sampling rates using the proposed method, as well as the closest obstacle, MAE and total repulsive force in each experiment.

Scan Rate [Hz]	$\mathbf{r_{buffer}}$ **[m]**	$\boldsymbol{\eta}$	$\boldsymbol{\Sigma} F_{rep}$	**Closest Obstacle [m]**	**MAE [m]**
1	4.62	6.9588	9.36e3	0	0.55
3	1.96	0.4175	9.76e3	0.73	1.64
10	1.02	0.0507	3.59e3	0.82	0.47
50	0.70	0.0175	2.44e3	0.48	0.25
200	0.64	0.0131	2.30e3	0.55	0.18

Analysing Table 5.2 reveals that the size of the buffer decreases with the increasing scan rate, resulting in a less repulsive field through the parametrisation. This in turn decreases the repulsive force, while the increased sample rate results in a lower MAE as the trajectory more closely tracks the MPP. It is important to note that the closest obstacle measurement is less than the safety buffer distance in every experiment. For the 1 Hz experiment, the closest obstacle is 0 m away, indicated that a collision must have occurred.

To characterise the lower bound of the scan rate consider that in the worst case the UAV is travelling towards an obstacle at the maximum velocity of 4 m/s. It would take 0.5 s to come to a stop at a deceleration of 8 m/s^2. In order to adequately sense and provide the repulsive command, the scan period (T_s) of the collision avoidance sensor should be less than 0.5 s, governed by this equation:

$$T_s < \frac{v_{max}}{a_{max}} \tag{5.2}$$

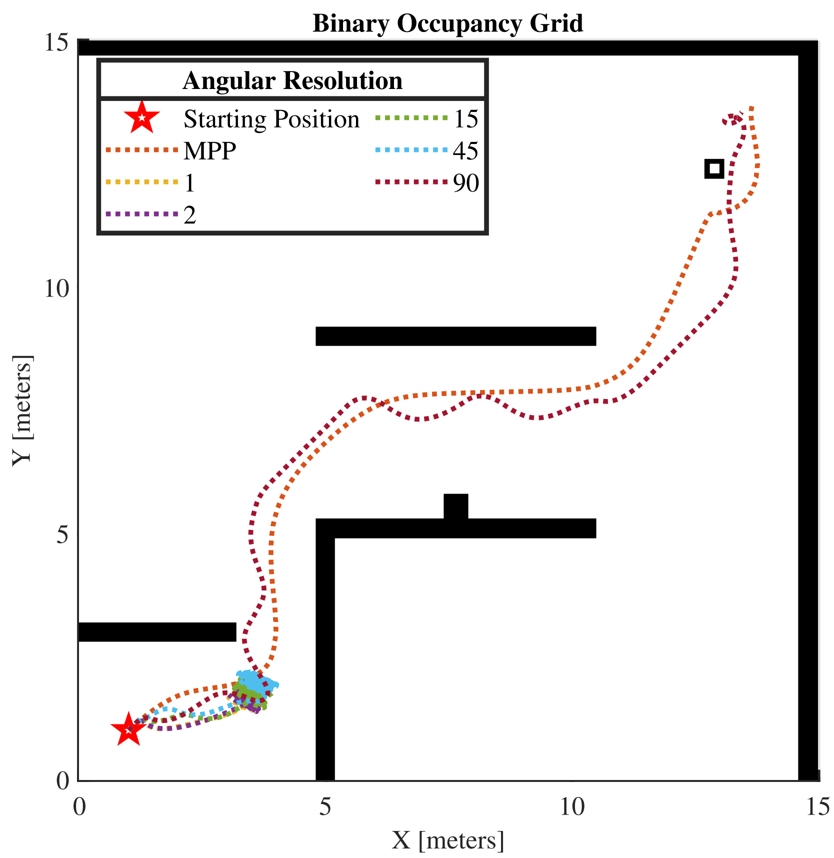
5.2.2 Angular Resolution

Rotating LiDAR technology shows a trade-off between scan rate and the angular resolution of the device. Smaller angular resolutions are achieved if the LiDAR is spinning slower, but a slow sample rate has been shown to be detrimental to performance. In fixed-sensor packages (not rotating), such as an ultrasonic array, the angular resolution is a limited by price, power and mass of each sensor, since many of them will be required to achieve a smaller angular resolution. This experiment is designed to determine how the angular resolution affects performance, to allow for better decisions around sensor choice.

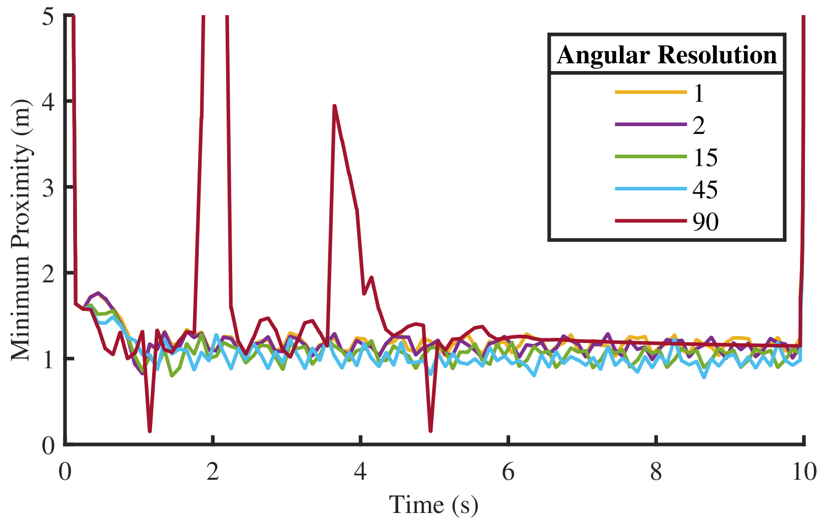
Once again, the repulsive gain varies with the chosen angular resolution, however, the safety buffer in this set of experiments is fixed at 1.02 m due to the fixed sample rate. This means it is impossible to pass through region A without violating the definition of safe flight.

In Figure 5.7a, the 1°, 2°, 15° and 45° runs show similar trajectories, despite a difference in the number of repulsive points. In each of these modes the UAV is stuck in a local minima, and ends up oscillating in region A without passing through. This is expected given the results of the scan-rate experiments, where at 10 Hz the system settled in a local minima. Table 5.3 also reveals that none of the experiments met the safe flight criteria, however this only occurred for a brief moment in the beginning of the flight as the vehicle approached region A after building up some velocity.

Increasing the angular resolution to 90° results in range readings in the 4 cardinal directions only, with the same repulsive gain. It is expected that obstacles between the sensors (North-West for example), would not be detected and not repel the UAV. This hypothesis is correct, as shown in Figure 5.7a, where the trajectory passes through region C in very-near proximity to the obstacle. Despite being the most North-Easterly trajectory, it does not meet the prime criteria of avoiding collisions.



(a) The trajectories of a UAV with different LiDAR angular resolutions, given in degrees.



(b) The closest obstacle to the UAV as recorded by the LiDAR with different angular resolutions.

Figure 5.7: The trajectories (a) and minimal proximities (b) during the angular resolution experiments.

This experiment highlights that the APF algorithm performs similarly for small angles ($< 15°$), since the parametrisation method compensates for the differing resolutions, as evidenced in Table 5.3. There is little to gain in terms of APF performance from selecting a LiDAR with extremely low (< 0.5) angular resolution, except perhaps the detection of small obstacles. However, if the angular resolution is too large, an obstacle may be undetected. If opting for a fixed-sensor package, the APF algorithm can work with a low number of sensors, which should then have a wide enough beam-pattern to cover the full 360° area around the UAV.

To avoid situations where an obstacle of minimum size is undetected by a LiDAR, the choice of angular resolution should be such that at least two LiDAR points can detect a minimally sized obstacle at the edge of the buffer. This is already expressed in Equation 4.3, and went un-noticed in the development of the parametrisation method, but requires some tweaking, since n_0 is now fixed at 2. For an obstacle width of 0.4 m, and a safety buffer distance of 1.02 m, the angular resolution of this system should be less than 11.46° to meet the above criteria. This value is computed by re-arranging Equation 4.3 to obtain the following limitation:

$$\theta_{res} < arctan(\frac{w}{2d}) \tag{5.3}$$

Table 5.3: The computed number of sensors and Repulsive Gain (η) for different sampling rates using the proposed method, as well as the closest obstacle, MAE and total repulsive force in each experiment.

Resolution [°]	**# Sensors**	$\boldsymbol{\eta}$	$\boldsymbol{\Sigma F_{rep}}$	**Closest Obstacle [m]**	**MAE [m]**
1	360	0.0507	3.59e3	0.83	0.5
2	180	0.0552	3.61e3	0.83	0.5
15	24	0.429	4.15e3	0.80	0.4
45	8	1.0553	4.53e3	0.78	0.3
90	4	1.0553	2.71e3	0.15	0.4

Table 5.3 echoes the analysis above. Notable is the closest obstacle in the 90° experiment, which is smaller than the UAV radius, indicating a collision. The table shows that even across different angular resolutions, the repulsive force, closest obstacle, and MAE change little. The repulsive gain however, changes a lot, indicating that the parametrisation method works well across multiple angular resolutions in order to obtain consistent behaviour even with a reduced number of sensors.

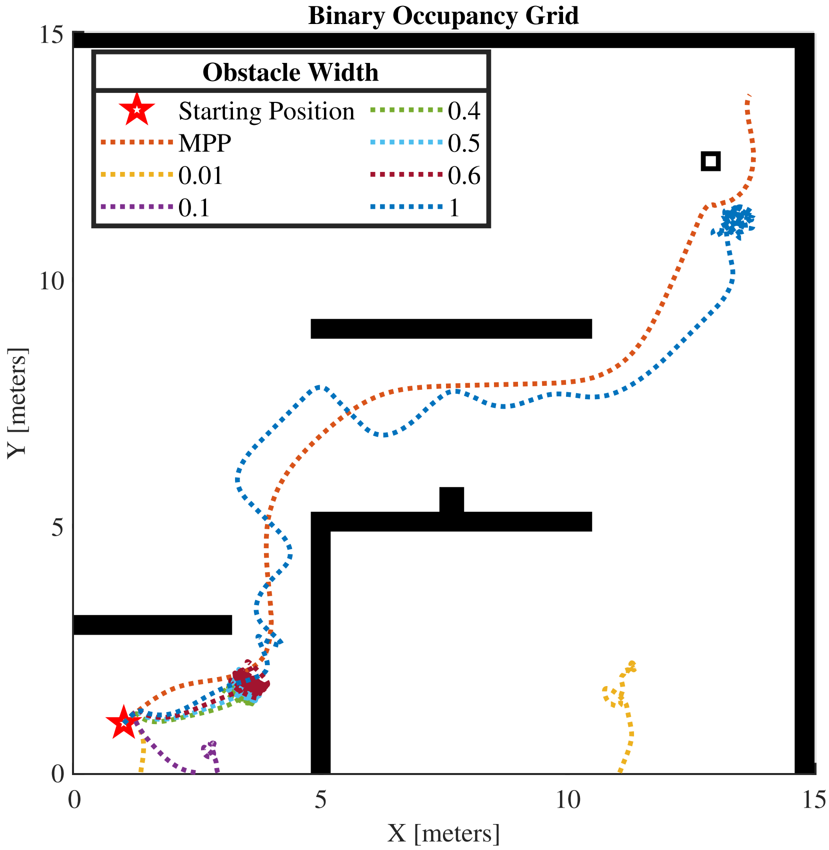
It is also noted that despite each obstacle in the environment being at least 0.4 m wide, the parametrisation does not create a sensitive enough field to ensure safe flight (the safety buffer distance is 1.02 m, while the closest obstacle in every experiment is less than that) with a minimum obstacle width of 0.4 m. However, if the repulsive gain is such that a single obstacle point creates maximum repulsion, the environment becomes too constrained to traverse, leading to impassable sections. The trade-off between controllability and safety is a recurring theme for APFs.

5.2.3 Minimum Obstacle Width

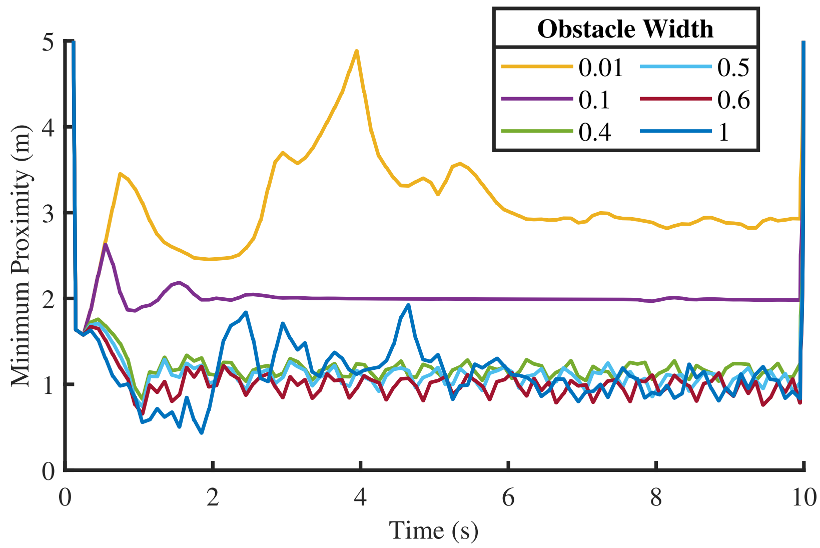
Instead of choosing the repulsive gain directly, the parametrisation dictated that the minimum avoidable obstacle width be chosen. This set of experiments observe how the field sensitivity varies with the changes in this value.

Figure 5.8a shows what is expected for this field, smaller minimum widths result in more sensitive fields, and it is only at 1 m that this environment is traversable. However, the 1.02 m safety buffer makes safe passage in region A and C impossible, which shows that for a broad range of obstacle widths this is maintained.

If the minimum obstacle width is chosen to be really small, as in the 0.01 m and 0.1 m experiments, the field becomes too sensitive and the vehicle is repelled out of the starting area. This behaviour is not desirable, since the operator input is completely disregarded despite the starting region providing enough space for safe flight.



(a) The trajectories of a UAV with different obstacle width parameters, given in meters.



(b) The closest obstacle to the UAV as recorded by the LiDAR with varying obstacle widths.

Figure 5.8: The trajectories (a) and minimal proximities (b) during the obstacle width experiments.

Larger widths (0.4 m to 0.6 m) produce more intended behaviour, all settling at different distances from region A, which is seen in Figure 5.8b. Each of these trajectories oscillate around the local minima in region A without passage. As expected, the proximity to the walls in region A decreases as the obstacle width increases due to the reduced field strength (repulsive gain), but never breach the wall effect buffer.

Table 5.4 and 5.2 together provide insight regarding repulsive gain. At the fixed angular resolution of 1°, repulsive gains smaller and equal to 0.024 allow for traversal of region A, but not safe flight. Repulsive gains more than 0.024 cannot pass through region A, and settles in a local minima before the doorway. The important outcome here is that the parametrisation works to ensure collision avoidance through the computation of repulsive gains across a wide range of obstacle widths. This particular environment however is too constrained in region A for this vehicle to pass through at the default settings. Table 5.4 summarises the sensitivity analysis for the obstacle

Table 5.4: The computed number of incident points and Repulsive Gain (η) for different obstacle widths using the proposed method, as well as the closest obstacle, MAE and total repulsive force in each experiment.

Width [m]	$\mathbf{n_0}$	$\boldsymbol{\eta}$	ΣF_{rep}	**Closest Obstacle [m]**	**MAE [m]**
0.01	1	1.1429	3.60e3	1.57	3.41
0.1	5	0.2288	3.22e3	1.57	1.63
0.4	23	0.1044	3.59e3	0.82	0.83
0.5	27	0.0435	1.58e3	0.74	0.44
0.6	33	0.0361	4.19e3	0.65	0.34
1	53	0.0240	4.72e3	0.43	0.49

width parameter. In the case of 0.01 m obstacle width, the MAE is high, attributed to the high repulsive gain that pushed the UAV out of region A and the visible grid. For obstacle widths 0.1 m-0.6 m, the quantitative results are similar, while the 1 m result shows a 0.43 m closest obstacle, which is a near miss for the 0.41 m wide UAV.

The results show that increasing the obstacle width decreases the flight safety. Increasing the minimum obstacle width to 1 m results in a field sensitivity too low to prevent breaching the wall effect buffer, which is unsafe. Widths smaller than 0.1 m result in a field too sensitive to enable pilot authority.

5.2.4 Relative Velocity

Kroghs relative velocity scaling factor as given in Equations 3.6 and 3.8 provides a powerful extension to the traditional APF [6], but it is not the only approach.

An alternative approach, which will be called Negative-Krogh, is described. Risk of collision increases with velocity, but when moving away from an object, it may be desirable to decrease the force as a means of reducing oscillation. This is done by modifying the original velocity scaling factor and is described in Equation 5.4. As in previous Chapters, the relative velocity is considered, since it is intractable to compute the actual air-velocity of the vehicle and surrounding obstacles.

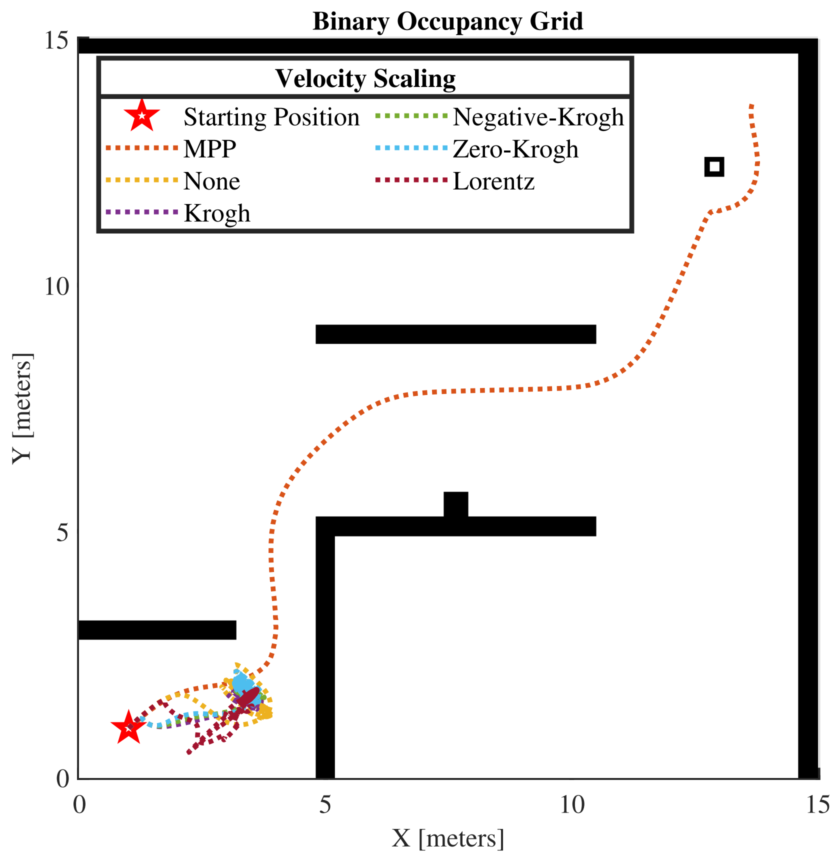
$$\lambda_i = \begin{cases} 1 + \frac{|\vec{v_i^t}|}{v_{max}} & \text{if } \vec{v_i^t} \leq 0; \\ 1 - \frac{|\vec{v_i^t}|}{v_{max}} & \text{if } \vec{v_i^t} > 0; \end{cases} \tag{5.4}$$

A separate take on this is to completely ignore obstacles that are moving away, in a mode referred to as Zero-Krogh. This velocity scaling negates the repulsive effects of obstacles moving away completely, as described in Equation 5.5.

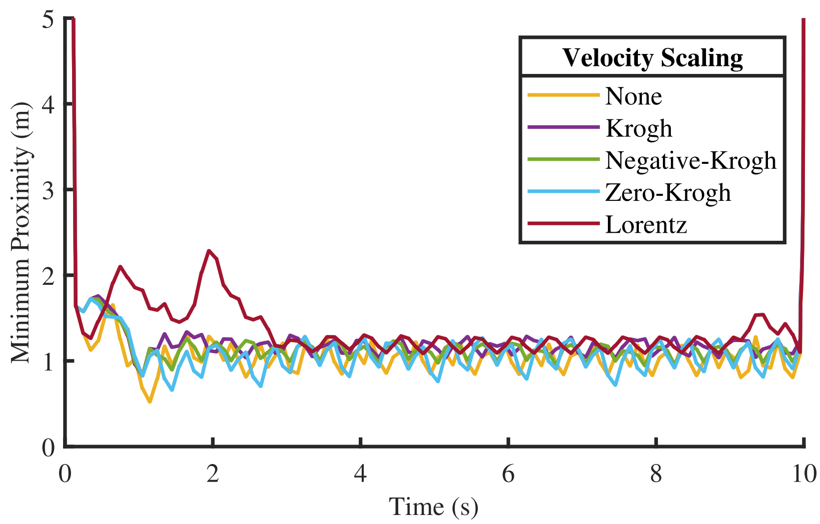
$$\lambda_i = \begin{cases} 1 + \frac{|\vec{v_i^t}|}{v_{max}} & \text{if } \vec{v_i^t} \leq 0; \\ 0 & \text{if } \vec{v_i^t} > 0; \end{cases} \tag{5.5}$$

Finally, it may be desirable to only compensate for obstacles that are approaching at close to the maximum velocity, while remaining relaxed at low velocities. This relationship is alike to the Lorentz Factor, which rapidly increases as the relative velocity of a particle approaches the speed of light. A similar formulation is considered in Equation 5.6 to obtain an exponentially stronger repulsion as the UAV approaches it's maximum speed.

$$\lambda_i = \begin{cases} \frac{1}{\sqrt{1 - (\frac{\vec{v_i^t}}{v_{max}})^2}} & \text{if } \vec{v_i^t} \leq 0; \\ 1 & \text{if } \vec{v_i^t} > 0; \end{cases} \tag{5.6}$$



(a) The trajectories of a UAV with different velocity scaling techniques.



