Delft University of Technology
Master of Science Thesis in Embedded Systems

Simultaneous drone localization and communication using visible light

Jasper Johannes Maria Lut

Simultaneous drone localization and communication using visible light

Master of Science Thesis in Embedded Systems

Networked Systems group
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
Van Mourik Broekmanweg 6, 2628 XE Delft, The Netherlands

Jasper Johannes Maria Lut

28th June 2023

Author
Jasper Johannes Maria Lut

Title
Simultaneous drone localization and communication using visible light
MSc Presentation Date
July 7th 2023

Graduation Committee

Dr. M.A, Zúñiga Zamalloa	Delft University of Technology
Dr. Qing Wang	Delft University of Technology
Ir. Talia Xu	Delft University of Technology

Abstract

Drones that perform complex autonomous movements require a perfect estimate of their current position. However, internal measurement unit (IMU) errors introduce drift in this estimate, leading to significant discrepancies between the predicted and actual location. Various solutions have been proposed to calibrate the IMU, including methods involving cameras and humans in the loop. This thesis suggests implementing a previously developed technique that involves projecting a precise static light polarisation grid into a room. Although this pattern is invisible to the human eye it can be observed using a polariser and colour sensor combination. A drone equipped with such a sensor setup can recalibrate for IMU drift by utilising the perceived polarisation patterns as optical landmarks.

The system design is further developed by exploring the potential of visible light communication (VLC) as an alternative to traditional radio frequency (RF) links for drone control. By leveraging the existing infrastructure used for the projection of the polarisation grid, a VLC link is integrated into the system. With the addition this work strives to fuse polarisation-based localisation and VLC, setting the first steps in creating a fully visible light-based drone platform.

To validate the system, a prototype is created that achieves real-time simultaneous localisation and communication on an embedded drone. This is accomplished through machine learning based classification, a drone motion model, an optimised polarisation pattern enabling fast localisation and a noise-resistant VLC link. Experiments show a median 2D tracking error of 10cm using only light-based methods and a VLC link range of up to 2.5 meters under various conditions.

Preface

This thesis shows the culmination of my academic journey and represents the hard work and passion I have poured into my research over the past year. During this time, I have encountered countless challenges and setbacks, followed by moments of joy and understanding. These experiences solidified my knowledge of the subject matter and allowed me to ultimately present this thesis to you.

This thesis was only possible with the help of some incredible people. I want to express my gratitude towards Dr. M.A. Zúñiga Zamalloa for his unwavering support and expertise on the subject. Ir. Talia Xu for her excellent advice and fast support whenever needed. Additionally, I would like to thank Dr. Qing Wang for his willingness to be part of my committee.

Finally, I would like to thank my friends and family for all their support during my studies, throughout the good times, as well as during the more difficult times. I specifically want to thank Dipal for her unwavering support and patience during the most demanding parts of my thesis.

Jasper Johannes Maria Lut

Delft, The Netherlands
28th June 2023

Glossary

- **2D** Two Dimensional
- **3D** Three Dimensional
- **ACC** Accelerometer
- **ADC** Analogue to Digital Converter
- **BOM** Bill Of Materials
- **CAD** Computer Aided Design
- **DMD** Digital Micro-mirror Display
- **DT** Decision Tree
- **EM** Electro Magnetic
- **FFT** Fast Fourier Transform
- **FSK** Frequency Shift Keying
- **GT** Ground Truth
- **HSV** Hue Saturation Value
- **I2C** Inter-Integrated Circuit
- **IMU** Internal Measurement Unit
- **IR** Infra Red
- **K-NN** K-Nearest Neighbours
- **LIDAR** Light Detection and Ranging
- **LC** Liquid Crystal
- **LCD** Liquid Crystal Display
- **LED** Light Emitting Diode
- **ML** Machine Learning
- **MLP** Multi-Layer Perceptrons
- **MM** Motion Model
- **OOK** On-Off Keying
- **PF** Particle Filter
- **RF** Radio Frequency
- **RFID** Radio Frequency Identification
- **RGB** Red Green Blue
- **RTOS** Real Time Operating System
- **SVC** Support Vector Classifiers
- **UAV** Unmanned Aerial Vehicle
- **VLC** Visible Light Communication

Contents

1 Introduction 1
- 1.1 Research question . . . 3
- 1.2 Contributions . . . 4
- 1.3 Thesis outline . . . 4

2 Background 5
- 2.1 Existing localisation techniques . . . 5
  - 2.1.1 Non-Light/vision-based indoor localisation methods . . . 5
  - 2.1.2 Light and vision-based localisation methods . . . 5
  - 2.1.3 Projection-based localisation techniques . . . 6
  - 2.1.4 "Augmenting Indoor Inertial Tracking with Polarized Light" 8
  - 2.1.5 Birefringence . . . 9
- 2.2 Communication techniques . . . 11
- 2.3 VLC communication . . . 12
- 2.4 Relevant UAV VLC works . . . 13

3 Localisation with constrained hardware 15
- 3.1 Selecting the correct drone . . . 16
- 3.2 Transmitter polarisation pattern . . . 17
- 3.3 Receiver . . . 20
  - 3.3.1 Colour detection . . . 20
  - 3.3.2 Colour selection . . . 22
  - 3.3.3 Colour classification . . . 23
  - 3.3.4 Prototype reduction . . . 25
  - 3.3.5 Drone localisation . . . 26
  - 3.3.6 The particle filter . . . 27
  - 3.3.7 Grid design and creation . . . 28
  - 3.3.8 Motion model . . . 29
- 3.4 Transmitter design and implementation . . . 32
  - 3.4.1 Final transmitter setup . . . 34
  - 3.4.2 Receiver design implementation . . . 35

4 Communication link integration 37
- 4.1 Modulation techniques . . . 38
- 4.2 FSK VLC link implementation . . . 39
  - 4.2.1 VLC hardware implementation: . . . 41
  - 4.2.2 Packet Layout . . . 41
  - 4.2.3 VLC interference . . . 42

5 Embedded software considerations 45
5.1 Colour detection implementation 46
5.2 VLC communication link implementation 47
5.3 Particle filter implementation 47
5.4 Debug and interface upgrades 48

6 Evaluation 49
6.1 Experimental setup 49
6.1.1 Ground-truth reference 50
6.2 In flight colour detection 50
6.3 Communication link 52
6.4 Motion model performance 55
6.5 Particle filter performance 56

7 Conclusion 59

8 Discussion 61
8.1 Limitations and Recommendations 61
8.1.1 3D tracking: 61
8.1.2 Implementation limitations on localisation 61
8.1.3 Pattern imperfections 62
8.1.4 The use of a more powerful source 62
8.1.5 Sensor orientation 62
8.1.6 VLC link speeds 63
8.2 Future extensions 63
8.2.1 Advanced Grid design 63
8.2.2 Adding a secondary projector 63
8.2.3 Making the transmitter mobile 63
8.2.4 Using retroreflectors for localisation 64

A Converting ΔRGB to ΔHSV 71

B Bill of materials 73

C Setup hardware 75

Chapter 1

Introduction

When flying a simple drone, e.g. a Quad-Copter, have you ever noticed that it does not hover in the same spot for prolonged periods? It starts slowly drifting away from its original position without commanding it to do so. What phenomenon causes it to start drifting?

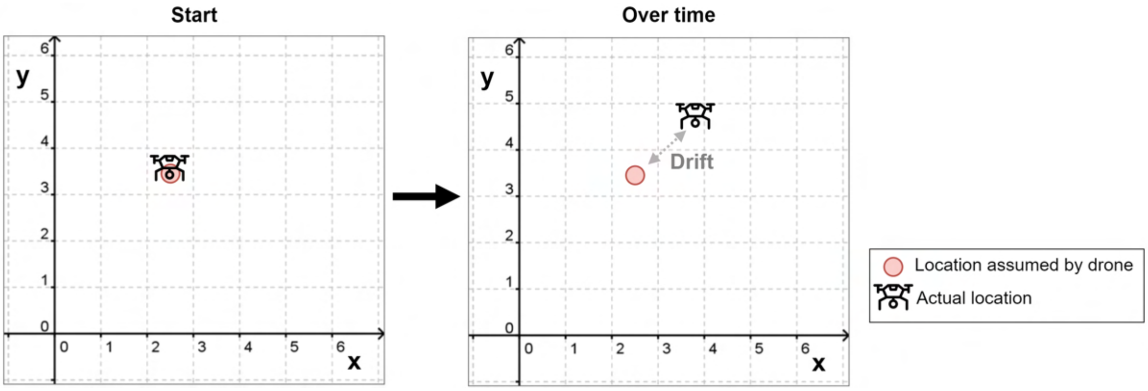


Figure 1.1: **Drone hovering and suffering from integration drift.**

Let us start at the root of the drifting problem illustrated by Figure 1.1. Drones that can hover independently usually contain an internal measurement unit (IMU). Such an IMU contains several sensor types, such as an *accelerometer* (acc), *gyroscope* (gyro), allowing the IMU to measure the acceleration and angular velocity. To control a drone, an estimate of the position and velocity is required. The IMU sensors do not directly measure these parameters, requiring internal integration algorithms to transform the IMU measurements into an estimate of the drone's velocity and position.

The problem is that the IMU measurements suffer from sensing errors varying from a bias to electrical or mechanical noise [44]. Therefore the integration algorithms also incorporate all the measurement errors and cause them to accumulate quickly until, over time, a substantial error is formed in the velocity and position estimate. This phenomenon is called integration drift and is illustrated in Figure 1.1. It causes the system's predicted velocity and position to diverge from the system's actual velocity and position. This difference results in the drone drifting away from its actual location without it knowing.

Over the years, many solutions have been proposed to counter IMU drift. Most techniques provide a known anchor used as a reference point for recalibration of the system's estimate. Some example techniques use GPS [17], a camera observing the environment [29], a human in the loop [35] or image/pattern projection [58, 59, 31].

Of the mentioned techniques pattern projection is especially interesting for drone implementations; the key aspect is that these projection setups allow the transmitter to contain most of the system's heft and complexity. An example is a cinematic projector that contains all heavy lenses and consumes large amounts of power, allowing the receiver to be a simple, lightweight, power-efficient sensor like a colour sensor or photodiode. Perfect for mounting on a mobile battery-powered platform like a drone.

This thesis builds upon a projection technique proposed in "Augmenting Indoor Inertial Tracking with Polarized Light" [48]. In this reference paper, the authors describe a novel projection-based IMU calibration technique where the main idea is to create a specialised, transparent grid pattern. This grid pattern is projected into a room and can be observed by a drone flying through the projection. When light passes through this grid pattern, the special birefringent [40] transparent material this grid is made of causes the electric field oscillation direction in the light waves (the polarisation) to change. This change in polarisation direction is imperceptible to humans [25]. However, when observing the grid through a material that only allows the light of 1 specific oscillation orientation to pass (a linear polariser), all cells in the grid projection obtain a colour. Figure 1.2 illustrates this principle of the cells in the grid colouring. Now, by using a polariser stacked on top of an RGB colour sensor mounted on the drone, these perceived colours are digitised and act as fine-grained colour landmarks for the drone to use in correcting for the IMU integration drift. The implementation allows a drone carrying RGB sensors to localise itself based

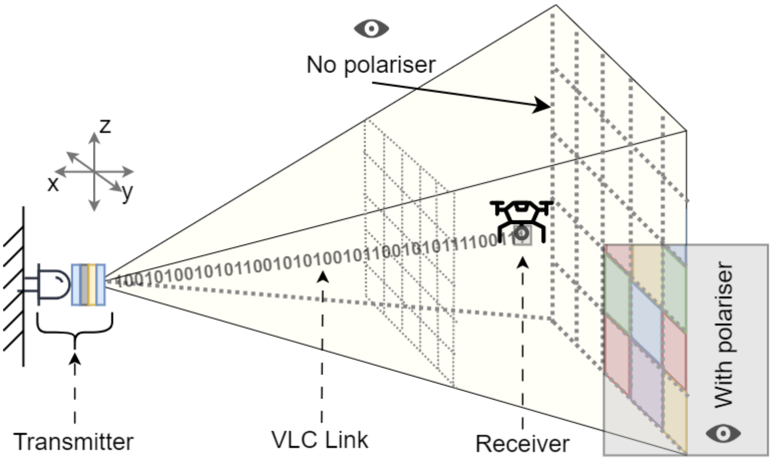


Figure 1.2: **Simplified overview of all major elements included in this thesis.**

on the corrected IMU data resulting in an accurate localisation estimate for autonomous flight.

After having achieved stability in the drone's position using the projection technique, the next question is; how can it be instructed to move? A common way to control a drone is by utilising a Radio Frequency (RF) link [39], the problem is that this RF link requires additional hardware on both the transmitter and receiver. This poses the question: Is it possible to use the existing localisation projection infrastructure for communication? Utilising the projection light source for communication leads to the emerging field of Visible Light Communication (VLC). This wireless technology modulates visible light from light-emitting diodes (LEDs) to encode and transmit data. By including rapid switching modulation capabilities into the projection setup and equipping the drone with a photodiode to detect the modulated light, it becomes possible to establish a light-based communication link with the drone. This link enables the drone to receive instructions and fly autonomously.

To summarise: this thesis introduces the system presented in Figure 1.2. The figure illustrates the projection-based localisation of a drone using the colours that appear when observing the projection through a polariser. Simultaneously, the system modulates the projection light source to provide instructions to the drone while traversing through the projected grid. Fusing the projection-based localisation and visible light communication allows for simultaneous localisation and communication. And sets the first steps in working towards a completely visible light-based drone platform.

1.1 Research question

The reference paper [48] has implemented and demonstrated the polarisation-based localisation technique in real-time on ground robots for both the 2D and 3D scenarios. This thesis investigates the feasibility of using this same technique to localise flying drones in an indoor environment. Additionally, this thesis explores the option of implementing a VLC communication link that utilises the existing infrastructure for localisation. Investigating the possibility of visible light-based simultaneous localisation and communication on flying drones is how this thesis adds novelty to the techniques used by the reference paper.

The research question poses: Can we achieve real-time simultaneous visible light localisation and communication on a drone? More specifically:

- Is it possible to implement the polarisation-based localisation technique presented in the reference paper [48] in real-time on a limited compute power drone?
- Is it possible to use the existing projection infrastructure to establish a VLC link with minimal changes and interference to the localisation system?

Answering the above questions poses several challenges. Some of the major challenges are:

- Obtaining reliable localisation in a drone context.
- Establish a reliable VLC link that does not interfere with the localisation mechanisms.
- Establish both the localisation and communication in a self-contained, real-time, limited compute power embedded drone context.

1.2 Contributions

This thesis presents several noteworthy contributions to the field of drone technology. First, it introduces implementing the projection-based polarisation pattern localisation described by the reference paper [48], accurately localising drones within the projection area of approximately 1.6×1.6m. Moreover, the localisation technique is successfully implemented in real-time on a limited compute power drone platform, demonstrating its practicality and implementation feasibility in real-world scenarios.

Additionally, this research explores the integration of Visible Light Communication (VLC) into the existing projection infrastructure without interfering with the localisation accuracy of the system. The achieved addition of a VLC link enables one-way communication from the transmitter to the receiving drone, with support for up to three drones operating in parallel. This fusion of projection-based localisation with a VLC link provides a significant step towards a future where drones rely only on visible light-based technologies for localisation, communication and navigation. The combination can advance the performance of drones in various applications, varying from autonomous navigation to secure communication.

1.3 Thesis outline

This thesis contains the following chapters: The Background provides information on relevant localisation techniques, both VLC and non-VLC based. Additionally, it explores the property of birefringence and provides several VLC communication options for drones. This chapter is then followed by the design and implementation chapters of this work's localisation and Visible light communication aspects. Rounding it off with a chapter on the embedded software considerations providing an overview of the major implementation optimisations. After this, a chapter evaluates the performance of the major systems, followed by the conclusion and discussion, where the discussion introduces some interesting future research directions.

Chapter 2

Background

This chapter provides information on relevant localisation techniques, both VLC and non-VLC based. Additionally, it explores the property of birefringence and describes several existing VLC communication options for drones.

2.1 Existing localisation techniques

As explained briefly in the Introduction, a drone or any other device that solely utilises an IMU for navigation will suffer from an integration drift. Numerous techniques have been developed and implemented to counter this phenomenon [59], these techniques vary wildly in implementation, and there exists not one solution that fits all problems. This thesis will focus on implementing the indoor localisation technique proposed in the reference paper [48]; therefore, the focus on localisation methods will be limited to indoor methods only. At first, some non-visible light developments will be mentioned, followed by a comprehensive overview of the latest light and projection-based localisation techniques.

2.1.1 Non-Light/vision-based indoor localisation methods

The paper by Yan [59] provides a comprehensive overview of some of the most recent developments in indoor localisation with a focus on wireless techniques. Devices that localise themselves use known sources either already present or purposely placed as anchors and determine their position based on these sources. Wireless signals form the basis for most localisation techniques because they are abundant in most indoor environments. Some examples are WiFi based [33, 50, 62], some Ultra Wide Band based techniques are summarised in [3],some use Sound [34], some (Geo)magnetic signals [47], and some use RFID tags [7]. Of these techniques, WiFi is probably the most extensively researched approach.

2.1.2 Light and vision-based localisation methods

Besides the previously mentioned techniques, light-based localisation is also considered an appealing approach for indoor environments as it has strong privacy properties since light waves cannot penetrate walls, making it more difficult for outside entities to intercept data compared to other non-light-based techniques.

Furthermore, most urban environments require some form of illumination, allowing light-based techniques to serve multiple purposes. The following surveys by Rahman [42] and Yang [59] provide a broad overview of the recent advancements in localisation with visible light. The research into visual light localisation can be subdivided into the following categories:

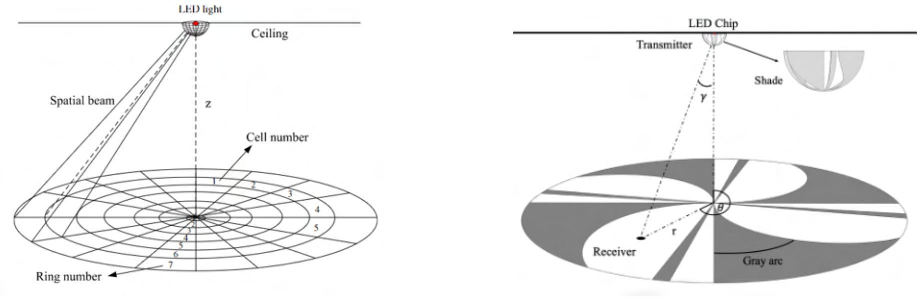
- Anchor points/ landmarks where usually LEDs are utilised to provide some form of uniquely identifiable signature combined with the Relative light intensity. These techniques' accuracy heavily depends on the density and the type of source, sometimes even utilising specialised LEDs [4, 57].
- Computer vision analysing images captured by camera equipment. The downside to these techniques is the high computing and sensing capabilities required when frequent high-resolution frames need to be analysed. In conjunction with this, there is also the sensitive topic of human privacy [43].
- Image or pattern projection. There exist several different approaches in this category. The first is a digital projector that can project a dynamic or static image into space. The second approach is a virtual reality setup utilising sweeping laser scanning technology, and finally, the third approach is a form of shadow casting where specific parts of a projection are turned on or off in rapid succession. No matter the projection technique, the signal is captured by receivers that use the information to estimate their location [58, 59, 31].
- Light detection and ranging (LIDAR). The final category utilises LIDAR, where lasers pulses are emitted, and reflections are collected. Allowing for simultaneous localisation and mapping (SLAM) [23].

The options above provide a wide range of possibilities for light-based localisation. This thesis aims to localise a drone, requiring minimal power consumption, compute power and weight to be on the drone side of the system. Based on these goals, the branch focussing on LIDAR is deemed too much weight to carry, and computer vision is too heavy to compute in real-time. Leaving pattern projection and Anchor points as possible options to localise a drone.

2.1.3 Projection-based localisation techniques

Of the remaining branches for light-based localisation, one exciting field applicable to drones is projection-based techniques. Most projection-based techniques have the pleasant property that the bulk of the power-consuming work is performed by the projecting or transmitting side of the system. Hence the receiving or localising side of the system can be a simple low-power sensor, E.g. photodiode or colour sensor. Both examples are easily implementable on a battery-powered drone system.Therefore, the idea used in this thesis for localisation belongs to the image and pattern projection category.

The first promising technique is SpinLight [58]. This paper introduces a precise light positioning system that consists of a signal transmitter with an infrared LED covered by a rotatable hemispherical shade with a unique cutout pattern. The receiver is equipped with a photodiode. By rotating the hemisphere, the transmitter emits a unique sequence of shaded and non-shaded light signals for each point in the covered space. Figure 2.1a provides a system overview. SpinLight achieves a median location error of 3.8 cm. The downside of this approach is the mechanical aspect, requiring a motor and a fast-spinning shade. CurveLight [59] provided in Figure 2.1b is similar to the paper on SpinLight, a rotatable hemispherical shade is used. However, the key innovation lies in the design of curves on the shade. Through experiments, the system achieves an average 2-3 cm accuracy and a 36 Hz update rate using a single transmitter. The mechanical limitation still holds.



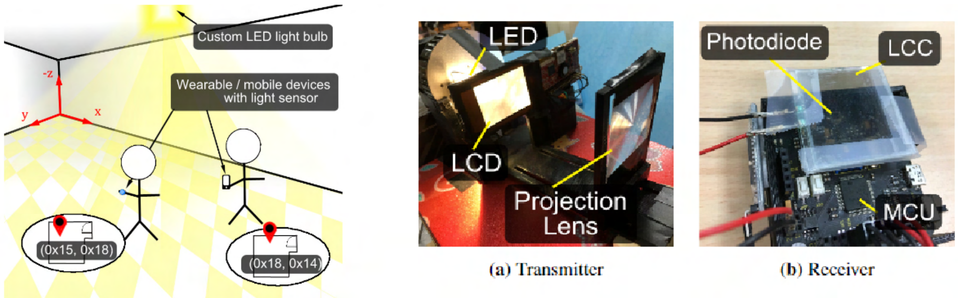
(a)

(b)

Figure 2.1: **(a) SpinLight overview [58]. (b) CurveLight overview [59].**

An approach that removes the mechanical parts while achieving a projection mechanism is CELLI [54]. This paper introduces an indoor visible light positioning system that uses a single LED as the transmitter and a simple light sensor as the receiver. CELLI utilises a small LCD at the transmitter side to project polarised light to form unique cells. The receiver can then identify its assigned cell by detecting the polarisation modulation signal of every cell. Experimental results show that CELLI achieves a median 3D error of 11.8 cm and a median 2D error of 2.7 cm. An overview of CELLI is provided in Figure 2.2. Instead of an LC, a digital micro-mirror display (DMD) can also be utilised: an example is provided in Figure 2.3 [31]. In this paper, the authors propose a fine-grained positioning system using a DMD projector as the transmitter. In this design, every pixel of the DMD carries a unique location signal that a simple photodiode can receive. They are reporting mm-level accuracy. The downside of this method is the compute-heavy, bulky and power-hungry transmitter.

By following the trend of pixel level accuracy, the following work [24] introduces



(a) Transmitter

(b) Receiver

Figure 2.2: **CELLI overview [54].**

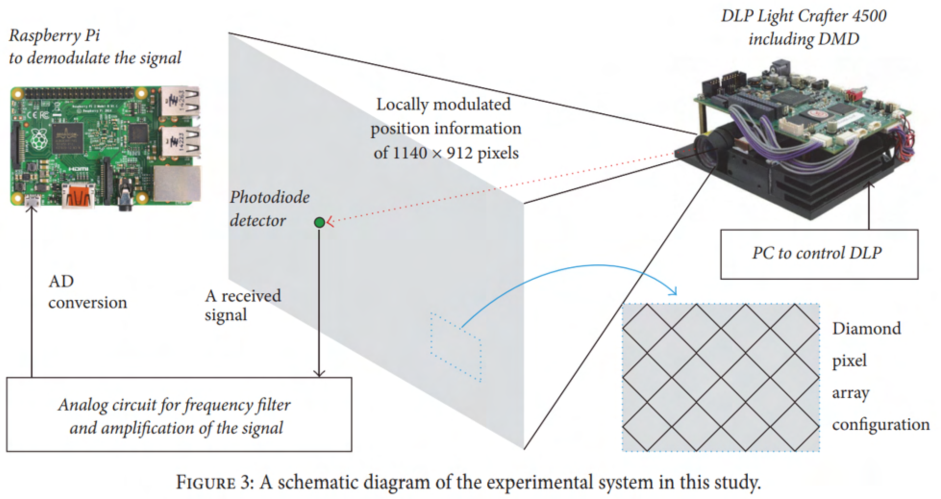


Figure 2.3: **DMD overview [31].**

a novel method for localising and navigating multiple mobile robots using coded light superimposed onto a visual image projected on the robots. The robots receive this encoded message, determine their localisations, and follow a target direction. The system requires a projector and can use any type of projection surface. The work has experimentally demonstrated localisation accuracy for stationary and moving robots. Figure 2.4 provides an overview of this work.

