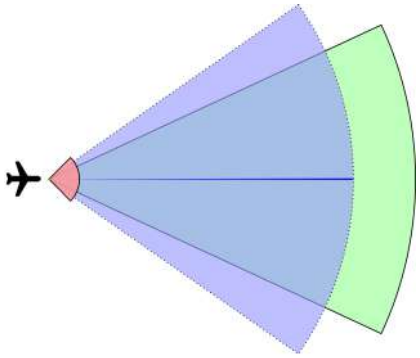
TÉCNICO LISBOA



Modeling and Optimization of an Obstacle Detection System for Small UAVs

Nuno Miguel Portal Alturas

Thesis to obtain the Master of Science Degree in

Aerospace Engineering

Supervisor: Prof. André Calado Marta

Examination Committee

Chairperson: Prof. Paulo Jorge Coelho Ramalho Oliveira
Supervisor: Prof. André Calado Marta
Member of the Committee: Prof. Bruno João Nogueira Guerreiro

January 2021

Dedicated to Catarina and Tiago.

Acknowledgments

Firstly, I want to express my gratitude to Prof. André Marta for his continuous support, guidance, availability and patience throughout the development of this work.

I want to thank my family because of the love and support they gave me through the years and my colleagues and friends for making these years more fun and fruitful.

Lastly, I want to thank my father for helping me proofread this work.

Resumo

Nos últimos anos, o mercado de veículos aéreos não tripulados (UAV) foi-se expandindo e diversificando substancialmente. Este trabalho apresenta uma solução para o aumento da segurança durante o voo de UAVs pequenos de asa fixa, no que diz respeito à deteção de obstáculos durante o voo. Esta tarefa foi alcançada através da realização de um estudo de mercado sobre os sensores disponíveis, de forma a encontrar os mais adequados a equipar no UAV e modelando-os, para que estes modelos pudessem ser integrados em simulações de deteção e prevenção de colisões. Também foi feito um estudo sobre diferentes filtros de rastreio e técnicas de fusão de sensores, onde o Filtro de Kalman de Medição Convertida e a técnica de Filtro Ponderado foram consideradas as melhores opções. Nas simulações realizadas, o método de evasão utilizado foi o dos Campos Potenciais por ser computacionalmente barato e por fornecer soluções viáveis em tempo real. Vários estudos paramétricos foram conduzidos para testar o desempenho dos sensores estudados e para averiguar como os diferentes parâmetros do sensor afetam o sucesso da evasão de obstáculos. Nestes testes, as características dos sensores foram consideradas adequadas para evitar obstáculos quando integrados em pequenos UAVs. Também foi realizado um estudo de otimização, utilizando um algoritmo genético, para encontrar, para várias configurações de sensores, as orientações que resultassem no melhor desempenho para um conjunto de cenários gerados aleatoriamente e, no geral, o sistema desenvolvido forneceu uma solução satisfatória.

Palavras-chave: Campos Potenciais, Algoritmo Genético, Filtro de Kalman, Conversão Imparcial, Fusão de Sensores, Estudo Paramétrico

Abstract

In the last years, the unmanned aerial vehicles (UAV) market has expanded and diversified substantially. This work presents a solution for the enhancement of safety during the flight of small fixed-wing UAVs, regarding the detection of obstacles during flight. This task was achieved by making a market study on available sensors to find the most suitable to equip a UAV and by modeling them, so that these models could be integrated into collision detection and avoidance simulations. A study was also made on different tracking filters and sensor fusion techniques, where the Converted Measurement Kalman Filter and the Weighted Filter technique were found to be the best options to implement. In the performed simulations, the used avoidance method was the Potential Fields for being computationally inexpensive and for providing feasible solutions in real time. Several parametric studies were conducted to test the performance of the studied sensors and to see how different sensor parameters affect the success of the obstacle avoidance. In these tests, the characteristics of the sensors were deemed adequate for avoiding obstacles when integrated into small UAVs. An optimization study was also conducted, using a genetic algorithm, to find the orientation of sensors, for different sets of sensors, that results in the best performance in a collection of random generated scenarios. It is shown that, overall, the developed system provided a satisfactory solution.

Keywords: Potential Fields, Genetic Algorithm, Kalman Filter, Unbiased Conversion, Sensor Fusion, Parametric Study

Contents

List of Tables

List of Figures

Nomenclature

Greek symbols

α	True azimuth.
β	Orientation of sensor.
$\boldsymbol{\nu}$	Observation noise vector.
$\boldsymbol{\omega}$	Process noise vector.
$\mathbf{\Phi}$	State transition matrix.
Δ	Variation.
Γ	Obstacle function.
λ	Bias compensation factor.
ω	Angular speed.
ϕ	Roll angle.
σ	Standard deviation.
θ	Angle between desired motion and obstacle.
$\tilde{\alpha}$	Azimuth error.

Roman symbols

α_{PF}	Weighing term.
$\hat{\mathbf{x}}$	State estimation vector.
a	Aceeleration vector.
B	Control-imput matrix.
$\mathbf{d}_0$	Vector pointing from obstacle to UAV.
f	Force.
H	Measurement sensitivity matrix.

K Kalman gain matrix.

L Rotation matrix.

m Desired direction of motion.

P State estimation covariance matrix.

Q Accumulated uncertainty matrix.

R Measurement error covariance matrix.

$\mathbf{s}_{dir}$ Swerling direction.

u Input vector.

v Velocity vector.

z Measurement vector.

$\tilde{r}$ Range error.

A, B Position of vehicles A and B.

a, b, c Obstacle size parameters.

a_1, a_2 Filter parameters.

acc Desired accuracy.

D Used measurement in filter.

d Distance.

e Filter's smoothing factor.

f Focal length.

$f(\theta)$ Fitness function.

G LIDAR gain.

g Standard acceleration due to gravity.

L Laser rangefinder measurements.

lb, ub Lower and upper bounds.

N Number of redundant sensors.

n Number of filter cycles.

N_1, N_2 Laser rangefinder norms.

O Measured position variation.

P Position of UAV.

p, q, r Obstacle shape parameters.

P_{close} Closest point of global path.

P_{next} Position of next waypoint.

r True range.

R_a Action radius.

R_c Collision radius.

R_d Detection radius.

R_s Safety radius.

S_{max} Maximum intensity.

t Time instant.

V Stereo vision measurements.

v Cartesian velocity.

W Filtered width.

W_L Laser rangefinder filter.

W'_L First term of laser rangefinder filter.

W_V Stereo vision filter.

x, y, z Cartesian position.

Subscripts

0 Initial.

α Azimuth.

A, B Vehicles A and B.

at Attractive.

CPA Closest point of approach.

cut_off Cut off.

k Current discrete measurement.

$k-1$ Previous discrete measurement.

m Measurement.

obs Observation.

rep Repulsive.

x, y Cartesian components.

C Camera frame.

L Local frame.

r Range.

s Standard.

u Unbiased.

Superscripts

\+ *A posteriori* estimate.

\- *A priori* estimate.

-1 Inverse.

T Transpose.

Glossary

ADS-B	Automatic Dependent Surveillance-Broadcast
ANAC	Autoridade Nacional da Aviação Civil
ATC	Air Traffic Control
BFGS	Broyden-Fletcher-Goldfarb-Shanno
BP	Back-projection
BVLOS	Beyond Visual Line of Sight
CAGR	Compound Annual Growth Rate
CAN bus	Controller Area Network
CCD	Charge-coupled Device
CMKF	Converted Measurement Kalman filter
CMOS	Complementary Metal-oxide Semiconductor
CPA	Closest Point of Approach
CR	Close Range
DARPA	Defense Advanced Research Projects Agency
DTM	Digital Terrain Models
EASA	European Union Aviation Safety Agency
EKF	Extended Kalman Filter
FAA	Federal Aviation Administration
FOV	Field of View
GA	Genetic Algorithm
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
HALE	High Altitude Long Endurance
HD	High Definition
IMC	Instrument Meteorological Conditions
IMU	Inertial measurement unit
INS	Inertial Navigation System
IR	Infra Red
KF	Kalman Filter
LADP	Low Altitude Deep Penetration

LALE	Low Altitude Long Endurance
LIDAR	Light Detection and Ranging
LOS	Line of Sight
MALE	Medium Altitude Long Endurance
MILP	Mixed Integer Linear Programming
MRE	Medium Range Endurance
MR	Medium Range
MTOW	Maximum Take-Off Weight
NASA	National Aeronautics and Space Administration
PDF	Probability Density Function
PID	Proportional-Integral-Derivative
PN	Proportional Navigation
PWM	Pulse Width Modulation
RADAR	Radio Detection And Ranging
RA	Resolution Advisory
RMS	Root Mean Square
SBAS	Satellite Based Augmentation System
SNR	Signal to Noise Ratio
SR	Short Range
S&A	Sense and Avoidance
T-CDMA	Time-Code Division Multiple Access
TA	Traffic Advisory
TCAS	Traffic Alert and Collision Avoidance System
UAV	Unmanned Aircraft Vehicle
UCAV	Unmanned Combat Air Vehicle
UKF	Unscented Kalman Filter
VLOS	Visual Line of Sight
VTOL	Vertical Take-off and Landing
WP	Waypoint

Chapter 1

Introduction

1.1 UAV Market Overview

Like many other technologies, Unmanned Aircraft Vehicles (UAVs) were initially developed for military purposes and have since made their way into the civil domain. Their origin dates back to World War I when the Navy and Army experimented with aerial torpedoes and flying bombs [1]. Since then, the UAV market has expanded and diversified substantially. Nowadays, UAV applications include but are not limited to:

- Commercial Photography and Video (Figure 1.1 (a)): Due to their small size and agility, UAVs can be used to obtain video footage that otherwise would have been impossible to acquire. Moreover, employing UAVs in video production and broadcasting opens up numerous opportunities for new forms of content, enhanced viewer engagement and interactivity [2];
- Precision Agriculture (Figure 1.1 (b)): UAVs can be used to estimate and predict health of crops to design targeted fertilizer treatment plans in order to improve productivity, farm profits and environmental quality [3];
- Border Control (Figure 1.1 (c)): UAVs can also be used to track and recognize illegal activities, unwanted infiltrators and unauthorized trespassers of national borders by using cooperative surveillance [4];
- Deliveries (Figure 1.1 (d)): The interest in using UAVs for transportation and delivery of goods has been increasing. UAVs can avoid obstacles that ground vehicles cannot and therefore the use of UAVs allows for reductions in time, effort and cost [5].

The market is still growing and projections show that non-military UAV production will total 14.3 billion dollars in 2028, while totaling 4.9 billion dollars in 2019, presenting a Compound Annual Growth Rate (CAGR) of 12.6% in constant dollars [10].

UAV classification is important to differentiate existing systems, since each category has different legal regulations, and also commercial and operational purposes. There is a large number of metrics

(a) UAV capturing video footage [6]

(b) UAV examining a crop [7]

(c) UAV patrolling the US border [8]

(d) UAV delivering a parcel [9]

Figure 1.1: Some UAV applications

that have been used for UAV classification, such as maximum take-off weight (MTOW), size, operating conditions, or any combination of these and other characteristics. MTOW is a good metric for regulatory purposes since it correlates well with the expected kinetic energy imparted at impact, which in turn is considered to be the primary factor affecting safety of operations. Basing UAV classes on flight altitude may also be of interest since they will dictate to a degree collision avoidance requirements.

One possible classification of UAV is presented in Table 1.1, where the variety of UAVs and the multiple dimensions of differentiation are displayed [11].

Table 1.1: UAV categorization

Category	MTOW (kg)	Range (km)	Flight Altitude (m)	Endurance (h)
Micro	<5	<10	250	1
Mini	<10/25/30/150 [1]	<10	150/250/300	<2
Close Range (CR)	25-150	10-30	3000	2-4
Short Range (SR)	50-250	30-70	3000	3-6
Medium Range (MR)	150-500	70-200	5000	6-10
MR Endurance (MRE)	500-1500	>500	8000	10-18
Low Altitude Deep Penetration (LADP)	250-2500	>250	50-9000	0.5-1
Low Altitude Long Endurance (LALE)	15-25	>500	3000	>24
Medium Altitude Long Endurance (MALE)	1000-1500	>500	3000	24-48
High Altitude Long Endurance (HALE)	2500-5000	>2000	20000	24-48
Unmanned Combat AV (UCAV)	>1000	1500	12000	2

[1] varies with national legal restrictions

In the United States, from the 1,563,263 UAVs registered in 2020, 441,709 are used for commercial purposes and 1,117,900 are used for recreational purposes [12]. The Federal Aviation Administration (FAA) conducted a survey to get a snapshot of the UAV categories owned by the general population [13]. By extrapolating the data obtained in this survey to the total commercial and recreational UAVs registered in the United States, we get the distribution presented in Figure 1.2.

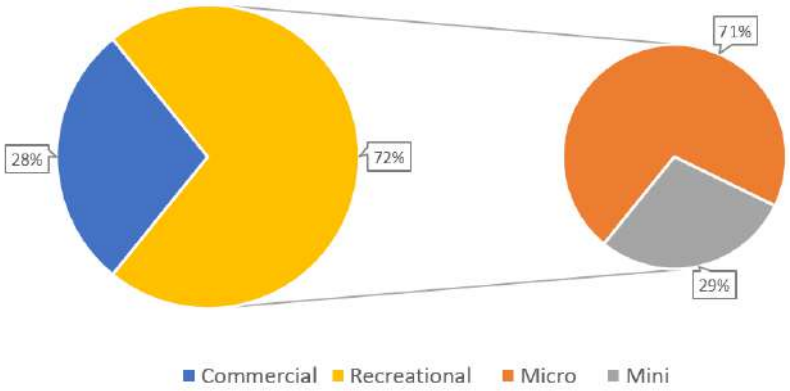


Figure 1.2: Distribution of UAVs in regard to use and category

There is not available data on the quantity of UAVs owned by the United States military but a forecast indicates that the USA will acquire over 1000 UCAVs in the next 10 years, which is a much smaller number than the 1,563,263 registered small UAVs [14]. The image and the data show the great interest in small UAVs from the general public, which surpasses, in volume, the commercial and military branches, thus motivating the study and development of systems to improve these micro and mini UAVs.

UAVs can also be classified based on their level of autonomy under the three following categories:

- Remotely piloted: A certified pilot controls the UAV either within line of sight (LOS) or with feedback from the UAV sensors;

- Remotely operated: The UAV is given high-level commands (waypoints, objects to track, etc.) but the decision making is delegated to a trained operator who monitors its performance;

- Fully autonomous: The UAV is given general tasks and is capable of determining how to accomplish them, even at the face of unforeseen events. If any faults occur, it can also react accordingly.

Regarding wing configuration, UAVs can have fixed wing or rotary wing. The configuration must be chosen according to the operational mission at hand. On the one hand, UCAVs may be required to operate at a higher speed, so they are likely to have a fixed wing with a low aspect-ratio and take-off from a long runway or be air-launched. On the other hand, the majority of civilian uses for UAVs will require the air vehicle to fly at speeds lower than 50 knots for much of its mission and many will require or benefit from the ability to hover, thus a rotary wing configuration is preferred [15].

1.2 Sensing Systems' Applications

Sensing technology has become very popular and has been used for a wide variety of purposes. In this section, a brief summary of two of these applications is made.

Back in the 1990's, mobile robot missions to the surface of Mars were being planned and, to prepare for these missions, several prototype rovers were constructed to be tested in desert fields. One of these prototypes was the Rocky 7 [16] (Figure 1.3 (a)). To provide a reliable measurement of the vehicle heading, this rover was equipped with sun sensors, accelerometers to determine sensor tilt and an on-board clock, since the Mars' magnetic field is too negligible to use a magnetic compass. For hazard avoidance, navigation telemetry and science data, black and white CCD (charge-coupled device) cameras were used extensively. Images from pairs of these cameras were captured simultaneously as stereo pairs and were processed on-board to provide depth maps of the environment. Afterwards, the stereo images were automatically analyzed for abrupt changes in height or high-centering hazards [17].

An application closer to the objective of this work was the Defense Advanced Research Projects Agency (DARPA) Grand Challenge of 2005, a driverless car race in which several universities competed to have the shortest time in a 212 km of of-road course. During the race, the engineers could not contact the vehicle, which could only rely on its sensors to detect and avoid obstacles. In particular, the Stanley vehicle, the winner of the race [18] (Figure 1.3 (b)), was equipped with coupled laser, vision and RADAR sensors: the laser sensor had great accuracy and a short range sufficient for slow motion, the camera captured denser data but was unable to provide range data and the RADAR provided range data for a range of up to 200 meters, but at a level of coarseness far inferior to the laser measurements. All raw sensor data was geo-referenced by the unscented Kalman filter (UKF) position estimates. This filter is a non-linear version of the Kalman filter that asynchronously integrates data from the Global Positioning System (GPS) systems, the Inertial Measurement Unit (IMU), and the Controller Area Network (CAN bus) [19, 20].

(a) Rocky 7 Mars rover [16]

(b) Stanley vehicle with its five laser sensors on the roof [18]

Figure 1.3: Vehicles possessing complex obstacle detection and avoidance systems

1.3 Motivation

With the development and increasing numbers of civil UAVs it becomes apparent that their deregulated and mass use can affect negatively the operational security of air navigation as well as people's safety.

Portuguese legislation regarding the use and operation of these kind of vehicles already exists and the *Autoridade Nacional da Aviação Civil* (ANAC) is the competent Portuguese authority that has approved the current enforced regulations [21]. A brief summary of these directives include:

- UAVs can only operate during the day, in Visual Line of Sight (VLOS) operation, up to 120 meters (400 feet) above the surface;
- UAVs must keep a safe distance from people and patrimonial assets to avoid damage in case of an accident;
- Priority must always be given to manned aircraft;
- UAVs must fly with identification lights always on;
- There are forbidden areas around military bases, airports and aerodromes that must be respected;
- For a flight to happen near the vicinity of aerodromes, several requirements must be met, depending on its airspace, such as: a special clearance, direct contact with the aerodrome staff and the ability to measure its altitude;
- Near aerodromes, flights cannot exceed the altitude of the highest obstacle within a radius of 75 meters centered on the UAV;
- An authorization from ANAC is required for operation during the night, Beyond Visual Line of Sight (BVLOS) flight or if the UAV weighs more than 25 kg.

To complement these regulations, the *Decreto Lei 58/2018* was issued in 2018 [22]. This decree made mandatory to have liability insurance on every UAVs that weighted more than 900 grams and required the registration of every UAV that weighted more than 250 grams, but at the time of writing, these new laws have not yet been enforced. These Portuguese regulations were promulgated in anticipation to European laws that would be issued in the foreseeable future by the European Aviation Safety Agency (EASA), and indeed, since July 1, 2020, new European regulations were enforced. These are resemblant to the Portuguese regulations already in place and will also impose the online evaluation of remote pilots and registration of UAVs that weight more than 250 grams [23].

For the UAVs to perform the functions described in subsection 1.1 efficiently, autonomous and BVLOS flight is essential, which is already foreseen by the Portuguese law. Furthermore, for an autonomous and BVLOS flight to be safe and effective, a reliable Detection and Avoidance system is needed. There are already numerous proposals for avoidance algorithms and sensor layouts but adapting these systems to certain UAV characteristics and keeping the cost low continues to be a challenge.

The present work is specifically aimed at fixed-wing mini UAVs (MTOW < 25 kg), due to the big share of the market that this category occupies (as seen in section 1.1) and due to their versatility and low-cost. Despite the bigger share of micro UAVs, weighing less than 5 kg makes them less qualified for outdoors operation. An example of a representative UAV, to which this work is aimed, is the AR4 (Figure 1.4), an autonomous, fixed-wing and mini UAV designed and manufactured by the Portuguese company Tekever. This aircraft has a maximum take-off weight of 4 kg, an endurance of 2 hours, a maximum speed of 15 m/s and it is hand launched for take-off [24].

Figure 1.4: Tekever AR4 UAV [24]

1.4 Objectives and Deliverables

With the rapid development of UAVs in recent years and the multiplicity of missions in which UAVs are being used for, new areas of research regarding these vehicles have been surfacing. An overview of some of these domains is displayed in Figure 1.5. This thesis will focus on the autonomy subsystem, specifically on the obstacle detection and sensor fusion aspects.

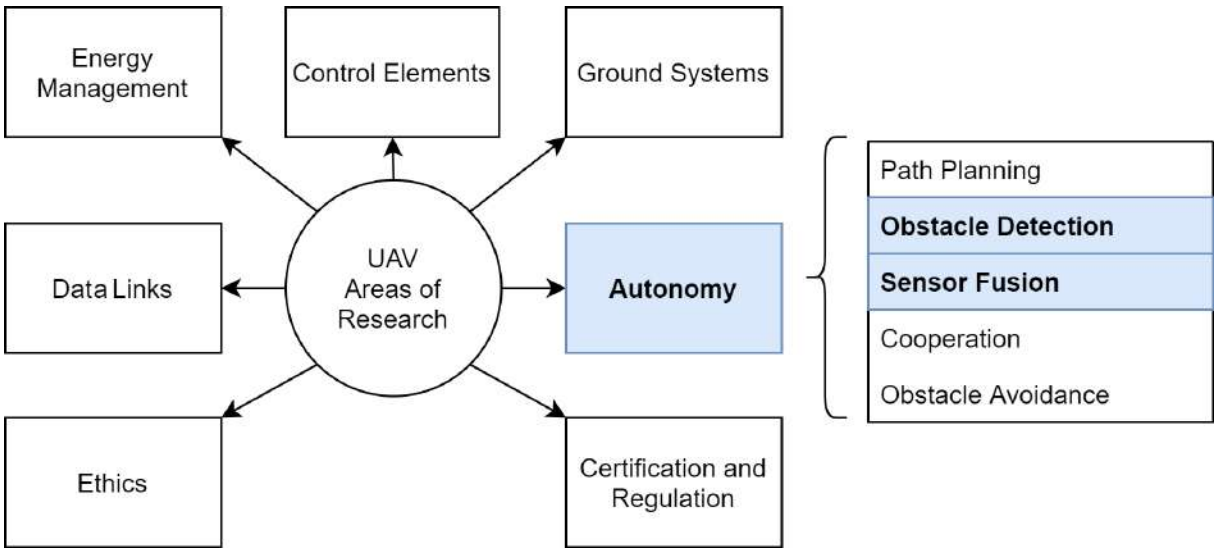


Figure 1.5: UAV areas of research

The main goal of this thesis is to improve the safety on low-cost fixed-wing mini UAVs regarding the detection of obstacles during their flight. It is part of an extensive obstacle detection and collision avoidance system, representing a two-stage "sense" and "avoid" problem, being the work focused solely on the former.

During the detection stage, the acquisition of information enables the UAV to detect threatening situations like proximity to ground obstacles, proximity to sensitive infrastructures, and route of collision with other manned or unmanned aircraft. This stage requires a set of sensors and communications devices to be available and the acquired information to be preprocessed and integrated (sensor fusion) to allow the detection of safety issues. The problem is divided in the following sub-goals:

1. Study existing rangefinders and chose which ones are to be integrated in the UAV platform to perform the detection of obstacles;
2. Chose suitable sensor fusion techniques to get the appropriate data from the several sensors for the detection of safety issues;
3. Evaluate the performance of the developed system in a simulated space to determine the drawbacks and advantages of the proposed methods;
4. Determine optimal sensor configurations that yield the best obstacle avoidance success rate.

In order to meet these proposed objectives, there is a set of deliverables that are expected to be outputted, namely:

1. Virtual simulation tools to evaluate the performance of the system configuration;
2. A methodology for getting an optimal configuration of systems adapted to the mission requirements and the aircraft's characteristics;

It is to be assumed that the obstacle identification data is to be fed to an existing system that will deal with the "avoidance" stage and take the appropriate evasive maneuvers.

1.5 Thesis Outline

The remainder of this thesis is organized as follows:

In chapter 2, each one of the blocks of the architecture of the sense and avoidance system is explained and a description of various cooperative and non-cooperative sensors is made by disclosing how each one acquires its data and by presenting a couple examples in which that sensor can be used in different fields of study. After introducing all the considered sensors, a comparative analysis is made in regards to several critical parameters. Lastly, the architecture of the proposed solution is presented.

Chapter 3 presents a benchmark of sensors commercially available and a quantitative comparison between them regarding several factors, which include range, field of view, cost and weight, is made so that the most suitable are chosen to be modeled.

Chapter 4 introduces the different Kalman filters that are used to track the obstacles detected by the different sensors and discloses the different ways in which each sensor is modeled in the simulated space. Finally, an approach on how to model the obstacles in the simulated environment is suggested.

Chapter 5 addresses different techniques to fuse the data from the used sensors that provide redundant data and the most suitable is chosen.

Chapter 6 addresses different possible collision detection and avoidance algorithms and the chosen geometric collision detection method and the potential fields method are described in detail.

In Chapter 7, several collision scenarios used to study the performance of the sensors are described and numerous parameters associated with the dynamics of the UAV and with the Kalman tracking are determined so that the UAV behaves realistically in the simulated space and the tracking phase is ran correctly. After, a sensor parametric study is made in order to see how different sensor parameters affect the success of the obstacle avoidance.

Chapter 8 presents an optimization study to find the orientation of sensors that result in the best performance in a collection of random generated scenarios, for different sets of sensors. The genetic algorithm used to perform this optimization is also described and the choice to use this algorithm is also justified.

Lastly, chapter 9 presents the conclusions of this work and recommends future work to be developed.

Chapter 2

Obstacle Detection

In this chapter, an overview of a typical architecture of Sense and Avoidance (S&A) systems will be discussed and each module will be briefly explained. Then, an analysis on several sensors that are commonly used for obstacle detection will be made. These will be divided into two categories: cooperative and non-cooperative sensors. Cooperative sensors require the existence of other vehicles in the airspace to be equipped with the same kind of sensors for information to be exchanged between them so as to conclude the aircraft/object detection. In contrast, both ground and airborne objects without any communication equipment can be detected by non-cooperative sensors. Cooperative sensors tend to have a higher cost but function much better in Instrument Meteorological Conditions (IMC) [25].

Finally, a qualitative comparative analysis of these sensors will be made and their specifications will be presented in a table to get a better idea of each sensor's positive and negative aspects.

2.1 Architecture of S&A Systems

The process of obstacle detection and avoidance can be divided into several modules represented in the functional block diagram in Figure 2.1.

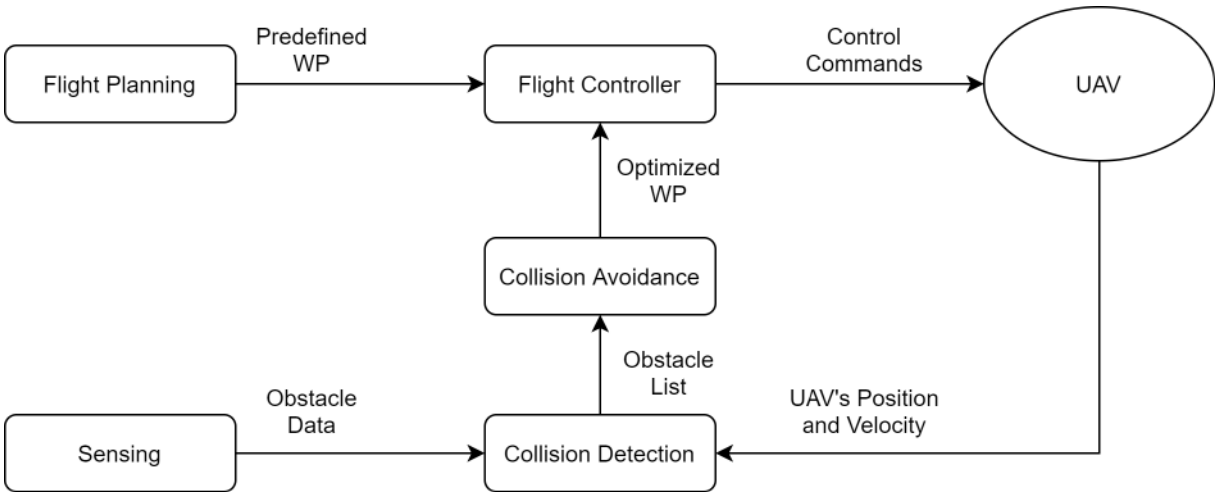


Figure 2.1: Block diagram of S&A systems' architecture

During the pre-flight phase, in the Flight Planning Module, the optimal path is computed, based on a set of predetermined waypoints (WP). To evaluate the best path, cost functions are formulated that take into consideration path lengths, flight altitude, danger zones, energy consumption, threats and flight time. To solve this optimization problem, several algorithms can be used such as Mixed Integer Linear Programming (MILP) which is effective despite being computationally demanding. However, this is not problematic, as this algorithm can be executed in high-end computers during the pre-flight phase, when there are no hard limitations on computational power and time [26]. Non-deterministic algorithms are also gaining popularity such as the Genetic Algorithm (GA), where the possible WP are comparable to genes and the possible solutions are comparable to chromosomes, which are subjected to crossover, mutation and where the fittest are favored [27].