(b) The minimum proximity as recorded by the 10 Hz LiDAR placed on the UAV.

The Negative-Krogh compensation performs very similarly to the standard, but the Zero-Krogh mode, although providing smooth trajectories, does collide after passing through region A since it is insufficiently repelled by the wall. Removing repulsion from obstacles that are relatively moving away is not a viable strategy.

The Lorentz compensation model does seem to work, but the non-linear nature results in a fairly complicated trajectory, especially when caught in a local minima. Testing the Lorentz compensation with less sensitive fields (where the trajectory is not stuck), reveals that the UAV traverses the course at a much slower ground speed, which is undesirable.

Table 5.5: The Repulsive Gain (η) for different velocity compensation methods, as well as the closest obstacle, MAE and total repulsive force in each experiment.

Velocity Compensation	ΣF_{rep}	**Closest Obstacle [m]**	**MAE [m]**
None	4.59e3	0.52	0.29
Krogh	3.59e3	0.82	0.51
Negative-Krogh	4.01e3	0.82	0.46
Zero-Krogh	5.38e3	0.66	0.43
Lorentz	1.55e4	1.09	0.60

Table 5.5 summarises experiments using no velocity compensation, the standard Krogh velocity compensation, and the 3 alternatives described above. Each experiment used the same parametrisation and settings as previous experiments. Quantitatively, the standard Krogh formulation performs best, showing the lowest total repulsive force in Table 5.5, which matches the observations in Figure 5.9a. Without any velocity compensation, the APF algorithm oscillates far more than any others. The higher amplitude of these oscillations is noticeable in Figure 5.9b.

5.3 Conclusion

A simulation environment is designed in order to perform a sensitivity analysis of APF parameters, as well as to benchmark the parametrisation methodology under many conditions. The metrics used to quantify these effects are the vehicle position, minimal proximity, MAE from the MPP and the total repulsive force, which act as measures of path smoothness and optimality.

A total of 20 experimental configurations are simulated, testing the effects of sample rate, obstacle width, velocity compensation and angular resolution on the performance of the APF collision avoidance algorithm. These experiments reveal:

- The scan rate of range-measurement devices should be as high as possible. The lower bound is given as a relationship between the maximum deceleration and velocity in Equation 5.2. An increased sample rate results in a smaller, less sensitive field (when using the proposed parametrisation method) which is better suited to constrained environments. Low sample rates are shown to be responsible for undamped oscillation, and in the worst cases, collision despite the increased repulsive sensitivity.
- The angular resolution of the range-measurement device does not play a major role in APF performance. The APF performs very similarly under the condition $\theta_{res} < arctan(\frac{w}{2d})$ when the parametrisation method is used. Larger angles run the risk of an obstacle not being detected, and therefore not avoided.
- The obstacle width parameter used to control field sensitivity behaves as expected: an obstacle of width w at the minimum distance d is maximally repelled. Increasing the minimum obstacle width reduces field sensitivity, enabling more constrained flight. The caveat is that a reduced field sensitivity also corresponds to a reduced risk sensitivity, resulting in more unsafe conditions. However, the minimal proximity does not breach the wall-effect buffer unless the obstacle width parameter is 1 m or more.
- The relative-velocity scaling developed by Krogh (the GPF) greatly outperforms the general APF in terms of obstacle proximity. Other formulations of Krogh's velocity scaling perform similarly but a Lorentz Factor scaling exhibits large oscillations.

- Unsafe flight conditions occur often in this environment, indicating that the parametrisation is unable to prevent unsafe flight always. However, these situations occur rarely, and for very brief moments before the effects of repulsion force the UAV away from the obstacle. Under the ideal conditions stipulated above, one can expect to never breach the wall effect buffer, which is enough to prevent collisions.

Chapter 6

Test Apparatus and Environment

For the purposes of realising and validating the experimental results, a real quadcopter is to be used. The platform in question is a GDS-Canary, which is equipped for research and development in underground mines. Outfitted with an ArduCopter flight controller and Raspberry Pi, the quadcopter can be controlled via standard 2.4 GHz radio or WiFi, accepting attitude inputs from the custom software.

The software is built on the ROS middleware and is split across two core modules. `Bringup` is a driver class, interfacing with ArduCopter, while also providing a multiplexor for various inputs. `Potential Field` is responsible for collision avoidance using the APF algorithm.

6.1 Hardware

The hardware configuration for the physical realisation of experiments is discussed in this section. On a high level there are two primary parts, the GCS and the quadcopter itself. Each are described in Figure 6.1 which details on a high level the minimal parts required to allow collision avoidance and teleoperation of a UAV in an underground environment.

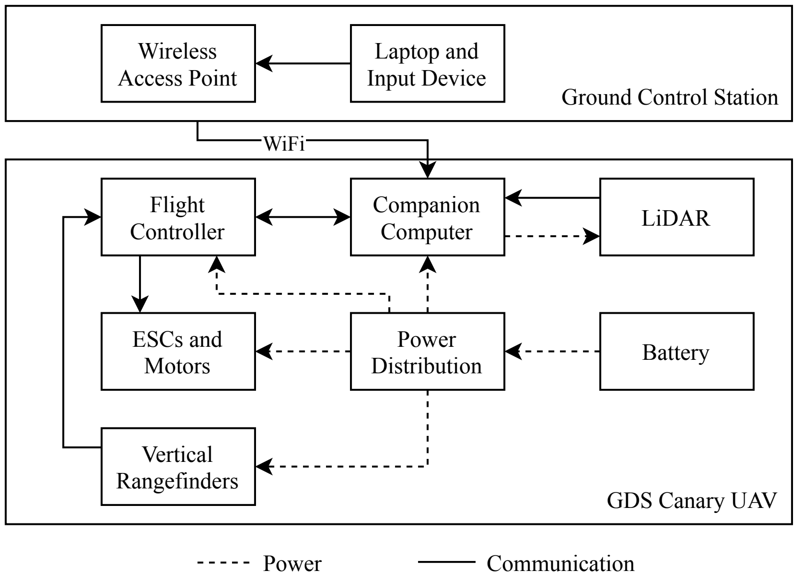


Figure 6.1: High level connection diagram of the GDS Canary platform used in this research.

Figure 6.1 shows two distinct platforms, the GCS and UAV. The GCS is simply comprised of a computer and wireless connection, allowing pilot input, visual feedback, remote control and logging. Central to the UAV connection diagram are the companion computer, which processes all LiDAR information for the APF algorithm, and communicates with the FC; and the Flight Controller (FC), which provides an interface to motors, altitude control, and attitude control.

Each of the major components, their function and features are detailed below.

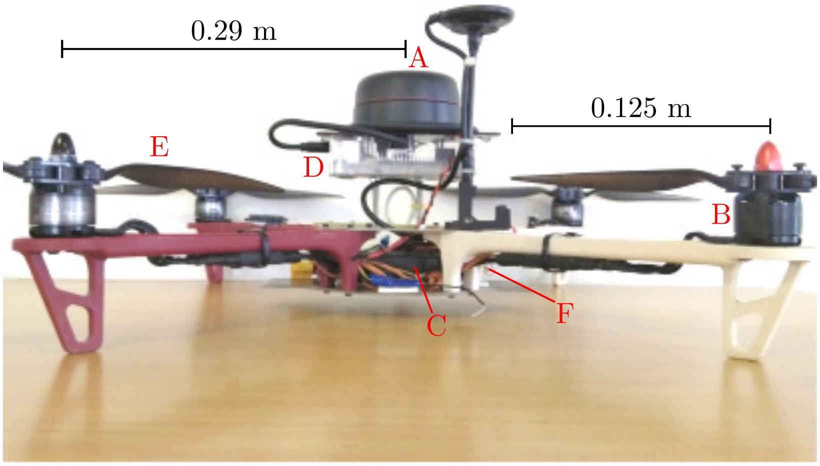


Figure 6.2: The GDS Canary hardware stack with (A) RPLiDAR A2M8, (B) motors, (C) FC, (D) Raspberry Pi 3B+, (E) 25 cm propellers and (F) downwards facing rangefinder. The image also shows dimensions of the center-to-tip dimension (29 cm) and propellers(12.5 cm) used to calculate the safety buffer size.

6.1.1 Platform

The testing platform is a GDS-Canary quadcopter, which has a wheel-base of 0.45 m and a mass of 2 kg. Being a research and development platform, it is highly configurable, and in the current configuration is fitted with parts as shown in Figure 6.2. Not shown is the battery, which is a 3-cell 2500 mAh LiPo, granting around 5 minutes of flight time as determined while testing with the complete payload.

The off-the-shelf FC is responsible for attitude stability, and would normally be used for manual consumer flight over a 2.4 GHz radio. Here, the UAV attitude controls are specified over a serial connection from the Raspberry Pi, which processes the collision avoidance logic, and communicates using built-in 2.4 GHz WiFi. This works in tandem with the RPLiDAR A2M8 laser scanner which is mounted on top of the platform, and provides a 360 scan of the area around the platform.

Up and down range finders fitted to the platform enable hover control and ceiling collision avoidance. Vertically constrained flight is not tested in this research, so the upwards facing sensor is disabled. However, the downwards facing range finder is used by the FC to maintain altitude.

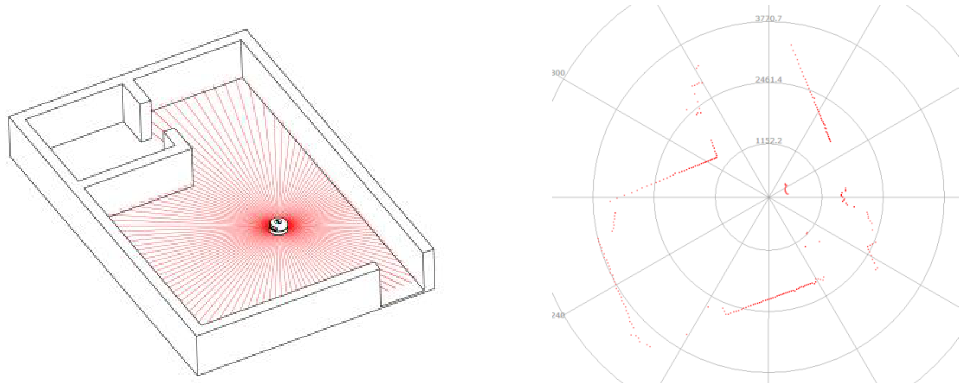
6.1.2 Laser Scanner

The key RPLiDAR A2M8 features are listed in Table 6.1. In Chapter 5 the default scan rate is 10 Hz, but it is noted that a higher sample rate is desired, as long as the angular resolution is not too large. The RPLiDAR can spin as fast as 20 Hz, while the laser samples at 4000 Hz. This means that 200 range samples are taken every revolution, with an angular resolution of 1.8°. The decreased angular resolution is not expected to impact APF behaviour, as explored in Chapter 5.

Table 6.1: Key features of the RPLiDAR A2M8 that make it suitable for the research.

Item	Unit	Min	Typical	Max
Range	m	0.15	-	8
Laser Sample Rate	Hz	2000	4000	4100
Scan Rate	Hz	5	10	20
Mass	g	-	190	-
Power Consumption	W	1	2.25	3

The datasheet does not provide a measure of precision for the laser, but through a static test with 250 samples, and assuming a guassian distribution, the standard deviation is estimated at 0.02 m. This noise estimate is 0.16% of the full range of the LiDAR, and shouldn't have much effect on the performance of the APF algorithm.



(a) An illustration of the RPLiDAR in an example environment, showing the full field of view.

(b) Real data from the RPLiDAR in an environment similar to (a) from a top view. This data is presented in the body frame.

Figure 6.3: The RPLiDAR range data represents a horizontal "slice" of the environment in (a), which can be represented by the top view in (b).

The LiDAR is rigidly mounted above the centre of the UAV, and scans continuously in the x-y plane of the body frame. Figure 6.3b shows an example of what this data would look like for an indoor environment.

6.1.3 Teleoperation

A joystick connected to the GCS allows for pilot input through an intuitive interface. Keyboard inputs are very binary compared to the fine control offered by a joystick, and addition buttons can be used for arming and disarming, as well as an emergency switch.

The WiFi capabilities present an opportunity for completely remote teleoperation, but latency is a concern, and is not tested in this research. Providing pilot control through the WiFi connection also removes the added interference of 2.4 GHz radios that is noted by [3], increasing the available bandwidth. This enables more telemetry to be relayed back to the GCS, which can be used for visual feedback, either through a top view as in Figure 6.3b, or an onboard camera.

6.2 Software

All computation takes place on the Raspberry Pi onboard, reducing latency and improving robustness against network failures over the remote setup in [3]. The software performing the task of collision avoidance is designed to interface with the ROS middle-ware, which provides many of the networking and time-keeping tasks involved with distributed robotics. The software is based around two control loops, which are detailed below.

6.2.1 Control Overview

The realisation of the APF algorithm as a closed-loop control system makes real-time implementation a reality. These control loops build on the levels of autonomy discussed in Chapter 2.

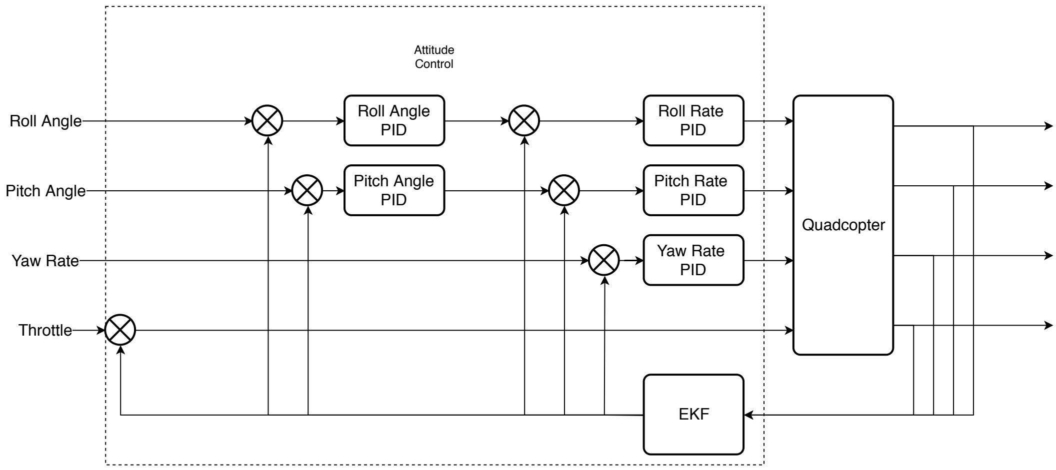


Figure 6.4: The attitude control loop with nested PID controllers commonly used for commercial FCs.

The inner control loop is a 400 Hz attitude stabilisation system, which provides "user-friendly" UAV readiness, responding rapidly to changes in roll, pitch, yaw and height by actuating the motors. Smoothed tate estimation is done using an Extended Kalman Filter (EKF) with data from an IMU, rangefinders and barometer, and nested PID controllers stabilise the platform. Figure 6.4 provides a block diagram of this setup. The inner loop is a popular research topic, and is comprehensively understood, so much so that many commercial and open source implementations exist, such as ArduCopter, which is a widely used autopilot that is feature rich and well tested. ArduCopter is discussed further in the following sections.

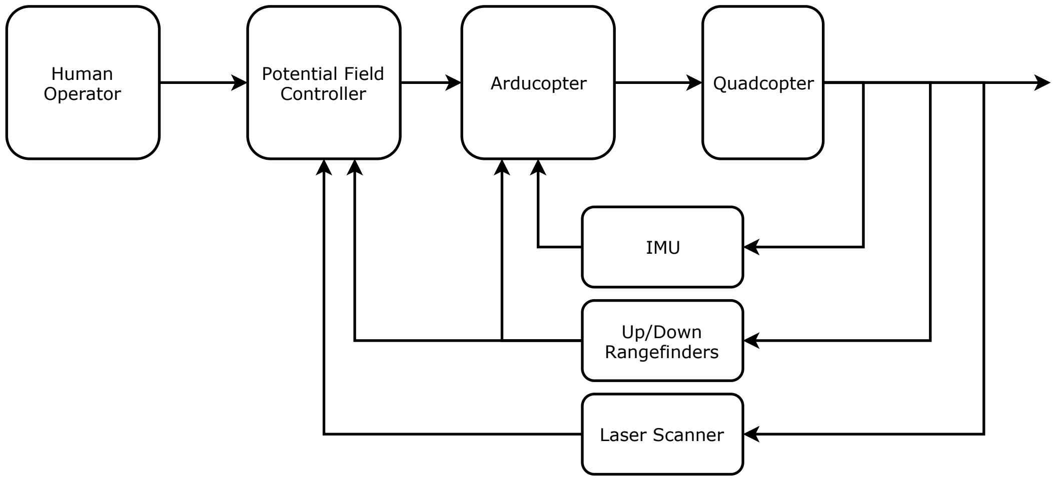


Figure 6.5: The outer control loop, which extends the level of autonomy available to the UAV through additional sensing and control. The inner loop operates independently on the FC, while the outer loop runs on the companion computer, receiving inputs from the pilot.

The outer control loop runs at 20 Hz, the rate of the LiDAR, and provides inputs to the inner control loop as in Figure 6.5. The outer loop is responsible for collision avoidance using the APF algorithm. This is designed such that another, slower loop could provide direction setpoints, which in the case of this research is the pilot via teleoperation. This outer loop presents challenges such as communication, device drivers, teleoperation and portability, which are all addressed by the ROS architecture, discussed next.

6.2.2 ROS

ROS is not an operating system, but provides the services that would be expected from one, “including hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management. It also provides tools and libraries for obtaining, building, writing, and running code across multiple computers.” [48].

For these purposes, it acts as a middle-ware for code executed at a high level to interact with devices at the hardware level. This is well suited to the research in question, and comes with many tools to aid development and debugging, such as `RVIZ`, which provides real-time data visualisation. ROS also integrates with `MATLAB`, which is useful for data capture and analysis.

Transforms in ROS

Also provided is the `tf` package, which is able to track co-ordinate frames over time in a tree-like structure. This is important for static transformations, such as the location of the LiDAR frame relative to the body frame; and dynamic transformations such as the body frame as it translates in the map frame. Figure 6.6 shows the reference frames and how they are connected.

The map frame provides a long-term reference of the UAV trajectory as it navigates the environment. This is the world-fixed frame, and the origin is found at the UAV starting position. This is achieved using the `gmapping`, which is available on ROS, and provides a fully fledged SLAM solution based on LiDAR information and odometry [40].

Odometry is computed by the `laser_scan_matcher`, also available as a ROS package, linking the odometry and base stabilised frames [39]. The base stabilised frame is ridigly attached to the platform, but does rotate as the UAV rolls, pitches and yaws. These rotations are captured by the base link frame, as recorded by the EKF within the FC. Finally, the LiDAR frame is a static transformation noting the physical placement on the platform.

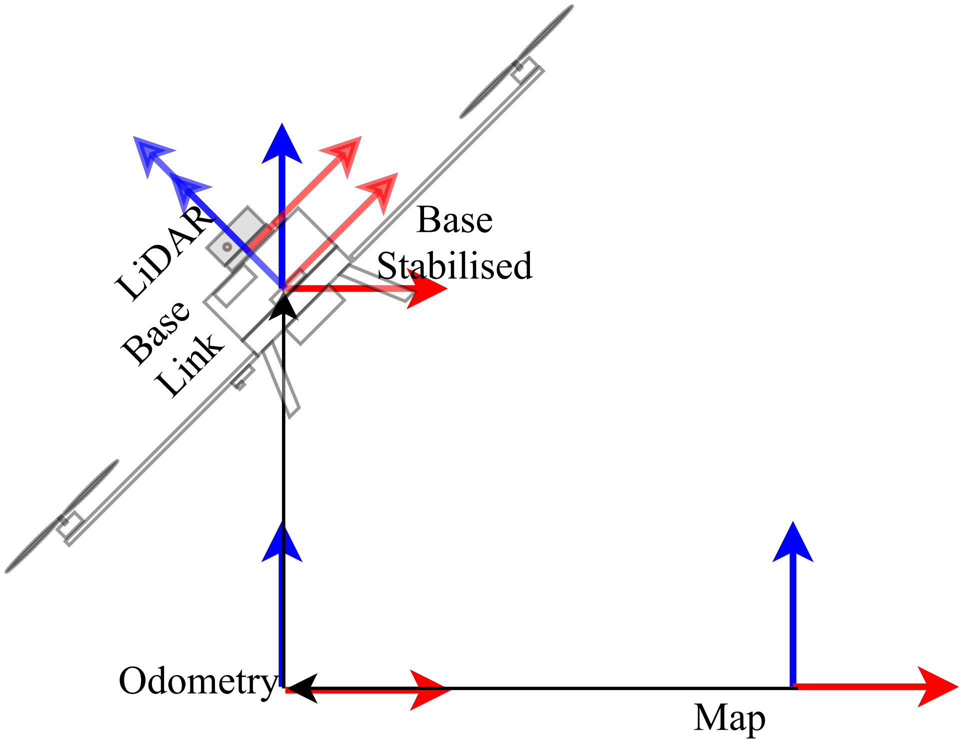


Figure 6.6: A 2D side-view of the frames of interest for the UAV, also known as a transform tree.

