Cooperative Simultaneous Localisation and Mapping for Unmanned Aerial Vehicles in Outdoor GPS-denied Environments

MONASH University

Brian Ramirez Espinosa

Masters of Engineering, Centro de Investigacion y Estudios Avanzados del IPN, Mexico
B.Sc. Engineering (Hons), Instituto Politecnico Nacional, Mexico

A thesis submitted for the degree of Doctor of Philosophy at Monash University in 2018
Faculty of Information Technology

November 2018

Copyright Notice

©Brian Ramirez Espinosa (2018).

Abstract

Swarms of autonomous Unmanned Aerial Vehicles (UAVs) will soon play a significant role in challenging tasks such as the exploration of large unknown geographical regions, mapping, and environmental monitoring. For an autonomous vehicle to be able to perform these tasks, it may need to navigate through an environment without explicit human control, often relying on embedded processing units, localisation algorithms and onboard sensors to do so. Although most UAVs depend on Global Positioning Systems (GPS) for outdoor navigation, it is unreasonable to assume GPS is available in all situations.

One of the most promising ways of enabling a UAV to autonomously explore a large unknown GPS-denied environment is through the use of simultaneous localisation and mapping (SLAM). However, a fundamental problem when performing SLAM is that, without absolute position measurements, the position estimations from SLAM may drift over time. Loop closure is a well-known method of addressing the problem. This reduces the drift in the position estimation by associating new observations of landmarks with those which are already found on the map. Loop closure is especially challenging in large outdoor environments, as the computational complexity of the SLAM algorithm grows quadratically with the number of observed landmarks. Also, incorrect landmark data association can cause catastrophic failures in SLAM.

Multi-session SLAM enables a UAV to avoid the problem associated with loop-closure by combining observations from multiple smaller SLAM missions performed repeatedly over time in the same environment. Even though this technique reduces the number of observed landmarks in each run, it requires a single UAV to perform multiple missions, which significantly increases total flight time and the risk of failure.

This PhD research aims to improve on the current limitations of SLAM and multi-session SLAM by using a decentralised swarm of UAVs to fly over and accurately map large GPS-denied outdoor environments in a single SLAM run. To achieve this task, the proposed system fuses position information from the UAV's sensors (inertial measuring unit, stereo visual SLAM and relative localisation system) via an Error State Extended Kalman Filter (ESEKF), thus creating a system that cooperatively reduces the drift of the position output from every member of the swarm.

A set of outdoor experiments were conducted in a GPS-denied environment to assess the error of the cooperative ESEKF's trajectory estimates compared with the trajectory estimates of the visual SLAM algorithm of each member of the swarm. These resulted in an average reduction in error of the trajectory of each UAV, of approximately 39%, 74% and 48% in the Longitudinal, Lateral and Height axes respectively. The algorithm collectively reduces the drift in the position estimation of each UAV while performing SLAM simultaneously, thus avoiding the need for multi-session SLAM or map-sharing. By using a swarm of UAVs, the system is robust and scalable, enabling mapping and surveillance to be conducted efficiently to a high degree of accuracy in GPS-denied environments.

Declaration

This thesis contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

Signature: .....

Print Name: Brian Ramirez Espinosa

Date: 23 November 2018

Publications During Enrolment

RAMIREZ, B., CHUNG, H., DERHAMY, H., ELIASSON, J., AND BARCA, J. C. Relative localization with computer vision and UWB range for flying robot formation control. In *Control, Automation, Robotics and Vision (ICARCV), 14th International Conference on* (2016), IEEE, pp. 1–6

Acknowledgements

First and foremost, I would like to acknowledge my indebtedness and give my warmest thanks to my supervisors: Dr John Betts, Dr Hoam Chung and Dr Jens Eliasson for their guidance, support and constructive comments during my candidature. I also want to thank Dr Jan Carlo Barca for his initial supervision.

Special thanks are due to the Mexican Government, especially the Consejo Nacional de Ciencia y Tecnologia (CONACYT), for supporting this research through a study abroad scholarship, and to the Faculty of Information Technology at Monash University, for supporting this research through tuition and top-up scholarships.

I wish to thank Mary-Jo O'Rourke, who provided copy-editing and proofreading services according to the university-endorsed national 'Guidelines for editing research theses'. Also to Julie Holden for her invaluable help with my English skills.

I would also like to thank Kal Backman for help building the quadcopters during summer projects. These were used for the outdoor experiments in Chapter 5.

I am especially grateful to my loving wife Karina and my daughter Nessie for their support during my candidature.

Finally, I want to thank my parents for all their support and guidance throughout my life, and to family and friends for their unwavering support.

Brian Ramirez Espinosa
Monash University
November 2018

Abbreviations

CoSLAM	Cooperative SLAM
CoRL	Cooperative Relative Localisation
DoF	Degrees of Freedom
EKF	Extended Kalman Filter
ESC	Electronic Speed Control
ESEKF	Error State Extended Kalman Filter
FoV	Field of View
GPS	Global Positioning System
INS	Inertial Navigation System
IMU	Inertial Measuring Unit
RTK	Real Time Kinematic
SLAM	Simultaneous Localisation and Mapping
SRS	Swam Robotics Systems
TDA	Time Difference of Arrival
ToF	Time of Flight

UAV	Unmanned Aerial Vehicles
UWB	Ultra-Wideband
VSLAM	Vision-based SLAM

Nomenclature

Δt_k The time interval.

$\delta \theta_b^w$ The error in the rotation of the IMU frame with respect to world frame.

Δp_b^w The difference between the real position and the estimated position of the IMU frame expressed in the world frame.

$\hat{p}_b^w$ The estimated position of the IMU expressed in the world frame.

$\hat{v}_{xc}$ The estimated velocity of the coloured marker on the image's x axis.

$\hat{x}$ Estimated state.

$\hat{x}$ The estimated state.

$\hat{x}_c$ The estimated position of the coloured marker on the image's x axis.

ω The true angular velocities of the IMU.

$\Omega(\omega)$ The quaternion-multiplication matrix of ω.

ω_m The angular velocity measurements of the IMU.

ω_{x^b} The angular rotations of in the x axis of the IMU frame.

ω_{y^b} The angular rotations of in the y axis of the IMU frame.

ω_{z^b} The angular rotations of in the z axis of the IMU frame.

$\otimes$ The quaternion multiplication.

θ_j The angle between the optical axis and d_{UWB}.

a The true linear accelerations of the IMU.

a_m The linear acceleration measurements of the IMU.

b The IMU frame or body frame.

b_ω The bias in the angular velocity measurements of the IMU.

b_a The bias in the linear acceleration measurements of the IMU.

c The camera frame.

F_c The continuous time system matrix.

g The gravity vector in the world frame.

G_c The continuous time noise matrix.

H The observation matrix.

h The input histogram.

$I(x_i, y_i)$ The intensity of the PDF of the image.

I_d The identity matrix.

K The Kalman gain.

l The relative localisation frame.

M_{00}, M_{10}, M_{01} The first moments of the search window.

n_ω White Gaussian noise of the angular velocity measurements.

n_a White Gaussian noise of the linear acceleration measurements.

pdf The probability distribution function.

Q The process noise covariance.

Q_c The continous time process noise covariance.

R The measurement noise covariance matrix.

$u = 1$ Hue channel.

$u = 2$ Saturation channel.

v The vision frame.

w The world frame.

x The true state.

x_c, y_c The mean location (centroid).

x_i, y_i The pixel location within the search window.

x_N x coordinate of the position of UAV_j in the NED frame.

y_E y coordinate of the position of UAV_j in the NED frame.

z_D z coordinate of the position of UAV_j in the NED frame.

C_q The rotational matrix corresponding to the quaternion q.

n_{b_ω} The noise in the angular velocity's bias.

n_{b_a} The noise in the linear acceleration's bias.

p_b^w The real position of the IMU with respect to world frame.

Definition of Terms

Cooperative SLAM: is the process by which a swarm of multiple mobile robots cooperatively and distributively build a map of an environment, and simultaneously reduce the computational complexity of the SLAM algorithm on each robot by sharing computational resources [99].

Multisession SLAM: combines the results of multiple SLAM missions performed over time over the same environment [45].

Loop closure: occurs when a robot returns to a previously mapped region and associates landmarks with new position estimates of those landmarks held in the map from different viewpoints to improve the robot's position in the map [30].

Relative localisation: is the process of obtaining the positions of a neighbour based on the range and bearing measurements of an observer [22].

SLAM: Simultaneous Localisation and Mapping is a process by which a mobile robot can build a map of an environment and simultaneously use the map to deduce its location [27].

Visual odometry: is the process of estimating distance travelled, based on changes in key visual features of sequential camera images. [53].

Unstructured environment: is used to refer to an environment without human made markers or features.

Table of Contents

List of Figures

List of Tables

Other Supporting Material

Video of experiments in Chapter 5 posted at https://youtu.be/QIqRG7jf3lQ.

1. Introduction

1.1 Overview

Map making has long been an important part of human activity for communicating useful information about the location of landmarks and other key features describing outdoor environments. Moreover, maps are used to help with critical tasks such as locating oneself in the environment (localisation), path planning, and navigation for applications such as exploration, search and rescue [53, 54]. Therefore, it is very important to construct accurate maps. It is easy to understand that the process of constructing accurate maps is time consuming, and even dangerous when performed manually in certain situations. Consequently, mobile robots have recently gained attention from research groups and industry because of their ability to be controlled remotely or automated, thereby giving their operator safer access to dangerous, distant or otherwise inaccessible places for the purpose of mapping.

Ground-moving robots and unmanned aerial vehicles (UAVs) have been detailed in the research literature as viable options for navigating and mapping outdoor unstructured environments [122, 14]. In this thesis, the term *unstructured environment* is used to refer to an environment free from man-made markers or features. In outdoor locations, mobile robots must navigate over a wide range of obstacles, such as rocks, trees, and changes in gradient. UAVs have an advantage over ground-moving robots for these types of environ-

ments, as they can fly over or under obstacles, and adapt to variations in terrain. UAV systems can explore an unknown environment without explicit human control (beyond setting a flight plan) by using onboard cameras, embedded central processing units (CPUs) and inertial measurement units (IMUs) to create a map of the environment and estimate their position within it. However, these position estimates may drift over time. To reduce this error, the global positioning system (GPS) is typically used to correct position estimation.

Although most UAVs depend on GPS for outdoor navigation, it is unreasonable to assume that GPS signals are available in all situations [28]. For example, mapping a dense forest may result in GPS outages when tree canopies obstruct the line of sight to the sky required to maintain a radio connection with GPS satellites [72]. To address this limitation of current navigation systems, this PhD research aims to develop methods and technologies that increase the accuracy of the position estimates of UAVs in outdoor GPS-denied environments compared with existing approaches. In this thesis, mapping is mentioned as the ultimate goal of the research, but this is beyond the current scope. However, localisation, which is a precursor for mapping, is fully explored.

1.2 Problem Statement

For a UAV to autonomously navigate through an unknown GPS-denied environment, it must be able to localise itself within a map of a given area in order to accurately estimate its location. However, maps may not be available in many scenarios in practice. Because of this limitation, UAVs must be able to sense the environment and create a map while navigating [26, 17]. This approach is called simultaneous localisation and mapping (SLAM) [2, 11] (see Fig. 1.1). However, when a UAV navigates through a large outdoor area, it must deal with a considerable number of landmarks and noisy distance estimates in order to localise itself, often resulting in a drift in the SLAM position estimate. One way to solve this problem

is to revisit previously measured landmarks and associate the new position measurements with those already held in the map [11, 26, 42, 62, 68]. This process is called loop closure [12, 18, 19]. Although loop closure reduces the drift in SLAM after a long run through an environment, the number of new landmarks that need to be associated with previously seen landmarks may increase the computational complexity of the SLAM algorithm to an unacceptably high level [3]. For this reason, multi-session SLAM is often employed to reduce the complexity of a single long SLAM run. This technique combines the results of multiple smaller SLAM runs that have been performed repeatedly over time in the same environment. By having fewer features in each run, the complexity of loop closure is reduced [9]. However, real-life implementations of this technique using UAVs pose many challenges. For example, UAVs have limited onboard computational and communication resources, battery life and sensor capabilities compared with ground-moving robots. This means that the resources available may not be sufficient to navigate multiple times through a large GPS-denied environment. Furthermore, a mapping system comprised of one UAV alone is not robust against collisions with other objects, such as trees, hardware or software failures, any of which could lead to the failure of the whole mission [31, 4].

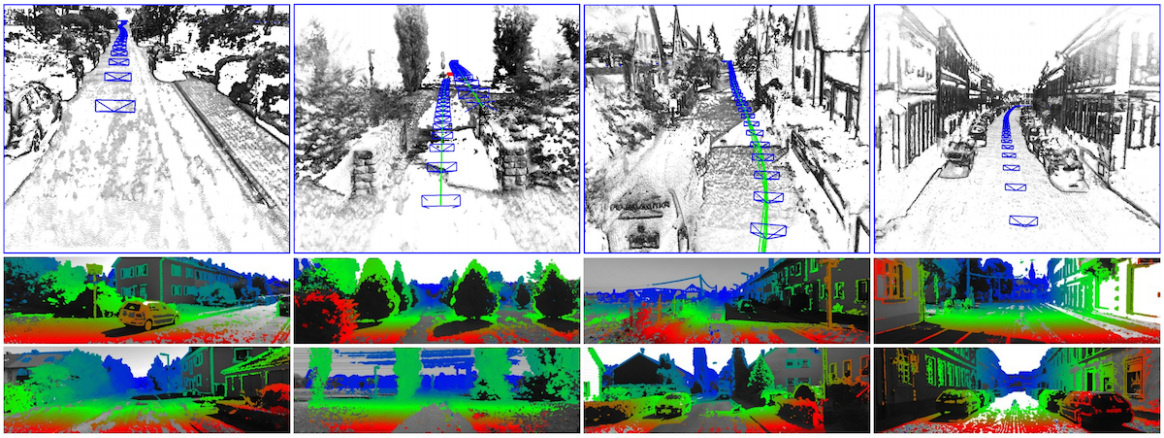


Fig. 1.1 Example of camera position estimate using SLAM in 3D space [148].

To address these issues, this thesis proposes the use of a Swarm Robotic System (SRS)

comprised of multiple UAVs to navigate in formation and map large outdoor GPS-denied environments in a single SLAM run. Using a swarm of UAVs enables the creation of a relatively simple system of agents using their limited resources cooperatively to improve the position estimation of each member of the swarm faster, more accurately and more robustly against hardware or software failures than a single agent could achieve [62, 66].

A swarm of UAVs can perform complex tasks cooperatively and efficiently without compromising the scalability of the system or computational resources [45, 27]. In the case of performing SLAM in an outdoor area, this means that regardless of the size of the swarm, the system can cooperatively reduce the drift in the SLAM position estimate of every member of the swarm in one run through the environment. However, to achieve this objective, it is necessary for each UAV to share position information about partly constructed maps with other members of the swarm. Because UAVs have limited communication bandwidth, sharing parts of the map may not be possible in all situations. For example, in flight, UAVs may not be able to maintain a stationary position long enough to allow the transfer of map data. To address this problem, this thesis proposes that the UAVs share their relative position estimates as an efficient alternative to map sharing. This is achieved by performing relative localisation, whereby each UAV measures the location of its nearest neighbours with respect to its own body frame. The relative position estimates of the neighbour are then transformed to a global coordinate frame, which allows each UAV in the swarm to share their relative position estimates with their neighbours. This, in turn, aids in the coordination of the swarm formation, in order to avoid collisions between members of the swarm or with external objects [45, 25].

To share and use the relative position measurements, an Error State Extended Kalman Filter (ESEKF) has been developed and implemented. The ESEKF fuses the information

available from the sensors on each UAV with the shared relative position estimates of the other members of the swarm. The filter reduces the drift of the position estimates compared with standalone SLAM position estimates. The resulting system collectively reduces the drift in the position estimation of every member of the swarm without adding to the computational complexity of the SLAM algorithm for any individual UAV.

1.3 Limitations of Existing Technology

Previous researchers have investigated robotic mapping implementations to address the difficulties associated with loop-closure in GPS-denied unstructured environments. The main approaches and their limitations are:

- Researchers have proposed the use of a multi-robot system comprised of a slow ground-moving robot fitted with a monocular camera, and a radio based ranging sensor for mapping. The robot estimated its position in an environment using the monocular camera and the SLAM algorithm without performing loop closure. This means that the position estimates of the robot contained drift. To reduce this error, the distance calculated by the robot between itself and a base station using a radio ranging sensor were fused with the SLAM position estimates [67]. However, a base station may not be available in all situations, for example war zone. Furthermore, the physical limitations of ground moving robots reduce their applicability in environments where changes in gradient or large obstacles may obstruct the navigation path.

- Other researchers have investigated the use of a single UAV system to perform several SLAM runs in a large environment. This approach reduces the computational complexity of mapping the environment, as the single system does not have to associate a large number of landmark positions with ones that have been already mapped [85]. However, this method is not robust to failures in the single UAV. Furthermore, a single

UAV system requires a larger amount of time to perform a SLAM run compared to a multi-UAV system, which may not always be possible, for example during an emergency.

- The use of a multi-UAV system to explore and map different regions of a large environment has been proposed by previous researchers. In this approach, the UAVs share and fuse partial individual maps through feature matching to create a global map. Although this approach reduces the amount of time needed to map a large unstructured environment compared with a single UAV, the multi-UAV system has to associate landmarks based on the inaccurate position estimations of UAVs in the system. This leads to inaccuracies in the global map. Moreover, the communication bandwidth needed to share the individual maps between all members may not be available in all situations because of the physical limitations of the UAVs. Furthermore, the time needed to fuse the partial maps complicates real-time SLAM implementations [78].

1.4 Research Objectives

To address the limitations of previous approaches, and create a system that will enable an autonomous swarm of UAVs to efficiently navigate through GPS-denied environments while improving the accuracy of SLAM, this PhD thesis aims to answer the fundamental question:

How can an unstructured GPS-denied environment be explored efficiently by a swarm of UAVs while avoiding multi-session SLAM?

To address this question, the following secondary research questions have been formulated:

- RQ1. How can a swarm of UAVs use the position estimation made by each UAV to improve the accuracy of the relative localisation performed by other members of the swarm?

  Without access to GPS measurements, each UAV in the swarm must rely on inertial measurements to estimate their position within the environment. However, stochastic errors in onboard IMU measurements cause the position estimates to drift. To address this, it is proposed that each UAV in the swarm cooperatively performs relative localisation in order to estimate the positions of their neighbours and in turn have their own position estimated. These relative position estimates are then shared between neighbours in order for each UAV to improve its own position estimates.

- RQ2. How can every agent in the swarm maintain or regain a visual line of sight with its neighbour to update its relative position measurement?

  For the swarm to maintain or recover a formation, the relative localisation system on each UAV needs to be robust against short term failures in tracking and distance estimations. For example, a UAV may be no longer able to locate the neighbour it was tracking because of a temporary break of the visual line of sight due to an obstacle. The system should be able to predict the position of the neighbour until the line of sight is restored or the system determines that the object has been lost. To achieve this, it is proposed to employ a sensor system comprised of a monocular camera with a colour tracking algorithm and Kalman Filter, as well as an ultra wideband ranging

sensor. These, in combination, can estimate and track the position of a colour marker on a neighbouring UAV even when the visual line of sight is temporarily broken.

- RQ3. How can a swarm use the relative position measurements of each member to improve the accuracy and the performance of SLAM?

  The SLAM algorithm scales quadratically with the number of features mapped, and may become too large to compute in a complex environment. Therefore, using these features exclusively incurs a high computational cost. However, each member of the swarm can cooperatively perform relative localisation with neighbouring members of the swarm and in turn, share these measurements. Sharing these measurements allows each member of the swarm to correct the drift of their SLAM position estimates in real time. This is achieved without performing multi-session SLAM or sharing partially constructed maps through the use of the Error State Extended Kalman Filter.

1.5 Model Scenario

The type of scenario motivating this research is the problem of mapping of a large outdoor area using UAVs, in an environment where GPS is unavailable. For example, a swarm of UAVs is sent to perform SLAM in a GPS-denied environment, such as in a forest, under tree canopies. In this scenario, the UAVs are assumed to have limited payload capacity, battery life and communication capabilities, and the area to be mapped is greater than what a single UAV can cover. Furthermore, in the presence of inherent sensor noise, the estimated positions of each UAV may drift. Inaccurate position information may cause each individual navigation system to fail, which could result in collisions with neighbours or obstacles. The objective is

to devise a cooperative SLAM method that allows the swarm to efficiently and accurately map this large unstructured environment without relying on constant GPS measurements.

1.6 Contributions of this Thesis

The main contributions of this research are as follows:

1. Cooperative SLAM has been proposed in recent literature as a way to reduce the computational complexity of SLAM, and improve the exploration efficiency of UAVs. To achieve this, UAVs share partial maps with their neighbours to create a global map and improve each UAV's individual position estimates. However, communication links may be sporadic, and so sharing large amounts of data such as a map may prove difficult in real time implementations. Therefore, this thesis proposes an innovative approach to performing cooperative SLAM whereby the ESEKF is implemented in every member of the swarm of UAVs. This decentralised ESEKF system cooperatively reduces the drift of the standalone SLAM position estimate for every member of the swarm without the need to perform loop closure, multi-session SLAM, or share a partial or a global map.

2. A critical aspect of the implementation of an autonomous UAV is the data-fusion of its inertial measuring unit and external sensors. Two options implemented in the research literature to fuse data are the use of the direct Extended Kalman Filter and the indirect approach, also referred to as Error State Extended Kalman Filter [137]. In the direct formulation, total states such as orientation and position are inputs to the filter, along with inertial measurements from the INS. The correction of the states is achieved using external source signals such as the position estimates from VSLAM [13]. However, there are some severe drawbacks to EKF implementation. In localisation applications especially, the filter may present inconsistencies in the position estimates caused by

accumulated errors in the linearisation process [13]. Therefore, this research proposes the design of a cooperative ESEKF that fuses the position and attitude estimates from SLAM with the inertial measurements of the UAV's embedded IMU and the shared relative localisation position estimates from neighbouring UAVs to improve on the UAV position estimate obtained by SLAM

3. Without access to GPS measurements, each UAV in the swarm must rely on its inertial navigation system to determine its position in the environment. However, random errors in onboard IMU measurements can cause the position estimates to drift exponentially [12]. Making matters worse UAVs have high constraints on payload capacity, computational power, and battery. Therefore, this thesis proposes the creation of a computationally lightweight relative localisation system that can be implemented in a UAV with limited payload and limited computational resources. This system provides reliable position estimates of nearby neighbours for real-time multi-robot navigation using a 180° fish-eye camera, an IMU and an ultra-wideband (UWB) radio.

1.7 Thesis Structure

This thesis is comprised of six chapters as outlined below:

Chapter 1: Explains the motivation for this research and the rationale behind the specific objectives to be addressed. The limitations of existing approaches and technologies are briefly outlined. The contributions to knowledge made in this thesis are briefly summarised.