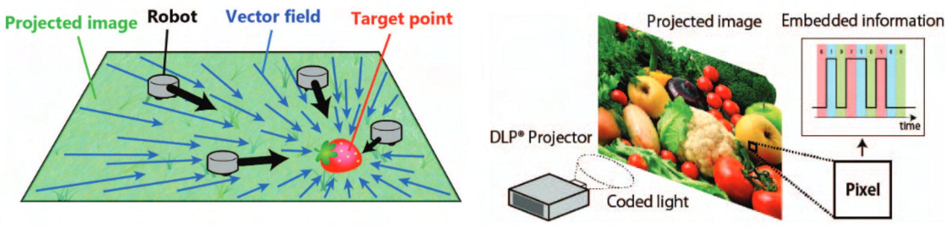


Figure 2.4: **Projection-based localization and navigation[24].**

Besides the provided examples, there are many more projection-based localisation works out there. Some interesting ones are Spotlight which uses a helicopter as the source [46], RainbowLight, which analyses birefringence patterns with a camera [36], PIXEL proposes polarisation based modulation relieving the computational load on visible light positioning devices [61], Localisation by modifying a single led lamp [37].

2.1.4 “Augmenting Indoor Inertial Tracking with Polarized Light”

As described before in the introduction, the localisation aspect in this thesis heavily relies on the principles presented in this paper: “Augmenting Indoor Inertial Tracking with Polarized Light” [48]. This work refers to this paper as the *“reference paper”*. In brief, this work proposes a method to improve the accuracy of indoor IMU integration tracking. It does this by projecting a static light polarisation pattern using birefringence (see Section 2.1.5).

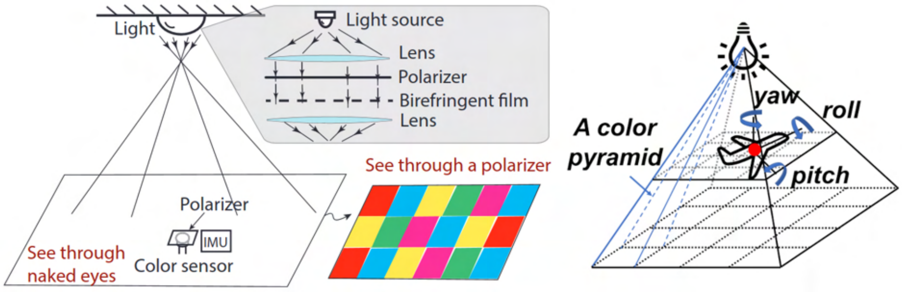


Figure 2.5: **Augmenting Indoor Inertial Tracking with Polarized Light [48].**

As illustrated in Figure 2.5, the transmitter passes light from a fixed LED through a lens, gathering the light in a straight beam. This beam then passes through a linear polariser leaving only the lightwaves with their orientation aligned with the polariser. After the polariser, the light passes through a transparent grid material with different thicknesses for the cells in the grid. The material used has birefringent properties. This causes the polarisation direction of the lightwaves to rotate depending on the thickness of the transparent birefringent material. Finally, a secondary lens projects the light into a room. The polarisation orientation of light is imperceptible to the human eye [25], making the projected grid pattern with the different polarisation orientations look like a regular light projected into a room.

However, because of the birefringent material and the rotation of the polarisation direction of the light waves caused by the material, different colours appear when observing the pattern through a secondary linear polariser, see Figure 2.5 for an example. A simple RGB colour sensor can digitise the perceived colour. These observed colours will act as fine-grained optical landmarks. These are then used to correct the IMU integration drift. This correction process uses a so-called particle filter [6] where the observed colours and IMU data are fused to estimate the current location of a robot within this polarisation pattern.

The birefringent transparent grid mentioned before uses regular transparent office tape as the birefringent material. This office tape is stacked to create different thicknesses and obtain different colours when observing through a linear polariser. This office tape is cheap, making it a cost-effective and simple solution. Experimental results demonstrate a median error of 4.3 cm for 2D and 8.1 cm for 3D tracking, as well as the robustness of the approach in various environments.

2.1.5 Birefringence

A frequently mentioned item in this thesis is the polarisation pattern made of transparent tape material and the resulting colours when observing it through a polariser. By referring to the article on “Principles of Birefringence” [40],

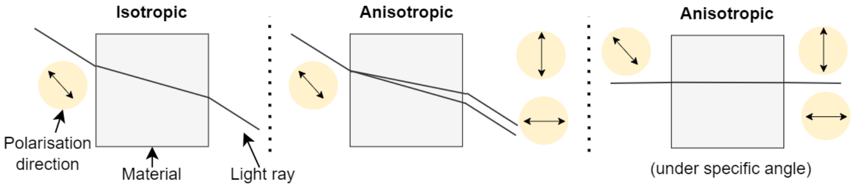


Figure 2.6: **Regular refraction, double refraction, and double refraction under specific angles.**

one can understand that when working with transparent solids, many have the property of being optically isotropic. This means that for light, the index of refraction is equal for all directions through the material. On the other hand, there are anisotropic, also known as birefringent materials. These materials interact with light in a manner that is dependent on the orientation of their molecular structure with respect to the incoming light. When light passes through this type of material, it splits into two components with different refractive indices, polarisation orientations and velocities. Figure 2.6 illustrates linearly polarised light entering both an isotropic and anisotropic material. As mentioned, the light splits into two separate rays when entering an anisotropic material, the so-called "fast ray", which travels at the same speed independent of the propagation direction and the "slow ray", which travels with a velocity depending upon the propagation direction within the material. Besides the difference in propagation speed, the material also causes the two rays to become polarised perpendicular to each other.

The following interesting effect happens when the incoming light is linearly polarised and aligned along a specific optical axis of the anisotropic tape material. As illustrated by Figure 2.6, instead of the ordinary and extraordinary rays diverging, they instead propagate in the same direction. Note that the rays are still perpendicularly polarised, and the propagation speeds are different inside the material.

When passing through the anisotropic tape material, the slow ray has reduced speed compared to the fast ray, which causes a shift between the two rays. This shift is called retardation, and the shift amount depends on the anisotropic material's thickness. E.g. the number of tape layers stacked.

With the above information, the simplified 2D representation given in Figure 2.7 (left) provides an illustration to show how the colours appear. In this figure blue monochromatic linearly polarised light is passed through the anisotropic tape material. As described within the material, the fast and slow rays become perpendicularly polarised, and retardation occurs, shifting the phase in the slow ray. Note that the two rays are separated for illustrative purposes. Because of the perpendicular polarisation the lightwaves can not interfere with each other. However, arriving at the secondary linear polariser, parts of the fast and slow rays pass through depending on how far their polarisation angle diverges from the polarisation direction of the polariser. After the polariser, their polarisation directions align, and the light waves start to interfere. Depending on the amount of retardation, the rays either construct or destructively interfere.

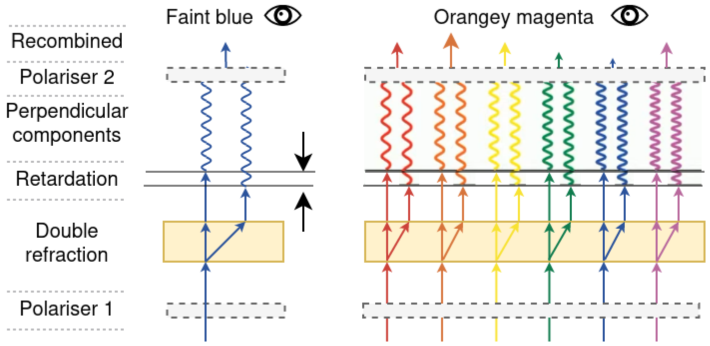


Figure 2.7: **Interference colours caused by the retardation of the light. (left) monochromatic blue light, (right) white light interfering, resulting in an orangey magenta.**

Now, when using white light containing all different wavelengths, the effect illustrated by Figure 2.7(right) occurs. Retardation is the same for every wavelength. Causing different types of interference to happen after the secondary polariser. This results in different wavelengths constructively interfering, such as orange, and others destructively interfering, such as blue. The resulting combination of colours is called the interference colour. This is the colour humans perceive and also the colour that is used for localisation in the reference paper [48] and this thesis. By varying the thickness of the anisotropic material e.g. a varying amount of stacked tape layers, different interference colours are obtained.

2.2 Communication techniques

This next section covers the essential information related to wireless communication required to understand this thesis.

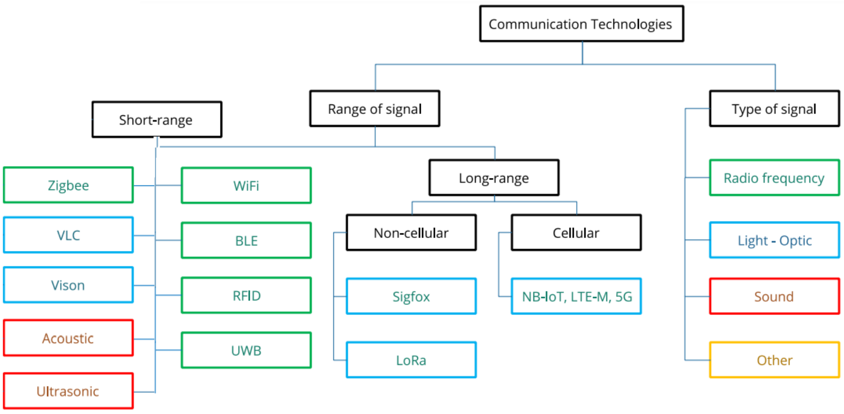


Figure 2.8: **Technologies enabling wireless communication [21].**

Wireless Communication is an old and broadly studied subject, having developed numerous techniques and protocols working with various transport mediums. Many of these wireless devices use EM waves for communication. The frequencies at wich such devices usually operate mainly depends on the free-to-use frequency bands provided by governmental bodies and the purpose of the device: low-bandwidth long-range data uses the 900 MHz and 2.4 GHz bands, whereas high-bandwidth communication like Video streaming uses 5.8 GHz or higher frequencies lacking in range but allow for high data throughput [2].

Using the available EM frequency ranges, many communication technologies have been developed that allow devices to communicate data reliably. Figure 2.8 shows a tree graph containing some of the many communication technologies available. These wireless technologies apply to many fields and are not device-specific. However, because this thesis relies on drones, the research will be limited to drones or unmanned aerial vehicle (UAV) implementations. Using drones, both short- and long-range technologies have been implemented, each with benefits and shortcomings. Some promising examples implemented for long-range communication techniques are: Cellular networking 2G (GSM), 3G (UMTS), and 4G (LTE/WiMAX) [19], Sigfox [21], and satellite networks[27].

Besides long-range communication, there is also dedicated short-range communication. Usually optimised for use in low power applications like ZigBee [11]. Or as close range high datarate links like WLAN(WiFi)in the frequency bands: 2.4 GHz, 3.6 GHz, 5 GHz, and 60 GHz [19]. The focus of this thesis will be reusing the existing light infrastructure used for projection-based localisation, therefore, the research will focus on short-range VLC, with a special focus on drone or UAV implementations.

2.3 VLC communication

Visible light communication, or VLC, is a technology that uses visible light as a transmission medium to transmit data between devices. VLC occupies the spectrum from 750–380 nm as illustrated by Figure 2.9.

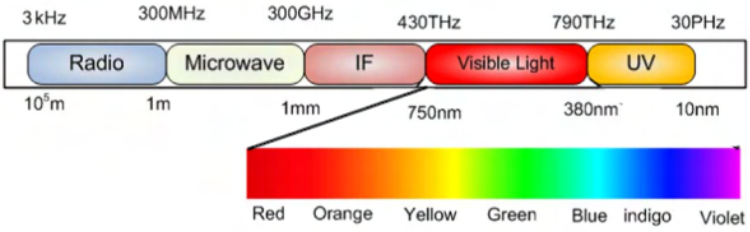


Figure 2.9: **VLC frequency spectrum [30].**

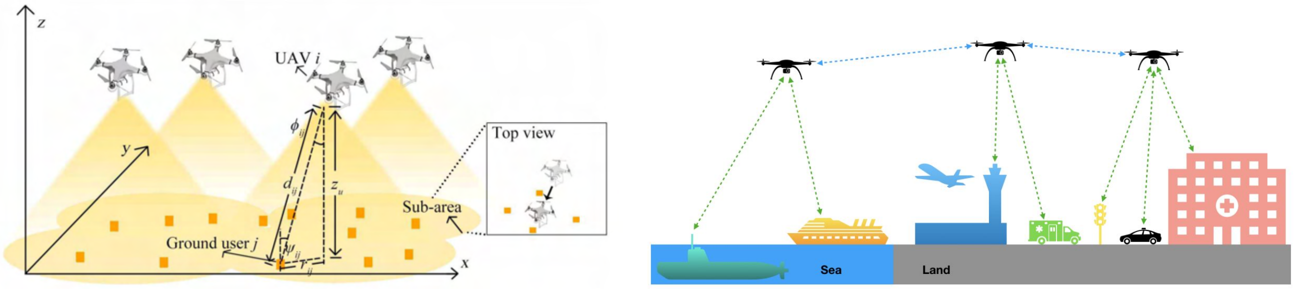
VLC is more advantageous because it provides higher data rates than other wireless communication technologies. This data rate advantage is because of the large bandwidth available in the visible light spectrum. Furthermore, VLC has a higher level of security compared to RF technologies since light waves cannot penetrate walls, making it more difficult for outside entities to intercept data. VLC also has some limitations. One disadvantage is that it requires a direct line of sight between the transmitter and receiver. This can be a problem in areas where obstacles such as walls, furniture or drones can block the light signal.

Despite these limitations, VLC is a promising technology that has potential applications in various fields, such as indoor positioning, automotive vehicle-to-vehicle communication, and Li-Fi [45].

2.4 Relevant UAV VLC works

Because of the high directiveness of VLC links compared to RF communication, most studies on using VLC consist of a static transmitter and receiver pair where precise alignment is a large requirement [8]. When narrowing down to mobile studies, some examples have investigated vehicle-to-infrastructure or vehicle-to-vehicle links like: [18] where they provide the current challenges and goals of VLC for use in vehicle communications. The problem is that cars are much larger and undergo smooth motions, unlike a drone mid-flight.

Only a few actual implementations show a VLC link implemented on a UAV or drone. Some studies proposed the use of VLC on drones to provide temporary Internet coverage to people [60]. Figure 2.10a illustrates this approach. A more recent study [20] assesses the link between a drone's camera and a ground light illustrated by Figure 2.10b.



(a)

(b)

Figure 2.10: **(a) The application scenarios of VLC wifi support with UAVs [60]. (b) Conceptual illustration of drone-based use-cases for LED to camera communication [20].**

Regarding the feasibility of establishing a communication link, previous works have investigated the option to run an FFT on a drone platform [4] where multiple LED sources are identified by their frequency using an FFT requiring only a small step to develop this further to establish communication. Finally, some progress has been made in modelling the channel between a ground station and a drone [52].

This thesis tries to build directly on the basic ideas of these works. Attempting to implement a one-directional VLC link between the transmitter and the receiver. However, this thesis has the added benefit that it also provides localisation while communicating.

Chapter 3

Localisation with constrained hardware

A flying drone will slowly drift due to noise in the IMU measurements accumulating over time. Because of this drifting, any tasks that require accurate autonomous flight are impossible, making it essential for these errors to be removed from the system to allow for precise localisation. Various techniques are available to remove the integration drift. As explained in the background, projection-based techniques are an exciting field applicable to drones.

This thesis focuses on implementing and further developing the projection-based localisation techniques described in “Augmenting Indoor Inertial Tracking with Polarised Light” [48] (the *“reference paper”*). Transferring the localisation technique first implemented on a robot cart to a flying drone. In summary, the reference paper proposes a method to improve the accuracy of indoor IMU integration tracking. It does this by projecting a static light polarisation pattern using the material property of birefringence (Section: 2.1.5). The projected

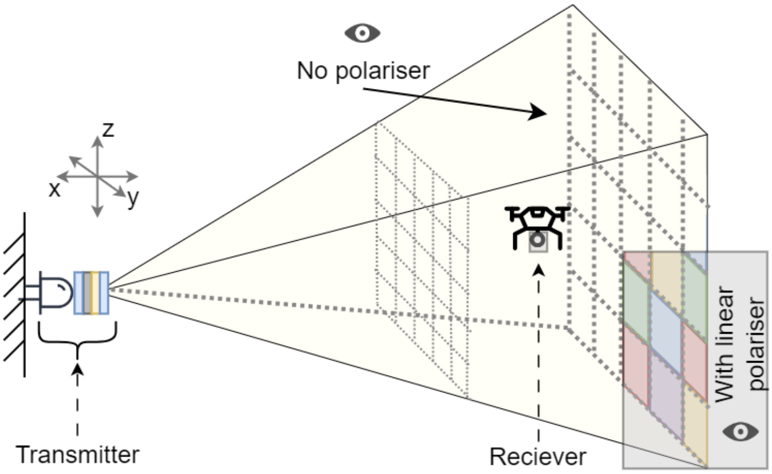


Figure 3.1: **Simplified overview of all major elements required for the localisation part of this thesis**

pattern is imperceptible to the human eye however, when observing the pattern through a polariser, different colours can be seen depending on the polarisation pattern. A simple RGB colour sensor with a polariser can record the received colour. These observed colours will act as fine-grained optical landmarks. These are then used to correct for the IMU integration drift and provide an accurate receiver localisation. The proposed approach utilises regular transparent tape stacked in various configurations and heights to create the grid polarisation pattern that can be observed as colours.

A global view of all essential components is required to better understand the different parts involved in the localisation part of this thesis. The two core parts are the transmitter and the receiver. Figure 3.1 presents this global overview. The purpose of the transmitter is to project the polarisation pattern into space. It comprises a light source, several lenses, a linear polariser, and a polarisation pattern made of regular tape. This polarisation pattern has a grid shape and is specially designed to enable rapid localisation. When the light passes through the transmitter stack, it forms a projection in the shape of a sideways pyramid where the polarisation pattern is uniformly projected, increasing in size over distance. The receiver is a drone, and the goal of the drone is to obtain an accurate position of itself free of drift. It does this by reliably detecting the colours the polarisation pattern creates when the incoming light is perceived through a linear polariser. This colour is then digitised, identified and processed for localisation.

Implementing the reference work on limited hardware (the drone) posed several challenges related to reliable colour detection, colour classification and particle filter localisation. Each of the problems will be tackled in this chapter while simultaneously adding novelty and improvements.

3.1 Selecting the correct drone

Before diving into the localisation details it is important to select the core hardware to this project such that all design decisions are made with this in mind. In this thesis, the core hardware is a drone. An important question then is: what drone to use? Currently, a wide variety of drones are available on the market, varying in size, lifting strength and flight duration. This section provides a list of requirements that narrows the selection for this thesis and a general approach for other works. The drone selection poses the following requirements:

- **Ground truth localisation**: Support for accurate X, Y, Z ground truth localisation to test the newly developed localisation algorithms.
- **Access to live IMU data**: Access to live IMU data is required to run the particle filter motion model.
- **Controllability**: Preference for a motion planner or trajectory control over individual motor control.
- **Carrying strength**: Capability to carry a lightweight sensor cluster.

- **Onboard compute power**: Sufficient amounts to implement a particle filter, fast Fourier transform, motion model and machine learning classification.
- **Community and support**: Must have an active community to answer possible questions during development.
- **Battery life**: A minimum of 5 minutes.
- **Out-of-the-box readiness**: Little extra work required for flight readiness.

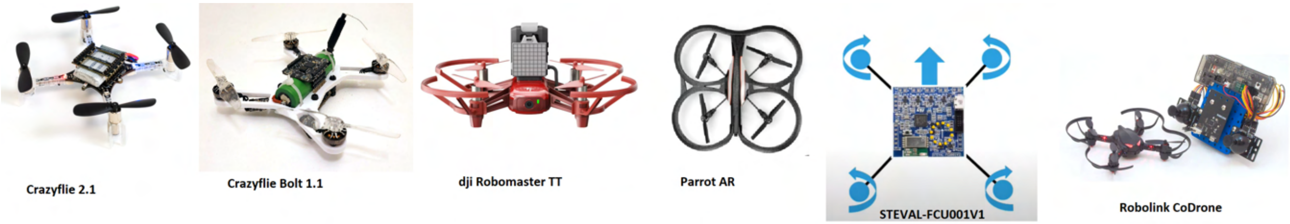


Figure 3.2: **Selection of drones aimed at academic purposes.**

Figure 3.2 provides a selection of drones aimed towards development currently available on the market. Of the listed drones, the following are removed from the selection for various reasons:
1) Discontinuation: Parrot AR, STEVAL-FCU001V1.
2) Limited hardware and software access: RoboLink.

The remaining drones are the Crazyflie 2.1, the Crazyflie Bolt, and Robomaster TT. All of which have been used in academic research before [4, 41, 9]. Based on the above requirements, table 3.1 shows the best choice being the Crazyflie 2.1, with the option to upgrade to the Crazyflie Bolt for additional carrying strength and battery life.

Table 3.1: **Drone comparison.**

Drone	Ground truth	IMU	Control	Strength	Compute	Support	Battery	Readiness
Cf 2.1	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Cf Bolt	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Robo TT	Third-party	No	Yes	Yes	Yes	No	Yes	Yes

3.2 Transmitter polarisation pattern

Observing Figure 3.1 the most essential task of the transmitter is the projection of the polarisation pattern. This pattern will play an essential role in the localisation of the drone. Using the principle of birefringence explained in Section 2.1.5 regular tape is used to create a range of colours when observed through a polariser more on this in Section 3.3.1. This section answers the question of how to utilise a finite set of colours for localisation by creating unique colour sequences in a grid. The reference paper [48] uses a similar technique. This thesis adds novelty to the implementation of the pattern generation algorithm taking a new approach while preserving the end goal. Additionally, this thesis adds to the manufacturing capabilities of creating the physical grid creating a more detailed projection with a larger number of smaller cells.

Ideally, the polarisation pattern would contain a unique colour for every cell allowing for immediate localisation based on the unique received colour. This immediate localisation is impossible because of the discrete thickness of the tape used to create the colours and the noise in identifying the colours. Hence only a limited set of colours can be used. In order to still allow for localisation, colour sequences have to be used. The design of these sequences aligns with the following goals:

1. Robust detection between cells.
2. Equal colours can't be placed side by side.
3. Minimal or no repeating sequences.

Figure 3.3 illustrates how these goals can be fulfilled. Achieving the first and second goals is done by positioning cells with high-contrast colours next to each other. This high contrast produces a sharp change in colour when crossing a boundary to a different cell, ensuring reliable detection.

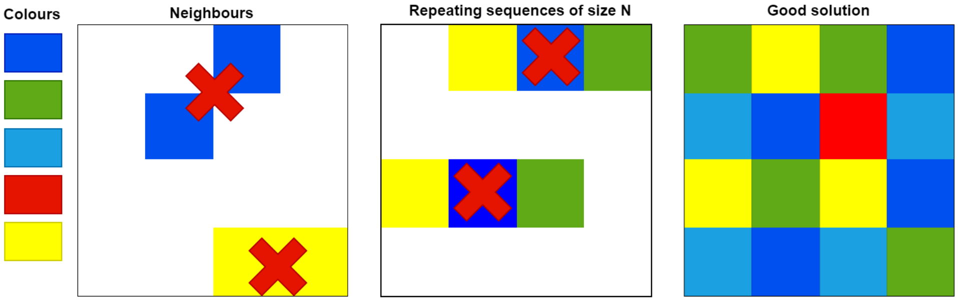


Figure 3.3: **Grid design with a total of five unique colours, giving examples of forbidden placements and sequences. Finishing with a complete 4×4 grid.**

The third goal is achieved by not allowing for repeating colour sequences of length n such that any straight trajectory through the cells results in a unique sequence. More specifically:

- Colour sequences of size n originating from a specific cell may not be the same in any of the eight cardinal directions (N, NE, E .. W, NW). E.g. the colour sequence "GBG" where n = 3 may only originate from the cell in 1 direction and not in any of the other directions.
- Colour sequences of size n in the global grid along a cardinal direction may not be the same. E.g., the Sequence "GBG" where n = 3 may exist

in the East direction only once. However, it is allowed to exist facing the south direction.

In order to generalise the grid design for any size grid, we design the following algorithm: we have a grid of $M \times N$ cells, L total colours (tape configurations), the set of all possible colour sequences of length n: S, where the grid designer sets the sequence length n. And finally, a function $D(L_1, L_2)$ that evaluates the difference between the colours. To be more specific: $D(L_1, L_2)$ evaluates the Euclidean distance on the ΔH, ΔS plane (more on this in Section 3.3.1).