Similar to the inertial frames introduced in the UAV working principle of Chapter 2, Figure 6.6 presents the transforms used by ROS. The map and odometry frame are analogous to **N**, the base stabilised is analogous to **I**, and the base link to **B**.

ROS Nodes

ROS nodes are responsible for computation, and communicate with each other by streaming topics via preset/custom message types. A node graph, like that in Figure 6.7, shows how information flows through various node. In this configuration, a number of existing packages provide nodes, which improves speeds up development. However, for implementing the APF algorithm with teleoperation, some new nodes are required. Two ROS packages are created, `canary_bringup` and `canary_potential_field`, which contain all the relevant nodes.

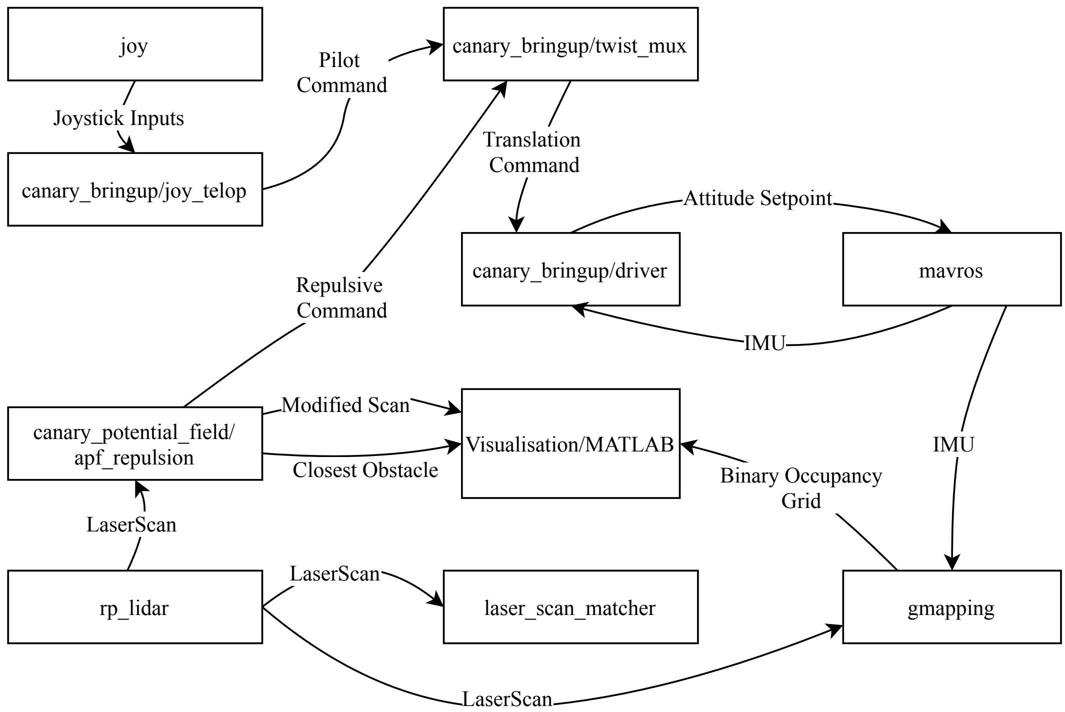


Figure 6.7: The node graph of all the ROS software showing which messages are being passed around to each node.

The bringup package is designed to allow free teleoperation of the UAV without collision avoidance enabled. The first node is the `joy_teleop` node, which is responsible for verifying pilot inputs from a handheld joystick or gamepad. Horizontal movement is input via a thumbstick and vertical velocity setpoints via up/down buttons. Yaw-rate is controlled via a second thumbstick, and is passed through as an angle to the driver. These are published as a `geometry_msgs/Twist` message at 25 Hz. This node also handles emergency stop, take-off and land commands.

The `twist_mux` node subscribes to these messages, as well as a `geometry_msgs/Twist` from the collision avoidance nodes. The messages are summed to get the total translation vector, which is published at a rate of 20 Hz.

The `driver` node subscribes to this translation command, converts it to a Quaternion, and publishes an attitude setpoint to be consumed by the `mavros` node. The maximum attitude angle that is allowed is 8°.

The potential field package contains only one node, `apf_repulsion`, which subscribes to the `sensor_msgs/LaserScan` message from the RPLiDAR. Each point of scan is processed by the APF algorithm, producing a commanded linear translation vector, which is published as a `geometry_msgs/Twist` message. This is the core collision avoidance processing node.

The APF node also publishes some metric data such as the closest obstacle in each scan, and a modified scan in which the intensity or "color" of each point has been adjusted to represent the magnitude of its relative repulsion (for visualisation purposes). All of these messages are published as fast as the LiDAR scan rate, which is 20 Hz.

APF parameters including repulsive gain, maximum repulsive distance, vehicle center-to-tip dimension and the safety buffer distance are modified using the ROS parameter system. This means they must be pre-computed or chosen, as in the previous chapter, using the proposed parametrisation method.

ArduCopter

ArduCopter is a controller from the ArduPilot package, designed for autonomous UAV flight. It provides the stable inner loop, and runs on many FCs. It is one of many commercial and open source avionics systems, but was chosen for it's maturity and ROS support. Specifically, it supports the `mavros` package, which provides a communication link between the FC and the ROS nodes running on the Raspberry Pi.

6.3 Testing Environment

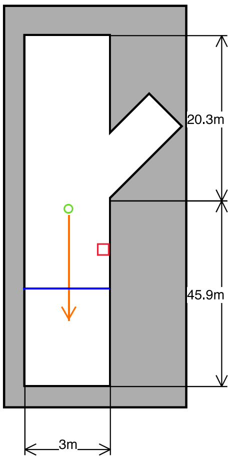
This section discusses the testing environment. The first experiments require a large open area in order to study the effects of only the obstacle configuration under test. The tunnel is not appropriate since the walls will impact the APF algorithm due to their proximity.

Figure 6.8: The atrium of the Chamber of Mines building with (A) the GDS Canary UAV, (B) the GCS, (C) WiFi access point and (D) the obstacle configuration.

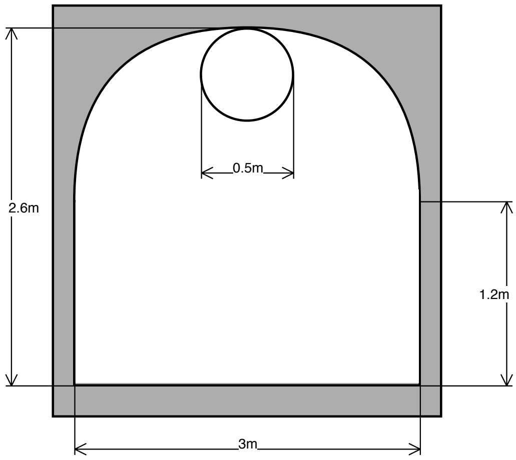
The largest available indoor area for this is the atrium of the Chamber of Mines Building, which provides an obstacle-free area in a radius of 10 m, pictured and annotated in Figure 6.8.

(a) The tunnel region of the WITS Mock Mine. The ventilation pipe can be seen on the ceiling.

(b) The blasting face section of the WITS Mock Mine.



(c) Top view of the mock mine showing the 45.9 m tunnel region, 20.3 m blast face region, and the forked offshoot which is an entrance.



(d) Cross section of the tunnel, with dimensions. The blasting face has the same dimensions.

Figure 6.9: The WITS Mock Mine pictured in (a) and (b), with dimensions given in (c) and (d).

A mine-like tunnel environment is used for the final experimental flights, as in Figure 6.9a and 6.9b. This tunnel features a 3 m wide, 46 m long section, a mine-ventilation system and an uneven rock-face region that emulates a blasting face. Ideally, the test environment would be a vertical shaft, but that is intractable outside of a real mine. The horizontal tunnel, will work as a proof of concept by placing the UAV in constrained mining conditions. Figure 6.9c shows the starting position of the UAV for the tunnel experiments (green circle), the piloted vector (orange), the obstacle (red square), and the finish line (blue).

6.4 Conclusion

The GDS-Canary custom quadcopter is identified as a suitable UAV for experimental flights in order to verify the APF algorithm in practice. The choice of LiDAR scanner provides a large amount of detail regarding the environment geometry which is processed on the Raspberry Pi 3B+ on board to reduce latency. Communication with the ground station is then achieved by connecting to the same WiFi network as the Raspberry Pi on a separate GCS.

The `ROS` middleware provides a simple means to connect UAV hardware to the additional sensors through software. This allows existing packages, namely `mavros`, `rp_lidar`, `laser_scan_matcher`, `gmapping` to be easily integrated with new software developed to enable remote teleoperation using potential fields.

Lastly, a testing environment that provides a large open space (the atrium) and a mine-like tunnel environment have been identified for the experimental setup.

Chapter 7

Experimental Results

The experimental results are presented and discussed here for the experiments proposed in Chapter 4. The primary outcomes of the research regarding the APF algorithm and it's applications to underground teleoperation are presented in detail.

Collision risk and safety in each experiment is assessed by the vehicle proximity to an obstacle, pilot compliance is measured as the error between the total applied command vector and the pilot vector. Oscillations are quantified in terms of the repulsive vector gradient. Together, the metrics are used to assess APF performance as a real time collision avoidance algorithm.

Additionally, the trajectory of the UAV in each environment is presented to give a holistic view of the interactions between the environment and the APF algorithm. This is in contrast to the experiments of Chapter 5, which studied the effects of parameters that are intrinsic to the APF algorithm (such as scan rate, angular resolution and minimum obstacle width) in a fixed environment.

7.1 Performance Metrics

To quantify the effects of obstacle proximity, pilot compliance and oscillation, the following metrics are used:

- Proximity records the closest obstacle point as measured by the LiDAR in every scan. Calculating the mean proximity across the experiments gives a measure of general safety, while the minimum proximity provides a measure of collision resistance.
- Pilot compliance is measured as the difference between the total applied vector and the pilot vector, which happens to be the repulsive vector. The Root-Mean Square (RMS) of the x and y components shows the average pilot compliance ($\mathbf{F_{rep}}$ **RMS**) of UAV during each experiment, with low values corresponding to high pilot compliance.
- Oscillatory behaviour is measured by computing the gradient (in terms of time) of the x and y components of the repulsive command with the RMS providing a summary of the experiment ($\mathbf{\dot{F}_{rep}}$ **RMS**). Since the pilot command will be constant, only the repulsive command will contribute to oscillation in the total applied force. A highly oscillatory repulsive command will result in a higher average gradient.

In addition to the metrics above, Appendix B also contains the proximity, repulsive vector and vector-gradient timeseries data from each experiment.

7.2 Atrium Experiments

The first set of experiments investigate the effects of minimal obstacle configurations, requiring an open area with a flat floor that is free of wind. This setup eliminates external perturbations as far as possible. The atrium is lined with trees along a segment with a diameter of approximately 10 cm, so this is chosen as the ”minimum obstacle width” for parametrisation. The increased field sensitivity as a result of this is also desirable considering the results of Chapter 5 regarding obstacle width and flight safety. This layout is shown in Figure 7.1. The boxes used in the atrium experiments are all 0.4 m wide. A conservative tuning for the real experiments is safer, but does mean that the tuning methodologies ability to avoid a minimally sized obstacle is not tested. Table 7.1 shows the default parameters.

Table 7.1: Default parameters that are used for parametrisation from both the UAV system and the testing environment.

Parameter	**Value**
UAV Radius	0.41m
UAV Maximum Acceleration	8 m/s^2
UAV Maximum Velocity	4 m/s
LiDAR Scan Rate	20 Hz
LiDAR Angular Resolution	1.8°
Smallest Avoidable Obstacle	10 cm
Repulsive Threshold Distance	12 m
Safety Buffer	0.82 m

The values in Table 7.1 show variables relevant to the platform dynamics obtained from the system modelling in Chapter 3, such as the maximum velocity and acceleration. The LiDAR data is also informed by the physical platform as discussed in Chapter 6. LiDAR parameters, and the chosen obstacle and threshold distances are all used to calculate the repulsive gain using the parametrisation methodology given in Chapter 4.

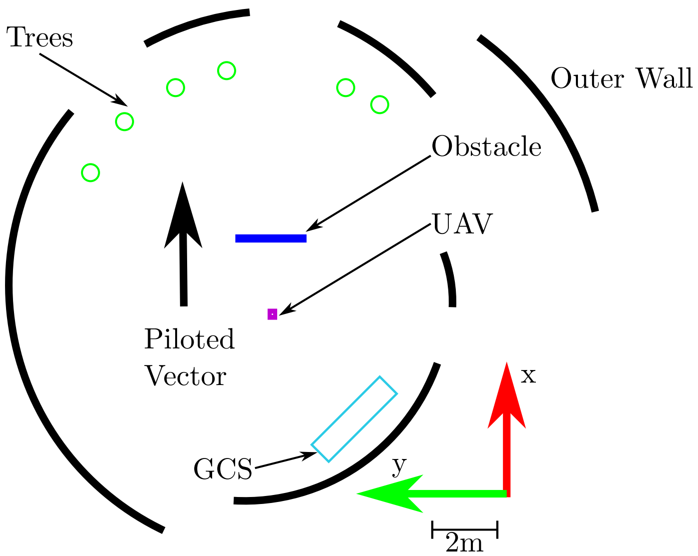


Figure 7.1: The atrium layout used for the single obstacle, obstacle gap and wall experiments.

The experiment layout for the atrium area, identified in Chapter 6, showing the local configuration of obstacles, the UAV and the GCS. Note that the origin of the map axis is located at the start position of the UAV.

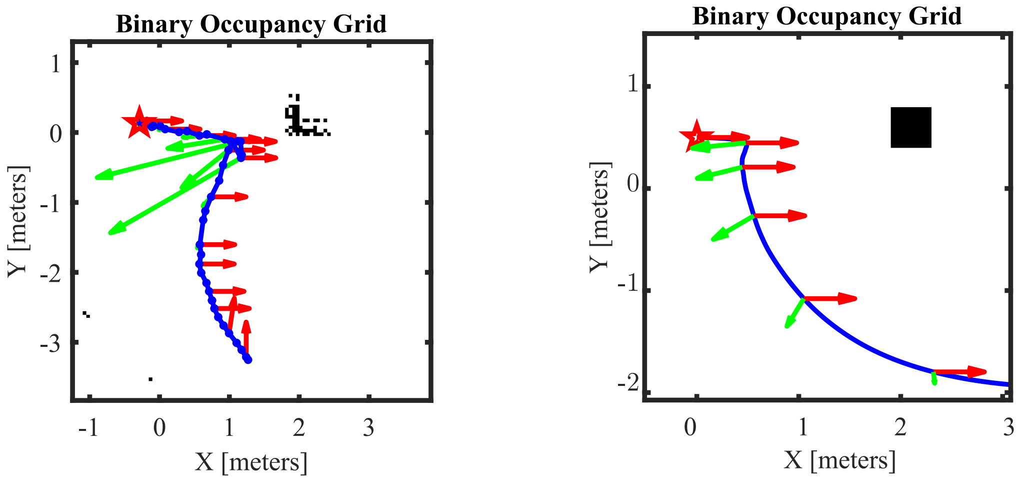
7.2.1 Single Obstacle

In this experiment the UAV is placed 2 m away from a 1.4 m tall obstacle with horizontal dimensions of 0.4 m x 0.4 m. The pilot then takes off to 1 m altitude and commands a forward vector directly into the obstacle. This tests the APF collision avoidance behaviour in the presence of a single obstacle. It is intended that obstacles such as people are acceptably avoided despite the piloted commands.

Table 7.2: Metrics from the single obstacle experiments. The data from the actual experiment is averaged from 100 samples, while the simulation provided 2000 samples.

	Proximity [m]			$\mathbf{F_{rep}}$ **RMS**		$\dot{\mathbf{F}}_{\mathbf{rep}}$ **RMS**	
	Mean	**Std. Dev.**	**Min**	**x**	**y**	**x**	**y**
Actual	2.02	0.62	0.77	0.70	0.31	1.62	0.71
Simulated	6.40	4.10	1.35	0.18	0.09	0.09	0.03

The metrics are summarised in Table 7.2 and the trajectories are shown in Figure 7.2. The primary outcome of this experiment is that the obstacle is avoided in the atrium experiment despite the pilot command. This is indicated by the magnitude of repulsive commands (recall that the total commanded vector has a maximum magnitude of 1) and proximity in Figure B.1 of Appendix B.



(a) (b)

Figure 7.2: The trajectory (in blue) of the UAV (a) in the atrium and (b) in the simulation

Figure 7.2 presents the simulated and actual trajectories for the single obstacle experiment. The pilot is intentionally commanding a vector towards the obstacle for 10 s starting at the red star. The pilot vector is shown as a red arrow, and the repulsive vector is indicated by the green arrow. The 0.4 m wide obstacle is avoided in both cases.

The x-component of the repulsive vector peaks 3 s into the flight cancelling the pilot vector and preventing a collision course. The minimum proximity is also achieved at 3 s where it breaches the safety buffer briefly (resulting in the large repulsive vectors in the negative x-direction).

Experiments in the simulated and real environment perform in a similar manner: the UAV approaches the obstacle, gaining speed; the repulsive force grows sharply as the proximity lessens, cancelling the pilot vector by commanding a leftwards vector; noise in the LiDAR as well as slight offsets in the relative positions of the UAV and obstacle invokes a downwards vector. Finally, the UAV escapes the effects of the repulsive field yielding instead to the pilot vector once again.

The same behaviour is observed in Figure 7.2 for both the actual and simulated trajectories. The simulated experiments display a greater degree of safety and pilot compliance than the actual experiments. Oscillations are also less prevalent in simulations, indicated by the smaller $\dot{F_{rep}}$ RMS value in Table 7.2. The repulsive vector changes more gradually, observed in Figure B.1 and B.2.

The disparity between the simulated and actual experiment is observed. The repulsive vector does cancel the pilot vector as expected. However, momentum carries the UAV closer, causing the repulsive vector to rise rapidly away from the obstacle[1]. In the simulation, the UAV stops faster upon approach, and therefore does not get so strongly repelled.

Note also that in the simulated environment, the repulsive vector cancels the pilot vector at 1.35 m away despite the parametrisation dictating a maximum repulsive vector at 0.82 m. However, the parametrisation is designed to avoid a 0.1 m wide obstacle and as such the UAV avoids a bigger obstacle (0.4 m) with a greater berth.

[1]The simulation does include momentum, but the velocity response is much faster than the real UAV. There is a section dedicated to this disparity further on

7.2.2 Obstacle Gap

In this set of experiments the UAV is placed 2 m away from a set of obstacles with a gap between them. The pilot then takes off to 1 m altitude and commands a forward vector through the gap for 10 s. Each experiment is performed once.

The APF algorithm notoriously settles into a local minima when attempting to pass between two obstacles [32]. The problem is not solved in this research, but it is desirable to investigate what size gaps are passable given the various APF parameters. In these experiments, the gap starts off 1 m wide, and is increased in increments of 0.4 m until the UAV passes through.

Table 7.3 provides an overview of the results for these experiments, while Figures 7.3 and 7.4 display the flight trajectories. The data shows that the actual UAV passes through a 2.2 m wide gap, and collisions are avoided, which are the primary findings.

The data in Table 7.3 shows that the mean proximity tends to decrease as the gap widens, up until 2.2 m where the UAV is able to pass through safely. The repulsive x command magnitude closely matches the pilot command (0.5) since they cancel out, leading to a low pilot compliance. The high repulsive command gradient is indicative of oscillatory behaviour in the actual experiments. The statistics over each actual experiment are from 100 samples, while the simulations produced 2000 samples.

Table 7.3: Summarised metrics of proximity, pilot compliance and oscillatory commands for the gap experiments.

	Proximity [m]			$\mathbf{F_{rep}}$ **RMS**		$\dot{\mathbf{F}}_{\mathbf{rep}}$ **RMS**		**Passage**
	Mean	**Std. Dev.**	**Min**	**x**	**y**	**x**	**y**	
Actual 1.0 m	2.28	0.83	0.89	0.87	0.18	1.33	1.30	No
Actual 1.4 m	1.63	0.36	0.99	0.58	0.06	0.76	0.29	No
Actual 1.8 m	2.00	0.86	0.74	0.64	0.36	0.68	1.27	No
Actual 2.2 m[2]	1.23	0.42	0.50	0.59	1.08	2.16	3.36	Yes
Simulated 1.0 m	2.02	0.27	1.95	0.48	3.0e-3	0.06	0.02	No
Simulated 1.4 m	1.96	0.16	1.87	0.48	3.6e-3	0.07	0.03	No
Simulated 1.8 m	1.87	0.20	1.77	0.47	5.0e-3	0.06	0.04	No
Simulated 2.2 m	1.78	0.23	1.65	0.47	7.5e-3	0.07	0.05	No

[2]The total applied vector magnitude is clamped between $[0- > 1]$, however, the repulsive magnitude is not bounded such that the pilot vector can be overridden in critical avoidance situations.