Once this phase is completed, the list of waypoints to follow will be loaded into the Flight Controller, which will issue control commands to the UAV actuators, so that the UAV can make the correct maneuver to proceed to the next WP.

The Sensing Module is responsible for searching continually for obstacles in the UAV's vicinity and sending the obstacle data to the Collision Detection Module at a constant frequency to be stored in a list. Each element of this list contains the obstacle three-dimensional position, height, velocity and radius of protection volume, which encompasses the obstacle and a safety margin.

In the Collision Detection Module, knowing the UAV's position, the distance to the obstacle can be computed and used to order the elements of the list from closest to farthest, since the closest obstacles pose the highest risk. However, if the obstacle is not static, its velocity must also be taken into account to define the priority order of the obstacles within the list.

If one of these poses a threat to the UAV's navigation, the Collision Avoidance Module will be triggered and it will identify possible solutions to avoid the danger by using one of several algorithms. When designing this module, several aspects must be considered, such as the maneuverability of the UAV, the surrounding environment, safety, cost of the path and energy to be spent in the segment of leg (path between current position of UAV and the next WP). These aspects are incorporated in the form of cost functions that need to be minimized, and in the form of constraints that a path must act in accordance with. Once a solution is found, the waypoint list is updated to feature the new optimized waypoint and passed on to the flight controller module to allow the tracking of the new path [28].

All this process (except for the Flight Planning phase) must be completed in a timely manner as this is a real-time system with a strict time constraint considering that the collision may not be avoided if the computations and actions take too long.

2.2 Cooperative Obstacle Sensing

2.2.1 Traffic Alert and Collision Avoidance System (TCAS)

This cooperative obstacle sensing system is mandatory on all large transport aircraft and has helped preventing many catastrophic accidents. Its surveillance is based on air-to-air interrogations broadcast

once per second by a mode S (select) transponder, which has a unique 24-bit identifier. Once the intruder's transponder receives this interrogation, it emits a response with its current altitude. When this response is received, the TCAS computes the range and the bearing of the intruder and if a collision is predicted to happen within the next 20 to 48 seconds (depending on the altitude), TCAS issues a traffic advisory (TA), which is presented in a display and communicated as a spoken message. The pilot is then supposed to search visually for the threat and prepare for an evasive maneuver. If the situation worsens (15 to 35 seconds before collision, depending on altitude), a resolution advisory (RA), which is coordinated through a data link to ensure that each aircraft maneuvers in a compatible direction, is issued and the pilot is commanded to turn the auto-pilot off and either climb or descent, following the spoken and displayed instructions [29]. The TA and RA regions are represented in Figure 2.2.

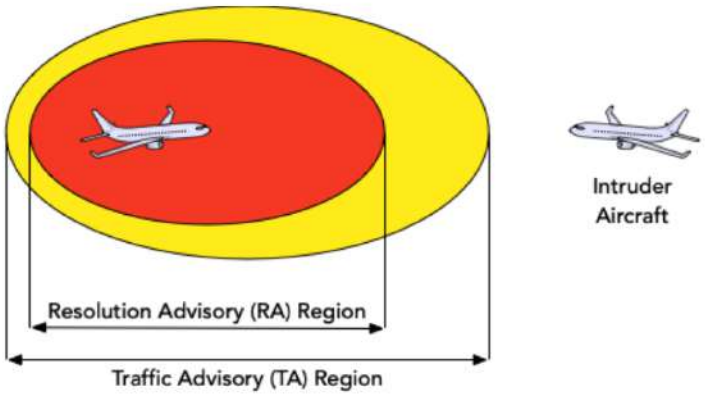


Figure 2.2: Traffic Advisory (TA) and Resolution Advisory (RA) regions [30]

Several versions of TCAS were developed over the years. TCAS I was intended to accommodate the general aviation (GA) community and the regional airlines. This system includes the traffic display, provides TAs and is mandatory on aircraft with 10 to 30 seats, although TCAS II may be installed instead. TCAS II is a more sophisticated system which provides the information of TCAS I, and also includes complex collision avoidance logic to provide vertical RAs to the flight crew. It is required internationally in aircraft with more than 30 seats or weighing more than 15,000 kg. TCAS III was envisioned as an expansion of the TCAS II concept to include horizontal RA capability. However, by 1995, it was determined that the concept was unworkable using available surveillance technology (due to the inadequacy of horizontal position information), and that horizontal RAs were unlikely to be invoked in most encounter geometries. In a continued effort to develop horizontal RA capability, a TCAS IV concept was studied in the mid 1990s. TCAS IV would use extended transponder messages to provide accurate position information from traffic navigation sources, such as inertial navigation systems or GPS, when requested. However, this concept was abandoned as ADS-B development started [31].

Nonetheless, using TCAS in UAVs is challenging as the payload is strictly limited. As an example, the TTR-4100 [32], a lightweight TCAS, requires 28 V DC to function and weighs 6.2 kg which already exceeds the Tekever AR4's MTOW. Moreover, TCAS has trouble in dealing with multiple aircraft and communication between manned and unmanned aircraft is also a drawback [25].

In references [33] and [34], the implementation of a collision avoidance system using TCAS in larger

UAVs is contemplated. The authors concluded that this system does not provide sufficiently precise directional information to use in unmanned aircraft as this system is intended to aid visual acquisition by indicating the proper sector to search out the cockpit, but does not provide sufficient bearing or altitude rate accuracy to support avoidance maneuvers by itself, requiring supplementary systems to provide precise directions.

2.2.2 Automatic Dependent Surveillance-Broadcast (ADS-B)

ADS-B, at its introduction, represented a completely new paradigm for air-traffic control and has gained so much prominence that since June 2020 every manned aircraft that weighs more than 5700 kg or has a maximum cruise speed greater than 250 knots will need to be equipped with ADS-B capabilities to be operated in European airspace [35], which is already required for operation in the U.S. airspace. Every aircraft that uses this system retrieves its position and velocity by using an onboard GPS receiver. Then the information is periodically broadcast by the transmitting subsystem ADS-B Out. This message is received and processed by the Air Traffic Control (ATC) as well as surrounding aircraft equipped with ADS-B In, as is depicted in Figure 2.3. With the knowledge of other aircraft's position, velocity and course, collisions can be avoided by the remote or auto-pilots [36].

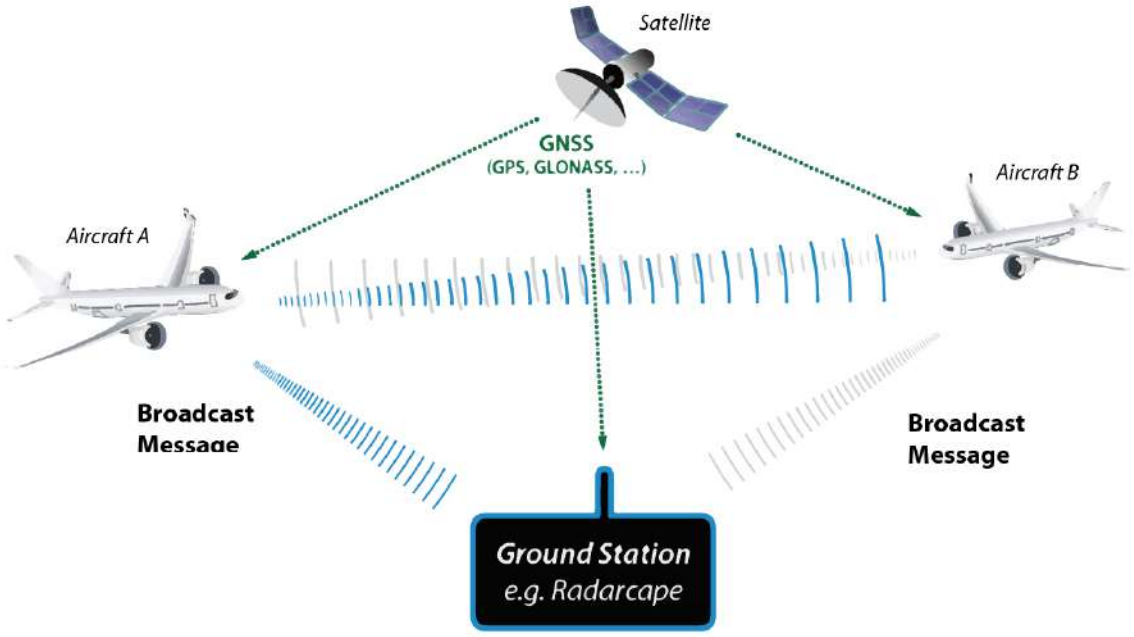


Figure 2.3: Operation of ADS-B system [37]

ADS-B is favorable for UAV S&A since it can provide accurate and reliable information of navigation variables, uses well-proven communication technology, possesses flexible structure for easy implementation and can even be integrated into existing traditional TCAS transponders. However, ADS-B is ineffective in the case of ground-based obstacles, such as terrain features, towers or power lines [25].

Lin et al. [38] developed a collision avoidance system to be used in helicopter/UAV cooperative disaster surveillance operations that makes use of quasi ADS-B systems employing the same general

operation principles of traditional TCAS. However, they made use of a different approach to TCAS, more suitable for helicopters and UAVs, by proposing an algorithm that uses a virtual sector in front of each aircraft instead of the conventional separation bubble algorithm, that issue TA and RA from the position and overlapped area of two aircraft's sectors. This was done as a result of the higher maneuverability of UAVs and helicopters, which can turn rapidly and violate the straight flight route assumed by conventional TCAS. Since this operation combines manned and unmanned aircraft, only the manned helicopters were supposed to change paths when alerted, while the UAVs were assigned a higher priority and only broadcast their position and following way-point through a quasi ADS-B out system.

In reference [39], an ADS-B radar system was designed to perform cooperative and non-cooperative obstacle sensing. This system broadcasts standard ADS-B messages to alert other aircraft, receives the same kind of information from others and uses the echo of its own transmissions to detect and track non-cooperative targets, such as terrain or aircraft without proper cooperative sensors. This system operates using standard ADS-B transponders which do not require any modification, despite an antenna array being required to detect the reflected signals. The needed components weigh less than 1.1 kg, have a size of 200 cm^3 and a power consumption of less than 40 W, which may be acceptable depending on the type and application of the UAV.

2.3 Non-cooperative Obstacle Sensing

2.3.1 Laser Rangefinder

Laser rangefinders are able to compute distances to obstacles by emitting a laser pulse and measuring the time it takes for the reflected beam to be detected, as represented in Figure 2.4. This kind of sensor is being used more and more due to its low price, portability and stability, despite being fairly impacted by illumination and atmospheric conditions [40].

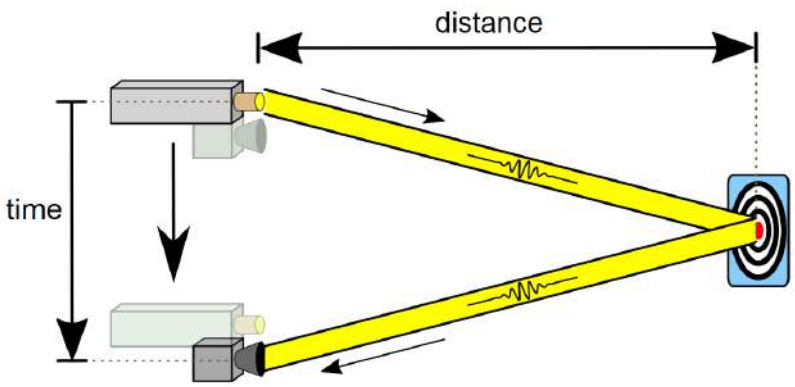


Figure 2.4: Operation principle of a laser rangefinder [41]

Saunders et al. [42] developed a static and dynamic obstacle avoidance for miniature air vehicles, by using a laser rangefinder, due to the weight and electrical power limitations in such small vehicles. The fixed sensor was mounted to detect obstacles directly in front of the UAV. In addition to a rapidly-exploring random tree algorithm used to generate waypoint paths around obstacles known *a priori*, a dynamic geometric algorithm is used to generate paths around detected obstacles. Regarding the sensing phase, there are two algorithms that run in turn to gather information on obstacle locations. The first scans the area of the UAV's expected trajectory and the second scans the dimensions of obstacles already partially detected in an attempt to provide enough information to avoid them.

In reference [43], an obstacle and avoidance system to perform in urban environments was implemented in a micro UAV by using a pair of miniature laser rangefinders and two proportional-integral-derivative (PID) controllers cooperating with an obstacle avoidance controller. For this micro UAV, using laser sensors was also the best solution due to its simplicity and power and weight constraints. In this case, using a single laser rangefinder was not feasible, as such configuration disables the possibility of flight in streets' canyons. As a result, the chosen configuration was to place both laser beams tangent to the UAV plane, forming a 60 degree "V" shape. The data was filtered to eliminate noise, disturbances and peaks caused by small obstacles that could bring instability to the system. Additionally, the fact that the sensors were fixed to the UAV body frame meant that a coordinate transformation was needed to obtain the real distances to the obstacles.

2.3.2 Light Detection and Ranging (LIDAR)

Laser rangefinders can also be used to achieve laser scanning, also known as LIDAR, if the sensor is attached to a scanning surface such as a servo or if an oscillating mirror deflects the laser beams. By doing so, a large number of distance measurements can be acquired for a small area, thus creating a 3-D point cloud, as represented in Figure 2.5.

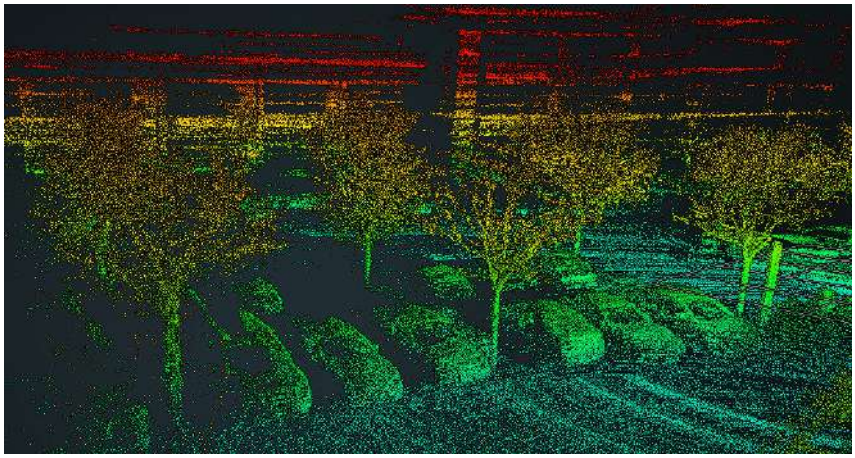


Figure 2.5: 3-D map of point cloud obtained from LIDAR [44]

To improve the surveying accuracy of the point cloud obtained from a LIDAR, its errors should be reduced as much as possible. The motion error associated with the movement of the UAV while the laser beam is rotating can be reduced by increasing the scanning speed (limited by hardware) or by

correcting the point cloud using position estimation via GPS or Inertial Navigation System (INS). These systems can be used to map the laser points into a desired coordinate system and a clustering algorithm can be used to cluster the point cloud with uneven density [45]. Additionally, the error resulting from the larger distance between consecutive detected points for farther obstacles, which results in smaller recorded dimensions, can be reduced by recording the maximum detected dimensions, comparing future measurements to these maxima and considering the largest of the two [46].

LIDAR sensors make use of the same technology as laser rangefinders, but require servos or oscillating mirrors to function, which results in an increase in weight and required power. The obtained measurements are precise and dense, although the use of these sensors may limit the UAV's speed, roll and pitch angles in order not to significantly affect the computed obstacle distances [40].

In reference [47], LIDAR sensors installed in UAVs were used for forest inventory. The information obtained from these sensors was fused with a GPS receiver, an IMU and a High Definition (HD) video camera. This fusion allowed for a more rigorous of the spatial accuracy of the final point clouds. LIDAR was ideal for this application due to the high temporal and spatial resolutions datasets and its low operational cost. The obtained point clouds allow for the use of multi-temporal surveys such as forest health and canopy closure monitoring.

In reference [48], LIDAR-equipped UAVs were used for building modeling, due to the flexibility of UAVs, which were capable of surveying the most inaccessible parts of buildings, and due to the great efficiency of laser sensors for geometric data acquisition, both in terms of time and precision. The UAV had also to be equipped with a GPS receiver and an IMU so that the two-dimensional point cloud obtained with the LIDAR sensor could be placed in its right position of the computed trajectory, thus forming a 3-D model of the building.

In the field of geology and geomorphology, in reference [49], airborne LIDAR and UAVs were used to analyze the kinematic evolution of the Montescaglioso landslide in southern Italy. The objective of this study was to compare pre- and post-event orthomosaics, and to evaluate the evolutionary behavior of the slope instability by analyzing the obtained digital terrain models (DTM). The study also showed the effectiveness of combining the two UAV-LIDAR methodologies. For this particular application, the LIDAR was not appropriate for near real-time monitoring due to the elaborate programming and processing phases.

2.3.3 Radio Detection and Ranging (RADAR)

Radar is a technology that transmits and receives electromagnetic waves in order to detect, locate, and gauge the speed of distant objects. Reflected waves are analyzed using mathematically derived equations, designed to overcome natural phenomena that affect the propagation of those waves. Radio waves move through the atmosphere at a known speed in a way that can be projected mathematically. An echo indicates that an object is out there somewhere, and analysis of the returning signal provides information about the direction of that object from the transmitter. By calculating the amount of time it takes for the radio signal to move from transmitter to target and back, the operator arrives at an accurate

measure of the distance. The operation of the RADAR is very similar to the LIDAR, where the main difference resides in the frequency of the emitted radiation. RADARs rely on the emission of radio waves while LIDARs predominantly use infra-red radiation [50].

In reference [51], the capability of an ultra-wideband software defined radar to produce high-resolution images of sub-surface landmine-like targets is demonstrated. The RADAR is supposed to be attached to a low-altitude UAV pointing to the ground for the radiation to penetrate the ground, considering the dispersive and refractive effects of the air-ground interface. To deal with these effects, a back-projection (BP) focusing algorithm was developed, which has the advantages of presenting flexibility and precision regarding phase and motion errors compensation.

In reference [52], a RADAR sensor was designed and its performance was evaluated in different scenarios in order to assess if all requirements regarding its real-time measurement capability, operational environment, payload constrains and air safety regulation were met. This RADAR was to be implemented in a smart UAV capable of high speed cruise and vertical take off/landing (VTOL). Since the atmospheric attenuation is the main reason for decreasing signal to noise ratio (SNR) in millimeter wave bands, the attenuation according to range was considered in the radar equation. The probability of detection for the designed RADAR was larger than 90% and the simulation results showed that, in case of radar error data, the safety margin boundary was well designed for securing the safety of the UAV.

Ajith Kumar and Ghose [53] propose a RADAR-assisted collision avoidance/guidance strategy for flight vehicles on low-altitude missions. The basic inputs of this system were the guidance command input, the vehicle state estimate provided by the on-board INS, and the range map provided by the obstacle detection active sensors, while the output was the commanded lateral acceleration. A classical proportional navigation (PN) guidance law was used, as it requires minimal computations and the needed inputs are easily available from the radar and the INS carried in the vehicle. The RADAR cut-off range R_c was carefully defined, as a large R_c could make the vehicle deviate too much from the desired trajectory because of radar returns from distant obstacles. At the same time, it had to be large enough to warn the vehicle well in advance of an impending collision. The actual selection of R_c had to depend on some knowledge of the distribution of the obstacles on the terrain.

2.3.4 Vision Sensors

UAVs can also detect obstacles by using vision cameras (or thermal cameras) and image processing equipment. The detection techniques should provide high detection probability for obstacles that can have the size of only a few pixels, while maintaining a low false alarm probability in the presence of noise and severe background clutter and working in a timely manner. Unlike ultrasonic and laser sensors, vision sensors have the advantage of being passive, while being lightweight, power efficient and inexpensive compared to scanning lasers, as these do not require mirrors or servos, therefore being more robust to vibration and shock. Vision does however rely on adequate texture and lighting of features in the scene for proper image processing and this image processing requires significant computational power. Also, the range accuracy decreases with distance squared from the camera [54].

In reference [55], a method was developed to use both thermal and vision cameras for search and rescue missions. Thermal cameras cannot detect humans at large distances because of low image resolution and quality. Likewise, human detection algorithms working with color imagery give best results at low distances. The technique presented in this paper first detects human-temperature silhouettes with the infra-red (IR) cameras and subsequently subjects the corresponding region to a human body classifier configured to allow for weak classifications in order not to miss potential targets. The small focus of the processed image allows for a high processing rate which in turn leads to the pruning of false positives and the collection of statistics about classified humans.

A vehicle detection and tracking system based on imagery collected by a UAV has been developed by Wang et al. [56]. In that work, the developed system uses consecutive frames to generate vehicle's dynamic information, such as positions and velocities, and four different modules were developed: the image registration module, where the movement of the UAV is subtracted from the obtained images to obtain only the real motion of the vehicle; the image feature extraction module, where the edge, optical flow and optical feature points are extracted from the images; the vehicle shape detecting module, where the previously obtained features are used to detect the boundaries of vehicles; and finally, the vehicle tracking module, where, once again, the optical flow and local feature point obtained in the previous module are used to track the vehicles, considering possible occlusions. An example of a vision system tracking moving vehicle is illustrated in Figure 2.6.

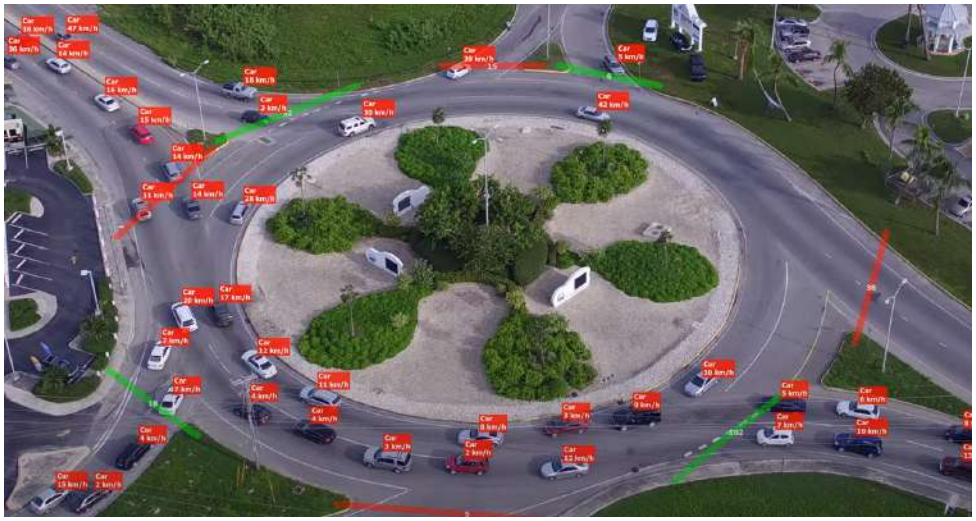


Figure 2.6: Vehicle tracking using UAVs equipped with vision sensors [57]

In the field of precision agriculture, vision sensors are also a great way to acquire field data in a fast and easy way. In reference [58], a Tetracam camera was mounted on a multi-rotor hexacoptor, while the data was processed with a photogrammetric pipeline to create orthoimages of the surveyed sites. These orthoimages were then used to extract several Vegetation Indices (VI) used to evaluate the characteristics of the crops. The image management, processing and exportation was achieved using the software PixelWrench2 (PX2), included in the camera. This software allowed the captured images to be exported as a triband image. However, the use of vision sensors in this field still entails several problems, as the well-established procedures to extract VI from remotely sensed imagery were

developed for airborne and satellite datasets. Also, to compute VI, reflectance values obtained from ground measurements by spectro-radimoters should be used. By only using low-cost cameras and on-board image processing, the calibrated radiance values cannot be obtained and the achieved results are mainly qualitative and not quantitative, due to being computed from raw indices that still give a decent description of the vegetation conditions [59].

The previous works rely on vision sensors to perform detection and tracking operations where the targets have known shapes and sizes, given the altitude of the flight. For our work, the target can be at any given distance to the camera and, because of that, have various sizes in the captured images. Since vision sensors can only provide two-dimensional images, computing the obstacle's distance can be challenging. The following examples provide solutions to this problem, by using complex image-processing techniques and by using stereo vision, where images are obtained from two different lenses. Fusing vision sensors with other sensors also solves this problem.

Carrillo et al. [60] developed a navigation system for a quad-rotor UAV by combining stereo vision with an INS. The adopted stereo odometry method consisted on searching for desired features on the left image and searching for the corresponding feature on the right image. Afterwards, the three-dimensional positions of the matched features are reconstructed using triangulation. This method can be used to obtain obstacles' range data or produce 3-D maps of the UAV's surroundings. In this particular work, it is used to determine the vehicle's relative motion in all six degrees of freedom, by comparing consecutive images. In Figure 2.7, a depth map obtained from stereo vision is represented, showcasing the ability of these cameras to obtain range data from targets.

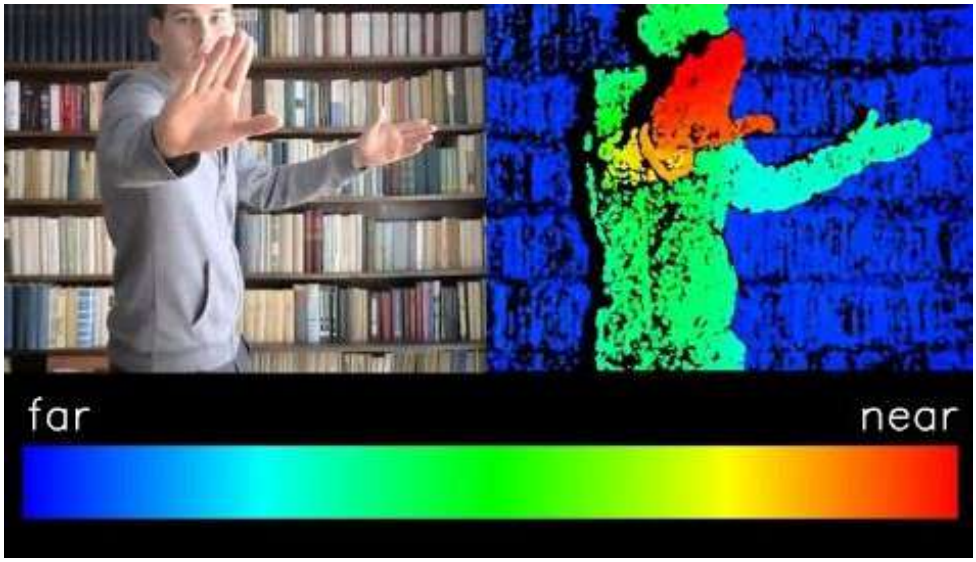


Figure 2.7: Depth map obtained from stereo vision [61]

The National Aeronautics and Space Administration (NASA) developed technologies for a new supersonic aircraft. One of the technological areas considered for the aircraft is the use of vision sensors and image-processing equipment to aid the pilot in detecting other aircraft in the sky. Several different algorithms were developed to detect obstacles in collision course and crossing trajectories: in case of a motionless background, the clutter is subtracted between consecutive frames. If the background moves,

an algorithm separates objects that have planar motion (background) from those with significant parallax motion (obstacles). In case of an object on collision course, morphological filtering is used because such objects can be nearly stationary in the image. Besides target translation, target expansion is also analyzed to get ranging information, as the increase of the image size of an object is inversely proportional to the time to collision [62].

2.3.5 Ultrasonic Sensors

Ultrasonic sensing is one of the direct ways to estimate the distance between the UAV and an obstacle. In principle, the sensor emits a sound, the sound wave hits the obstacle, it is reflected and the returning echo is recorded in the sensor. If the speed of the emitted sound in the air medium is known, then the distance to the object can be computed. The sonar operation replicates the echolocation performed by bats to locate their preys, as represented in Figure 2.8.