Within the grid, every cell contains a local set T containing all used colour sequences of length n originating from that cell in all eight cardinal directions:

$$T = [T_N : [S_1], T_{NE} : [S_2], \cdots, T_{NW}[S_8]]$$

The grid has a global set G, containing all used colour sequences of length n along the eight cardinal directions.

$$G = [G_N : [[S_1], [S_2], \cdots, [S_i]], G_{NE}[[S_{1+i}], [S_{2+i}]], \cdots, G_{NW}[\cdots]]]$$

Example: Figure 3.3 shows a completed grid with unique sequence length $n = 3$. The top left cell (grid[0,0]) in Figure 3.3 contains in the local set:

$$T_{0,0} = [T_E[G, Y, G], T_{SE}[G, B, Y], T_S[G, LB, Y]]$$

The global set G (only showing the East direction) would contain:

$$G = [G_E[[G, Y, G], [Y, G, B], [LB, B, R], [B, R, LB], [Y, G, Y], [G, Y, B], [LB, B, LB], [B, LB, G]]]$$

To solve any $M \times N$ grid, the algorithm:

- Maximises: $D(L_1, L_2)$
- Minimises: Repeated colour sequences in G and T

In practice, the algorithm iteratively adds more colours to the grid, starting in the top left with a user-specified colour. The algorithm then repeats the following sequence of events until all cells are filled:

1. Pick the next cell to place a colour filling the current row first.
    (a) Look at all neighbouring cells and remove the colours already in use from the set of possible colours L.
    (b) Look at all neighbouring cells and with the distance function $D(L_1, L_2)$, pick the colour with the largest average contrast to all neighbours.
    (c) With the new colour, check the local set T for uniqueness.
    (d) With the new colour, check the global set G for uniqueness.
2. If there are no violations, place the colour and proceed to step 1.
3. If there are violations, remove the colour from the available set L and go back to step 1b.
4. No colours left? Perform backtracking and go to step 1.

The backtracking part of the algorithm cycles back a random amount of cells, removes all cells ahead and picks the second best option from the set L aiming to avoid the deadlock caused by the previous sequence.

With the algorithm finalised, we can not yet create the colour (polarisation) grid. The final grid design and avaiable colours will be provided in Sections 3.3.7 and 3.3.2 as knowledge of the receiver is essential in this process.

3.3 Receiver

The receiver's goal is to reliably detect the colours created with the polarisation pattern when the light is perceived through a linear polariser. This section starts with colour selection, detection and classification. Followed by localisation and the actual polarisation grid design.

3.3.1 Colour detection

The polarisation pattern is projected in space. The challenge now is to detect this pattern reliably. Simply put, it is the process of detecting a colour observed by a colour sensor behind a secondary polariser. The colour sensor returns the light intensity values in the Red, Green, and Blue (RGB) channels. The key challenges during the detection are:

1. Dealing with (varying) ambient light interference
2. Varying signal intensities directly related to the distance from the transmitter.

This thesis implements the same techniques to solve these issues as the reference paper [48]: a dual-sensor approach. The novelty added by this thesis to this section is the custom lightweight sensor PCB design presented in Section 3.4.2.

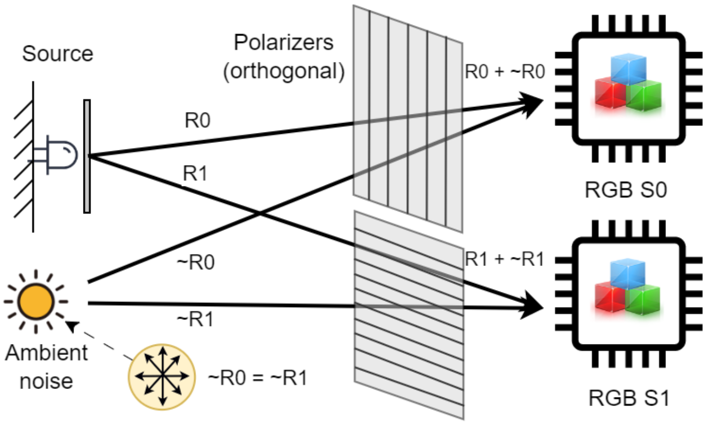


Figure 3.4: **Dual sensor approach where both ambient and source light reach the RGB colour sensors. Note that only the R colour channel is presented in this figure.**

The dual sensor approach works by having two identical colour sensors, each with a separate polariser orientated in perpendicular directions: 0° and 90°. As described in Section 2.1.5 on birefringence, a different colour can be observed by rotating the polariser 90°. Figure 3.5 illustrates this phenomenon where both sensors observe the same tape configuration but observe different colours.

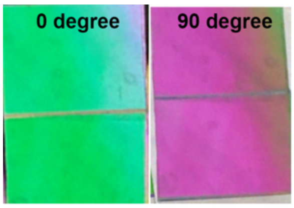


Figure 3.5: **Colours appearing when observing the same tape layers and only rotating the polariser by** $90°$**.**

In order to remove the ambient noise, the following assumption is taken: the recieved ambient light noise is unpolarised, containing an equal mixture of polarised light in all polarisation directions. Because of this assumption, the ambient light noise received by the colour sensors is the same for each sensor despite the different polarisation directions. The ambient light noise is then removed by taking the difference between two simultaneous sensor measurements. Allowing the differential colour values to be written as:

$$\begin{aligned}\Delta R &= (R_0 - \tilde{R}_0) - (R_1 - \tilde{R}_1)\\ \Delta G &= (G_0 - \tilde{G}_0) - (G_1 - \tilde{G}_1)\\ \Delta B &= (B_0 - \tilde{B}_0) - (B_1 - \tilde{B}_1)\end{aligned} \tag{3.1}$$

Here R_0, G_0, B_0 and R_1, G_1, B_1 are measurements that indicate the Red, Green, and Blue light received by sensors 0 and 1 respectively. They are different and won't cancel out due to the different polariser orientations. The $\tilde{R}_0, \tilde{G}_0, \tilde{B}_0$ and $\tilde{R}_1, \tilde{G}_1, \tilde{B}_1$ is the ambient Red, Green, Blue light noise received by both sensors 0 and 1. These do cancel out as they are the same. Figure 3.4 further illustrates this principle. Note that for the assumption of ambient light to hold, both sensors have to be placed as close as possible to each other.

The second remaining problem is the varying light intensity. This problem can be solved by performing a transformation from the RGB to the *hue*, *saturation*, *value* (HSV) domain as seen in the reference paper [48] and illustrated in Figure Fig3.6. In this domain, the *value* is directly related to the received intensity in the RGB domain. Hence *hue* and *saturation* contain all the received colour information. Note that because of the dual sensors approach, the obtained colours are in the form of Δ RGB values and a slight change is required in the ΔRGB to ΔHSV domain specifically for the *saturation* transform given in Equation 3.2.

$$\Delta S = \frac{max(\Delta R, \Delta G, \Delta B) - min(\Delta R, \Delta G, \Delta B)}{2 \times max(|\Delta R|\,|\Delta G|\,|\Delta B|)} \tag{3.2}$$

The new calculation guarantees a positive value within $[0, 1]$, and preserves the orthogonality with hue and value. Using the RGB to HSV transform, whenever the system takes a dual sensor ΔRGB colour measurement, results in a ΔHSV measurement. Appendix A provides the equations for the complete ΔRGB to ΔHSV transform.

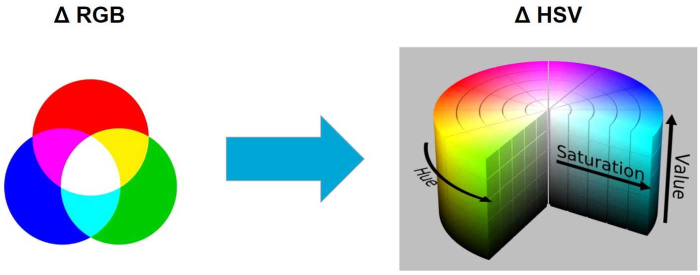


Figure 3.6: **Illustration of the colour domain transform between RGB and HSV [56].**

3.3.2 Colour selection

Since only a discrete number of colours can be obtained using tape, the aim is to find colours with high contrast such that they are easily differentiable while also obtaining a sufficient number of colours to generate a polarisation grid using the algorithm explained in Section 3.2.

The first step is the selection of tape material to use. Figure 3.7a shows the three different tapes that were analysed: 1) basic clear no-brand office tape, 2) basic white opaque office tape, and 3) cellulose tape. The first two are readily available in any office supply store. The first tape is the same as the reference paper, and the third tape is chosen based on a paper analysing birefringence where the tape of choice was cellulose tape [12]. Figure 3.7b shows a side-by-side comparison of the different colours that are obtained by stacking different tape layers in either vertical or horizontal orientation on a thin isotropic transparent backplane. Tape 1 shows the most promising results, whereas tape 3 lacks contrast. Finally, tape 2 is ruled out as stacking multiple opaque layers reduced the ability of light to pass through the material. This analysis results in this thesis using the same tape material as the reference paper.

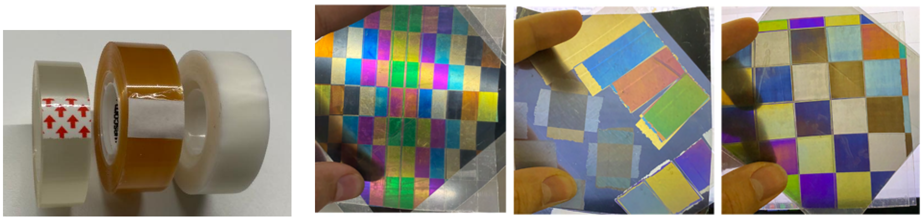

(a) (b)

Figure 3.7: **(a) The three different tape materials analysed. From left to right: 1,3,2. (b) Initial contrast comparison stacking layers of tape in various orientations From left to right tape material: 1, 2 and 3.**

With tape material 1, a total of 11 different tape orientations are proposed. While keeping the backplane material in the same orientation, the layers are stacked up to seven thick in the vertical direction and up to four horizontally. Note that the orientation of the tape layers is with respect to the backplane. Table 3.2 provides an overview of the different tape configurations used. These tape configurations will be referred to as the *Colour ID* throughout this thesis. Figure 3.8 provides a snapshot of some of the colours as seen when observing the tape configurations through a linear polariser.

Table 3.2: **Colour ID and corresponding tape configuration, where ← indicates horizontal layering, and ↑ indicates vertical layering.**

ID	0	1	2	3	4	5	6	7	8	9	10	11	Amb
Layers	0	←1	←2	←3	←4	←5	←6	←7	↑ 1	↑ 2	↑ 3	↑ 4	nothing

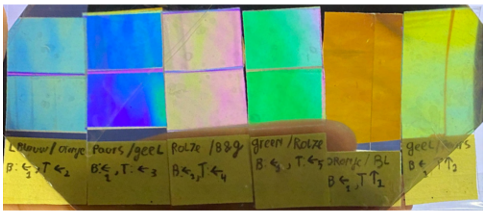


Figure 3.8: **A snapshot of several different tape orientations viewed through the secondary polariser.**

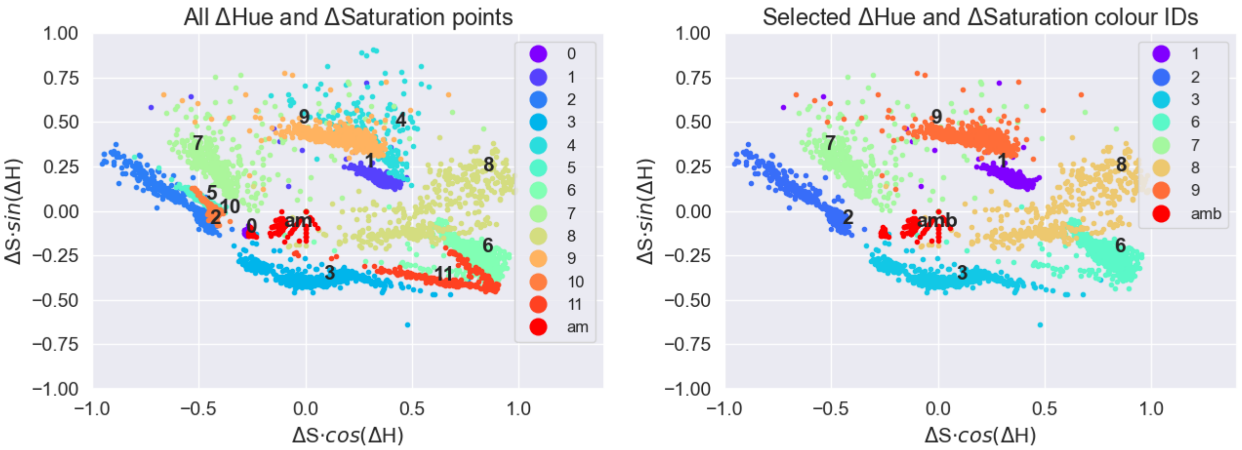
In order to analyse the different tape configurations, sample measurements are taken with the dual sensor design explained in Section 3.3.1. The receiver is facing the transmitter and moved between 0.5 and 2.5 meters from the source. In addition, the receiver is rotating the sensor face a maximum of 15° away from the source, mimicking all sorts of possible drone movements. The dual sensor raw RGB data is converted to the ΔHSV plane. From this point, only the Δhue and Δsaturation values are used, plotting these values on a cartesian coordinates system results in Figure: 3.9a. Every tape configuration from Table 3.2 forms a unique cluster where the distance between clusters is directly related to the colour contrast. In addition, some clusters are isolated, see IDs 7 and 8, and others overlap, see IDs 2,5 and 10. Some are compact, see IDs 1 and 2. Others have a larger spread, see ID 4.

We can ensure the highest probability of successful detection by picking non-overlapping and high-separation clusters. This results in Figure 3.9b. Resulting in a total of seven unique ΔHS colours to work with. This is more colours than the reference paper as they use five colours. However, they do not provide any details on why they chose to use only five colours or if using more colours is also optional in their implementation.

Note that the *amb* cluster is present in both figures. This cluster is light measured when: the source is off, the source is on with no birefringent film present, and finally, when only the transparent backplane is present. It is not used as a colour but aimed to be a reference for the drone to detect if it were to fly outside the projection or is stuck on a boundary-crossing between colours.

3.3.3 Colour classification

Currently, the data consists of labelled data points that belong to several ΔHS colour clusters. This section aims to build an algorithm to predict the colour ID



(a)

(b)

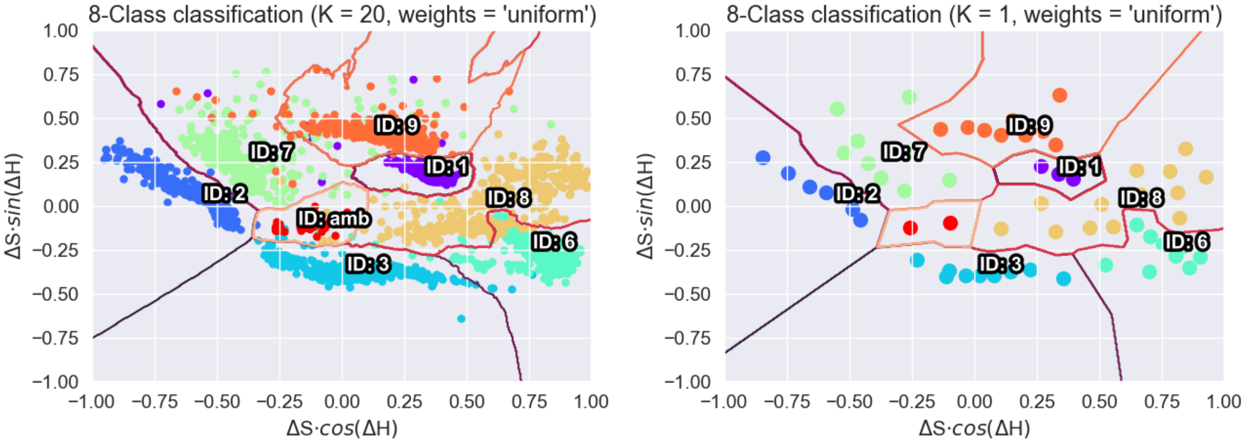
Figure 3.9: **(a) All ΔHue and ΔSaturation colour IDs in Carthesian coordinates. (b) Selected ΔHue and ΔSaturation colour IDs in Carthesian coordinates.**

of newly received unlabeled ΔHSV measurements. This classification has to be performed in real-time on the drone making it essential that this can be done fast using only the limited embedded computing power available.

The reference paper approaches this problem by creating a model of the incoming light using Jones calculus, and then they use system measurements and gradient descent to fit the model parameters to the setup. This thesis takes a different approach since the data obtained is nicely separated into identifiable clusters. This separation allows for the use of Machine learning (ML) to classify the data. Instead of using the Jones calculus model, an ML algorithm builds a model based on sample data, also known as training data. After training the model, it can make predictions without being explicitly programmed.

The colour detection problem is a labelled data-set attempting to solve a multiclass classification problem with supervised learning. The following promising ML algorithms are further analysed: the Multi-Layer Perceptrons (MLP), K-Nearest Neighbours (K-NN), Support Vector Classifiers (SVC) and the Decision Tree (DT).

- **MLP:** Used in Deep Learning, a MLP has an input and output layer with in between one or more hidden layers containing many neurons stacked together. A library called TensorFlow Lite exists to run pre-trained neural networks on an embedded platform. The problem is that the used drone, the Crazyflie 2.1, has limited memory left, forcing the network to fit in 9 kb of memory. This is possible [10]. However, in conjunction with a particle filter would cause memory issues requiring a different approach.
- **K-NN:** This method uses proximity to classify an individual data point. Observing the K closest neighbouring data points and taking the majority classification ID. The downside is the compute-heavy scaling, where the



(a)

(b)

Figure 3.10: **(a) 20-NN colour classification decision boundary of all points. (b) 1-NN colour classification decision boundary of points after prototype reduction.**

distance to all data points has to be determined for every classification [26].

- **DT:** The goal of a DT is to create a model that predicts the classification by learning simple decision rules. A DT can be seen as a piecewise constant approximation [1]. The problem is the shape complexity of the clusters observed in Figure 3.9b results in a tree sizing over 250 branches.
- **SVC:** This method works by finding an optimal decision boundary that separates data points into different classes. SVMs maximise the margin between the decision boundary and the nearest data points [1]. The SVM is a promising solution. However, there exists no easily implementable multiclass classification embedded solution. Designing one would be considered out of the scope of this project.

The K-NN method is deemed the best move forward. It allows for high accuracy when sufficient training data is presented. The algorithm is simple to implement, removing the need for complicated embedded libraries. The only downside is the heavy computing load on the embedded platform. To counter this problem several mitigations are implemented in the next section. The best configuration with clean decision boundaries is obtained using 20 neighbours (20-NN). Figure 3.10a provides this 20-NN implementation. This configuration forms a good balance between very jagged boundaries ($k < 20$) and too smooth boundaries ($k > 20$).

3.3.4 Prototype reduction

As mentioned in the previous section, the K-NN method requires some work to optimise memory usage and computational complexity. Storing the 3500+ floating-point training set in memory would require more than the available 9KB of memory. Additionaly the requirement of computing the distance so all training points would unnecessary strain the system. Hence the training data set must be reduced while preserving the decision boundaries.

A method frequently used to achieve this is *Prototype reduction.* It attempts to reduce the training data by only preserving relevant prototypes (training points) and removing all other non-relevant points. This smaller set of prototypes can consist of existing data points (condensing), newly added data points placing them based on the existing data (addition) or a hybrid form of both. This work by Garcia [22] provides an in-depth overview of the many condensing, addition and hybrid techniques researchers have developed.

In this thesis, K-Means clustering is utilised. According to Scikit-learn [1] "The K-Means algorithm clusters data by trying to separate samples in n groups of equal variance, minimising a criterion known as the inertia or within-cluster sum-of-squares". Reducing the data to 60 clusters and matching the clusters to the correct colour ID results in Figure 3.10b. Observe that the decision boundary is well preserved, especially in regions of high training point density. The 60 points are easily stored in the memory and allow for rapid real-time classification using K-NN.

3.3.5 Drone localisation

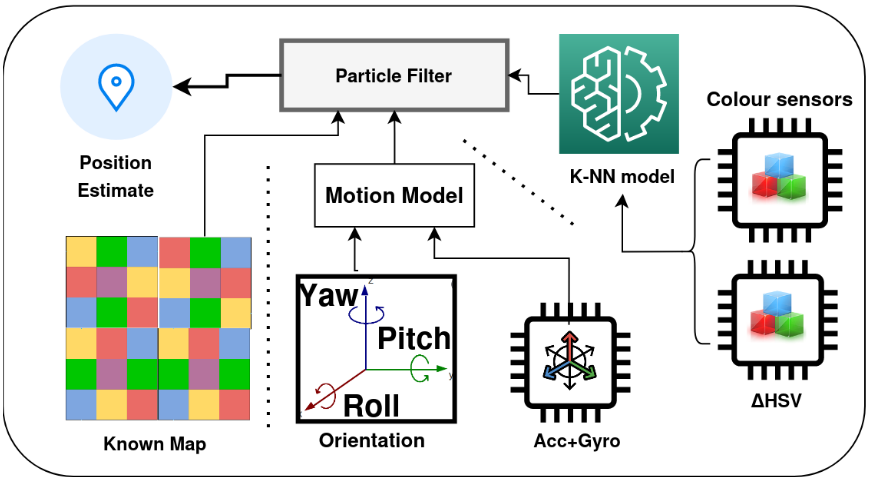


Figure 3.11: **Overview of all parts required by the particle filter to work.**

Now that the receiver detects the correct colour ID independent of the receiver distance and ambient noise, the next step is to use this information to make an accurate position estimate.

Position estimation by fusing the colour data with the IMU measurements is part of the statistical branch: Estimation theory [55]. This branch provides several theoretical frameworks in which values are estimated based on (noisy) empirical measurements containing a random component. Kalman and Particle filters (Sequential Monte Carlo) [6] are prominent examples from this field. There are several implementations where both Kalman and particle filter techniques are used for localisation. Bitcraze, the developers of the Crazyflie drone used in this thesis, provide an outstanding example implementation of the Kalman filter. Their framework fuses the sensor information from the IMU and several other sensor setups, such as the Flow deck or the Lighthouse Deck, to obtain a position estimate with cm level accuracy [14].

The problem with implementing a Kalman filter is that it requires a Gaussian probability distribution of the input and output state estimate. Because of the repeating colours in the polarisation pattern, a non-Gaussian probability distribution for the output position estimate is obtained. Every colour projected into space matching the colour detected by the colour sensor contains an equal probability of being correct. Hence a filter that can handle non-Gaussian probability distributions is required. One such example is the Particle filter. This is also the type of filter used by the reference paper. However, a particle filter is very compute intensive, and for it to work in this thesis, some novel computational optimisations must be applied to the design.

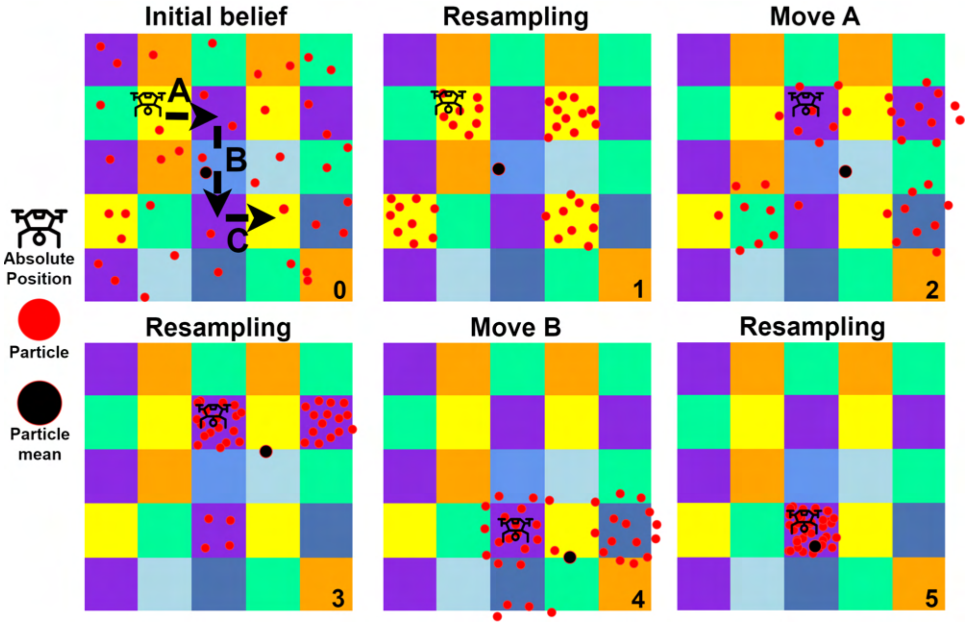


Figure 3.12: **Particle filter progress over time.**

3.3.6 The particle filter

A particle filter works by placing a set of particles on a known map of the environment. Here each particle represents a possible state of the system. E.g. the actual location of the drone. When combined, they estimate the probability distribution of the system state. A particle filter then works by alternating between 2 states.

1. The resampling state: Every particle obtains a new position based on the current probability distribution of the location.
2. The motion state: The particles advance to a new location based on a motion model created with IMU sensor information. More on the motion model later.

Before switching back to the resampling state, a new probability distribution is generated with the new position information of all particles and the observed colour information. The next resampling cycle then uses this updated distribution.