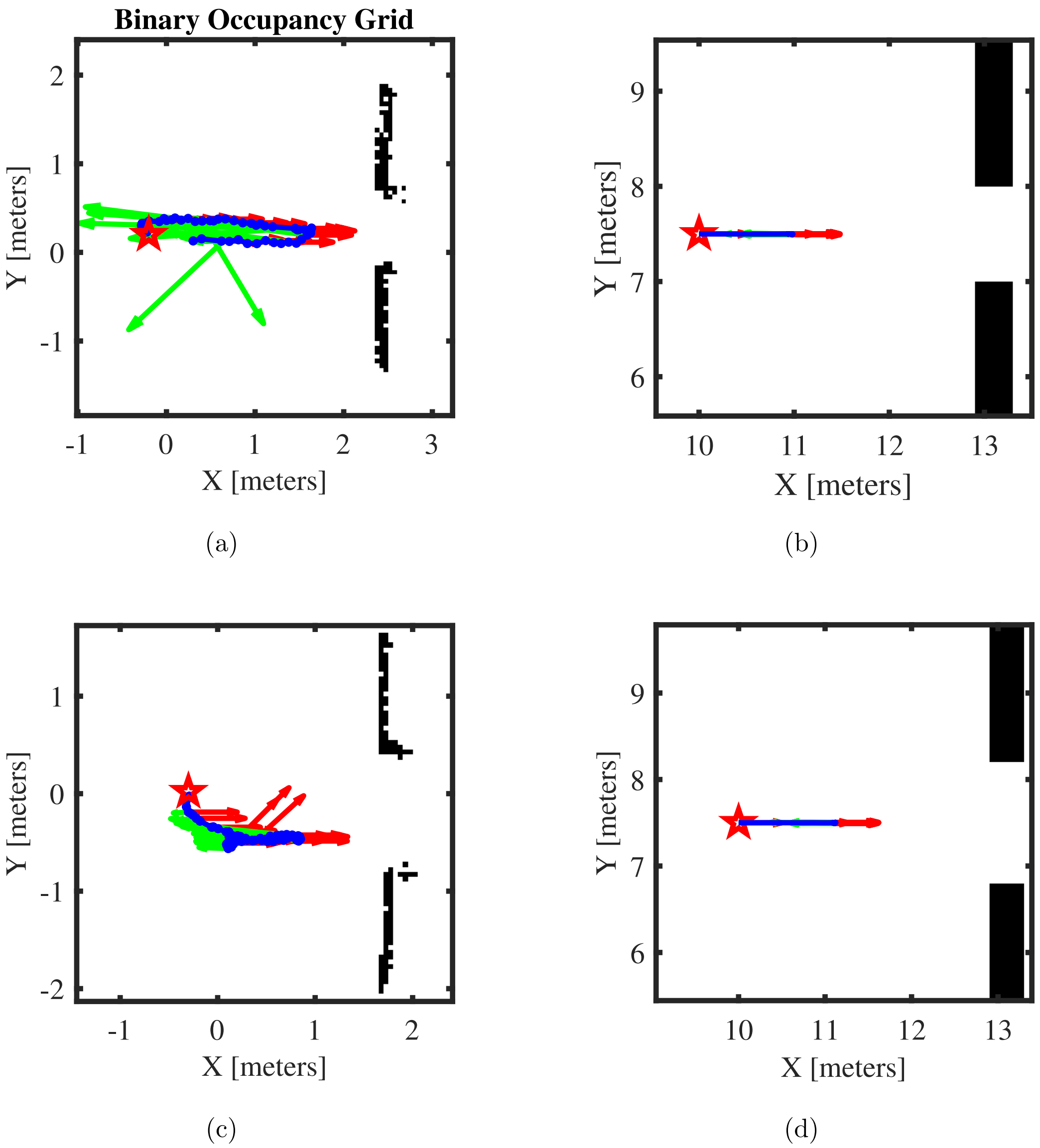


Figure 7.3: Trajectories (in blue), pilot vectors (red arrows) and repulsive vectors(green arrows) in the gap experiments. The trajectory starts at the red star. The configurations shown are (a) 1 m gap, atrium; (b) 1 m gap, simulation; (c) 1.4 m gap, atrium and (d) 1.4 m gap, simulation.

The trajectory in Figure 7.3a shows the UAV rebounding from the gap clearly as it moves right from the red star and is subsequently repelled from the gap, moving left. Another key behaviour is noted in Figure 7.3c as the UAV starts offset from the gap center and is ”pushed” downwards by repulsive forces of the top-most wall. The UAV is also repelled as in the 1 m gap experiment. Also notable in Figure 7.3c is the pilot’s attempts at offsetting the downwards repulsive force to re-align the UAV with the gap center.

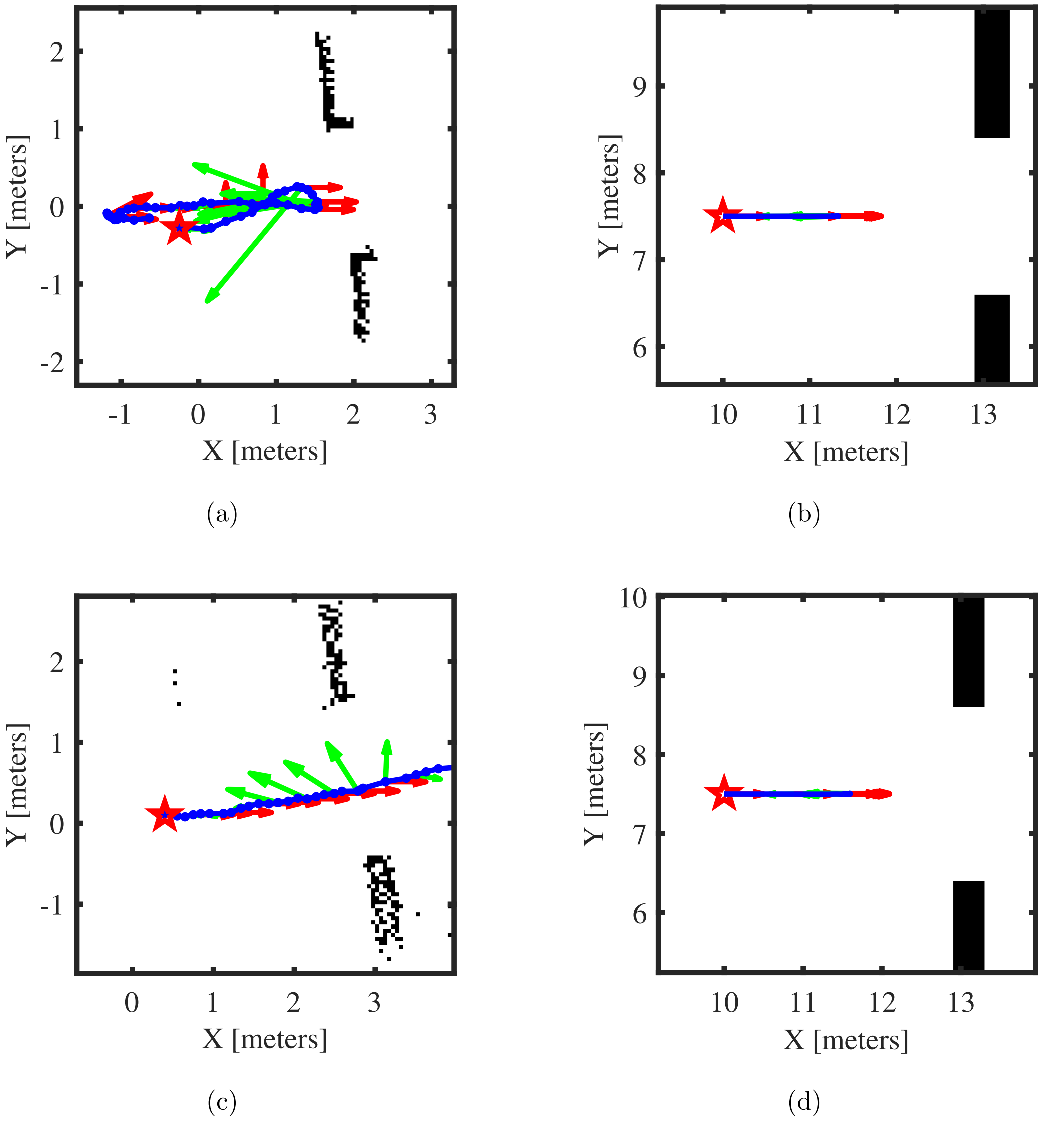


Figure 7.4: Trajectories (in blue), pilot vectors (red arrows) and repulsive vectors(green arrows) in the gap experiments. The trajectory starts at the red star. The configurations shown are (a) 1.8 m gap, atrium; (b) 1.8 m gap, simulation; (c) 2.2 m gap, atrium and (d) 2.2 m gap, simulation.

The trajectory in Figure 7.4a shows more of the rebounding behaviour observed in previous experiments. The UAV makes it closer to the gap, and is then strongly repelled, indicated by the long green vectors above. The saddle shape between the gaps, noted in Chapter 3, is congruent also with the figure-of-eight shape of the trajectory as it is repelled first by the top wall on the approach and then the bottom wall on the retreat. Figure 7.4d simply shows the UAV passing through the gap.

From a proximity perspective, the vehicle avoided collisions in all cases. Across both simulated and actual experiments, as the gap widened, the mean proximity decreased. This result is expected since the net repulsive force in the x-direction that opposes the pilot vector would be lessened as the obstacles are moved further away. Once again the simulated experiments are observed to have a higher mean proximity than the actual, indicating safer flight.

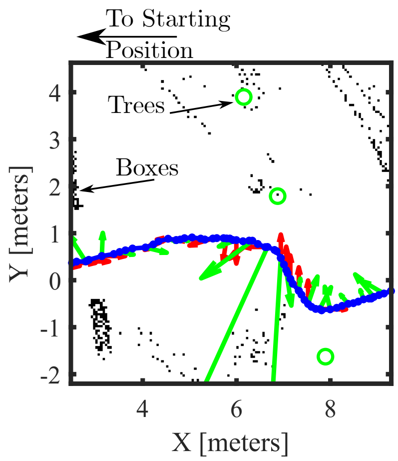


Figure 7.5: The UAV trajectory after flying through the 2.2 m gap on the left, where it is propelled towards a tree (indicated by green circles).

The 0.5 m minimum proximity of the actual 2.2 m experiment occurs after passing through the gap, where it encounters a tree and is very strongly repelled from it as in Figure 7.5. The result that a minimally sized obstacle is avoided successfully is positive, however the minimum proximity of 0.50 m indicates that the repulsive field may need to be more sensitive. Additionally, the occurrence of a minimally sized obstacle may skew the average metrics. Data before 3 s in Figure B.9 shows the behaviour when passing through the gap.

The repulsive force vectors shown for each experiment are consistently bigger in the x direction, since that is the direction being opposed. However, the rebounding effect noted in the single obstacle experiment does manifest in the actual flights where the gap is not passed. The metrics highlight this oscillatory behaviour through a high repulsive force gradient. This result is far less prevalent in the simulated environments where the UAV comes to a complete stop without rebounding/oscillating.

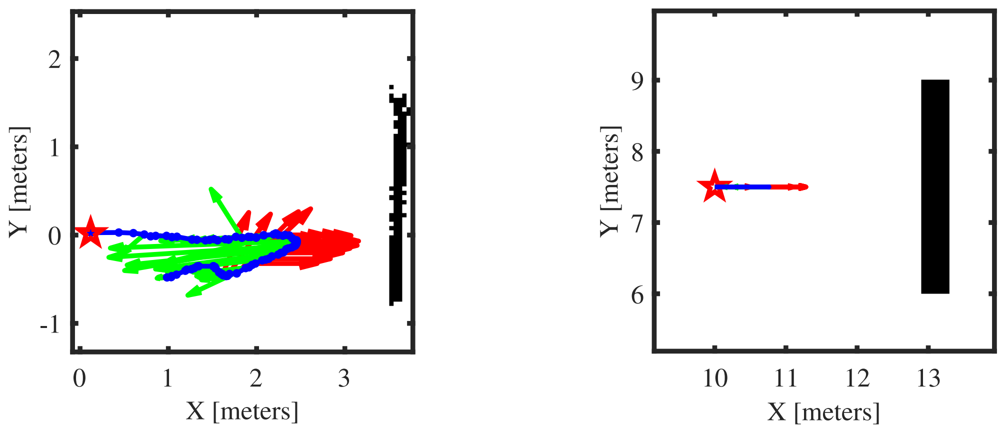
7.2.3 Wall

In this experiment the APF is tested against a solid wall of obstacles with no clear path through or around them. The expected behaviour is to stop at some steady state distance from the wall as observed in [13, 42]. The UAV is flown at an altitude of 1 m and approaches the wall from 3 m away with a piloted vector to the right.

Table 7.4: Summarised metrics for the wall experiments. The data from the actual experiment is from 100 samples, while the simulation provided 2000 samples.

	Proximity [m]			$\mathbf{F_{rep}}$ **RMS**		$\dot{\mathbf{F}}_{\mathbf{rep}}$ **RMS**	
	Mean	**Std. Dev.**	**Min**	**x**	**y**	**x**	**y**
Actual	1.70	0.79	1.00	0.78	0.17	0.85	1.12
Simulated	2.15	0.10	2.1	0.48	2e-3	0.07	0.01

The experiment results are summarised quantitatively in Table 7.4 and the trajectories for simulated and actual experiments are given in Figure 7.6. As expected collision with the wall is avoided with a minimum proximity of 1 m. The simulated experiment shows more sensitivity to the wall and has a greater mean proximity. The oscillatory behaviour of the actual experiment is shown to be greater by the magnitudes of $\dot{\mathbf{F}}_{\mathbf{rep}}$ RMS.



(a) (b)

Figure 7.6: The trajectory of the UAV (a) in the atrium and (b) in the simulation, where the pilot is intentionally commanding a vector (red vector) towards a wall. As the LiDAR senses the wall, it generates a repulsive vector (in green). The trajectory starts at the red star.

Figure 7.6a shows the UAV rebounding behaviour again as the wall approaches and then retreats from the wall despite the constant pilot vector towards the obstacle. The rebounding effect is not visible in Figure 7.6b. Neither experiment as shown in Figure 7.6 collides with the wall.

The RMS value of the repulsive magnitude in the x-direction is indicative of low compliance since the pilot vector is cancelled. Having the repulsive magnitude be greater than the pilot vector magnitude (0.5) is expressed once again as a rebounding from the wall, leading to a higher vector gradient.

The oscillation metric (also vector gradient) is disproportionately high in the y-direction despite little observed movement in the trajectory (the y deviation is at most 0.5 m). The reason for this is seen in Figure B.11, which presents a high degree of erroneous data picked up by the LiDAR which results in high variability in the repulsive vector, especially in the y-direction.

The simulated UAV behaviour matches the expected result observed in previous work, settling 2.1 m away from the wall. The measured compliance and oscillation metrics in the y-axis are non-zero due to the noise in the LiDAR data.

7.3 Tunnel Experiments

A 10 m stretch of the tunnel in Figure 6.9a is used for the underground experiments. The tunnel has a width of 3 m. 5 m from the starting position is an obstacle perturbation of size 0.4 m, which creates a gap of size 2.6 m. The starting position is marked midway between the walls, and the pilot commands a vector down the tunnel at an altitude of 1 m, before landing.

The experiment is performed twice in both the real and simulated environments, first with the obstacle width parameter set to 0.1 m, as in the atrium experiments, and then 0.4 m. This results in a repulsive gain of 0.1 and 0.02 respectively. The reasoning for repeating the experiments at different gains is discussed below.

Table 7.5: Averaged metrics of proximity, pilot compliance and oscillations for the tunnel experiments. The data shows a better APF performance when the repulsive gain is lower. The data from the actual experiment is from 100 samples, while the simulation provided 2000 samples.

	Proximity [m]			**$\mathbf{F_{rep}}$ RMS**		**$\dot{\mathbf{F}}_{\mathbf{rep}}$ RMS**	
	Mean	**Std. Dev.**	**Min**	**x**	**y**	**x**	**y**
Actual High Gain	1.3	0.24	0.81	0.36	1.32	0.74	2.6
Simulated High Gain	1.37	0.09	1.09	0.16	0.26	0.26	1.57
Actual Low Gain	1.4	0.14	1.00	0.02	0.12	0.07	0.20
Simulated Low Gain	1.39	0.06	1.09	0.04	0.02	0.07	0.11

The results for the higher-gain experiments are shown in Figure 7.7a and 7.7b and in Table 7.5. Before the perturbation, the real high-gain experiment oscillates with an amplitude of 30 cm, but after the perturbation begins to oscillate with an amplitude of 1 m. The direction of oscillation is in the y-axis, indicated by the high RMS repulsive gradient. The source of the oscillation is attributed to the APF, as evidenced by Figure B.13 in Appendix B.

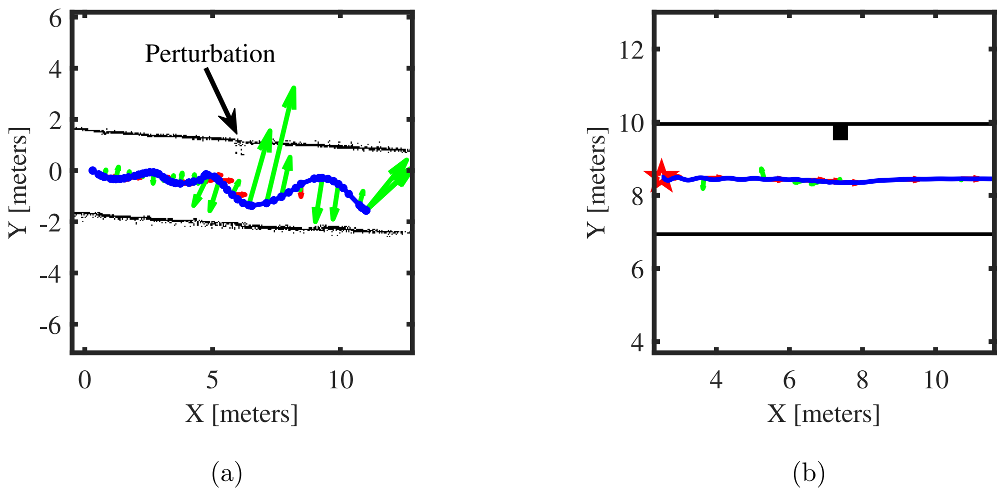
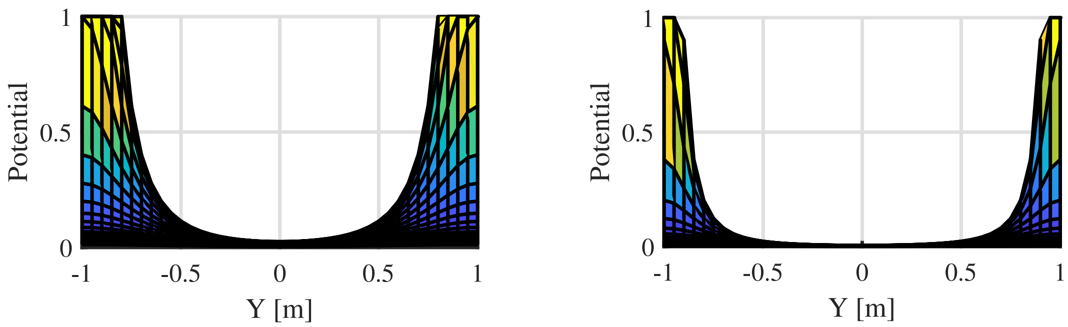


Figure 7.7: Trajectories (in blue), pilot vectors (red arrows) and repulsive vectors(green arrows) in the tunnel experiments. The configurations shown are (a) 0.1 repulsive gain, tunnel; (b) 0.1 repulsive gain, simulation.

The simulated UAV does oscillate in the high gain experiment, but the amplitude is 10 cm. Kroghs velocity scaling method is able to dampen the oscillations in the simulation, but the effect of the perturbation does increase the oscillation amplitude.

The oscillations are unsurprising, and their origin is discussed in Chapter 3. However, they are still a problem, increasing the pilot effort rather than reducing it. Although collisions are being avoided (the minimum proximity is 0.82 m in the high-gain tunnel experiment) the reduced pilot authority is undesirable.

To address this, consider Figure 7.8 which shows the computed potential field for a 2 m wide tunnel with a gain of 0.1 and 0.02. The region of influence of the higher-gain potential field is larger which reduces pilot authority along the Y-axis, but also increases the opportunity for oscillations since the potential fields are closer together. The smaller potential field has a wider saddle region increasing the pilot authority. By reducing the repulsive gain the potential field is made smaller which may reduce oscillations at the cost of higher collision risk. In Figure 7.8, the potential



(a) (b)

Figure 7.8: A cross-cut view of a potential field represented as a surface in a 2 m wide tunnel.

field is calculated using a repulsive gain of (a) 0.1 and (b) 0.02, showing how a bigger gain (Figure 7.8a) results in a larger field which is responsible for oscillations. Figure 7.8b shows a high potential near the walls, which is desirable to avoid collisions, but contains a much flatter region close to the midpoint than the higher-gain field in Figure 7.8a. Note that the gradient of the surface is used to compute the repulsive force used in the APF algorithm, and as such the wider saddle region in (b) provides more pilot authority along the y-axis.