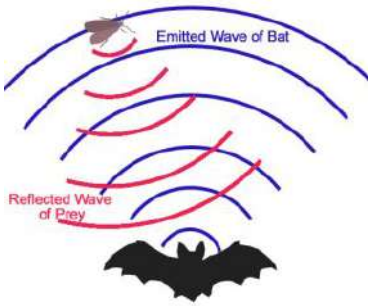


Figure 2.8: Bat echolocation [63]

Ultrasonic sensors are very effective under poor lighting conditions such as smoke or fog. However, these are proximal sensors, which means that their signal attenuates fast and the distance measurement ability is mostly below 10 m. Moreover, they cannot detect sound absorbing surfaces such as clothes. Therefore, these sensors are not reliable for detecting people. Because of this, it is fairly common to fuse these kind of sensors with infra-red sensors as their simultaneous use solves the drawbacks of each one [64, 65].

In reference [66], a sonar sensor model was designed to safeguard landings of UAVs, in which obstacles in the landing field are detected and, in case of obstacles higher than the UAV's landing legs, the procedure is aborted. Two different procedures were tested. Firstly, four ultrasonic sensors were attached to the UAV to identify the landing field plane and its tilt angle. Secondly, rotary ultrasonic sensors rotate along the vertical of the UAV, thus being able to map the landing plane and search for obstacles. The instability of the UAV due to the ground proximity affected the accuracy of the measurements but the results were acceptable.

In another work [67], the use of a hybrid acoustic and optical indoor positioning system for 3D positioning of UAVs was proposed. The acoustic sensing module was based on a Time-Code Division Multiple Access (T-CDMA) scheme, where the sequential emission of five spread spectrum ultrasonic codes was performed to compute the horizontal vehicle position following a 2D multilateration procedure. The received echoes were captured by a portable receiver and were then transmitted to an external com-

puter where the data fusion and iterative processing took place. Once again, the sensor fusion resulted in a 70-80% improvement of the overall accuracy. This approach differs significantly from the present work, since the processing will have to take place in an on-board computer.

2.4 Sensors' Comparative Analysis

With the information presented in this chapter, several inferences can be made. First and foremost, using cooperative detection is inadequate to the intended mission due to their high cost and weight, which are two of the critical parameters when designing an obstacle detection system to be used by mini UAVs. There are ADS-B systems adequate for UAV use, but since cooperative sensors are only able to detect other aircraft equipped with the same equipment, static obstacles like the ground, trees and unequipped aircraft remain undetectable.

Secondly, laser sensors and sonar sensors are fairly complementary. While laser sensors have a considerable range, sonars suffer from high signal attenuation and can only measure distances up to 10 m. Moreover, laser sensors do not perform well under low visibility conditions and ultrasonic sensors are not able to detect sound absorbing surfaces, but their simultaneous use can solve their singular drawbacks and thus, fusing these kind of sensors proves to be advantageous. These also have a significantly lower price compared to other sensors while being stable and easy to use. The main drawback of regular laser rangefinders is the narrow field of view (FOV), which only allows these sensors to detect what is directly in front of them. To solve this issue, a LIDAR can be used, which consists of a multidirectional laser range finder. However, these represent an increase in cost, weight and required power, since an electrical servo or oscillating mirror system also needs to be installed.

Most of the RADARs that can be used by UAVs comprise a patch or a scanning antenna, meaning that these sensors can be directional or multidirectional. This technology is also very power efficient while providing a large FOV and a range comparable to that of a laser rangefinder. This technology is very similar to that used by laser rangefinders and LIDARs, where the main difference resides in the frequency of the emitted radiation: RADARs emit 24 GHz microwave radiation while lasers use 300 THz infrared light.

Vision sensors also provide a good range, power efficiency, weight and cost, but are not capable of providing range information, so it is common practice to fuse vision with other sensors or to use stereo vision. These heavily rely on the texture and lighting of the objects and require a significantly more complex processing stage than other non-cooperative sensors, and also cooperative ones, since the range information is simply broadcast to these.

All these inferences are qualitatively represented in Table 2.1 and, in chapter 3, a more quantitative study will be made for each of the sensor types.

Table 2.1: Qualitative comparison of UAV sensor types

Sensor	Weight	Electric Power	Signal Processing	Cost	Range	Directionality	FOV
TCAS	high	high	very simple	high	very high	omnidirectional	-
ADS-B	high	low	very simple	low	very high	omnidirectional	-
Laser Rangefinder	low	low	simple	low	high	directional	very narrow
LIDAR	medium	medium	simple	medium	high	multidirectional	very narrow
RADAR	medium	low	simple	medium	high	(multi)directional	broad
Vision Sensors	low	low	complex	low	low	directional	variable
Ultrasonic Sensors	low	low	simple	low	low	directional	medium

2.5 Proposed Architecture

To detect and avoid obstacles in the UAV's path, various steps need to be taken in sequence but including several feedback links, so that the operation can be completed successfully. In the following chapters, each step will be analyzed and several options for each one will be given. The steps are the following:

1. Obstacle data acquisition from installed sensors: at the end of this phase, the controller will have access to the points that correspond to detected surfaces;

2. Data filtering and fusion: the obtained data can be filtered and fused with data obtained from other sensors in order to reduce present noise and obtain more accurate data;

3. Assignation of obtained data to a surface or shape: to apply any of the avoidance algorithms, a list of obstacles to avoid is needed and simply feeding a large number of detected points to this algorithm would not be feasible. So, in this stage, the sets of points are assigned to surfaces and shapes defined by certain parameters to be outputted to the following stages in form of list;

4. Collision detection: having the obstacles' list is not enough to start the avoidance maneuver because the list can be too extensive, as not every one of the detected obstacles will interact with the UAV's path. Consequently, the obstacles need processing in regard to their capability of interaction with the UAV. Also in this phase, the obstacles are sorted by risk, so that the more immediate threats are avoided first. The trimmed and ordered list that includes only the obstacles with high probability of interaction with the UAV will then be fed to the next and final stage;

5. Collision avoidance: finally, the UAV performs an avoidance maneuver, according to the proximity, velocity and type of obstacle.

As it can be seen in the flowchart representing this process in Figure 2.9, there are two feedback loops: a smaller one connecting the collision avoidance module to the collision detection module and a bigger one connecting the collision detection to the obstacle data acquisition module. As the avoidance maneuver is taking place, the UAV's state changes with it. This change affects the collision detection module, as obstacles that were not threats in the previous state can become threatening with the UAV's new position and orientation, which justifies the existence of the smaller control loop.

The first three phases occur sequentially, as they all consist of data acquisition and processing. However, a feedback loop needs to connect the collision detection module to the first one because the later modules can take much longer than the previous ones, due to the high operating frequency of the sensors, which can lead to new threats being detected while the last two modules are operating.

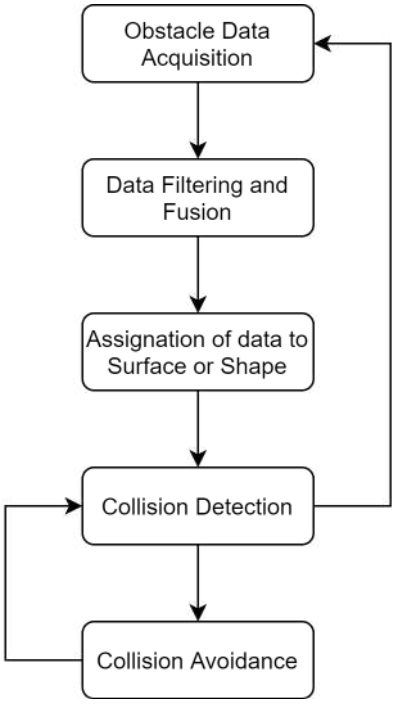


Figure 2.9: Flowchart representing chosen architecture

Chapter 3

Benchmark of Sensors

In this chapter, several models of previously described sensors will be compared in regard to their size, range, sensibility and other specifications so that the best suited sensors for our UAV can be chosen. Lastly, the sensed area of a sensor of each category will be graphically represented to get a better idea of what each of them can detect.

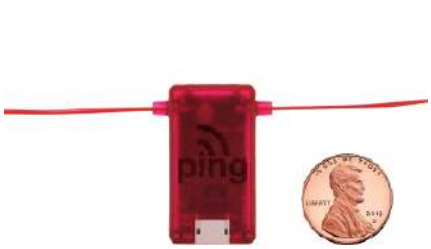
3.1 ADS-B

uAvionix has developed several dual-frequency ADS-B traffic sensors, namely pingRX, a receiver for UAV sense and avoid, which receives position reports from surrounding aircraft and displays them to the UAV pilot [68]. This sensor is extremely light, low-priced and communicates with the autopilot by MAVLink [69], a serial protocol most commonly used to send data and commands between vehicles and ground stations. MAVLink messages have a length of 263 bytes at most, can be sent over almost any serial connection and do not depend upon the underlying technology (WiFi, 900mhz radio, etc). The biggest drawback of this sensor is the fact that it does not include the ADS-B Out subsystem, hence cannot transmit information to other aircraft, being only able to receive data from others. uAvionix distributes other sensors able to send and receive aircraft data, but they are much more expensive and demand more voltage than pingRX. Ping1090i, for example, has Satellite Based Augmentation System (SBAS), GPS and a precision barometric sensor integrated to gather data to transmit to other ADS-B Out equipped systems [70].

The TR-1W produced by Aerobits is also an ADS-B Out transmitter/receiver with a more reasonable price than the Ping1090i, also compatible with any UAV using Mavlink protocol and also equipped with a barometer and Global Navigation Satellite System (GNSS) receiver [71]. The technical specifications of these sensors are displayed in Table 3.1 for easy consultation and the mentioned sensors are displayed in Figure 3.1.

Table 3.1: Comparison of different ADS-B sensors

	uAvionix pingRX	uAvionix Ping1090i	Aerobits TR-1W
ADS-B Out	no	yes	yes
Mass (g)	5	26	30
Input Voltage (V)	4-6	11-28	5
Sensitivity on 1090 MHz (dBm)	-84	-88	-72
Cost (€)	230	1835	725



(a) uAvionix pingRX next to a coin for reference [68]

(b) uAvionix ping1090i [70]

(c) Aerobits TR-1W [71]

Figure 3.1: ADS-B sensors

3.2 Laser Rangefinders/LIDAR

Lightware is one of the leading companies in the development and distribution of laser range-finder technology for application across numerous industries. They offer a wide variety of products, namely the LW20/C model, which has a 100 meter range in sunlit conditions, a 1 cm resolution, an accuracy of 10 cm, weights 20 grams and is confined within a water-proof enclosure [72].

The LW20/C runs from a single 5 Volt power source and has two operations modes: it can be used as a distance measuring sensor which allows this device to be used like an altimeter, or be attached to a digital servo to create LIDAR maps that sense the world as two dimensional images, thus becoming a low cost, lightweight collision avoidance sensor. In scanning mode, 48 to 388 measurements per second can be made and the data can be internally processed or streamed either by Serial or I2C ports in the form of 32 bit ASCII strings, while the servo is controlled by a pulse-width modulated (PWM) signal coming from a control line of the LW20/C. Additionally, two alarms can be set so that warnings are issued once obstacles are within a certain distance.

Other models produced by this manufacturer include the SF30/C [73] and the SF30/D [74], which can also operate as rangefinders or LIDARs and have the same resolution, accuracy and power supply parameters as those of the LW20/C. The sensors are displayed in Figure 3.2 and their differences are displayed in Table 3.2. It is important to note that despite being the only sensor with a protective casing, the LW20/C is still the lightest model.

Table 3.2: Comparison of different laser rangefinders

	Lightware LW20/C	Lightware SF30/C	Lightware SF30/D
Protective casing	yes	no	no
Mass (g)	20	35	35
Range (m)	100	50	200
Maximum measurement frequency (Hz)	388	20010	20000
Cost (€)	255	275	365

(a) Lightware LW20/C [72]

(b) Lightware SF30/C [73]

(c) Lightware SF30/D [74]

Figure 3.2: Laser/LIDAR sensors

3.3 RADAR Sensors

Aerotenna is a company powered by Ainstein specialized in RADAR sensing and processing solutions for UAVs. They developed, among others, the μSharp Patch Sense and Avoid RADAR, which is very lightweight and requires a very low power consumption [75]. Regarding its performance, the μSharp Patch provides high resolution distance sensing in the range of 0.5 m to 120 m. It is able to detect both moving and stationary objects and delivers a remarkable performance in all weather and light conditions. It has a horizontal FOV of 50° and a vertical FOV of 30° , which is fairly adequate for fixed-wing UAVs.

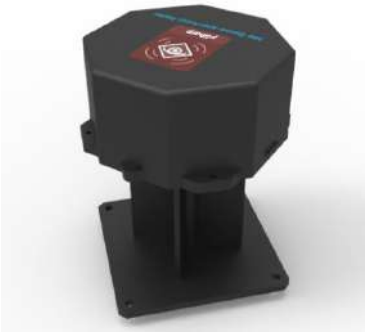
Another model provided by Aerotenna is the μSharp - 360° Sense-and-Avoid RADAR, which scans 360° without blind spots, locating targets on the horizon for reliable and quick reactions to change flight course [76]. It is also very compact and power efficient, despite having a smaller range (40 m) and weighing considerably more due to the mechanical equipment needed for scanning.

Aerotenna's sister company, Ainstein, produces the SRD-D1, an object detection RADAR sensor suitable for small UAVs [77]. In addition to range and relative position, this sensor is also able to compute the velocity of the detected targets. It does not experience degradation in performance under low-light and adverse meteorological conditions. Regarding other performance characteristics, the SRD-D1 is weaker than the previous two sensors: it has a worse accuracy, a lower update rate and weighs almost as much as the scanning RADAR. Conversely, it has a wider horizontal FOV than the μSharp Patch. All these sensors only require 5V of input voltage and have a UART connection interface. These sensors are displayed in Figure 3.3 and their specifications are represented in Table 3.3.

Table 3.3: Comparison of different RADAR sensors

	Aerotenna μSharp Patch	Aerotenna μSharp - 360°	Ainstein SRD-D1
Range (m)	120	40	30
Mass (g)	43	243	200
Horizontal FOV (°)	50	360	60
Accuracy (cm)	22	22	60
Update rate (Hz)	90	80	10
Cost (€)	630	7665	530

(a) Aerotenna μSharp Patch [75]



(b) Aerotenna μSharp - 360° [76]

(c) Ainstein SRD-D1 [77]

Figure 3.3: RADAR sensors

3.4 Vision Sensors

The chosen vision sensors for this study include two monocular sensors unable to compute range data directly and one stereo sensor. All these sensors employ the complementary metal-oxide semiconductor (CMOS) technology in which the charge is converted to voltage at the pixel, as opposed to charge-coupled device (CCD) in which each pixel's charge packet is transferred to a common output structure which converts the charge to voltage, once the exposure is complete. CMOS sensors offer superior integration, power dissipation and system size at the expense of image quality (particularly in low light) and flexibility, but the cost of these technologies are approximately equal [78]. Despite offering superior image quality and flexibility, CCD cameras' development has slowed down and this technology is near its end of life, which entailed the choice of comparing only CMOS sensors.

There is also a distinction to be made regarding the type of shutter used. With rolling shutters, the image is captured by scanning the area rapidly. These are cheaper and easier to implement, which results in a good SNR, but can lead to distorted pictures if the target or the camera itself moves too quickly, as seen in the example presented in Figure 3.4. On the other hand, cameras with global shutters capture the whole scene at the same time to prevent this effect, but require one to three more transistors per pixel, which can induce noise [79].

Lumenera is one of the global market leaders regarding digital cameras, providing an extensive range of high quality vision sensors with different resolution, speed and sensitivity. Lt-C1950 and Lt-C1900 are two of the sensors offered by this company. Lt-C1900 is a CMOS sensor with a rolling shutter and a

Figure 3.4: Distortion caused by a rolling shutter in the rotating blades of an airplane [80]

60 fps frame rate, providing great color reproduction, dynamic range, low noise and a high sensitivity which makes imaging easier in environments with variable or uncontrolled lighting. The Lt-C1900 and the Lt-C1950 models look and weigh the same but the Lt-C1950 possesses a slightly higher resolution and a much higher frame rate (162 fps). Moreover, this model has a global shutter, which despite all the advantages cited above means an increased cost of 470 € (The cost of the Lt-C1900 model was not available but it can be assumed lower than 470 €) [81].

Intel has produced several stereo cameras that can be used for UAV navigation such as the Depth Camera D435 that is equipped with two imagers, whose data is sent to the Vision Processor, which calculates depth values for each pixel in the image by correlating points on the left image to the right image, and via shift between a point on the left image and the right image. It has a 10 m range, a 90° x 65° FOV (H x V) and is also equipped with a RGB camera and an infra-red projector that projects non-visible static IR pattern to improve depth accuracy in scenes with low texture. Another great attribute of this camera is the fact that a vision processor is also integrated in the small form module, making it easier to implement and reducing the load the UAV will have to transport. By having global shutter sensors, this camera works adequately in low-light environments [82].

The different sensors are displayed in Figure 3.5 and their different specifications are presented in Table 3.4. The Intel D435 is a more suitable model for our application, as it contains a global shutter, can provide depth information at close range and has an IR projector while weighing less than the other models and having a significantly lower cost.

Table 3.4: Comparison of different cameras

	Lumenera Lt-C1900	Lumenera Lt-C1950	Intel D435
Type of camera	monocular	monocular	stereo
Type of shutter	rolling	global	global
Mass (g)	88	88	72
RGB Resolution (MP)	2.1	2.3	2.1
Cost (€)	N/A	470	160

(a) Lumenera Lt-C1900/Lt-C1950 [81]

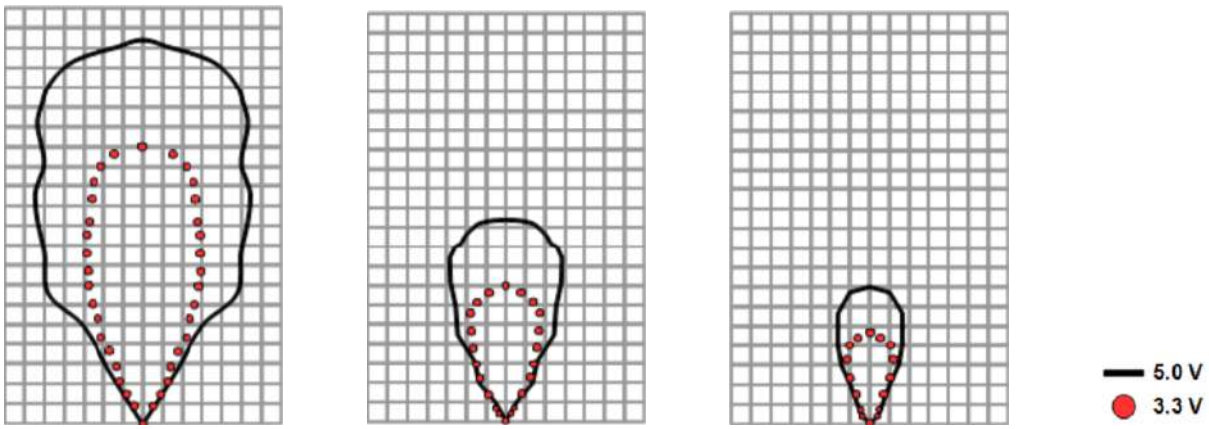
(b) Intel D435 [82]

Figure 3.5: Vision sensors

3.5 Ultrasonic Sensors

MaxBotix designs and manufactures ultrasonic sensors, namely the ones belonging to the I2CXL-MaxSonar-EZ series. These sensors have a high acoustic power output along with real-time auto calibration for changing conditions. They require low power operation (3V to 5.5V), have a centimeter resolution, can detect obstacles from 20 cm to 765 cm, have a 40 Hz read rate and have a small form factor. The communication is made using the I2C protocol and the sensor can only receive three different commands: take range reading, report last range value and change sensor address. Because of this, the controller needs to periodically send these first two commands in order to obtain a stream of data [83].

The models of ultrasonic sensors belonging to this series include the MB1202, the MB1222 and the MB1242. All these different models operate the same way and differ only on their range, cost and beam pattern, which are represented in Figure 3.6 and Table 3.5, while a typical sensor belonging to this series is represented in Figure 3.7.



(a) MaxBotix MB1202

(b) MaxBotix MB1222

(c) MaxBotix MB1242

Figure 3.6: Different models' beam patterns for a 8.89 cm diameter dowel [83]

The beam patterns of Figure 3.6 show the capability of the different models to detect a small dowel and so, represent their sensitivity and beam wideness but not their range. From this figure and the sensors' datasheets [83], we can infer that the MB1202 has the highest sensitivity and widest beam

Table 3.5: Comparison of the sonar models' range and cost

	MaxBotix MB1202	MaxBotix MB1222	MaxBotix MB1242
Range (cm)	770	810	655
Cost (€)	40	35	35

sensor, making it the best suited sensor of this series for people detection, despite having the lowest noise tolerance. The MB1222 model offers a good balance between wide and narrow beam sensors, and large and narrow object detection, which makes it suitable for nearly all conceivable applications. The MB1242 has the highest noise tolerance and the narrowest beam of the whole series, being able to provide stable range readings to large targets even in electrically and acoustically noisy environments.

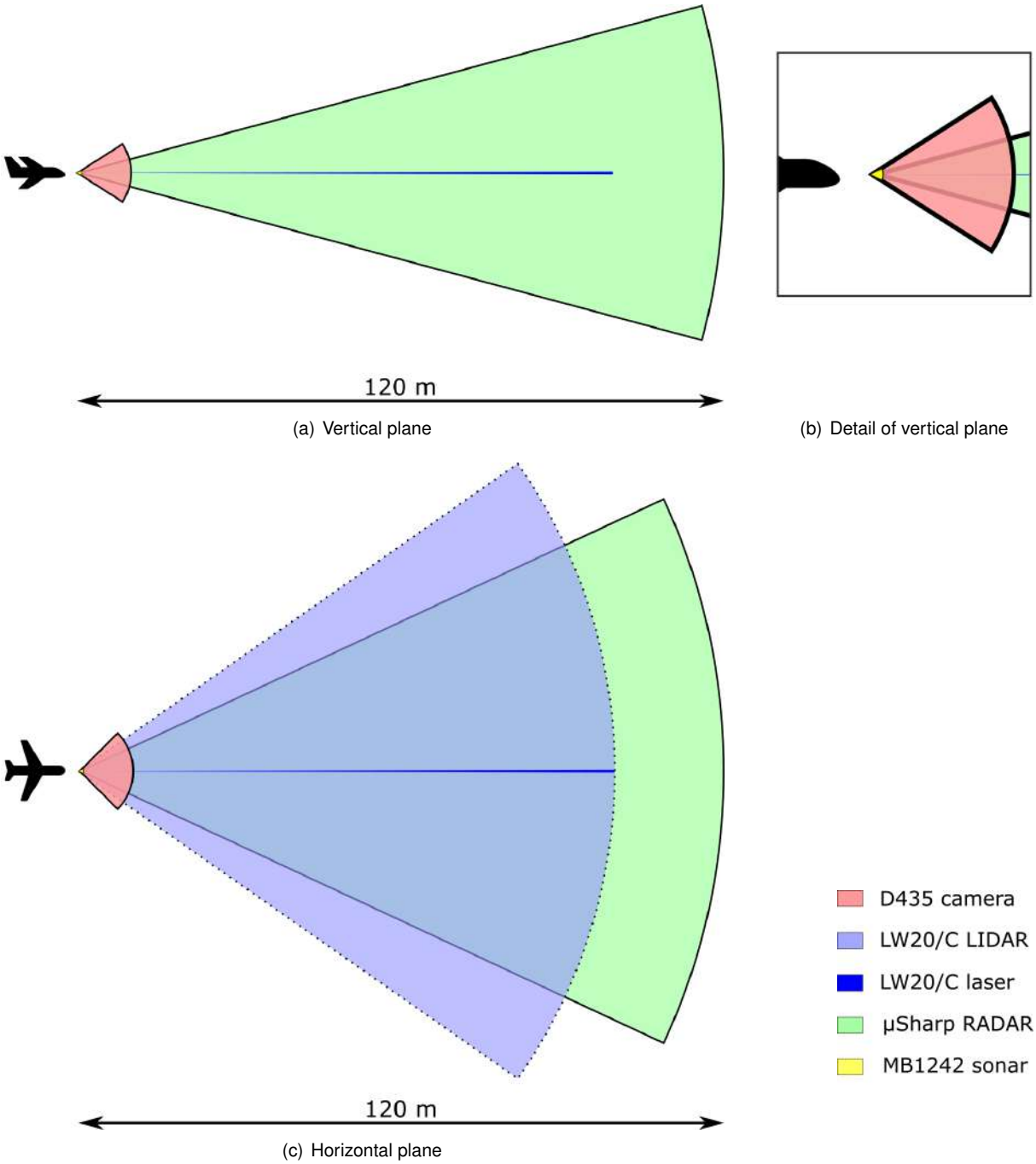
Figure 3.7: MaxBotix I2CXL-MaxSonar-EZ sensor [83]

3.6 Chosen Sensors Comparison

In this section, one of each sensor type was picked to be compared to the others, regarding their range and FOV (horizontal and vertical), so that their attributes and flaws could be better showcased. The picked sensors for this comparison were the uAvionix pingRX ADS-B sensor, the Lightware LW20/C laser rangefinder, the Aerotenna μSharp Patch RADAR, the Intel D435 stereo camera and the MaxBotix MB1242 sonar. All these sensors' sensed areas are represented in Figure 3.8, except for the ADS-B sensor, due to its omnidirectionality and the fact that its range depends on the power of the other aircraft's emitting signal. The sonar's FOV represented in the figure was based on the beam width present in Figure 3.6 (c), which is approximately 50° . This sensor's FOV is not very visible as a result of its small range (two orders of magnitude below the RADAR's range), so a detail was included in Figure 3.8 (b).

It is important to note that the LIDAR sensor, due to its multidirectionality, can scan with different angular apertures, and its area is only limited by the used scanning servo. In Figure 3.8, an arbitrary 70° horizontal FOV was chosen and, because of this, the limits of the sensed area were dotted. The LW20/C's area, when the scanning mode is not activated, is represented in dark blue, where its 0.3° beam divergence can be observed. This value is also the vertical FOV for this sensor whether its scanning mode is activated or not, as the servo only scans the horizontal plane.

The camera's sensed area is associated with its depth image sensors. The infra-red and color



(a) Vertical plane

(b) Detail of vertical plane

(c) Horizontal plane

Figure 3.8: Comparison of several sensors' ranges and FOVs (UAVs not to scale)

cameras have their own range and FOV but these are more applicable to complement other sensors as they do not provide depth data.

Based on this comparison, the ultrasound and stereo vision sensors were not modeled in chapter 4 due to their small range, as identifying obstacles when they are only at a distance of 10 m or 0.765 m, when the UAV is traveling at a maximum cruise speed of 15 m/s, generally, does not result in successful avoidance maneuvers.

Chapter 4

Sensor and Obstacle Models

Before analyzing the models for the different sensors, a study of Kalman filters (KF) must be made, due to its widespread use and effectiveness in tracking the status of dynamic systems with random disturbances. After introducing the different Kalman filters, the different sensor models are analyzed and a possible way to model the obstacles in the environment is also made.

4.1 Kalman Filters

The Kalman filter is a real-time estimator that employs a predictor-corrector method because it propagates the linear least mean squares estimate $\hat{\mathbf{x}}$ and its covariance of estimation uncertainty $\mathbf{P}$ forward in the time between measurements, predicting the estimate of the state variables along its covariance of estimation uncertainty before the next measurement is used. Then, the results of the measurements are used to correct the predicted values to reflect the influence of the information gained from the new measurements. If no new measurements are obtained, then the update is not performed [84].

In Figure 4.1, an example of the use of a Kalman filter is illustrated, in which a green sphere passes through a cardboard box that hides the sphere from the camera. In both parts of the image, all the video frames that include the sphere are overlaid. However, the right part of the image also includes the estimation of the sphere's position obtained from a Kalman filter applied to a constant acceleration model.