Chapter 2: Reviews the research literature and summarises the limitations of existing technologies relevant to the project. This includes recent developments in swarm robotics, relative localisation, cooperative SLAM and cooperative data fusion.

Chapter 3: Describes the theoretical approach and technology systems underpinning the design and implementation of a novel relative localisation sensor for UAVs. The testing and validation of the relative localisation system compared with ground truth position measurements is reported.

The concepts and some of the results presented in this chapter have been published in:
RAMIREZ, B., CHUNG, H., DERHAMY, H., ELIASSON, J., AND BARCA, J. C. Relative localization with computer vision and UWB range for flying robot formation control. In *Control, Automation, Robotics and Vision (ICARCV), 14th International Conference on* (2016), IEEE, pp. 1–6

Chapter 4: Presents the mathematical theory underpinning the design of an Error State Extended Kalman Filter (ESEKF), which fuses all the available position estimates from the sensors embedded in a UAV. The chapter presents a preliminary implementation of the filter by testing its application in two UAVs within an indoor environment.

Chapter 5: Describes the implementation of the ESEKF, developed in Chapter 4, in a cooperative swarm of flying UAVs. The accuracy of the new system is evaluated by comparing position estimates from the cooperative ESEKF with ground truth measurements obtained from GPS.

Chapter 6: Concludes the thesis by summarising the contributions and limitations of this research and outlines future research directions.

2. Literature Review

This chapter focuses on one of the primary application of autonomous mobile robotics, that is, the mapping of GPS-denied unstructured outdoor environments. In this environment, a single robot may not be robust to overcome its limited computational and communication resources, limited payload and the possibility of being lost or significantly damaged [28]. In these situations, multiple UAVs can be used, but this introduces the problem of coordination, map sharing and communication. Therefore, the following chapter provides a literature review on the state of the art of autonomous UAV systems developed for mapping large outdoor GPS-denied environments, as well as the terminology and basic concepts necessary to understand the motivations behind this research.

Section 2.1 introduces cooperative SLAM in order to understand mapping in GPS-denied unstructured environments. The Implementation of UAVs in these environments is discussed in Section 2.2. Section 2.3 introduces the concept of swarm robotics to overcome the limitations of UAVs followed by Section 2.4, which reviews recent approaches to Relative Localisation required to coordinate the actions of the members in a swarm. To better understand the coordination approaches of a swarm visual SLAM is discussed in Section 2.5. Section 2.6 reviews the most recent data-fusion methods that aim to improve the position estimation of visual SLAM. Section 2.7 presents the limitations on current implementations of cooperative SLAM and summarises the findings.

2.1 Introduction: Cooperative SLAM

Simultaneous Localisation and Mapping (SLAM) was developed as a practical solution for navigating and mapping unknown unstructured environments without the need for GPS [12, 35]. The most robust type of SLAM for outdoor environments is vision-based SLAM (VSLAM), which records detailed feature information about the surrounding environment [139]. However, Gayathri et al. [48] noted that there are two major problems related to VSLAM for UAV navigation. The first problem is that the resulting map will contain drift because of inherent noisy sensor measurements. The second problem is to perform loop closure, which requires high computational power to associate previously seen features with new features and reduce the drift in the SLAM position estimation.

To avoid the computational complexity of loop closure in large environments, many researchers have investigated how to use multiple smaller SLAM runs to generate a global map [1, 28]. This approach generally uses a single UAV to map a large area by successive runs. However, if the UAV gets lost or damaged, the whole SLAM run may fail. When a swarm of UAVs is used instead, it is possible to take advantage of the cooperative nature of the system to map the region faster, more accurately and more robustly than a single UAV system [83]. Gamage et al. [47] called this approach cooperative SLAM (CoSLAM). This approach decreases the computational complexity of SLAM and increases the robustness of the operation to catastrophic failure either from software or hardware. The main challenge in CoSLAM is, sharing parts of the map or relative position measurements between agents and fusing this information to reduce the drift in the position estimate of each SLAM .

This review introduces the methods and technology developed by researchers to implement CoSLAM for the exploration and mapping of unstructured environments.

2.2 Unmanned Aerial Vehicles

The two main robotic systems that have been detailed in the research literature as viable technologies for mapping large unstructured environments are ground-moving robots and UAVs [146]. Ground-moving robots can be equipped with high-powered processing units, arrays of sensors and high-capacity batteries because this type of robot can carry a higher payload without compromising maneuverability [109, 115]. However, this type of robot has to overcome ground obstacles such as rocks and bushes, as well as changes in the terrain's gradient [83, 88]. By contrast, UAVs can efficiently travel through complex outdoor environments, as they are typically lightweight and fast-moving, and can fly over or under obstacles [96]. However, compared to ground-moving robots, UAVs are constrained by payload capacity and battery life, which limits the computational, sensory and communication resources that can be equipped in the system [97]. A subset of UAVs that manages to balance navigation efficiency in cluttered environments and payload capacity are quadrotors (see Fig. 2.1) [100, 3].

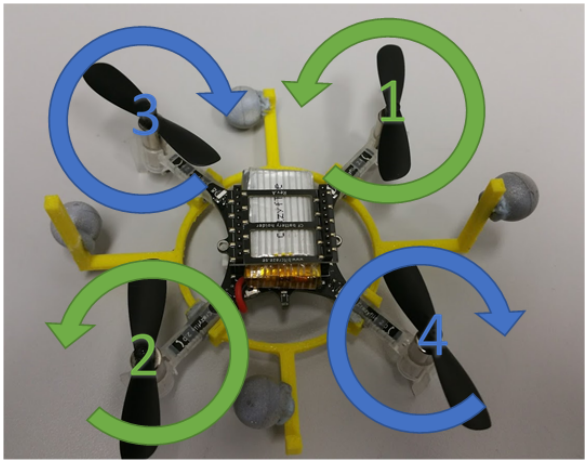


Fig. 2.1 Quadrotor showing propeller direction to generate lift.

Quadrotors have four rotors, which allow movement with six degrees of freedom (6DoF), which includes up, down, left, right, forward and back. [119]. This configuration enables vertical take-off and landing and offers high maneuverability. Because the weight of the UAV is distributed over the four motors, quadrotors have a relatively high payload capacity compared with fixed-wing aircrafts and require less space to take off and land. They are also able to hover at a particular point in space, for example when a particular area is in need of a more detailed inspection. The relative speeds of each rotor enable the UAV to modify its pitch, roll and yaw angles, and to modify its position at any time (see Fig. 2.2) [112]. These attitude changes allow the UAV to map and navigate efficiently in unstructured environments. Because of advances in electronics, quadrotors can now be manufactured in a large range of sizes, which means that they can be scaled to suit a wide variety of applications such as search and rescue, mineral exploration, environmental monitoring (see Fig. 2.3) [59, 141, 39].

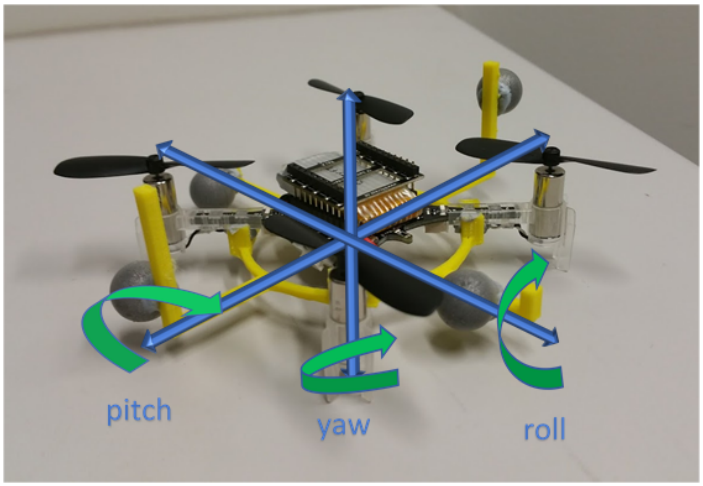


Fig. 2.2 Quadrotor pitch, roll, and yaw angles.

To develop an autonomous UAV system that can navigate through an environment, the UAV must accurately sense or measure its position within the environment. To perform these measurements, recent research literature has detailed two primary sensors: proprioceptive

sensors such as IMUs that determine the position of the system based on inertial measurements [103, 42]. Exteroceptive sensors such as cameras and laser rangefinders that measure the position of objects in relation to the pose of the sensor in order to avoid collisions with obstacles in the environment or other UAVs [71].

Fig. 2.3 Picture taken at the Drone/UAV Search and Rescue Challenge May 17th 2014 Marshall Virginia US [86].

2.3 Swarm Robotics

Multi-UAV systems can share individual resources and coordinate actions to provide greater functionality, robustness to individual failures, better area coverage and shorter mission completion times compared with single-UAV systems [18]. However, the implementation of multi-UAV systems also creates additional challenges, such as overcoming the uncertainty about the collective state of the team [31]. For a multi-UAV system to perform autonomous cooperative tasks, each team member has to maintain information about its state in the environment, the state of the neighbouring UAVs as well as potential changes in the environment.

Many coordination algorithms for multi-robot systems have been based on biological behaviour-based control, which exploits the social characteristics of insects and animals such as swarm behaviour [27]. This type of control is implemented individually in every UAV, detailing communication and resource-sharing mechanisms, as well as reflective and reactive actions to avoid collisions. This method allows interactions between each UAV as well as between the environment, leading to the emergence of a system with intelligent, cooperative behaviour [119, 97].

2.3.1 Swarm Cooperative Behaviour

The deployment of autonomous UAV swarms has the potential to enable fast information gathering, such as mapping, and more efficient coverage of vast areas, by sharing computational resources, sensor position measurements and by coordinating actions in 3D space. Some applications of autonomous UAV swarms detailed in the recent research literature are:

- Manipulating large structural elements in environments that are difficult for humans to access, such as deserts or over oceans [96].
- Collaborating between different types of robots to achieve a common goal. For example, in a large-scale disaster UAVs can locate areas where victims are likely to be found while ground-moving robots search these areas [140].
- Performing simultaneous measurements in different locations, such as sensing mineral deposits in different parts of an outdoor unstructured area [5] and monitoring pollution after an industrial accident [100].

Coordination between the agents in a swarm is essential in order for the system to perform any autonomous cooperative task. However, the implementation of coordination algorithms in the real world UAV systems poses many technical challenges, ranging from communication, task allocation and distributed sensor fusion [71, 119].

2.3.2 Coordination Topologies

There are two main approaches to designing cooperative systems composed of multiple UAVs: centralised and distributed approach. The centralised approach uses one of the UAVs or an external node to coordinate the system in order to map an unstructured environment [97]. The advantage of a centralised system is that it is relatively simple to implement in UAVs. However, the main disadvantage is that the centralised approach has computational complexity, which limits the size of the system. Because adding members into the system increases the processing load of the central UAV where all the information and communication are concentrated [28]. Another disadvantage of a centralised system is that, if the central UAV stops working, the entire communication network breaks down. Therefore, this approach is not robust against the failure of a single component [54].

In a distributed approach, independent UAVs work together to achieve the common goals of the entire population, without the need of a central node. This type of approach models the group's coordination generally based on animal behaviour. For example, Colorni et al. [86] detailed how ants optimise their path to food sources by distributing task among the members of the colony. In another example, Vásárhelyi et al. [133] how birds localise neighbours to create different formations to reduce wind resistance during flight. Like ant colonies and birds, cooperative and decentralised UAV systems are generally scalable, meaning that the system will not increase its computational complexity with the addition or reduction of members to perform a common task [53]. However, effectively coordinating the system is governed by internal and external factors. For example, a few of the internal factors are the communication set-up used to share position measurements with other members and the implemented sensors that enable the UAVs to navigate through an environment. Some external factors are obstacles that change the direction of the system to avoid them and the type of environment mapped by the system, [59, 142].

To achieve the coordination of a swarm of UAVs, it is generally an advantage to have each sensory resource in the system as a resource available to any member of the swarm. The ability to share sensory information, appropriately translated to another UAV's perspective, can extend the capabilities and robustness of a swarm [72]. However, in practice, this is difficult to achieve because each UAV's sensory resource generally provides information based only on that UAV's reference perspective, which is typically its moving coordinate system or body frame. A simple solution detailed in the research literature is to determine a stationary global reference frame that any individual UAV in the swarm can orient its sensor measurements in order to be interpreted by other UAVs [96].

In swarm robotics research literature, relatively simple nature-inspired coordination mechanisms are presented that make distributed systems feasible:

- Information-based coordination: where the interaction is in the form of communications between swarm members. The exchange of information can be either direct, by explicitly passing electronic messages, or indirect, by placing messages in the environment, such as modifying an aspect or adding a distinctive feature to the environment [141].

- Physical coordination: where individual members of the swarm interact at a mechanical level, either directly or indirectly. For example, each member of the swarm directly exerts attracting or repelling forces on each other, such as potential fields or indirectly manipulating a common object, such as modifying the colour of a visible landmark. [142].

Recent research on multi-UAV systems has proposed several methods for coordinating an SRS. These include leader–follower, graph theory and virtual structures (formations) [3].

Meng et al. [86] detailed the leader-follower approach, where the UAV designated as the leader transmits location and orientation information to its follower UAVs. However, leaders do not receive any information from the other members of the system, which means that communication is one sided [8]. Therefore if the communication link is broken the network fails. Also, this system is not robust against the failure of the leader as it solely depends on it to guide the swarm's motion [54].

Coordination using graph theory treats each robot as a node in a graph [100]. The graph models reroute the information of all the other nodes through information-exchange methods which are algebraic graph theory and proximity graphs. These methods assume that the multi-robot system consists of several agents, evolving in a N-dimensional state and every agent is a vertex in a graph. These vertices are interconnected in order to communicate information to a leader from any vertex [113]. These methods play a direct role in the improvement of performance, stability and robustness to variation in the communication topology. For example while using this approach the swarm can select a new leader in case of the failure of the current leader [71]. However, this approach relies on constant communication and knowledge of each UAV's localisation. Therefore, it is not robust against communication interference or when communication dropouts occur [59].

Virtual structures are formations that behave as a single entity, where all the members of the swarm measure the distance and angle with other neighbouring members in order to maintain a predefined geometric relation with its neighbours (see Fig. 2.4) and to communicate position estimates. UAVs in a formation typically move in a certain direction and orientation, maintaining the required geometrical relationship with its neighbouring members [1]. This coordinated motion can, therefore, be considered a system of virtual springs and dampers, which adapt to the changes in the shape of the unstructured environment [122].

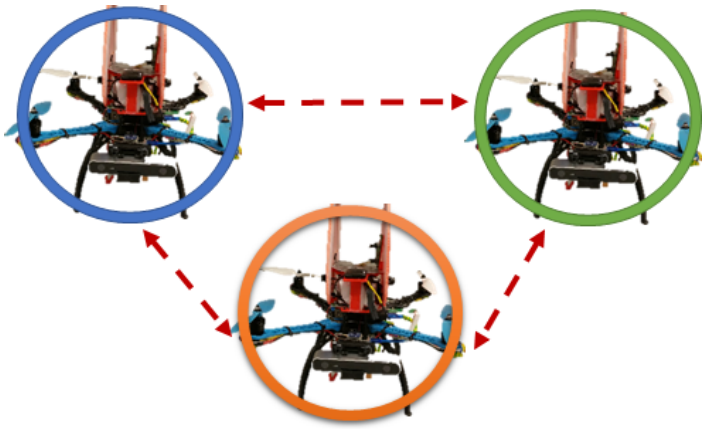


Fig. 2.4 Example of a triangular formation

For example, M. Anthony et al. [145] detailed that in a formation each robot performs an identical optimisation computation of its next position and orientation based on the position of other neighbouring robots in the formation and the maximum and minimum separation between members that has to be maintained for the formation to be stable. Furthermore, the communications requirements to maintain a formation are a function of four factors: 1) the number of robots in the formation, 2) the bits needed to encode the position of a single robot, and 3) the update period of the robot control cycle. Swarms have the flexibility to choose any geometrical formation depending on the environment. For example, if a swarm of UAVs that is navigating in a triangular formation is required to map an area that is not wider than two UAVs. The system can change its triangular formation into a linear formation to fit the desired area and recover its original state at a later time [133, 119].

2.4 Relative Localisation

For formations of UAVs to coordinate their movements, all the UAVs in a formation must know the location of neighbouring members relative to their position based on a common reference frame [136]. Traditionally in outdoor environments, UAVs have access to the global frame via GPS. However, GPS signals may be lost in unstructured outdoor environments such

as a forest. Approaches to overcome GPS occlusions, for example optical motion-tracking systems, use cameras placed in the environment to determine the position of an UAV [136]. Another example is relative localisation, where a UAV or an external node can determine the position of its neighbour based on its position in relation to a global coordinate frame [143, 30].

Without access to GPS measurements, each UAV in the swarm must rely on its inertial navigation system (INS) to perform dead reckoning, which estimates the UAV's position relative to a starting point. Dead reckoning uses the linear accelerations and angular velocities measured from the UAV's IMU to determine its position in the environment. However, random errors in onboard IMU measurements can cause the position estimates to drift exponentially [12]. As a result, the drift in the IMU's inertial frame cannot be modelled deterministically [9, 34]. Therefore, external localisation systems, such as radios, ultrasound and (multi-) camera systems [34, 15, 78] are still widely used for closed-loop control and ground truth position measurements. However, these systems are prohibitively expensive and difficult to set up in unstructured outdoor environments. Moreover, many of these systems cannot determine the position and rotation of a large number of UAVs, as they only allow localisation of up to 30 UAVs in real time [130].

Relative localisation is another way to localise a UAV. This is achieved by measuring and then estimating the position of the UAV of interest based on neighbouring UAVs current positions in the environment [76]. In a swarm setting, each UAV estimates the position of its immediate neighbours in relation to a fixed global coordinate system [89, 95]. Typically an relative localisation system measures the range and angle of nearby UAVs, in order to establish inter-UAV relative position measurements [77, 84].

2.4.1 Cooperative Relative Localisation

Two main types of relative localisation have been proposed in the research literature. They are network relative localisation and cooperative relative localisation (CoRL) [99]. Network relative localisation places static nodes and moving nodes, such as neighbouring UAVs, in an attempt to reconstruct the relative positions of one UAV. This is achieved by triangulating the distance measurements made between the UAV and the neighbouring UAVs. However, this approach may not be feasible in unstructured or unknown outdoor environments, because the accurate positions of the measuring nodes have to be known to the system to function properly. Alternatively, CoRL considers the different moving body frames attached to each UAV and then attempts to estimate the relative distance, and angle among all neighbouring moving body frames [99, 104]. This approach gives more flexibility to the system, as all the UAVs in the system can estimate the positions of their neighbours and in turn have their own position estimated, all while in motion as shown in Fig. 2.5.

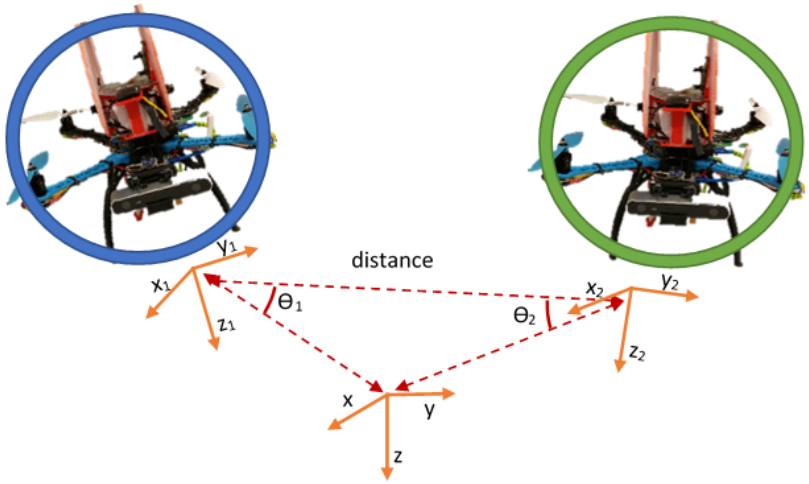


Fig. 2.5 Example of distance and angle measurements from each UAV body frame based on a global coordinate frame

One of the earliest implementations of relative localisation was reported by Sperati et al. [124], in their approach, ground-moving robots were split into two groups. One group of

robots moved while the other remained stationary, essentially serving as landmarks for the first group of robots to locate themselves. Although this implementation could determine the position of a robot, applying it to UAVs might prove difficult, because of the UAVs limitations on battery life. Another method was detailed by Stegagno et al. [125], in this approach, a multi-robot system performed CoRL by measuring the distance between all neighbouring robots to later share these measurements with concerning robots. Therefore, allowing the multi-robot system to fuse the distance measurements of all the members in the system and estimate the bearing of each robot. This approach was implemented in decentralised Extended Kalman Filter (EKF), where each robot shared the computational load of estimating the relative localisation of all the robots in the system. However, when an update to all filters occurred, the system required all robots to communicate with each other and share their position estimates.

Wanasinghe et al. [135], detailed an approach where a mobile robot determined the bearing of a stationary robot from a series of different distance and angle measurements. However, the system required that the robot being localised remain stationary. This approach might not be possible when applied to UAVs because a UAV cannot hover in one place for long periods of time due to battery constraints. Mourikis et al.[89] detailed a more flexible approach to CoRL. This work described how two ground-moving robots performed a circular motion at alternating times in order to be identified and tracked. Although this method allowed the robots to remain mobile, the requirement to perform circular motions may be incompatible with operational objectives.

2.4.2 Sensors used for Relative Localisation

To perform CoRL in any environment, it is highly desirable to have precise and high-frequency sensor measurements, such as the position measurements from sonar, cameras and

laser rangefinders [129]. Using sensor readings, a measuring UAV can accurately measure the relative range and angle of other neighbouring UAVs based on the measuring UAV's location in the environment [27].These measurements can then be fused with the measuring UAV's own inertial sensor data (IMU) to relate the relative localisation measurements to a single coordinate frame [15].

Sonar is one of the most frequently used sensors for localisation, as it is simple, relatively inexpensive and provides distance information. However, sonar has difficulty in measuring the accurate bearing of objects. Monocular cameras are another option detailed in the research literature to localise neighbouring UAVs, as they are inexpensive compared to laser-rangefinders. Cameras can identify and determine the position of specific colours or shapes of a UAV, when paired with visual tracking algorithms [45]. For example, Dugas et al. [34] used wide-angle cameras to create a panoramic image with a 360° horizontal Field of View (FoV), where neighbouring robots could be identified and tracked in order to compute their relative positions.

Active and passive markers have been detailed by researchers to identify and visually track the positions of UAVs [4, 59]. For example, Ioannis et al. [78] noted that this approach could lead to tracking systems that have low computational cost because the system has only to identify the colour placed on the robot and not the individual physical characteristics of the robot. Moreover, Luft et al. [77] developed a computationally lightweight localisation system equipped with target modules, which included four active light emitting diodes (LED) emitters arranged in a particular shape and a monocular camera to identify the predefined shape. However, in outdoor environments the light emitted from the LEDs could be washed out by sunlight and obstacles could occlude the predefined shape, making the tracking of the UAVs difficult. Alternatively, passive markers with distinctive colours can be tracked,

even when occlusions are present, using colour-tracking algorithms that can detect colour in any lighting conditions and a Kalman Filter to improve the position estimation [137, 25]. However, this method requires additional distance measurements to complete the 3D relative position estimation.