The effect of decreasing the repulsive gain to 0.02 is shown in Figure 7.9a and 7.9b. The simulated environment presents no oscillations, and passes the obstacle perturbation without problems. Oscillations in the real environment are present, but dampened as the UAV self centers before the perturbation, where it is less sharply repelled than in the higher gain experiments.

The effects of reducing the repulsive gain are also evident in the quantitative results in Table 7.5 which shows increased pilot authority (by the reduced repulsive magnitude) and reduced oscillatory effects (by the repulsive vector gradient).

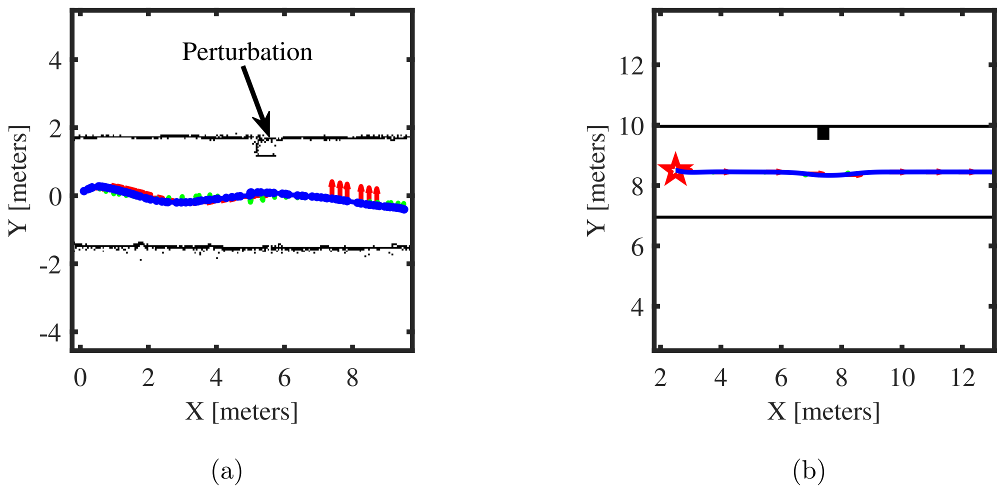


Figure 7.9: Trajectories (in blue), pilot vectors (red arrows) and repulsive vectors(green arrows) in the tunnel experiments. The configurations shown are (a) 0.02 repulsive gain, tunnel; (b) 0.02 repulsive gain, simulation.

Figure 7.9 shows fewer oscillations in the actual trajectory than Figure 7.7. There is a marked decrease in the y-axis oscillations, even after the perturbation, indicating that reducing the gain sufficiently helps dampen oscillations. The effect is also visible in the simulation, albeit at a smaller scale since there are few oscillations to begin with.

7.4 Conclusion

7.4.1 Simulation Disparity

Each of the experiments note a difference in performance of the simulated UAV to the actual system. While in general the behaviour is similar, the simulation does not fully capture some things. For example, the minimum proximity in all simulated environments is on average 0.77 m more than the actual experiments.

The first observation is that flying the UAV outdoors in the `LOITER` mode provided by the FC uses a Global Positioning System (GPS) to perform position and velocity control. Flight in this mode is required in order to measure the velocity inputs and outputs. An important feature of this mode is braking, which rapidly stops the UAV translation.

The atrium and tunnel experiments however are performed in the `GUIDED_NOGPS` flight mode (since GPS does not work underground). This mode has no position or velocity control, nor does it feature braking. The identified model features this braking, providing a fast 2nd-order response to velocity inputs, while the actual underground UAV does not. An improved model would either be achieved analytically, or by fitting a system that maps the tilt angle input to the translational acceleration while in the `GUIDED_NOGPS` mode.

Secondly, the system identification for the UAV model maps the velocity reponse to the velocity input which is not instantaneous. However, there are other potential sources of delay that have not been modelled. The software is layed out such that the LiDAR data flows as follows:[3] Repulsive Field Calculation->Multiplexing with Pilot Command->Driver->mavros->Ardupilot.

Table 7.6 presents the timing results after injecting a short range (0.5 m) LiDAR point in the middle of the function responsible for calculating the total repulsive force. The short range will dominate the repulsive response which propagates through the system. The delay between the injection and the resulting attitude command can then be observed. Performing 5 of these experiments shows an average delay of 74.5 ms between receiving the LiDAR data and sending the attitude setpoint to `mavros`. The delay between measuring a perturbation and acting on it is greater than the LiDAR sample time (50 ms at 20 Hz) which makes it considerable.

[3]The delay of the light reflecting of the surface being measured is neglible at roughly 8e-8 s at the maximum LiDAR range of 12 m.

Table 7.6: Characterising the delay between receiving the scan from the `rp_lidar` ROS node, to injecting a short range reading which results in a large pulse in the repulsive force, then the total force, then the attitude command after some delay.

	Time since scan received [seconds]					
Run	**A**	**B**	**C**	**D**	**E**	**Average**
Inject Impulse	0.0219	0.0215	0.0181	0.0202	0.0274	0.0218
Publish F_{rep}	0.0438	0.0463	0.0381	0.0395	0.0512	0.0437
Publish F_{tot}	0.0827	0.0628	0.0539	0.0625	0.0644	0.0653
Publish Attitude	0.0904	0.0738	0.0652	0.0681	0.0749	0.0745

There are additional delays as the UAV rotational response required to generate translational velocity is not instantaneous, but these are considered lumped as part of the originally identified system. However, the delays identified above are substantial considering that the LiDAR scan rate is 20 Hz. The simulations are also performed with zero delay, which may be an additional cause of disparity between the actual and simulated behaviour. Finally, while Gaussian noise is used to model the LiDAR range noise, it does not model the misreadings observed in Figure B.11. The misreadings appear to be a result of the laser scanner detecting parts that are on the frame, and could be modelled by random fault injection. However, the APF exhibits good immunity to the errors. Further experimentation regarding the noise immunity of the APF algorithm is required.

7.4.2 Critical Analysis

The purpose of this chapter is an experimental validation of the APF algorithm on a physical platform. The goal of a local collision avoidance system is just that: to avoid collisions. In that respect, the APF algorithm performs well, supported by the proximity metric, which was at all times greater than the vehicle center-to-tip dimension (which would indicate a collision). However, in terms of the safety buffer definition of flight safety in Chapter 4, some situations lead to unsafe conditions. In particular, approaching a tree of diameter 0.1 m lead to the smallest proximity which breached the safety buffer, as well as the wall effect buffer. Regardless, collisions are still avoided, and the APF algorithm performs adequately to avoid intentional pilot-commanded collisions. Additionally, the selection of parameters to avoid collisions using the parametrisation method is verified.

From a pilot authority perspective, the APF algorithm is particularly sensitive to environmental constrained-ness as well as the repulsive gain. Avoiding obstacles requires a reduced pilot authority, however in situations where there is no obstacle along the pilot trajectory (as in the gap experiments), better compliance of the pilots commands is desirable for ease of use. However, even a gap-finding algorithm must set a minimum gap size, whereas in this case it is found empirically to be 2.2 m. The experiments and metrics do not capture the pilot ease of use, which is an important metric if the APF algorithm is to be adopted for value-adding use-cases in mining environments.

Oscillatory behaviour is perhaps the greatest draw-back of the APF algorithm in terms of pilot ease of use. When approaching an obstacle, the pilot expects the UAV to slow down and then come to a stop at a safe distance from the obstacle, a behaviour noted in the simulation. However, many of the atrium experiments show a rebounding effect, quantified by the high repulsive magnitude gradients shown in the single obstacle, wall, and gap tests.

A possible recourse for this is to modify the repulsive function to vary linearly with distance, rather than quadratically. The effect is that breaching the safety buffer, even by a small distance, does not result in a much larger rebounding force.

The experiments also confirmed the trade off between flight safety and pilot authority. Had the repulsive gain been reduced in the gap experiments, a narrower gap would likely have become passable. However, the risk of collision with smaller obstacles then increases. In an underground mining environment the UAV would detect both a hanging chain and a truck, and it would be desirable to avoid both with equal proximity. Each detected obstacle should be capable of maximally repelling the UAV, which is currently not true since all repulsive forces are additive, resulting in larger objects creating larger repulsive forces.

A potential mitigation for this is to cluster points together as a form of obstacle detection, calculating the repulsive force from the centroid of the cluster [49]. Clustering can be computationally expensive, so a second alternative is to limit the total repulsive force in any one direction. For example, a small obstacle to the left produces a repulsive force to the right, while a large wall on the right produces a repulsive force to the left. Limiting the effect of the bigger obstacle means the smaller obstacle can cancel the repulsive force, resulting in a centering behaviour.

In conclusion the atrium experiments show that a single obstacle 0.4 m wide and a wall are both avoided despite an intentional pilot collision course. Applying the parametrisation method with an obstacle width of 0.1 m, the UAV flies through a gap of 2.2 m. In the 3 m wide tunnel, boths walls are avoided, and the pilot is able to fly down the tunnel past a perturbation. Choosing a larger minimum obstacle size results in less oscillation before the perturbation, with no significant response to the perturbation (as in the more sensitive test).

Chapter 8

Conclusion and Future Work

8.1 Conclusion

Chapter 4 proposes three research questions/objectives designed to add to the body of knowledge surrounding real time collision avoidance using potential fields. A parametrisation method has been developed in order to choose APF repulsive gain and maximum repulsive range in a more robust manner than trial and error.

As shown in Chapter 5, the parametrisation is robust across a range of sample rates, angular resolutions and repulsive gain/minimum obstacle width. The Chapter also discusses the sensitivity of the APF algorithm to each of the above variables, and provides a comparative measure of flight safety, pilot authority and oscillation. The primary results of this sensitivity analysis is that a higher scan rate is desirable-even at the risk of a coarser angular resolution- since it reduces oscillatory behaviour. A minimum obstacle width between 0.1 m and 1 m provides a good balance between pilot authority and flight safety.

The parametrisation is then applied to a real UAV in order to assess the algorithm performance for teleoperation in a constrained environment. Chapter 6 describes the hardware and software components required to enable collisions-free flight while addressing the technical implementation challenges.

Then, by choosing the minimum obstacle width; applying the sensor scan rate and maximium range; UAV maximum velocity and dimensions; as well as angular resolution as inputs to the parametrisation, a repulsive gain is computed.

Physical testing in an open area shows that the repulsive gain is sufficient to avoid an obstacle of size 0.4 m in the direct path of the piloted command. The UAV also avoids a wall with no clear trajectory around it. In terms of constrainedness, the UAV is able to pass between obstacles with a 2.2 m wide gap between them.

Further validation of the APF algorithm for teleoperation in an underground mine is performed. A 10 m stretch of 3 m wide tunnel is flown, with an obstacle perturbation along one of the tunnel walls. The UAV is able to fly collision free in the constrained environment.

8.2 Contributions

The contributions of the research are summarised as follows:

- Developed teleoperable UAV capable of flying in a constrained environment, collision free and in real time using the APF algorithm, previously only performed on a UGV and in simulation on a UAV to the authors knowledge.
- Developed a method for choosing the repulsive parameters of an APF that provide a balanced tuning to prevent collisions.
- Demonstrated through simulation the use of the parametrisation method across a broad range of sensor settings, also revealing the limits of the parametrisation.
- Validated through experimentation the use of the parametrisation to counteract intentional pilot collision commands. The vehicle also passes through a gap 2.2 m wide, and flies in a mine-like environment 3 m wide.
- Performed system identification to find a second order model of a pilot-friendly UAV.
- Proposed unique, quantifiable metrics for comparing collision avoidance algorithms.

8.3 Future Work

A number of possible research avenues are identified during the research, given below:

- Chapter 2 discusses three sensors, lidar, sonar, and the camera, which are lightweight enough for a UAV. However, early prototyping suggests the sensitivity of these devices should be researched. Specifically, their sensitivity to dust, moisture, vibration, electromagnetic interference and the movement of air around the propellers must be measured to identify the best candidate for a UAV in the long term.
- The effects of wind on an APF-enabled UAV are explored by Steele [50], and a validation of their results on a real platform in a wind tunnel is required.
- The advent of autonomous vehicles is causing a vast amount of research and development into Three-dimensional (3D) LiDAR. The APF algorithm can be extended in 3D to provide completely omnidirectional collision avoidance.
- There are many collision avoidance algorithms that can be applied to the same or similar platform and directly compared using the given metrics for proximity, pilot authority and oscillation.
- The rebounding problem is not adequately solved using Krogh velocity scaling alone. Implementation of a global velocity controller (rather than relative) is a possible avenue for stabilising the collision avoidance in a constrained environment. This may require external positioning systems (costly) or onboard SLAM, however, velocity can also be estimated by averaging the velocity computed in Equation 3.6. Another option is a fixed braking manoeuvrer such that a braking command aggressively opposes the direction of motion, preventing the UAV from breaching the safety buffer and rebounding.
- Reducing the vehicle size and improving the LiDAR scan rate and delay parameters, even more constrained environments can be considered. This is however more of an engineering problem than a research task.
- The system identification in this research is performed outdoors, with a well tuned position and velocity controller using GPS. The flight dynamics indoors, without a position or velocity controller, are different as observed in the previous chapter. An accurate system identification would model the input tilt angle against the output velocity.

References

[1] D. Msiza, “Statement on the 2018 Mine Health and Safety Statistics, Delivered by Chief Inspector of Mines, Mr David Msiza,” Mar. 2019. Library Catalog: www.dmr.gov.za.

[2] P. Corke, J. Roberts, J. Cunningham, and D. Hainsworth, “Mining Robotics,” pp. 1127–1150, Jan. 2008.

[3] Edwards, Stuart Robert, *Autonomous 3D mapping and surveillance of mines with MAVs*. PhD thesis, University of the Witwatersrand, Johannesburg, July 2017.

[4] E. Yanmaz, S. Yahyanejad, B. Rinner, H. Hellwagner, and C. Bettstetter, “Drone Networks: Communications, Coordination, and Sensing,” *Ad Hoc Networks*, vol. 68, 2017.

[5] S. Barman, R. Rouf, M. S. Islam, and M. M. Hasan, “Proposed Design for Automated Agro-Quadcopter ” Agro-ropter ”,” 2017.

[6] B. KROGH, “A generalized potential field approach to obstacle avoidance control,” *Proc. SME Conf. on Robotics Research : The Next Five Years and Beyond, Bethlehem, PA, 1984*, pp. 11–22, 1984.

[7] J. Green and K. Hlophe, “Underground mine navigation using an integrated IMU/TOF system with unscented Kalman filter,” 2011.

[8] Jeremy Green, “Requirements elicitation techniques for the development of an underground remote piloted aerial system,” 2017.

[9] R. Stopforth, G. Bright, and R. Harley, “Performance of the improvements of the CAESAR robot,” *International Journal of Advanced Robotic Systems*, vol. 7, 2010.

[10] C. Kanellakis, S. Sharif Mansouri, G. Georgoulas, and G. Nikolakopoulos, “Towards Autonomous Surveying of Underground Mine using MAVs,” 2018.

[11] P. Gohl, M. Burri, S. Omari, J. Rehder, J. Nikolic, M. Achtelik, and R. Siegwart, "Towards autonomous mine inspection," in *Proceedings of the 2014 3rd International Conference on Applied Robotics for the Power Industry*, (Foz do Iguassu, Brazil), pp. 1–6, IEEE, Oct. 2014.

[12] G. Hoffmann, D. Rajnarayan, S. Waslander, D. Dostal, J. Jang, and C. Tomlin, "The Stanford testbed of autonomous rotorcraft for multi agent control (STARMAC)," in *The 23rd Digital Avionics Systems Conference (IEEE Cat. No.04CH37576)*, (Salt Lake City, UT, USA), pp. 12.E.4–121–10, IEEE, 2004.

[13] T. Lam, H. Boschloo, M. Mulder, and M. van Paassen, "Artificial Force Field for Haptic Feedback in UAV Teleoperation," *IEEE Trans. Syst., Man, Cybern. A*, vol. 39, pp. 1316–1330, Nov. 2009.

[14] European Union, "Directive 2014/34/EU of the european parliament and of the council of 26 february 2014 on the harmonisation of the laws of the member states relating to equipment and protective systems intended for use in potentially explosive atmospheres (recast) (text with EEA relevance)," vol. L 96, pp. 309–357.

[15] S. Garrido, "Path Planning for Mobile Robot Navigation using Voronoi Diagram and Fast Marching," 2011.

[16] H. Choset and J. Burdick, "Sensor-Based Exploration: The Hierarchical Generalized Voronoi Graph," *The International Journal of Robotics Research*, vol. 19, pp. 96–125, Feb. 2000.

[17] P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, pp. 100–107, July 1968.

[18] A. Viquerat, L. Blackhall, A. Reid, S. Sukkarieh, and G. Brooker, "Reactive Collision Avoidance for Unmanned Aerial Vehicles Using Doppler Radar," in *Field and Service Robotics: Results of the 6th International Conference* (C. Laugier and R. Siegwart, eds.), Springer Tracts in Advanced Robotics, pp. 245–254, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[19] J. Minguez, L. Montano, and O. Khatib, "Reactive collision avoidance for navigation with dynamic constraints," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, pp. 588–594 vol.1, Sept. 2002.

[20] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robot. Automat. Mag.*, vol. 4, pp. 23–33, 1997.

[21] D. Bareiss, J. v. d. Berg, and K. K. Leang, "Stochastic automatic collision avoidance for tele-operated unmanned aerial vehicles," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4818–4825, Sept. 2015.

[22] B. Du and S. Liu, "A common obstacle avoidance module based on fuzzy algorithm for Unmanned Aerial Vehicle," in *2016 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, pp. 245–248, June 2016.

[23] J. A. Steiner, X. He, J. R. Bourne, and K. K. Leang, "Open-sector rapid-reactive collision avoidance: Application in aerial robot navigation through outdoor unstructured environments," *Robotics and Autonomous Systems*, vol. 112, pp. 211–220, Feb. 2019.

[24] M. Nieuwenhuisen, D. Droeschel, M. Beul, and S. Behnke, "Obstacle detection and navigation planning for autonomous micro aerial vehicles," in *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 1040–1047, May 2014.

[25] P. Stegagno, M. Basile, H. H. Bülthoff, and A. Franchi, "Vision-based Autonomous Control of a Quadrotor UAV using an Onboard RGB-D Camera and its Application to Haptic Teleoperation," *IFAC Proceedings Volumes*, vol. 46, no. 30, pp. 87–92, 2013.

[26] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," *IEEE Robotics Automation Magazine*, vol. 13, pp. 99–110, June 2006.

[27] B. Vedder, H. Eriksson, D. Skarin, J. Vinter, and M. Jonsson, "Towards collision avoidance for commodity hardware quadcopters with ultrasound localization," in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 193–203, June 2015.

[28] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots," *IEEE Trans. Syst., Man, Cybern.*, vol. 19, pp. 1179–1187, Sept. 1989.

[29] M. Y. Chen, D. H. Edwards, E. L. Boehmer, N. M. Eller, J. T. Slack, C. R. Speck, S. M. Brown, H. G. Williams, S. H. Wilson, C. S. Gillum, G. C. Lewin, M. S. Sherriff, and G. T. Garner, "Designing a spatially aware and autonomous quadcopter," in *2013 IEEE Systems and Information Engineering Design Symposium*, pp. 213–218, Apr. 2013.

[30] G. A. Kumar, A. K. Patil, R. Patil, S. S. Park, and Y. H. Chai, "A LiDAR and IMU Integrated Indoor Navigation System for UAVs and Its Application in Real-Time Pipeline Classification," in *Sensors*, 2017.

[31] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *1985 IEEE International Conference on Robotics and Automation Proceedings*, vol. 2, pp. 500–505, Mar. 1985.

[32] Y. Koren and J. Borenstein, "Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation," vol. 2, pp. 1398–1404 vol.2, May 1991.

[33] M. C. Lee and M. G. Park, "Artificial potential field based path planning for mobile robots using a virtual obstacle concept," in *Proceedings 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2003)*, vol. 2, pp. 735–740 vol.2, July 2003. ISSN: null.

[34] J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *IEEE Trans. Robot. Automat.*, vol. 7, pp. 278–288, June 1991.

[35] R. J. van Breda, "Vector field histogram star obstacle avoidance system for multicopters - Semantic Scholar," Dec. 2016.

[36] J. Sfeir, M. Saad, and H. Saliah-Hassane, "An improved Artificial Potential Field approach to real-time mobile robot path planning in an unknown environment," in *2011 IEEE International Symposium on Robotic and Sensors Environments (ROSE)*, pp. 208–213, Sept. 2011.

[37] J. Mok, Y. Lee, S. Ko, I. Choi, and H. S. Choi, "Gaussian-mixture based potential field approach for UAV collision avoidance," in *2017 56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, pp. 1316–1319, Sept. 2017.