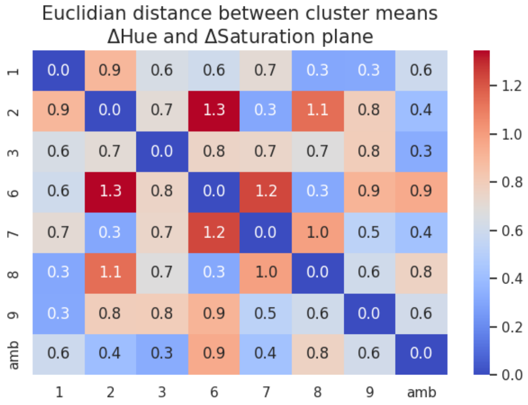
To illustrate the 2 alternating steps of the particle filter Figure 3.12 provides an example. In this example, the drone follows a path within a known colour map. At first, an initial belief of particles is uniformly distributed over the complete map meaning that the drone can be anywhere. While following the path [A, B, C], the motion state and re-sampling state keep alternating, discarding particles at wrong colour locations and resulting in the particles converging on the absolute position of the drone.

In this thesis, such a particle filter is implemented in real-time on the drone. However, for it to work, several components have to be provided. First, a known map is required. Additionally, the detected colour is needed. These two sources are used by the particle filter resampling state. Secondly, a motion model is needed. This motion model attempts to model the motion of the drone based on the IMU data and is used in the motion state. Furthermore, to run the particle filter, some computational optimisations were made to run a larger number of particles in real-time on an embedded platform. Figure 3.11 provides an overview of all components that work together in this particle filter to obtain the position estimate of the drone. The following sections will provide information regarding these components.

3.3.7 Grid design and creation

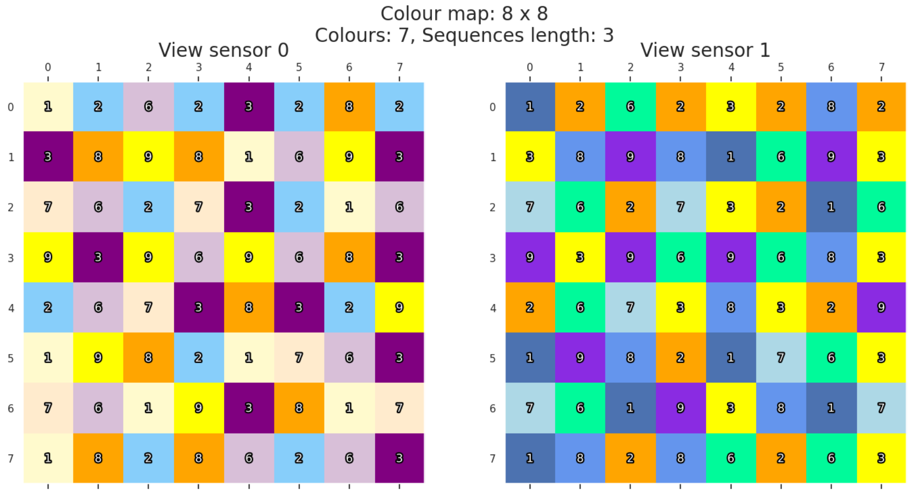
Sections 3.3.1 and 3.3.2 resulted in seven uniquely identifiable colours. With these seven colours, we create the localisation map. This map follows the algorithm designed in Section 3.2. The first step is determining the contrast between the different colours. This difference is specified using the function $D(L_1, L_2)$ that evaluates the Euclidean distance on the ΔHS plane. Calculating this distance between the means of every colour cluster results in Figure 3.13. Using the Euclidian distances, the following 8×8 map in Figure 3.14 is created. Note that both sensors 0 and 1 observe different colours due to the $90\circ$ rotation in polariser orientation. This results in two different colour representations as observed by sensors 0 and 1 of the same polarisation grid.

This thesis innovates on the reference work [48] by developing techniques for manufacturing a more fine-grained grid design. The cell sizes are only 7.5×7.5 mm compared to the original 10 mm. This smaller size is achieved using a specialised tape-cutting jig. Note that smaller grid sizes up to 5 mm are also possible. For the polarisation projection, the 8 by 8 grid presented in Figure 3.3 is manufactured. This results in a total grid size of 60×60 mm. The cells are glued to a 3D-printed mesh backplane that provides structural stability and



	1	2	3	6	7	8	9	amb
1	0.0	0.9	0.6	0.6	0.7	0.3	0.3	0.6
2	0.9	0.0	0.7	1.3	0.3	1.1	0.8	0.4
3	0.6	0.7	0.0	0.8	0.7	0.7	0.8	0.3
6	0.6	1.3	0.8	0.0	1.2	0.3	0.9	0.9
7	0.7	0.3	0.7	1.2	0.0	1.0	0.5	0.4
8	0.3	1.1	0.7	0.3	1.0	0.0	0.6	0.8
9	0.3	0.8	0.8	0.9	0.5	0.6	0.0	0.6
amb	0.6	0.4	0.3	0.9	0.4	0.8	0.6	0.0

Figure 3.13: **Euchlidian distance between the means of every colour cluster on the ΔHS plane.**



	0	1	2	3	4	5	6	7
0	1	2	6	2	3	2	8	2
1	3	8	9	8	1	6	9	3
2	7	6	2	7	3	2	1	6
3	9	3	9	6	9	6	8	3
4	2	6	7	3	8	3	2	9
5	1	9	8	2	1	7	6	3
6	7	6	1	9	3	8	1	7
7	1	8	2	8	6	2	6	3

	0	1	2	3	4	5	6	7
0	1	2	6	2	3	2	8	2
1	3	8	9	8	1	6	9	3
2	7	6	2	7	3	2	1	6
3	9	3	9	6	9	6	8	3
4	2	6	7	3	8	3	2	9
5	1	9	8	2	1	7	6	3
6	7	6	1	9	3	8	1	7
7	1	8	2	8	6	2	6	3

Figure 3.14: **Map showing the different colours observed by the drone.**

hides edge imperfections while minimising grid obstruction. Figure 3.15 provides an image of the backplane, square cutter and grid in the manufacturing phase. An analysis of the colour classification and localisation performance with the resulting grid is provided in Chapter 6.

3.3.8 Motion model

While using a particle filter for localisation, the filter alternates between the resampling and motion states. During the motion state, the particles are allowed to move. The goal is that particles should precisely mimic the drone's motion. This mimicking is approximated using a motion model. The motion model uses the IMU's rapid observations to estimate the system's state, e.g. the current velocity and position. The reference paper [48] estimated the evolution of the

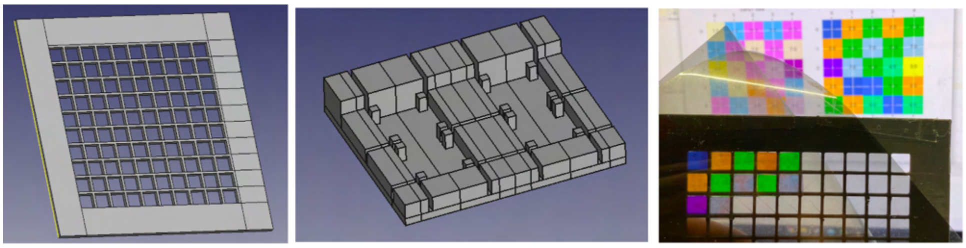


Figure 3.15: **(a)The grid backplane, (b) Square tape cutter, (c) Final grid creation.**

motion model using the following set of equations:

$$\boldsymbol{v}_k = \boldsymbol{v}_{k-1} + \boldsymbol{a_k} \cdot \Delta t + \boldsymbol{n}_k^{\boldsymbol{v}}$$
$$\boldsymbol{x}_k = \boldsymbol{x}_{k-1} + \frac{1}{2}(\boldsymbol{v}_{k-1} + \boldsymbol{v}_k)\Delta t + \boldsymbol{n}_k^{\boldsymbol{x}} \tag{3.3}$$

Here $\boldsymbol{v}_k$ is the velocity vector for x, y, z in the global reference frame at time step k, $\boldsymbol{x}_k$ is the position for x, y, z in the global reference frame at time step k, and $\boldsymbol{n}_k^{\boldsymbol{v}}$ and $\boldsymbol{n}_k^{\boldsymbol{x}}$ are normally distributed process noises. It is necessary to realise that every particle contains such a model and that after every IMU update, all particles have to be updated. If sufficient computational overhead is available, this is not an issue. However, in a limited embedded environment, this is an issue. This thesis proposes the following design changes to relieve some of the computational requirements of the motion model, allowing for a greater number of particles.

The first novel addition is to split the particles into two types of particles:

1. Motion particles.
2. Regular particles.

Of the complete set of particles, there is only one motion particle, and all other particles are regular particles (Figure 3.16). The main difference is that the motion particle contains a complete motion model, while the regular particles only know their current position. The motion particle accumulates the rapid IMU updates and estimates the change in velocity and position over a time window T, e.g. in 200 ms, the IMU measurements accumulate, and the motion model estimates that the drone's position has changed by 0.1 m. After T time, this accumulated position change and some additional Gaussian noise transfers

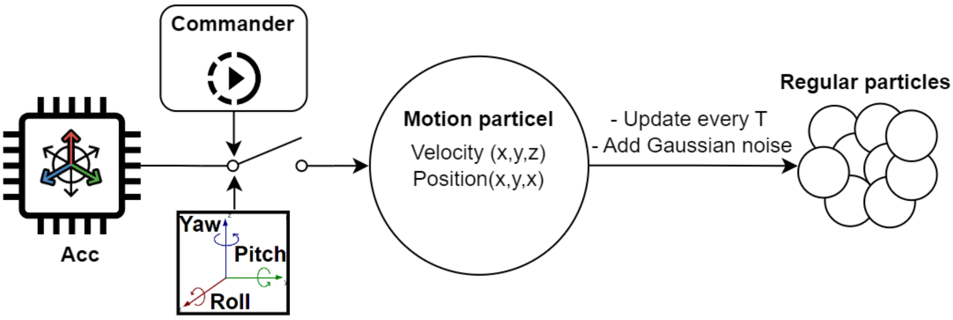


Figure 3.16: **Optimised motion model and novel additions.**

to the regular particles. This process is called the transfer step and happens multiple times during the motion state.

Adding Gaussian noise during the transfer step represents the second change to the motion model. Previously this noise was added during every IMU update to all particles. This approach no longer works with the accumulation approach. Hence the noise is added at the transfer step.

The above changes represented in Figure 3.16 reduce the number of calculations required for the motion model during every IMU time step from all N regular particles to a single motion particle. Additionally, the simplifications reduce the Gaussian noise generation for all N normal particles during every IMU time step to only at the end of the time window T. This all brings the computation steps down from approximately 400 calculations to just 2 for every IMU cycle. The downside is that the noise is accumulated at the transfer step, losing some of the accuracy in the system.

The following set of equations describes the updated motion model contained by the motion particle:

$$\boldsymbol{v}_k = \boldsymbol{v}_{k-1} + \boldsymbol{a_k} \cdot \Delta t$$

$$\boldsymbol{x}_k = \boldsymbol{x}_{k-1} + \frac{1}{2}(\boldsymbol{v}_{k-1} + \boldsymbol{v}_k)\Delta t \tag{3.4}$$

With the two changes implemented, observe that the system behaves like the reference paper implementation of the motion mode by setting T to the time window of a single IMU sample.

Motion model additions

The current implementation of the motion model only uses information obtained from the accelerometer. As explained before, these measurements suffer from noise and integration drift. In case of the Crazyflie drone, this is so much drift that all particles would diverge rapidly and cause the particle filter to never converge. E.g. speeds of more than 10 m/s within several seconds. This is not the case in the reference paper, as a higher-quality IMU is used. However, in order to reduce this rapid drift for this drone implementation, the motion model is expanded to incorporate more information. Two promising additions are:

- Yaw, Pitch and Roll estimates in the global reference frame. In a drone context, Yaw, Pitch and Roll indicate if motion is about to start or stop. E.g. stopping a drone from moving along the Y axis requires a rolling motion. This rolling motion provides an additional indicator beside the accelerometer that the motion has stopped.
- The received flight command. Just like the previous point, it provides a clear indication of the start of a movement.

By fusing these two indicators into the motion model, the system can disable the motion model when no movement is expected. E.g. the drone is not receiving a move command, nor does the roll axis detect anything means effectively to keep all particles in the same spot only allowing them to move when resampling to a different colour. This addition prevents the particles from drifting away uncontrollably and allows the model to converge to the correct location. Figure 3.16 provides an overview of all elements involved in the motion model. The evaluation Chapter 6 offers more information on the performance of this motion model during a flight of the Crazyflie 2.1.

3.4 Transmitter design and implementation

The implementation process uses various materials and components listed in detail in the Bill of Materials (BOM) provided in Appendix B. This approach ensures clarity and avoids unnecessary clutter in the main body of text.

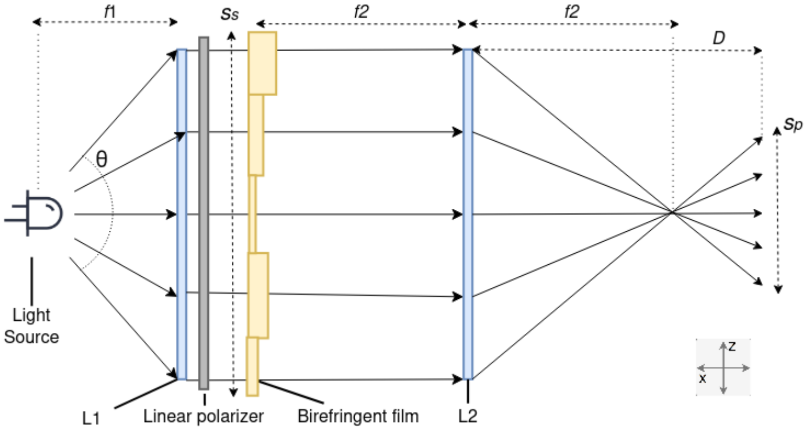


Figure 3.17: **Projection configuration.**

The transmitter aims to project a sideways pyramid with a uniform intensity polarisation pattern into space. Figure 3.1 provides an example of such a sideways pyramid. The projector contains three major components: The light source, the lenses, and the polarisation pattern designed in Section 3.3.7. The alignment of the included components is illustrated in Figure 3.17. First, L_1 collimates the light into a uniform beam. The light then passes through a linear polariser and the birefringent film containing the polarisation pattern. Lens L_2 then determines how fast the light diverges to form the pyramid projection. The divergence rate follows the mathematical relation:

$$S_p = \frac{D - f_2}{f_2} \cdot S_s \tag{3.5}$$

Here D is the distance from L_2, and f_2 is the focal length of L_2. S_p is the projected grid size, and S_s is the starting size. For example, a small focal point for f_2 results in a fast diverging projection. Resulting in a wide pyramid in the Z and Y axis but a limitation in accuracy over longer distances caused by the rapid growth of the polarisation pattern.

Regarding the correct component choice in this projector design, the critical factors that decide most dimensions are:

- The polarisation pattern grid size S_s of 60 × 60 mm sets the minimum diameter size for the lenses to be at least the size of the diagonal length of the polarisation grid. This size is required to project the complete grid from corner to corner. It results in a minimum lens diameter of $\sqrt{60^2 + 60^2} = 84.85$ mm.
- The LED beam angle Θ combined with focal point f_1 set the upper limit to the polarisation grid size S_s. They follow the relation given in Equation 3.6

$$S_s = 2 \cdot tan(\frac{\Theta}{2}) \cdot f_1 \tag{3.6}$$

The beam angle Θ of an LED source determines the width and distribution of the emitted light. It indicates the angle at which the intensity drops to 50% of its maximum. Staying well within the beam angle ensures minimal intensity variation in the collimated beam of light. Providing minimal intensity differences across the grid projection. The system can deal with the remaining intensity variation because of the RGB to HSV domain transformation explained in Section 3.3.1.

Lens choice

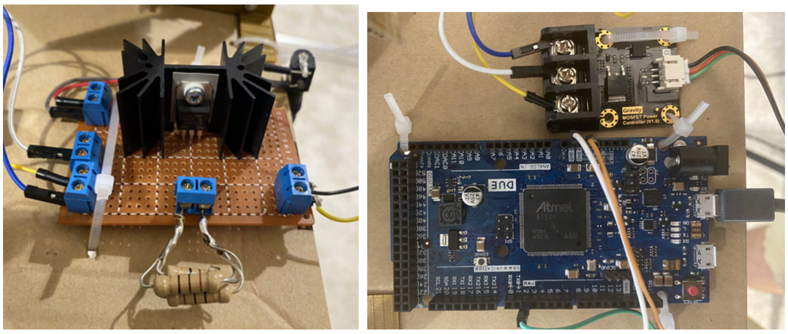
The two main parameters required in picking a lens are the lens sizes and the focal points of both lenses L_1 and L_2. The polarisation grid's size sets the lens size lower bound of the diameter to be at least $84.85mm$. The divergence rate of the projected pyramid is set by the focal point of the L_2 lens. Similar to the reference paper, this thesis aims to achieve a cell size of approximately 200mm at a 2.5-meter distance, providing a good balance between accuracy and range. Following 3.5, a focal point of approximately 90mm is required for lens f_2. The resulting lenses for L_1 and L_2 are 100mm square plano-convex Fresnel lenses with a focal length of 90mm. The use of Fresnel lenses significantly reduces both the cost and weight of the system. While causing only minimal distortions in the final projection.

Light Source

The goal of the light source is to provide sufficient illumination to project the polarisation pattern into space. Based on this, the following requirements are set:

- Minimum projection distance of 2.5m, similar to the reference work [48].
- High luminosity, the polarisers absorb at least half of the emitted light.
- Uniform intensity across the polarisation pattern.
- Rapid power switching capabilities to enable communication.
- Eye safe.

Additionally, with a diagonal grid size of 84.85mm, and a focal point of 90mm the beam angle Θ must be wider than $tan^{-1}(\frac{84.85mm}{2 \cdot 90mm}) = 50,34°$. Based on the requirements, a 1 Watt wide-angle LED is chosen. A constant current (CC)



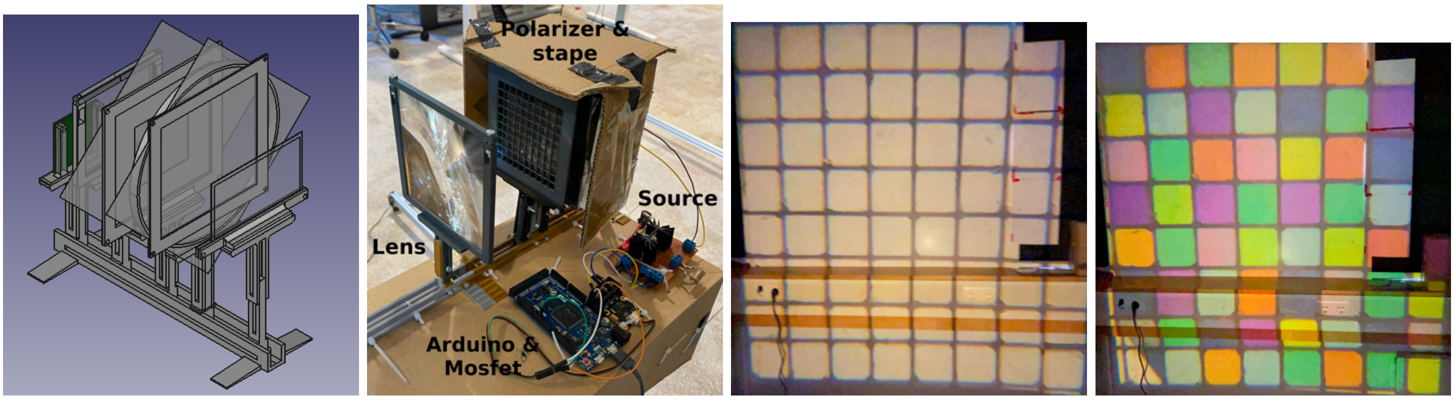
(a) (b)

Figure 3.18: **(a) Constant current LED source. (b) Arduino setup and MOSFET switch.**

source drives this LED to provide a constant luminosity (Figure 3.18a). Limiting the LED at 1W makes the operating range after the polarisers larger than 2.5 meters. In addition, the intensity is still within eye-safe limits. The wide angle ensures complete coverage of the first lens and the polarisation pattern.

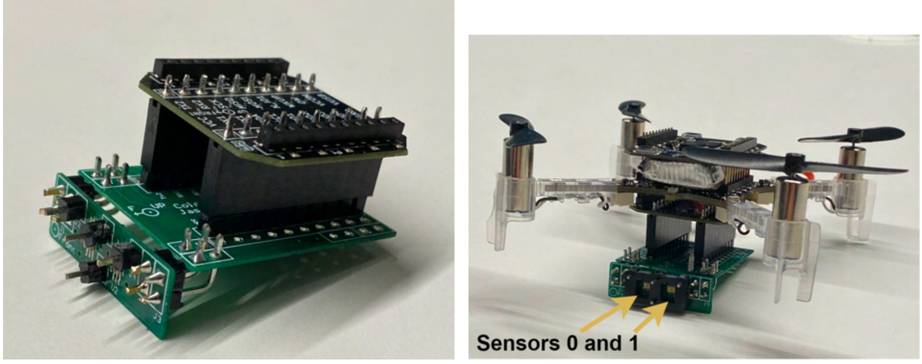
3.4.1 Final transmitter setup

Finally, the setup is realised using a 3D printed setup designed in computer aided design (CAD) software that holds all elements at the correct height and distance. Figure 3.19 shows both the CAD and actual implementations. Including the resulting projection with and without a secondary polariser present at the source. The secondary polariser allows the colours to be visible when projected. The grid lines in between colours originate from the backplane. In an ideal situation, the grid would be self-supporting, requiring no backplane.



(a) (b) (c) (d)

Figure 3.19: **(a) 3D model. (b) Projector hardware setup. (C) Grid without secondary polariser. (d) Grid with secondary polariser at projector side.**



(a) (b)

Figure 3.20: **(a) Sensor side view. (b) Sensor mounted under drone.**

3.4.2 Receiver design implementation

The dual-sensor ΔHSV Colour design proposed in the design Section 3.3.1 solves two important sensing issues:

- Ambient noise from sources other than the transmitter.
- Light intensity variations.

However, the critical assumption that the dual-sensor approach makes is that both sensors receive (almost) the same light. To approximate this state, the colour sensors must be placed in very close proximity to each other. This is where this thesis further elaborates on the work in the reference paper [48]. Instead of placing two prototype boards approximately 3 centimetres apart, this work proposes a custom PCB that places the colour sensors 0.5 cm apart. See Figure 3.20.

The sensors require I2C to communicate with the Crazyflie 2.1. However, due to an I2C address clash, the TCA9548a I2C Multiplexer is integrated to orchestrate the communication protocol. Atop the colour sensors, a 3D-printed clamp holds the two polarisers in place and provides shielding against light pollution from the sides and cross-contamination between the sensors. The PCB design allows for direct mounting underneath the drone using friction pins, and due to the PCB layout, the sensors are mounted at a right angle with respect to the bottom of the drone. This layout allows the colour sensor detection faces to always point towards the transmitter in normal flight conditions. Provided in the Appendix C are the schematic and PCB files of both the transmitter and receiver.

Chapter 4

Communication link integration

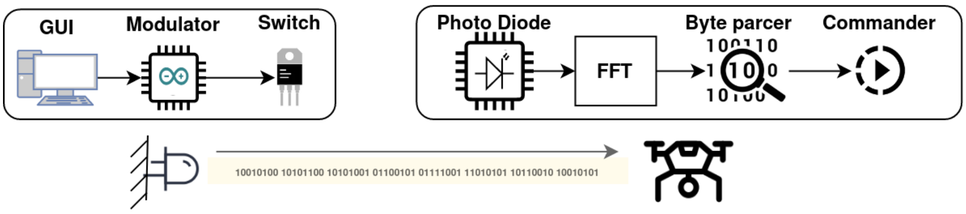


Figure 4.1: **Schematic design of the VLC communication link.**

A drone that can localise itself and only hover in one spot is impractical. For it to become useful, it requires communication with a base station to receive commands. Hence the next logical step is to expand the current system with a communication link.

There are countless ways to establish a communication link with a drone, many of which utilise the radio frequency (RF) spectrum. The Background 2.2 goes into great depth, exploring the many options available. The problem with using, for example, the RF spectrum is that dedicated communication hardware is required on both the transmitter and receiver side, making the system more complex. This asks the question, is it possible to use the existing projection-based localisation infrastructure from the last chapter to establish a communication link?

For this to be possible, the link must use the light projection infrastructure already in the system. Hence the communication link must use some form of Visible Light Communication (VLC) technique (see Section 2.3). For such a VLC addition to work, the following requirements have to hold:

- A source capable of rapid on-off switching.
- No interference with the localisation mechanisms. E.g. the rapid on-off switching while transmitting should not have any adverse effect on the

localisation accuracy.

- The link should be robust against dynamic ambient light changes similar to the RGB sensors.
- The working range should be the same or larger than the localisation range. Currently set to 2.5 meters.
- The receiver needs to be capable of rapid sampling. At least 2000 Hz, the ADC sampling rate of the Crazyflie 2.1 drone.