For a monocular camera to determine the 3D relative localisation of a marker, it has to be paired with another monocular camera to create a stereoscopic system [46]. However, this approach will increase the computational complexity of the system, because it computes the position of the marker based on the disparity between the two images. Alternatively, radio range measurements like ultra wide-band (UWB) radios paired with a monocular camera can avoid the computational overhead of determining the 3D location of objects [77]. For example, Zihajehzadeh et al. [148] described a system which used UWB radios and Multi-camera system to determine the position of an object. The UWB radios determined the distance to the object being tracked in order to fuse this information with position estimates of the cameras to improve their accuracy. Therefore, improving the position estimates, without the need for more cameras. UWB radios offer interesting properties and capabilities such as low weight, low power consumption, high bandwidth, low sensitivity for multi-path interference and centimetre accurate ranging [144].

2.5 Outdoor SLAM

Absolute position measurements are desirable for outdoor UAV navigation. GPS has been the preferred sensor used to acquire this type of information. However, GPS is unreliable if the line of sight to the sky is occluded, thereby nullifying the effectiveness of the sensor in unstructured environments such as a forest [35]. An alternative localisation approach detailed in the research literature is to estimate a UAV's position through dead reckoning [50]. Although, as stated in Section 2.4, in this approach the position estimate drifts over

time, nevertheless, localisation is still possible by performing Simultaneous Localisation, and Mapping (SLAM) which improves on dead reckoning position estimates [38].

SLAM is performed without the need for any *a priori* knowledge of location [39]. To perform SLAM, the UAV requires two types of sensors. As stated in Section 2.2, these are proprioceptive and exteroceptive sensors [40] (as shown in Fig. 2.6). To perform SLAM, a UAV estimates its current position by measuring the distances between itself (x_k) and environmental landmarks (z_{ik}), then associates these measurements with a bearing command given or control vector to the UAV (u_k), at a discrete time k, as the following quantities are defined [35, 12]:

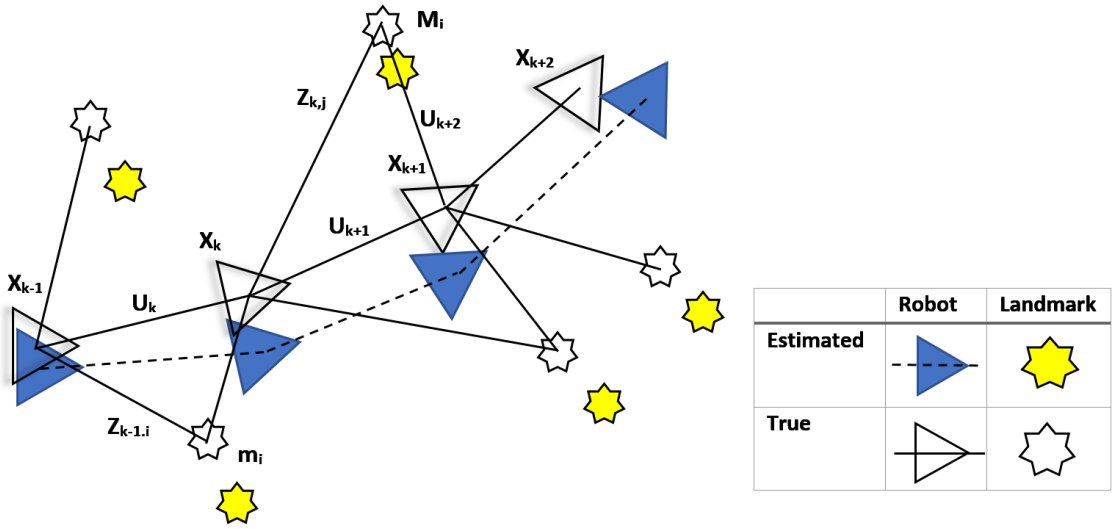


Fig. 2.6 Diagram representing the simultaneous estimate of both UAV and landmark locations, and the inherent error in the position estimation due to noisy measurements [35, 12].

- x_k: the state vector describing the location and orientation of the vehicle.
- u_k: the control vector, applied at time $k-1$ to drive the UAV to a state x_k at discrete time k.
- m_i: a vector describing the location of the ith landmark whose actual location is assumed time-invariant.

- z_{ik}: an observation taken from the UAV of the location of the ith landmark at a discrete time k.

In Fig.2.6, it can be seen that for the most part, the error between the estimated and true landmark locations is approximately the same between all landmarks position estimates. This is because of errors in the UAV's dead reckoning position estimates when the landmark observations are made [49, 35]. To estimate these errors, the state–space model of the system comprising the observation model and the motion model have to be defined. The observation model describes the probability of making a measurement z_k when the vehicle location and landmark locations are known and are typically described as below. The motion model for the vehicle can be described as a probability distribution on the previous position and control vector as described below.

$$P(z_k|x_k, m) \tag{2.1}$$

$$P(x_k|x_{k-1}, u_k) \tag{2.2}$$

Estimating the position of the UAV based on measured landmark locations involves finding an appropriate representation of the observation model and the motion model. The most common representation of these models is in the form of a state–space model with additive Gaussian noise, leading to the use of a Kalman Filter to reduce the error or drift in SLAM position estimates [45, 51].

2.5.1 Loop Closure

SLAM can be used to determine the local position of a UAV in an environment when absolute position measurements are unavailable. However, the estimated position will drift over time because of noise in the sensor measurements. To overcome this error, the UAV returns to a previously mapped region after a long excursion to correctly associate visible landmarks

with landmarks held in the map, thereby closing a loop. This method has been named in the research literature as loop-closure [52]. Although performing loop closure can reduce the drift, errors in associating previously mapped landmarks locations with current position measurement can lead to the failure of the SLAM algorithm [56].

Most loop-closure detection approaches are appearance based because they exploit the distinctiveness of the environment in a captured image [65, 115]. Appearance based loop-closure detection is done by comparing all the features of previous images with the most recently captured image features [85]. Implementing this approach in real time requires to analyse the incoming images faster than the time required to acquire them [61, 118]. Thus, as the map grows, the time required for loop-closure detection increases and computational complexity scales quadratically, with the number of features held in the map. This complexity eventually limits the size of the environment that can be mapped and the applicability of the system in real time [90].

Research literature has detailed sub-mapping as an approach to address the computational complexity of loop closure. Sub-mapping avoids computational scaling by combining smaller SLAM runs, and thus involves smaller data associations [101]. However, the UAV must start in an already mapped portion of the environment. Alternatively, the UAV can initialise a new map with a referential map on start-up, when a previously visited location is encountered, a transformation between the two maps is computed to merge the maps [43]. The transformations between the maps can be saved explicitly, with unique nodes called anchor nodes, or implicitly, with links added between each map to facilitate map merging [102, 116].

A type of sub-mapping detailed in the research literature is multi-session SLAM, which is the task of aligning two partial maps of the environment collected by the UAV during

different periods of operation [107, 109] (see Fig. 2.7). In multi-session SLAM, concurrent autonomous navigation is required, in which one UAV performs small SLAM runs over different parts of the same environment until the whole area has been mapped. For example, McDonald et al. [85] noted that it is challenging for a single UAV to perform multi-session mapping given the limitation inherent to UAVs such as battery life and payload capacity, which hinder the UAV system ability to complete map an environment. Therefore, the need for a better solution is required

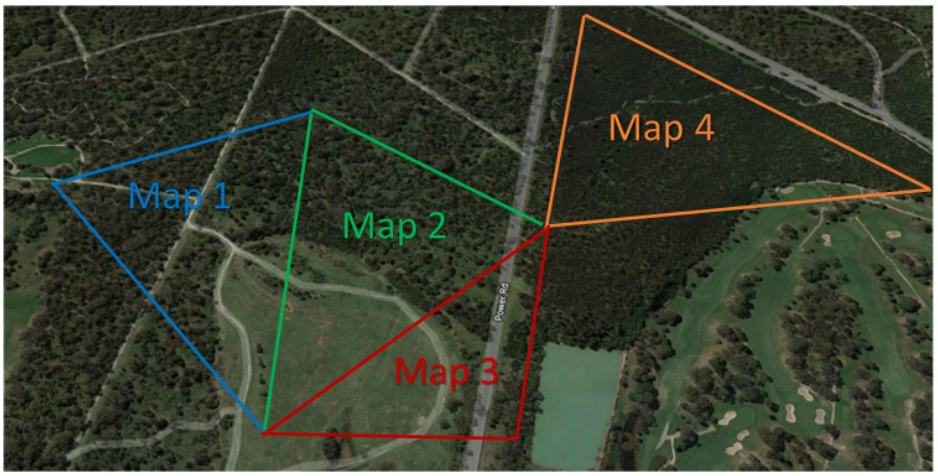


Fig. 2.7 Example of overlapping regions of sub-maps to create a global map.

2.5.2 Visual SLAM

In order to perform appearance based loop closure, cameras offer a larger bandwidth of information compared to laser-rangefinders or sonar. Cameras can capture unique features of landmarks in unstructured outdoor environments, which allows for data association through feature recognition [26, 42], thus creating Visual SLAM (VSLAM) [43, 128]. However, depending on the method for recognising features, VSLAM may be computationally expensive in applications that need to run in real time [61].

Two main approaches have been developed to identify landmarks on an image and to determine their positions based on the position of the camera. They are the direct method and the feature-based method [62]. The direct method exploits all the information from the environment, even from areas where intensity gradients are small. Although this method is capable of providing accurate position estimates, it is computationally intensive, because SLAM scales quadratically with the number of mapped landmarks [68]. For real-time implementations in unstructured environments, where the potential number of landmarks may be unlimited, the computational complexity of SLAM is potential limitation .

The feature-based method uses a set of features dependent on the environment as landmarks, such as points and lines. These are then extracted in each image in successive frames. However, features may appear different in consecutive images depending on the position from which the camera made the observation [139]. To overcome the changing appearance of the features is with the use of scale-invariant feature descriptors, where image gradients are measured at a selected scale in the region around key features (points and lines). The key features are then transformed into a representation that allows for the visual system to recognise them even when shape distortions occur [81, 128].

To recognise the features as landmarks, two approaches have been detailed in the research literature. The first approach groups the extracted features from the scene as landmarks, rather than the features themselves, thereby reducing the number of landmarks to a manageable set, which in turn reduces the computational complexity of the algorithm [91]. The second approach to VSLAM landmark recognition is model-based, where images are searched in a database that contains the images of previously seen landmarks. Although this method reduces computational complexity, it may not be suitable in unstructured outdoor environments as features differ in different lighting conditions [103].

Based on feature recognition, a navigating UAV embedded with a camera can determine the distance and direction it has travelled. This method is called visual odometry, where the position of the UAV is determined using a sequence of images [114]. This method is often used in conjunction with stereo cameras or monocular cameras, where the features third dimension is used in combination with their appearance in consecutive images to determine the distance travelled by the camera [113, 136].

Monocular cameras can provide the positions of outdoor landmarks, but at a high computational cost [146]. The camera captures a single image of the environment where landmarks are located. This is performed at every time step during its movement, the features from consecutive images are then extracted and combined in order to determine the distance between the sensor and the features [143]. For example, a position estimate can be computed using images captured from two UAV at two different time steps, to determine the location of a landmark. However, both UAVs need to know precise relative localisation between them at the two time steps and to associate them with the captured images [43, 91].

Jakob and Schöps et al. [38], presented a monocular VSLAM algorithm. This work used pixel intensity errors of image patches, instead of traditional point feature detection. In this approach, a bearing vector and distance were identified parametrised features in the environment. A Kalman Filter was then designed to estimate the position of the camera, based on the intensity errors. Eryong et al. [39] presented a monocular VSLAM system for a UAV to map a GPS-denied environment. This approach followed a hierarchical structure from the observations of the camera module, meaning that the system calculated the position only using keyframes, thereby losing important image information in the process. Finally, Edmundo et al. [51] developed a low-cost quadrotor that was capable of monocular visual

navigation in unstructured environments by using scale invariant features and off-board processing.

Stereo cameras provide numerous advantages, For example, the set baseline between the cameras perceives the scene at a solid angle, allowing the development of accurate 3D VSLAM approaches [134, 8]. For example, Pradeep et al. [103] presented a visual position estimation system from multiple cameras in a 360° configuration on board a robot, to determine its accurate position within the area as well as possible obstacles. Congdao et al. [26], mounted a stereo camera on board a robot, while the system used scale-invariant features as landmarks. Although using this type of features is computationally complex, this approach makes motion estimation accurate and robust against large translations and rotations. [31, 101].

Despite substantial research progress in VSLAM, many issues remain to be solved before a robust visual mapping and navigation solution can be widely deployed on UAVs [40, 12]. A key issue is that of persistence, which is the capability of a UAV to operate for long periods of time without collisions or hardware failure as well as reliably associating previously seen landmarks with new ones [107, 52]. Because of battery and payload constraints this key issue still remains, as UAVs cannot treat the mapping of an outdoor area as a single large mission, instead, data should be collected from multiple SLAM runs and then processed to determine the accurate locations of the UAV and landmarks [85, 9].

2.5.3 Cooperative SLAM

Cooperative SLAM (CoSLAM) is an efficient framework for solving the problem of drift in SLAM. This method is based on the cooperation of multiple UAVs to estimate each UAV positions in the environment as well as to build a map of the area. The individual maps built by the multiple UAVs should then be merged to obtain a global map [109]. CoSLAM

also increases the navigation system's robustness against failure, reduces the computational complexity and improves the exploration efficiency and range [57]. In the research literature, special attention has been paid to estimation consistency and robust data association involving CoSLAM. To optimally map an unstructured environment cooperatively, different implementations of estimation techniques have been proposed, including EKFs, information filters and particle filters [23].

CoSLAM is closely related with CoRL and performing state estimations within sensor networks, where the positions of all UAVs are cooperatively estimated. The main difference is that, in CoSLAM, UAVs need to share and fuse the partial maps of all agents by using distinctive overlapping features or landmarks between the maps, as well as the estimated UAV position [50]. However, communication links may be sporadic and sharing large amounts of data like a map will complicate real time implementations.

Several multi-robot CoSLAM approaches have been devised in the recent research literature. For example, Lemaire et al. [43] proposed a collaborative vision system for localisation and mapping by using IMUs and RGBD (red, green, blue and depth) sensors. A monocular VSLAM algorithm was used for localisation tasks, and depth data processed from the sensor was used to solve the scaling of the mapped features [58]. The position estimates and maps from multiple robots were then transmitted to a ground station where, in the case of sufficient overlap between robot views, the maps were merged based on a global coordinate frame [64]. Similarly, León et al. [69] employed a fleet of UAVs to form a collaborative stereo camera for the localisation of robots in an environment. The sensors used in the proposed scheme were a monocular camera, an IMU and a sonar for each robot. The sensor position measurements were fused in an EKF for state estimation. Finally, a formation control algorithm was developed to maximise the overlapping FoV of the individual cameras embedded in the robots.

Leung et al. [70], presented an approach for accomplishing multi-UAV CoSLAM, where every UAV in the system was equipped with an IMU and a stereo camera system. A VSLAM algorithm was implemented each UAV, and the position estimates from VSLAM were filtered through a non-linear controller. The system accuracy for both the position of the vehicle and the map cartography depended on feature re-observation, where a UAV observed features already registered by another robot. Madhavan et al. [80] presented a multi-robot mapping method based on multiple position graphs. This work utilised anchor nodes, equivalent to base nodes to decompose the mapped area into sub-maps and optimise the position estimates with a batch optimisation approach, called Tectonic Smoothing and Mapping. Roumeliotis et al. [110] also presented a SLAM algorithm where robots only communicated parts of the map of the environment, in order to reduce communication bandwidth. However, this method can yield highly conservative estimates compared to sharing each robot's complete map.

To fuse the information from multiple maps, Rashid et al. [108] introduced a sparse extended information filter to merge key features of the map, in this approach robots, do not need to be aware of each other's starting location as the filter will match similar visual features. Similarly, in the approach proposed by Zhan et al. [145], a robot localises itself in another robot's map to confirm their relative positions before merging their maps. As detailed above, most of the CoSLAM methods are based on either map or image merging, which requires overlapping exploration areas and significant amounts of data to be transmitted. To avoid these issues, Paull et al. [98] presented a method in which the relative positions between the two rovers were estimated by using mono cameras and range measurements, without transmitting any image or feature point. One robot's mono camera estimated a landmark's position as well as a neighbouring robot's position. The estimated position of a neighbour with respect to a landmark as well as the distance between robots was then shared with the

other robot. This information is the fused in order to improve the position estimates of the robot.

Recent research literature details EKF-based cooperative localisation algorithms to reduce the communication overhead of CoSLAM's map sharing. This cooperative approach is not dependent on timely updates as position estimates can be back-propagated once they are shared and rely only on pairwise communication in the context of the relative position measurements [145]. Approaches in which the UAVs treat the positions of other neighbouring UAVs as deterministic parameters and cross-correlations entail the risk of becoming overconfident or forces the robots to follow certain motion patterns. A more elaborate approach is to treat incorporation of relative measurements as a fusion of estimates with unknown correlations [69, 110].

2.6 Data Fusion to Improve SLAM

UAVs autonomously exploring an unstructured environment must estimate their location within it. However, none of the existing localisation technologies alone can meet the desired performance specifications, such as accuracy and speed [5]. For example, dead reckoning uses high refresh rate IMUs to estimate the fast changing position of a UAV. However, this position estimate will drift because of noise in the inertial measurements [16]. The integration of additional sensor measurements has been detailed in the research literature as a possible way to improve the accuracy of IMUs' position estimates [82]. One example is the use of the position and rotation estimates from VSLAM as an aiding sensor measurement to reduce the drift in dead reckoning [20]. The integration of additional sensor measurements is called sensor data fusion, as it combines measurements from different information sources to overcome the disadvantages of each sensor with the advantages of other aiding sensors [22].

The most widely used data-fusion methods in robotics originated in the fields of statistics, Bayes' theorem, estimation and control [33]. Probabilistic Bayesian methods are now considered the standard approach to data fusion in all robotics applications for combining measurements in order to estimate a UAV's true position. These estimation methods are implemented in a number of ways such as the Kalman Filter, sequential Monte Carlo estimators or functional density estimates [36].

An estimator is a decision rule which takes as an argument a sequence of observations or measurements (IMU, stereo camera) and whose action is to compute a value for the state of interest (position) [41]. Sensor state models are required by the estimator in order to understand what type of information comes from the sensor. Environment state models are also required in order to relate the position measurements from the sensors to states to be estimated. Finally, some concept of information value or gain is needed in order to judge the performance of the measurements [63]. For example:

- The state of the system is $x \in X$, this quantity describes an environment or process such as the position estimate from dead reckoning. A state model comprises a set of possible states together with any knowledge of how the elements of this set are related [63].

- The information about the observations or measurements that a sensor yield describes a quantity z. For example, the position estimate from VSLAM is a single realisation $z \in Z$ from this set. For each specific state of nature, $x \in X$ an observation model is required that describes what observations the system will make $z = z(x) \in Z$ [63].

- Given the information obtained through observation z, the goal of the data fusion process is to infer the underlying state x. To do this, a gain or decision rule is described K which maps observations to states, $K(z) \rightarrow x \in X$ and enables the prediction or the update to be fused [63].

In summary, the estimator incorporates information about the nature of the observation process, beliefs about the state and the value placed on the accuracy of the measurement [79]. The decision rule is essential in data-fusion problems, it is the function that takes in all information and produces a single decision or resulting estimate. The most important problem in data fusion is the definition of appropriate models of uncertainty associated with both the state and observation models [82].

2.6.1 Kalman Filter

The Kalman Filter assumes that the states and their noise have Gaussian distribution and that the current state is linearly dependent on the previous state [13]. The Kalman Filter is an optimal estimator in the sense that it minimises the mean-squared error under the linear-Gaussian system assumption [29]. The simple and robust nature of this recursive algorithm has made it particularly appealing in the field of mobile robotics [10].

The Kalman Filter employs an explicit statistical model of how the parameter of interest $x(k)$ (prediction) evolves over a discrete time interval k and an explicit statistical model on how the measurements $z(k)$ (correction) that are made are related to the prediction. The gain or decision rule K employed in a Kalman Filter is chosen to ensure that, the observation and state models used, provide a resulting estimate $\hat{x}(k)$ minimises the mean-squared error, rather than a most likely value [60].

The Kalman Filter may be considered a specific instance of a recursive Bayesian filter for the case where the probability densities of states are Gaussian. The starting point for the filter is to define a model for the states to be estimated in the standard state-space form at a discrete time k as shown below [60]:

$$x(k) = F(k)x(k-1) + B(k)u(k) + G(k)v(k) \tag{2.3}$$

Where $x(k)$ is the state vector of interest, $u(k)$ is a known control input, $v(k)$ is a random variable describing uncertainty in the evolution of the state and where $F(k)$, $B(k)$ and $G(k)$ are matrices describing the contribution of states, control input if it exists and noise to state transition respectively. An observation (measurement) model is also defined in the standard state-space form as shown below [60, 73]:

$$z(k) = H(k)x(k) + R(k)w(k) \tag{2.4}$$

Where $z(k)$ is the observation vector, $w(k)$ is a random variable describing uncertainty in the observation, and where $H(k)$ and $R(k)$ are matrices describing the contribution of state and noise in the observation respectively at a discrete time k [60, 74]. When the measurement model and the state model are defined, the Kalman Filter then proceeds recursively to estimate the true state, as shown in Fig. 2.8.