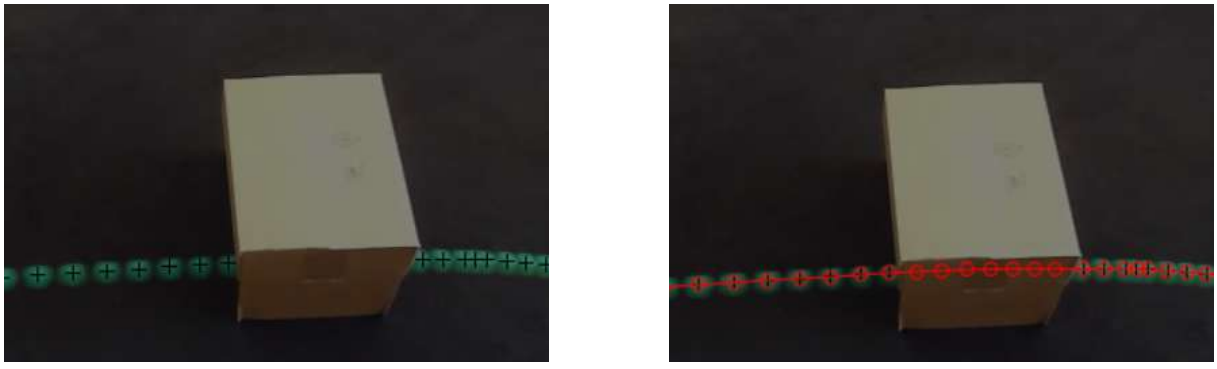


Figure 4.1: Tracking of a green sphere with a Kalman filter, when occluded [85]

All of the considered sensors for obstacle detection can only provide the relative position between the target and the UAV, therefore the vehicle needs to be equipped with a set of sensors that would allow it to get the absolute state of the obstacles. The Kalman filters can be used to track them.

The essential implementation equations for the update part of the Kalman filter are given by [84]

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{R}_k + \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T)^{-1} \tag{4.1}$$

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-) \tag{4.2}$$

$$\mathbf{P}_k^+ = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_k^- \,, \tag{4.3}$$

while the equations associated with the prediction are

$$\hat{\mathbf{x}}_k^- = \mathbf{\Phi}_k \hat{\mathbf{x}}_{k-1}^+ + \mathbf{B}_k \mathbf{u}_k \tag{4.4}$$

$$\mathbf{P}_k^- = \mathbf{\Phi}_k \mathbf{P}_{k-1}^+ \mathbf{\Phi}_k^T + \mathbf{Q}_k \,. \tag{4.5}$$

The parameters and variables used in these equations are described in Table 4.1. To apply this algorithm, the matrices need to be adequately defined and initial estimations of the state vector and the covariance matrix need to be made.

Table 4.1: Parameters and variables used in the Kalman filter equations

Symbol	Description	Dimension
n	Dimension of state vector	1
ℓ_k	Dimension of *k*th measurement vector	1
j_k	Dimension of *k*th input vector	1
k	Discrete measurement index	1
t_k	Discrete time of *k*th measurement	1
$\hat{\mathbf{x}}_{k-1}^+$	*A posteriori* state estimate at time t_{k-1}	$n \times 1$
$\mathbf{P}_{k-1}^+$	*A posteriori* state estimation covariance at time t_{k-1}	$n \times n$
$\mathbf{\Phi}_k$	State transition matrix from t_{k-1} to t_k	$n \times n$
$\mathbf{B}_k$	Control-input matrix at time t_k	$n \times j_k$
$\mathbf{Q}_k$	Uncertainty accumulated between t_{k-1} and t_k	$n \times n$
$\mathbf{u}_k$	Input vector at time t_k	$j_k \times 1$
$\hat{\mathbf{x}}_k^-$	*A priori* state estimate at time t_k	$n \times 1$
$\mathbf{P}_k^-$	*A priori* estimation covariance at time t_k	$n \times n$
$\mathbf{z}_k$	Measurement vector at time t_k	$\ell_k \times 1$
$\mathbf{H}_k$	Measurement sensitivity matrix at time t_k	$\ell_k \times n$
$\mathbf{R}_k$	Measurement error covariance at time t_k	$\ell_k \times \ell_k$
$\mathbf{K}_k$	Kalman gain matrix at time t_k	$n \times \ell_k$
$\hat{\mathbf{x}}_k^+$	*A posteriori* state estimate at time t_k	$n \times 1$
$\mathbf{P}_k^+$	*A posteriori* estimation covariance at time t_k	$n \times n$

The classic Kalman filter is a very effective linear estimator. However, since many of the estima-

tion problems of practical interest are nonlinear, several approximations were developed to apply the more successful linear estimation methods to less-than-linear problems, namely the first-order Extended Kalman Filter (EKF) and the Unscented Kalman Filter (UKF) [84]. The EKF, specifically, uses first-order partial derivatives evaluated at the estimated value of the state vector. Thus, if the state vector is nonlinear, this vector and the state transition matrix are defined as

$$\mathbf{x}_k = f_k(\mathbf{x}_{k-1}, \mathbf{u}_k) + \boldsymbol{\omega}_k \tag{4.6}$$

$$\boldsymbol{\Phi}_k = \left.\frac{\partial f_k}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_{k-1}^{+}}, \tag{4.7}$$

where $\boldsymbol{\omega}_k$ is the process noise, assumed to be a zero-mean multivariate Gaussian noise with covariance $\mathbf{Q}_x$. Due to the loss of linearity, equation (4.4) is no longer valid in the prediction phase of the filter. Because of this, it is replaced by the previous computation of the state transition matrix, equation (4.7), and by

$$\hat{\mathbf{x}}_k^{-} = f_k(\hat{\mathbf{x}}_{k-1}^{+}, \mathbf{u}_k) \tag{4.8}$$

On the other hand, if the measurement vector is nonlinear, this vector and the measurement sensitivity matrix are defined as

$$\mathbf{z}_k = h_k(\mathbf{x}_k^{-}) + \boldsymbol{\nu}_k \tag{4.9}$$

$$\mathbf{H}_k = \left.\frac{\partial h_k}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_k^{-}}, \tag{4.10}$$

respectively, where $\boldsymbol{\nu}_k$ is the observation noise, assumed to be a zero-mean multivariate Gaussian noise with covariance $\mathbf{R}_x$. Because of the nonlinearity of the measurement vector, equation (4.2) of the update phase is no longer valid and it is replaced by the previous computation of the sensitivity matrix, equation (4.10), and by

$$\hat{\mathbf{x}}_k^{+} = \hat{\mathbf{x}}_k^{-} + \mathbf{K}_k(\mathbf{z}_k - h_k(\mathbf{x}_k^{-}))\,. \tag{4.11}$$

Obviously, if the state and measurement vectors are simultaneously nonlinear, both changes given by equations (4.7 - 4.10) need to be made.

4.2 Sensor Models

A sensor model is an abstraction of the actual sensing process that describes the information a sensor can provide, how this information is limited by the environment and how it can be enhanced by data obtained from other sensors. According to [86], having an effective model of the sensor capabilities is the key to intelligent fusion of disparate sensors, as having the sensors' capabilities well estimated *a*

priori leads to the development of sensor strategies more in line with the application's requirements.

For the developed simulation, different sensors need to be modeled in order to compare their behavior and find the combination of sensors that achieve the best results. The sensors are characterized by their range, FOV, accuracy and data frequency. The values used for these parameters are the ones presented in section 3.6, which were obtained from their technical manuals or inferred from available data. The considered parameters are presented in Table 4.2.

Table 4.2: Characteristics of the different sensors used in simulations

	LIDAR	Laser rangefinder	RADAR
Range (m)	100	100	120
Horizontal FOV (°)	variable	0.3	50
Accuracy (m)	0.2	0.2	0.22
Maximum Frequency (Hz)	388	388	90

4.2.1 Laser Rangefinder/LIDAR model

Fayad and Cherfaoui [46] presented an approach to solve the problem of tracking partially hidden objects by a single layer laser scanner to be used in driving situations. The proposed method to cluster and estimate results from the point cloud provided by the sensor can be adapted to perform obstacle modeling so that the obstacle parameters can be fed to the collision detection module.

A scanning laser rangefinder can only detect objects, either dynamic or static, that have sufficient reflectivity and that are inside its visible area. That is to say, the objects are within range, inside the FOV and are not obstructed by other objects. All these different zones present in the scanning plane are displayed in Figure 4.2.

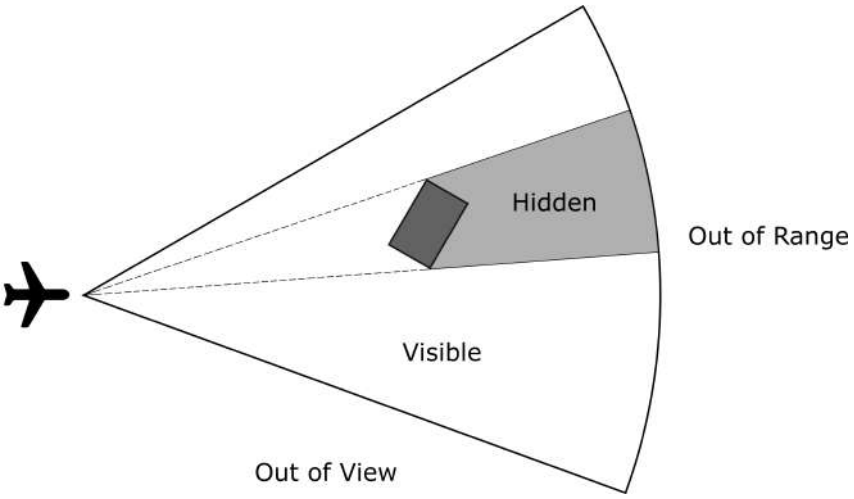


Figure 4.2: Different areas of the scanning plane of a LIDAR

The used approach consists of representing a detected obstacle with the smallest parallelepiped that encloses all the detected points, where the generated parallelepipeds are aligned horizontally, the scanning direction, as a result of the minute vertical field of view that lasers possess. So, the obstacle's

length and width could be detected while a prediction could be made to estimate its height according to its classification considering its position and detected dimensions.

One problem associated with obstacle detection using laser rangefinders is the fact that only one or two sides of the obstacle can be detected, but the detected points are not uniformly distributed along the detected surfaces. Because of this, using the center of gravity of the detected points can introduce large errors and discontinuities. A better approximation consists of identifying the geometric center of the obtained points and associating it to the gravity center.

To compute the object's dimensions and the position of its geometric center, the obstacles need to be reconstructed. If an object is totally visible, it is considered that its half was detected and the remaining of the obstacle is reconstructed assuming symmetry, where the center of symmetry is the medium point of the segment connecting the first and last point of the cluster. The described process is represented in Figure 4.3. In subsequent scannings, this approximation is updated according to the object's position and new data. If the object is completely visible but only one side of the obstacle is detected instead of two, it is not possible to estimate the object's length, so the reconstruction only takes place in following detections, after more data is gathered.

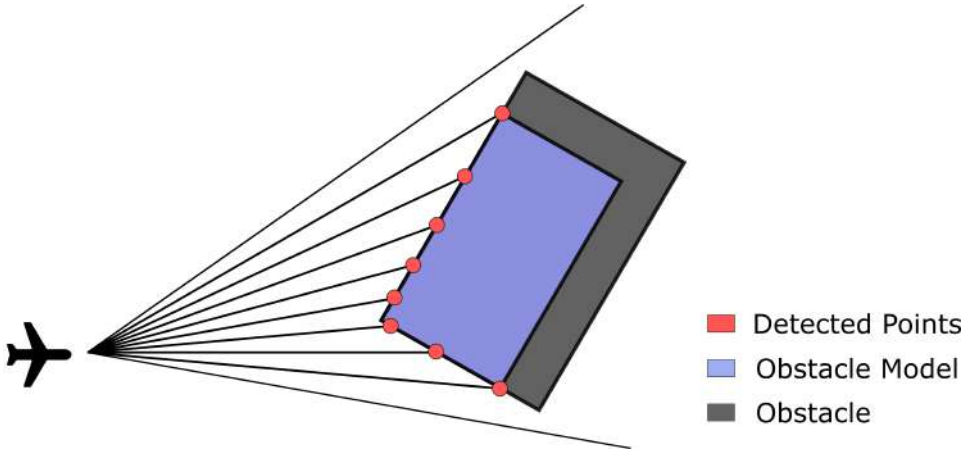


Figure 4.3: Obstacle reconstruction using a LIDAR

If the obstacle is partially outside the visible zone, these estimations take into consideration the data history collected of the obstacle if the object was detected in a previous cycle in the totally visible zone. If the present cycle corresponds to the first time the object was detected, the measured parameters are saved in the object's history to be used in following cycles, where the object is better positioned. The recognition of any hidden part of the object is based on the position of the object within the visible zone and in regard to other obstacles present in the scene.

Fayad and Cherfaoui [46] also provide a solution to the errors caused by the higher distance between consecutive points in farther obstacles which results in smaller detected dimensions, as seen in Figure 4.3, where the modeled obstacle is considerably smaller than the real obstacle. Instead of saving the maximum detected dimensions and updating these only if larger ones are detected, the dimensions are passed by a time filter. This was done because the maximum saving method fails easily if noisy measurements occur, as a large and noisy measurement can be saved and never replaced by real, future and smaller measurements.

So, in each measurement the detected length L and/or the detected width W are filtered by

$$W_k = W_{k-1} + G(W_m - W_{k-1}) \tag{4.12}$$

$$L_k = L_{k-1} + G(L_m - L_{k-1})\,, \tag{4.13}$$

where G (0<G<1) is the filter gain, W_k and L_k are the filtered dimensions at instant t_k, W_{k-1} and L_{k-1} are the filtered dimensions at instant t_{k-1}, and W_m and L_m are the measured dimensions at instant t_k.

The gain needs to be carefully selected as it impacts the speed of the variation of the dimensions. A small gain corresponds to a slow variation and it is preferable for noisy environments but not suitable for high relative speed objects. The gain can be determined by

$$G = 1 - \sqrt[n]{1 - acc}\,, \tag{4.14}$$

where acc corresponds to a fraction that represents the desired accuracy of the dimensions and n corresponds to the number of filter cycles required to get an accuracy of acc.

Regarding the tracking phase, classical Kalman filters were used, where the motion of detected obstacles is considered to be linear and constant between consecutive scans. This simplification describes the state of the targets with an acceptable error, considering a high scanning frequency. The coordinates of the obstacles need to be known for the filter to be applied.

Using this model, the state vector and the state transition matrix can be defined, respectively, as

$$\mathbf{x}_k = \begin{bmatrix} x_k \\ y_k \\ v_{x_k} \\ v_{y_k} \end{bmatrix} \text{ and } \mathbf{\Phi}_k = \begin{bmatrix} 1 & 0 & \Delta t_k & 0 \\ 0 & 1 & 0 & \Delta t_k \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.15}$$

where Δt_k is the sampling time between two scans at iteration k, (x_k, y_k) is the position of the detected obstacle at instant t_k according to a fixed frame of reference and (v_{x_k}, v_{y_k}) is the obstacle's velocity at instant t_k. This model assumes a LIDAR that only scans horizontally, but if the rangefinder was to be attached to a gimbal with two degrees of freedom, it would have to be extended to include the third dimension.

The laser range finder is only able to measure the position of the obstacle. Therefore, the measurement vector and measurement sensitivity matrix are defined, respectively, as

$$\mathbf{z}_k = \begin{bmatrix} x_k \\ y_k \end{bmatrix} \text{ and } \mathbf{H}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \tag{4.16}$$

The measurement error covariance matrix $\mathbf{R}_k$ should be calculated from the laser scanner characteristics, while the uncertainty between two measurements $\mathbf{Q}_k$ should take into consideration all errors imported by the approximation of the real motion by the equation of the model.

4.2.2 Camera Model

One of the simplest models that describes the operation of a camera is the pinhole camera model. This model describes a relationship between the coordinates of a point in the three-dimensional world and its projection onto the image plane of an ideal pinhole camera, where no lenses are used to focus light and the aperture of said camera is considered punctual. This model is very simple and does not take into account several effects caused by the presence of lenses and finite-sized apertures, such as geometric distortions and the blurring of unfocused objects. In spite of this, it is a reasonable model since these effects are small enough to be neglected if a camera with a high enough quality is used [87, 88].

In Figure 4.4, the geometry of the pinhole model is represented, where the obstacle's position X_{obs} is unknown to the vehicle, unlike its position projected into the image plane Z_{obs}. The image plane is parallel to the axes Z_c and Y_c, intercepts the principal axis X_c at the principal point P and is located at a distance f (focal length) from the origin.

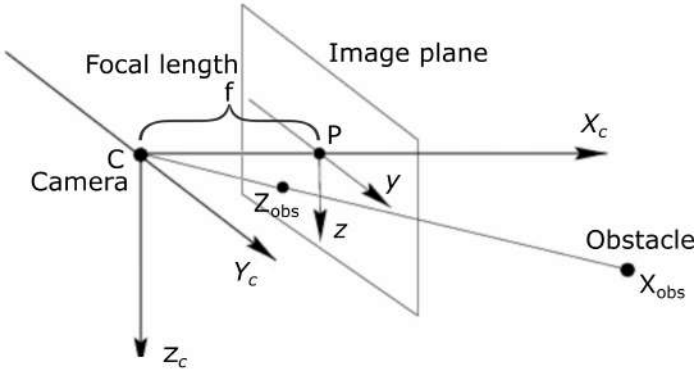


Figure 4.4: Pinhole camera model (adapted from [88])

Considering the vector $\mathbf{x}_c = \mathbf{L}_{CL}\mathbf{x} = [X_c, Y_c, Z_c]^T$ containing the coordinates of the point X_{obs} in a frame aligned with the camera's attitude (rotated by $\mathbf{L}_{CL}$ from the local frame to the camera frame), and the vector $\mathbf{z} = [z_y, z_z]^T$ containing the coordinates of Z_{obs} in the image plane, by recognizing the similar triangles in the aforementioned figure, the 2-D measurement of the obstacle position in the image plane at a k-th time step can be given by

$$\mathbf{z}_k = \frac{f}{X_{c_k}} \begin{bmatrix} Y_{c_k} \\ Z_{c_k} \end{bmatrix} + \boldsymbol{\nu}_k = h_k(\mathbf{L}_{CL_k}\mathbf{x}_k) + \boldsymbol{\nu}_k \tag{4.17}$$

where f is the focal length of the camera and ν_k is a zero mean Gaussian discrete white noise process with covariance matrix $\mathbf{R}_k = \sigma^2\mathbf{I}$ [89].

The measurement model is nonlinear with respect to the relative state. Thus, an EKF with a linear state vector is applied to estimate the relative position of the obstacles. By using equation (4.10), the measurement sensitivity matrix can be computed as

$$\mathbf{H}_k = \frac{1}{\hat{X}_{c_k}^-} \begin{bmatrix} -h(\hat{\mathbf{X}}_{c_k}^-) & \mathbf{I}_{2\times 2} \end{bmatrix} \mathbf{L}_{CL_k} \tag{4.18}$$

and equation (4.4) is rewritten as

$$\hat{\mathbf{x}}_k^- = \mathbf{\Phi}_k \hat{\mathbf{x}}_{k-1}^- - \mathbf{v}_{k-1}\Delta t_{k-1} - \frac{1}{2}\mathbf{a}_{k-1}\Delta t_k^2 \,, \tag{4.19}$$

where Δt_k is, once again, the sampling time and and $\mathbf{v}_{k-1}$ and $\mathbf{a}_{k-1}$ are the velocity and acceleration vectors of the vehicle, respectively, in the local fixed frame at instant t_{k-1}. The omission of the term **Bu** is due to the lack of known control inputs. Instead, it is replaced by the new terms, which represent the effects caused by unknown inputs. Furthermore, the state transition matrix is approximated to the identity matrix ($\mathbf{\Phi}_k = \mathbf{I}$), which is valid for stationary obstacles and a sufficiently small sampling time. Lastly, the covariance matrix of the process noise is defined as

$$\mathbf{Q}_k = \sigma_X^2 \mathbf{I}_{2\times 2}\Delta t_k \,. \tag{4.20}$$

The biggest disadvantage of this model is its inability to deal with moving obstacles. This can be achieved by applying an adaptive estimator for the relative state of moving obstacles. However, regardless of the approach, obtaining the distance between the UAV and any other moving vehicle by relying only on vision-based data is a difficult task, as the accuracy of the range estimations depends on the camera translation motion: it is at its highest when the UAV is moving in parallel to the image plane and is at its lowest when its motion is perpendicular to this plane.

Another associated problem is the influence of the target's acceleration on the estimations provided by the EKF. These unknown accelerations act as unmodeled disturbances on the estimation process, which results in biased or even diverging estimates [89].

4.2.3 RADAR Model

To evaluate the system performance, the RADAR sensor was modeled in the context of the Sense and Avoid system. So, this model addresses the angular accuracy, update rate, range and field of view of the RADAR, rather than being a lower-level model that would deal with signal and environment modeling.

Assuming the RADAR sensor outputs the range, bearing and elevation of the detected obstacles, the state estimation becomes more complicated than the estimations used in previous models, as these outputs are polar, whereas the intruder dynamics are best expressed in rectangular coordinates. Tracking in Cartesian coordinates using polar measurements can be handled in two ways. The first method consists in converting the polar measurements to a Cartesian frame of reference so that a linear Kalman filter can be used, whereas the second approach is to use an extended Kalman filter, which incorporates the original measurements in a nonlinear fashion into the target state estimation, resulting in a mixed coordinate filter [90].

In both techniques, the inaccuracy of the measurements when converted to a Cartesian frame of reference must be accounted for properly. In the first method, the measurement error covariance matrix **H** must be computed in every iteration of the filter, while in the second approach, the initial state covariance

depends on the accuracy of the initial converted measurements and the gains depend on the accuracy of the subsequent linearization.

The chosen radar model was the converted measurement Kalman filter (CMKF), due to its easier implementation. Once again, for simplicity's sake, all the following equations reflect a two-dimensional model that can easily be extended to 3-D.

Standard Conversion

Using the standard conversion, described in [91], the Cartesian coordinate measurements x_m and y_m to be processed by the KF are obtained from

$$x_m = r_m \cos(\alpha_m) \tag{4.21}$$

$$y_m = r_m \sin(\alpha_m)\,, \tag{4.22}$$

where α_m is the measured azimuth and r_m is the measured range in polar coordinates, obtained from

$$r_m = r + \tilde{r} \tag{4.23}$$

$$\alpha_m = \alpha + \tilde{\alpha}\,, \tag{4.24}$$

where r and α are the true values of range and azimuth, respectively, and $\tilde{r}$ and $\tilde{\alpha}$ are the measurement errors assumed to be independent Gaussian white noises with zero mean and standard deviation σ_r and σ_a respectively. Using Taylor first-order expansion, the measurement errors in Cartesian coordinates are

$$\tilde{x} = x_m - x \approx \tilde{r}\cos(\alpha) - \tilde{\alpha} r \sin(\alpha) \tag{4.25}$$

$$\tilde{y} = y_m - y \approx \tilde{r}\sin(\alpha) + \tilde{\alpha} r \cos(\alpha)\,, \tag{4.26}$$

where $x = r\cos(\alpha)$ and $y = r\sin(\alpha)$. From equations (4.25) and (4.26), the Cartesian covariance matrix for the standard conversion can be built, but since the true values (r, α) are practically unavailable, these are replaced by the measured values in the expression. So, the resulting covariance matrix for the standard conversion is given by

$$\mathbf{R}_s = (x_m, y_m | r_m \alpha_m) = \begin{bmatrix} R_s^{11} & R_s^{12} \\ R_s^{21} & R_s^{22} \end{bmatrix}\,, \tag{4.27}$$

where

$$R_s^{11} = \sigma_r^2 \cos^2(\alpha_m) + r_m^2 \sigma_{\alpha_m}^2 \sin^2(\alpha_m) \quad (4.28)$$

$$R_s^{22} = \sigma_r^2 \sin^2(\alpha_m) + r_m^2 \sigma_{\alpha_m}^2 \cos^2(\alpha_m) \quad (4.29)$$

$$R_s^{12} = R_s^{21} = 0.5(\sigma_r^2 - r_m^2 \sigma_{\alpha_m}^2) \sin(2\alpha_m)\,. \quad (4.30)$$

Unbiased Conversion

The standard conversion is the simplest way to implement the CMKF. However, it gives biased inconsistent estimates for certain levels of cross-range measurement error owing to the nonlinear transformation of the noisy bearing. Because of this, several studies were made to compensate the present bias within the Kalman filter framework. Here, the unbiasing method developed in [92] is described.

Taking expectation in equations (4.22) and (4.22), we have

$$E(x_m) = \lambda_\alpha r \cos(\alpha) \quad (4.31)$$

$$E(y_m) = \lambda_\alpha r \sin(\alpha)\,, \quad (4.32)$$

where λ_α is the bias compensation factor given by

$$\lambda_\alpha = E(\cos(\tilde{\alpha})) = e^{-\sigma_\alpha^2/2}\,. \quad (4.33)$$

Thus, the conversion is biased if $\lambda_\alpha \neq 1$ and an unbiased conversion can be given when $\lambda_\alpha \neq 0$, as the polar-to-Cartesian unbiased conversion is given by

$$x_m^u = \lambda_\alpha^{-1} r_m \cos(\alpha_m) \quad (4.34)$$

$$y_m^u = \lambda_\alpha^{-1} r_m \sin(\alpha_m)\,. \quad (4.35)$$

The compensation of the bias is, therefore, multiplicative.

Similarly to what happened in the standard conversion, the elements of the covariance matrix are conditioned on the true measured values, which are unknown in practice. To solve this problem, the authors suggest that the squared error is averaged directly conditioned on the observations. In this new approach, the approximate measurement covariance matrix is given by

$$\mathbf{R}_u = \begin{bmatrix} R_u^{11} = var(x_m^u | r_m, \alpha_m) & R_u^{12} = cov(x_m^u, y_m^u | r_m, \alpha_m) \\ R_u^{21} = R_u^{12} & R_u^{22} = var(y_m^u | r_m, \alpha_m) \end{bmatrix}\,, \quad (4.36)$$

where

$$R_u^{11} = (\lambda_\alpha^{-2} - 2) r_m^2 \cos^2(\alpha_m) + 0.5(r_m^2 + \sigma_r^2)(1 + \lambda_\alpha^4 \cos(2\alpha_m)) \tag{4.37}$$

$$R_u^{22} = (\lambda_\alpha^{-2} - 2) r_m^2 \sin^2(\alpha_m) + 0.5(r_m^2 + \sigma_r^2)(1 - \lambda_\alpha^4 \cos(2\alpha_m)) \tag{4.38}$$

$$R_u^{12} = (\lambda_\alpha^{-2} - 2) r_m^2 \cos(\alpha_m)\sin(\alpha_m) + 0.5(r_m^2 + \sigma_r^2)\lambda_\alpha^4 \sin(2\alpha_m)\,. \tag{4.39}$$

4.3 Obstacle Models

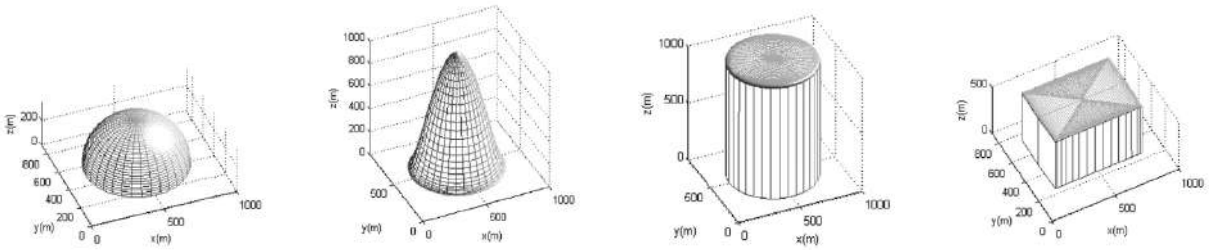
In a real flight environment, the UAV must avoid a wide diversity of obstacles while performing its designated application, which can vary from mountains and buildings to other aerial vehicles. These obstacles and threats can be modeled by many methods.

In reference [93], each point of the environment has a value associated that corresponds to the risk of flying in said zone. This value is computed from a normal distribution of the probabilistic risk of the area.

In reference [94], the obstacles are modeled in a more pragmatic way, where each one is defined by several parameters which impact its position, size and shape. In particular, the function

$$\Gamma = \left(\frac{x - x_0}{a}\right)^{2p} + \left(\frac{y - y_0}{b}\right)^{2q} + \left(\frac{z - z_0}{c}\right)^{2r} \tag{4.40}$$

was adopted to simplify the model of the obstacle, where a, b and c are the size parameters, p, q and r are the shape parameters and (x_0, y_0, z_0) is the position of the center of the obstacle. Points with $\Gamma = 1$ correspond to the surface of the obstacle while points with $\Gamma < 1$ are inside and points with $\Gamma > 1$ are outside. Several kinds of obstacles can be obtained by varying these parameters, including spheres and cylinders, as seen in Figure 4.5.