A nice benefit of the static polarisation pattern used in the reference paper's polarisation-based localisation technique is that the light source is independent of the polarisation pattern projected into space. This independence means that various light sources can be used without reducing localisation accuracy. Therefore, source independence allows for the use of rapid-switching LED-based light sources and fulfils the first requirement.

Because the system uses the HSV colour domain approach explained in Section 3.3.1, it has an inherent resistance against varying light intensity. Hence the reduced intensity while sending information should have minimal to no effect on the localisation accuracy and satisfies the second requirement.

4.1 Modulation techniques

In order to send information on a VLC Link, it needs to be encoded. This encoding can be achieved in numerous ways, each having different properties that work best in specific scenarios. In this scenario, the third requirement requires the VLC link to be robust against varying ambient light conditions. To attempt to meet this requirement, a collection of the different encoding techniques used in VLC links are summarised into four different categories:

- Pulse-based modulation: This method involves varying the intensity of the light source by turning it on and off rapidly to create a binary code.
- Frequency domain modulation: involves modulating the frequency of the light signal to encode information.
- Continous time: involves multi-source modulation schemes.
- Colour domain: transmits data through the variation of the colours emitted at the source.

Figure 4.2 provides a tree graph showing several existing modulation schemes within each category. Important here is that some techniques require a single source, whereas others require multiple or specialised sources. The techniques that only use a single source of white light are relevant to this thesis. Hence the categories: Pulse-based modulation and frequency domain modulation provide optional solutions. Where techniques like On-Off Keying (OOK), and Frequency Shift Keying (FSK) are frequently implemented examples.

By design, FSK is the most robust against ambient light noise as it does not require detection thresholds unlike OOK, but instead identifies the frequencies in the received light. The variations in ambient light are expected to be around

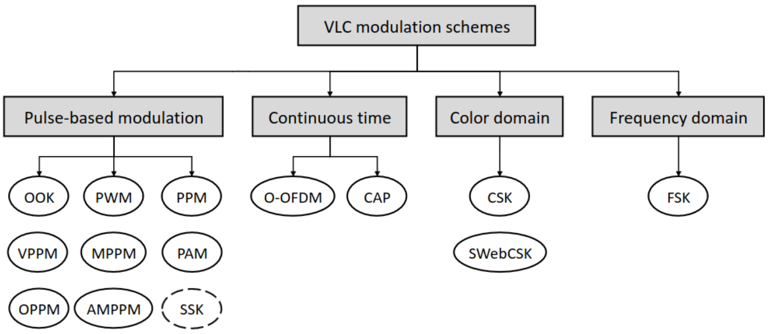


Figure 4.2: **VLC modulation schemes [53].**

0.1 to 50Hz, with sources varying from a cloud moving in front of the sun to 50 Hz AC lightbulbs. This makes FSK the best approach for this VLC link when listening for frequencies higher than these noise source frequencies.

The last two requirements are heavily sensor dependent. It requires finding the balance between sensors that are too sensitive and saturate when too close to the projector vs sensors that are not sensitive enough, and the signal disappears below the noise floor prematurely when distance increases. Finally, the sensors should also have a sampling rate faster than 2000Hz. The original colour sensor can not be used as their sampling rate is not fast enough, only around 2 Hz. Therefore based on empirical results, a sensor that provides the right balance is the OPT101. Furthermore, with minor modifications such as a MOSFET switch to the light source, the LED source can provide the rapid switching required for FSK. This MOSFET is then switched using a dedicated driver circuit controlled by an Arduino Due connected to a PC with a graphical user interface (GUI) controlled by the user. Figure 4.1 provides a schematic overview of the resulting communication design showing all steps from PC to transmitter to receiver. Additionally, Figure 3.19b and Appendix C provide the implemented hardware setup.

4.2 FSK VLC link implementation

FSK works by modulating the carrier frequency between several discrete frequencies. These discrete frequencies represent the digital information. In order to keep the communication link straightforward and robust, the simplest form of FSK: Binary-FSK, is chosen. In Binary-FSK, the carrier signal shifts between two discrete frequencies f_0 and f_1, representing a binary 0 and 1, respectively. E.g. the sequence: $[f_1, f_0, f_1, f_1, f_0, f_0, f_1, f_0] = bin$:10110010. In this case, the carrier signal is a square wave emitting at either f_0 or f_1 and arrives at the receiver as light with oscillating intensity. Then the receiver samples this light, and using a fast Fourier transform (FFT) the original transmitter frequency is retrieved, obtaining the original binary sequence. Simply sending a sequence of frequencies does not provide a reliable method of sending information. Several additions must be made to the signal such that the receiver knows when the packet starts sending and when to start listening for the bits.

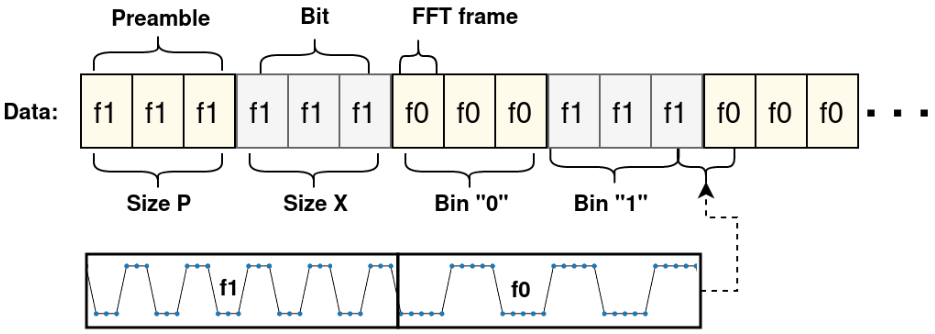


Figure 4.3: **Layout of a single VLC data packet encoded using FSK.**

In order to successfully transmit data, the following communication scheme is proposed. Figure 4.3 shows the layout of a single data packet: every f_0 or f_1 represents a fixed width time window of N samples a ”frame”. Within that frame, a square wave of either frequency is transmitted. When at the receiver side, a fast Fourier transform (FFT) is performed on the frame, the transmitted frequency is recovered.

A packet starts with a simple preamble sending f_1 frames a set P amount of times. The receiver counts until P frames have been received. At that point, the sending of the data packet starts where every bit is represented by X consecutive frames. At the receiver, these frames are decoded using the FFT, and a majority vote is performed on the X consecutive received frequencies.

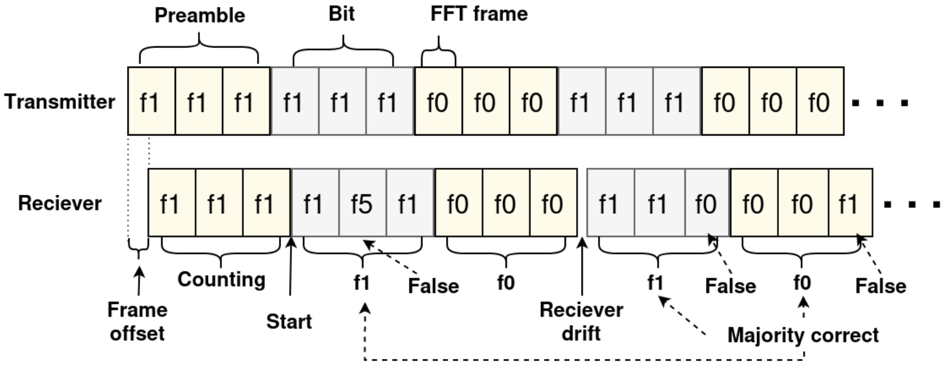


Figure 4.4: **Illustrating the error correcting properties of this communication protocol**

Figure 4.4 illustrates the reasoning behind sending X frames and picking the majority: When sending information, the sender and receiver are most likely not in sync with sampling their frames. Hence the frame at the receiver side may start with a frame offset. This can be only a quarter frame or, worst-case, almost half of the receiver frame. This frame offset is not an issue since the majority frequency of the received frame will be detected and used in the decoding. E.g. a received frame with a large frame offset consisting of 51% f_0 and 49%

f_1 will result in the detection of f_0. This offset will propagate, and with exact timings, the data is received correctly.

It becomes an issue if a slight drift in either systems is present. This drift may cause the frame to shift over the detection threshold resulting in a false frequency being detected. This would be an issue if a single frame is sent. However as seen in Figure 4.4, sending multiple frames allows for majority voting and catching of the error. Furthermore, if an unknown frequency is detected, say an f_5, this would also be resolved without having to discard the packet. A larger value for X would result in more redundancy but slows down the transmission rate. An analysis of the optimal number of frames required for correct transmission will be provided in Section 6.3 for now it's assumed to be $X = 3$.

4.2.1 VLC hardware implementation:

As mentioned, the projection-based localisation setup developed in the previous chapter is modified by adding an OPT101 photodiode, a switching power MOSFET, and an Arduino Due connected to a PC for controlling the switching behaviour and interfacing with the user. With the hardware known, the parameters mentioned in Section 4.2 are set. Table 4.1 provides an overview of these parameters.

Table 4.1: **The parameters used in the FSK VLC link**

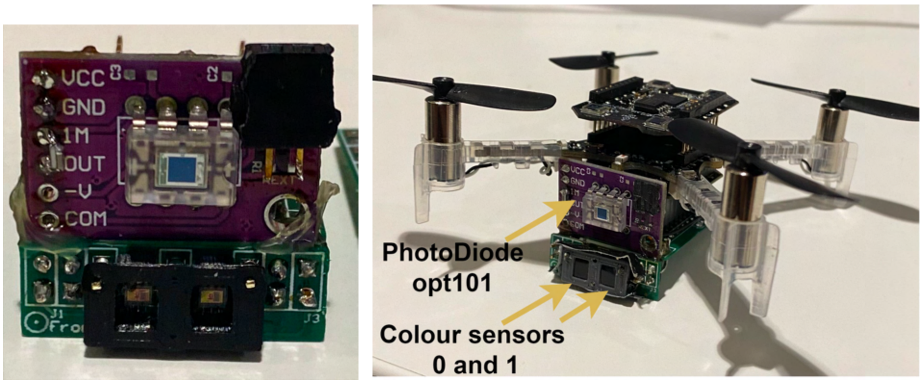
Frequency 0	Frequency 1	Samples per frame	Preamble frames	Bit frames	Sampling rate drone	Bytes per second
$f_0 = 125Hz$	$f_1 = 156Hz$	$N = 64$	$P = 3$	$X = 3$	$f_s = 2000Hz$	$Bps = 1.3$

The f_0 and f_1 are set to 125Hz and 156Hz, respectively. The problem is that frequencies below $300Hz$ are perceived as flickering [16] by humans, causing unwanted interference. Therefore higher frequencies are desired; however, at frequencies higher than $156Hz$, the MOSFET caused unwanted frequency noise. Therefore, the frequencies are set lower. The software supports 16, 64, and 256 samples for each FFT frame. Only frames with the number of samples larger than 64 ($N >= 64$) reliably obtained the frequencies where $N = 256$ takes too long to compute. Finally, The sampling rate f_s is limited by the drone to 2000Hz. Due to the frame size, multi-frame redundancy and "slow" sample rate of the drone, the Bytes per second (Bps) is only 1.3. This is slow compared to other VLC link speeds [8], but it does not require precise alignment of the transmitter and receiver.

In the hardware implementation, the photodiode mounts on the PCB together with the colour sensors to always face the transmitter. Figure 4.5 shows the placement of the sensor. Additionally, appendix C provides the connection diagram for all parts included.

4.2.2 Packet Layout

How information is best transferred between transmitter and receiver is very project specific. However, in general, some form of messaging protocol is used.



(a) (b)

Figure 4.5: **(a) OPT101 mounted on the sensor PCB, (b) Updated sensor mounted under drone.**

In brief, a messaging protocol is a set of rules that governs the communication. An example is the MAVLink [49] communication protocol developed for drone-specific implementations. The problem with MAVLink is the 8 bytes overhead required to send a single message. The current VLC implementation only sends at 1.3 Bps, causing the 8 bytes required by MAVlink to take too long. Hence a custom single-byte communication protocol is developed. The custom protocol enables communication with up to 3 simultaneous drones, supports 32 unique commands and has a single parity check bit. A single byte has the layout presented in table 4.2. The first 2 bytes are the drone ID's, with "$0b11xxxxxx$" reserved for broadcasting to all drones at once. The next 5 bits are commands, followed by a single-bit checksum.

Table 4.2: **The single byte layout used in the FSK VLC link**

ID	Command	Check
XX	XXXXX	X
3 drones	32 commands	1 bit check

Some examples of the commands currently implemented are: Take off, Land, Forward, Backward, Left, Right, Up and Down.

4.2.3 VLC interference

While flying the drone uses a ground truth system the: *Lighthouse Deck* [13]. This system is used as an absolute reference for the localisation algorithm validation. Therefore it is active at all times during a flight. The Lighthouse deck system uses the SteamVR Base Stations. These base stations emit an intense Infra-Red (IR) light when sweeping across the room. This sweep causes the OPT101 photodiode to saturate temporarily. Although this saturation happens only in a very short amount of time, this saturation is enough to cause the FFT to misinterpret the received frequency, severely increasing the chances of a faulty packet to almost 50% of the time. Figure 4.6 shows such an IR saturation event.

Two solutions are proposed:

- **Physical bandpass light filter**: such a filter placed in front of the OPT101 removes the IR spectrum from the received signal. This filter solves the issue but causes the received signal strength from the source to be more than halved. Additionally, the filters available are too heavy and bulky for the Crazyfie 2.1.
- **Software-based solution**: A simple moving average filter detects rapid saturations (spikes) in the received signal. When a spike is detected, the signal is averaged to the signal mean throughout the spike. This averaging is not ideal because it causes other frequency components in the FFT to grow. This can be compensated for by taking a larger number of samples n in a single FFT frame. E.g. the 64 samples instead of the smallest amount of 16 the system is capable of.

Because of the weight constraints, only the second proposal is implemented. The communication link works better than without the software filter although the filtering is not perfect. In Section 6.3, a performance evaluation is provided with the effects of the ground truth system on the communication link. Furthermore, in Chapter 6 an analysis is given of the protocol in different scenarios, analysing range, and the performance of the communication in conjunction with the localisation.

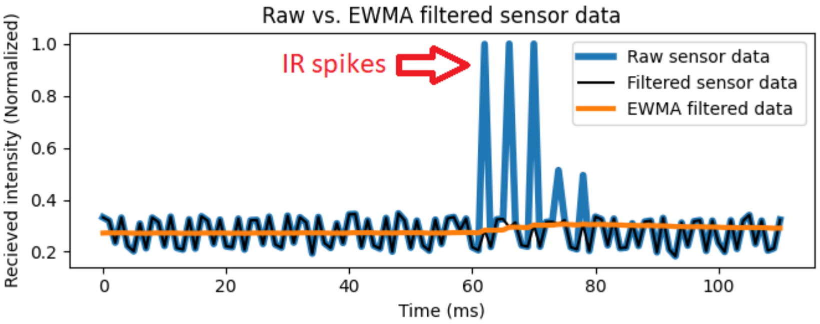


Figure 4.6: **IR interference caused by the Lighthouse ground truth system used for the Crazyflie 2.1. Showing both unfiltered and filtered data.**

Chapter 5

Embedded software considerations

The above sections conclude the hardware implementations of this thesis. Nevertheless, a lot of the work is hidden within the code. Working in a limited compute power environment forces some primary design considerations to be made with computational efficiency in mind. This section aims to illustrate the core structure, dependencies and embedded focused decisions to the reader without going on a line-by-line basis through the code.

Most of the code developed is directly integrated into the Crazyflie embedded firmware. The developers use the real time operating system (RTOS): FreeRTOS to schedule all elements in their code dynamically. Simply put, FreeRTOS is an open-source RTOS designed for embedded systems. It follows a preemptive scheduling model and prioritises tasks based on their assigned priorities. FreeRTOS supports interrupt handling, software timers for scheduling recurring tasks and many more. Despite adding some overhead, FreeRTOS simplifies adding new functionalities down to creating a task and setting its priority.

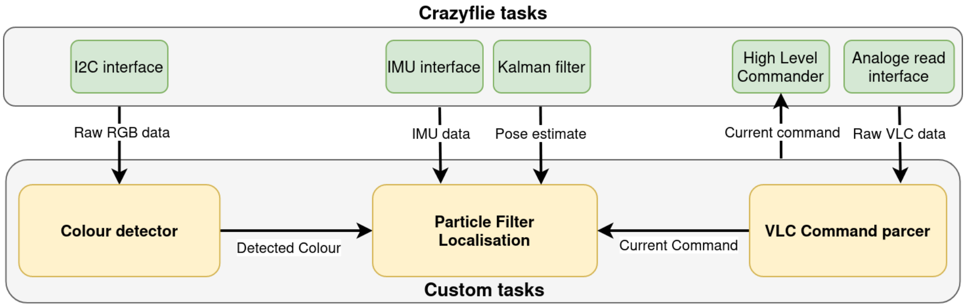


Figure 5.1: **FreeRTOS task interconnection.**

This thesis adds three significant tasks to the Crazyflie firmware: Colour detector, Particle filter, and VLC Command parser. The newly designed tasks are not completely standalone and do require some help from the tasks implemented by the Crazyflie developers. Some examples are the I2C driver or accelerometer data acquisition. Figure 5.1 illustrates the connectivity between the Crazyflie firmware and custom firmware.

All the custom tasks written for this thesis consist primarily of 2 parts: the *tick()* functions and the *update()* functions. The tick functions are time-critical high-priority sections. It interrupts all other code and only runs for a short time. Some example tick function tasks are ADC reads or setting a data ready flag. The update functions run the non-time critical compute-heavy steps like the FFT or a K-NN classification algorithm. The non-critical sections do have the requirement to finish the update function before the next start of the update cycle.

5.1 Colour detection implementation

Figure 5.2 provides the code structure of the colour detection task. Note that the task contains both the tick and update sections running at specific frequencies. The tick only sets data-ready flags and the update section takes care of the I2C read/write, RGB to HSV transforms and the K-NN classification. The primary embedded optimisation in this task is the prototype reduction explained in 3.3.4, reducing the data set from 3500+ to only 60 points.

This task publishes the detected colour based on an averaging algorithm. This is because single false colour detections can severely mess with the particle filters resampling algorithm. The filter uses a circular buffer [28], if the buffer contains the same colour three times, a new colour must be present. This result is then published to the other tasks in the system.

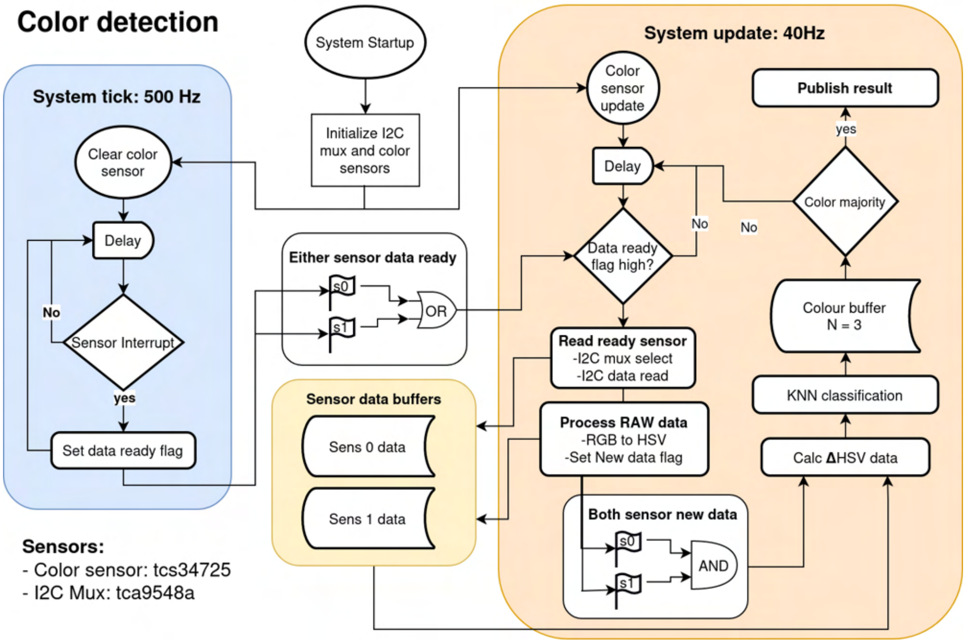


Figure 5.2: **Color detection code flow diagram.**

5.2 VLC communication link implementation

Figure 5.3 provides an overview of the code architecture used in this VLC link. The tick section samples and saves the value from the photodiode in 1 of the two available buffers. There are two buffers because the FFT implementation requires a fixed frame of samples. Hence while performing the FFT the buffer is locked, and writing data to this buffer during an interrupt is impossible. The solution alternates read and write access between buffers freeing the buffer after the FFT and locking it when filled with new samples. A full buffer is processed using the ARM FFT library [5] processing frames of 64 samples at once, and the resulting frequency spectrum is filtered leaving the majority frequency f_0 or f_1. By using the multi-frame approach explained in Section 4.2 and the Boyer-Moore majority voting algorithm proposed in [38], the emitted frequency is retrieved and saved in a buffer for binary parsing. The parsed results are then published to other tasks.

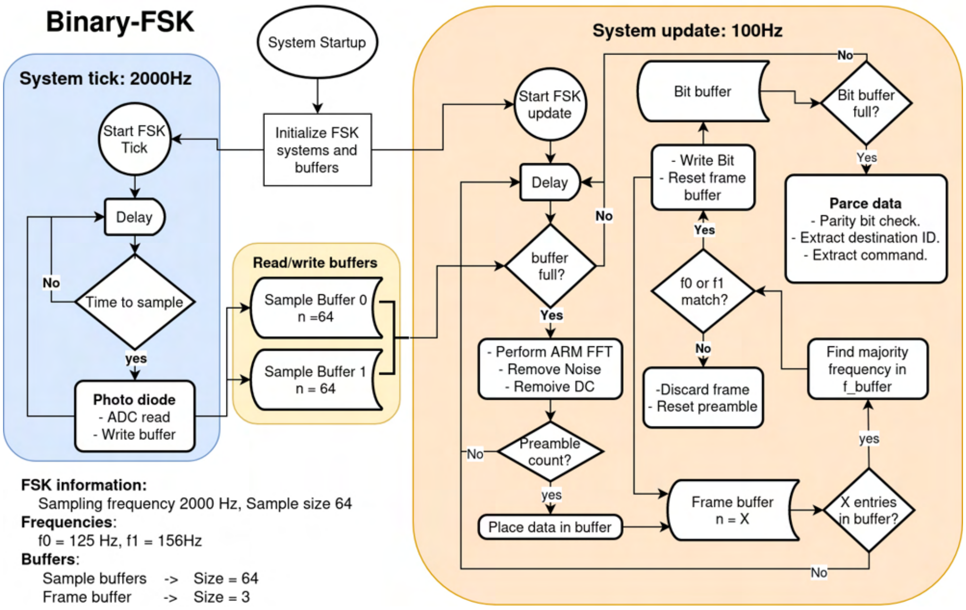


Figure 5.3: **VLC code flow diagram.**

5.3 Particle filter implementation

The final substantial task addition is the Particle filter. Figure 5.4 provides the software architecture overview. The significant contribution to this thesis is the separation of the regular particle model into motion particles and regular particles, as was explained in Section 3.3.8. The single motion particle gets updated at a very high frequency in the tick section. And the accumulated position change + random noise only gets transferred to the regular particles at a much slower interval during the update section. This separation of particle types significantly reduces the number of calculations that would have needed to be performed on all the particles every tick cycle.

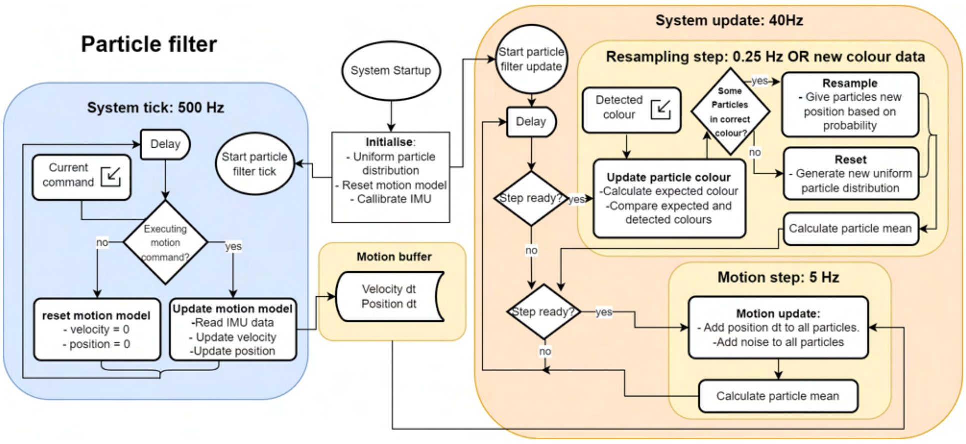
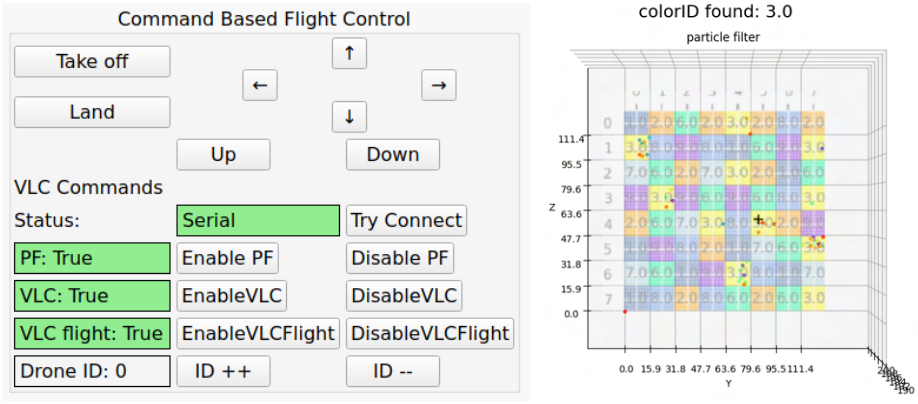


Figure 5.4: **Particle filter code flow diagram.**

5.4 Debug and interface upgrades

Several quality-of-life tools have been developed for efficient interfacing and VLC link testing. These tools are an addition to the existing flight interface [15] provided by the developers of the Crazyflie 2.1. The first tool visible in Figure 5.5a shows the addition of dedicated VLC commands in the GUI. Next to the regular flight commands, there is now the option to send commands over the VLC link. The second addition is the real-time visualisation of the particle filter presented in Figure 5.5b. Due to bandwidth limitations, only a maximum of 50 particles are visible and updated at 1 Hz. Not ideal, but it allows for efficient tuning of the particle filter.