In the real world, the state model and the observation model may not be linear, because of the intrinsic properties of the sensors used in a UAV such as drift in the bias of the input sensor measurements. When the state model, the observation model or both are non-linear, an Extended Kalman Filter (EKF) can be employed. This filter assumes a non-linear Gaussian

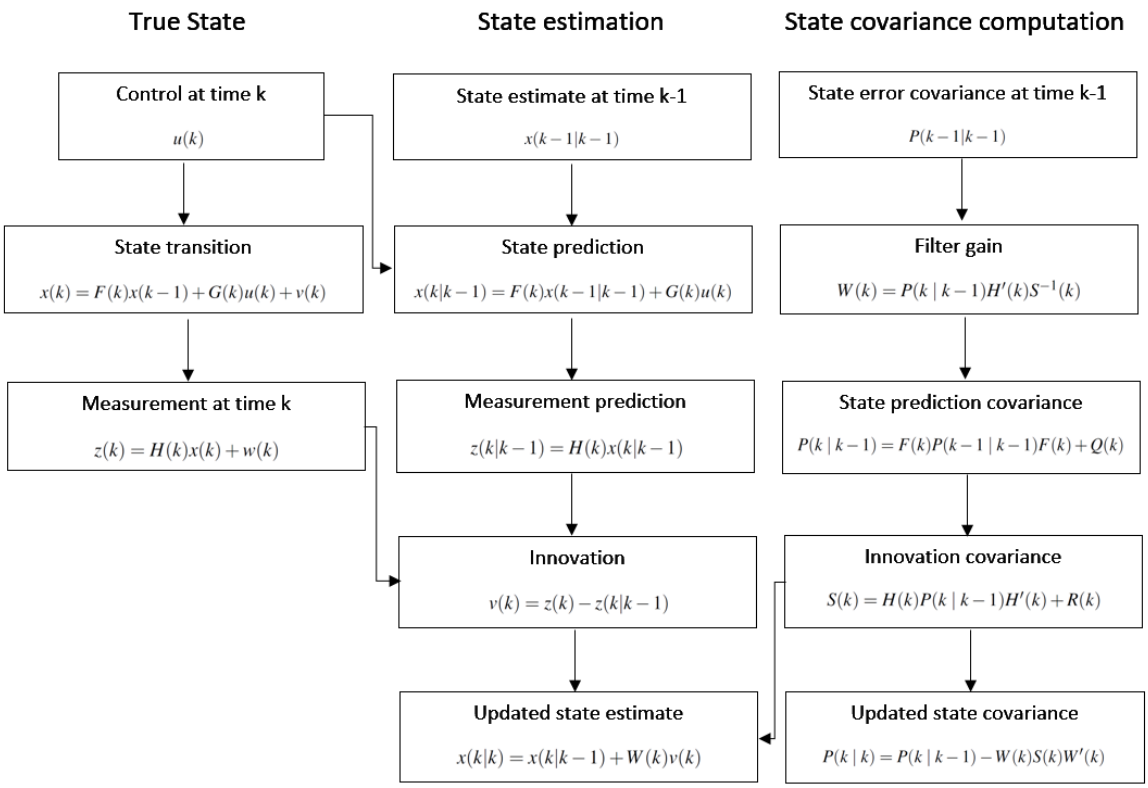


Fig. 2.8 Block diagram of the Kalman filter cycle [60]

process model and a non-linear measurement model and applies the first-order Taylor series approximation to linearise both models [75].

2.6.2 Extended Kalman Filter

The EKF has several features which make it ideally suited when dealing with complex multi-sensor position estimation. The explicit description of processes and observations allows a wide variety of different sensor models to be incorporated within the basic algorithm [120]. Also, the consistent use of statistical measures of uncertainty makes it possible to quantitatively evaluate the role that each sensor plays in overall system performance [131].

In multi-sensor data fusion, a group of sensors can be considered as a single sensor with a large and possibly complex observation model [132]. This approach may be limited to relatively small numbers of sensors that can be grouped [87]. A second approach is to consider each position observation made by each sensor as independent according to a specific observation model. The observations can be sequentially incorporated into the estimate.

Single-sensor estimation techniques can be applied to the sequential formulation of the multi-sensor estimation [111]. However, this sequential approach requires that a new prediction and gain matrix be calculated for each observation from each sensor at every time step, and so it is computationally expensive for real time applications [121]. Another approach is to explicitly derive equations for integrating multiple sensor position measurements made at the same time into a common state estimate. Starting from the formulation of the multi-sensor EKF algorithm and employing a single model for a group of sensors. Therefore, a set of recursive equations for integrating individual sensor observations can be derived [123].

2.6.3 Error State Extended Kalman Filter

A critical aspect of the UAV implementation of a data-fusion filter in conjunction with an INS is the use of the direct (EKF) or the indirect approach, also referred to as Error State Extended Kalman Filter (ESEKF) [137]. In the direct formulation, total states such as orientation and position are among the variables in the filter, such as inertial measurements from the INS. The correction of the states come from external source signals such as the position estimates from VSLAM [13].

However, there are some severe drawbacks to EKF implementation. Especially in localisation applications, the filter may present inconsistencies in the position estimates caused by accumulated errors in the linearisation process [13]. An EKF implemented in the navigation system loop of a UAV, the filter has to maintain explicit, accurate awareness of the vehicle's angular motion. For instance by incorporating the dynamic model of the UAV which may not be accurate, as well as attempting to suppress noisy and erroneous data at a relatively high frequency. [29].

In the case of the ESEKF, the errors in orientation and position are among the estimated variables, and each measurement presented to the filter is the difference between the inertial measurements of an IMU and position estimates from external sensors such as a stereo camera or relative localisation. In the case of a failure in the position estimate of an external sensor, the ESEKF can continue to provide estimates by acting as an integrator of the inertial measurements of the IMU [29, 60]. The IMU itself can follow the high-frequency motions of the vehicle very accurately, and there is no need to model these dynamics explicitly in the ESEKF compared with EKF approaches [74, 20].

ESEKF uses two states: the estimated state $\hat{x}$ which is the estimation of the true state x, and the error state $\tilde{x}$ which is the difference between the true state and the estimated state $\tilde{x} = x - \hat{x}$ [10]. The ESEKF determines the error recursively, and then uses the error to calculate the corrections which must be performed on the estimated state [147, 132].

In summary, the ESEKF allows for a continuous vehicle position estimation by integrating the IMU inertial measurements with correcting sensors such as a stereo camera or relative localisation. Practical applications of this type of solution to real vehicles with 6DoF can be found in a UAV's navigational system, where all available sensors aid in the position estimation [138].

2.7 Summary

This chapter has summarised recent research literature and technology relevant to the application of UAVs for mapping outdoor unstructured GPS-denied environments. The topics reviewed include the basic concepts of UAVs and their limitations, such as limited payload capacity, processing power and battery life, as well as SLAM and its recent implementations in outdoor environments and its challenges. These challenges include the increasing computational complexity of the algorithm with the number of observed landmarks and errors in data association which may cause drift in the position estimates of SLAM. This chapter has also detailed the most recent methods that use swarms robotics to solve the data-association problems of SLAM. For example, the cooperative behaviour of a swarm can aid in accurate localisation of every agent by fusing all available measurements of the system, and ultimately improving the position estimation of the system mapping a large outdoor area.

From the literature review, the following major research challenges have been identified:

Outdoor GPS-denied mapping and robustness against failures. UAVs navigating in GPS-denied environments must reason over, and act under, the uncertainty about the state of the environment. Making matters worse, the low payload capacity of UAVs limits the sensors and the processing power that can be carried on board. It is not hard to envision a single UAV performing several runs to thoroughly explore and map an unstructured environment without the need for a GPS. However, if this single UAV gets lost or damaged in the environment, the whole mapping mission may fail.

Cooperative decentralised systems. The key challenges in a swarm robotic system are sharing information about the environment generated by individual UAVs and coordinating the members in the system. The underpinning architecture needed to perform these tasks determines the applicability of the swarm to helping each member navigate the unstructured outdoor environment. In the research literature, CoSLAM implemented in UAVs relies heavily on a centralised architecture to share and fuse individual map estimates into an accurate global map [118]. A centralised architecture requires a reliable communication network, often depending on external infrastructure such as anchors. This limits the usability of such systems in applications like outdoor mapping. Moreover, centralisation reduces the scalability and robustness of the system against failures of a single member or node.

CoSLAM for unstructured outdoor environments. CoSLAM has been proposed to reduce the loop-closure computational complexity, improve the exploration efficiency and range in unstructured environments. In CoSLAM, a swarm of UAVs share partial maps to all the members in order to be merged by every UAV, in order to create a global map and improve each member individual position estimates. However, communication links may be

sporadic and so sharing large amounts of data such as a map may prove difficult in real time implementations.

To address these research problems, this PhD proposes to develop a computationally lightweight system for relative localisation and to implement it in a swarm of UAVs for cooperative SLAM via an ESEKF. This would enable a swarm of UAVs equipped with stereo cameras and other localisation systems to efficiently and accurately estimate the position of each member of the swarm in order to navigate unstructured GPS-denied outdoor environments.

3. Relative Localisation for Swarms of UAVs

When a swarm of UAVs performs a complex task such as mapping, the system must be able to coordinate the movements of its members to avoid internal collisions with neighbouring UAVs or with external obstacles such as trees. To achieve the coordination of the swarm, it is necessary for each UAV to measure or estimate the relative positions between itself and its immediate neighbours in relation to common navigation frame or world frame [15, 76]. This is called relative localisation.

This chapter presents a relative localisation system, that is suitable for use by every UAV in the swarm. The system measures the angle and distance between two neighbouring UAVs (a measuring or sensing UAV, and a target UAV). These measurements are then used by the measuring UAV to estimate relative position of a neighbouring UAV in three-dimensional (3D) space. It will be further shown in Chapter 4 how this relative localisation system can then be used to reduce the drift in the position estimates of VSLAM, using an Error State Extended Kalman Filter (ESEKF).

The chapter begins by introducing the proposed approach for estimating the relative positions of UAVs in a swarm. This section includes the methods used to measure the angle and distance between UAVs even when the visual line of sight is temporarily broken. The platform design of the UAVs, as well as the hardware used to create a swarm for later experimentation, is described in Section 3.2. Section 3.3 describes the sets of experiments

performed to evaluate the accuracy of the relative localisation system in two settings: with and without visual occlusions. The results obtained from the experiments are discussed in Section 3.4. A summary of the chapter is presented in Section 3.5.

3.1 A Proposed Relative Localisation Method

For a swarm of UAVs to perform relative localisation in any environment, it is necessary for each UAV in the swarm (UAV_j, $j = 1, 2, 3, ..., N$) to estimate the position of a neighbouring UAV, UAV_{-j}, in relation to UAV_j's frame of reference or body frame (IMU frame). Note: the subscript $-j$ is used to denote a neighbouring UAV. In order for all members of the swarm to be able to share position estimates, UAV_j must rotate and translate the position estimates of UAV_{-j} to conform to the common navigation frame or world coordinate frame.

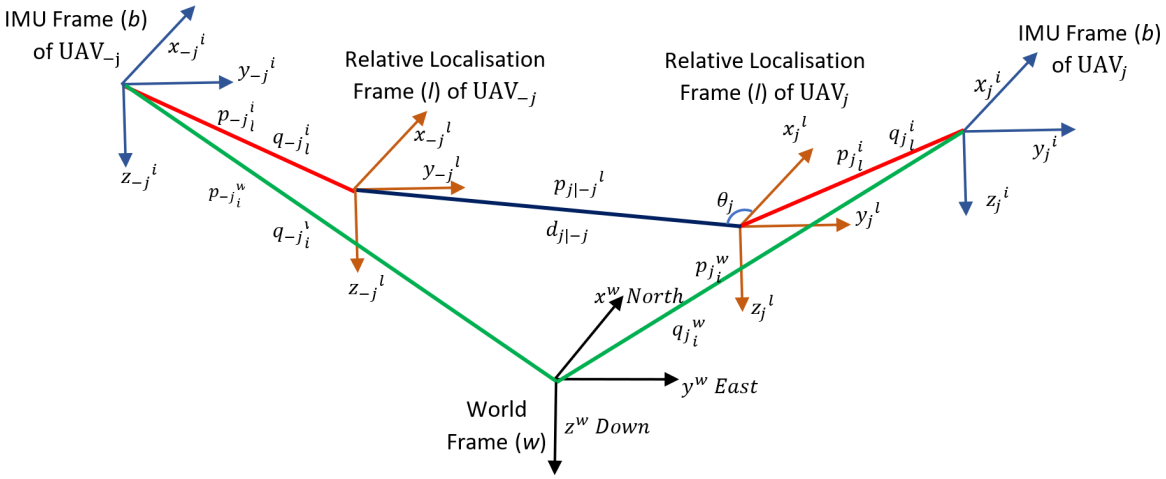


Fig. 3.1 Relative localisation coordinate frames.

The formulation of the relative localisation system is illustrated by the situation where UAV_j estimates the position of UAV_{-j} according to the world frame North, East, and Down (NED). Fig. 3.1 shows the three coordinate frames that comprise the relative localisation

system. These are: w, which represents the world coordinate frame, b, which represents the IMU frame or body frame of a UAV, and l, which is the coordinate frame of a UAV's relative localisation system. This figure also shows the position (p) and rotation (q) between the coordinate frames, ${p_j}_b^w$ denotes the position of the origin of UAV$_j$'s IMU frame expressed in the world frame, ${p_{j|-j}}^l$ denotes the relative position of UAV$_{-j}$ measured by UAV$_j$'s relative localisation system based on the angle θ_j, and distance $d_{j|-j}$ between the UAVs. The first step to developing a system that can perform relative localisation is to identify the moving IMU and relative localisation frames of each UAV as well as the transformations between them. The second step is to measure θ_j and $d_{j|-j}$ in order to estimate ${p_{j|-j}}^l$.

Relative localisation is performed as follows. A monocular camera placed on UAV$_j$ was used in combination with a colour tracking algorithm to estimate the position of UAV$_{-j}$ on an image as described in Subsection 3.1.1. Two ultra-wideband (UWB) radios were used to measure the distance between UAV$_j$ and UAV$_{-j}$ as described in Subsection 3.1.2. Using the position UAV$_{-j}$ on an image and distance measurements between UAVs, the angle θ_j can be calculated by using the geometric pinhole camera model. This angle was then be used to triangulate the real world position of UAV$_{-j}$ relative to UAV$_j$ body frame. In order to share the position estimates UAV$_j$ then uses the attitude or orientation estimations from its IMU (accelerometer, gyroscope and magnetometer) to transform the relative localisation position estimates of UAV$_{-j}$ with respect to the NED frame. This is described in Subsection 3.1.3.

When performing relative localisation, it is desirable to estimate the positions of the neighbouring UAVs even if obstacles temporarily break the visual line of sight between the UAVs. This can be achieved by using a Kalman Filter in the colour tracking algorithm. The filter predicts the positions of the tracked UAV in the image at a time step t_{k+1} by estimating the

velocity of the UAV at time t_k. The filter continues to predict the position of the UAV until the visual line of sight is recovered. This is described in Subsection 3.1.4.

3.1.1 Establishing a Visual Line of Sight for UAV Tracking

For UAV$_j$'s camera to track the position of UAV$_{-j}$ on an image, a passive coloured marker was attached to UAV$_{-j}$ to distinguish it from the environment. A colour that was not predominant in an unstructured outdoor environment was chosen in order that it could be easily detected and consequently tracked. The Camshift colour tracking algorithm [7] was used to track the coloured marker on UAV$_{-j}$. This algorithm is included as a part of the OpenCV set of libraries [93].

The Camshift algorithm is an adaptation of the Meanshift algorithm for colour tracking. It is a non-parametric method, robust to changes in the shape and lighting of a marker that is to be tracked [76]. The Meanshift algorithm is used to locate the maxima of a Probability Distribution Function (PDF), representing the marker's colour histogram or an image's histogram. Therefore, the Meanshift algorithm requires a colour histogram as input in order to convert the image into a probability distribution relative to the input colour's histogram. The Camshift algorithm builds on this method as it uses continuously adaptive probability distributions, which means the colour's probability distribution will be recomputed in every frame. This allows the position of the colour on the image can be computed even if the marker's size, shape and appearance change in every frame or image [31].

The Camshift Algorithm can be summarised in the following steps [7]:

1. Set a region of interest within the entire image, which contains the marker's colour in order to determine the PDF of interest.

2. Select an initial location of a search window. The selected location should be the target probability distribution (histogram) to be tracked.

3. Calculate the colour probability distribution (histogram) of the region centred at search window.

4. Iterate to find the centroid with the highest probability. Store the zero-*th* moment (distribution area) and centroid location.

5. For the following frame, centre the search window at the mean location found in Step 4 and set the window size to a function of the zero-*th* moment.

To determine the PDF, an initial histogram is computed from the colour in the image's region of interest. Although colour histograms are typically computed according to the RGB (red, green, blue) colour space, these are not robust to brightness changes, which change the colour tone. Therefore, the HSV (hue, saturation, value) colour space was used, which is robust to changes in lighting conditions. This makes it especially suitable for outdoor applications where lighting condition may change over time. The hue and saturation channels represent the main characteristics of any colour. Thus, they were used to calculate the input histograms [33]. The input histograms are quantised into bins, which reduces the computational complexity and allows similar colour values to be clustered together. The histogram bins are then scaled between the minimum and maximum PDF of the image intensities as:

$$pdf_u = min\left(\frac{255}{max(h_u)}h_u, 255\right), u = 1,2\ , \tag{3.1}$$

where h denotes the histogram, pdf denotes the probability distribution function, and the subscript $u = 1,2$ represents the hue and saturation channels respectively.

Histograms are rescaled from [0, $max(q_u)$] to the new range [0, 255], where the image pixels with the highest probability of being in the sample histogram will appear as visible

intensities in the image's histogram back projection. This is an operation that associates the pixel values in the image with the value of the corresponding histogram bin. The back projection of the target histogram with any consecutive frame generates a probability image where the value of each pixel characterises probability that the input pixel belongs to the histogram that was used [7, 76].

The mean location (centroid) within the search window (x_c, y_c) of the image's PDF, computed in Step 3, is the probable location of the UAV's colour marker projected on the image. This is found using the zero-th and first moments (M_{00}, M_{10}, M_{01}) of the search window as follows:

$$\begin{aligned} M_{00} &= \sum_{x_i}\sum_{y_i} I(x_i, y_i) \\ M_{10} &= \sum_{x_i}\sum_{y_i} xI(x_i, y_i) \\ M_{01} &= \sum_{x_i}\sum_{y_i} yI(x_i, y_i) \end{aligned} \tag{3.2}$$

$$x_c = \frac{M_{10}}{M_{00}} \quad ; \quad y_c = \frac{M_{01}}{M_{00}} \,,$$

where $I(x_i, y_i)$ is the intensity of the PDF of the image at (x_i, y_i) pixel location within the search window.

The Camshift algorithm continually recomputes new values of x_c, and y_c until there is no significant shift in their position. Because the limit of the accuracy is one pixel, a convergence criterion is set as a shift of one pixel in either the x or y axis of the image. A further criterion of convergence to limit the number of iterations. This was set at 10 to 20 iterations per frame in order to avoid infinite calculations of the centroid positions [7].

3.1.2 Ultra Wide-Band Radio Ranging

To measure the distance between both UAVs, the ranging system was constructed using the Decawave DW1000 radio and the Mulle wireless sensor platform. The DW1000 radio is an IEEE802.15.4-2011 UWB wireless transceiver. It can measure the distance between itself and another DW1000 radio at an accuracy of $+/-0.10$ m over a distance up to 300 m within line of sight, and with a resolution of 0.00001 m [32]. This allowed the relative localisation system to measure the range between UAVs over a large distance accurately. The Mulle wireless sensor platform consists of two main components: a wireless sensor module and an internet gateway device. These allow the processing and transmitting the ranging measurements from the Decawave UWB radio directly to an embedded CPU.

The Mulle wireless sensor module is equipped with a microcontroller having an IEEE 802.15.4 radio (868 MHz radio frequency). The Mulle sensor node runs the open-source Contiki or RIOT operating systems. Both operating systems feature a full IPv6 (Internet Protocol version 6) stack [37]. This means that the Mulle platform allows the swarm robotic system to have a maximum size of 2^{128} members. The Mulle can be connected to the internet or a CPU via the gateway device. This contains an IEEE 802.15.4 radio and a USB port that allows for wired connections. The gateway contains an ARM Cortex A9 processor, which runs a full Linux system. Therefore, the Mulle platform can interpret and compute the distance information from the UWB radio in order to transmit the measurements to an embedded CPU [37].

The Decawave UWB radio supports two forms of distance calculations: Time Difference of Arrival (TDA), and Time of Flight (ToF). TDA uses three radios or nodes to calculate the distance between the nodes. The initial time of sending the initiating node or central node is not required to be known to the three-node system, only the time the signal was received by two target nodes, and the signal's speed. This means that the clocks in the three nodes

have to be synchronised. Once the signal is sent by the initiating node and received by the two targets, the time difference of arrival can be calculated in order to determine the distance through triangulation. On the other hand, ToF measures the time a signal takes to travel from the one initiating node to a target node and back to the initiating node. For this research, ToF was used for distance calculations as it only required a minimum of two radios to perform measurements, and also because the times of transmission and reception of each pair are only determined by the initiating node. This means that ToF measurements do not rely on clock synchronisation between nodes compared with the TDA method. This approach makes the relative localisation system robust to clock offsets and allows the swarm system to be decentralised.

The UWB radios can be set-up in two modes: as an anchor or as a tag. Anchors are actively listening/waiting for messages from any tag to start the ranging process. There are a total of six messages between the two nodes used in the ranging process: 1-blink, 2-init, 3-poll, 4-response, 5-final, and 6-measurement report [106]. The ranging process is listed below, and summarised in Fig. 3.2.

1. The tag starts in sleep mode, and when is required, it broadcasts a blink message searching for an anchor within range. (The anchor is always listening for messages). This blink is the discovery message to start the distance measurement.

2. Once the anchor receives a blink, the tag will begin a handshaking process by sending an init message.

3. The tag responds with a poll message at time T_{sp} and is received by the anchor at time T_{rp}.

4. The anchor then sends a response message at time T_{sr}, which is received by the tag at time T_{rr}

5. The tag, in turn, responds with a final message at time T_{sf}, which is received by the anchor at time T_{rf}.

6. The sixth and final message is a measurement report sent from the anchor to the tag. This message contains the ToF calculated from the four timed messages from T_{sp} to T_{rf} as well as the IP address or ID from tag, and the number of measurement (M_s $s = 1, 2, 3...N$).

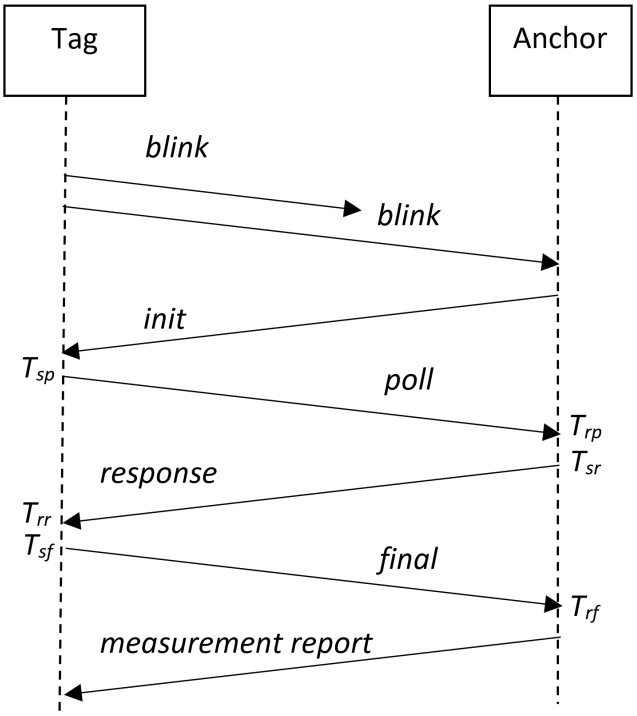


Fig. 3.2 UWB ranging procedure [106].

A tag can range with several anchors, and this process will generate measurement reports that contain the information from all the anchors visible to the tag. This allows the tag to keep a record of the anchors around it as well as the rates at which the measurements occur. However, having one tag and several anchors would make the relative localisation system centralised because only a tag can initiate the ranging process. Therefore, to create a decentralised system, in the experiments conducted for this research the tags and anchors

were setup to switch roles at given time intervals. This enabled every member of the swarm to initiate the ranging process with its immediate neighbours. Role switching also has the advantage of reducing the strain on the UAV's battery because it reduces the amount of time a UAV is listening for a blink message.