(a) Sphere (a=b=c, p=q=r=1)

(b) Cone (a=b, p=q=1, r<1)

(c) Cylinder (a=b, p=q=1, r>1)

(d) Cuboid (p>1, q>1, r>1)

Figure 4.5: Models of typical obstacles [94]

One big flaw of this model is the fact that it does not take the obstacle's velocity into consideration. However, associating a three-dimensional vector to the obstacle is enough to solve this problem.

The shape of the modeled obstacle also has a close relation with the kind of sensor used in its detection. A scanning laser rangefinder can only detect points of the obstacle in its scanning plane, so assuming a cylindrical shape for the obstacles detected with this kind of sensor may be an acceptable approximation model, which would not be the case for obstacles detected by three-dimensional sensors.

Chapter 5

Multisensor Data Fusion Techniques

When the sensing system is composed of multiple sensors, the input data from the sensors needs to be merged in some way. In this chapter, the concept of sensor fusion is explained and justified. Then, the weighted filter and the Parzen-like estimator techniques are explained and finally, one of the approaches is chosen, considering their advantages and drawbacks.

5.1 Data Fusion Overview

Multisensor data fusion consists on taking data obtained from a set of different sensors and combining them in order to achieve more accurate measurements and more specific inferences than a single sensor could achieve. Additionally, obtaining redundant observations of physical phenomena results on statistical advantages due to the improved obtained estimates.

In reference [95], the benefits of multisensor data fusion are showcased. The objective of said work was to sense possible collisions with moving vehicles detected by both a pulsed radar and stereo vision. The radar was able to measure the accurate distance to the obstacle but struggled to determine its angular direction with precision. By contrast, the imaging sensor had sufficient lateral resolution to find the boundaries of the obstacle, but was unable to measure range as accurately. The correct association of the observations obtained from both of these sensors resulted in a reduced error region which prompted an improved determination of the obstacle's location and occupying areas. The different error areas are displayed in Figure 5.1.

The data fusion techniques can be characterized by the inference level associated with the transformation between the observed quantity and the decision produced by the fusion estimation process regarding the existence, characteristics or identity of the obstacle, as seen in Figure 5.2 (a). The inference level of this transformation ranges from the observed signals to progressively more abstract concepts and depends on the application under consideration. For our application, the fusion is at a "position and/or velocity" level of inference, as the situation assessment and threat analysis are not to be conducted in this phase of the process.

Data fusion methods can also be categorized in regard to the level at which the process takes place.

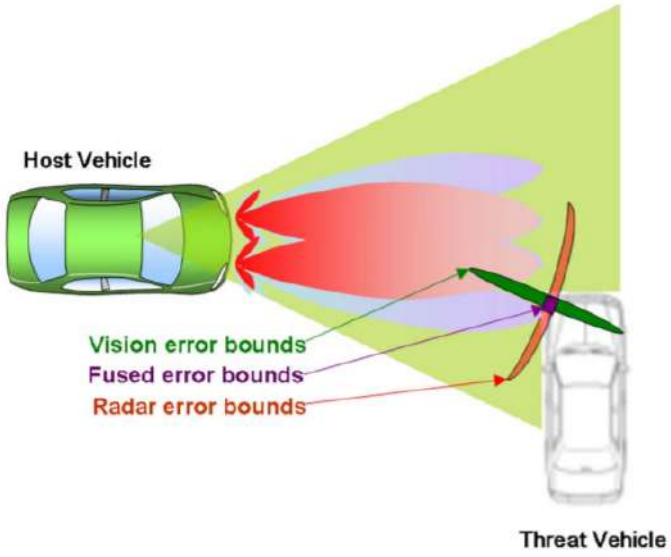
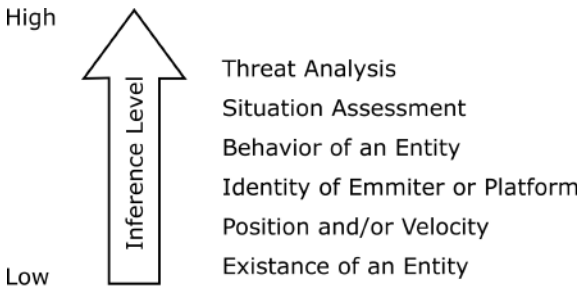
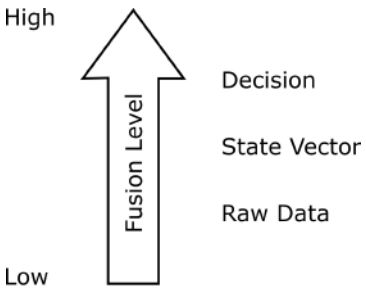


Figure 5.1: Error regions of a radar, a stereo vision sensor and the fusion of both (adapted from [95])

The data can be combined at a raw data level, at a state vector level or at a decision level, as illustrated in Figure 5.2 (b). Fusion at a raw data level is more common when several sensors of the same type are used and typically involve classic detection and estimation methods. State vector fusion is associated with the extraction of representative features from different sensor observations are joined into a single concatenated feature. Finally, decision level fusion is made after each sensor has already processed its data and computed the obstacle's location and attributes. Because sensors of different types are to be used in our work, the sensor fusion is to be made at a decision level, which will be more thoroughly detailed in following sections [96].



(a) Inference hierarchy of sensor fusion data (adapted from [96])



(b) Sensor fusion specificity hierarchy

Figure 5.2: Hierarchies that characterize sensor fusion techniques

5.2 Weighted Filter Technique

The weighted filter is a data fusion or data selection technique used in reference [64] to obtain the best distance measurements from multiple sensors. In this approach, a weight is computed for each of the sensors, which represents its reliability. Having computed all the sensor weights, the sensor with the

best result is picked to provide the measurements in the present iteration of the process.

To compute the weights, the UAV needs to be also equipped with reference data sensors, which provide information about the UAV's state. IMUs and optical flow sensors are examples of reference data sensors used to evaluate the reliability of the main sensor data and help to decide between those sensors, based on the rationale that changes in distance to obstacles correspond to analogous changes in the UAV's position. If the obstacles are stationary, these variations should coincide. If the obstacles are moving, this information becomes corrupted, but it is unlikely that this motion corresponds better to randomly wrong measurements.

The weights are computed by comparing all possible sensor combinations of main data and reference data using a differential norm, where the obstacle distance corresponding to the sensor with the lowest weight is considered the final result, while the remaining are rejected based on the idea that they are corrupted. However, if the computed weights have a low variation, the sensor values are fused according to their weights.

As an example, considering a UAV equipped with a stereo camera, a laser rangefinder and an optical flow sensor, the needed norms the laser rangefinder are

$$N_1 = |O_k - (L_k - L_{k-1})| \tag{5.1}$$

$$N_2 = |O_k - (L_k - D_{k-1})|\,, \tag{5.2}$$

where O_k is the position variation of the UAV between the two measurements computed from the data given by the optical flow sensor, L_k and L_{k-1} are the measurements obtained from the laser rangefinder at instants k and $k-1$ and D_{k-1} is the measurement used in the previous iteration by the chosen sensor (or fused result). With these norms computed, an exponential moving average filter can be applied with a smoothing factor e, so that the history can be accounted for. So, the first term of the filter is given by

$$W'_{L_k} = a_1 \times N_1 + a_2 \times N_2\,, \tag{5.3}$$

where a_1 and a_2 must be tuned to determine the influence of the different sources, and the filter itself is given by

$$W_{L_k} = e \times W'_{L_k} + (1 - e) \times W_{L_{k-1}}\,, \tag{5.4}$$

which corresponds to the weight for the laser rangefinder.

The weight for the stereo vision $W_{V_k}$is computed analogously and, if the difference between both weights is less than 10%, the overall distance is computed by the weighted average

$$D_k = \frac{L_k \times W_{L_k} + V_k \times W_{V_k}}{W_{L_k} + W_{V_k}}\,, \tag{5.5}$$

where V_k is the measurement obtained from the stereo vision sensor.

5.3 Parzen-like Estimator Technique

The Parzen estimator is a non-parametric method for estimating probability density functions without making any assumptions about the nature of the distribution. For a given set of sensors' data, this estimator uses a set of parametric functions such as Gaussian functions that are centered on the sensors' data. Then, these functions are added up and normalized. The resulting probability density function (PDF) reflects the distribution of the sensors' data with more energy concentrated where most data points exist [97].

Considering the parametric function as the sum of the true measurement value with additive white Gaussian noise with a standard deviation estimated from the noise level of the sensor's data, the PDF produced by this method is given by

$$PDF(x) = \frac{1}{N}\sum_{k=1}^{N}\frac{1}{\sigma_k\sqrt{2\pi}}\exp\left(-\frac{(x-x_k)^2}{2\sigma_k^2}\right), \tag{5.6}$$

where N is the number of redundant sensors, x_k is the k-th sensor data and σ_k is the standard deviation associated with the k-th sensor, estimated based on each sensor's specifications. In Figure 5.3, an example of a PDF obtained from three different sensor measurements is presented. In this example, all measurements were performed by similar sensors with the same standard deviation (1 m).

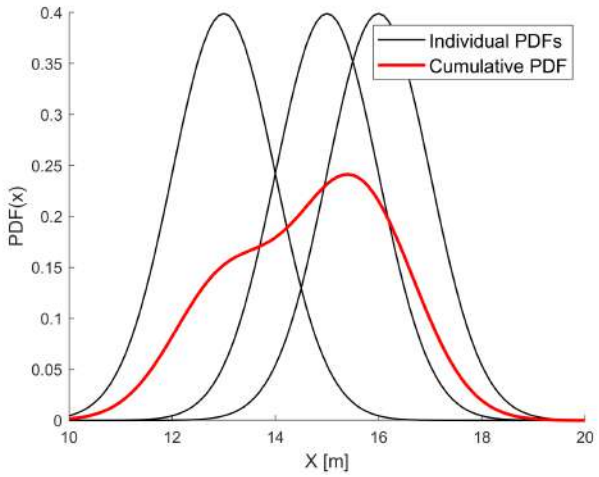


Figure 5.3: PDFs corresponding to measurements and corresponding cumulative PDF

Knowing that the centroid of the PDF can be computed from

$$\frac{\int_X x\,PDF(x)\,dx}{\int_X PDF(x)\,dx}, \tag{5.7}$$

a value resulting from the fused data can be extracted from the computed PDF, following the steps [97]:

1. Find the range X which contains 95% of the PDF's energy;
2. Find the centroid of the PDF;
3. Find the area on each side of the centroid of the PDF;

4. Get the value of the measurand that corresponds to the supreme of the PDF on the side of the centroid with highest area.

5.4 Benchmark of Techniques

The chosen technique for our work was the weighted filter. Firstly, this technique is less computationally demanding than the Parzen-like estimator, as only some computations need to be made in order to calculate the weight associated with each sensor. Moreover, in the second approach, the estimated value does not explicitly reflect the degree of agreement between the sensors. This problem could be solved by introducing the computation of a confidence level associated with each sensor that would impact the weight given to its PDF [97]. However, this would require more calculations in each step of the detection phase and an *a priori* study of each sensor's data distribution to obtain its initial confidence level.

It would be feasible to perform these additional steps but the weighted filter methodology is simpler and already adequate to perform the data fusion.

Chapter 6

Obstacle Detection and Avoidance Algorithms

In this chapter, an analysis is made on several collision detection methods, since not all detected obstacles interfere with the UAVs trajectory and require an avoidance maneuver to be made. In the next section, an overview of several approaches to change a predefined reference path in order to avoid detected objects is made and the attributes and disadvantages of each one are discussed. Subsequently, the collision detection and obstacle avoidance algorithms used in a previous work [98] are explained in greater detail. Portions of the code needed to be changed to account for different sensor characteristics and these modifications are also discriminated in this chapter.

6.1 Collision Detection

After receiving the obstacle data from the equipped sensors, this data needs to be processed, since only a few will require an avoidance maneuver to take place. The existence of this processing phase helps to save computing time and power during the avoidance phase, considering that some of the avoidance algorithms can be very demanding. Moreover, some of these algorithms are impracticable when run on their global form but feasible on their local form. Also in this phase, the list of obstacles is ordered according to the imminence of the detected collisions. The layout of a typical and general collision detection module is displayed in the flowchart of Figure 6.1 and several approaches described in the survey [99] are presented in the following.

The distance estimation is a very common collision detection method due to its simplicity and easy implementation. In this method, a safety radius is defined around the UAV and the shortest distance between the UAV and the obstacle is computed, whether it is a static or a dynamic obstacle. If this distance is smaller than the defined safety radius, then an evasive maneuver is made.

The act as seen approach is an even simpler collision detection method, which consists on acting defensively as soon as an obstacle is detected. This method is mainly used by UAVs equipped with vision sensors and it must be used when the avoidance method is equally simple, as more complex

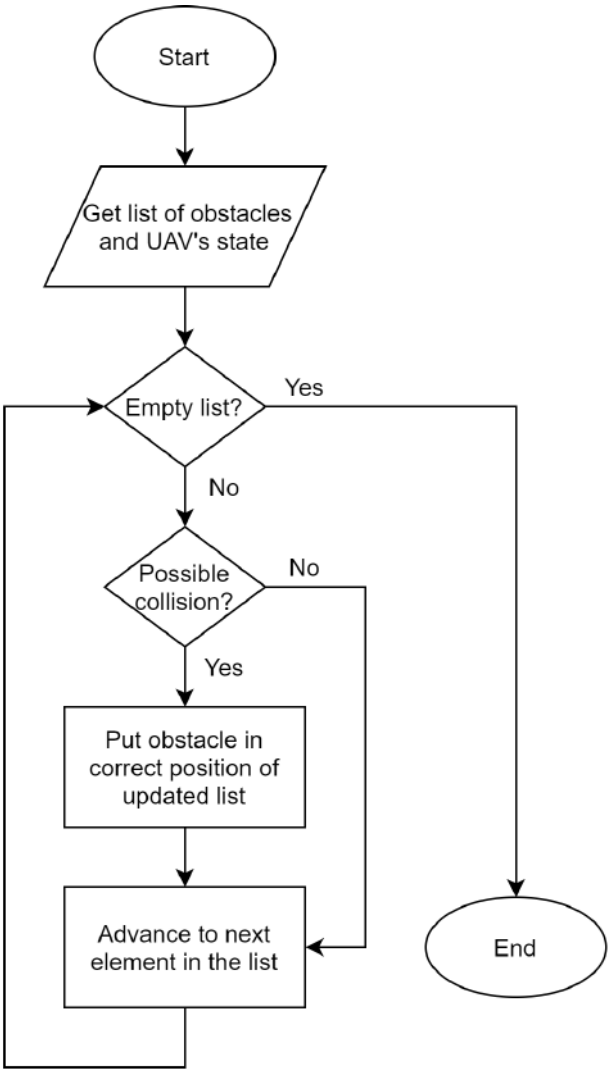


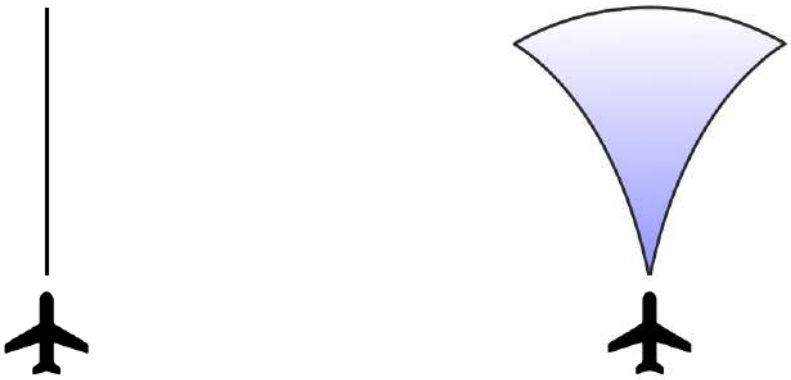
Figure 6.1: Flowchart of a general collision detection phase

avoidance algorithms could be easily overwhelmed by having to compute avoidance strategies to every single detected object [100].

In the previous methods, the trajectory of moving obstacles is not taken into account and if one wishes to consider the future position of these obstacles, several predictions can be made with varying degrees of simplicity and effectiveness.

The simplest projection method is the straight projection method, where the current state of the obstacle is projected into the future along a straight trajectory made at a constant velocity (6.2 (a)). This method can be only used in situations in which the aircraft trajectories are very predictable and for a short period of time. The other extreme is the worst case scenario, which assumes the obstacle will perform any possible maneuver after being detected (6.2 (b)). If any of the possible trajectories comes into conflict with the UAV, a collision is predicted and an avoidance maneuver is performed. This approach is the most effective but the most inefficient. Therefore, it should also be limited to a short period of time to limit the computation requirement for risk assessment [101].

The other projection method is the probabilistic one, where uncertain factors and their probability



(a) Straight projection (b) Worst case projecion

Figure 6.2: Two methods used for projecting the UAVs trajectory

distributions act as variables. In reference [102], the considered factors were the UAV's and obstacle's lateral position, vertical position, along-track speed, cross-track position, the obstacle's heading change and altitude change. Using this model, the probability of collision is computed using Monte Carlo simulations. In Monte Carlo simulations, risk analysis is performed by establishing models of possible results by substituting a range of values for any factor that has inherent uncertainty. It then calculates results repeatedly, each time using a different set of random values from the probability functions. The drawback of this approach is the excessive computational power required to run the Monte Carlo simulations several times with such a number of uncertain factors, which would be unfeasible for a mini UAV in real time.

Based on the previous remarks, the code used to perform our simulations uses the straight projection method which performs well while being the one that requires the least computational power and time of all the discussed approaches (excluding the act as seen approach which would not be feasible). Besides this, elements from the distance estimation approach are also used, not only due to its simplicity but also because the chosen avoidance algorithm relies heavily on the distance between the UAV and the obstacle, as discussed in the next section. Additionally, the picked method will be discussed in detail in section 6.3.

6.2 Obstacle Avoidance Algorithms

After having the obstacle information returned by the sensors and a possible collision detected, the UAV must apply its avoidance strategy to ensure that a safety distance is kept between itself and the detected obstacles. To achieve this, a local path is generated so that the threatening obstacles are avoided and only when there are no more obstacles representing potential collisions does this module export its generated local path to the flight controller, so that it can be followed. Before describing the next stage of the avoidance algorithm, it is important to distinguish global WPs from WPs belonging to the global path: global WPs are considered mandatory and the UAV must cross them to complete the

present flight phase, while WPs belonging to the global path connect two global WPs because these are the waypoints that define the trajectory and they are to be followed by the flight controller but are not mandatory to cross, as the UAV can deviate from the global path if an obstacle is blocking it. Because of this, following the obstacle avoidance, if an obstacle was sensed blocking the next global waypoint, the algorithm is also responsible for generating a local path to the missed global WP in order to complete the present flight phase. However, if the blocking obstacle is static, the global waypoint is considered unreachable and it is discarded from the flight plan, so that the UAV can continue its mission by moving to the next global WP. Only the current global WP can be excluded in this phase, as discarding future global WPs would be too computationally demanding if the global path had to be recalculated and would be inconsequential otherwise. Considering all this, the flowchart of a general collision avoidance module is displayed in Figure 6.3.

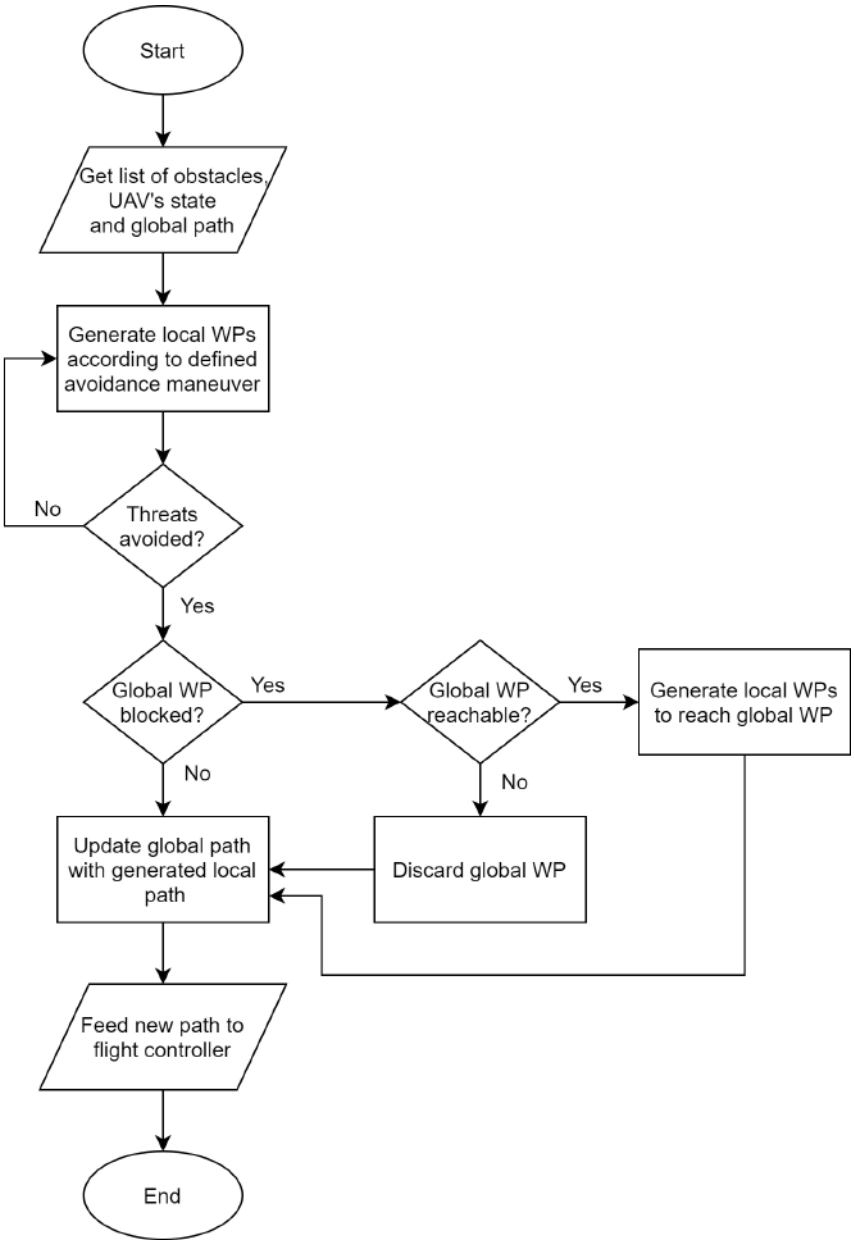


Figure 6.3: Flowchart of a general collision avoidance phase

Several avoidance techniques can be used to generate a local path around the obstacles and, in references [59, 99, 103], these approaches are divided in different categories, which include geometric, path planning, bearing angle based and potential field, all of which will be briefly described next.

In the geometric approach, the collisions are detected and avoided through a geometric analysis by simulating the trajectories of the UAV and the obstacles. To accomplish this, both vehicles' velocity, heading and position are needed and because of this, this method is most associated with the use of ADS-B sensors. A typical use of this approach is demonstrated in reference [104] where, given the vehicles' motion in a two-dimensional environment and considering they keep linear trajectories, the subtraction of their movement vectors at the Closest Point of Approach (CPA) is computed and, in case of a possible collision, their trajectories are changed to widen this vector. However, this conflict resolution process fails if a high number of UAVs have to avoid each other, since applying this algorithm to each obstacle, one at a time, does not always result in a successful conflict avoidance.

The path planning approach is a grid based method that utilizes the path re-planning algorithm with graph search algorithm to find a collision free trajectory during the flight. It has some similarities with the previous approach due to relying on some trajectory calculation in a geometric way. However, by using this approach, the generated trajectory is usually the most optimized one, where all obstacles are avoided while the predefined trajectory is still maintained at a short distance. This approach is more suited for static obstacles and their size and position must be known. This approach was adopted in reference [103], where the environment was discretized by dividing the map into a grid, represented by a weighted graph. This grid enables finding a collision free path by using a graph search algorithm like A* [103], such that in case of an obstacle detection, a path re-planning is performed by calculating the shortest path from the actual position to the goal while avoiding the edges of the graph that connect cells within the obstacle's safety distance. An example of this resolution is displayed in Figure 6.4.

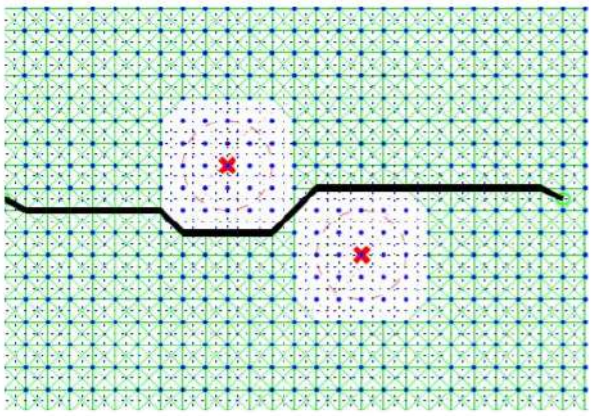


Figure 6.4: Obstacle avoidance by path planning with A* search [103]

The bearing angle based approach relies on visual sensors and their ability to return relative angle of obstacles toward the UAV. To avoid the collision using this method, the obstacle has to be kept at a "safe" position in the sensor's FOV, which results in spiral flight paths. This method has all the disadvantages of the vision sensors, being affected by external conditions that can affect the image and relying on image processing techniques. In reference [100], computing the distance and time to collision is not

required, since this method relies completely on the obstacle's bearing and UAV's heading to avoid obstacle. The maneuver starts as soon as the obstacle is detected, while the stopping decision is based on thresholding a cost function.

Lastly, in the potential fields approach, the waypoints and obstacles are considered charged particles with associated potentials that affect the UAV's movement according to its proximity to obstacles or to a waypoint. However, computing the potential of each point of the plane would not be feasible, as it would require too much time and computing power. Because of this, the potential gradient is instead used, which correlates to a force to be exerted on the UAV. A potential field generated by a waypoint and an obstacle is represented in Figure 6.5. In reference [105], an analytically tractable potential field model of free space is presented, which assumes that the border of every two-dimensional region is uniformly charged. The Newtonian potential function was used and simulation results showed that the collision avoidance was effective and the resulting path was smooth.

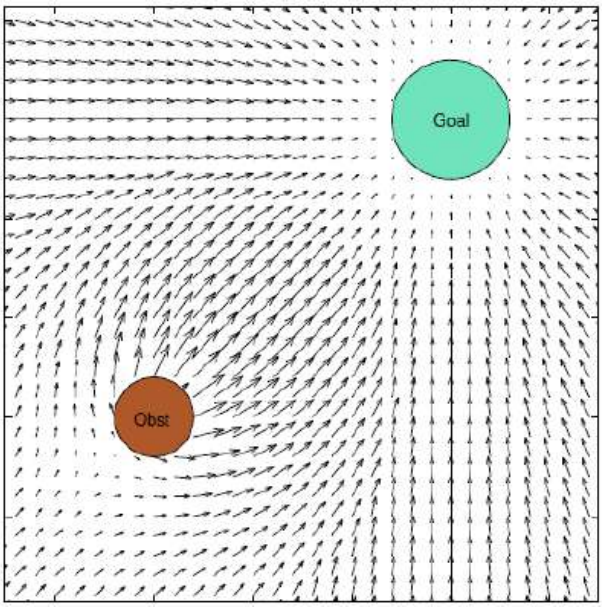


Figure 6.5: Potential field generated by an obstacle and a goal [106]

The approach used in our legacy code [98] is the potential field method, although the collision detection method, as described before, shares several similarities with the geometric approach. The other methods were not so suited to the resolution of our problem, as the path planning approach is more appropriate for avoiding previously known static obstacles and the bearing angle based approach restricts the kind of sensors used due to relying heavily on vision sensors. The actual approach will be described in greater detail in the following section.

6.3 Description of Adopted Algorithm

6.3.1 Safety Zones

In the developed algorithm, each detected obstacle has several safety zones associated with it, which play a role in the collision detection phase as well as in the avoidance phase. The obstacles were modeled as spheres and, as such, the collision radius (R_c) defines the volume of the obstacle and a collision is said to occur if this radius is trespassed. The safety radius (R_s) defines the minimum distance that should be maintained between the UAV and the obstacle to take into account possible deviations and uncertainties that could happen during the detection and path prediction phases. The action radius (R_a) is the distance from which the replanned paths begins to depart from the original path given by the global planner. Lastly, the detection radius (R_d) represents the distance from which an obstacle is considered by this algorithm. The R_s should be similar to the UAV's size and depend on the UAV's speed and type of obstacle approach, as demonstrated in reference [98]; the R_a should be comparable to the R_s and the R_d should correspond to the range of the sensors used. A representation of the described safety zones is displayed in Figure 6.6.