Status: Serial

PF: True

VLC: True

VLC flight: True

Drone ID: 0

(a)

(b)

Figure 5.5: **(a) Updated GUI with VLC command options. (b) Particle filter visualisation where drone is detecting in colour ID 3 (yellow).**

Chapter 6

Evaluation

The evaluation section is split into subsystems and aims to provide experimental results of all systems. We do this splitting because the setup consists of many parts that must work flawlessly for the complete system to work correctly. As a guide, the flowchart in Figure 6.1 is provided that shows what parts of the systems are evaluated as subsystems, ending with the testing of the complete system.

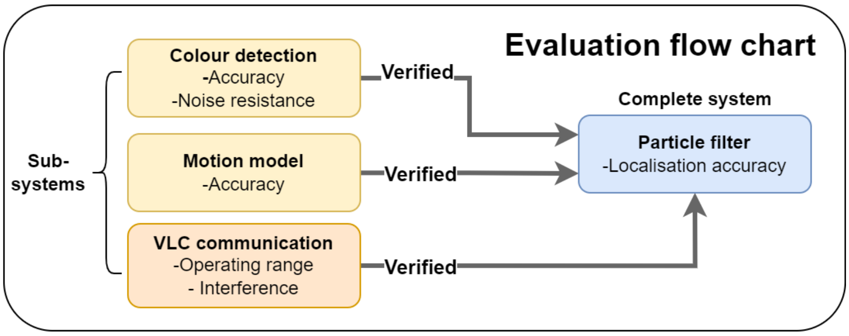
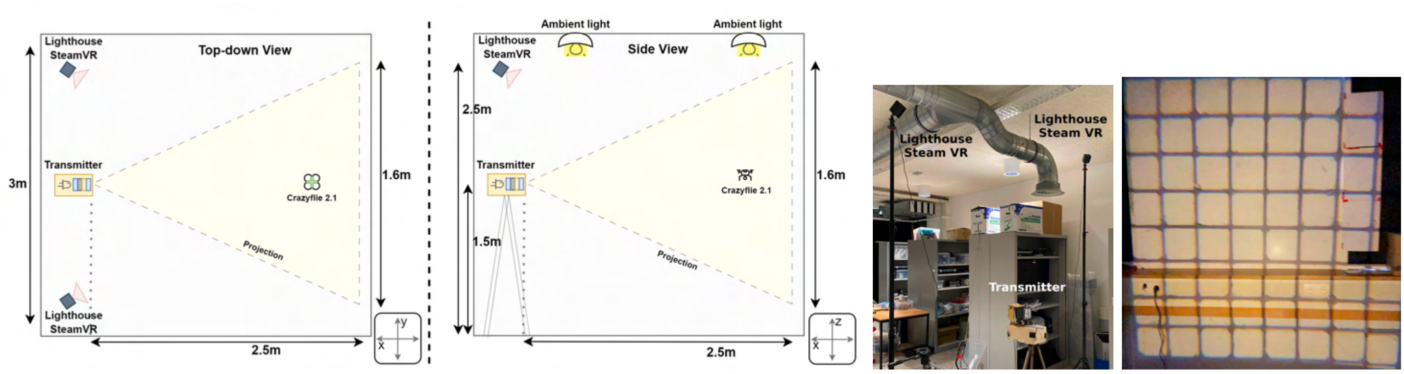


Figure 6.1: **Flow chart for the evaluation of the system and subsystems.**

6.1 Experimental setup

The experiments are conducted in the experimental setup depicted in Figure 6.2. It consists of the transmitter projector setup with the polarisation pattern on a tripod ∼1.5 meters above the ground. The transmitter projects the polarisation pattern towards a wall ∼2.5 meters away from the source. The Crazyflie flies with the PCB sensor array within the projected pyramid shape. The ambient light is adjustable using a dimmable switch ranging between 0 (almost off) to 400 lux. In Figure 6.2c, the interference colours are invisible to the naked eye. However, the structural backplane used for the polarisation pattern casts shadows around the squares, making them visible. Ideally, these shadows would be removed or even thinner.



(a) (b) (c)

Figure 6.2: **(a) Top-down and side view of setup. (b) Test setup transmitter side. (C) Grid backplane casting a shadow.**

6.1.1 Ground-truth reference

A critical part of verifying the behaviour of a localisation setup is the availability of ground-truth information. Hence it was a significant selection criterion in the choice of drone. The Crazyflie supports several state-of-the-art systems utilising Light, RF, and vision-based approaches. Similar to previous works [4], this thesis utilises the infrared (IR) light-based Lighthouse positioning deck based on its proven ability to work, providing cm-level accuracy. During the evaluation of the systems, the lighthouse keeps the drone from drifting away whilst also providing a reference for the particle filter localisation accuracy.

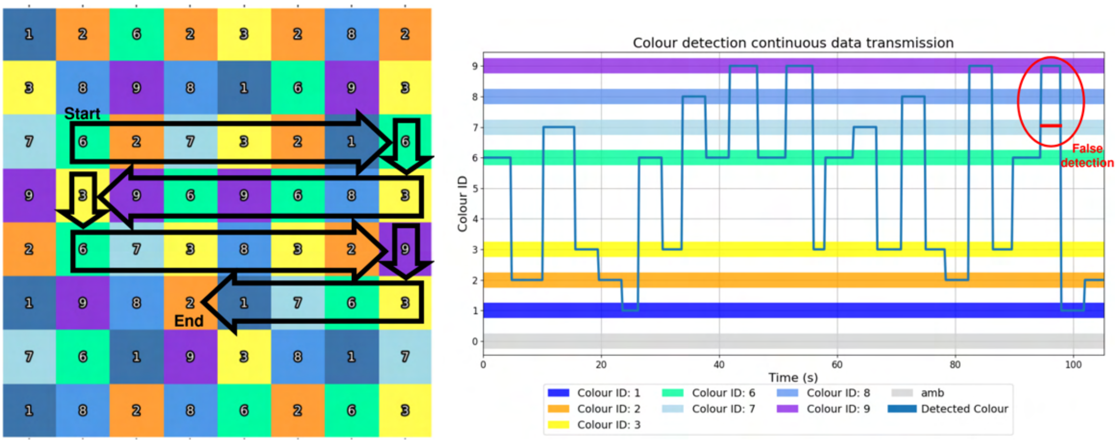
In short, the ground truth system works in the following manner. The Lighthouse system consists of two parts, a receiving lighthouse deck PCB mounted on the Crazyflie and 2-4 SteamVR Base Stations mounted high above the floor (see Figure 6.2a). These base stations emit laser sweeps within the operational area. The lighthouse deck on the drone consists of four receivers that provide the complete pose of the Crazyflie based on the received laser sweep timings. The position and pose calculations are performed internally within the Crazyflie, eliminating the need for external communication.

6.2 In flight colour detection

The evaluation starts with the colour detection and classification subsystem. Much of this thesis is devoted to successfully classifying the acquired colour information. It is very important that a correct classification is presented to the particle filter, as false detections can cause the filter to lose the location estimate. To recap: the colour classification uses a dual RGB sensor approach to remove ambient noise and the RGB to HSV transformation to make the system intensity independent. Furthermore, the K-NN machine learning technique classifies the raw data to the correct colour ID.

To validate the colour detection, the experiments will use the following setup: The drone flies a fixed path at a 2-meter distance from the transmitter. The path is displayed in Figure 6.3a. This path ensures that all colours are encountered at least two times. Note that during the flight, the flying commands are set manually from the PC using a radio link, making the graphs slightly vary in length. And finally, at a distance of 2 meters, the received light intensity captured by the sensors is smaller than 1 lux.

The following scenarios are run first the colour detection at 0 ambient lux without the VLC link to set a baseline. This baseline is followed by varying the ambient lux to validate whether the system resists ambient noise. Finally, an experiment is run where the VLC link is continuously sending information to identify if the link causes any interference in the colour classification.



(a) (b)

Figure 6.3: **(a)Path followed through the grid at 2 meters from the transmitter (b) Shows the detected colours over time as the drone flies a pre-defined pattern. This baseline measurement has 0 ambient noise lux**

Baseline colour detection

Figure 6.3b provides the resulting colour detections of the baseline measurement following the predefined trajectory. All colours are detected correctly except for a single miss classification at cell [5,5]. Based on this information, the baseline is correctly detecting colours during flight. The miss classification is most likely caused by the fact that colour IDs 7 and 9 are close together on the ΔHS plane indicated by Figure 3.9. This false classification illustrates nicely the importance of colour separation. When colours are too identical, overlaps might occur, resulting in cases of false detection. Although small, such a miss classification can severely hamper the particle filter performance. To counter this, a wider spread of particles is required in the particle filter.

Colour detection under ambient light noise

To analyse if ambient noise has any effect on the colour classification, Figure 6.4 shows the results of an experiment that varies the ambient lux intensity to observe the system's robustness against the ambient light noise. Based on these results, the system still works at 50 lux ambient light. At 200 Lux, the colour detection fails. This failure indicates that the dual-sensor approach is not perfect at hight ambient noise light intensities much larger than the recieved source intensity of only 1 lux. Some causes for the failure could be that the ambient light is not entirely uniformly polarised or both sensors are not receiving the exact same amount of light.

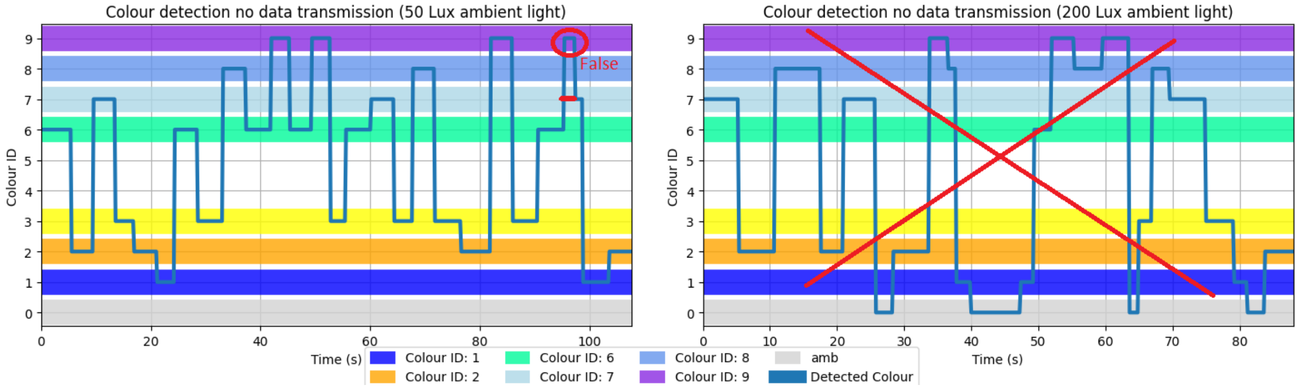


Figure 6.4: **Shows the detected colours show over time as the drone flies a pre-defined pattern, Left is at 50 Lux ambient light and right is at 200 lux ambient light.**

VLC link interference on colour detection

To investigate further the performance implications of the VLC link on the colour classification, the baseline situation was flown again. However, this time, the VLC transmitter was continuously sending data. This flight resulted in Figure 6.5. Observe that the sequence of colour detections is correct and identical to the baseline result. Again, the same classification error at cell [5,5] indicates a K-NN model fault. Based on this information, the VLC link does not affect the colour classification; hence no effect on the localisation accuracy of the particle filter.

6.3 Communication link

The purpose of this section is to validate the performance of the VLC link implemented using the existing infrastructure. The following aspects are analysed:

- The range of the link under various ambient light conditions.
- Effect of the FSK frames developed in Section 4.2.
- Interference with the existing ground truth setup.

The experiments are performed by placing the drone on a stationary mount halfway between the edge and the centre of the projection. Then depending on the specific experiment, the distance is varied, the ambient light is varied, or the ground truth system is enabled or disabled.

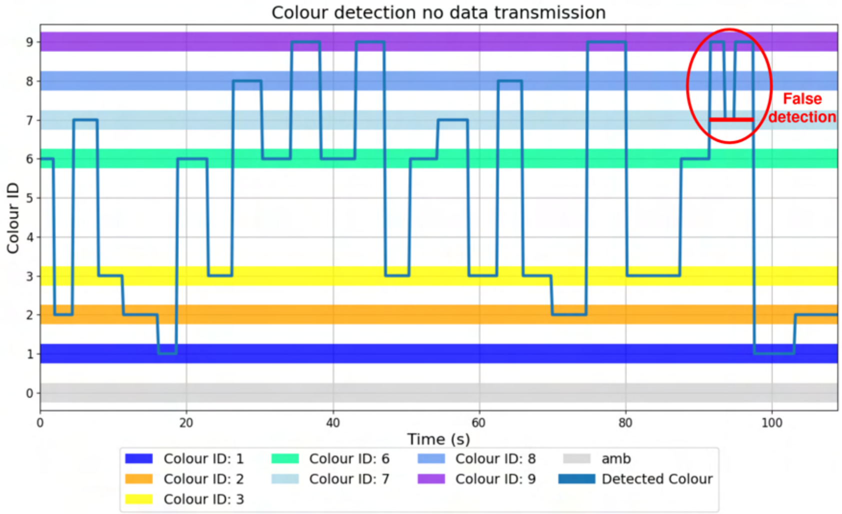


Figure 6.5: **Shows the detected colours over time as the drone flies a pre-defined pattern during continuous transmission of data on the VLC link. This measurement has 0 ambient noise lux.**

Effect of the FSK fames and ground truth interference

To recap: the sensor implemented to establish the VLC link is the OPT101. The sensor suffers severely from the interference caused by the ground truth system that sweeps the room using IR lasers. To solve this, Section 4.2.3 proposes a software solution, but how good is this solution compared to the situation when no ground truth system is active? Furthermore, does the addition of more repeating frames in the FSK VLC link proposed in Section 4.2 help establish a (more) stable link?
To analyse these effects, the drone is placed on a tripod at a 2-meter distance from the transmitter, and the ambient light is reduced to less than 1 Lux. By transmitting random VLC bytes to the drone, Figure 6.6 is created. The figure shows the packet reception rate with varying frame lengths with and without the ground truth system active.

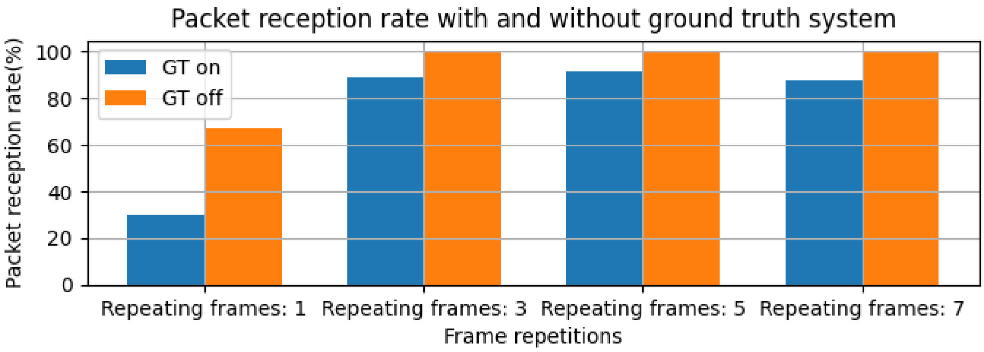


Figure 6.6: **Effect ground truth system effect on the packet reception rate at 2-meter distance with varying frame sizes.**

The system reaches a packet reception rate of 100% at a frame size of three. This reception rate shows that the VLC link works using the existing localisation infrastructure under the conditions that the ground truth system is disabled. Additionally, based on this information, repeating frame sizes larger than three do not provide additional benefits.

The activation of the ground truth system degrades the communication link from 100% at repeating frame sizes larger than 3 to an 85-90% packet reception rate. Indicating that the software solution proposed in Section 4.2.3 provides some improvement over the original 50% but does not resolve all issues. Because the ground truth system must be active while running the particle filter in flight, around 10-15% of the commands will not be received while flying using VLC commands. This does not have a significant effect as the drone will keep hovering in place and only requires manually resending of the command.

Range under various ambient light conditions

The VLC link implementation uses FSK modulation to encode the information. The main reason behind FSK is its stability against varying ambient light noise. Covering the complete localisation space requires a working range between 0.5 and 2.5 meters. In order to verify the link range and noise resistance, the range is increased from 0.5 to 3 meters, and the ambient lux at the drone's location varies from 0 to 200 lux. Figure 6.7 shows the experimental results. The figure indicates a reliable operating range from 0.75 to 2.5 meters under 0 lux conditions. The sensor saturating causes the breakdown at 0.5 meters, likely caused by the transmitter projecting too much light on the sensor due to the focusing of the lenses. The range is reduced when increasing the ambient light, showing resilience but not immunity to ambient light changes. The drop over distance is most likely because the source intensity perceived by the sensor at 2 meters is less than 1 lux dropping it below the noise floor when the ambient light intensity is too large. Keeping the ambient lux around 0 allows the complete system to work in the complete localisation space using the VLC link.

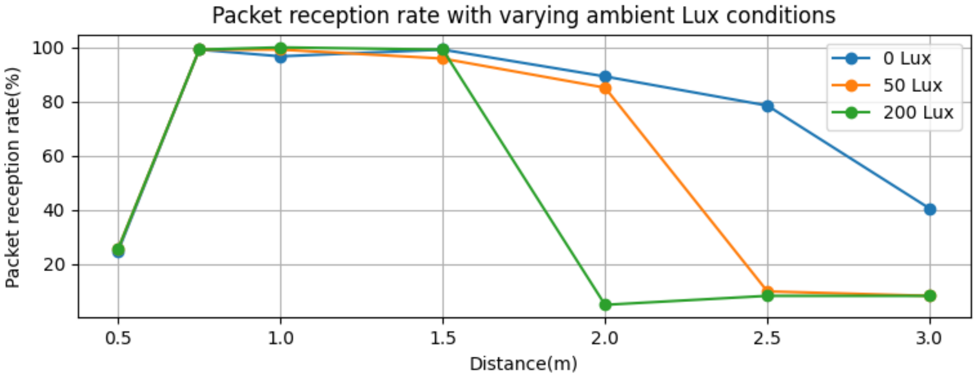


Figure 6.7: **Increasing range with varying lux intensity (ground- truth enabled).**

6.4 Motion model performance

It is now established that the colour detection and the VLC link do not interfere with each other. This leaves the motion model as the final sub-system to be evaluated before the complete system can be evaluated. To recap, the motion model allows the particles in the particle filter to behave like drones. It closely mimics the drone's movement by analysing the IMU, orientation and command data and transferring this estimate to the particles.

This section will evaluate the motion model by flying 2 meters from the receiver and performing stepwise movements in random directions along the separate x, y and z axes. Then, using the ground truth system as a reference to form an estimate of how closely the model follows the actual movement. Where in an ideal situation, we have an exact 1 to 1 match.

By operating the motion model described in Section 3.3.8 during flight manoeuvres, the following figure is created: Figure 6.8. The figures show the ground truth (GT) and motion model (MM) lines along the three axes. Observe that the motion model follows the ground truth in general lines. When the drone hovers, there is no exploding or slow drifting behaviour. It does, however, not precisely follow the movement during dynamic movement along the three axes. The estimation overshoots or undershoots by approximately 0 to 5 cm on a step of 10 cm. This error accumulates, and the estimate diverges from the absolute position. To compensate for this uncertainty in the motion model, the particle filter should add sufficient noise to the system and have many particles to ensure that some particles to still follow a correct trajectory.

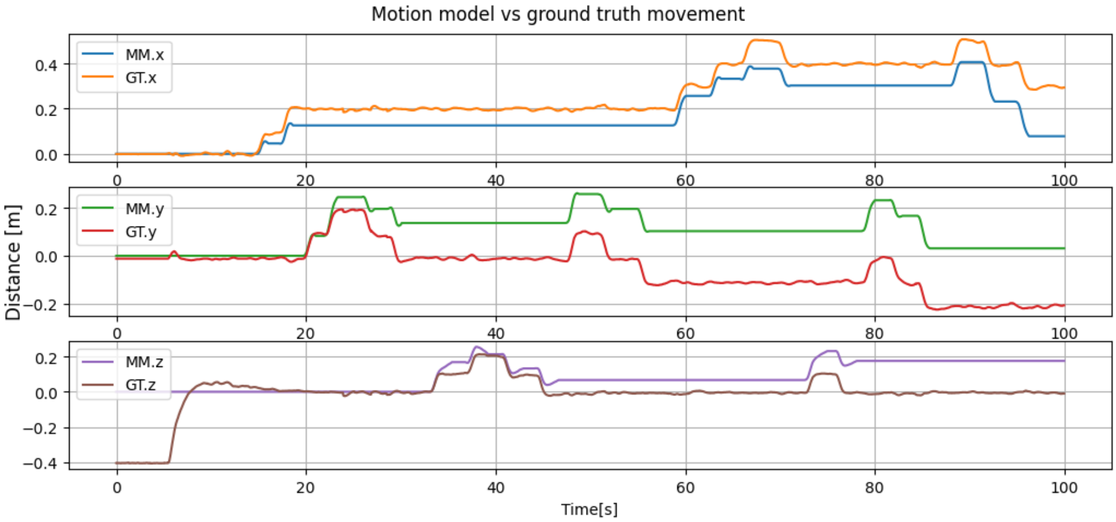


Figure 6.8: **Motion model X, Y and Z dimension performances. Where MM is the motion model, GT is the ground truth.**

6.5 Particle filter performance

With the individual subsystems evaluated and considered to be working, the complete system can be evaluated and compared to the reference paper [48]. Throughout this section, the complete system particle filter is evaluated with all sub-systems working together and providing the necessary information required by the particle filter. The system is evaluated in 2D by flying an arbitrary pattern through the polarisation grid at a fixed 2 meters distance using the 0 lux ambient light case. The commands are sent to the drone using the VLC link with human input. At the start of the pattern, the particle filter is enabled and compared to the absolute position. The result of the 2D localisation is presented in Figure 6.9.

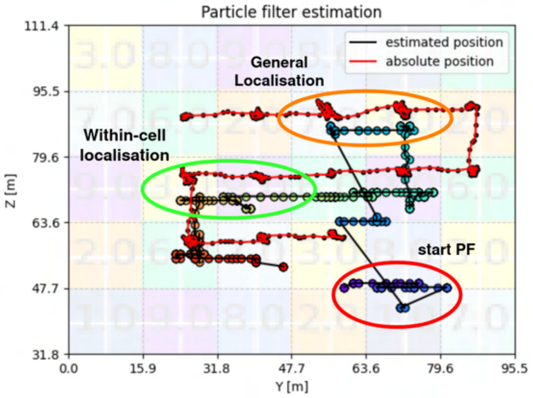


Figure 6.9: **Particle filter location estimate compared to the absolute position.**

In this figure, two lines indicate the position estimate by the particle filter(black) and the absolute position using the ground truth (red). The position estimate of the particle filter is the mean of all particles. Observe how the particle filter estimate first starts somewhere in the middle of the map. It does not know where it is because all particles are uniformly spread over the complete map making the middle the best estimate. After moving through 3 colour cells, the particle filter converges to the same area as the absolute position. This convergence indicates an actively working particle filter. Additionally, it shows that the unique colour sequence length of three colours proposed in Section 3.3.7 works. Only requiring a sequence of 3 colours to approximate the drone's current position.