3.1.3 3D Location Based on the Common Navigation Frame

Once the distance and image data (location) of the target UAV have been obtained, the location of the target UAV in 3D space can be estimated. Fig. 3.3 shows the geometric pinhole camera model [44], which was used to triangulate the position of UAV$_{-j}$'s marker in 3D space. This geometric model is comprised of the distance (d_{UWB}) between UAVs measured by the Mulle-UWB sensor, the focal length of the camera lens (f), the pixel coordinates of UAV$_{-j}$'s marker projected onto the image plane (x_c, y_c), the angle (θ_j) between the optical axis and d_{UWB}, and the real world coordinates of UAV$_{-j}$'s marker according to the relative localisation frame (x_{rl}, y_{rl}, z_{rl}) [44]. The geometric pinhole camera model uses the location of the pixel coordinates of UAV's colour marker on the image plane (x_c, y_c) relative to the image's centre point (0,0) and the focal length (f) to calculate the angle (θ_1) between the optical axis and d_UWB. This allows the real world coordinates x_{rl}, y_{rl}, and z_{rl} of UAV$_{-j}$'s marker to be determined as:

$$\begin{aligned} \theta_j &= arctan\left(\frac{y_c}{x_c}\right) \\ y_{rl} &= d_{UWB}\ \sin\theta_j \\ z_{rl} &= \frac{d_{UWB}\ \sin\theta_j\ f}{y_c} \\ x_{rl} &= \frac{x_c}{f} z_{rl}\ . \end{aligned} \tag{3.3}$$

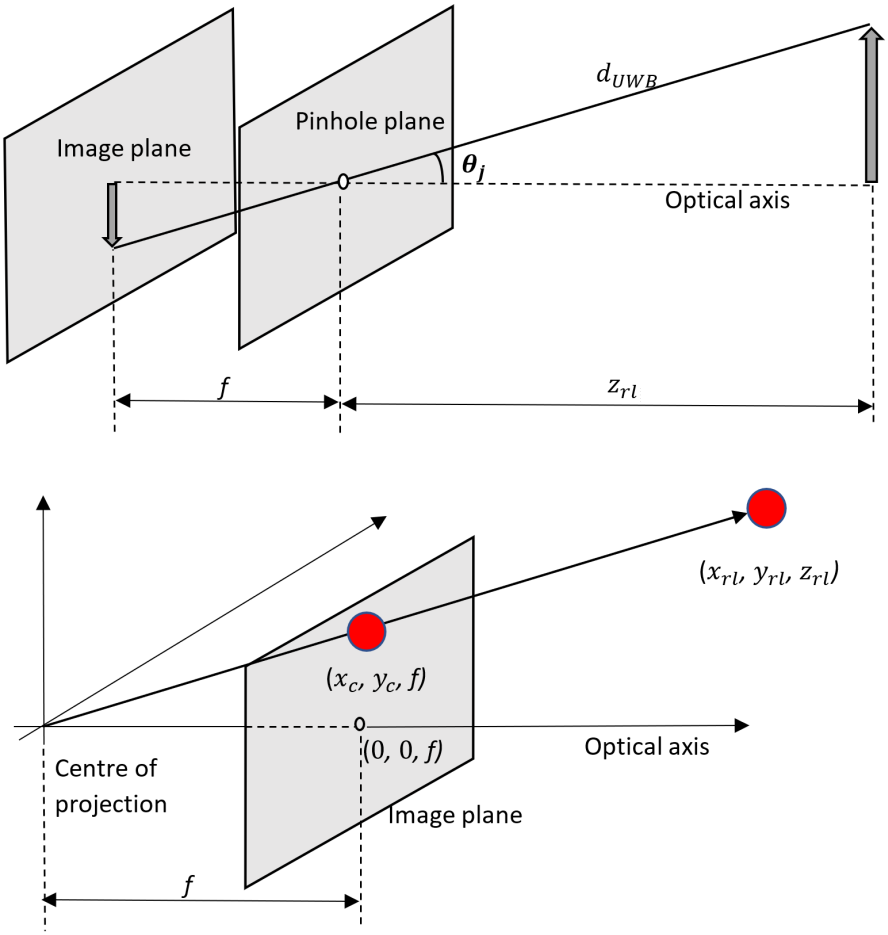


Fig. 3.3 Geometric pinhole camera model [44].

The 3D position calculations (${p_{j|-j}}^{l}(x_{rl}, y_{rl}, z_{rl})$) obtained using the relative localisation system are highly sensitive to the change of the direction in which the camera is pointing. Small changes in the attitude (pitch, roll or yaw) of UAV$_j$ will modify the position estimates of the relative localisation system. However, rotating the position measurements with respect to the NED frame enables the system to estimate the position UAV$_{-j}$ relative to common coordinate frame. This is achieved by determining the rotation of the l frame expressed in the w frame (${q_{jl}}^{w}$). From Fig.3.1 it can be seen that the rotation of the l frame, with respect to the w frame is comprised of the rotation ${q_{jl}}^{b}$ and the rotation ${q_{jb}}^{w}$. Also, it can be deduced that the l frame is fixed to the b frame, therefore the resulting rotation is defined as ${q_{jl}}^{b} = (1,0,0,0)$. The rotation between the b and w frames can be obtained through the UAV$_j$'s IMU attitude measurements in quaternions (${q_{jb}}^{w} = (q_{b_0}, q_{b_1}, q_{b_2}, q_{b_3})$). Therefore, ${q_{jb}}^{w}$ is equal to ${q_{jl}}^{w}$. This rotation can be implemented using the rotation matrix (Rot) to orient x_{rl}, y_{rl}, z_{rl} to the NED frame as follows:

$$\begin{bmatrix} x_N \\ y_E \\ z_D \end{bmatrix} = Rot \begin{bmatrix} x_{rl} \\ y_{rl} \\ z_{rl} \end{bmatrix} , \tag{3.4}$$

where

$$Rot = \begin{bmatrix} q_{b_0}^2 + q_{b_1}^2 + q_{b_2}^2 + q_{b_3}^2 & 2(q_{b_1} + q_{b_2} - q_{b_0} q_{b_3}) & 2(q_{b_1} + q_{b_3} - q_{b_0} q_{b_2}) \\ 2(q_{b_1} + q_{b_2} - q_{b_0} q_{b_3}) & q_{b_0}^2 + q_{b_1}^2 + q_{b_2}^2 + q_{b_3}^2 & 2(q_{b_2} + q_{b_3} - q_{b_0} q_{b_1}) \\ 2(q_{b_1} + q_{b_3} - q_{b_0} q_{b_2}) & 2(q_{b_2} + q_{b_3} - q_{b_0} q_{b_1}) & q_{b_0}^2 + q_{b_1}^2 + q_{b_2}^2 + q_{b_3}^2 \end{bmatrix} ,$$

and x_N, y_E and z_D are the coordinates of the position of UAV_{-j} in the NED frame.

3.1.4 Tracking the UAV when the Line of Sight is Temporarily Broken

The Camshift algorithm can estimate the position of a colour marker even if its size, shape and appearance change on the image. However, the position calculation of the search window's centroid (x_c, y_c) may be affected by the lighting conditions in the environment. Moreover, if the visual line of sight to the marker is temporarily broken, the algorithm can no longer be used to determine the position of the marker until the visual line of sight is restored. To overcome these potential limitations of the Camshift algorithm, the position estimates x_c and y_c were tracked using a Kalman Filter [76]. This filter predicts the positions of x_c and y_c based on the kinematics of the UAV's movement. The kinematics of the UAV projected on the image are described as:

$$\begin{aligned} x_{c_{t_{k+1}|t_k}} &= x_{c_{t_k|t_k}} + \dot{x}_{c_{t_k|t_k}} \Delta t \\ v_{x_{c_{t_{k+1}|t_k}}} &= v_{x_{c_{t_k|t_k}}} + n_x \\ y_{c_{t_{k+1}|t_k}} &= y_{c_{t_k|t_k}} + \dot{y}_{c_{t_k|t_k}} \Delta t \\ v_{y_{c_{t_{k+1}|t_k}}} &= v_{y_{c_{t_k|t_k}}} + n_y \ , \end{aligned} \tag{3.5}$$

and the dynamics of the system described as:

$$\begin{aligned} v_{x_{c_{t_k}}} &= \left(x_{c_{t_k}} - x_{c_{t_k-1}} \right) \Delta_t \\ v_{y_{c_{t_k}}} &= \left(y_{c_{t_k}} - y_{c_{t_k-1}} \right) \Delta_t \ , \end{aligned} \tag{3.6}$$

where n_x and n_y denote the noise in the measurement in the image's x and y axes respectively, Δtk is the time interval between position measurements, $t_k|t_k$ denotes the current measurement.

The state variables in Equation (3.5) are the position (x_c, y_c) and velocity (v_{x_c}, v_{y_c}) per axis. In order to estimate the position of the marker at time t_{k+1}, the filter has two steps: the prediction step and the correction step. The prediction step estimates the states at t_{k+1} from Equation (3.5) at t_k. The correction step then improves the estimation of the prediction step, whenever the position measurement and calculated velocity are available.

The prediction step can be expressed as:

$$\begin{aligned} \hat{x}_{t_{k+1}|t_k} &= F_{\Delta t}\hat{x}_{t_k|t_k} \\ P_{t_{k+1}|t_k} &= F_{\Delta t}P_{t_k|t_k}F_{\Delta t}^T + Q\,, \end{aligned} \tag{3.7}$$

where

$$\hat{x} = \begin{bmatrix} \hat{x}_c \\ \hat{y}_c \\ \hat{v}_{xc} \\ \hat{v}_{yc} \end{bmatrix} \quad ; \quad F_{\Delta t} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad ; \quad Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \sigma_{n_x}^2 & 0 \\ 0 & 0 & 0 & \sigma_{n_y}^2 \end{bmatrix}, \tag{3.8}$$

$\hat{x}$ is the estimated state, $\hat{x}_c$ is the estimated position of the coloured marker on the image x axis, and $\hat{v}_{xc}$ is the estimated velocity of the coloured marker. The state ($\hat{x}_{t_{k+1}|t_k}$) is predicted according to the process $F_{\Delta t}$ and the current predicted state $\hat{x}_{t_k|t_k}$. The process covariance $P_{t_{k+1}|t_k}$ is calculated according to the process $F_{\Delta t}$, and the process noise covariance Q.

The next step in the Kalman Filter formulation corrects the prediction by adjusting the predicted state according to the error between the predicted and measured state. This is achieved by:

$$
\begin{aligned}
K_{t_{k+1}|t_{k+1}} &= P_{t_{k+1}|t_k} H^T (H P_{t_{k+1}|t_k} H^T + R)^{-1} \\
\hat{x}_{t_{k+1}|t_{k+1}} &= \hat{x}_{t_{k+1}|t_k} + K_{t_{k+1}|t_{k+1}} (x_{t_{k+1}} - \hat{x}_{t_{k+1}|t_k}) \\
P_{t_{k+1}|t_{k+1}} &= (I_d - K_{t_{k+1}|t_{k+1}} H) P_{t_{k+1}|t_k} \,,
\end{aligned}
\tag{3.9}
$$

where

$$
H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad ; \quad R = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \,, \tag{3.10}
$$

K is the Kalman gain, H is the observation matrix, R is the measurement noise covariance matrix, $x_{t_{k+1}}$ is the measurement of x_c and y_c made by the CamShift algorithm at a time t_{k+1}, and I_d is the identity matrix. R is defined as the identity matrix because velocity is calculated based on the position of the UAV on the planar image. This means that both position and velocity measurements can be trusted equally in this situation.

3.1.5 Algorithm for the Proposed Relative Localisation Method

The complete process of determining the relative localisation of UAV$_{-j}$ is given in Algorithm 1, which is summarised as follows:

- The first step to determine the relative location of UAV$_{-j}$ based on UAV$_j$'s body frame is to measure the angle between UAVs. This is achieved by using the Camshift colour tracking algorithm to estimate the position of UAV$_{-j}$'s coloured marker on an image. The algorithm requires the hue and saturation histograms of the marker's colour in order to find its most likely location on the image (see Equation (3.2)). This step is performed in lines 3 through 10 of Algorithm 1.

- The second step is to reduce the noise in the position estimates from the camera as well as to predict the states ($\hat{x}$) of the marker on the image. This is achieved by implementing the Kalman Filter prediction step (Equation (3.7)), and the correction step (Equation (3.9)). This step is performed in lines 11 through 20 of Algorithm 1.

- The third step is to obtain the measurement report. This contains the measured distance ($d_{j|-j}$), UWB$_{-j}$'s IP address (ID), and measurement number (M_s). The relative localisation system uses this information to determine when a new measurement is obtained in order to triangulate the position of the marker (Equation (3.3)). The position estimate of UAV$_{-j}$ is then rotated to the NED frame (Equation (3.4)). This step is performed in lines 21 through 29 of Algorithm 1.

3.2 Hardware design

A prototype UAV was designed and built. This was then replicated to create a swarm system capable of performing relative localisation, and cooperative SLAM for experiments in following chapters. Each UAV was constructed from the following components:

- A glass fibre quadcopter frame (48 cm $\times$ 48 cm $\times$ 48 cm) was used as the flying platform to mount the sensors.

- Two ELP $1920x1080P$p 180° FoV cameras [6], which have a 2 MP sensor and can record video at 30 fps were used to determine the angle between the UAV and two of its nearest lateral neighbours.

- One Mulle-UWB ranging sensor measured the distance to a neighbouring UAV's Mulle-UWB ranging sensor at a rate of 4 Hz.

- A passive colour marker was used to differentiate the UAV from the environment.

Algorithm 1 Relative Localisation

Input: histogram (q_u) ($u = 1,2$, where 1=hue and 2=saturation), image $I(x,y)$, prediction process $F_{\Delta t_k}$, process noise covariance Q, measurement noise covariance R, focal length of the camera f, IMU's roll,pitch,yaw measurements in quaternions $(q_{bo}, q_{b1}, q_{b2}, q_{b3})$ and UWB measurement report (d_{UWB}, ID, M_s)

Output:North, East, Down position coordinates x_N, y_E, z_D of UAV_{-j}

$t_k \leftarrow 1$

loop

Colour Tracking

Select the colour of the marker in the image, to determine the hue and saturation histograms

$p_u = min\left(\frac{255}{max(q_u)} q_u, 255\right), u = 1,2$

$M_{00} \leftarrow \sum_{x_i}\sum_{y_i} I(x_i, y_i)$

$M_{10} \leftarrow \sum_{y_i}\sum_{y_i} xI(x_i, y_i)$

$M_{01} \leftarrow \sum_{y_i}\sum_{y_i} yI(x_i, y_i)$

$x_c \leftarrow \frac{M_{10}}{M_{00}}$

$y_c \leftarrow \frac{M_{01}}{M_{00}}$

Kalman Filter

$v_{x_{c_{t_k}}} \leftarrow x_{c_{t_k}} - x_{c_{t_k-1}} \Delta_t$

$v_{y_{c_{t_k}}} \leftarrow y_{c_{t_k}} - y_{c_{t_k-1}} \Delta_t$

$\hat{x}_{t_{k+1}|t_k} \leftarrow F_{\Delta t} \hat{x}_{t_k|t_k}$

$P_{t_{k+1}|t_k} \leftarrow F_{\Delta t} P_{t_k|t_k} F_{\Delta t}^T + Q_{t_k}$

if a new measurement $x_{t_{k+1}}$ is available **then**

$K(t_{k+1}) \leftarrow P_{t_{k+1}|t_k} H^T (H P_{t_{k+1}|t_k} H^T + R)^{-1}$

$\hat{x}_{t_{k+1}|t_{k+1}} \leftarrow \hat{x}_{t_{k+1}|t_k} + K(t_{k+1})(x_{t_{k+1}} - \hat{x}_{t_{k+1}|t_k})$

$P_{t_{k+1}|t_{k+1}} \leftarrow (I - K(t_{k+1})H) P_{t_{k+1}|t_k}$

end if

3D Position Estimation

Algorithm 1 Relative Localisation (continued)

if a new M_s is available **then**
  $\theta \leftarrow arctan(\hat{y}_{t_{k+1}}/\hat{x}_{t_{k+1}})$
  $y_{rl_{t_{k+1}}} \leftarrow d_{UWB}(t_{k+1}) * sin(\theta_j)$
  $z_{rl_{t_{k+1}}} \leftarrow d_{UWB}(t_{k+1}) * sin(\theta_j) f/\hat{y}_{t_{k+1}}$
  $x_{rl_{t_{k+1}}} \leftarrow \hat{x}_{t_{k+1}}/f * z_{rl_{t_{k+1}}}$
  Use $(q_0(t_{k+1}), q_1(t_{k+1}), q_2(t_{k+1}), q_3(t_{k+1}))$ to calculate the rotation matrix *Rot* as described in Equation(3.4)
  $[x_{N_{t_{k+1}}}, y_{E_{t_{k+1}}}, z_{D_{t_{k+1}}}]^T \leftarrow Rot(t_{k+1})[x_{rl_{t_{k+1}}}, y_{rl_{t_{k+1}}}, z_{rl_{t_{k+1}}}]^T$ Rotate according to ID
end if
$t_k \leftarrow t_k + 1$
end loop

- A Pixhawk autopilot [105] was used to stabilise the UAV in the air as well as to use its IMU (accelerometer, gyroscope and magnetometer) to determine the attitude of the UAV with respect to the NED frame.

- An ODROID-XU4 CPU [55] was used to compute the relative localisation estimates.

- An external IMU (UM7) [24] was used to measure the linear accelerations and angular velocities of the UAV at a 100Hz (described in Chapter 4).

- A Stereolabs ZED stereo camera [126] was used to capture the images of the environment in order for the VSLAM algorithm from the ZED Software Development Kit (SDK) to estimate the position of the UAV (described in Chapter 4).

- An Nvidia TX1 Graphical Processing Unit (GPU) [92] was used to implement the ZED SDK VSLAM algorithm and process the images from the ZED stereo camera (described in Chapter 4).

- A SwiftNav Real-time Kinematic differential GPS (RTK-DPS) [127] was used to determine the position of the UAV in an outdoor environment. This RTK-DGPS is comprised of a GPS antenna and an RTK-DGPS receiver to process the signals from the satellites (described in Chapter 5).

- A Wi-Fi dongle was used to remotely initialise the system through a ground station via Secure Shell.
- A radio transmitter that was used to control the flight of the UAV manually.
- Four motors, four propellers and four Electronic Speed controllers for propulsion of the UAV.
- A 6200 mAh lithium polymer battery was used to power the whole system including sensors, the processing units, ESCs and the motors.

The connections between the sensors, CPU, GPU, ESCs, and radio manual control are shown in Fig.3.4. The fisheye cameras were calibrated in order to estimate their intrinsic parameters: the focal length, skew, distortion, and the image centre. This enabled the fisheye lens distortion to be corrected in order for the relative localisation system to use the geometric pinhole camera model. Because of the limited space on the UAVs, all the equipment had to be mounted in a 10 cm $\times$ 10 cm $\times$ 10 cm volume. To reduce the interference between the Mulle-UWB and ODROID-XU4, and between the GPS antenna and the RTK-DGPS receiver, a ferrite choke was placed on each communication line. This suppressed high frequency noise and maintained the functionality. With all the equipment placed on the UAV, the weight of the system was approximately 2.6 kg. This allowed a maximum flight time of 2 minutes, enabling a UAV to travel approximately 80 meters forward at a speed of 1.4 m/s for the experiments described in Chapter 5.

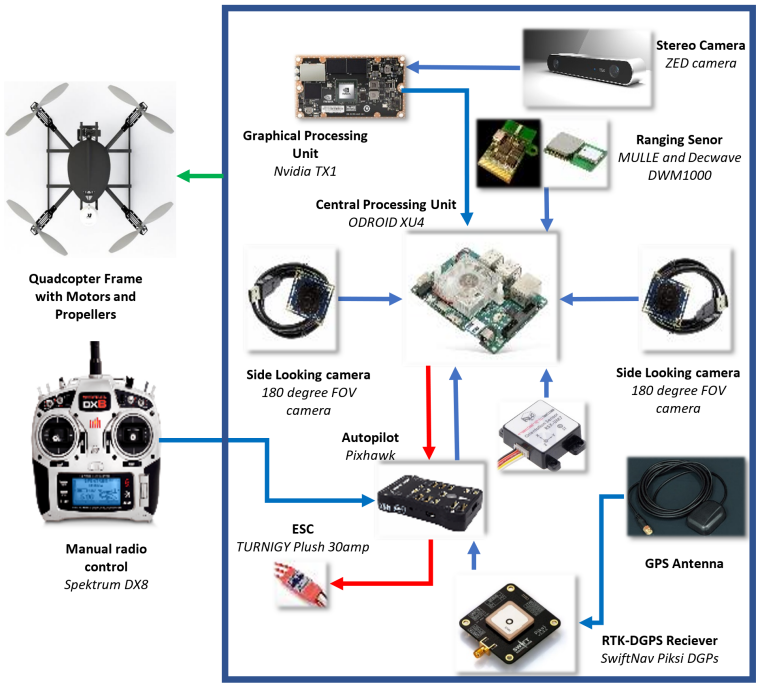
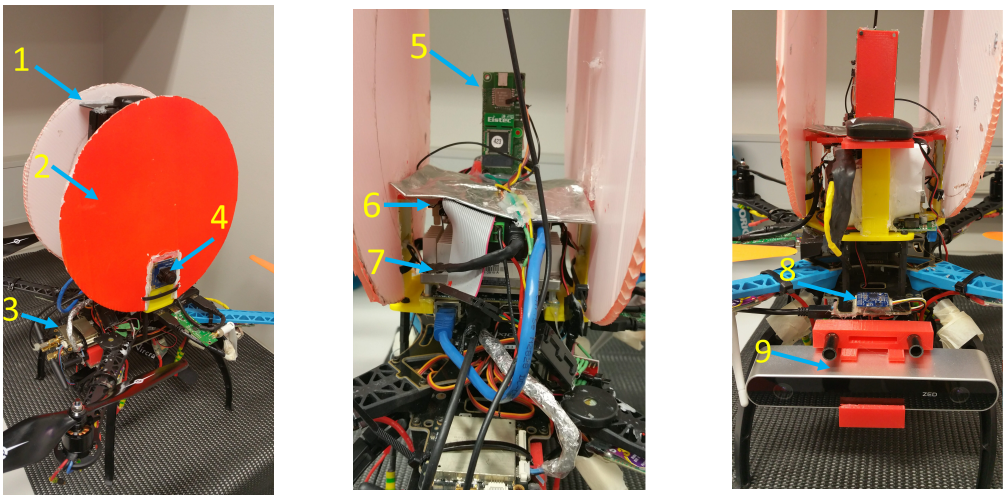


Fig. 3.4 Connections between the sensors, CPU, GPU, ESCs, and radio manual control.