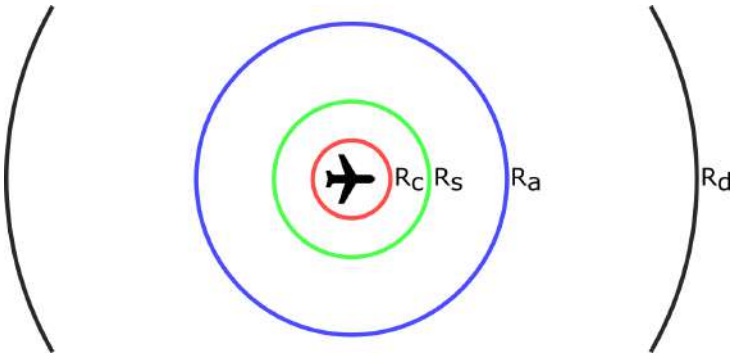


Figure 6.6: Representation of the safety zones around an obstacle

6.3.2 Geometric Collision Detection Method

As seen in section 6.1, the chosen collision detection method computes straight projections of the obstacles, considering future distances between the obstacles and the UAV, like in the distance estimation approach, which results in a collision detection method similar to the geometric approach (the collision avoidance method).

As such, the resulting collision detection method consists of computing the closest point of approach (CPA) between the UAV and the target, assuming that both vehicles will maintain constant velocities and rectilinear paths. With these considerations, the motion of two vehicles, A and B, can be described as $A(t) = A_0 + \mathbf{v}_A t$ and $B(t) = B_0 + \mathbf{v}_B t$ and the distance between them at an instant t is given by $||A(t) - B(t)||$. By solving its derivative, the instant corresponding to the minimum distance between the two points (t_{CPA}) is given by

$$t_{CPA} = \frac{-(A_0 - B_0) \cdot (\mathbf{v}_A - \mathbf{v}_B)}{||\mathbf{v}_A - \mathbf{v}_B||^2}. \tag{6.1}$$

Knowing the value of t_{CPA}, the minimum distance between the two vehicles can be easily computed, as

$$d_{CPA} = ||A(t_{CPA}) - B(t_{CPA})||. \tag{6.2}$$

Finally, if this distance is smaller than the safety radius, an evasive maneuver must be performed, otherwise the obstacle is not considered a threat to the UAV. Since the linear projection is not the most effective, these computations will have to be made regularly to take into account maneuvers performed by the UAV itself and by non-threatening obstacles that can become threatening after said maneuver. Furthermore, the installed sensors can detect new obstacles, whose risk will also have to be evaluated.

In case of multiple collisions being detected, the obstacles are sorted according to their t_{CPA}, so that the obstacles associated with possible collisions that would happen first are avoided before the remaining ones.

The adopted algorithm with all of the elements described above is presented in Pseudo-code 1.

Input: UAV's position A, UAV's speed $\mathbf{v}_A$, obstacle's position B, obstacle's speed $\mathbf{v}_B$, obstacle safety radius R_s
Output: Flag indicating possible colision $flag$, t_{CPA}, Obstacle's position at CPA B_{CPA}
begin
  set $t_{CPA} = \frac{-(A_0 - B_0)\cdot(\mathbf{v}_A - \mathbf{v}_B)}{||\mathbf{v}_A - \mathbf{v}_B||^2}$
  set $B_{CPA} = B + \mathbf{v}_B \times t_{CPA}$
  set $A_{CPA} = A + \mathbf{v}_A \times t_{CPA}$
  set $d_{CPA} = ||A_{CPA} - B_{CPA}||$
  if $t_{CPA} > 0$ *and* $d_{CPA} < R_s$ **then**
    set $flag = 1$
  else
    set $flag = 0$
  end
end

Pseudo-code 1: Collision avoidance algorithm

6.3.3 Avoidance Strategy

Before explaining the avoidance method, the avoidance strategy needs to be defined. As stated in section 1.3, UAVs must always give way to manned aircraft and, since the sensing is assumed to be non-cooperative, the UAV will always make an evasive maneuver to avoid a collision, while respecting the Rules of the Air [107]. So, in case of an imminent collision, the avoidance strategy is the following:

- In level flight, if the intruder is in a head-on collision path or to the right of the UAV, the avoidance algorithm makes the UAV turn right;
- In level flight, if the intruder is approaching from the left, the UAV turns left to avoid being in front of the intruder;
- While climbing, upon detecting an intruder with a high risk of collision, the UAV levels off until the obstacle is overcome;

- While descending, when a target is detected, the UAV increases its rate of descent (unless it is at its maximum), which is easier than leveling off due to the vehicle's inertia;
- In case of a static obstacle, the direction in which the obstacle is circled is the one that corresponds to the smaller path to reach the goal.

6.3.4 Potential Fields Method

To solve the local path planning problem, the Potential Fields approach is used, where the waypoints and obstacles are considered charged particles. This approach was used in the previous works [98, 108]. Considering this analogy, the waypoints generate an attractive field, the obstacles a repulsive field and the sum of all forces is used to generate the direction of motion. However, punctual charges are not enough for the UAV to have an appropriate behavior. Because of this, several changes to this method needed to be done so that the UAV could follow a path and avoid an obstacle in a smooth and regular manner.

Attractive potential

The attractive potential is given by

$$\mathbf{f}_{at} = \alpha_{PF} \frac{P_{close} - P}{||P_{close} - P||} + (1 - \alpha_{PF}) \frac{P_{next} - P_{close}}{||P_{next} - P_{close}||}, \tag{6.3}$$

where the first term is responsible for guiding the UAV to the nearest point of the global path and the second term is responsible for guiding the UAV to the next defined waypoint. P is the UAV's position and P_{close} is the closest point of the global path. Subtracting both positions and dividing by its norm results in a unit vector pointing from the UAV to the closest point of the path. The second term functions under the same principles, where P_{next} is the position of the next waypoint. The α_{PF} term is responsible for giving more or less predominance to each term.

An example of a global path to a waypoint and its corresponding attractive potential field is represented in Figure 6.7 for $\alpha_{PF} = 0.7$.

Repulsive potential

Using a simple repulsive potential to avoid obstacles is not feasible since that would lead to irregular motion around the obstacle. Adding a swirling motion to the potential flow solves this problem and makes the evasion start sooner, since the UAV will evade the obstacle instead of just keeping the distance to the obstacle.

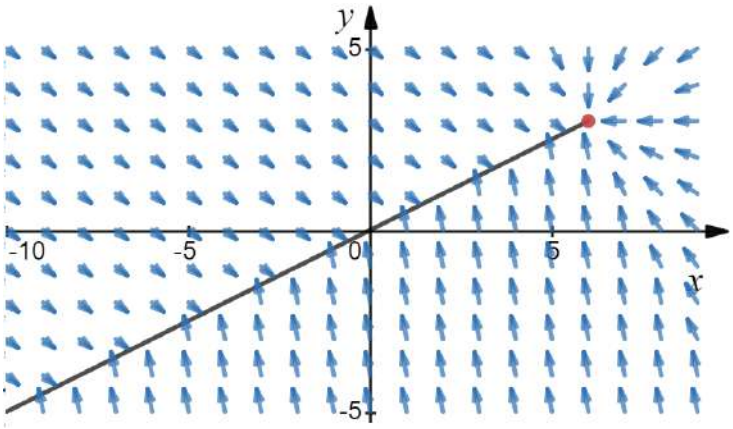


Figure 6.7: Attractive field for a linear path with α_{PF}=0.7

The swirling term is given by

$$\mathbf{s}_{dir} = \frac{\hat{\mathbf{k}} \times \mathbf{d}_0}{||\mathbf{d}_0||}, \tag{6.4}$$

where $\hat{\mathbf{k}}$ is the unit vector in the z direction (0,0,1) and $\mathbf{d}_0$ is the vector pointing from the obstacle to the UAV. Computing $\mathbf{s}_{dir}$ this way results in a vector that makes the UAV circumvent the obstacle counterclockwise. When the obstacle needs to be circled in the other direction, the symmetric of $\mathbf{s}_{dir}$ is used.

To avoid the UAV being trapped around the obstacle, the generated field needs to become zero once the obstacle is overcome. To check this condition, an angle θ between the desired direction of motion ($\mathbf{m}$) and the direction of the obstacle needs to be computed from

$$\theta = arccos\left(\frac{\mathbf{m} \cdot \mathbf{d}_0}{||\mathbf{m}||||\mathbf{d}_0||}\right). \tag{6.5}$$

If this angle is larger than a $\theta_{cut-off}$, the obstacle will have been overcome, making the field null.

Knowing this, the potential associated to the obstacle is represented in Figure 6.8 and it is described by

$$\mathbf{f}_{rep} = \begin{cases} \infty \frac{\mathbf{d}_0}{||\mathbf{d}_0||} & \text{, if } ||\mathbf{d}_0|| \leq R_c \\ S_{max}\mathbf{s}_{dir} & \text{, if } R_c < ||\mathbf{d}_0|| \leq R_s \\ S_{max}\frac{R_a - ||\mathbf{d}_0||}{R_a - R_s}\mathbf{s}_{dir} & \text{, if } R_s < ||\mathbf{d}_0|| \leq R_a \\ 0 & \text{, if } ||\mathbf{d}_0|| \geq R_a \vee \theta \leq \theta_{cut-off} \end{cases}. \tag{6.6}$$

This field is different according to the distance between the obstacle and the UAV. If the UAV is in the collision zone, the field will be repulsive with infinite intensity. If it is in the safety zone, the field will have the direction of $\mathbf{s}_{dir}$, previously defined, and the intensity of S_{max}, a constant to be defined depending on the velocity of the UAV. In the action zone, the field is similar to the previous one but with the addition of a gradient term that ensures the intensity of the field decreases linearly with the distance of the UAV to the obstacle until becoming null for $||\mathbf{d}_0|| = R_a$. Lastly, outside the action zone, the obstacle has no

influence in the motion of the UAV, thus the filed intensity is null.

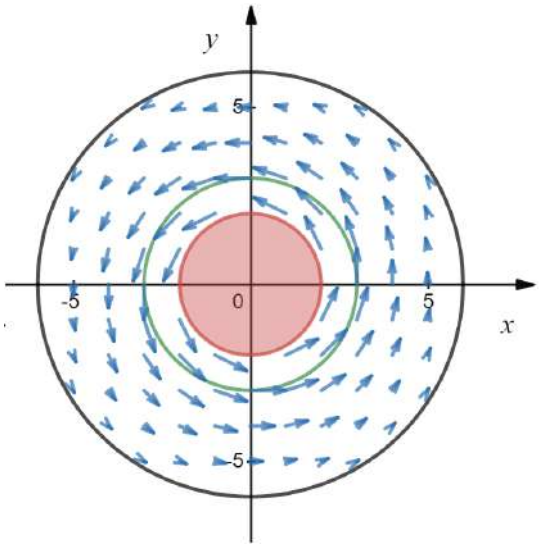


Figure 6.8: Repulsive field for an obstacle with $R_c = 2$, $R_s = 3$ and $R_a = 6$

6.3.5 Pseudo-code of Avoidance Algorithm

The avoidance algorithm is represented in pseudo-code 2 and it includes two functions: $P_{close} =$ find_closest$(P, path)$, which finds the point of the global path closest to the position of the UAV; and **v** =get_velocity$(\mathbf{f}, \mathbf{v})$, which computes the direction of the force F and creates a velocity vector with this direction and the magnitude of V. The variable mov is computed previously and indicates the direction the UAV will take in order to avoid the obstacle according to their positions. Moreover, as stated before, the constants S_{max} and θ_{cut} need to be defined according to the size and speed of the UAV in order to obtain the most effective avoidance maneuvers.

Input: Position of UAV P, speed of UAV $\mathbf{v}$, position at CPA of obstacle P_{CPA}, radius of obstacle R_c, position of next global WP P_{next}, safety radius R_s, action radius R_a, weighing term α_{PF}, points of global path $path$, time step dt, variable indicating avoidance strategy mov
Output: Flag indicating if collision was avoided $flag$, next point of local path P_L
begin
set P_{close} =find_closest$(P, path)$
set $\mathbf{f}_{at} = \alpha_{PF} \frac{P_{close}-P}{||P_{close}-P||} + (1-\alpha_{PF}) \frac{P_{next}-P_{close}}{||P_{next}-P_{close}||}$
set $\mathbf{d}_0 = P - P_{CPA}$
if $mov == 1$ **then**
set $\mathbf{s}_{dir} = \frac{\hat{\mathbf{k}} \times \mathbf{d}_0}{||\mathbf{d}_0||}$
else
set $\mathbf{s}_{dir} = -\frac{\hat{\mathbf{k}} \times \mathbf{d}_0}{||\mathbf{d}_0||}$
end
if $||\boldsymbol{d}_0|| \leq R_c$ **then**
$\mathbf{f}_{rep} = \infty \frac{\mathbf{d}_0}{||\mathbf{d}_0||}$
else if $R_c < ||\boldsymbol{d}_0|| \leq R_s$ **then**
$\mathbf{f}_{rep} = S_{max} \mathbf{s}_{dir}$
else if $R_s < ||\boldsymbol{d}_0|| \leq R_a$ **then**
$\mathbf{f}_{rep} = S_{max} \frac{R_a - ||\mathbf{d}_0||}{R_a - R_s} \mathbf{s}_{dir}$
else
$\mathbf{f}_{rep} = 0$
end
set $\theta = arccos\left(\frac{\mathbf{f}_{at} \cdot \mathbf{d}_0}{||\mathbf{f}_{at}|| ||\mathbf{d}_0||}\right)$
if $\theta > \theta_{cut}$ **then**
set $\mathbf{f} = \mathbf{f}_{at}$
set $flag = 1$
else
set $\mathbf{f} = \mathbf{f}_{at} + \mathbf{f}_{rep}$
set $flag = 0$
end
set $\mathbf{v}$ = get_velocity$(\mathbf{f}, \mathbf{v})$
set $P_L = P + \mathbf{v} \times dt$
end

Pseudo-code 2: Collision avoidance algorithm

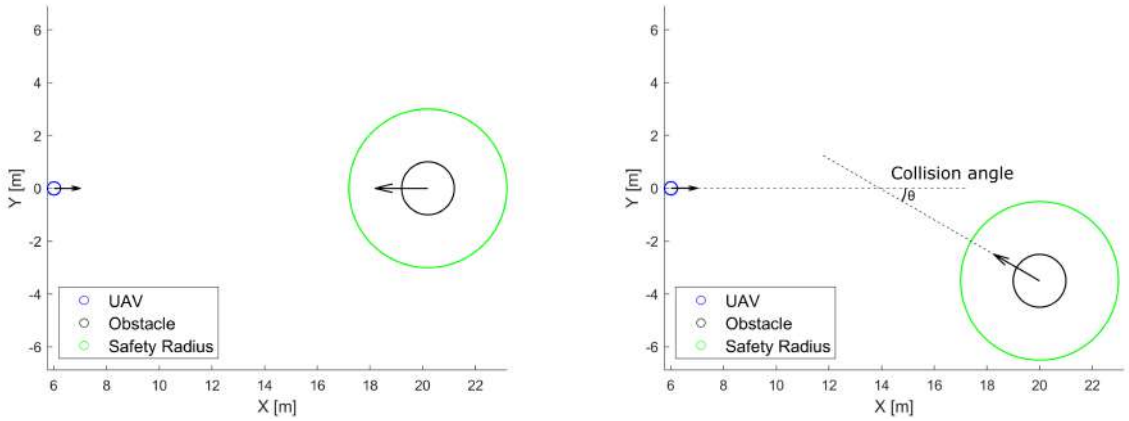
Chapter 7

Sensor Parametric Studies

The main focus of this chapter is on studying the response of the UAV to an imminent collision, when it is equipped with sensors with different parameters. However, before making this study, the collision scenarios need to be defined and several parameters associated with the UAV's dynamics and the Kalman tracking need to be carefully determined, namely the UAV's speed, maximum angular velocity and the measurement error covariance matrix. All the simulations were performed using MATLAB R2018a.

7.1 Collision Scenarios

The possible scenarios, in which the simulations present in this chapter were performed, include a head on collision course (Figure 7.1 (a)) and an angled collision course (Figure 7.1 (b)).



(a) Head-on approach

(b) Angled approach

Figure 7.1: Scenarios used for the simulations

The positions and velocities of the vehicles in these figures are arbitrary and are better defined in the sections referring to the particular simulations. For example, in the case of the angled approach, the initial position of the obstacle needs to be adjusted in accordance to the chosen velocities, so that the obstacle crosses the UAV's path dangerously.

7.2 UAV's Speed and Angular Velocity

Before performing the various simulations, some UAV characteristics need to be defined. Since the simulations were performed in a two-dimensional environment, the two most important performance parameters to define are the UAV's speed (V) and the maximum angular velocity (ω) of its turns.

Considering the example of the Tekever AR4, described in section 1.3, the UAV is considered to travel with a speed between 8 m/s and 15 m/s. The faster the UAV moves, the larger its angular velocity needs to be so that the obstacles can be effectively avoided. To prove this concept and to get the lowest maximum angular velocity needed for different possible UAV speed values, a series of simulations were performed. In these simulations, the UAV was set in a head-on collision course with an obstacle with a 2 m radius and a safety radius of 2 m. The obstacle approaches with the same speed as the UAV, which is equipped with a RADAR with a 50° FOV and a 120 m range. For each of the tested speeds, the maximum angular velocity was decreased until the UAV could not perform the avoidance maneuver without breaching the safety zone around the UAV. The results obtained from these tests are presented in Figure 7.2, in which a linear dependency can be recognized.

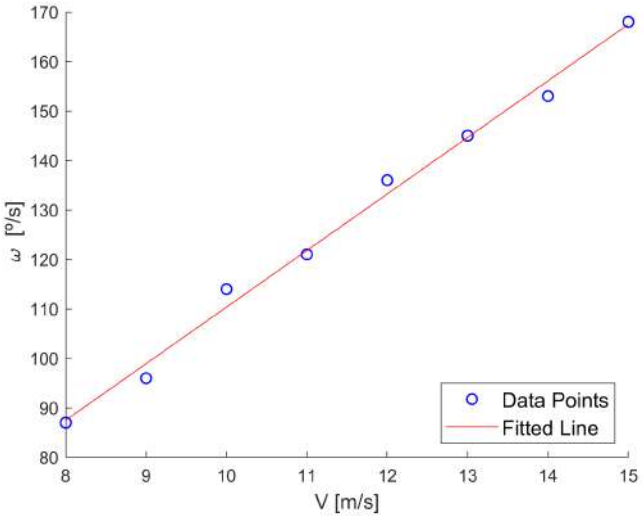


Figure 7.2: Angular velocity needed to avoid obstacle safely for different speeds

To check the validity of these results, the avoidance maneuvers can be approximated to coordinate turns. A coordinate turn is a turning maneuver made at a vertical and constant angular velocity and in which the lateral component of the resulting force is null. Considering that there is no wind, the slip is almost null and the angle of attack and climb angle are very small, the speed and the angular velocity are related by

$$tan(\phi) = \frac{\omega V}{g}\,, \tag{7.1}$$

where ϕ is the correspondent roll angle and g is the standard acceleration due to gravity ($\approx$ 9.81 m/s^2) [109].

The avoidance maneuver is not performed at a constant angular velocity, but this approximation can

be made to check the roll angles corresponding to the speeds and angular velocities obtained. All the obtained values are displayed in the Table 7.1.

Table 7.1: Roll angles for each maneuver considering the performance of a coordinate turn

Speed (m/s)	Angular Velocity(°/s)	Roll Angle (°)
8	87	51.1
9	96	57.0
10	114	63.8
11	121	67.1
12	136	71.0
13	145	73.4
14	153	75.3
15	168	77.4

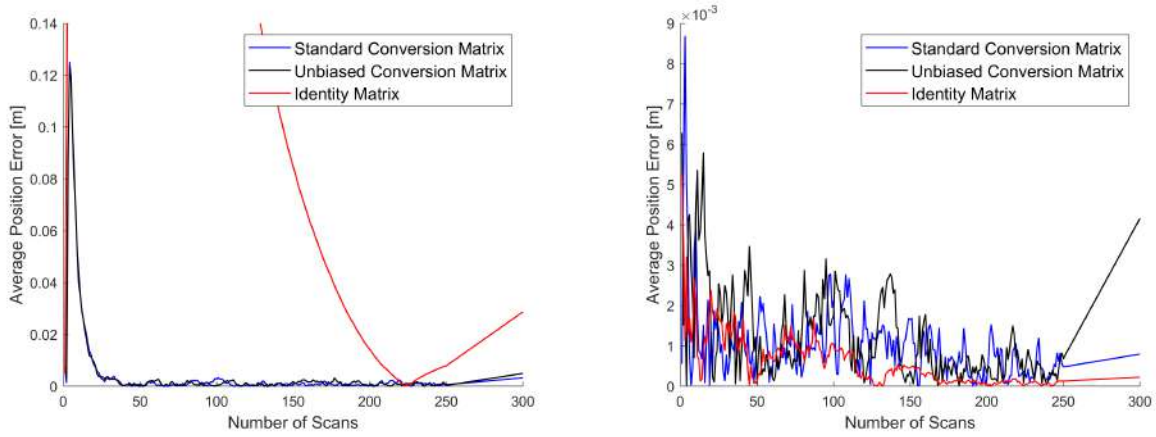
The obtained roll angles are acceptable considering the urgency of the maneuvers needed to avoid obstacles. For the highest speed of 15 m/s, the corresponding roll angle is already pretty high but achievable nonetheless. So, for the next simulations, the maximum angular velocity of the turns was set to 168°/s.

7.3 Measurement Error Covariance Matrices

As described in section 4.2.3, using polar measurements and a Cartesian state space leads to inaccuracies when tracking the obstacles with a Kalman filter. To test the two measurement error covariance matrices described in that section, the UAV was put in a head-on collision course (Figure 7.1 (a)), where the UAV is moving at 8 m/s and the obstacle is moving at 10 m/s. The UAV was equipped with a RADAR with the specifications of the Aerotenna μSharp Patch (100 m range, 50° FOV and 0.22 m accuracy). The noise was divided into a radial and an angular component, where both components were modeled as a zero-mean Gaussian noise, with the corresponding variance chosen so that 99.73% of the set would be within the accuracy range. The angular accuracy was considered at half the sensor range (50 m).

To test the standard conversion matrix and the unbiased covariance matrix, one hundred simulations were performed for both matrices and the average position errors were computed for both Cartesian coordinates. The identity matrix was also tested as a control group. The results are presented in Figure 7.3.

For the two studied matrices, during the first scans, the position errors are considerably high, but after only 50 scans, they are below the 0.01 m mark for the x coordinate and remain at this order of magnitude for the remaining scans. For the y coordinate, the initial overshoot is much smaller compared to the one of the x coordinate, due to the nature of the chosen scenario, where neither the obstacle nor the UAV have velocity in the y direction. For this reason, the identity matrix produces a very small error in the y direction but, in the x direction, produces an error one order of magnitude above the other matrices.



(a) X coordinate

(b) Y coordinate

Figure 7.3: Average position errors for different measurement error covariance matrices

After 250 scans, the UAV initiates the avoidance maneuver and the obstacle is no longer detected by the sensor. So, after this point, the uncertainties present in the tracked velocity starts to propagate, which results in linear increases of the position errors. In the x direction, the slopes of the average position errors are very similar for both matrices but, in the y direction, using the unbiased conversion results in a much faster deviation. For both coordinates, using the standard conversion matrix results in a slower deviation, despite both deviations being acceptable, considering that 50 scans later, the average position errors are still two order of magnitude below the obstacle's radius.

The root mean square (RMS) deviation for the three matrices and for both coordinates is represented in Table 7.2. Only the first 250 scans were considered in the computations, so that the points where the obstacle is not detected would not influence this metric.

Table 7.2: Root mean square deviations for each of the used matrices

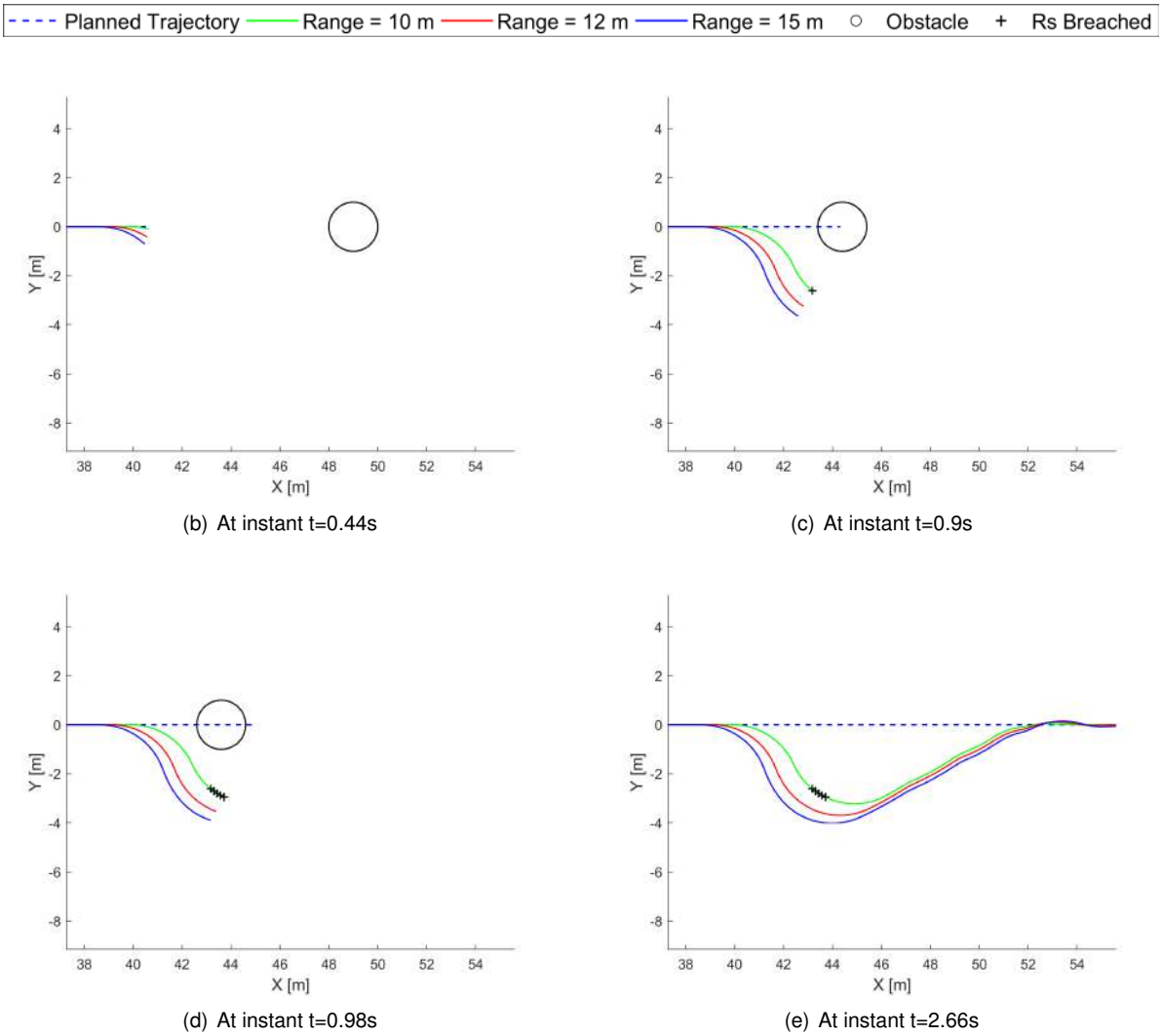
RMS [m]	x axis	y axis
Standard conversion matrix	0.0165	0.0013
Unbiased conversion matrix	0.0163	0.0016
Identity matrix	0.5993	0.0009

From Table 7.2, one can conclude that using the standard conversion or the unbiased conversion result in very similar outcomes. This may be due to the particular conditions of our study, where the sensor range and the noise variance are not very high. In spite of the similar results, the unbiased conversion matrix was used in all the following simulations.