Localisation accuracy

It is important now to differentiate between general localisation (within a cell size away) and within-cell localisation (within the same cell). The moment the particle filter converges, it approximates the same location as the ground truth. This estimate can be within the current cell or just outside the current cell, meaning it has achieved general localisation. The longer the convergence holds, the more of the straggling particles dotted around the map also converge to the

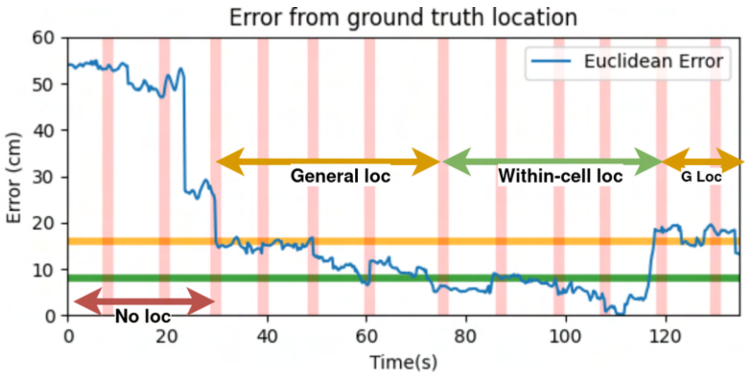


Figure 6.10: **Euclidean distance error red is a colour boundary crossing, orange is single cell size, green is half a cell size.**

current cell making the mean a more accurate representation of the current location. By accurately following the motion model information, the location can be approximated within a single cell, achieving within-cell localisation. Figure 6.10 illustrates the estimation error well. In this figure, the red bars indicate colour crossings, the orange bar is the size of a single cell, and the green bar is half a cell size. Values under the red line are considered general localisation, and under the green line is within-cell localisation. In the Figure the estimate converges to general localisation after three crossings. Furthermore, after crossing an additional four times, the system achieves within-cell localisation. This event is further illustrated by Figure 6.11 where the absolute current cell colour is given against the particle filter estimated cell colour. At first, the colours do not align. During general localisation, the estimated colour leads or lags the absolute colour, and finally, during within-cell localisation, the colours almost match. Providing a second indicator of a close match between the estimated and absolute location.

Something interesting happens around the 120-second mark in Figure 6.10. The particle filter mean has lost its within-cell localisation accuracy. This means that some particles are currently at the wrong location, causing the PF mean to be less accurate. Losing the lock is caused by one or multiple sub-systems providing false information. It is unknown exactly which one caused it in this scenario. However, it only caused some particles to lose their location, allowing the PF to recover over time once the drone traverses more cells. This, however, did not happen in this figure due to a drained battery.

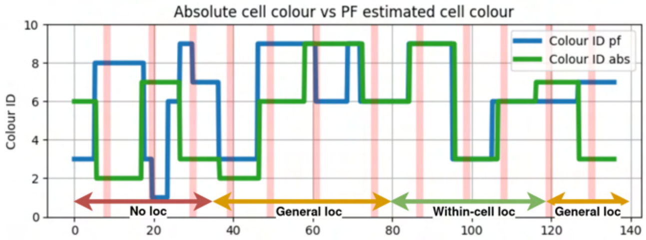


Figure 6.11: **Absolute cell colour vs particle filter predicted cell colour**

Once the system is converged to general localisation, the median error is 10 cm, and the 90th percentile is 17.8 cm. The reference work [48] indicates a 2D median of 4.3 cm, and the 90th percentile is 8.1 cm. This means the current implementation is slightly worse than the reference implementation. The difference in accuracy can be explained in several ways: The first one is that the particle filter still requires more tuning because it converges to general localisation but then requires more time to converge to within-cell localisation. The second difference is the difference in number of particles used by the filters. Despite all optimisations, this implementation can only run 150 particles in parallel, whereas the reference paper runs 1024 particles. Additionally, their work indicates that accuracy saturation only occurs at 1024 particles. Providing a clear indication that this thesis would also benefit from a more significant number of particles.

Chapter 7

Conclusion

Throughout this thesis, we developed a system for real-time drone localisation inspired by the work on [48]. The system uses a custom transmitter to project an optimised polarisation colour pattern into the space. A receiving drone traverses this projection and identifies the unique colour sequences perceived when traversing the projected grid. It localises itself using a particle filter with a known map of the environment, the received colour information from a custom PCB, and an accurate model of the drone's motion. By using the existing infrastructure with minimal modifications, a one-way VLC link is added that allows the drone to navigate on light-based commands.

To answer the main research question: Can we achieve simultaneous visible light localisation and communication on a drone? The relevant subquestions must be answered and combined to form the answer to the main question.

- Is it possible to implement the polarisation-based localisation technique presented in the reference paper [48] in real-time on a limited compute power drone? The analysis shows that real-time localisation is possible. The Crazyflie 2.1 can localise itself with a median 2D accuracy of 10 cm in a 1.6× 1.6 m area. This is less accurate than the 4.3 cm in the reference paper, but this thesis still provides the correct cell once fully converged. It uses a computationally optimised particle filter running 150 particles in parallel to achieve this localisation. Less than the 1024 particles run by the reference paper likely explaining the difference in localisation accuracy. Similar to the reference paper, the data indicates that the projected colours are detected with great accuracy in varying ambient noise lighting conditions.
- Is it possible to use the existing projection infrastructure to establish a VLC link with minimal changes and interference to the localisation system? The results clearly show a successful implementation of a VLC link using the existing hardware. The system incorporates an FSK VLC link resilient to ambient changes and offers a communication range of approximately 2.5 meters, covering the entire localisation area. Notably, the VLC link operates without interfering with the localisation mechanisms. It only requires the addition of a simple MOSFET switch to enable the rapid On-Off switching required by the VLC link. It is worth mentioning that the VLC link does have limitations, including a link speed of only 1.3 bytes per second and the requirement of an additional fast-sampling photodiode to capture the FSK. Finally, the addition of the IR-based ground-truth system for localisation verification also causes the packet reception rate to be only around 90% when the ground-truth system is active.

With the above subquestions thoroughly answered, the main research question is also answered and deemed completed successfully. Yes, it is possible to achieve simultaneous visible light localisation and communication on a Crazyflie 2.1 drone.

Chapter 8

Discussion

In this chapter, we critically reflect on our work by describing the limitations of the system. Furthermore, we identify and outline several possible directions for future work.

8.1 Limitations and Recommendations

8.1.1 3D tracking:

Currently the particle filter is only capable of tracking in 2D. Using the plane perpendicular to the projection at 2-meter distance. To increase the localisation capabilities of the model to 3D, several approaches can be used. Of these ideas, the last two are novel ideas not implemented by the reference paper.

1. Using the Lambertian model [32] to estimate the light intensity in relation to distance throughout the 3D pyramid to estimate the current distance from the source. This approach is implemented by the reference paper.
2. Collect data from the light intensity under various conditions throughout the map. Then use an ML technique like K-NN or others to make an estimate of the current distance from the source.
3. Use the motion model of the drone to estimate how big the squares are when traversing the grid. E.g. the drone speed equals 0.5m/s traversing a square in 0.5 seconds results in a cell size of 25cm. By using the size over distance formula 3.5 for the projection size of the current square results in an estimate of the current distance of the drone.

Any of the above ideas or a fusion of the combined ideas could result in an accurate 3D localisation estimate.

8.1.2 Implementation limitations on localisation

With the design of the particle filter completed, the question remains what to do with the resulting position estimate? This point is where the implementation is currently limited. Stable flight requires corrections to compensate for drift. In a complete system, the particle filter position estimate provides these corrections to prevent the drone from drifting. Figure 8.1 provides an example of how this correction would work. The drone hovers in a blue square. Slowly it drifts away due to integration drift. The particle filter realises this drift and corrects it. Implementing this correction is not trivial and is considered future work for this paper. Currently, the ground truth system and integrated Kalman filter perform this correction procedure while flying.

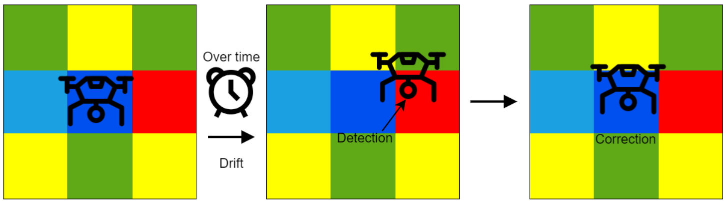


Figure 8.1: **Correcting for the drift using the colour detection and particle filter estimate.**

8.1.3 Pattern imperfections

With the current implementation of the polarisation grid, several dark boundary crossings are observed between the cells. This shadow is cast by the 3D-printed back-plane that provides structural integrity to the polarisation grid. These shadow imperfections can cause the drone to temporarily get stuck on the crossings, unable to receive VLC commands. The problem could be eliminated by using more advanced manufacturing techniques to remove the grid pattern.

8.1.4 The use of a more powerful source

The current implementation uses a 1W LED source. The recommendation is to use a more powerful source in future system implementations despite the requirements stating that the light must be eye safe. A more powerful source would be capable of producing more light. This extra light would greatly benefit the system due to the severe light intensity losses caused by the polarisers. At a distance of 1 meter, only around 10 lux arrives at the drone. This light then still has to pass through a secondary polariser leaving very little for the RGB sensors, especially in conditions where the ambient light arriving at the sensor is 50 lux or more.

8.1.5 Sensor orientation

In the current design, the RGB and the photodiode sensors must face the transmitter projection. If the drone were to perform a yawing motion rotating away from the projection, the particle filter would lose the localisation lock, and the VLC link would not receive new commands. Several approaches could solve this, for example, multiple sensors located around the drone or a freely-rotating sensor automatically aiming towards the transmitter.

8.1.6 VLC link speeds

Due to the limited sample rate of the Crazyflie, only a slow link speed of 1.3 bytes/s is achieved. To improve the link speed, a new flight deck for the Crazyflie can be developed that handles the communication with a much faster sampling rate and provides the received commands directly to the Crazyflie. It might even be possible to integrate existing hardware implementations like OpenVLC [51], an open-source, flexible and low-cost Visible Light Communication System platform, into the Crazyflie.

8.2 Future extensions

There are several interesting directions this thesis can go some of the most promising directions are explained below.

8.2.1 Advanced Grid design

Currently, the grid size is limited by manufacturing capabilities. More advanced techniques allow for very fine-grained patterns with more colour variation. Furthermore, cell shapes other than squares, e.g. hexagons, would allow for more boundary-crossings hence a faster convergence.

8.2.2 Adding a secondary projector

Instead of relying on the ideas proposed in Section 8.1.1, a completely different approach would be to use a secondary projection. Combined, they would provide the 3D position of the drone in the areas where the projections meet. This approach only requires a secondary projection setup and the addition of 2 extra colour sensors perpendicular to the original sensors to work.

8.2.3 Making the transmitter mobile

What would happen if we made the projection source mobile? For example, the source can be aimed using servos or is entirely detached from any fixed anchor

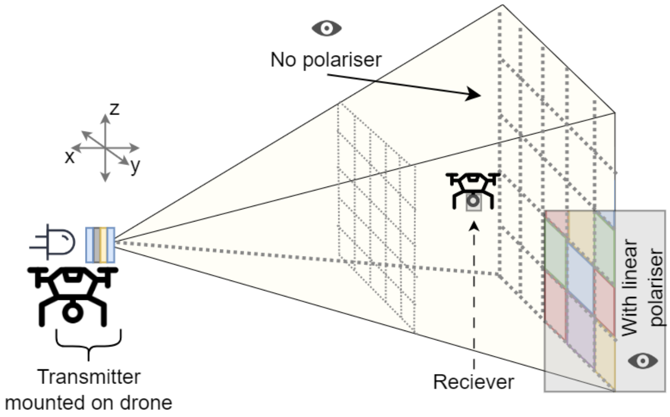


Figure 8.2: **Making the source mobile by placing it on a drone.**

using a secondary drone to carry the projection setup. The receiving drone now localises itself with respect to the transmitting drone. In a static condition, nothing much would change. However, in dynamic conditions, a swarm of drones could be following the transmitting drone using the projection to calculate their trajectories. Figure 8.2 shows an example of what such a portable transmitter would look like.

Because the transmitter is moving, its no longer required for the receiver to move for localisation. Figure 8.3 illustrates this. The received projection will change when the transmitting drone performs pitching, rolling or yawing motions. This change means that the colours of surrounding cells can be observed without having to change position allowing for stationary localisation.

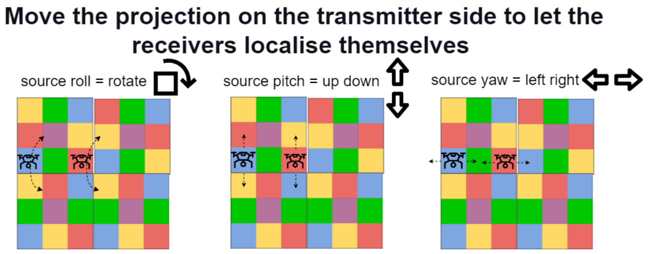


Figure 8.3: **Transmitter movement from the perspective of the receiver.**

8.2.4 Using retroreflectors for localisation

The final possible future extension of this thesis utilises retroreflectors. By placing a polariser in front of a retroreflector, the colour is reflected back at the transmitter. By observing the pattern of reflected colours and using the motion of the drone to estimate the size of the cells, the location can be determined. Now by observing multiple retroreflectors, more accurate estimates can be formed. Figure 8.4 illustrates this idea.

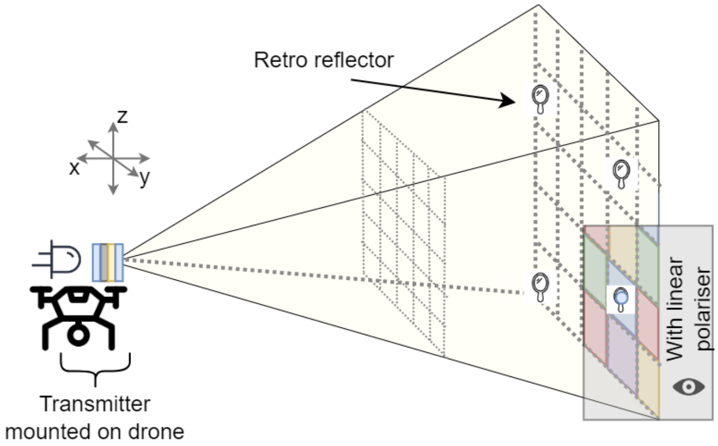


Figure 8.4: **Localisation using retroreflectors.**

Bibliography

[1] Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research}, 2011.*

[2] 911 Security. Drone Communication - Data Link. URL: https://www.911security.com/learn/airspace-security/drone-fundamentals/drone-communication-data-link.

[3] Abdulrahman Alarifi, Abdulmalik Al-Salman, Mansour Alsaleh, Ahmad Alnafessah, Suheer Al-Hadhrami, Mai A. Al-Ammar, and Hend S. Al-Khalifa. Ultra wideband indoor positioning technologies: Analysis and recent advances, 5 2016. doi: 10.3390/s16050707.

[4] Ricardo Ampudia Hernándezhern´hernández, Talia Xu, Yanqiu Huang, and Marco A Z ´ U ˜ Niga Zamalloa. IEEE TRANSACTIONS ON MOBILE COMPUTING Firefly: Localizing Drones with Visible Light Communication and Inertial Sensors. Technical report.

[5] ARM. Fast Fourier Transforms FFTs Introduction . URL: https://developer.arm.com/documentation/101004/2304/Fast-Fourier-Transforms-FFTs/Fast-Fourier-Transforms-FFTs-Introduction.

[6] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Transactions on Signal Processing*, 50(2):174–188, 2002. doi: 10.1109/78.978374.*

[7] Association for Computing Machinery. Special Interest Group on Data Communications. SIGCOMM '13 : proceedings of the ACM SIGCOMM 2013 Conference Applications, Technologies, Architectures, and Protocols for Computer Communication : August 12-16, 2013, Hong Kong, China*.*

[8] Ahmad Helmi Azhar, Tuan Anh Tran, and Dominic O'Brien. A gigabit/s indoor wireless transmission using MIMO-OFDM visible-light communications. IEEE Photonics Technology Letters*, 25(2):171–174, 2013. doi: 10.1109/LPT.2012.2231857.*

[9] Songnan Bai and Pakpong Chirarattananon. SplitFlyer Air: A Modular Quadcopter That Disassembles Into Two Bicopters Mid-Air. IEEE/ASME Transactions on Mechatronics*, 27(6):4729–4740, 12 2022. doi: 10.1109/TMECH.2022.3164886.*

[10] *Bart Duisterhof. Learning to Seek: autonomous source seeking using deep reinforcement learning onboard a Crazyflie, 10 2019. URL:* `https://www.bitcraze.io/2019/10/learning-to-seek-autonomous-source-seeking-using-deep-reinforcement-learning-onboard-a`

[11] *Maik Basso, Iulisloi Zacarias, Carlos Eduardo Tussi Leite, Haijun Wang, and Edison Pignaton de Freitas. A practical deployment of a communication infrastructure to support the employment of multiple surveillance drones systems.* Drones*, 2(3):1–12, 9 2018.* `doi:10.3390/drones2030026`.

[12] *Augusto Beléndez, Elena Fernández, Jorge Francés, and Cristian Neipp. Birefringence of cellotape: Jones representation and experimental analysis.* European Journal of Physics*, 31(3):551–561, 5 2010.* `doi:10.1088/0143-0807/31/3/012`.

[13] *Bitcraze. Lighthouse positioning deck. URL:* `https://www.bitcraze.io/products/lighthouse-positioning-deck/`.

[14] *Bitcraze. state estimation. URL:* `https://www.bitcraze.io/documentation/repository/crazyflie-firmware/master/functional-areas/sensor-to-control/state_estimators/`.

[15] *Bitcraze. Userguide cfclient GUI. URL:* `https://www.bitcraze.io/documentation/repository/crazyflie-clients-python/master/userguides/userguide_client/`.

[16] *Rens Bloom, Marco Zúñiga Zamalloa, and Chaitra Pai. LuxLink. In* Proceedings of the 17th Conference on Embedded Networked Sensor Systems*, pages 166–178, New York, NY, USA, 11 2019. ACM.* `doi:10.1145/3356250.3360021`.

[17] *Cheng Bo, Taeho Jung, Xufei Mao, Xiang Yang Li, and Yu Wang. SmartLoc: sensing landmarks silently for smartphone-based metropolitan localization.* Eurasip Journal on Wireless Communications and Networking*, 2016(1), 12 2016.* `doi:10.1186/s13638-016-0603-7`.

[18] *Alin Mihai Cailean and Mihai Dimian. Current Challenges for Visible Light Communications Usage in Vehicle Applications: A Survey, 10 2017.* `doi:10.1109/COMST.2017.2706940`.

[19] *Mitch Campion, Prakash Ranganathan, and Saleh Faruque. Uav swarm communication and control architectures: A review.* Journal of Unmanned Vehicle Systems*, 7(2):93–106, 2019.* `doi:10.1139/juvs-2018-0009`.

[20] *Bhawana Chhaglani, Abhay Sheel Anand, Nakul Garg, and Ashwin Ashok. Evaluating LED-camera communication for drones. In* LIOT 2020 - Proceedings of the 2020 Light Up the IoT, Part of MobiCom 2020*, pages 18–23. Association for Computing Machinery, Inc, 9 2020.* `doi:10.1145/3412449.3412889`.

[21] *Pooyan Shams Farahsari, Amirhossein Farahzadi, Javad Rezazadeh, and Alireza Bagheri. A Survey on Indoor Positioning Systems for IoT-Based Applications.* IEEE Internet of Things Journal*, 9(10):7680–7699, 5 2022.* `doi:10.1109/JIOT.2022.3149048`.

[22] *Salvador Garcia, Joaquin Derrac, Jose Ramon Cano, and Francisco Herrera. Prototype Selection for Nearest Neighbor Classification: Taxonomy and Empirical Study.* IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):417–435, 3 2012.* `doi:10.1109/TPAMI.2011.142`*.*

[23] *Wolfgang Hess, Damon Kohler, Holger Rapp, and Daniel Andor. Real-time loop closure in 2D LIDAR SLAM.* Proceedings - IEEE International Conference on Robotics and Automation*, 2016-June:1271–1278, 6 2016.* `doi:10.1109/ICRA.2016.7487258`*.*

[24] *Takefumi Hiraki, Shogo Fukushima, and Takeshi Naemura. Projection-based localization and navigation method for multiple mobile robots with pixel-level visible light communication. In* SII 2016 - 2016 IEEE/SICE International Symposium on System Integration*, pages 862–868. Institute of Electrical and Electronics Engineers Inc., 2 2017.* `doi:10.1109/SII.2016.7844108`*.*

[25] *Gábor Horváth and Dezső Varjú.* Polarized Light in Animal Vision*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.* `doi:10.1007/978-3-662-09387-0`*.*

[26] *IBM. K-Nearest Neighbors Algorithm . URL:* `https://www.ibm.com/topics/knn#:~:text=The%20k%2Dnearest%20neighbors%20algorithm%2C%20also%20known%20as%20KNN%20or,of%20an%20individual%20data%20point`*.*

[27] *Iridium. Iridium Simplifies Adoption of IoT Solutions with Amazon Web Services. Technical report, 2019. URL:* `https://www.iridium.com/home/iridium-cloudconnect-contact-form/`*.*

[28] *Johnston P. Creating a Circular Buffer in C and C++, 5 2017. URL:* `https://embeddedartistry.com/blog/2017/05/17/creating-a-circular-buffer-in-c-and-c/`*.*

[29] *Jonathan Kelly and Gaurav S. Sukhatme. Visual-inertial simultaneous localization, mapping and sensor-to-sensor self-calibration.* Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation, CIRA*, pages 360–368, 2009.* `doi:10.1109/CIRA.2009.5423178`*.*

[30] *Latif Ullah Khan. Visible light communication: Applications, architecture, standardization and research challenges, 5 2017.* `doi:10.1016/j.dcan.2016.07.004`*.*

[31] *Motoi Kodama and Shinichiro Haruyama. A Fine-Grained Visible Light Communication Position Detection System Embedded in One-Colored Light Using DMD Projector.* Mobile Information Systems*, 2017, 2017.* `doi:10.1155/2017/9708154`*.*

[32] *T. Komine and M. Nakagawa. Fundamental analysis for visible-light communication system using LED lights.* IEEE Transactions on Consumer Electronics*, 50(1):100–107, 2 2004.* `doi:10.1109/TCE.2004.1277847`*.*

[33] *Manikanta Kotaru, Kiran Joshi, Dinesh Bharadia, and Sachin Katti. SpotFi: Decimeter Level Localization Using WiFi.* Computer Communication Review*, 45(4):269–282, 8 2015.* `doi:10.1145/2785956.2787487`.

[34] *Dong Li, Jialin Liu, Sunghoon Ivan Lee, and Jie Xiong. FM-track: Pushing the limits of contactless multi-target tracking using acoustic signals. In* SenSys 2020 - Proceedings of the 2020 18th ACM Conference on Embedded Networked Sensor Systems*, pages 150–163. Association for Computing Machinery, Inc, 11 2020.* `doi:10.1145/3384419.3430780`.

[35] *Fan Li, Chunshui Zhao, Guanzhong Ding, Jian Gong, Chenxing Liu, and Feng Zhao. A Reliable and accurate indoor localization method using phone inertial sensors.* UbiComp'12 - Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 421–430, 2012. URL:* `https://www.researchgate.net/publication/262211952_A_Reliable_and_accurate_indoor_localization_method_using_phone_inertial_sensors`, `doi:10.1145/2370216.2370280`.

[36] *Lingkun Li, Pengjin Xie, and Jiliang Wang. RainbowLight: Towards low cost ambient light positioning with mobile phones. In* Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM*, pages 445–457. Association for Computing Machinery, 10 2018.* `doi:10.1145/3241539.3241545`.

[37] *Song Liu and Tian He. Smartlight: Light-weight 3D indoor localization using a single LED lamp. In* SenSys 2017 - Proceedings of the 15th ACM Conference on Embedded Networked Sensor Systems*, volume 2017-January. Association for Computing Machinery, Inc, 11 2017.* `doi:10.1145/3131672.3131677`.

[38] *lokeshpotta20. Boyer-Moore Majority Voting Algorithm, 9 2022. URL:* `https://www.geeksforgeeks.org/boyer-moore-majority-voting-algorithm/`.

[39] *Huan Lv, Fang Liu, and Nai Chang Yuan. Drone Presence Detection by the Drone's RF Communication.* Journal of Physics: Conference Series*, 1738(1):012044, 1 2021. URL:* `https://iopscience.iop.org/article/10.1088/1742-6596/1738/1/012044https://iopscience.iop.org/article/10.1088/1742-6596/1738/1/012044/meta`, `doi:10.1088/1742-6596/1738/1/012044`.

[40] *Douglas B. Murphy, Kenneth R. Spring, Thomas J. Fellers, and Michael W. Davidson. Principles of Birefringence. URL:* `https://www.microscopyu.com/techniques/polarized-light/principles-of-birefringence`.

[41] *James A. Preiss, Wolfgang Honig, Gaurav S. Sukhatme, and Nora Ayanian. Crazyswarm: A large nano-quadcopter swarm. In* 2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3299–3304. IEEE, 5 2017.* `doi:10.1109/ICRA.2017.7989376`.

[42] *A. B.M.Mohaimenur Rahman, Ting Li, and Yu Wang. Recent advances in indoor localization via visible lights: A survey, 3 2020.* `doi:10.3390/s20051382`.