(a) Side view (b) Back view (c) Front view

Fig. 3.5 Sensors mounted on all UAVs, where 1. GPS antenna, 2. Marker, 3. RTK-DGPS receiver, 4. 180° Field of view camera, 5. Mulle-UWB, 6. ODROID-XU4, 7. NVIDIA TX1, 8. IMU (UM7), 9. ZED stereo camera

3.3 Design of Experiments

To determine the accuracy of the relative localisation system, two set of experiments were conducted. Each used two UAVs: a measuring UAV (UAV_1), and a target UAV (UAV_2). The first experiment was performed in order to determine the accuracy of the relative localisation system position estimates without occlusions. The second experiment was performed to determine the accuracy of the relative localisation system position estimates when occlusions temporarily broke the visual line of sight of the 180° field of view camera. This second experiment simulated the situation when the UAVs pass through trees. Each experiment consisted of 15 trials, conducted in an indoor environment with constant lighting.

In each trial, the Root Mean Square Error (RMSE) was used to quantify the deviation of the UAV estimated position from the externally measured true position as determined by a ground truth motion capture system. This metric is also referred to as the absolute trajectory error as it measures the overall consistency of the estimated trajectory of the UAV compared with the ground truth measured trajectory. This method takes as inputs the estimated trajectory, $\hat{P}_1,...,\hat{P}_N$, and the ground truth trajectory measurements, $P_1,...,P_N$, of UAV_2. The RMSE calculation requires both set of measurements to be aligned to the same coordinate frame, time synchronised, with samples of equal length. The RMSE is calculated as [21]:

$$RMSE = \sqrt{\frac{1}{N}\sum_{tk=1}^{N}(P(x,y,z)_{tk} - \hat{P}(\hat{x},\hat{y},\hat{z})_{tk})^2} \tag{3.11}$$

In both experiments, the position of UAV_2 was estimated according to the measurements made by UAV_1. These measurements were then represented in the world frame. UAV_2 could move in all directions (6-DoF) while UAV_1 remained stationary and its bearing aligned with the world frame. To calculate the RMSE, the true positions of UAV_2 were measured with

a motion capture system (Optitrack [94]), to give ground truth position measurements. In all trials, the calculations of the relative localisation system were performed in real time by UAV_2's ODROID-XU4 CPU. The environment where the experiments were performed had a volume of approximately 84 m^3. The world frame in all experiments was oriented according to the Longitudinal, Lateral, Height coordinate system (see Fig. 3.6), which in this case was aligned to the North, East, and Down (NED) frame.

Before the start of the experiments, the internal clock on UAV_1 and the motion capture system's clock were synchronised by using a network time protocol (NTP) server running on UAV_1. This clock synchronisation protocol allowed the clock in the motion capture system to converge to a set time in the NTP server. The clock synchronisation between UAV_1 and the motion capture system ensured that all position estimates were recorded on the same time frame. For the image captured at the start of the tests, a region of interest that contained the colour of the maker on UAV_2 was selected by an operator. This was then used to determine the hue and saturation histograms required by the CamShift algorithm.

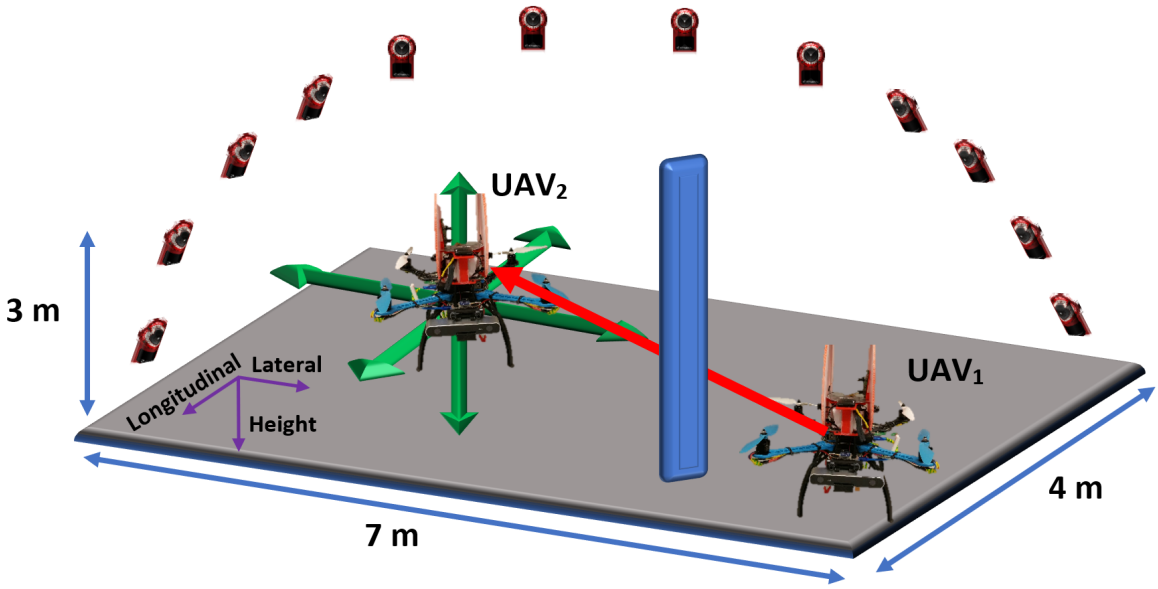


Fig. 3.6 Experimental set-up for the relative localisation system with occlusions.

3.4 Results and Discussion

Fig. 3.7a and 3.7b show one representative trial for each experiment (with and without occlusion respectively). The figures show UAV_2's estimated trajectory along each axis by the relative localisation system in UAV_1. The figures also show UAV_2's trajectories recorded by the motion capture system. It can be seen in the figures that the relative localisation system can estimate the trajectory of UAV_2 in every direction, even if a temporary occlusion occurs. It can also be noticed in the figures that the relative localisation system's estimated trajectories of UAV_2 in the Longitudinal and Height axes are smoother than the estimated trajectory in the Lateral axis. In the Lateral axis, the trajectory estimated by the relative localisation system shows small oscillations along the true trajectory measured by the motion capture system. This is because the Longitudinal and Lateral axes are approximately aligned with the North and the East axes. This means that the relative localisation frame and the world frame are aligned and that no rotation of the estimated positions was necessary.

From Equation (3.3) it can be noted that the calculated angle (θ_j) is more important in the estimation of UAV_2's position in Longitudinal and Height axes than the distance measurements. Also, Equation (3.3) shows that the angle calculations are based on the Kalman Filter's position estimate of UAV_2 on the image ($\hat{x}_c$,$\hat{y}_c$), which explains the smoothing of the estimated trajectories. This figure also shows that the Kalman Filter cannot follow fast changes in the trajectory of UAV_2's in the Longitudinal and Height axes compared to the relative localisation system's estimated trajectory in the Lateral axis. This is because the the distance measurement, which is more accurate than the estimation of $\hat{x}_c$ and $\hat{y}_c$, is more important than the angle measurement in the estimation of the position in the Lateral axis.

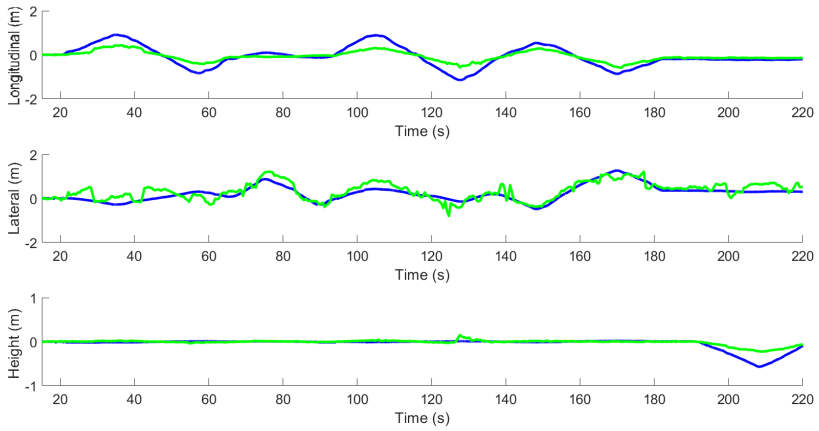
Fig. 3.7 shows that the error in the relative localisation system's estimated trajectories of UAV_2 do not increase significantly when the visual line of sight is temporarily broken, compared with the estimated trajectories when the visual line of sight is unimpeded. It is shown in the figures that even though the error in the trajectory estimates from the relative localisation system increases when the visual line of sight is broken, the trajectories are still being estimated by the system. Fig. 3.8a and 3.8b show the error between UAV_2's actual and estimated trajectories. In Fig. 3.8 it is noticeable that the error in the trajectory estimates increase when UAV_2 is temporarily occluded. The increase in the error is greater in the Longitudinal and Lateral axes than in the Height axis. This is because the movement of UAV_2 was kept at a constant height during the trial. From the error plots, it can also be seen that the error in the estimated lateral trajectory oscillates with a smaller magnitude than in the estimated longitudinal and height trajectories.

Table 3.1 presents the average RMSE of the relative localisation system's trajectory estimates of UAV_2 over the 15 trials in each experiment. The 15 trials without occlusions yield an average RMSE of 0.38 m, 0.18 m, and 0.26 m in the Longitudinal, Lateral, and Height

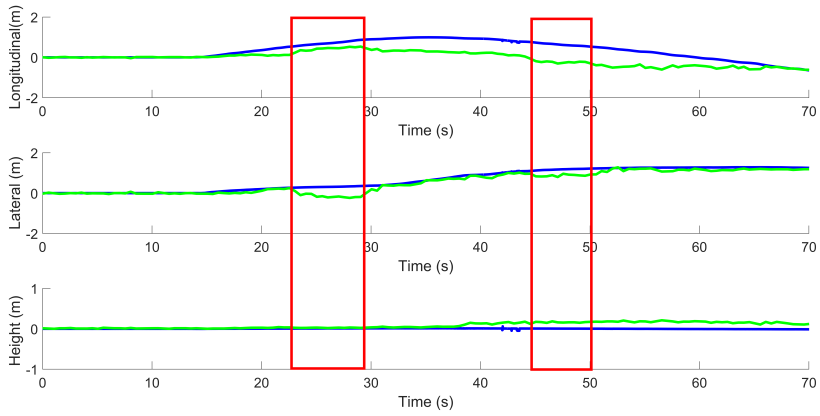
axes respectively. These show that the relative localisation system's estimated trajectories of UAV_2 in the Longitudinal and Height axes have lower accuracy than the estimated trajectory in the Lateral axis. The 15 trials with occlusions yield an average RMSE of 0.45 m, 0.20 m and 0.27 in the Longitudinal, Lateral, and Height axes respectively. These show the degradation of the accuracy of the relative localisation system's trajectory estimates when the UAV_2 is temporarily occluded. This are consistent with the results in Fig. 3.7 and 3.8, for a single trial, and mean that the relative localisation system can estimate the trajectory of a UAV to reasonable level of accuracy even when the visual line of sight is temporarily broken.

It can be observed from Fig. 3.7, 3.8, and Table 3.1 that when an occlusion is present, the worst degradation of accuracy happens in the Longitudinal axis. This is because the Kalman Filter predicts the position $(\hat{x}_c, \hat{y}_c)$ of UAV_2 according to the measurements of its last position $(\hat{x}_{c_{t_k|t_k}}, \hat{y}_{c_{t_k|t_k}})$ and velocity $(\hat{v}_{x_{c_{t_k|t_k}}}, \hat{v}_{y_{c_{t_k|t_k}}})$ before the occlusion occurs. This means that when UAV_2's marker is occluded, the estimated velocity is assumed to be constant as in Equation (3.5). It can also be seen that if the navigation trajectory of the swarm is aligned with the world frame, the position estimates in the Longitudinal and Height axis are mainly calculated by the CamShift algorithm and the Kalman Filter, with the Mulle-UWB ranging sensor mainly responsible for calculating the position estimates in the Lateral axis.

Fig. 3.9 shows the distribution of the RMSE of the relative localisation system's position estimates of UAV_2 from the 15 trials for each experiment. It can be seen in the figure that the RMSE is approximately normally distributed in both experiments an in all axes. These distributions yield a standard deviation in the experiment without occlusions of 0.003 m, 0.002 m, and 0.005 m in the Longitudinal, Lateral, and Height axes receptively. In the experiment with occlusions yield a standard deviation of 0.008 m, 0.004 m, and 0.007 m in the Longitudinal, Lateral, and Height axes receptively. These results show that errors in all



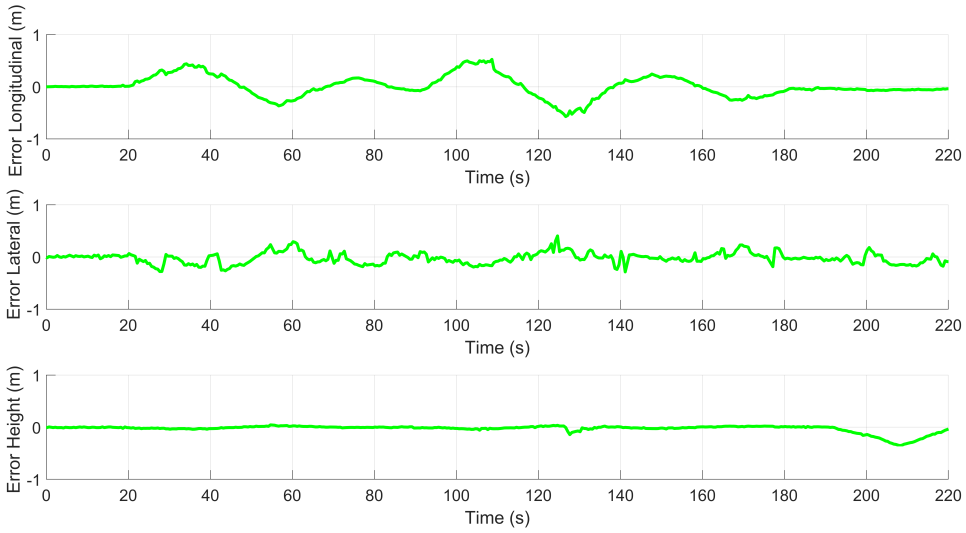
(a) No Occlusions present



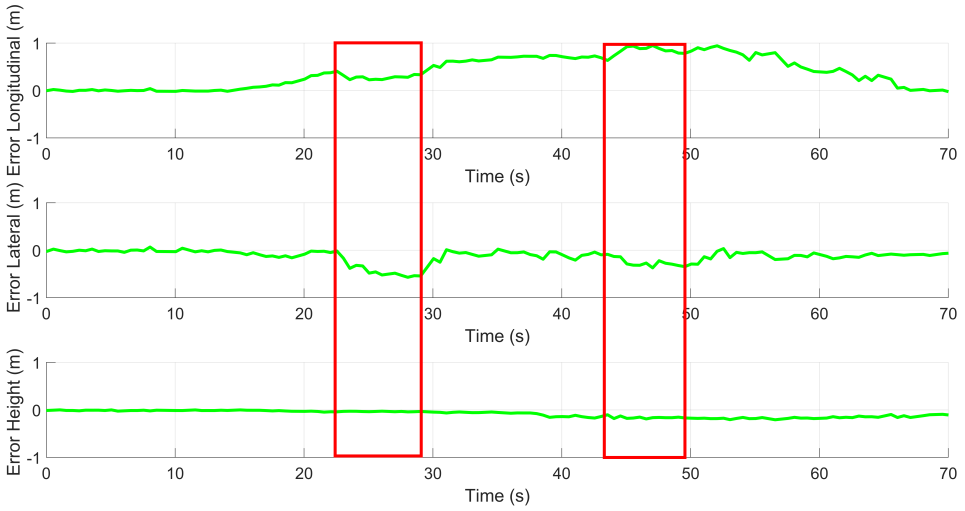
(b) Occlusions present

Fig. 3.7 UAV_2 actual and estimated trajectories. Blue: Optitrack trajectory measurements. Green: relative localisation estimated trajectories. Red Box: Presence of an occlusion.

axes, with and without occlusions are relatively consistent within this experimental setting.



(a) No Occlusions present



(b) Occlusions present

Fig. 3.8 Error in the relative localisation system's estimated trajectories. (a) Without occlusions. (b) With occlusions Red Box: Presence of an occlusion.

Axis	Average RMSE without occlusions (m)	Average RMSE with occlusions (m)
Longitudinal	0.3811	0.4532
Lateral	0.1845	0.2006
Height	0.2560	0.2667

Table 3.1 Average RMSE of the relative localisation system's estimated trajectories, with and without occlusions.

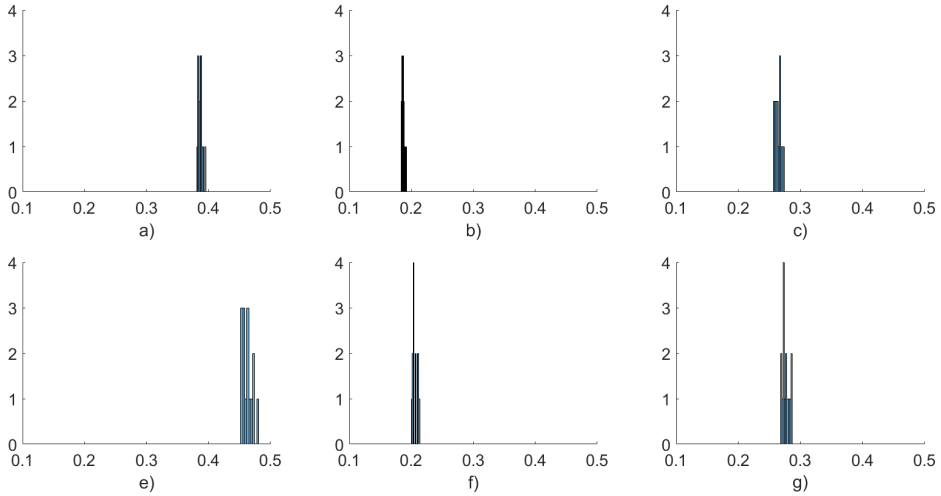


Fig. 3.9 Distribution of the RMSE a) Longitudinal axis without occlusions; b) Lateral axis without occlusions; c) Height axis without occlusions; d) Longitudinal axis with occlusion; e) Lateral axis with occlusions; f) Height axis with occlusions.

3.5 Summary

This chapter has described the development and testing of the proposed relative localisation system for UAVs. The system estimates the position of a neighbouring UAV (UAV_2) according to the angle and distance measurements of a UAV (UAV_1). The measured relative position is then transformed into the world frame. Two types of experiments with 15 repetitions each were performed in an indoor environment while the ground truth was measured by the motion capture system. The first experiment comprised of UAV_1 estimating the position of UAV_2 without any occlusions. The second experiment comprised of UAV_1 estimating the position of UAV_2 when an obstacle temporarily broke the visual line of sight between UAVs.

The results from these experiments show that the relative localisation system estimates the trajectory even when the visual line of sight is temporarily broken to a reasonable level of accuracy. The estimated trajectories of the relative localisation system have a greater error in Longitudinal and Height axes than in the Lateral axis. Therefore, this relative localisation system is an effective option when applied to a swarm of UAVs that fly in a linear formation. This is because the estimation of the position in the Lateral axis is important in order to avoid collisions between members. However, this is only the case when the relative localisation frame is aligned with the NED frame. Consequently this approach was taken in the experiments performed in the following chapters.

4. Error State Extended Kalman Filter for Cooperative Sensor Data Fusion

Choosing an appropriate position estimation method is one of the main challenges when developing a UAV's navigation system in order to explore GPS-denied environments. This is because there is not one method (dead reckoning or VSLAM) that alone can meet the accuracy required for a UAV to navigate such environments. Therefore, Error State Extended Kalman Filters (ESEKF) have been used by previous researchers to fuse the position estimates from all the available sensors in a UAV in order to improve the accuracy of the UAV's position estimates. Moreover, if a swarm system comprised of UAVs is used, each UAV in the swarm can take advantage of cooperation to share its position estimates with neighbouring UAVs. For example, by using the relative localisation system described in Chapter 3.

This chapter presents the design and implementation of a cooperative ESEKF to reduce the drift in the position estimates of the VSLAM algorithm. The chapter begins by describing the use of Kalman filters applied to data fusion. Section 4.2 explains the motivation to develop a cooperative ESEKF. Section 4.3 describes the mathematical design of the proposed ESEKF. The design of experiments in an indoor environment is given in Section 4.4. The results from the experiments are discussed in Section 4.5. Section 4.6 summarises the finding from this chapter.

4.1 Kalman Filter

One efficient way to reduce the drift in the position estimates of VSLAM is to fuse them with the position estimates of other available sensors. For example, In the Kalman Filter formulation, it is possible to relate position estimates from multiple sensors to a vector of internal states containing the parameters of interest, such as position, velocity, and acceleration [17]. The basic idea of using a Kalman Filter to fuse the position estimates of several sensors is to increase the trust in the sensors measurements where they each perform best. Therefore, the system provides more accurate and stable position estimates than a system based on one sensor alone. Although, the Kalman Filter can fuse the position estimates from different sensors, the UAV system has to be modelled linearly, which is not always feasible. Thus, the Extended Kalman Filter (EKF) has been derived. This filter uses the multivariate *Taylor Series* expansions, to linearise the state model. However, the Taylor series expansion is a poor approximation of most non-linear functions. The accuracy of the linearisation depends on two factors: the degree of uncertainty and the amount of local non-linearity in the functions being approximated. Therefore, the EKF is only as good as its approximation about the mean of the estimated state [15].

Another formulation of the EKF is the Indirect Extended Kalman Filter or Error State Extended Kalman Filter filter (ESEKF) shown in Fig. 4.1. This filter was devised to address the Markov assumption used in the EKF that the current estimate encapsulated the current state, and all of the previous states of the system. Therefore, the EKF assumed that the measurements from the sensors or IMU estimated the state of the system exactly, that is, without error or bias [84]. However, the error is a part of the state of the system [84]. Thus, in the error state space (indirect) formulation, the errors in the indicated position and velocity

are among the estimated variables. This means that the ESEKF continually estimates the error based on sensor measurements to correct the inertial position estimates [117].

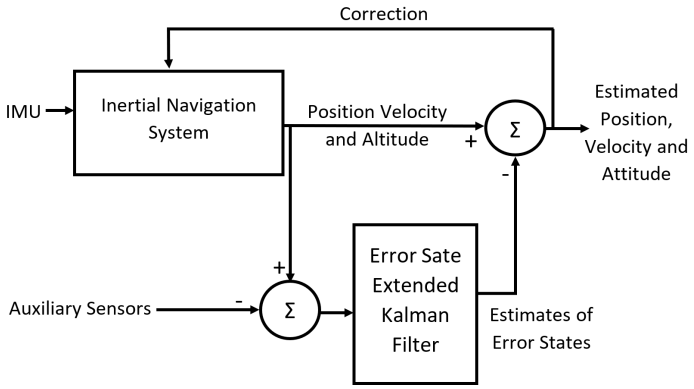


Fig. 4.1 Representation of the ESEKF [117].