[38] C. Kanellakis, S. Sharif Mansouri, G. Georgoulas, and G. Nikolakopoulos, "Towards Autonomous Surveying of Underground Mine Using MAVs: Proceedings of the 27th International Conference on Robotics in Alpe-Adria Danube Region (RAAD 2018)," pp. 173–180, 2019.

[39] A. Censi, "An ICP variant using a point-to-line metric," in *2008 IEEE International Conference on Robotics and Automation*, pp. 19–25, May 2008. ISSN: 1050-4729.

[40] G. Grisetti, C. Stachniss, and W. Burgard, "Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters," *IEEE Transactions on Robotics*, vol. 23, pp. 34–46, Feb. 2007. Conference Name: IEEE Transactions on Robotics.

[41] H. Han, C. Xiang, B. Xu, and Y. Yu, "Aerodynamic performance and analysis of a hovering micro-scale shrouded rotor in confined environment," *Advances in Mechanical Engineering*, vol. 11, p. 168781401882332, Apr. 2019.

[42] J.-H. Cho, D.-S. Pae, M.-T. Lim, and T.-K. Kang, "A Real-Time Obstacle Avoidance Method for Autonomous Vehicles Using an Obstacle-Dependent Gaussian Potential Field," 2018.

[43] J. Meyer, A. Sendobry, S. Kohlbrecher, U. Klingauf, and O. v. Stryk, "Comprehensive Simulation of Quadrotor UAVs using ROS and Gazebo," in *3rd Int. Conf. on Simulation, Modeling and Programming for Autonomous Robots (SIMPAR)*, p. to appear, 2012.

[44] M. Nieuwenhuisen, M. Schadler, and S. Behnke, "Predictive Potential Field-Based Collision Avoidance for Multicopters," *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. XL-1/W2, pp. 293–298, Aug. 2013.

[45] F. Bordin, F. G. Müller, E. C. Teixeira, S. B. A. Rolim, F. M. W. Tognoli, L. G. da Silveira Júnior, M. R. Veronez, and M. Scaioni, "An intensity recovery algorithm (IRA) for minimizing the edge effect of LIDAR data," *European Journal of Remote Sensing*, vol. 49, pp. 301–315, Jan. 2016.

[46] Y. K. Hwang and N. Ahuja, "A potential field approach to path planning," *IEEE Transactions on Robotics and Automation*, vol. 8, pp. 23–32, Feb. 1992.

[47] B. Krogh and C. Thorpe, "Integrated path planning and dynamic steering control for autonomous vehicles," in *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, vol. 3, (San Francisco, CA, USA), pp. 1664–1669, Institute of Electrical and Electronics Engineers, 1986.

[48] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "Ros: an open-source robot operating system," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA) Workshop on Open Source Robotics*, (Kobe, Japan), May 2009.

[49] L. Zheng, P. Zhang, J. Tan, and F. Li, "The Obstacle Detection Method of UAV Based on 2D Lidar," *IEEE Access*, vol. 7, pp. 163437–163448, 2019. Conference Name: IEEE Access.

[50] A. Steele, *Development of a close quarters collision-protected aerial drone*. PhD thesis, Stellenbosch University, Stellenbosch, Dec. 2019.

Appendix A

System Identification Data

Chapter 5 details how the system is innervated and how the `MATLAB` System Identification Tool is used to generate a second and third order model of the UAV response to velocity inputs.

The x and y velocity setpoint data is given in Figure A.1, showing the erratic inputs used for excitation. Figure A.2 shows a periodogram of the velocity input (top) and output (bottom) of the plant excitation signal, which shows a flat response over a broad frequency range from around 0.1 Hz to 5 Hz.

Figure A.1 shows the x-and y-velocity excitation signals used for the system identification and modelling. Important features of an excitation signal for system identification are a large variability in the magnitude of the driving setpoint, as well as a broad range of frequencies.

The plant excitation signal spectral density is analysed using a periodogram in Figure A.2. The frequency response shown indicates no distinct frequency spikes up to 4 Hz, indicating that it is a good broad-band signal for system identification.

The two identified systems are described by:

$$G_2(s) = \frac{10.13s + 2.86}{s^2 + 5.106s + 3.172}$$

and,

$$G_3(s) = \frac{397.7s^s + 390.9s + 154.9}{s^3 + 262.1s^2 + 195.8s + 155.2}$$

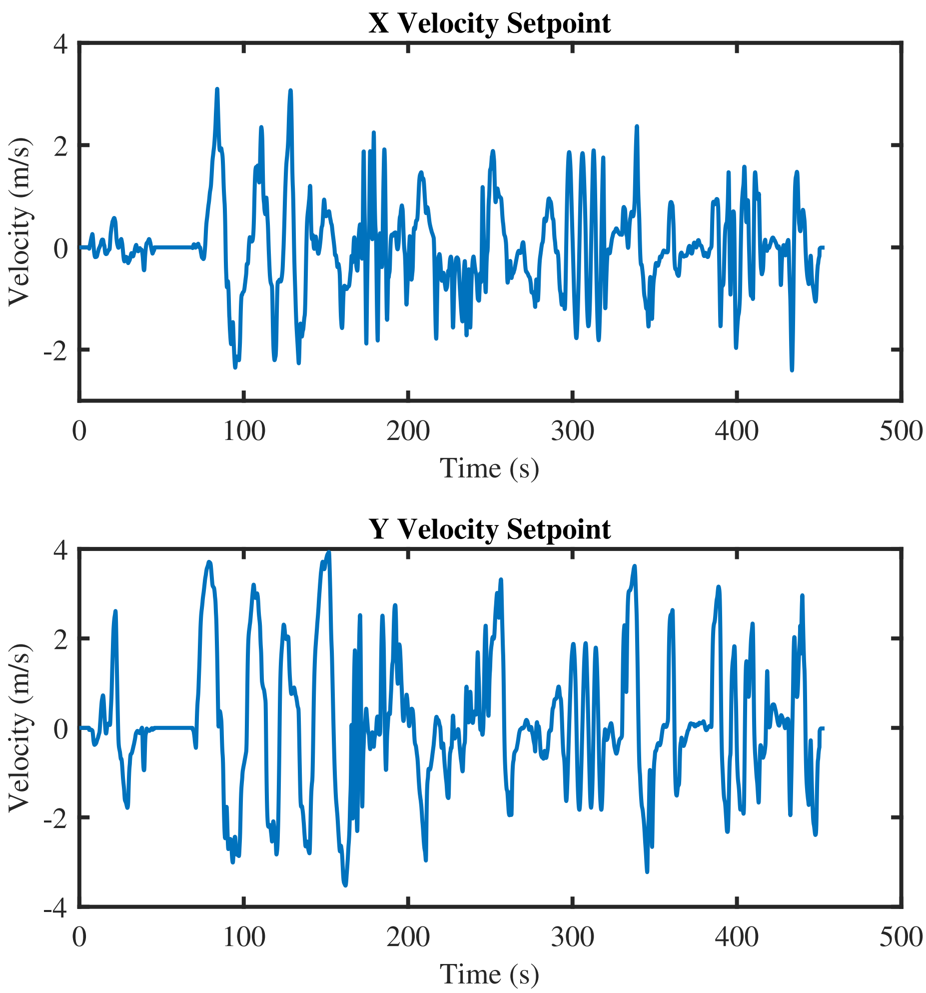


Figure A.1: Plant excitation signals used in the system identification in both horizontal directions. These inputs exhibit a broad amplitude and frequency range of inputs.

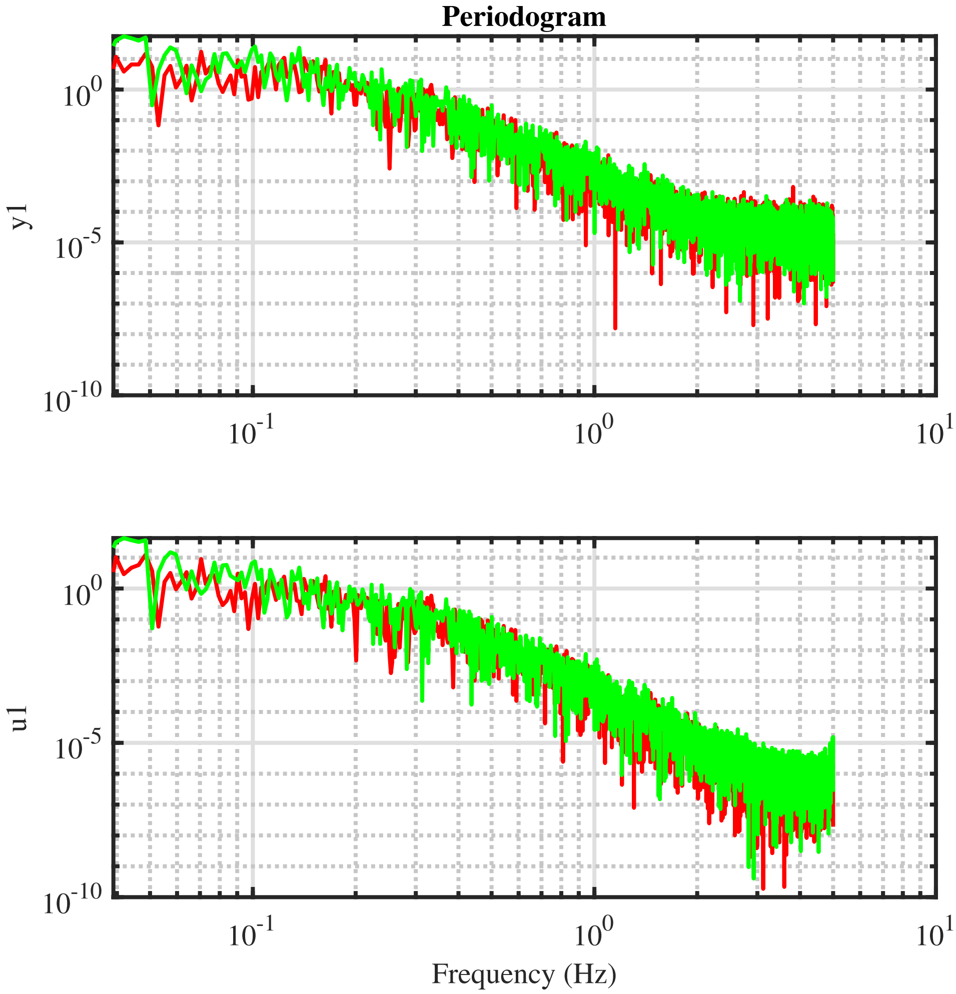


Figure A.2: Periodogram of the input and output signals.

Appendix B

Experiment Data

Data from each experiment in the simulation, atrium and tunnel is presented below. Each figure presents the proximity of the UAV to the nearest measured obstacle in panel one. The second panel contains the magnitude of the difference between the actual command and the pilot command in both the x and y axis of the body frame. This serves as a measure of pilot authority. The third frame presents the gradient of this error magnitude, providing a measure of oscillatory behaviour. Each of these variables are given as a function of time.

They are arranged such that experiment data precedes the simulation data for the same configuration. Shown first is the single obstacle experiment, followed by the 4 gap experiments, the wall experiment and finally the two tunnel experiments.

B.1 Single Obstacle Experiment

Atrium (Real) Data

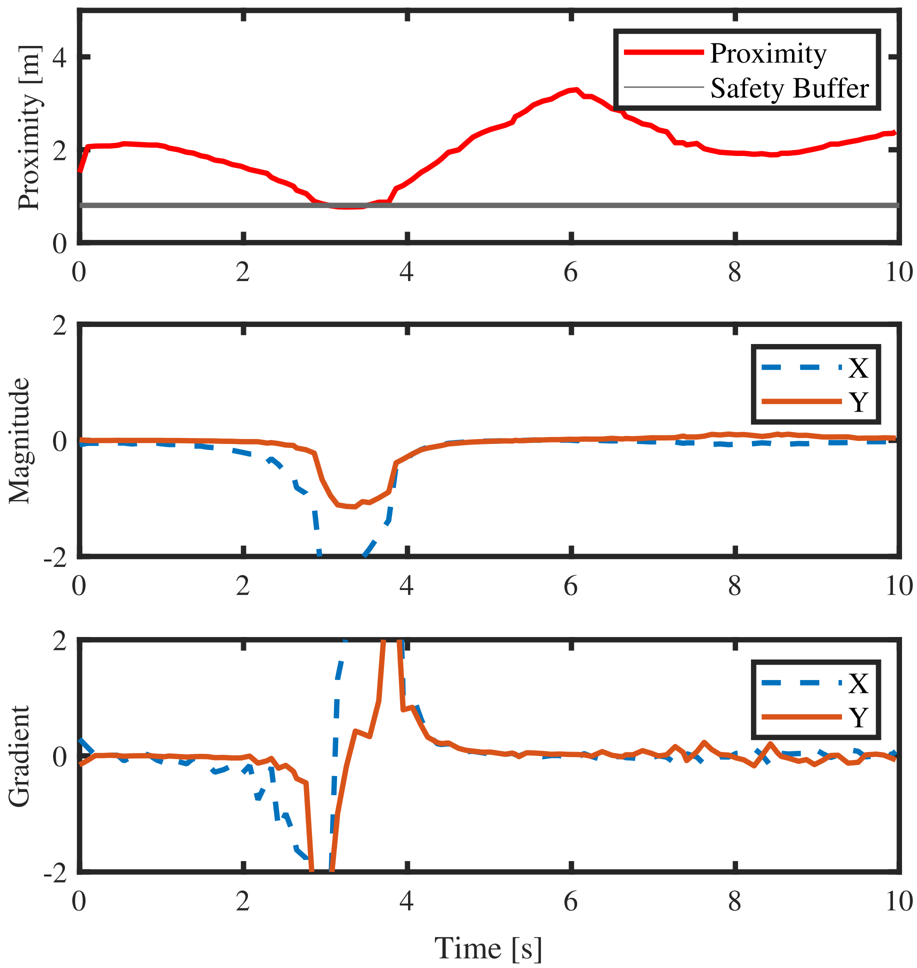


Figure B.1: The UAV proximity in the first panel, the repulsive vector magnitude in the second panel, and the repulsive vector magnitude gradient during the single obstacle experiment in the atrium.

Figure B.1 shows that the UAV is at its closest (0.77 m) at 3 s, briefly breaching the safety buffer. Note also that as the UAV approaches the edges of the atrium at 6 s, it detects a new obstacle as the closest. Nearing the obstacle at 3 s causes the UAV to deviate quickly, which is clear in the third panel. as the repulsive vector gradient spikes rapidly.

Simulation Data

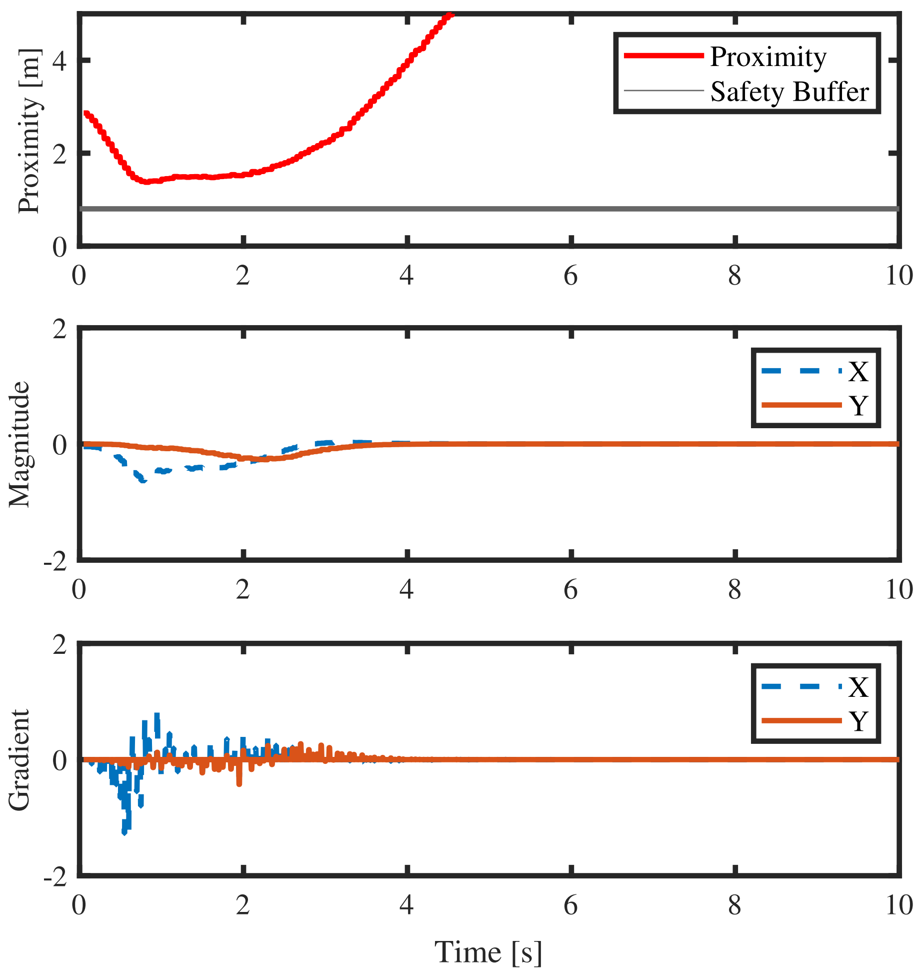


Figure B.2: The UAV proximity in the first panel, the repulsive vector magnitude in the second panel, and the repulsive vector magnitude gradient during the single obstacle experiment in the simulation.

In Figure B.2, the proximity reaches the minimum sooner than the actual experiment, at 0.5 s, since the simulated UAV flies faster. This repulsive magnitude and gradient show that the simulated APF algorithm is less oscillatory since the UAV does not approach the obstacle as closely as the actual experiment, and is therefore not rebounded as much.

B.2 Gap Experiments

B.2.1 1 m Gap

Atrium (Real) Data

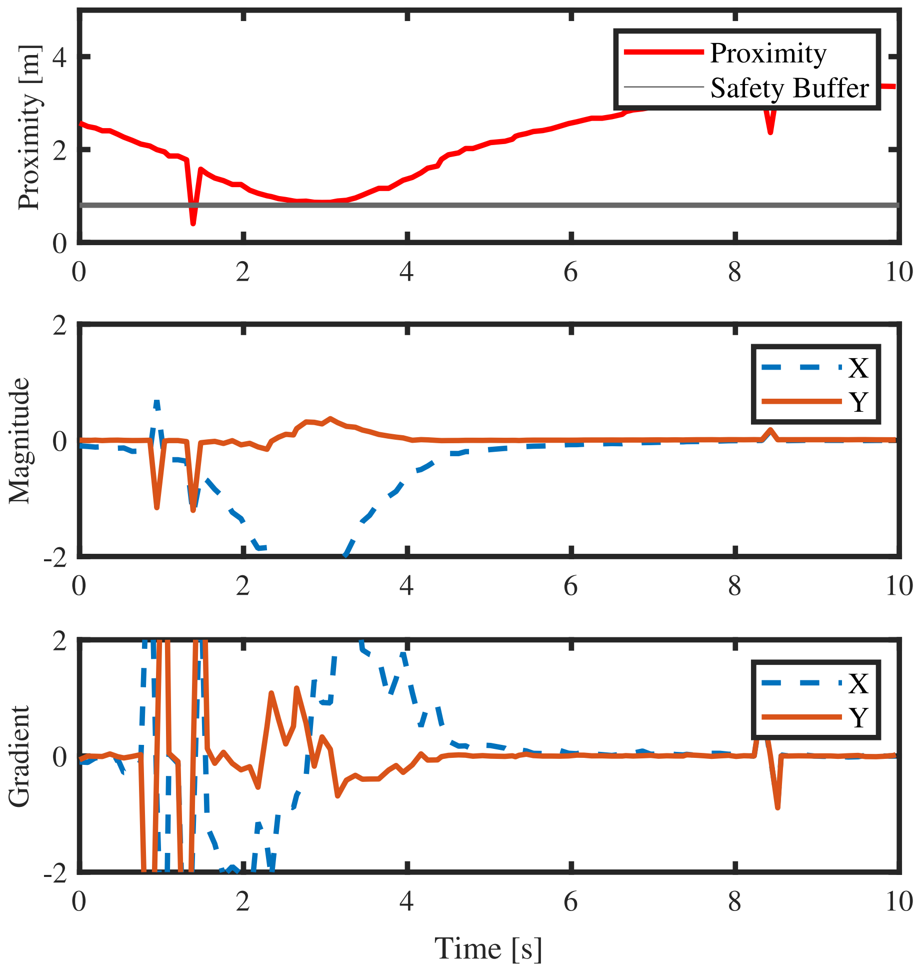


Figure B.3: Data from the 1 m wide gap experiments in the atrium used for quantifying proximity (in panel 1), pilot authority (as repulsive magnitude in panel 2) and oscillation (as the repulsive vector gradient in panel 3).

Figure B.3 shows that the minimum proximity of 1 m is encountered at 3 s, whereafter the UAV is repelled from the obstacles on either side of the gap. Note the spike in proximity at 1.5 s is an error. Panel 2 reveals the repulsive magnitude is much greater in the x-direction than the y-direction, since the pilot is commanding a collision course in the positive x-direction.