7.4 Varying Range Simulations

To study how the sensor's range influences the response of the UAV to detected obstacles, the UAV was set in a head-on collision course with an obstacle with a radius of 2 m, a safety radius of 4 m and animated with a speed of 10 m/s. The UAV is equipped with a RADAR sensor with a FOV of 50° but,

for this kind of experience, the type of sensor used does not affect the results much. In Figure 7.4, the behavior of the UAV when animated with a speed of 8 m/s is illustrated.



(b) At instant t=0.44s

(c) At instant t=0.9s

(d) At instant t=0.98s

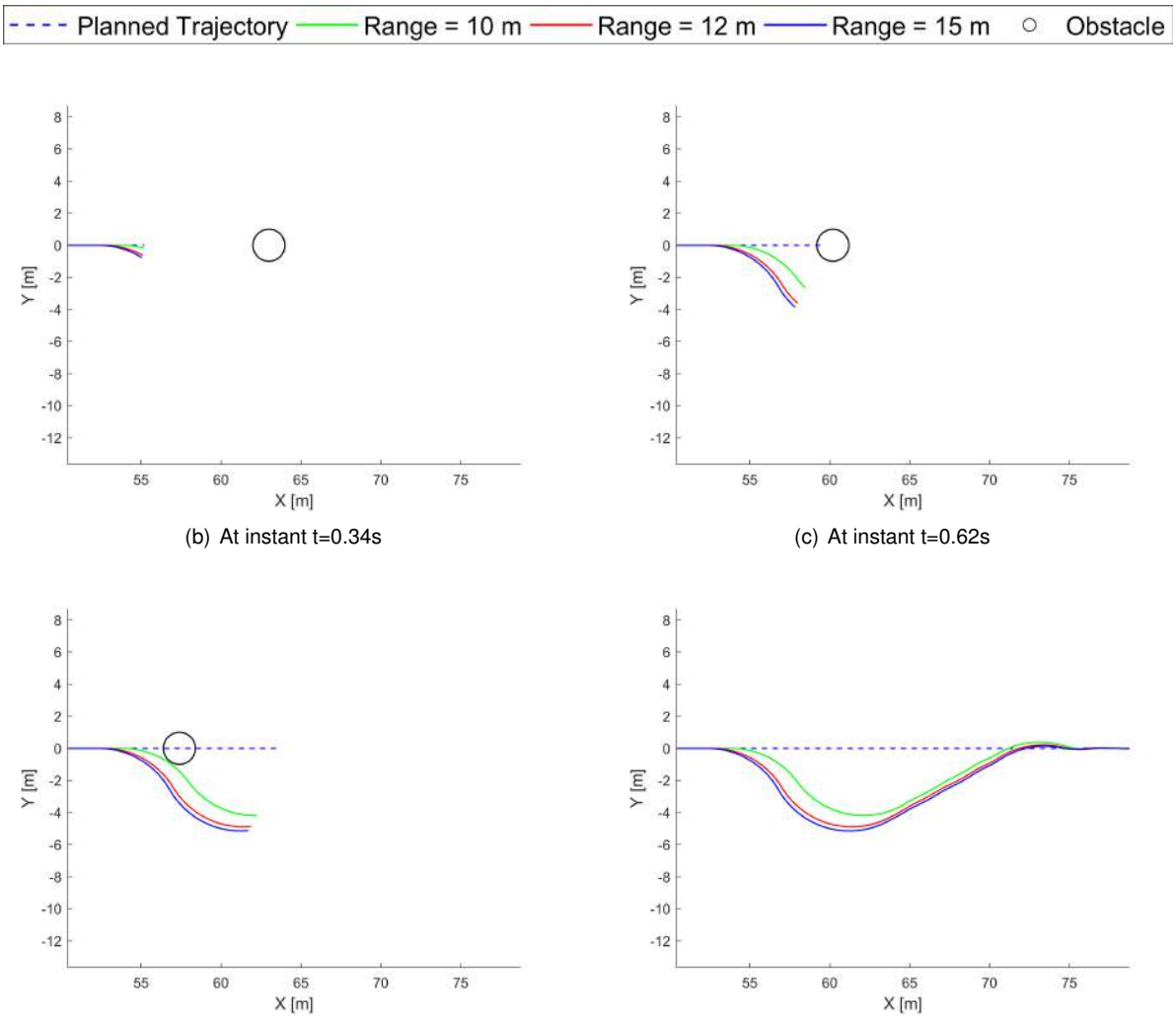
(e) At instant t=2.66s

Figure 7.4: Avoidance trajectories for UAVs with a speed of 8 m/s equipped with a RADAR with different ranges for a head-on collision threat

The UAV behaves very similarly for all illustrated cases. The obstacle is always properly tracked and avoided, which results in similar tight maneuvers. If the UAV is equipped with a RADAR with a range of 10 m, the UAV is already inside the action radius of the obstacle when the obstacle is detected. Because of this, the UAV immediately initiates the avoidance maneuver but cannot avoid breaching the safety radius of the obstacle. For this example, ranges greater than 15 m result in identical trajectories.

The same scenario for a UAV with a velocity of 15 m/s is depicted in Figure 7.5. The UAV's behavior is similar to that of the UAV with the lower speed, although the higher speed results in a higher produced potential, which forces the UAV to perform a less conservative maneuver that prevents the UAV from entering the obstacle's safety radius when it is equipped with a RADAR with a range of 10 m.

From these results, one can conclude that for a UAV moving at a maximum speed of 15 m/s, the range provided by the Aerotenna μSharp Patch is more than enough to prevent head-on collisions, assuming that the obstacles are also moving at a maximum speed of 15 m/s.



(b) At instant t=0.34s

(c) At instant t=0.62s

(d) At instant t=0.9s

(e) At instant t=2.12s

Figure 7.5: Avoidance trajectories for UAVs with a speed of 15 m/s equipped with a RADAR with different ranges for a head-on collision threat

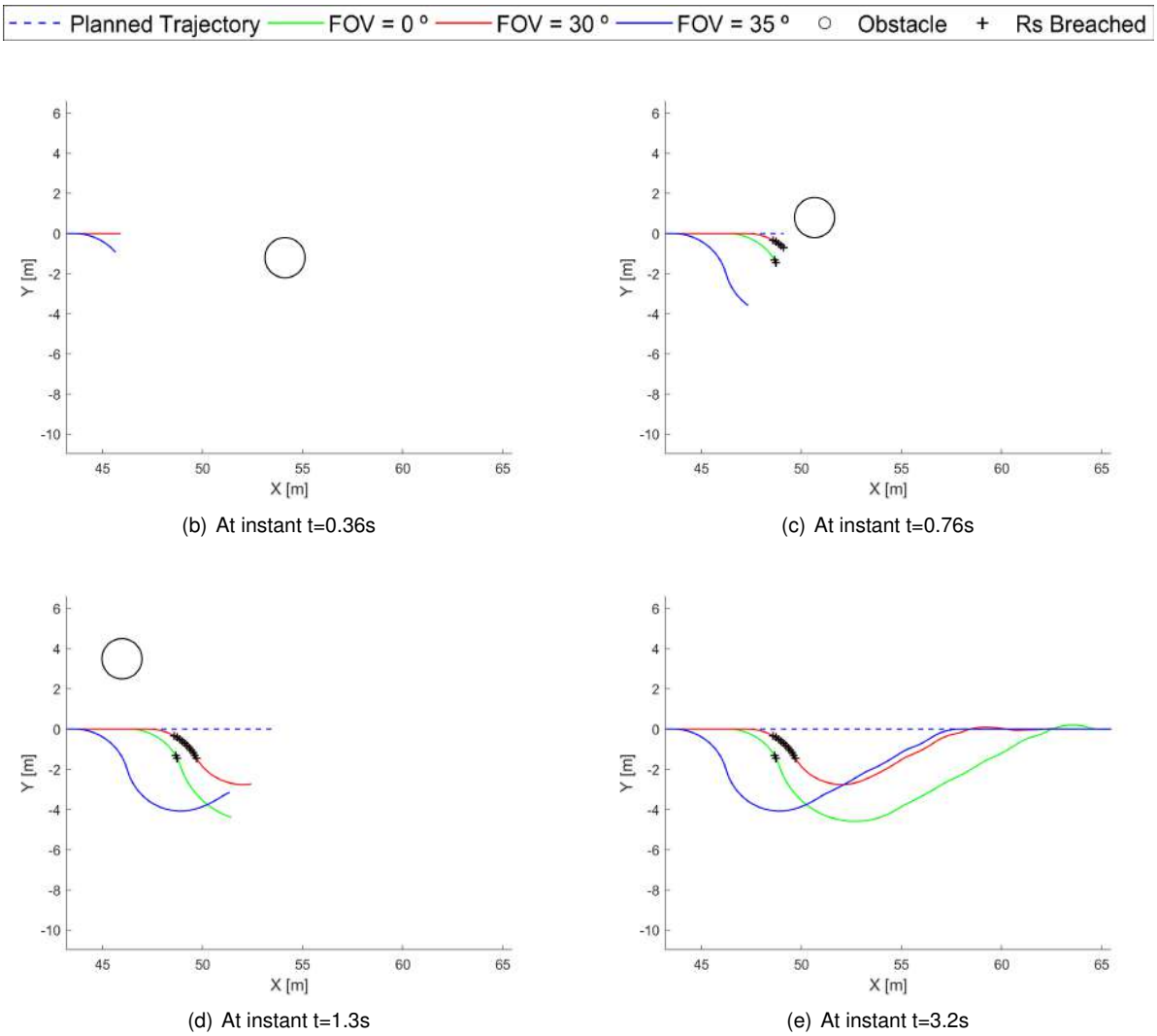
7.5 Varying Field of View Simulations

To test the effect of the FOV of the sensor on the avoidance capabilities of the UAV, the UAV was set in a θ=30° angled collision course (Figure 7.1 (b)) with an obstacle moving at a speed of 10 m/s, while being equipped with a LIDAR that performs a measurement every half degree with a range of 100 m. The responses of the UAV for different FOVs, when it is moving at a speed of 8 m/s, is displayed in Figure 7.6.

All the obstacles were simulated as circles. Therefore, equations (4.12) and (4.13) were adapted to filter the obstacle's radius instead of its length and width. The LIDAR gain from equation (4.14) also needed to be defined. For the sensor to reach 99% of the real dimensions, acc is set to 0.99 and, to get this precision before the obstacle transverses 10% of the 100 m range, the filter needs to perform 20 iterations (n), considering the LIDAR is working at a 50 Hz frequency and assuming the obstacles can move at the same speed of the UAV, which results in a maximum relative speed of 30 m/s. Knowing n and acc, the minimum gain to be used can be computed using said equation, it being 0.2057.

For a FOV of 30° , the obstacle is detected when it is already dangerously close to the UAV, so the UAV breaches the safety radius for several points, despite avoiding a collision. For FOVs greater than 35° , the obstacle is properly tracked in advance, which results in a proper safe maneuver. Finally, for a FOV of 0° , which corresponds to a fixed laser rangefinder, the UAV detects the obstacle only when it is directly in front of it, which causes a quick breach of the safety radius. Because of the singular nature of this case, the obstacle's velocity can only be tracked with a radial component and so, the obstacle is tracked as being in a head-on collision course with a small velocity. This leads to a wide maneuver which results in the UAV leaving the obstacle's safety radius after only a couple points.

The obstacles were avoided by the UAVs with different sensors, but the success of the maneuver depends on the approach angle of the obstacle (θ). For an obstacle closing in at a θ=30° angle, a FOV of 35° is sufficient to properly track the obstacle but that would not be the case if the obstacle would close in at a wider angle. Small FOVs also result in inadequate velocity tracking, which cause the UAV to unreasonably distance itself from the planned trajectory.



(b) At instant t=0.36s

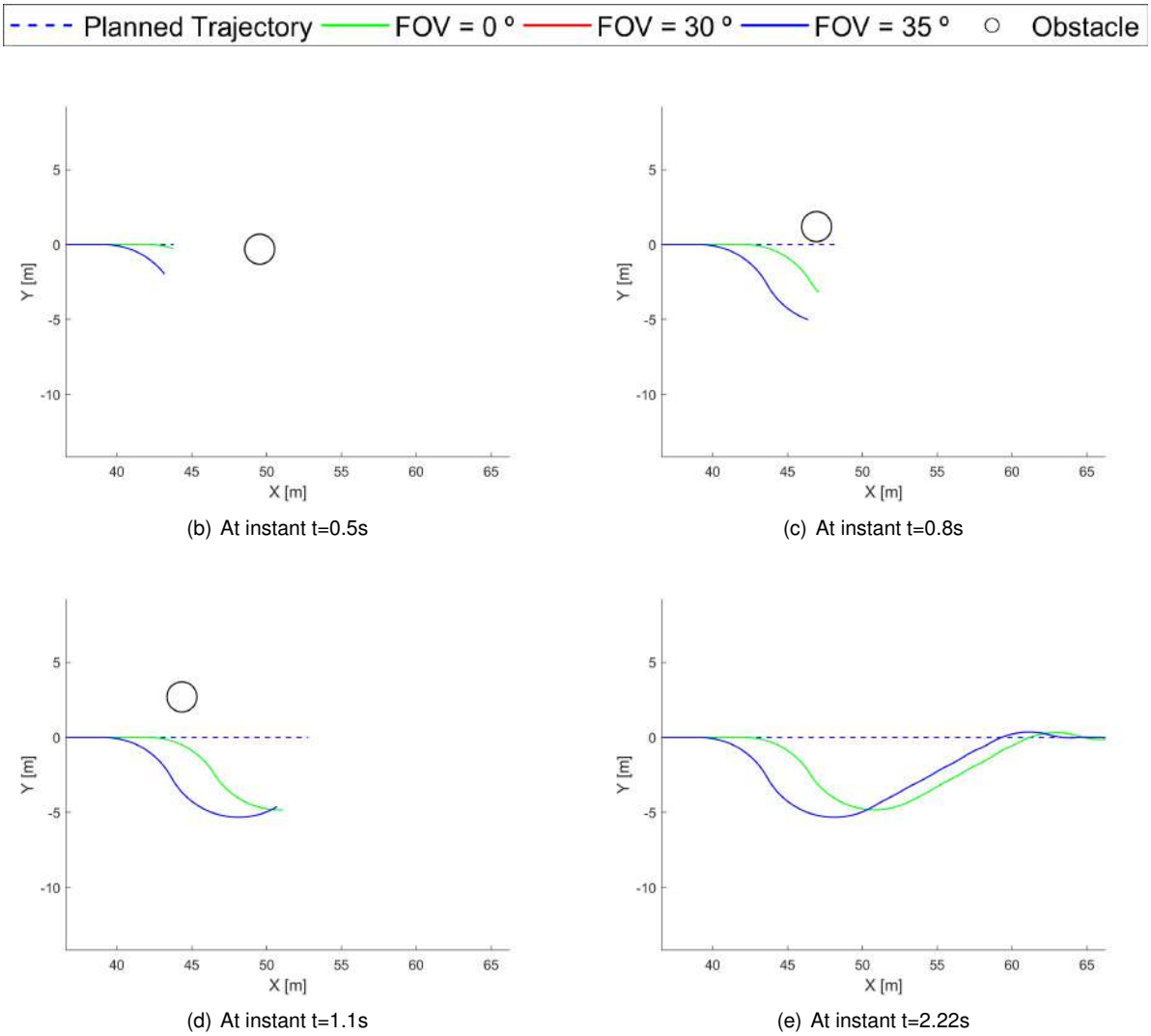
(c) At instant t=0.76s

(d) At instant t=1.3s

(e) At instant t=3.2s

Figure 7.6: Avoidance trajectories for UAVs with a speed of 8 m/s equipped with a LIDAR with different FOVs for an angled collision threat

As seen in Figure 7.7, if the UAV is moving with a speed of 15 m/s, the maneuvers for FOVs of 30° and 35° become identical, as the obstacle is successfully tracked in both cases, which results in a successful avoidance for both cases. The maneuvers also become wider, which also leads to a successful maneuver for a FOV of 0° .



(b) At instant t=0.5s

(c) At instant t=0.8s

(d) At instant t=1.1s

(e) At instant t=2.22s

Figure 7.7: Avoidance trajectories for UAVs with a speed of 15 m/s equipped with a LIDAR with different FOVs for an angled collision threat

Concluding, the success of an avoidance of an obstacle coming from an angled approach is highly dependent on the speed of the UAV, the speed of the obstacle and the angle of approach. However, the wider the FOV, the higher the chance of being able to avoid said obstacle, thus it is better to equip the UAV with a sensor with a wider FOV.

Chapter 8

Optimal Sensing System

In this section, an optimization study is made in order to find the best sensor configuration, for different sensor sets. First, the problem is defined, the algorithm used to perform the optimization is described and its use is justified. Then, the results obtained from the optimizations are displayed and explained. All the sensors used in this chapter have the specifications of the sensors present in the Table 4.2. Therefore the only parameter subject to optimization is their individual orientation.

8.1 Problem Formulation

Firstly, a large number of scenarios was generated, where a random number of obstacles with random velocities are moving when the UAV is traveling through a linear trajectory to a waypoint. From this set of scenarios, fifty that resulted in imminent collisions were selected to perform the subsequent tests. Then, a function $f(\beta)$, to be minimized and dependent on the sensor orientation β, was created. The function is initialized to null and all the generated scenarios are run, where, for each instant, $f(\beta)$ is incremented by

$$\begin{cases} f(\beta) = f(\beta) + 50 & \text{, if } ||\mathbf{d}_0|| \leq R_c \\ f(\beta) = f(\beta) + 1 & \text{, if } R_c < ||\mathbf{d}_0|| \leq R_s \end{cases}, \tag{8.1}$$

where R_c is the collision radius, R_s is the safety radius and $||\mathbf{d}_0||$ is the distance between the UAV and a given obstacle. If a collision occurs, this metric increases intensely, if the safety radius is breached, it increases lightly and it is not incremented otherwise. After running all the scenarios, the cumulative value of $f(\beta)$ can be evaluated: the lower its value, the less failures and close-calls happened.

Gradient-based methods, such as the Broyden-Fletcher-Goldfarb-Shanno (BFGS) were deemed inadequate to minimize this function, since in the chosen function, the variables have null derivatives for long stretches of their domains, which prevented the function from converging to its minima. As a consequence, the Genetic Algorithm (GA), a gradient-free method, was used instead [110].

The Genetic Algorithm is a gradient-free, population-based method, which, instead of working with a single solution candidate, deals with a set of solutions that are updated simultaneously from iteration to iteration, which increases the likelihood of finding the global optimum. As such, this algorithm mimics the

evolution or behavior of a population, where population is understood as the set of solutions. Therefore, population-based methods might be more efficient in exploring the whole search space at the cost of a higher computational load and more complex structures.

In the Genetic Algorithm, the design point associated with an individual is represented as a chromosome. At each generation, the chromosomes of the fitter individuals are passed on to the next generations after undergoing the genetic operations of crossover and mutation. New candidates for the solution are generated with a mechanism called crossover which combines part of the genetic patrimony of each parent and then applies a random mutation. If the new individual, called child, inherits good characteristics from his parents, it will have a higher probability to survive. In the context of genetic algorithms, the value of the objective function is termed the fitness and the variables need to be bounded.

The problem can then be posed in standard form as

$$\begin{aligned} \text{Minimize} \quad & f(\beta) \\ \text{w.r.t.} \quad & \beta\,, \\ \text{subject to} \quad & lb < \beta < ub\,, \end{aligned} \tag{8.2}$$

where $f(\beta)$ is the sensing system metric defined in equation (8.1), β is the orientation of the sensor and lb and ub are the lower and upper bounds of β, respectively, to be defined for each particular case.

Before performing the simulations, several optimization parameters needed to be defined.

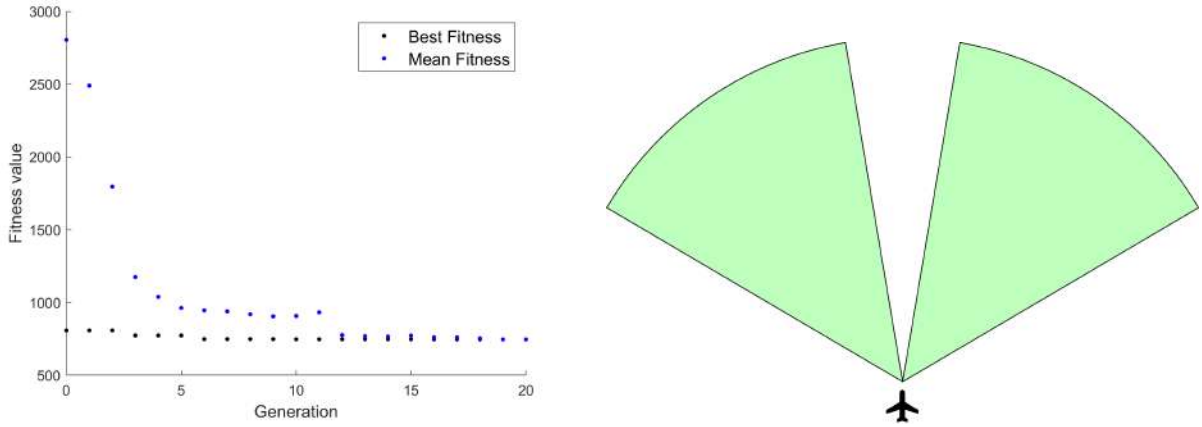
- The initial population was set to be created with a uniform distribution;
- The crossover function was set to create 80% of the population in each generation;
- Because the variables are bounded, the mutation function randomly generates directions that are adaptive with respect to the last successful or unsuccessful generation, where the chosen direction and step length satisfy the set bounds;
- The algorithm was set to stop after 20 generations, enough for one bounded variable to converge until the genetic diversity is very small;
- The population size was set to 30, enough to find the global minima in a timely manner.

These parameters were chosen following the best practices given in reference [111].

8.2 Set of two RADARs

The set of two RADARs was the first to be optimized. The orientation of each sensor was bounded between 0° and 90° from the longitudinal axis, in the horizontal plane and, to simplify the problem in order to reduce the computation time, the two RADARs were considered to have a symmetrical orientation about the UAV's longitudinal axis, which is an acceptable approximation. This way, the algorithm only had one variable to optimize.

As seen in Figure 8.1 (a), the algorithm performed well, as the best fitness value decreases slightly as the generations progress. Eventually, the mean fitness converges to the best fitness and, at the twentieth generation, when the optimization is halted, the best fitness is 746 and the mean fitness is 746.67. The expected result of this simulation was for the sensor orientation to be close to 25° , which would wield the same result as if the UAV was equipped with a RADAR sensor with a doubled FOV. However, the optimal value reached by the algorithm was a value of 34.6° , as seen in Figure 8.1 (b).



(a) GA results (b) Optimal configurtion

Figure 8.1: Results from the optimization of the two RADARs set

A comparison of performance between the optimal orientation, a 25° orientation and a single RADAR pointing forward is presented in Table 8.1, where a failure happens when the UAV collides with an obstacle and a close call happens if the UAV breaches the safety radius of an obstacle.

Table 8.1: Comparison of performance for different orientations for two RADARs

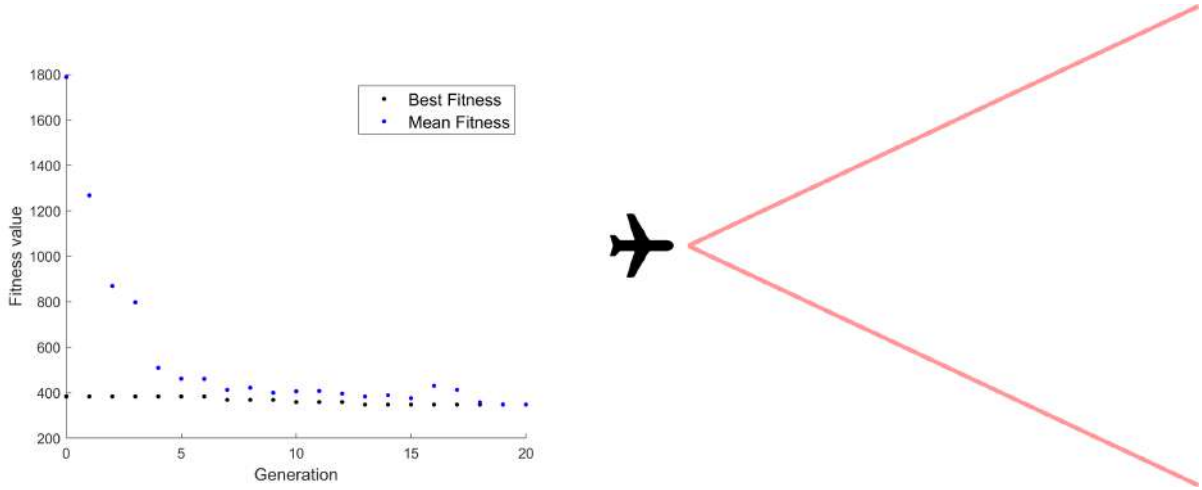
Orientation	Fitness Value	Failure Rate	Close Call Rate
0° (one RADAR)	1625	8%	22%
25° (doubled FOV)	1051	6%	26%
34.6° (optimal solution)	746	6%	20%

All of the failures that occur when the UAV is equipped with one RADAR pointing forward do not happen for the optimal solution because, in those cases, obstacles would approach the UAV from an angle that would not allow their detection by a single RADAR. However, because the optimal solution cannot detect obstacles directly in front of the UAV, two scenarios that were successes for one RADAR become failures but, because the random scenarios include a larger number of obstacles approaching from a wide angle than approaching head-on, the optimal solution corresponds to an angle larger than 25° . Moreover, overlapping FOVs would result in the use of the weighted filter data fusion technique and, consequently, more accurate measurements, but obstacles approaching from wider angles would not be detected.

8.3 Set of two Laser Rangefinders

Like in the previous optimization, in this one, the two sensors were considered to be symmetrical about the UAV's longitudinal axis, so that only one variable bounded between 0° and 90° needed to be optimized.

After 20 generations, both the best fitness and the mean fitness were 348. So, after this generation, only through mutation would a better solution be found. The evolution of the population is presented in Figure 8.2 (a) and the configuration that produced the best fitness (25.1°) is presented in 8.2 (b).



(a) GA results (b) Optimal configuration

Figure 8.2: Results from the optimization of the two laser rangefinders set

As seen in Table 8.2, the optimal configuration results in no collisions in all scenarios.

Table 8.2: Comparison of performance for different orientations for two laser rangefinders

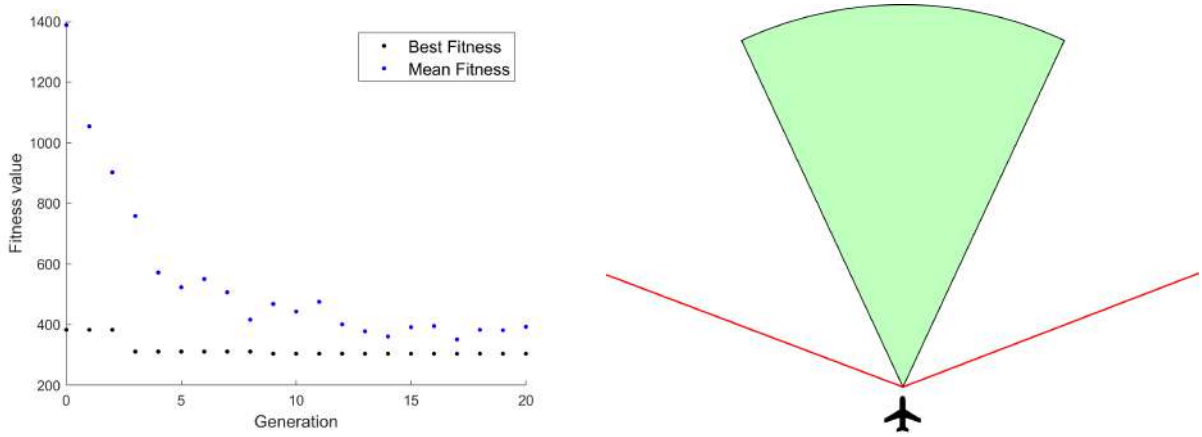
Orientation	Fitness Value	Failure Rate	Close Call Rate
0° (one laser)	1753	12%	40%
25.1° (optimal solution)	348	0%	44%

The optimal solution results in a much lower fitness value than when only one laser rangefinder pointing forward is used, as obstacles coming from an angled approach can be detected before the collision. However, despite the zero collisions, in 44% of the scenarios, the safety radius of obstacles was breached because when collisions with moving objects are imminent, a UAV equipped only with laser rangefinders is incapable of properly tracking the obstacles.