4.2 Motivation for a Cooperative Error State Extended Kalman Filter

UAVs have been equipped with IMUs to measure linear acceleration and angular velocity in order to estimate the position of the UAV. These measurements are performed at a high refresh rate to follow the fast changing movements of the UAV. However, these position estimates drift exponentially over time due to noise in the measurements. Therefore, VSLAM algorithm and estimate the position of the UAV within an environment. Although the position estimates of the VSLAM algorithm are more accurate than the position estimates from the IMU, its refresh rate is slower. Moreover, without loop closure, the position estimates of the VSLAM can drift due to error accumulation and can grow unboundedly with time.

In this situation, an ESEKF can be implemented to fuse the position estimates from each sensor (stereo camera and IMU). This allows the correction of the drift in the position estimates from the IMU with the position estimates of the VSLAM algorithm. This is done

while maintaining a high refresh rate of the estimation. However, this correction alone is insufficient as the position estimates by the VSLAM still drift over time, meaning that the correction performed on the inertial position estimate comes from an erroneous source. One method to reduce the positional drift of VSLAM is to use a swarm of UAVs, which can perform cooperative relative localisation. This allows the ESEKF to reduce the drift in each UAV's VSLAM position estimates. It is performed by implementing the ESEKF in each UAV, which enables the system to perform cooperative SLAM without map sharing.

4.3 Cooperative Error State Extended Kalman Filter Design

The filter described in this section is an ESEKF, which fuses the position estimates from an IMU, VSLAM algorithm and relative localisation system. The ESEKF requires two states of the UAV's motion: the estimated state ($\hat{x}$), which is the prediction of the true state (x); and the error state ($\tilde{x}$), where $\tilde{x} = x - \hat{x}$. To define the states of the swarm system, the coordinate frames of relevant sensors on a UAV must be defined, as well as the kinematics of the UAV. Therefore, five frames: World, IMU, Vision, Stereo camera, and Relative Localisation are introduced as shown in Fig. 4.2.

The design of the ESEKF is based on the work performed by Weiss et al. [138], Post et al. [102], Trawny et al. [131], and Maybeck et al. [84]. These works refer to the formulation of an ESEKF to estimate the position and attitude of a UAV by using an IMU and VSLAM. The contribution of this section is the fusion of a relative localisation system's position estimates (as develop in Chapter 3) with the inertial measurements of an IMU and the position and rotation estimates of the VSLAM. This allows the ESEKF to correct the drift of the position estimates of each UAV's VSLAM algorithm with the position estimates from the relative localisation system in a neighbouring UAV. Thus, creating a cooperative ESEKF.

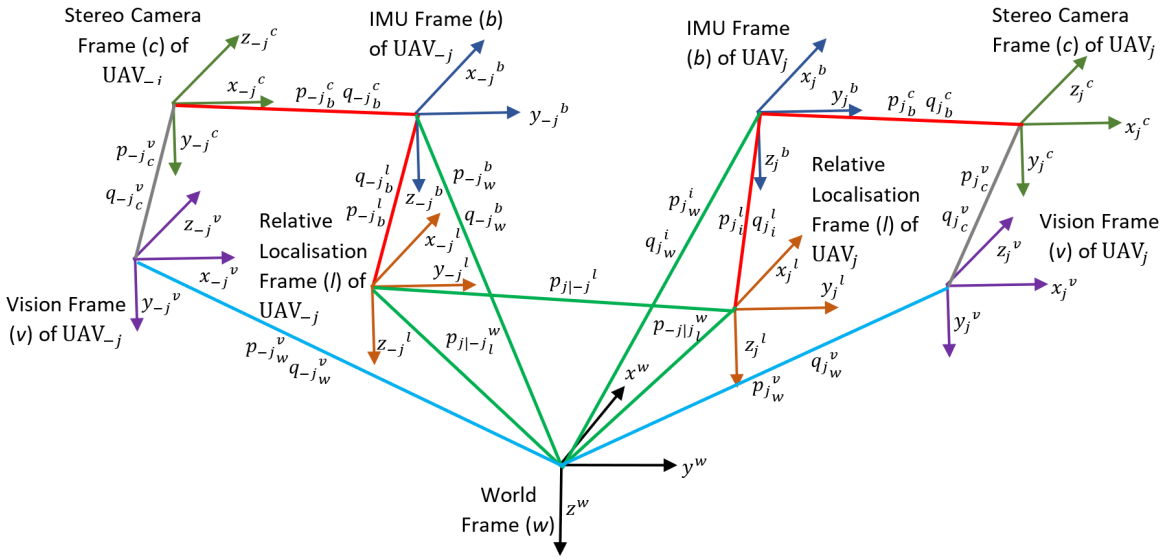


Fig. 4.2 Visualisation of the different coordinate frames

Fig. 4.2 shows the five coordinate frames necessary for the ESEKF formulation. w, represents the origin of the swarm system (which is the world frame), b, is the IMU frame (the UAV's body frame), c, is the camera frame (which is the centre of projection of the stereo camera), v, is the vision frame (which is the position and rotation of the camera when the first image is captured), l, is the relative localisation frame (which is the origin of the relative localisation system). The transformations between these frames requires translation (p) and the rotation (q). Superscripts denote the reference frame of the transformation and subscripts denote where the transformation is being applied. For example, ${p_{jc}}^{b}$, denotes the position of the origin of the camera frame with respect to the IMU frame of the $j\text{-}th$ UAV. ${q_{jc}}^{b}$ is the quaternion representation of the orientation of the camera frame with respect to the IMU frame from the $j\text{-}th$ UAV. The same holds for the transformations between the other frames. ${p_{j|-j}}^{w}_{l}$ describes the relative position of UAV$_{-j}$ measured by the relative localisation system on UAV$_j$, which is expressed in the world frame.

It is assumed that the angular velocity measurements (ω_m) and the linear acceleration measurements (a_m) of the IMU contain biases b_ω and b_a respectively. It is also assumed the inertial measurements have white Gaussian noise (n_ω, n_a). Therefore, the true angular velocities (ω) and the true accelerations (a) in the body frame can be defined as [138]:

$$\omega = \omega_m - b_\omega - n_\omega \qquad a = a_m - b_a - n_a, \tag{4.1}$$

With these assumptions, a kinematic model of the motion (6-DoF) of a UAV can be derived. This kinematic model can be expressed as a set of differential equations based on the b and w frames as:

$$\begin{aligned} \dot{p}_b^w &= v_b^w \\ \dot{v}_b^w &= C_{q_b^w}(a_m - b_a - n_a) - g \\ \dot{q}_b^w &= \tfrac{1}{2}\Omega(\omega_m - b_\omega - n_\omega)q_b^w \qquad , \\ \dot{b}_\omega &= 0, \dot{b}_a = 0, \dot{p}_c^b = 0 \\ \dot{q}_c^b &= 0, \dot{p}_w^v = 0, \dot{q}_w^v = 0, \dot{p}_l^w = 0 \end{aligned} \tag{4.2}$$

where C_q is the rotational matrix corresponding to the quaternion q, g is the gravity vector in the world frame, and $\Omega(\omega)$ is the quaternion-multiplication matrix of ω [138], which is defined as

$$\Omega(\omega) = \begin{bmatrix} -\omega_{x^b} & -\omega_{y^b} & -\omega_{z^b} & 0 \\ 0 & \omega_{z^b} & -\omega_{y^b} & \omega_{x^b} \\ -\omega_{z^b} & 0 & \omega_{x^b} & \omega_{y^b} \\ \omega_{y^b} & -\omega_{x^b} & 0 & \omega_{z^b} \end{bmatrix}, \tag{4.3}$$

where ω_{x^b}, ω_{y^b}, ω_{z^b} are the angular rotations of in each axis in the body frame.

The ESEKF algorithm consists of two steps: a prediction step and a correction step. The prediction step uses the IMU's linear accelerations and angular velocities to estimate the position of the UAV according to its kinematic model (Equation (4.2)). In this step, the pose (position and rotation) estimate of the stereo camera's VSLAM algorithm and the relative localisation position estimates are used to compensate for the IMU's bias and noise along the three axes. This process is summarised in Algorithm 2

4.3.1 Prediction Step

Using the five coordinate frames of reference shown in Fig. 4.2 and Equation (4.2), the estimated state vector $\hat{x} \in \mathbb{R}^{30}$, and the error state vector $\tilde{x} \in \mathbb{R}^{30}$ can be now defined. To simplify the notation of the states in the formulation of the ESEKF, the index j is omitted in the description below as it is for UAV$_j$ and the index will be only shown when necessary. These vectors are defined as:

$$\hat{x} = \begin{bmatrix} \hat{p}_b^w \\ \hat{v}_b^w \\ \hat{q}_b^w \\ \hat{b}_\omega \\ \hat{b}_a \\ \hat{q}_w^v \\ \hat{p}_w^v \\ \hat{q}_c^b \\ \hat{p}_c^b \\ \hat{p}_l^w \end{bmatrix} \quad ; \quad \tilde{x} = \begin{bmatrix} \Delta p_b^w \\ \Delta v_b^w \\ \delta\theta_b^w \\ \Delta b_\omega \\ \Delta b_a \\ \delta\theta_w^v \\ \Delta p_w^v \\ \delta\theta_c^b \\ \Delta p_c^b \\ \Delta p_l^w \end{bmatrix}, \tag{4.4}$$

where $\Delta p_b^w = p_b^w - \hat{p}_b^w$, $\delta\theta_b^w$ is the error in the rotation of the IMU with respect to world frame, which can be computed by

$$\delta\theta_b^w = \hat{q}_b^{w-1} \otimes q_b^w \tag{4.5}$$

where the operator $\otimes$ represents quaternion multiplication.

As is the case for the real states, the predicted states and the error states are governed by a set of differential equations as:

$$\begin{aligned}
\dot{\hat{p}}_b^w &= \hat{v}_b^w \\
\dot{\hat{v}}_b^w &= C_{\hat{q}_b^w}(a_m - \hat{b}_a) - g \\
\dot{\hat{q}}_b^w &= \tfrac{1}{2}\Omega(\omega_m - \hat{b}_w)\hat{q}_b^w \\
\dot{\hat{b}}_\omega &= 0, \dot{\hat{b}}_a = 0, \dot{\hat{p}}_c^b = 0 \\
\dot{\hat{q}}_c^b &= 0, \dot{\hat{p}}_w^v = 0, \dot{\hat{q}}_w^v = 0, \dot{\hat{p}}_l^w = 0
\end{aligned} \tag{4.6}$$

$$\begin{aligned}
\Delta\dot{p}_b^w &= \Delta v_b^w \\
\Delta\dot{v}_b^w &= -C_{\hat{q}_b^w}\lfloor a_m - \hat{b}_a \rfloor_\times \delta\theta_b^w - C_{\hat{q}_b^w}\Delta b_a - C_{\hat{q}_b^w} n_a \\
\delta\dot{\theta}_b^w &= -\lfloor \omega_m - \hat{b}_\omega \rfloor_\times \delta\theta_b^w - \Delta b_\omega - n_\omega \\
\Delta\dot{b}_\omega &= n_{b_w}, \Delta\dot{b}_a = n_{b_a}, \Delta\dot{p}_c^b = 0 \\
\delta\dot{q}_c^b &= 0, \Delta\dot{p}_w^v = 0, \delta\dot{q}_w^v = 0, \Delta\dot{p}_l^w = 0
\end{aligned} \quad , \tag{4.7}$$

where $\lfloor x \rfloor_\times$ is the skew-symmetric matrix composed from the vector x.

The error state dynamics can be described in the linearised continuous time error state as:

$$\dot{\tilde{x}} = F_c\tilde{x} + G_c n, \tag{4.8}$$

with

$$n = \begin{bmatrix} n_a \\ n_{b_a} \\ n_\omega \\ n_{b_\omega} \end{bmatrix} \quad ; \quad G_c = \begin{bmatrix} 0_3 & 0_3 & 0_3 & 0_3 & 0_{3\times 15} \\ -\hat{C}_{(q_b^w)} & 0_3 & 0_3 & 0_3 & 0_{3\times 15} \\ 0_3 & 0_3 & -I_3 & 0_3 & 0_{3\times 15} \\ 0_3 & 0_3 & 0_3 & I_3 & 0_{3\times 15} \\ 0_3 & I_3 & 0_3 & 0_3 & 0_{3\times 15} \\ 0_{15\times 3} & 0_{15\times 3} & 0_{15\times 3} & 0_{15\times 3} & 0_{15} \end{bmatrix} \quad ; \tag{4.9}$$

$$F_c = \begin{bmatrix} 0_3 & I_3 & 0_3 & 0_3 & 0_3 & 0_{3\times 15} \\ 0_3 & 0_3 & -C_{\hat{q}_b^w} \lfloor a_m - \hat{b}_a \rfloor_\times & 0_3 & -C_{\hat{q}_b^w} & 0_{3\times 15} \\ 0_3 & 0_3 & -\lfloor \omega_m - \hat{b}_\omega \rfloor_\times & -I_3 & 0_3 & 0_{3\times 15} \\ 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_{3\times 15} \\ 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_{3\times 15} \\ 0_{15\times 3} & 0_{15\times 3} & 0_{15\times 3} & 0_{15\times 3} & 0_{15\times 3} & 0_{315} \end{bmatrix},$$

where the coefficients for Q_c were obtained from the manufacturer's specifications of the IMU.

For the implementation of the ESEKF in the discrete time domain, the system Equation (4.8) needs to be discretised. From the Zero-Order-Holder equivalence:

$$F_d = exp(F_c \Delta t) = I_d + F_c \Delta t + \frac{1}{2!} F_c^2 \Delta t_k^2 + \dots , \tag{4.10}$$

where $\Delta t = t_k - t_{k-1}$.

A repetitive and sparse structure can be found in the matrix expansion, which allows F_d to be

expressed in the discrete time domain [138]. Accordingly,

$$F_d(t_k) = \begin{bmatrix} I_3 & \Delta t & A & B & -C_{\hat{q}_b^w}\frac{(\Delta t)^2}{2} & 0_{3\times 15} \\ 0_3 & I_3 & C & D & -C_{\hat{q}_b^w}\Delta t & 0_{3\times 15} \\ 0_3 & 0_3 & E & F & 0_3 & 0_{3\times 15} \\ 0_3 & 0_3 & 0_3 & I_3 & 0_3 & 0_{3\times 15} \\ 0_3 & 0_3 & 0_3 & 0_3 & I_3 & 0_{3\times 15} \\ 0_{15\times 3} & 0_{15\times 3} & 0_{15\times 3} & 0_{15\times 3} & 0_{15\times 3} & I_{15} \end{bmatrix} \tag{4.11}$$

with

$$\begin{aligned} A &= -C_{(\hat{q}_b^w)}\lfloor a_m - \hat{b}_a \rfloor \times \left(\tfrac{(\Delta t)^2}{2} - \tfrac{(\Delta t)^3}{3!}\lfloor \omega_m - \hat{b}_w \rfloor \times + \tfrac{(\Delta t)^4}{4!}(\lfloor \omega_m - \hat{b}_w \rfloor \times)^2\right) \\ B &= -C_{(\hat{q}_b^w)}\lfloor a_m - \hat{b}_a \rfloor \times \left(\tfrac{-(\Delta t)^3}{3!} + \tfrac{(\Delta t)^4}{4!}\lfloor \omega_m - \hat{b}_w \rfloor \times - \tfrac{(\Delta t)^5}{5!}(\lfloor \omega_m - \hat{b}_w \rfloor \times)^2\right) \\ C &= -C_{(\hat{q}_b^w)}\lfloor a_m - \hat{b}_a \rfloor \times \left(\Delta t - \tfrac{(\Delta t)^2}{2}\lfloor \omega_m - \hat{b}_w \rfloor \times + \tfrac{(\Delta t)^3}{3!}(\lfloor \omega_m - \hat{b}_w \rfloor \times)^2\right) \\ D &= -A \\ E &= I_3 - \Delta t \lfloor \omega_m \rfloor \times + \tfrac{(\Delta t)^2}{2!}(\lfloor \omega_m \rfloor \times)^2 \\ F &= -\Delta t + \tfrac{(\Delta t)^2}{2!}\lfloor \omega_m \rfloor \times - \tfrac{(\Delta t)^3}{3!}(\lfloor \omega_m \rfloor \times)^2 \end{aligned} \tag{4.12}$$

The discrete time system noise covariance matrix Q_d can now be calculated [84]. Thus,

$$Q_d(t_k) = \int_{t_k}^{t_{k+1}} F_d(\tau) G_c Q_c G_c^T F_d(\tau)^T d\tau. \tag{4.13}$$

During the prediction step, the position and rotation of the UAV is estimated according to the inertial measurements of the IMU. For every new IMU measurement the following steps are performed:

1. Predict the state variables according to their difference equations.
2. Calculate $F_d(t_k)$ according to Equation (4.11)
3. Calculate $Q_d(t_k)$ according to Equation (4.13)

4. Compute the predicted error state covariance matrix as:

$$P_{t_{k+1}|t_k} = F_d(tk)P_{t_k|t_k}F_d^T(tk) + Q_d(tk) \tag{4.14}$$

In the discrete time domain the position and velocity are estimated according to:

$$\begin{aligned} \hat{p}^w_{i_{t_{k+1}|t_k}} &= \hat{p}^w_{b_{t_k|t_k}} + \dot{\hat{p}}^w_{b_{t_k|t_k}}\Delta t \\ \hat{v}^w_{i_{t_{k+1}|t_k}} &= \hat{v}^w_{b_{t_k|t_k}} + \dot{\hat{v}}^w_{b_{t_k|t_k}}\Delta t \end{aligned} \tag{4.15}$$

To estimate the quaternion, a first order quaternion integrator can be used [131]. This gives

$$\begin{aligned} \hat{q}^w_i(t_{k+1}) = &\left(exp\left(\frac{1}{2}\Omega(\dot{\omega}_m)\Delta t\right)\right. \\ &\left. + \frac{1}{48}\Big(\Omega(\omega_m(t_{k-1}))\Omega(\omega_m)(t_k) - \Omega(\omega_m(t_k))\Omega(\omega_m)(t_{k-1})\Big)\Delta t^2{}_k\right)\hat{q}^w_b(t_k), \end{aligned} \tag{4.16}$$

where $\Omega(\dot{\omega_m})$ is:

$$\Omega(\dot{\omega_m}) = \Omega\left(\frac{\Omega(\omega_m)(t_{k-1}) - \Omega(\omega_m)(t_k)}{\Delta t}\right). \tag{4.17}$$

The prediction step in the ESEKF is summarised in lines 5 through 11 of Algorithm 2.

4.3.2 Correction Step

If the prediction step is repeated, the covariance of the error state will grow, which means that estimation becomes increasingly inaccurate. Measurements from other sensors (stereo camera and relative localisation) can provide extra information to correct the position estimates. These measurements are:

The VSLAM algorithm estimates position as well as the rotation of the stereo camera.

Therefore, the position of the stereo camera p_c^v can be computed by

$$z_p = p_c^v = C_{q_w^v}(p_b^w + C_{q_i^w} p_c^b), \tag{4.18}$$

and the rotation of the camera expressed in the vision frame (q_c^v) by

$$z_q = q_c^v = q_w^v \otimes q_b^w \otimes q_c^w. \tag{4.19}$$

The relative localisation system position estimate of UAV$_j$ is modelled as:

$$z_l = P_{\_j|jl}{}^w = P_{jb}^w + C_{\hat{q}_{jb}^w} P_{jl}^b. \tag{4.20}$$

From these equations, the observation matrix H, and the measurement noise covariance matrix R, can be determined. The observation matrices for the position and rotation of the camera, $H_{\_st_p}$ and $H_{\_st_q}$, can be defined by Equations (4.18) and (4.19) respectively [102]. The observatility matrix for the position estimates of the relative localisation system $H_{\_l_p}$ is defined by Equation (4.20). Detailed definitions are as shown below.

$$H\_st_p = \begin{bmatrix} \hat{R}_w^v \\ 0_{3\times3} \\ -C_{\hat{q}_w^v} C_{\hat{q}_b^w} \lfloor \hat{p}_c^b \rfloor \times \\ 0_{3\times3} \\ 0_{3\times3} \\ -C_{\hat{q}_w^v} \lfloor (\hat{p}_b^w + C_{\hat{q}_b^w} \hat{p}_c^b) \rfloor \times \\ I_{3\times3} \\ 0_{3\times3} \\ C_{\hat{q}_w^v} C_{\hat{q}_b^w} \\ 0_{3\times3} \end{bmatrix}^T \quad ; \quad H\_st_q = \begin{bmatrix} 0_{3\times3} \\ 0_{3\times3} \\ \frac{1}{2} C_{\hat{q}_b^c} \\ 0_{3\times3} \\ 0_{3\times3} \\ \frac{1}{2} C_{\hat{q}_b^c} C_{\hat{q}_w^b} \\ 0_{3\times3} \\ \frac{1}{2} I_{3\times3} \\ 0_{3\times3} \\ 0_{3\times3} \end{bmatrix}^T \tag{4.21}$$

$$H\_l_p = \begin{bmatrix} I_{3\times3} \\ 0_{3\times3} \\ 0_{3\times3} \\ 0_{3\times3} \\ 0_{3\times3} \\ 0_{3\times3} \\ 0_{3\times3} \\ 0_{3\times3} \\ 0_{3\times3} \\ C_{\hat{q}_{j_b}^w} \end{bmatrix}^T \tag{4.22}$$

H_st_p, H_st_q, and H_l_p are computed separately and then combined into one observation

matrix H, where

$$H = \begin{bmatrix} H_{\_st_p} \\ H_{\_st_q} \\ H_{\_l_p} \end{bmatrix}.$$

The residual, $\tilde{z}$, is calculated as $z - \hat{z}$. Thus

$$\tilde{z} = \begin{bmatrix} z_p - C_{\hat{q}_w^v}(\hat{p}_b^w + C_{\hat{q}_b^w}\hat{p}_c^b) \\ (\hat{q}_c^v)^{-1} \otimes z_q \\ z_l - \hat{p}_b^w \end{bmatrix}. \tag{4.23}$$

The measurement noise covariance matrix, R, is based on the position and rotation measurements of the VSLAM algorithm and the position shared by the relative localisation system in UAV$_{_j}$. Accordingly

$$R = \begin{bmatrix} R_{p_3} & 0_3 & 0_3 \\ 0_3 & R_{q_3} & 0_3 \\ 0_3 & 0_3 & R_{l_3} \end{bmatrix}, \tag{4.24}$$

where

$$R_{p_3} = \begin{bmatrix} \sigma^2_{z_{px}} & 0 & 0 \\ 0 & \sigma^2_{z_{py}} & 0 \\ 0 & 0 & \sigma^2_{z_{pz}} \end{bmatrix} \quad ; \quad R_{q_3} = \begin{bmatrix} \sigma^2_{z_{qx}} & 0 & 0 \\ 0 & \sigma^2_{z_{qy}} & 0 \\ 0 & 0 & \sigma^2_{z_{qz}} \end{bmatrix} \quad ; \tag{4.25}$$

$$R_{l_3} = \begin{bmatrix} \sigma^2_{z_{l_x}} & 0 & 0 \\ 0 & \sigma^2_{z_{l_y}} & 0 \\ 0 & 0 & \sigma^2_{z_{l_z}} \end{bmatrix}.$$

The magnitude of the covariances determine the level of trust in the corresponding measurements, with a smaller covariance implying higher level of trust. It is difficult to compute covariances from experimetal results becasuse of errors in sensor measurements. Because of

this uncertainty covariance values have been tuned by trial and error to control the behaviour of the ESEKF.