[43] *Niranjini Rajagopal, Patrick Lazik, and Anthony Rowe. Visual light landmarks for mobile devices. In* IPSN 2014 - Proceedings of the 13th International Symposium on Information Processing in Sensor Networks (Part of CPS Week)*, pages 249–260. IEEE Computer Society, 2014.* `doi:10.1109/IPSN.2014.6846757`*.*

[44] *Xu Ru, Nian Gu, Hang Shang, and Heng Zhang. MEMS Inertial Sensor Calibration Technology: Current Status and Future Trends, 6 2022.* `doi:10.3390/mi13060879`*.*

[45] *Signify Holding. Signify LiFi, 6 2023. URL:* `https://www.signify.com/global`*.*

[46] *Radu Stoleru, Tian He, John A. Stankovic, and David Luebke. A high-accuracy, low-cost localization system for wireless sensor networks. In* SenSys 2005 - Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems*, pages 13–26. Association for Computing Machinery, 2005.* `doi:10.1145/1098918.1098921`*.*

[47] *Kalyan Pathapati Subbu, Brandon Gozick, and Ram Dantu. LocateMe: Magnetic-fields-based indoor localization using smartphones.* ACM Transactions on Intelligent Systems and Technology*, 4(4), 2013.* `doi:10.1145/2508037.2508054`*.*

[48] *Zhao Tian, Yu Lin Wei, Wei Nin Chang, Xi Xiong, Changxi Zheng, Hsin Mu Tsai, Kate Ching Ju Lin, and Xia Zhou. Augmenting indoor inertial tracking with polarized light. In* MobiSys 2018 - Proceedings of the 16th ACM International Conference on Mobile Systems, Applications, and Services*, pages 362–375. Association for Computing Machinery, Inc, 6 2018.* `doi:10.1145/3210240.3210340`*.*

[49] *Tridgell Andrew. MAVLink. URL:* `https://github.com/mavlink/mavlink`*.*

[50] *Deepak Vasisht, Swarun Kumar, and Dina Katabi.* Open access to the Proceedings of the 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI '16) is sponsored by USENIX. Decimeter-Level Localization with a Single WiFi Access Point Decimeter-Level Localization with a Single WiFi Access Point. *URL:* `https://www.usenix.org/conference/nsdi16/technical-sessions/presentation/vasisht`*.*

[51] *Qing Wang, Domenico Giustiniano, and Daniele Puccinelli. OpenVLC. In* Proceedings of the 1st ACM MobiCom workshop on Visible light communication systems*, pages 15–20, New York, NY, USA, 9 2014. ACM.* `doi:10.1145/2643164.2643167`*.*

[52] *Shui Wang, Kehan Zhang, Bingcheng Zhu, Wei Wang, and Zaichen Zhang. Visible Light Communications for Unmanned Aerial Vehicle: Channel Modeling and Experimental Validation.* IEEE Communications Letters*, 27(6):1530–1534, 6 2023.* `doi:10.1109/LCOMM.2023.3267426`*.*

[53] *Wang Q. Slides_Lecture_02_Pulse-based Modulation, 2022.*

[54] *Yu Lin Wei, Chang Jung Huang, Hsin Mu Tsai, and Kate Ching Ju Lin. CELLI: Indoor positioning using polarized sweeping light beams. In* MobiSys 2017 - Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, pages 136–147. Association for Computing Machinery, Inc, 6 2017.* `doi:10.1145/3081333.3081352`*.*

[55] *Wikipedia contributors. Estimation theory, 2 2023. URL:* `https://en.wikipedia.org/w/index.php?title=Estimation_theory&oldid=1138539907`*.*

[56] *Wikipedia contributors. HSL and HSV, 6 2023. URL:* `https://en.wikipedia.org/w/index.php?title=HSL_and_HSV&oldid=1161414004`*.*

[57] *Bo Xie, Kongyang Chen, Guang Tan, Mingming Lu, Yunhuai Liu, Jie Wu, and Tian He. LIPS: A light intensity-based positioning system for indoor environments.* ACM Transactions on Sensor Networks*, 12(4), 9 2016.* `doi:10.1145/2953880`*.*

[58] *Bo Xie, Guang Tan, and Tian He. SpinLight: A High Accuracy and Robust Light Positioning System for Indoor Applications. URL:* `http://dx.doi.org/10.1145/2809695.2809713.`*,* `doi:10.1145/2809695.2809713`*.*

[59] *Shangyao Yan, Zhimeng Yin, and Guang Tan. CurveLight: An Accurate and Practical Indoor Positioning System. In* SenSys 2021 - Proceedings of the 2021 19th ACM Conference on Embedded Networked Sensor Systems*, pages 152–164. Association for Computing Machinery, Inc, 11 2021.* `doi:10.1145/3485730.3485934`*.*

[60] *Yang Yang, Mingzhe Chen, Caili Guo, Chunyan Feng, and Walid Saad. Power Efficient Visible Light Communication With Unmanned Aerial Vehicles.* IEEE Communications Letters*, 23(7):1272–1275, 7 2019.* `doi:10.1109/LCOMM.2019.2916998`*.*

[61] *Zhice Yang, Zeyu Wang, Jiansong Zhang, Chenyu Huang, and Qian Zhang. Wearables can afford: Light-weight indoor positioning with visible light. In* MobiSys 2015 - Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, pages 317–330. Association for Computing Machinery, Inc, 5 2015.* `doi:10.1145/2742647.2745924`*.*

[62] *Moustafa Youssef, Mohamed Abdallah, and Ashok Agrawala. Multivariate analysis for probabilistic WLAN location determination systems. In* MobiQuitous 2005: Second Annual International Conference on Mobile and Ubiquitous Systems -Networking and Services*, pages 353–362, 2005.* `doi:10.1109/MOBIQUITOUS.2005.41`*.*

Appendix A

Converting ΔRGB to ΔHSV

Note: The content of this document is heavily inspired by this document.

We start out by retrieving the red (R), green (G), blue (B) values, in a scale from 0 to 1, inclusively, for both sensors. Then the ΔRGB values are determined using the following equation:

$$\begin{aligned} \Delta R &= (R_0 - \tilde{R_0}) - (R_1 - \tilde{R_1}) = R_0 - R_1 \\ \Delta G &= (G_0 - \tilde{G_0}) - (G_1 - \tilde{G_1}) = G_0 - G_1 \\ \Delta B &= (B_0 - \tilde{B_0}) - (B_1 - \tilde{B_1}) = B_0 - B_1 \end{aligned} \tag{A.1}$$

Here R_0, G_0, B_0 and R_1, G_1, B_1 are measurements that indicate the Red, Green, and Blue light received by sensors 0 and 1 respectively.

We set the following definitions for the Min, Max, absolute Max, and the difference of the ΔRGB values.

$$m_{\max} = \max(\Delta R, \Delta G, \Delta B) \tag{A.2}$$

$$m_{\min} = \min(\Delta R, \Delta G, \Delta B) \tag{A.3}$$

$$|m_{\max}| = \max(|\Delta R|, |\Delta G|, |\Delta B|) \tag{A.4}$$

$$\Delta_c = m_{\max} - m_{\min} \tag{A.5}$$

ΔHue is now defined as:

$$\Delta H = \begin{cases} \text{undefined}, & \text{if } \Delta_c = 0 \\ \frac{\Delta G - \Delta R}{\Delta_c} & \text{if } m_{\max} = \Delta R \\ \frac{\Delta B - \Delta R}{\Delta_c} + 2 & \text{if } m_{\max} = \Delta G \\ \frac{\Delta R - \Delta G}{\Delta_c} + 4 & \text{if } m_{\max} = \Delta B \end{cases} \tag{A.6}$$

Value is based on the brightest colour channel and is defined as:

$$V = m_{\max} \tag{A.7}$$

Finally saturation is defined as:

$$\Delta S = \frac{\Delta_c}{2 \times |m_{\max}|} \tag{A.8}$$

Appendix B

Bill of materials

Table B.1: **Bill of materials**

Name	**Short description**	**Units**	**Preferred link**
Receiver			
TCS34725FN	RGB colour sensor	6	RS components
Pull-up resistors	10k, 0603 Resistor ±1% 0.05W	30	RS components
Capacitor	0.1 µF, 50 V, 0603, ± 5%	20	Farnell
TCA9548A	I2C mux	2	LCSC
Pin headers	2 row right angle	15	Farnell
Transmitter			
Polarising film	150 x 150mm Linear Polariser	2	Emund
CC driver	LM317BT	2	Farnell
HeatSink	Heat Sink, 4.7 °C/W, TO-220	2	Farnell
Power Resistor	Through Hole Resistor, 1 ohm, 2 W	10	Farnell
Fresnel lens	100x100mm square, f90mm	2	AliExpress
Aduino Due	Micro controller	1	Arduino
MOSFET switch	Gravity: MOSFET Power Controller	1	TinyTronics
LED	1W, warm light	1	AliExpress
Drone			
Crazyflie happy hacker bundle	Crazyflie 2.1 kit	1	Bitcraze
Lighthouse positioning system	Ground-truth	1	Bitcraze

Appendix C

Setup hardware

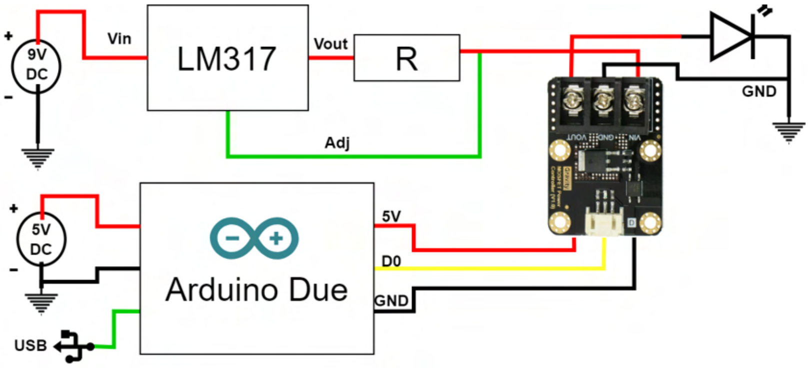


Figure C.1: **Transmitter hardware schematic.**

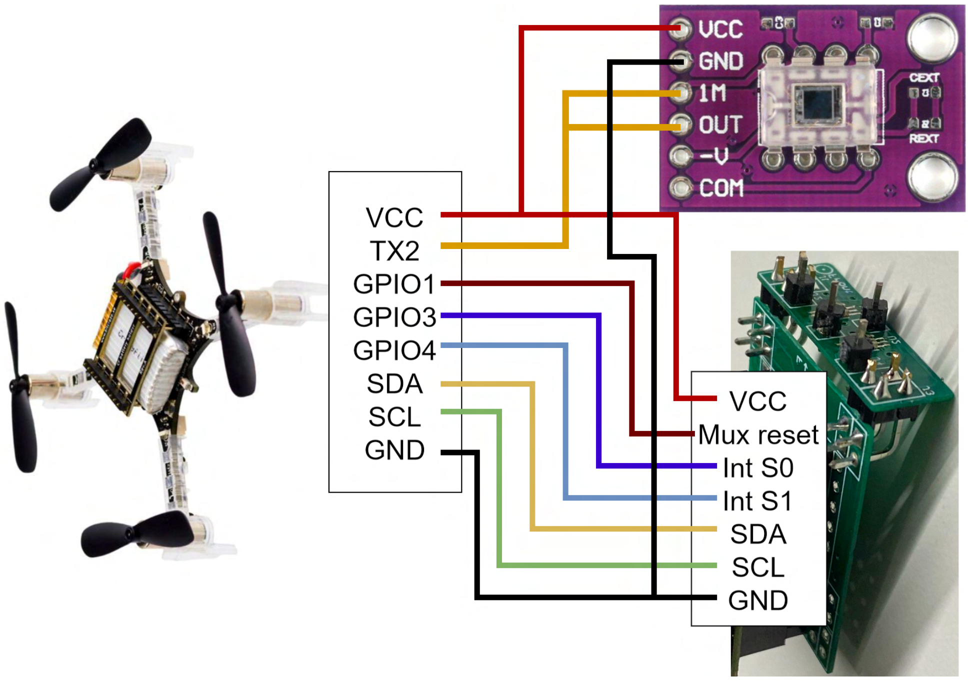
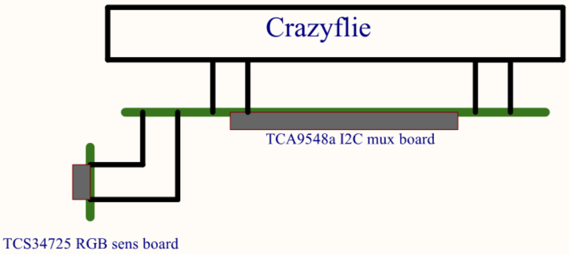
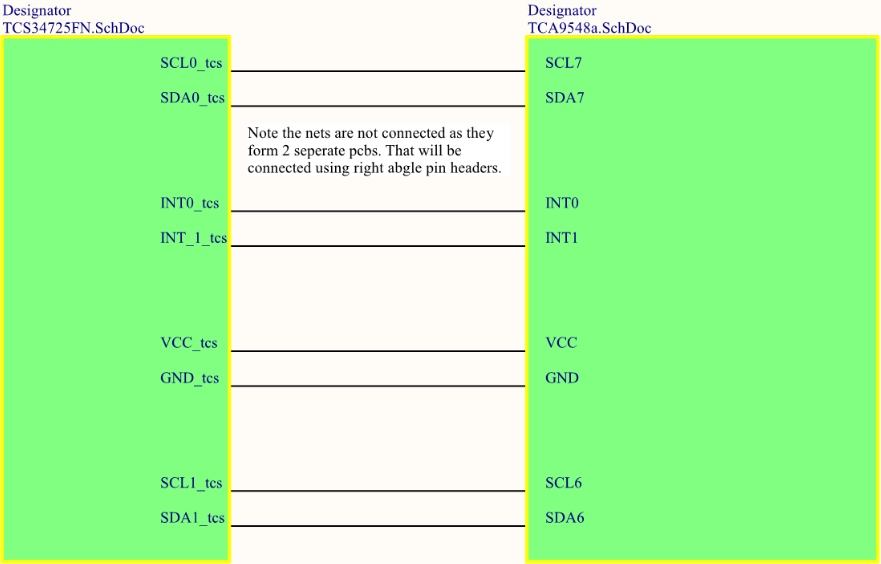


Figure C.2: **Receiver hardware schematic.**





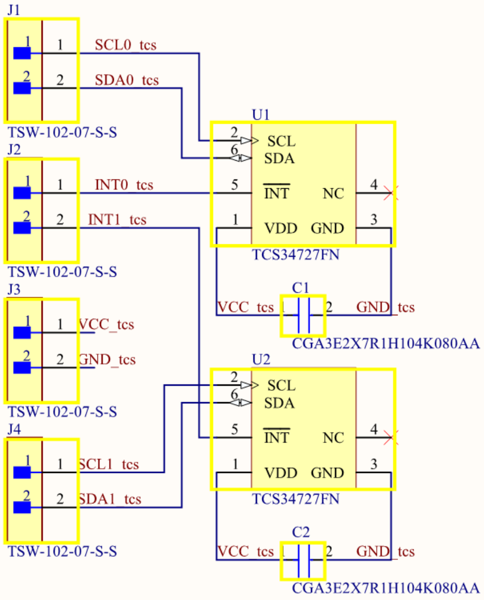
Note the nets are not connected as they form 2 seperate pcbs. That will be connected using right abgle pin headers.

Title	Crazyflie Color Deck	
Size A4	Number	Revision 1.0
Date:	6/20/2023	Sheet of
File:	D:\Documents Data drive\..\Overview.SchDoc	Drawn By: Jasper-Jan Lut

TCS34727FN Sensors

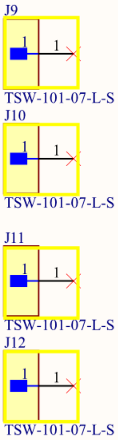
Included 2 TCS RGB sensors. And pinheader connections to the TCA i2c mux board.

This board is intentionaly left to a bare minimum as we would like to have a small footprint.



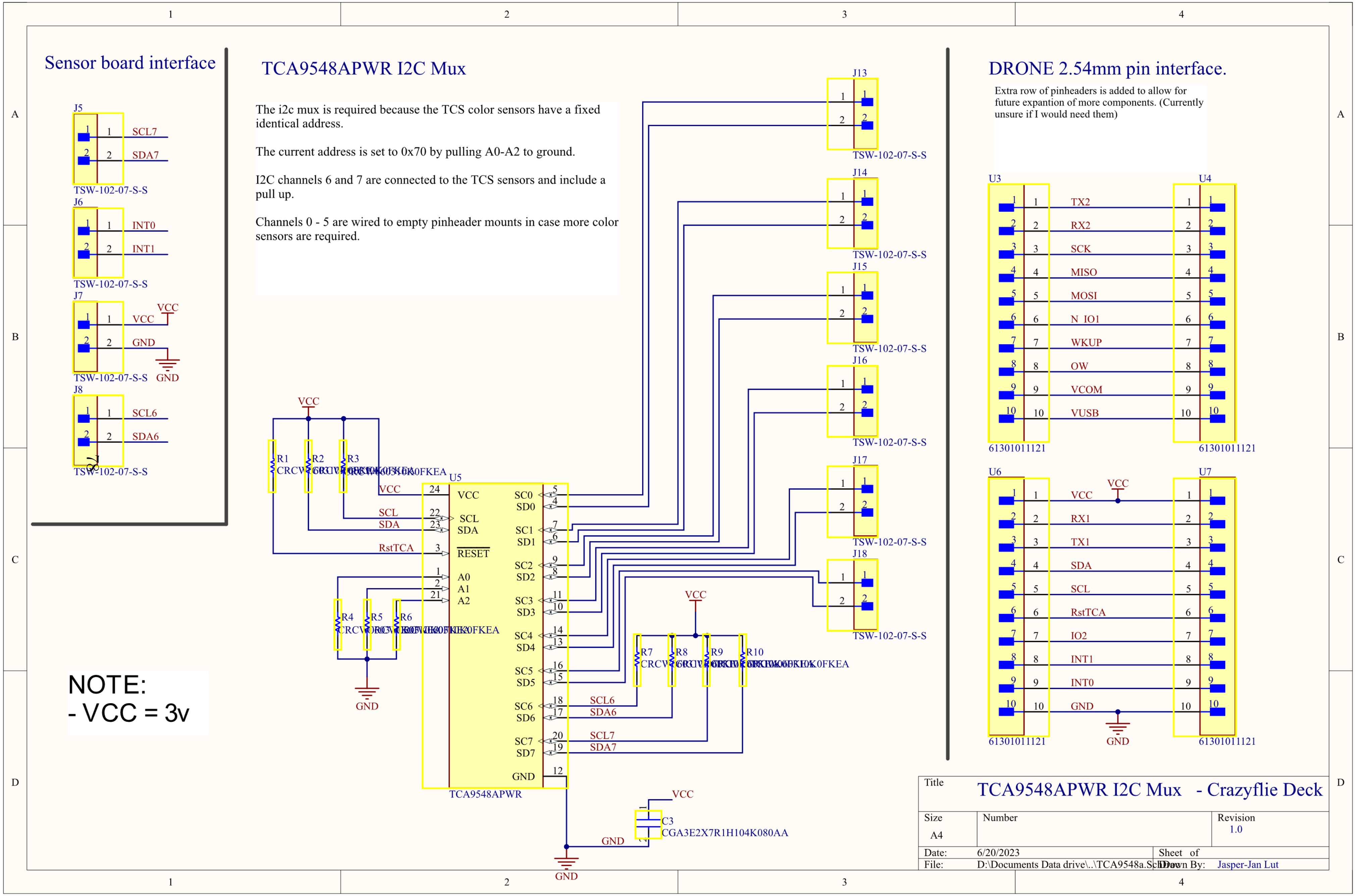
Structural pin headers

These pins are not connected and will be used to position and hold the polarizer sheets above the TCS sensors.



Title TCS34727FN Sensor board - Crazyflie Deck

Size A4
Number
Revision 1.0
Date: 6/20/2023
Sheet of
File: D:\Documents Data drive\..\TCS34725FN.SchDoc
Drawn By: Jasper-Jan Lut



Sensor board interface

TCA9548APWR I2C Mux

The i2c mux is required because the TCS color sensors have a fixed identical address.

The current address is set to 0x70 by pulling A0-A2 to ground.

I2C channels 6 and 7 are connected to the TCS sensors and include a pull up.

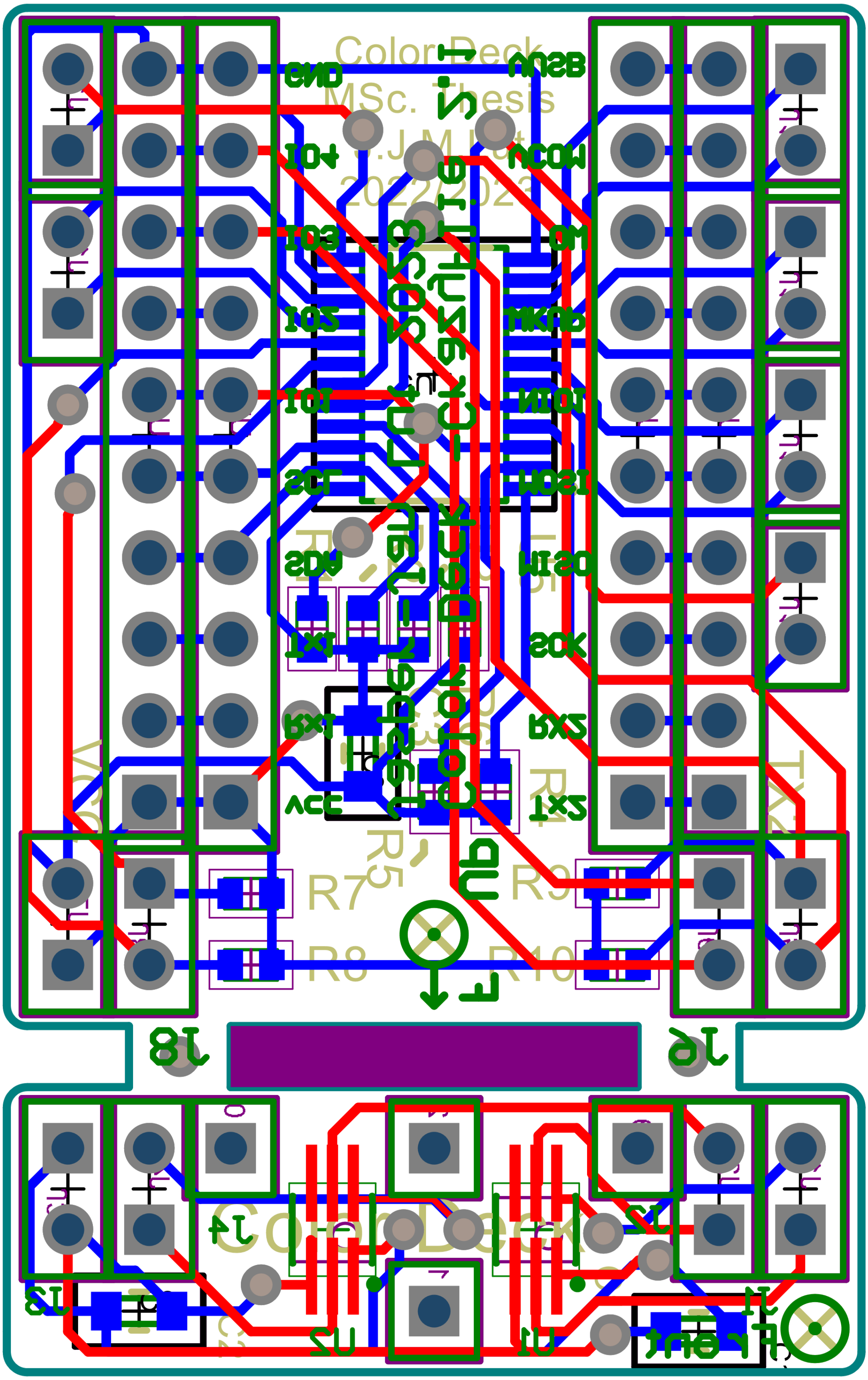
Channels 0 - 5 are wired to empty pinheader mounts in case more color sensors are required.

DRONE 2.54mm pin interface.

Extra row of pinheaders is added to allow for future expantion of more components. (Currently unsure if I would need them)

NOTE:
- VCC = 3v

Title	TCA9548APWR I2C Mux - Crazyflie Deck	
Size A4	Number	Revision 1.0
Date:	6/20/2023	Sheet of
File:	D:\Documents Data drive\..\TCA9548a.SchDoc	Drawn By: Jasper-Jan Lut


ColorDeck
MSc. Thesis
2022/2023
GND
VUSB
IO4
VCOM
IO3
OW
IO2
WKUP
IO1
SCL
MOSI
SDA
MISO
TX1
SCK
RX1
RX2
VCC
TX2
R4
R5
R7
R8
R9
R10
J8
J6
J4
J3
J1
U2
U1