8.4 Set of two Laser Rangefinders and one RADAR

This optimization involved three sensors: one fixed RADAR pointing forward and two laser rangefinders symmetrical about the UAV's longitudinal axis, whose orientation was bonded between 0° and 90° . As seen in Figure 8.3 (a), the mean fitness did not converge to the best fitness as much as in the previous simulations, but the best fitness remained unchanged for the last eleven generations. Therefore,

stopping the optimization at the twentieth generation was still appropriate. When the algorithm halted, the mean fitness was 392.8 and the best fitness was 304 for an orientation of 69.2°. The obtained optimal configuration is represented in Figure 8.3 (b).



(a) GA results (b) Optimal configurtion

Figure 8.3: Results from the optimization of the two laser rangefinders and one RADAR set

In Table 8.3, the performance of the optimal solution is compared to the performance of the solutions that would result from a UAV being equipped with only type of sensor, it being a RADAR pointing forward or two symmetrical laser rangefinders with an orientation of 69.2°.

Table 8.3: Comparison of performance for the optimal solution when using different sensors

Sensors	Fitness Value	Failure Rate	Close Call Rate
One RADAR	1625	8%	22%
Two lasers	1660	8%	62%
One RADAR and two lasers	304	2%	18%

Despite the two lasers configuration and the one RADAR configuration having very similar fitness values, a UAV with only two lasers has almost the triple of close calls than a UAV with one RADAR because the UAV will only be able to detect most of the obstacles when the safety radius is already breached. By contrast, the optimal solution results in only one failure and a reduced number of close calls.

The obtained solution was expected because obstacles approaching in a head-on collision course could be detected by the RADAR, while the ones approaching in an angled collision course could be detected by the heavily displaced laser rangefinders. Once again, the optimal solution did not involve overlapping sensors, which would increase the accuracy of the measurements through the use of the chosen data fusion algorithm, but would not allow the detection of obstacles approaching from a wider angle, which is more favorable.

Chapter 9

Conclusions

9.1 Achievements

This work was developed with the goal of enhancing the safety in the flight of fixed-wing mini UAVs, in regard to the detection of obstacles during flight. The focus was on the first stage of the S&A phase, responsible for the acquisition of the necessary information that allows the vehicle to detect threatening situations like proximity to sensitive infrastructures and route of collision with other manned or unmanned aircraft. The system was developed supposing that the collected data would be fed to an existing avoidance algorithm, responsible for taking the appropriate evasive maneuvers.

Several possible sensors were considered to be used by the developed system and, ultimately, after studying the operation of each type of sensor and making a market study on available sensors to know which sensors had the most adequate attributes (range, FOV, accuracy, cost), ultrasound and stereo vision sensors were rejected due to the small range they provided, TCAS sensors were rejected because of the high payload and power they required and ADS-B sensors were rejected for only detecting other aircraft when they were also equipped with this cooperative sensor, which would not fit the scope of this work, despite existing small models that could be implemented in mini UAVs. Of the studied available models, the laser rangefinder/LIDAR Lightware LW20/C and the RADAR Aerotenna μSharp Patch were the ones that would best meet the needs of the system.

This work was part of a comprehensive obstacle detection and collision avoidance system, so a brief study on existing collision detection and avoidance algorithms was made and an in-depth description of the potential field approach, the chosen algorithm, was also made. Knowing the sensors and the avoidance algorithm to be used, a S&A architecture was proposed and developed that would guarantee that only obstacles that represent impeding collisions generate avoidance maneuvers and that the vehicle returns to the predefined path or to a missed global WP once these obstacles were cleared.

To integrate the adequate sensors into the avoidance system already developed, they needed to be modeled. These models took into consideration the range, FOV and accuracy that the real sensors were characterized by. To allow the tracking of the detected obstacles, a study on different Kalman filters took place. For laser rangefinders and LIDARs, classic Kalman filters were sufficient to guarantee

adequate tracking. However, for the RADAR sensors, a Converted Measurement Kalman Filter with unbiased conversion was required, due to the conversion of the measurements from polar to Cartesian coordinates. Additionally, to fuse the data obtained from different redundant sensors at a decision level, two alternatives were presented and the weighted filter technique was selected due to its simplicity and effectiveness.

Having modeled the sensors, several parametric studies were made, where the impact of the range, field of view, and speed of the vehicle in the avoidance of obstacles from predetermined scenarios was made clear. From these simulations, the specifications of the studied sensors were verified as more than acceptable for avoiding obstacles at the considered speed range. Additionally, an optimization study was conducted to determine the best orientation that the sensors should have when the UAV is equipped with different sets of sensors. The optimization process was accomplished by using the genetic algorithm to minimize a function closely associated with the success and failure of the avoidance maneuvers. Because the sensors in the tested sets were redundant, none of the optimal solutions involved using the selected sensor fusion method, as having the sensors pointing at different directions is more valuable than having them pointing at the same obstacle and fusing the results.

Overall, the developed system provided a satisfactory solution to the obstacle detection problem of mini UAVs in a simulated environment.

9.2 Deliverables

The defined goals and deliverables for this thesis were mostly achieved: The software to test the performance of a sensor was developed, which can be used to tune the sensors' parameters in order to find the configuration that results in the the best avoidance of obstacles for any given scenario. The software to obtain the optimal configuration of a set of sensors was also developed, taking into account the aircraft characteristics and mission constraints.

9.3 Future Work

There are several improvements and future work that can be made. The simulations can be adapted to a space that better integrates the vehicle dynamics, so that the used models better reflect the reality. This new simulated environment should also be able to generate more complex obstacles that better reflect the diversity found in the real world. A study can also be made regarding the fusion of range sensors with image sensors, since this work only dealt with the fusion of redundant sensors.

Ground and flight tests should also be made to test how a vehicle would behave when using the chosen avoidance algorithm with different sensor configurations in the real world.

Finally, this work was developed with the objective of increasing the UAV's autonomy, but interaction with the operator should also be considered, so that more suitable decision systems can be developed.

Bibliography

[1] J. Keane and S. Carr. A brief history of early unmanned aircraft. *Johns Hopkins Apl Technical Digest*, 32:558–571, Dec. 2013.

[2] I. Mademlis, V. Mygdalis, N. Nikolaidis, M. Montagnuolo, F. Negro, A. Messina, and I. Pitas. High-level multiple-UAV cinematography tools for covering outdoor events. *IEEE Transactions on Broadcasting*, 65(3):627–635, 2019. doi: 10.1109/TBC.2019.2892585.

[3] P. Tokekar, J. V. Hook, D. Mulla, and V. Isler. Sensor planning for a symbiotic UAV and UGV system for precision agriculture. *IEEE Transactions on Robotics*, 32(6):1498–1511, Dec. 2016. doi: 10.1109/TRO.2016.2603528.

[4] S. Berrahal, J.-H. Kim, S. Rekhis, N. Boudriga, D. Wilkins, and J. Acevedo. Border surveillance monitoring using quadcopter UAV-aided wireless sensor networks. *Journal of Communications Software and Systems*, 12(1):67–82, Mar. 2016. doi: 10.24138/jcomss.v12i1.92.

[5] B. D. Song, K. Park, and J. Kim. Persistent UAV delivery logistics: MILP formulation and efficient heuristic. *Computers & Industrial Engineering*, 120(4):418–428, June 2018. doi: 10.1016/J.CIE.2018.05.013.

[6] M. Zhang. CNN gets permission from FAA to experiment with camera drones in news gathering. `https://petapixel.com/2015/01/13/cnn-gets-permission-faa-test-using-drones-news-imagery/`, 2015. Accessed: 20-12-2020.

[7] Delair. Delair unveils UX11 Ag, new UAV optimized for large scale surveying in agriculture and forestry. `https://petapixel.com/2015/01/13/cnn-gets-permission-faa-test-using-drones-news-imagery/`, 2018. Accessed: 20-12-2020.

[8] CBS News. U.S. drones now patrol half of the mexican border. `https://www.cbsnews.com/news/u-s-drones-patrol-nearly-half-of-mexico-border/`, 2014. Accessed: 12-04-2020.

[9] Drone delivery operations underway in 27 countries. `https://www.unmannedairspace.info/latest-news-and-information/drone-delivery-operations-underway-in-26-countries/`, Apr. 2019. Accessed: 24-03-2020.

[10] P. Finnegan. *2019 World Civil Unmanned Aerial Systems Market Profile & Forecast*. Teal Group Corporation, Aug. 2019.

[11] K. Dalamagkidis. Classification of UAVs. In K. P. Valavanis and G. J. Vachtsevanos, editors, *Handbook of Unmanned Aerial Vehicles*, pages 83–91. Springer Netherlands, 2015. doi: 10.1007/978-90-481-9707-1_94.

[12] U.S. Dept. of Transportation, Federal Aviation Administration, Office of Aviation Policy and Plans. Uas by the numbers. https://www.faa.gov/uas/resources/by_the_numbers/, Mar. 2020. Accessed: 25-03-2020.

[13] O. o. A. P. U.S. Dept. of Transportation, Federal Aviation Administration and Plans. FAA aerospace forecasts 2019-2039. https://www.faa.gov/data_research/aviation/aerospace_forecasts/media/FY2019-39_FAA_Aerospace_Forecast.pdf, July 2019. Accessed: 25-03-2020.

[14] D. Sabbagh. Killer drones: how many are there and who do they target? *The Guardian*, Nov. 2019. https://www.theguardian.com/news/2019/nov/18/killer-drones-how-many-uav-predator-reaper Accessed: 25-03-2020.

[15] R. Austin. *Unmanned Aircraft Systems: UAVS Design, Development and Deployment*. Wiley, first edition, Sept. 2011.

[16] NASA Jet Propulsion Laboratory-Caltech. Rocky 7. https://www.nasa.gov/topics/technology/features/Rocky-7.html, 2011. Accessed: 10-05-2020.

[17] R. Volpe. Mars rover navigation results using sun sensor heading determination. In *Proceedings 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human and Environment Friendly Robots with High Intelligence and Emotional Quotients (Cat. No.99CH36289)*, volume 1, pages 460–467, Oct. 1999. doi: 10.1109/IROS.1999.813047.

[18] Stanford Racing Team. High resolution desert pictures. https://cs.stanford.edu/group/roadrunner/old/presskit.html, 2005. Accessed: 10-05-2020.

[19] DARPA. DARPA grand challenge 2005 rules, Aug. 2004.

[20] Stanford Racing Team. Stanford racing team's entry in the 2005 DARPA grand challenge. Technical report, Stanford University, 2005. https://cs.stanford.edu/group/roadrunner/ Accessed: 30-04-2020.

[21] Regulamento 1093/2016. *Diário da República, II série*, 238, 10 Dec. 2016.

[22] Decreto lei 58/2018. *Diário da República, I série*, 140, 23 July 2016.

[23] Regulation (EU) 2019/947 on the rules and procedures for the operation of unmanned aircraft. *Official Journal of the European Union*, L 152/45, 11 June 2019.

[24] TEKEVER AR4. http://uas.tekever.com/ar4-evo//. Accessed: 24-11-2020.

[25] X. Yu and Y. Zhang. Sense and avoid technologies with applications to unmanned aircraft systems: Review and prospects. *Progress in Aerospace Sciences*, 74:152 – 166, 2015. doi: https://doi.org/10.1016/j.paerosci.2015.01.001.

[26] S. Lin, X. Kong, and L. Liu. Development of an intelligent UAV path planning approach to minimize the costs in flight distance, time, altitude, and obstacle collision. In *2019 19th International Symposium on Communications and Information Technologies (ISCIT)*, pages 238–243, Sept. 2019. doi: 10.1109/ISCIT.2019.8905119.

[27] J. Joseph, M. Radmanesh, M. N. Sadat, R. Dai, and M. Kumar. UAV path planning for data ferrying with communication constraints. *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC), Consumer Communications & Networking Conference (CCNC), 2020 IEEE 17th Annual*, pages 1 – 9, Jan. 2020. doi: 10.1109/CCNC46108.2020.9045416.

[28] P. Krishnan and K. Manimala. Implementation of optimized dynamic trajectory modification algorithm to avoid obstacles for secure navigation of UAV. *Applied Soft Computing Journal*, Jan. 2019. doi: 10.1016/j.asoc.2020.106168.

[29] J. Kuchar and A. C. Drumm. The traffic alert and collision avoidance system. *Lincoln Laboratory Journal*, 16(2):277, 2007.

[30] M. Schäfer, X. Olive, M. Strohmeier, M. Smith, I. Martinovic, and V. Lenders. Opensky report 2019: Analysing tcas in the real world using big data. 09 2019.

[31] International Civil Aviation Organization. *Airborne Collision Avoidance System (ACAS) Manual*, 1 edition, 2006. Doc 9863 AN/461. Accessed: 16-05-2020.

[32] *T TR-2100/T TR-4100 airborne collision avoidance system/traffic computer*. Collins Aerospace, 2019.

[33] J. K. Kuchar. Safety analysis methodology for unmanned aerial vehicle UAV collision avoidance systems. In *USA/Europe Air Traffic Management R&D Seminars*, volume 12, June 2005.

[34] H.-C. Lee. Implementation of collision avoidance system using TCAS II to UAVs. In *24th Digital Avionics Systems Conference*, volume 2, Oct. 2005. doi: 10.1109/DASC.2005.1563410.

[35] Commission implementing regulation (EU) laying down requirements for the performance and the interoperability of surveillance for the single European sky. *Official Journal of the European Union*, L 305/35(1207/2011), 2011.

[36] M. Strohmeier, V. Lenders, and I. Martinovic. On the security of the automatic dependent surveillance-broadcast protocol. *IEEE Communications Surveys Tutorials*, 17(2):1066–1087, 2015. doi: 10.1109/COMST.2014.2365951.

[37] What is ADS-B, which requirements? `https://shop.jetvision.de/Blog/What-is-ADS-B-What-Requirements`, . Accessed: 07-04-2020.

[38] C. Lin, L. Ya-Hsien, and L. Fang-Ju. UAV collision avoidance using sector recognition in cooperative mission to helicopters. *2014 Integrated Communications, Navigation and Surveillance Conference (ICNS) Conference Proceedings, Integrated Communications, Navigation and Surveillance Conference (ICNS), 2014*, pages 1–9, Apr. 2014. doi: 10.1109/ICNSurv.2014.6819986.

[39] C. Lai, Y. Ren, and C. Lin. ADS-B based collision avoidance radar for unmanned aerial vehicles. In *2009 IEEE MTT-S International Microwave Symposium Digest*, pages 85–88, June 2009. doi: 10.1109/MWSYM.2009.5165638.

[40] A. Stulgis, L. Ambroziak, and M. Kondratiuk. Obstacle detection and avoidance system for unmanned multirotors. *23rd International Conference on Methods & Models in Automation & Robotics (MMAR)*, pages 455 – 460, Aug. 2018. doi: 10.1109/MMAR.2018.8485911.

[41] RCraig09. Time-of-flight principles applied to laser range-finding. `https://commons.wikimedia.org/w/index.php?curid=89646954`, 2020. Accessed: 07-05-2020.

[42] J. Saunders, B. Call, A. Curtis, R. Beard, and T. McLain. Static and dynamic obstacle avoidance in miniature air vehicles. Sept. 2005. doi: 10.2514/6.2005-6950.

[43] C. Kownacki. Obstacle avoidance strategy for micro aerial vehicle. In *Advances in Aerospace Guidance, Navigation and Control*, pages 117–135. Springer, Berlin and Heidelberg, 2011. doi: 10.1007/978-3-642-19817-5_10.

[44] R. LaDue. Drones and lidar a match made in cartometric modeling heaven. `https://geoawesomeness.com/drones-and-lidar-a-match-made-in-cartometric-modeling-heaven/`, 2019. Accessed: 14-04-2020.

[45] L. Zheng, P. Zhang, J. Tan, and F. Li. The obstacle detection method of UAV based on 2D lidar. *IEEE Access*, 7:163437–163448, Nov. 2019. doi: 10.1109/ACCESS.2019.2952173.

[46] F. Fayad and V. Cherfaoui. Tracking objects using a laser scanner in driving situation based on modeling target shape. In *2007 IEEE Intelligent Vehicles Symposium*, pages 44–49, June 2007. doi: 10.1109/IVS.2007.4290089.

[47] L. Wallace, A. Lucier, C. Watson, and D. Turner. Development of a UAV-LiDAR system with application to forest inventorys. *Remote Sensing*, 4(6):1519–1543, May 2012. doi: 10.3390/rs4061519.

[48] D. Roca, J. Armesto, S. Lagüela, and L. Díaz-Vilariño. Lidar-equipped UAV for building information modelling. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 45:523–527, May 2014. doi: 10.5194/isprsarchives-XL-5-523-2014.

[49] P. Roberta, A. Ilenia, M. Paola, S. Giuseppe, M. Cosimo, E. Ruggero, and A. Ciro. UAV and airborne LiDAR data for interpreting kinematic evolution of landslide movements: The case study of the Montescaglioso landslide (southern Italy). *Geosciences*, 9(6):248, 2019.

[50] M. S. Ameigh. Radar. *Salem Press Encyclopedia of Science*, 2019.

[51] S. Prager and M. Moghaddam. Application of ultra-wideband synthesis in software defined radar for UAV-based landmine detection. In *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, pages 10115–10118, July 2019. doi: 10.1109/IGARSS.2019.8899149.

[52] Y. K. Kwag and C. H. Chung. UAV based collision avoidance radar sensor. In *2007 IEEE International Geoscience and Remote Sensing Symposium*, pages 639–642, July 2007. doi: 10.1109/IGARSS.2007.4422877.

[53] B. Ajith Kumar and D. Ghose. Radar-assisted collision avoidance/guidance strategy for planar flight. *IEEE Transactions on Aerospace and Electronic Systems*, 37(1):77–90, Jan. 2001. doi: 10.1109/7.913669.

[54] S. Hrabar. 3D path planning and stereo-based obstacle avoidance for rotorcraft UAVs. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 807–814, 2008. doi: 10.1109/IROS.2008.4650775.

[55] P. Rudol and P. Doherty. Human body detection and geolocalization for UAV search and rescue missions using color and thermal imagery. In *2008 IEEE Aerospace Conference*, pages 1–8, Mar. 2008. doi: 10.1109/AERO.2008.4526559.

[56] L. Wang, F. Chen, and H. Yin. Detecting and tracking vehicles in traffic by unmanned aerial vehicles. *Automation in Construction*, 72(3):294 – 308, 2016. doi: /10.1016/j.autcon.2016.05.008.

[57] DataFromSky. Vehicle speed tracking on drone traffic video from cayman islands. `https://www.youtube.com/watch?v=xwQky6aM_lo`, 2016. Accessed: 21-04-2020.

[58] M. Dubbini, S. Candiago, F. Remondino, M. De Giglio, and M. Gattelli. Evaluating multispectral images and vegetation indices for precision farming applications from UAV images. *Remote Sensing*, 7(4):4026–4047, Apr. 2015. doi: 10.13140/RG.2.1.3430.3527.

[59] H. Shakhatreh, A. H. Sawalmeh, A. Al-Fuqaha, Z. Dou, E. Almaita, I. Khalil, N. S. Othman, A. Khreishah, and M. Guizani. Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges. *IEEE Access*, 7:48572–48634, Apr. 2019. doi: 10.1109/ACCESS.2019.2909530.

[60] L. R. G. Carrillo, A. E. D. López, R. Lozano, and C. Pégard. Combining stereo vision and inertial navigation system for a quad-rotor UAV. *Journal of intelligent & robotic systems*, 65(1-4):373–387, Aug. 2012. doi: 10.1007/s10846-011-9571-7.

[61] W. Mormul. Stereo vision - depth map. `https://www.youtube.com/watch?v=bsA6RKUUA3M`, 2016. Accessed: 06-05-2020.

[62] T. Gandhi, M.-T. Yang, R. Kasturi, O. Camps, L. Coraor, and J. McCandless. Detection of obstacles in the flight path of an aircraft. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2304–2311, Jan. 2000. doi: 10.1109/CVPR.2000.854819.

[63] Bats use echolocation to find prey. https://dickinsoncountyconservationboard.com/2018/10/15/bats-use-echolocation-to-find-prey/, . Accessed: 07-04-2020.

[64] N. Gageik, P. Benz, and S. Montenegro. Obstacle detection and collision avoidance for a UAV with complementary low-cost sensors. *IEEE Access*, 3:599–609, May 2015. doi: 10.1109/ACCESS.2015.2432455.

[65] M. Schirrmann, A. Hamdorf, A. Giebel, F. Gleiniger, M. Pflanz, and K.-H. Dammer. Regression kriging for improving crop height models fusing ultra-sonic sensing with UAV imagery. *Remote Sensing*, 9(7):665–683, July 2017. doi: 10.3390/rs9070665.

[66] U. Papa and G. Del Core. Design of sonar sensor model for safe landing of an UAV. In *2015 IEEE Metrology for Aerospace (MetroAeroSpace)*, pages 346–350, June 2015. doi: 10.1109/MetroAeroSpace.2015.7180680.

[67] J. A. Paredes, F. J. Álvarez, T. Aguilera, and J. M. Villadangos. 3D indoor positioning of UAVs with spread spectrum ultrasound and time-of-flight cameras. *Sensors*, 18(1):89, 2018. doi: 10.3390/s18010089.

[68] *PING-RX ADS-B Dual Receiver*. uAvionix, 2019.

[69] A. Koubaa, A. Allouch, M. Alajlan, Y. Javed, A. Belghith, and M. Khalgui. Micro air vehicle link (mavlink) in a nutshell: A survey. *IEEE Access*, 7:87658–87680, 2019.

[70] *PING-1090i ADS-B Transceiver*. uAvionix, 2019.

[71] *TR-1W*. Aerobits, 2018.

[72] *LW20 / SF20 LiDAR sensor Product manual*. Lightware, 2018. Rev 9.

[73] *SF30 Accelerated laser rangefinder Product manual*. Lightware, 2018. Rev 9.

[74] *SF30/D LiDAR sensor Datasheet*. Lightware, 2018. Rev 0.

[75] *μSharp Patch Collision Avoidance Radar*. Aerotenna, 2018.

[76] *μSharp 360° Sense-and-Avoid Radar*. Aerotenna, 2018.

[77] *Drone products lineup*. Ainstein, 2018.

[78] D. Litwiller. CCD vs. CMOS. *Photonics spectra*, 35(1):154–158, 2001.

[79] J. Leconte. Areascan cameras: How to choose between global and rolling shutter. *ATMEL J*, pages 37–39, 2006.

[80] J. Mullins. Blades. https://www.flickr.com/photos/jasonmullins/4919797867/, 2010. Accessed: 19-06-2020.

[81] *Lt Camera Series Compact, Lightweight USB3 Cameras*. Lumenera Industrial Cameras, 2020.

[82] *Intel RealSenseTM D400 Series (DS5) Product Family*. Intel, 2018. Rev 001.

[83] *MB1242 Datasheet*. MaxBotix, 2012.

[84] M. S. Grewal, A. P. Andrews, and C. G. Bartone. Kalman filtering. pages 355–417. Wiley Telecom, 2020.

[85] MathWorks. Use Kalman filter for object tracking. `https://www.mathworks.com/help/vision/ug/using-kalman-filter-for-object-tracking.html`, 2013. Accessed: 19-09-2020.

[86] H. F. Durrant-Whyte. Sensor models and multisensor integration. In *Autonomous robot vehicles*, pages 73–89. Springer, New York, NY, 1990. doi: 10.1007/978-1-4613-8997-2_7.

[87] Y. Watanabe, A. Calise, and E. Johnson. Vision-based obstacle avoidance for UAVs. *AIAA Guidance, Navigation and Control Conference and Exhibit*, June 2012. doi: 10.2514/6.2007-6829.

[88] Y.-Y. Chuang. Camera calibration. In *Digital Visual Effects*. Citeseer, 2005.

[89] A. J. Calise, E. N. Johnson, R. Sattigeri, Y. Watanabe, and V. Madyastha. Estimation and guidance strategies for vision based target tracking. In *Proceedings of the 2005, American Control Conference*, volume 7, pages 5079–5084, June 2005. doi: 10.1109/ACC.2005.1470821.

[90] D. Lerro and Y. Bar-Shalom. Tracking with debiased consistent converted measurements versus EKF. *IEEE Transactions on Aerospace and Electronic Systems*, 29(3):1015–1022, July 1993. doi: 10.1109/7.220948.

[91] W. Mei and Y. Bar-Shalom. Unbiased kalman filter using converted measurements: Revisit. In *Signal and Data Processing of Small Targets 2009*, volume 7445, page 74450U. International Society for Optics and Photonics, 2009.

[92] M. Longbin, S. Xiaoquan, Z. Yiyu, S. Z. Kang, and Y. Bar-Shalom. Unbiased converted measurements for tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 34(3):1023–1027, 1998. doi: 10.1109/7.705921.

[93] B. Zhang, W. Liu, Z. Mao, J. Liu, and L. Shen. Cooperative and geometric learning algorithm (CGLA) for path planning of uavs with limited information. *Automatica*, 50(3):809 – 820, 2014. doi: 10.1016/j.automatica.2013.12.035.

[94] P. Yao, H. Wang, and Z. Su. Cooperative path planning with applications to target tracking and obstacle avoidance for multi-UAVs. *Aerospace Science and Technology*, 54:10 – 22, 2016. doi: 10.1016/j.ast.2016.04.002.

[95] S. Wu, S. Decker, P. Chang, T. Camus, and J. Eledath. Collision sensing by stereo vision and radar sensor fusion. *IEEE Transactions on Intelligent Transportation Systems*, 10(4):606–614, Oct. 2009. doi: 10.1109/TITS.2009.2032769.

[96] D. L. Hall and J. Llinas. An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85 (1):6–23, Jan. 1997. doi: 10.1109/5.554205.

[97] M. Abdelrahman, P. Kandasamy, and J. Frolik. A methodology for the fusion of redundant sensors. In *Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No.00CH36334)*, volume 4, pages 2922–2926, 2000. doi: 10.1109/ACC.2000.878745.

[98] J. Alves. Path planning and collision avoidance algorithms for small RPAS. Master's thesis, Instituto Superior Técnico, June 2017.

[99] H. Pham, S. A. Smolka, S. D. Stoller, D. Phan, and J. Yang. A survey on unmanned aerial vehicle collision avoidance systems. *arXiv preprint arXiv:1508.07723*, 2015.

[100] A. Mcfadyen, A. Durand-Petiteville, and L. Mejias. Decision strategies for automated visual collision avoidance. In *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2014. doi: 10.1109/ICUAS.2014.6842316.

[101] B. Albaker and N. Rahim. Unmanned aircraft collision detection and resolution: Concept and survey. In *2010 5th IEEE Conference on Industrial Electronics and Applications*, pages 248–253. IEEE, 2010.

[102] K.-Y. Kim, J.-W. Park, and M.-J. Tahk. UAV collision avoidance using probabilistic method in 3-D. In *2007 International Conference on Control, Automation and Systems*, pages 826–829, Oct. 2007. doi: 10.1109/ICCAS.2007.4407015.

[103] A. Alexopoulos, A. Kandil, P. Orzechowski, and E. Badreddin. A comparative study of collision avoidance techniques for unmanned aerial vehicles. In *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pages 1969–1974, Oct. 2013. doi: 10.1109/SMC.2013.338.

[104] J. Park, H. Oh, and M. Tahk. UAV collision avoidance based on geometric approach. In *2008 SICE Annual Conference*, pages 2122–2126, Aug. 2008. doi: 10.1109/SICE.2008.4655013.

[105] Jen-Hui Chuang and N. Ahuja. An analytically tractable potential field model of free space and its application in obstacle avoidance. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 28(5):729–736, Oct. 1998. doi: 10.1109/3477.718522.

[106] H. Safadi. Local path planning using virtual potential field. `https://www.cs.mcgill.ca/~hsafad/robotics/`, 2007. Accessed: 13-06-2020.

[107] International Civil Aviation Organization. Rules of the air, annex 2 to the convention on international civil aviation, July 2005.

[108] D. Ruivo. Formation of unmanned vehicles with collision avoidance capabilities. Master's thesis, Instituto Superior Técnico, Nov. 2015.

[109] B. Etkin and L. D. Reid. *Dynamics of flight*, volume 2. 3^{rd} edition, 1959.

[110] M. J. Kochenderfer and T. A. Wheeler. *Algorithms for optimization*. Mit Press, 2019. ISBN:9780262039420.

[111] Mathworks ®. Global optimization toolbox ™: User's guide (r2020b). https://www.mathworks.com/help/pdf_doc/gads/gads.pdf, 2020. Accessed: 29-12-2020.