Simulation Data

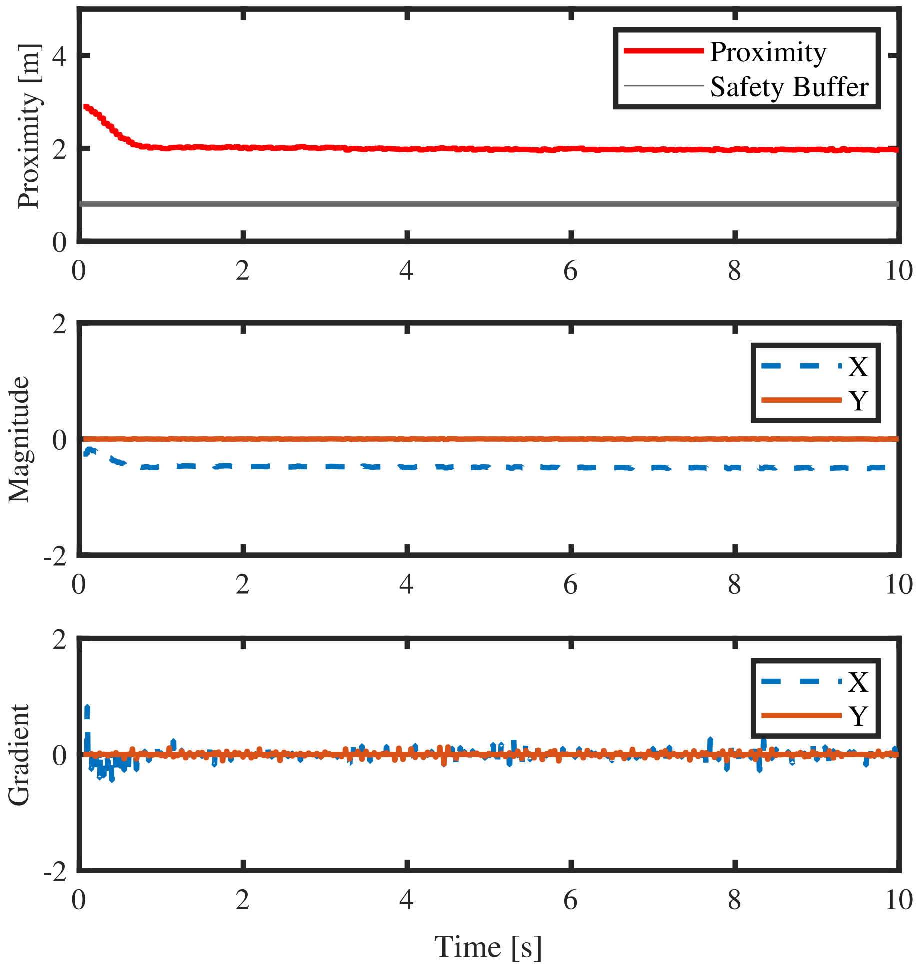


Figure B.4: Data from the 1 m wide gap experiments in the simulation used for quantifying proximity (in panel 1), pilot authority (as repulsive magnitude in panel 2) and oscillation (as the repulsive vector gradient in panel 3).

Figure B.4 provides little extra information, besides the almost complete lack of oscillation as the UAV becomes trapped in the local minima, indicated by the flat response after 1 s in panels 1 and 2.

B.2.2 1.4 m Gap

Atrium (Real) Data

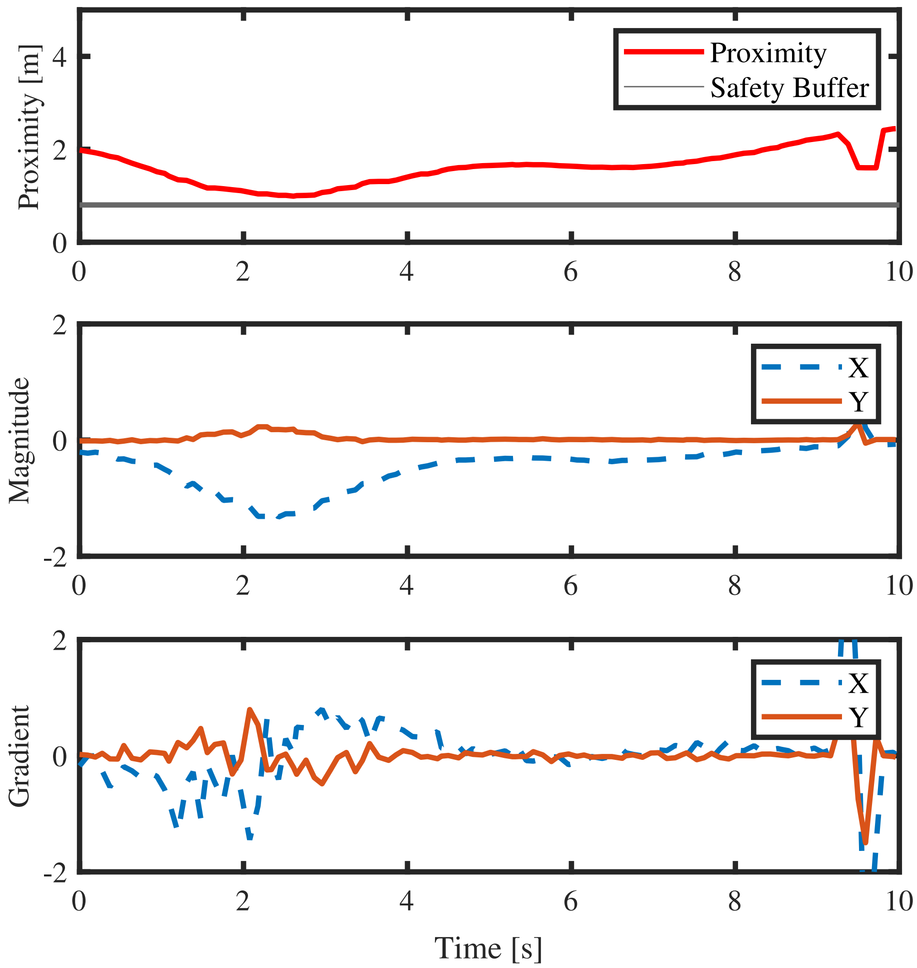


Figure B.5: Data from the 1.4 m wide gap experiments in the atrium used for quantifying proximity (in panel 1), pilot authority (as repulsive magnitude in panel 2) and oscillation (as the repulsive vector gradient in panel 3).

In Figure B.5, the minimum proximity of 1 m is encountered at 3 s, whereafter the UAV is repelled from the obstacles on either side of the gap. Panel 2 reveals the repulsive magnitude is much greater in the x-direction than the y-direction, since the pilot is commanding a collision course in the positive x-direction. At 10 s, an obstacle behind the vehicle triggers a slight response.

Simulation Data

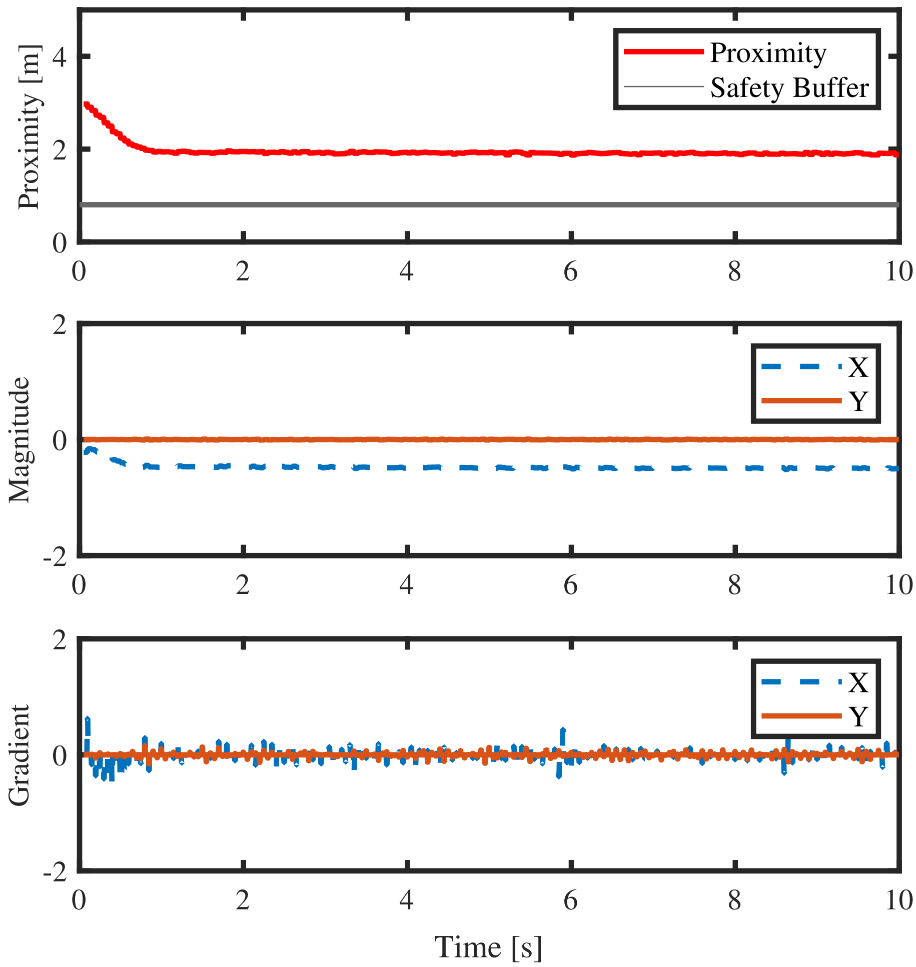


Figure B.6: Data from the 1.4 m wide gap experiments in the simulation used for quantifying proximity (in panel 1), pilot authority (as repulsive magnitude in panel 2) and oscillation (as the repulsive vector gradient in panel 3).

Figure B.6 shows that the repulsive magnitude and gradient are lower than the actual experiment since the UAV starts exactly in line with the centre of the obstacle, and there are no external obstacles or perturbations. Additionally, the UAV does not oscillate in the x-axis.

B.2.3 1.8 m Gap

Atrium (Real) Data

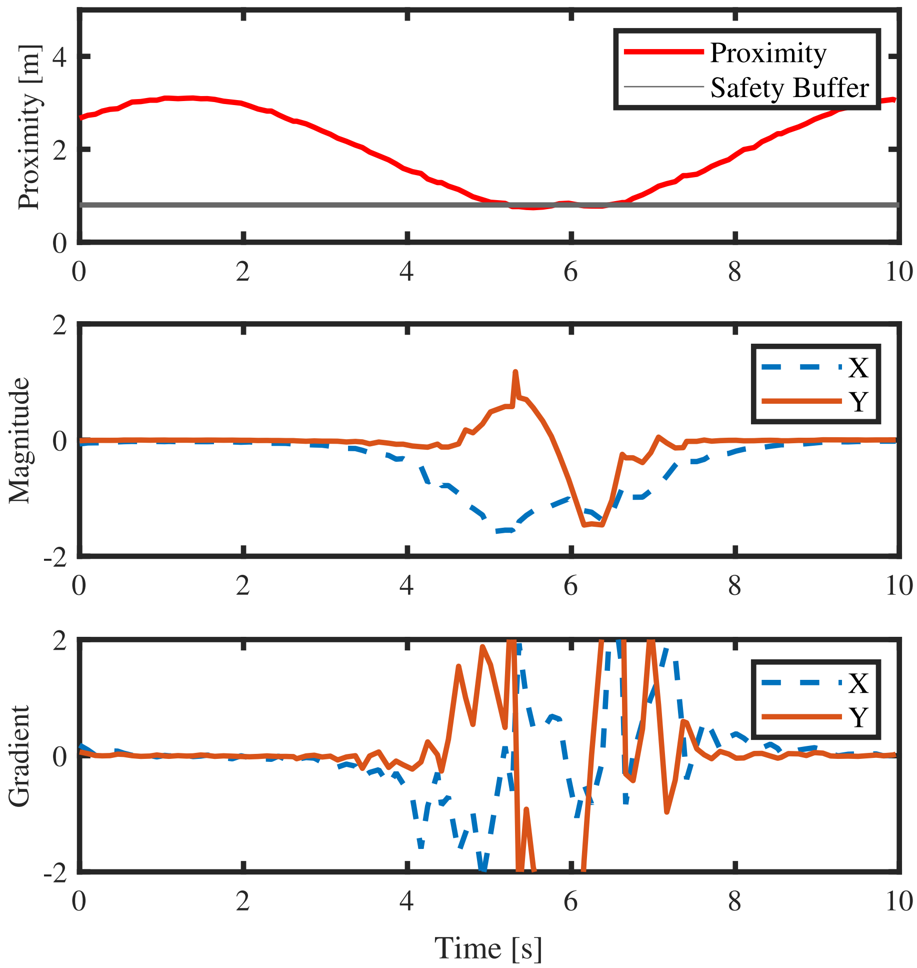


Figure B.7: Data from the 1.8 m wide gap experiments in the atrium used for quantifying proximity (in panel 1), pilot authority (as repulsive magnitude in panel 2) and oscillation (as the repulsive vector gradient in panel 3).

In Figure B.7 the minimum proximity occurs at 6 s where the safety buffer is breached, resulting in a strong repulsion in the negative x direction. The y-axis oscillation occurs as the walls either side of the gap push to center the UAV.

Simulation Data

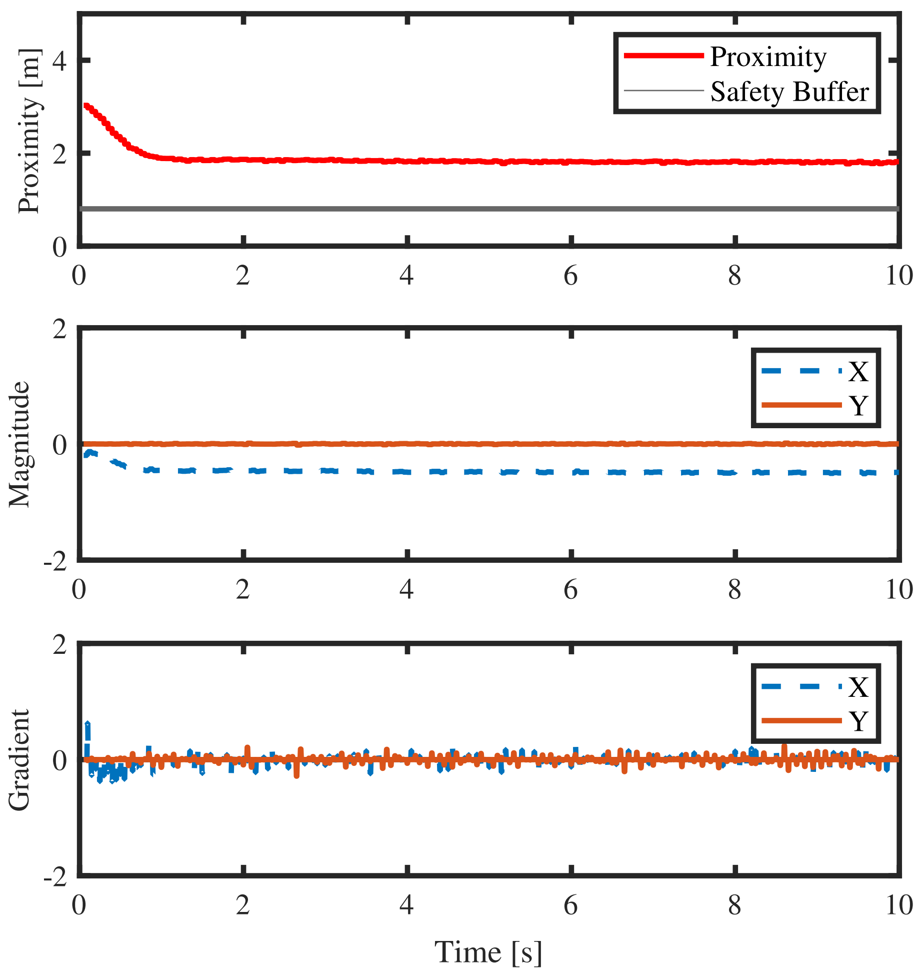


Figure B.8: Data from the 1.8 m wide gap experiments in the simulation used for quantifying proximity (in panel 1), pilot authority (as repulsive magnitude in panel 2) and oscillation (as the repulsive vector gradient in panel 3).

Figure B.8 shows how in contrast to the actual experiment, the UAV settles 2 m away from the gap. The repulsive force vector and gradient are lower than the actual experiment since the UAV starts exactly in line with the centre of the obstacle, and there are no external obstacles or perturbations.

B.2.4 2.2 m Gap

Atrium (Real) Data

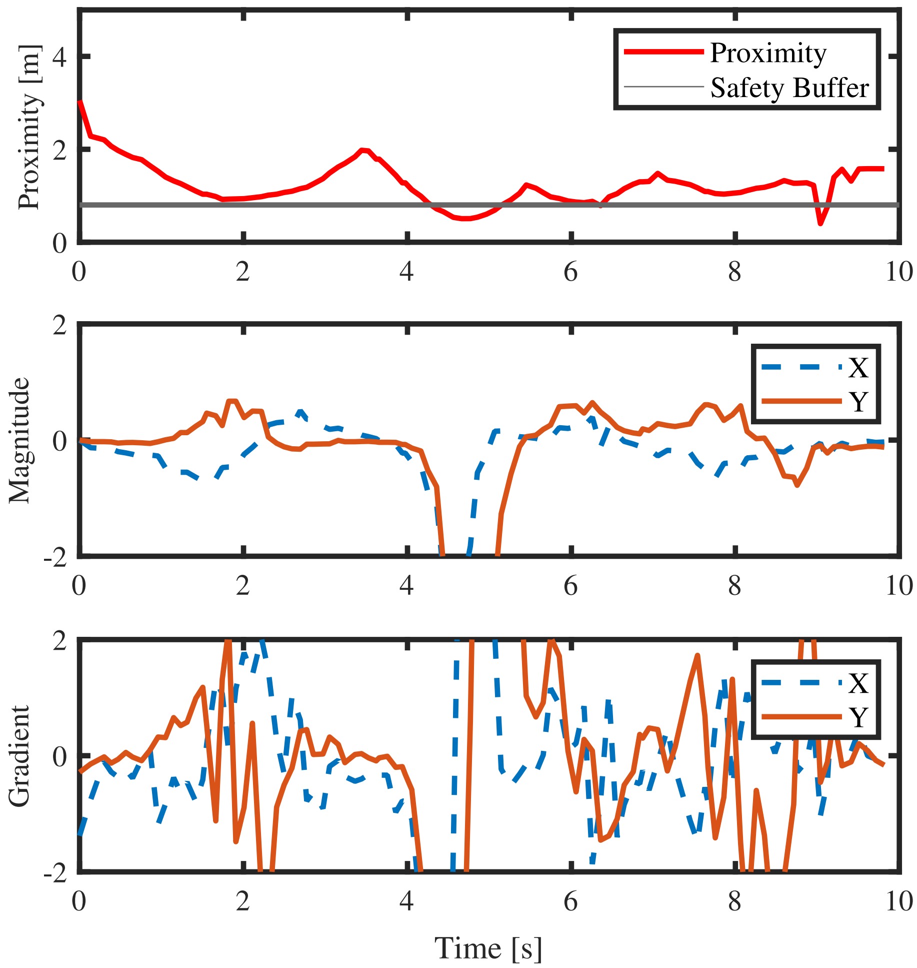


Figure B.9: Data from the 2.2 m wide gap experiments in the atrium used for quantifying proximity (in panel 1), pilot authority (as repulsive magnitude in panel 2) and oscillation (as the repulsive vector gradient in panel 3).

In Figure B.9 the UAV passes the gap at 2 s, then encounters a tree at 5 s and again at 6 s resulting in a large repulsive force and low temporary proximity. The encounters after passing the gap result in skewed data for comparison with the previous gap experiments.

Simulation Data

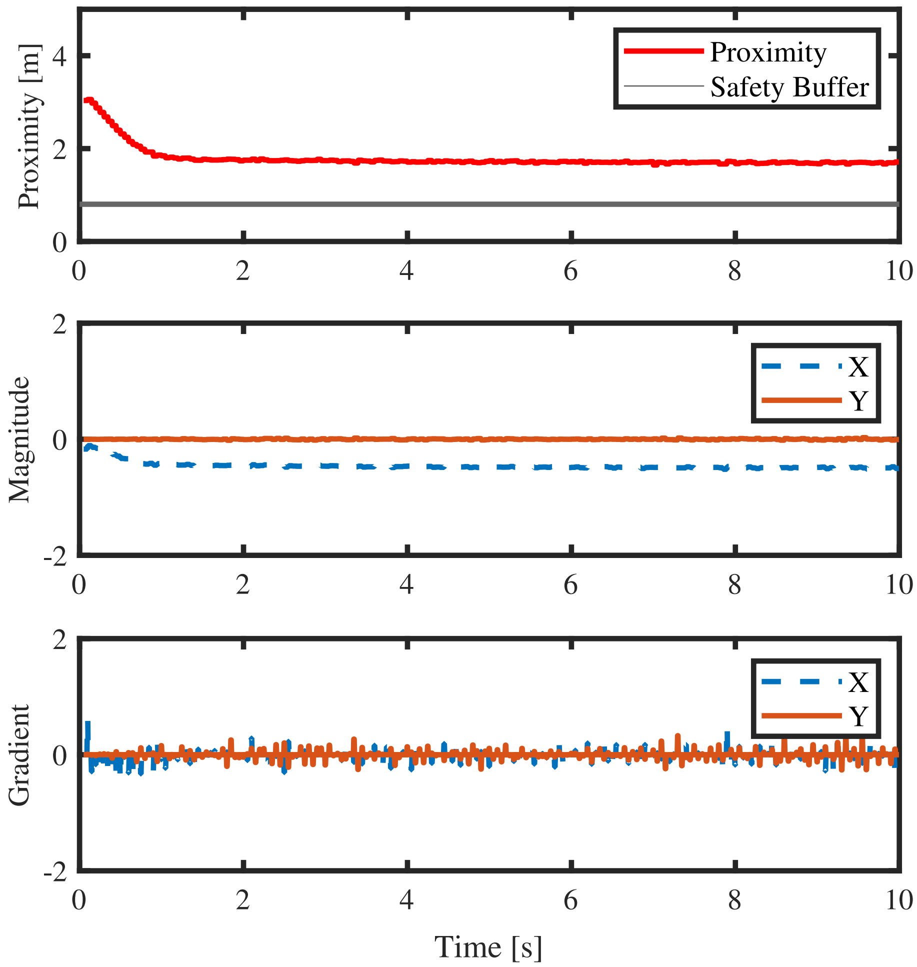


Figure B.10: Data from the 2.2 m wide gap experiments in the simulation used for quantifying proximity (in panel 1), pilot authority (as repulsive magnitude in panel 2) and oscillation (as the repulsive vector gradient in panel 3).

Figure B.10 shows that the UAV does not pass through the gap in this case, instead settling into a local minima 1.8 m away from the gap.

B.3 Wall Experiment

Atrium (Real) Data

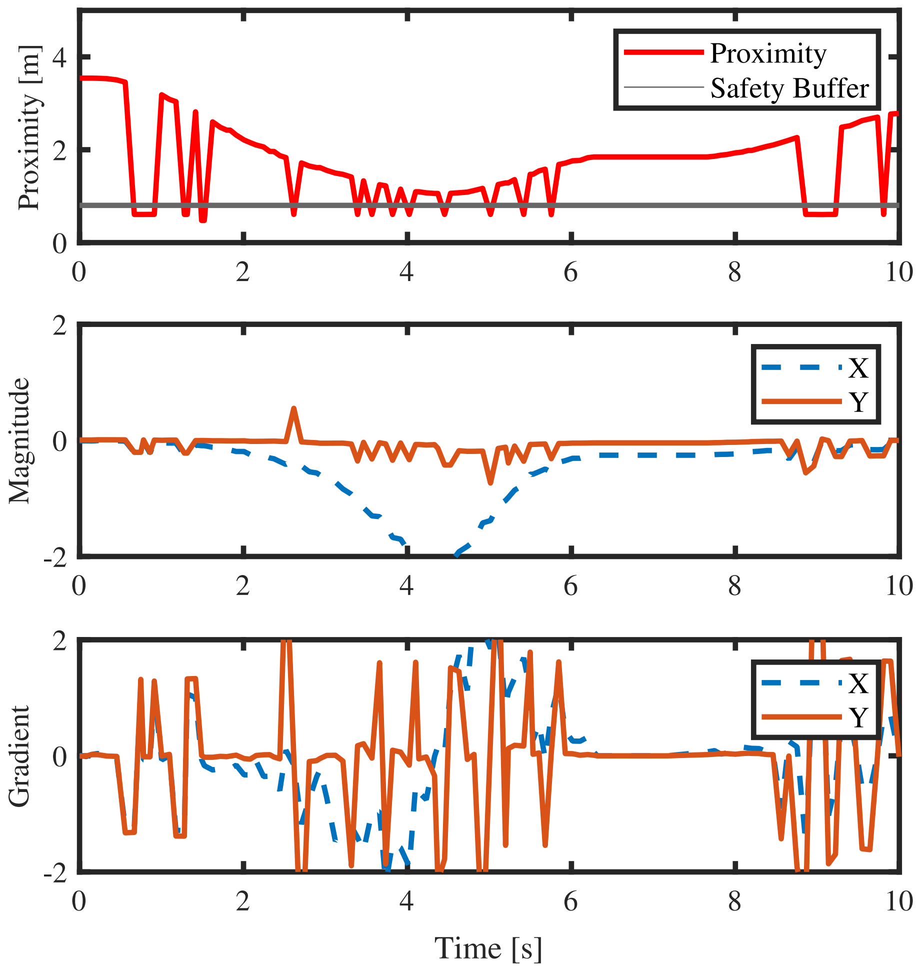


Figure B.11: Data from the wall experiments in the atrium used for quantifying proximity (in panel 1), pilot authority (as repulsive magnitude in panel 2) and oscillation (as the repulsive vector gradient in panel 3).

Key in Figure B.11 is that the LiDAR exhibits a large degree of noise shown as spikes in the proximity data. The true minimum proximity follows the parabolic shape as in previous experiments. The LiDAR misreadings result in spikes in the repulsive vector in the y-axis. However, the trajectory is not affected by these short errors.

Simulation Data

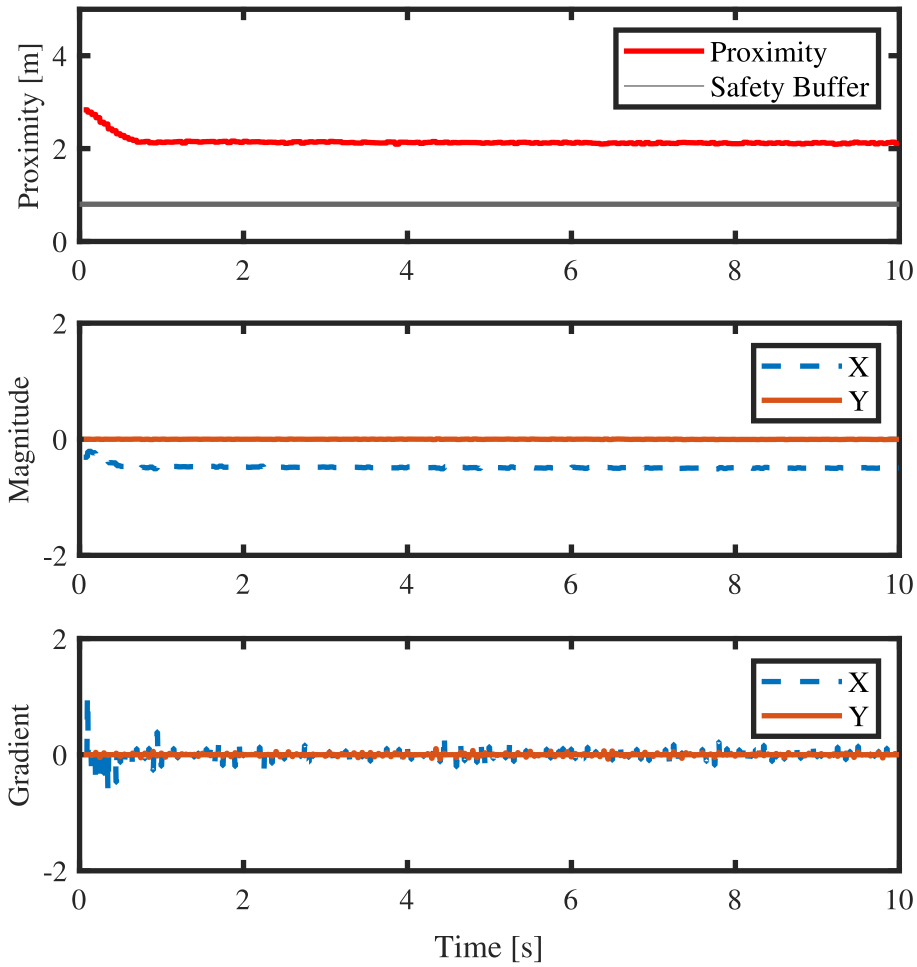


Figure B.12: Data from the wall experiments in the simulation used for quantifying proximity (in panel 1), pilot authority (as repulsive magnitude in panel 2) and oscillation (as the repulsive vector gradient in panel 3).

Unlike the actual experiment, the simulated UAV settles 2 m away from the obstacle without rebounding, as shown in Figure B.12.

B.4 Tunnel Experiment

B.4.1 High Gain

Tunnel (Real) Data

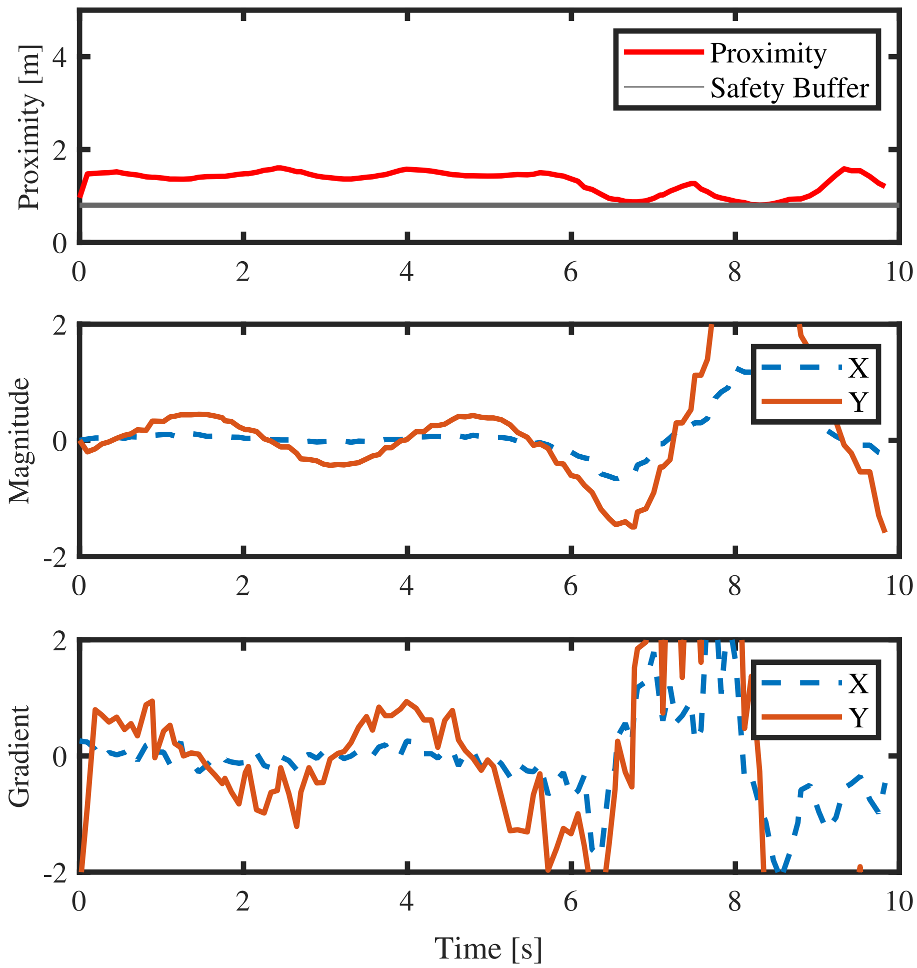


Figure B.13: Data from the high gain tunnel experiment in the mock mine used for quantifying proximity (in panel 1), pilot authority (as repulsive magnitude in panel 2) and oscillation (as the repulsive vector gradient in panel 3).

Figure B.13 shows that the repulsive command in the y-axis is oscillatory, and after the UAV passes the obstacle at 6 s, the oscillation amplitude increases. The obstacle perturbation causes a disturbance that is undamped in the y-axis.

Simulation Data

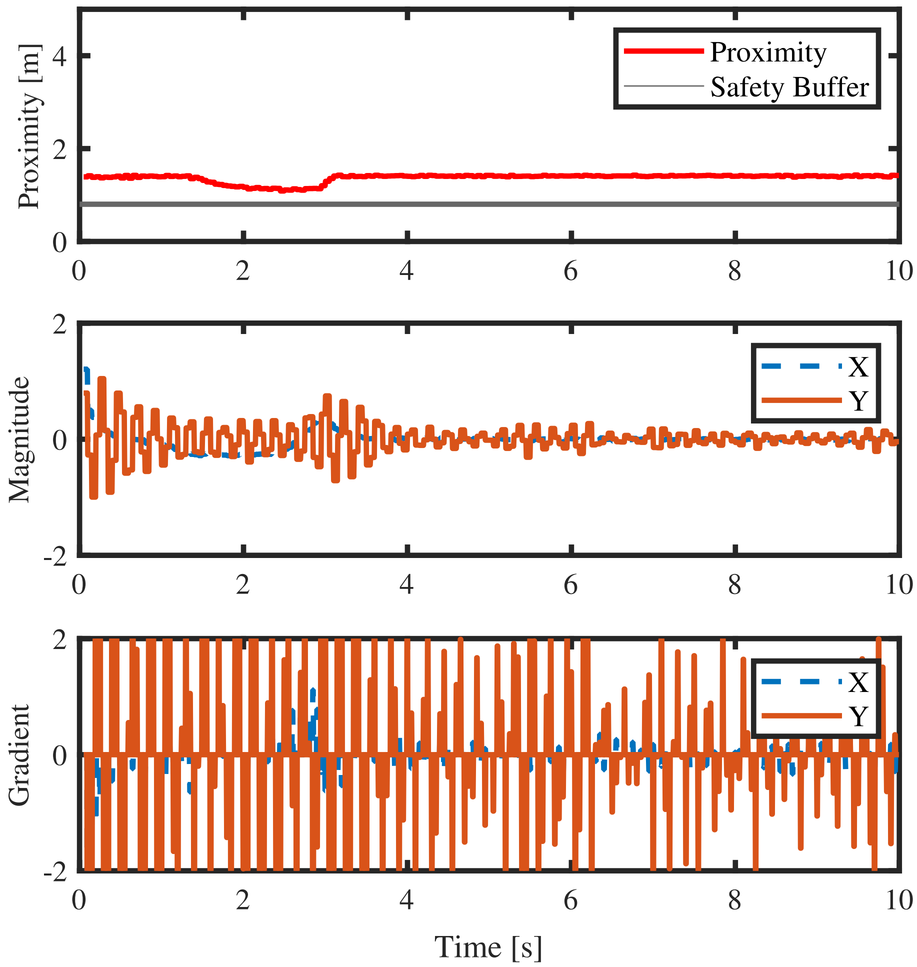


Figure B.14: Data from the high gain tunnel experiment in the simulation used for quantifying proximity (in panel 1), pilot authority (as repulsive magnitude in panel 2) and oscillation (as the repulsive vector gradient in panel 3).

In Figure B.14, it is observed that the narrow tunnel results in damped oscillatory commands in the y-axis, while the obstacle perturbation results in a disturbance at 3 s.

B.4.2 Low Gain

Tunnel (Real) Data

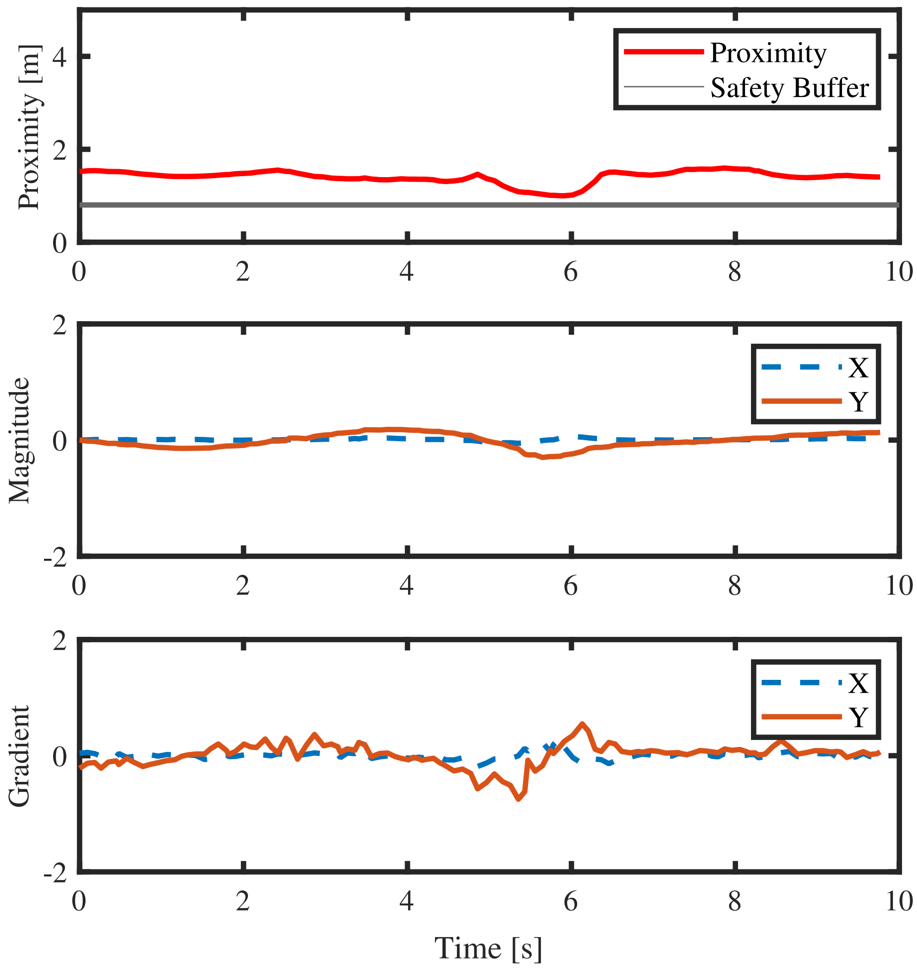


Figure B.15: Data from the low gain tunnel experiment in the mock mine used for quantifying proximity (in panel 1), pilot authority (as repulsive magnitude in panel 2) and oscillation (as the repulsive vector gradient in panel 3).

Figure B.15 shows that some oscillatory behaviour is visible in the y-axis, but at a lower amplitude than the high gain experiment. Passing the obstacle at 6 s does not result in a noticeable disturbance in the y-axis, but the x-axis of the repulsive command is perturbed slightly on the approach and retreat of the UAV to and from the obstacle.

Simulation Data

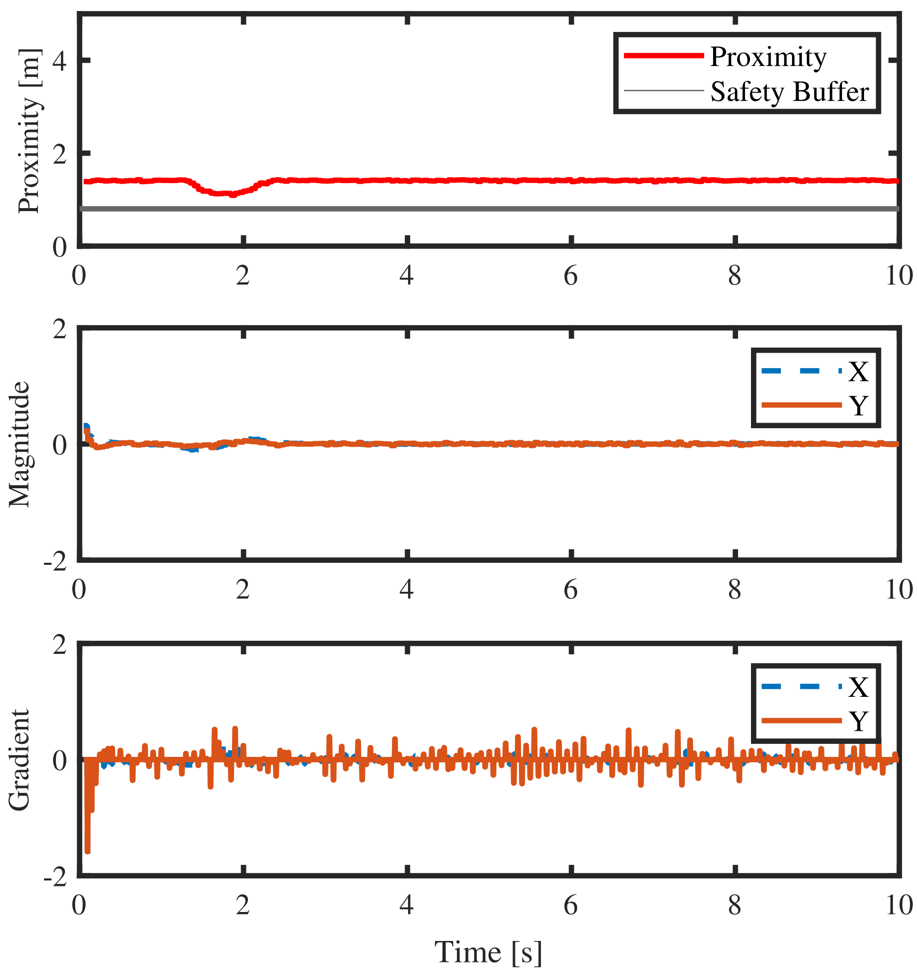


Figure B.16: Data from the low gain tunnel experiment in the simulation used for quantifying proximity (in panel 1), pilot authority (as repulsive magnitude in panel 2) and oscillation (as the repulsive vector gradient in panel 3).

In Figure 7.9b, after passing the obstacle at 2 s, the proximity and oscillatory behaviour is limited, and presents the ideal behaviour when flying in a tunnel. The pilot authority is also good, since the repulsive magnitude in panel 2 is low.