The ESEKF procedure for the correction step is as follows:

1. Compute the residual: $\tilde{z} = z - \hat{z}$
2. Compute the innovation: $S = HPH^T + R$
3. Compute the Kalman gain: $K = PH^T S^{-1}$
4. Compute the correction: $\hat{\tilde{x}} = K\tilde{z}$
5. Update states: $\hat{x}_{t_{k+1}|t_{k+1}} = \hat{x}_{t_{k+1}|t_k} + \hat{\tilde{x}}$
6. Update the error state covariance matrix:
   $P_{t_{k+1}|t_{k+1}} = (I_d - KH)P_{t_{k+1}|t_k}(I_d - KH)^T + KRK^T$

The correction stage of the ESEKF is summarised in lines 12 through 29 of Algorithm 2. It is important to note that in Algorithm 2, $H_{\_st_p}$ and $H_{\_st_q}$ are the main measurements used to predict the position of the UAV, and $H_{\_l_p}$ is used to correct the prediction. If $H_{\_l_p}$ were to be used as the main sensor, the prediction of the UAV's position would become unstable because the refresh rate of the relative localisation is not frequent enough to sustain a stable flight path.

Algorithm 2 Error State Extended Kalman Filter

Input: IMU measurements (a_m, ω_m), VSLAM position and rotation measurements (z_p, z_q), relative localisation position estimates (z_l), measurement noise covariance matrix R, process noise covariance matrix Q_c
Output: $\hat{x}$

1: $k \leftarrow 0$
2: $t_k \leftarrow 0$
3: $\hat{x}_{t_k} \leftarrow 0$

Algorithm 2 Error State Extended Kalman Filter (continued)

loop
  Prediction step
  $\hat{p}^{w}_{b_{t_{k+1}|t_k}} \leftarrow \hat{p}^{w}_{b_{t_k|t_k}} + \dot{\hat{p}}^{w}_{b_{t_k|t_k}} \Delta t_k$
  $\hat{v}^{w}_{b_{t_{k+1}|t_k}} \leftarrow \hat{v}^{w}_{b_{t_k|t_k}} + \dot{\hat{v}}^{w}_{b_{t_k|t_k}} \Delta t_k$
  compute $\hat{q}^{w}_{b}(t_{k+1})$ according to the first order quaternion integrator (Equation (4.16))
  compute $F_d(t_k)$ according to Equation (4.11)
  compute $Q_d(t_k)$ according to Equation (4.13)
  $P_{t_{k+1}|t_k} \leftarrow F_d(t_k) P_{t_k|t_k} F_d^T(t_k) + Q_d(t_k)$
  Correction step
  if any measurement update is available **then**
    if a new measurement from the VSLAM algorithm is received **then**
      $\tilde{z} \leftarrow z - \hat{z}$
      $S \leftarrow HPH^T + R$ based on the observation matrix $H = \begin{bmatrix} H_{\_st_p} \\ H_{\_st_q} \\ 0 \end{bmatrix}$
      $K \leftarrow PH^T S^{-1}$
      $\hat{\tilde{x}} \leftarrow K\tilde{z}$
      $\hat{x}_{t_{k+1}|t_{k+1}} \leftarrow \hat{x}_{t_{k+1}|t_k} + \hat{\tilde{x}}$
      $P_{t_{k+1}|t_{k+1}} \leftarrow (I_d - KH) P_{t_{k+1}|t_k} (I_d - KH)^T + KRK^T$
    end if
    if a new measurement from the the relative localisation system in UAV$_{_j}$ is shared **then**
      $\tilde{z} \leftarrow z - \hat{z}$
      $S \leftarrow HPH^T + R$ based on the observation matrix $H = \begin{bmatrix} 0 \\ 0 \\ H_{\_l_p} \end{bmatrix}$
      $K \leftarrow PH^T S^{-1}$
      $\hat{\tilde{x}} \leftarrow K\tilde{z}$
      $\hat{x}_{t_{k+1}|t_{k+1}} \leftarrow \hat{x}_{t_{k+1}|t_k} + \hat{\tilde{x}}$
      $P_{t_{k+1}|t_{k+1}} \leftarrow (I_d - KH) P_{t_{k+1}|t_k} (I_d - KH)^T + KRK^T$
    end if
  else
    $P_{t_{k+1}|t_{k+1}} \leftarrow F_d(t_k) P_{t_k|t_k} F_d^T(t_k) + Q_d(t_k)$
    $k \leftarrow k + 1$
  end if
end loop

4.4 Design of Indoor Experiments

In order to test the ESEKF's ability to improve position estimates of VSLAM a set of experiments similar to those conducted in Chapter 3, but now including the ESEKF were run. For these experiments, UAV_2 estimated its position using the VSLAM algorithm. At the same time, UAV_1 estimated the position of UAV_2 using the relative localisation system developed in this chapter, including the ESEKF, to measure the position of UAV_2. The experimental setup is shown in Fig. 4.4. This experimental setup was chosen because the indoor environment has constant lighting, and is rich in static and distinctive visual features that improve the accuracy of the VSLAM algorithm.

For the experiments, three sensors were used: an IMU for the prediction step, a stereo camera and the relative localisation system (Chapter 3) for the correction step. To select the IMU, it was necessary to take into account the refresh rate of the measurements to be performed by the sensor. A choice was made to select the UM7-IMU [24], which filters the high frequency noise of the inertial measurements at 100 Hz. This high refresh rate enables the ESEKF to estimate the fast changes in motion of a UAV during the prediction step. For VSLAM, the ZED stereo camera and the NVIDIA TX1 GPU were used. Stereo images from the ZED camera were processed by the ZED Software Development Kit (SDK), which included a VSLAM algorithm that ran on the TX1 GPU. The VSLAM algorithm estimated the position and rotation of the camera at a rate of 20 Hz. To correct the estimated position from the VSLAM algorithm, the position estimates from the relative localisation system described in Chapter 3 was also used. The relative localisation system estimated the position of a neighbouring at 4 Hz. Therefore, the proposed ESEKF is a multi-rate state estimator with three different sample rates. The different sampling times of measurements are illustrated in Fig. 4.3. In the case of real-time applications, the inherent time offset

between when a position estimate is made by the sensors and the transfer of the data to a processing unit can delay the correction step of the ESEKF. This time delay can be reduced by back-propagating the correcting measurement. This can be achieved by time stamping all the sensor measurements and re-computing correction step of the ESEKF based on the correct time given by the time-stamps and propagate the correction until it reaches the most recent prediction.

With this experimental set-up, 15 repetitions (trials) of the experiment were performed. The

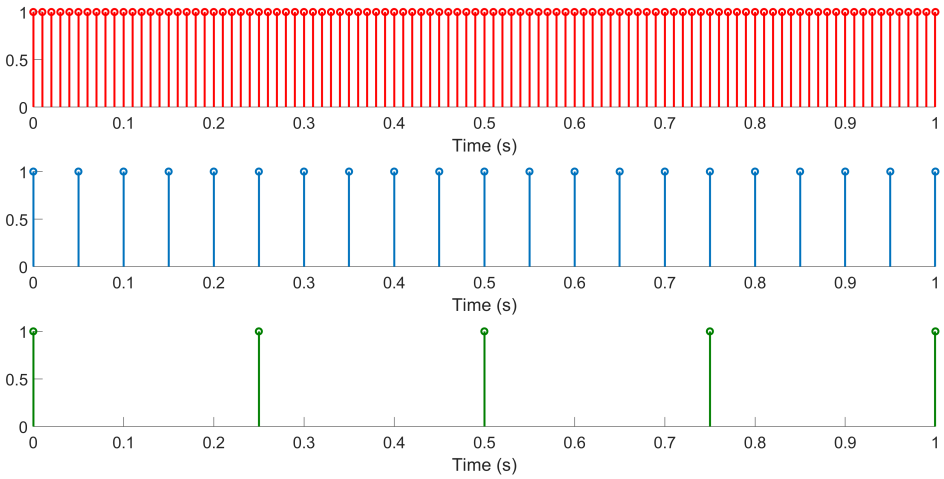


Fig. 4.3 Measurement refresh rates. Red is the IMU = 100 Hz, blue is the ZED stereo camera = 20 Hz and green is the relative localisation system = 4 Hz.

internal clocks of UAV_1, UAV_2, and the Optitack motion capture system were synchronised using an NTP server running in UAV_1. To process the data compute the position estimates, the proposed ESEKF was coded in Matlab 2017b and ran on a Core i7 computer with 8 Gb of RAM.

4.5 Experimental Results and Discussion

The experiment allowed the measurement noise covariance matrix, R, in Equation (4.24), to be tuned in order to improve the performance of the ESEKF trajectory estimates. During the

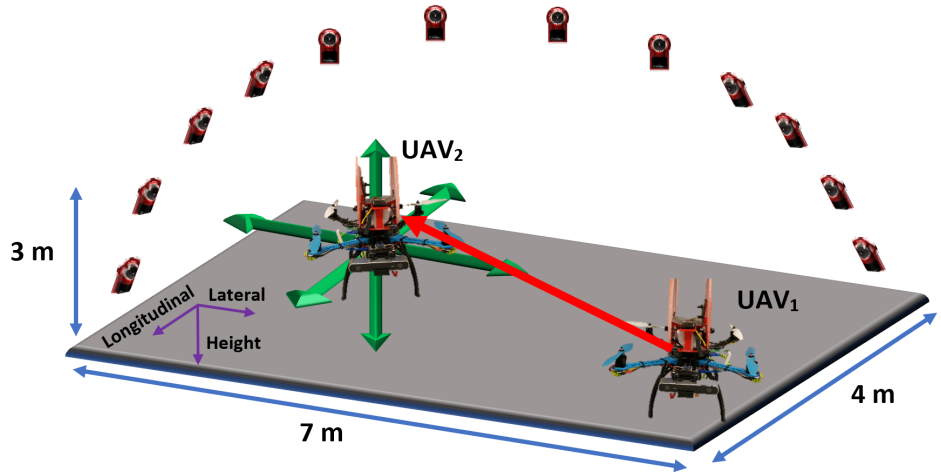


Fig. 4.4 This image shows how the indoor experiments were performed. UAV_2 moved in any direction (green arrows), performing VSLAM. UAV_1 was stationary and its bearing aligned with the world frame. UAV_1 estimated the position of UAV_2 while it was in motion (red arrow). UAV_2 moved within a volume of 84 m^3 while the Optitrack motion capture system measured the true positions of the two UAVs.

tuning of R, it was found that trusting the position estimates and the attitude estimates of the VSLAM algorithm more than the position estimates of the relative localisation system gave smoother trajectory estimates. By contrast, trusting the relative localisation system's position estimates more than those of the VSLAM algorithm produced more oscillatory trajectory estimates. This is because the covariances dictate the amount of correction performed by each sensor. Therefore, if the correction arrives at a slow rate compared to the prediction, oscillatory trajectory estimates can result.

In this case, the testing environment has static and distinctive visual features. Thus, it can be assumed that the VSLAM's estimates can be trusted more than the relative localisation system's position estimates. From the position and orientation of UAV_1, it can be derived the noise covariance of the relative localisation system's position estimates remain constant. Also, as the Longitudinal and Lateral axes were approximately aligned with the North and the East axes, the diagonal form of the covariance matrix can be used for the relative localisation system's position estimates. Moreover, from the findings in Chapter 3 the relative localisation

system's lateral position estimates can be trusted more than the position estimates in the Longitudinal and Height axes. Based on this reasoning the noise covariance matrix for UAV_2 was estimated as:

$$R_2 = \begin{bmatrix} 1.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & .5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & .5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & .5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5.4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6.6 \end{bmatrix} \tag{4.26}$$

It is worth mentioning, that the calculations needed to estimate the position of the UAV remain the same even with the added measurement of the relative localisation. However, the speed at which the ESEKF has to compute a correction when a relative localisation measurement is available does increase momentarily. This means that when a relative localisation measurement becomes available between two VSLAM measurements the ESEKF needs to compute the correction.

Table 4.1 presents the results from the 15 trials showing the average RMSE for the two set of position estimates: ESEKF's trajectory estimates, and the VSLAM algorithm's trajectory estimates compared with the ground truth measurements. It can be seen that there is an

overall reduction of the RMSE when the ESEKF is used. This reduction is approximately 24%, 75% and 11% in the Longitudinal, Lateral and Height axes respectively.

Fig. 4.5 shows the distribution of the RMSE of the ESEKF's trajectory estimates of UAV_2 from the 15 trials. It can be seen in the figure that the RMSE is approximately normally distributed in all axes. These distributions yield standard deviations of 0.005 m, 0.002 m, and 0.006 m in the Longitudinal, Lateral, and Height axes respectively. The results show that variations of the error in the 15 trials is relatively small. When estimates of the relative localisation system are shared from a stationary UAV (UAV_1).

The calculated RMSE shows that the trajectory estimates of the relative localisation system can reduce the drift of the VSLAM algorithm. Fig. 4.6 and Fig. 4.8 show one representative trial of the ESEKF experiment. In these figures, it can be seen that the ESEKF's trajectory estimates of UAV_2 tend to converge to the true trajectory measurements of the motion capture system, whereas the trajectory estimates of the VSLAM algorithm drift over time. This tendency is more evident in the Lateral axis where the trajectory estimates from the VSLAM algorithm have the biggest error.

Fig. 4.7 shows the error of the ESEKF's trajectory estimates of UAV_2 and the error of the VSLAM algorithm's trajectory estimates of the same trial as in Fig. 4.6 and Fig. 4.8. These are consistent with the RMSE summarised in Table 4.1 where the ESEKF's trajectory estimates of UAV_2 in the Lateral axis are more accurate than in other axes. These results are similar to the findings in Chapter 3, which shows that the trajectory estimates with the highest degree of accuracy from the relative localisation system are on the Lateral axis.

Axis	Average RMSE of the VSLAM algorithm's trajectory estimates (m)	Average RMSE of the ESEKF's trajectory estimates (m)
Longitudinal	0.0986	0.0752
Lateral	0.2613	0.1316
Height	0.0426	0.0380

Table 4.1 RMSE of the trajectory estimates of the VSLAM algorithm and RMSE of the trajectory estimates of the ESEKF

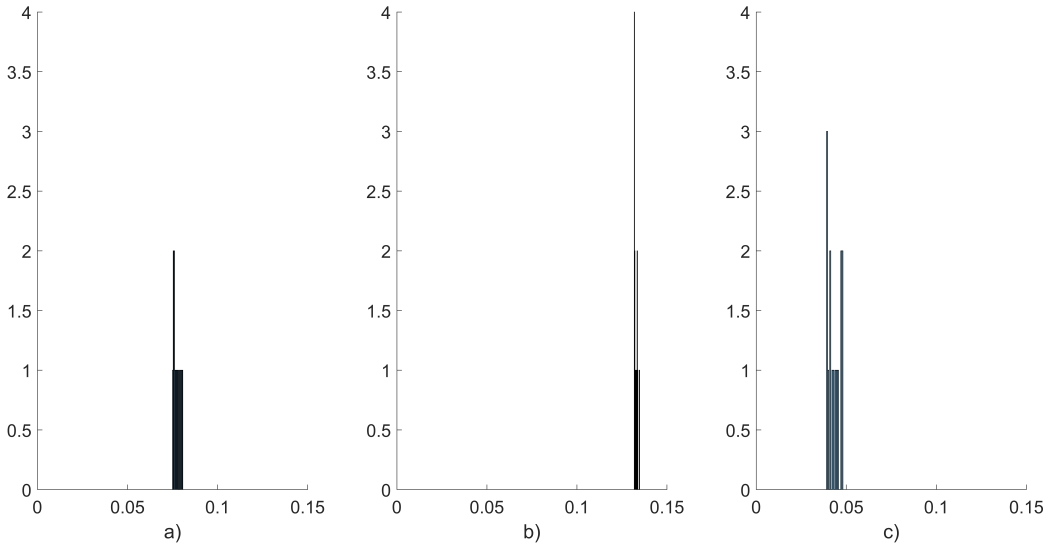


Fig. 4.5 Distribution of the ESEKF RMSE a) Longitudinal axis; b) Lateral axis; c) Height axis.

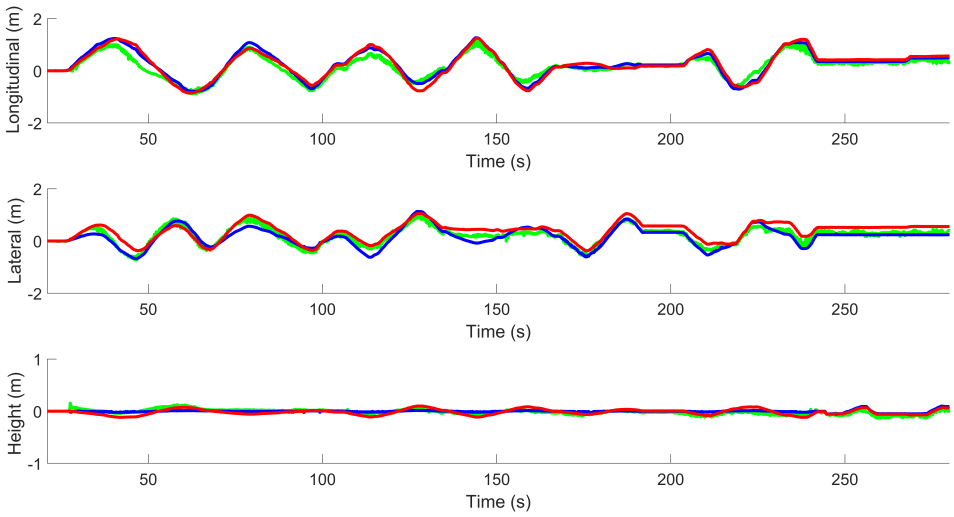


Fig. 4.6 ESEKF's estimated trajectories in the indoor environment. Blue: Optitrack trajectory measurements, Red: VSLAM algorithm's trajectory estimates, Green: ESEKF's trajectory estimates.

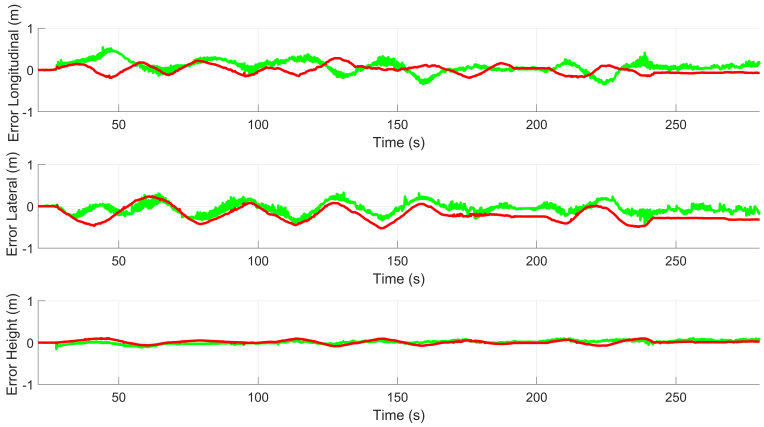


Fig. 4.7 Error of the ESEKF's estimated trajectories vs. error the VSLAM algorithm's trajectory estimates. Red: Error of the VSLAM algorithm's trajectory estimates, Green: Error of the ESEKF's trajectory estimates.

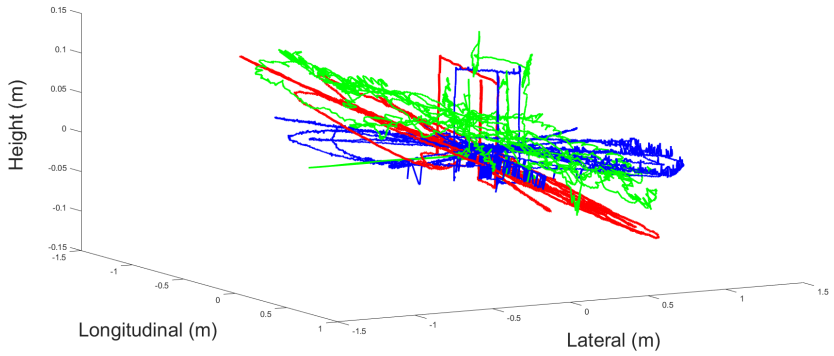


Fig. 4.8 ESEKF's 3D estimated trajectory. Blue: Optitrack 3D trajectory measurement, Red: VSLAM algorithm's 3D trajectory estimate, Green: ESEKF's 3D trajectory estimate.

4.6 Summary

This chapter has presented the design and testing of the improvement in position estimates when an ESEKF and relative localisation system is used for cooperative SLAM. The filter aims to reduce the drift of the VSLAM algorithm's position estimates in each UAV in the swarm. To achieve this, the ESEKF can be implemented individually in every member of the swarm. The ESEKF fuses the inertial measurements from the UAV's IMU with the position and rotation estimates from its VSLAM algorithm, and the shared position estimates from neighbouring UAVs relative localisation system. An experiment comprising of 15 trials was performed to determine the improvement of the position estimates when the ESEKF was used. The experiment consisted of UAV_2 estimating its position in the environment with the VSLAM algorithm while UAV_1 remained stationary estimating the position of UAV_2 through relative localisation.

The results from the experiments show that the ESEKF reduces the drift of the estimated trajectory of the VSLAM algorithm when compared with the true trajectory measurements of the motion capture system. This reduction was approximately 24%, 75% and 11% in the

Longitudinal, Lateral and Height axes respectively. It can also be noted that in the Lateral axis the ESEKF's trajectory estimates are more accurate than the trajectory estimates in the Longitudinal and Height axes. This experiments show that the proposed ESEKF is a viable option to be implemented in a swarm of UAVs for GPS-denied outdoor navigation. Therefore, the following chapter describes the implementation of this filter in multiple UAVs as well as testing in outdoor environments.

5. Cooperative Localisation in GPS-denied Environments

This chapter describes the implementation of the ESEKF, designed in Chapter 4, and testing with a swarm of UAVs in outdoor experiments. The chapter begins by describing the implementation the ESEKF in a cooperative swarm. Section 5.2 describes the design of the outdoor experiments. The results of the outdoor experiments are discussed in Section 5.4. Section 5.5 summarises and concludes the chapter.

5.1 A Proposed Cooperative Localisation Approach

In order for a swarm of UAVs to perform cooperative SLAM without map sharing, a cooperative ESEKF algorithm is implemented in each member of the swarm, UAV_j ($j = 1,2,3,...,N$). In each UAV_j the ESEKF fuses the sensor measurements (IMU, VSLAM) of UAV_j with the shared position estimates made by neighbouring UAVs using each of their relative localisation systems, at a time t_k ($k = 1,2,3,...,N$). The cooperative ESEKF algorithm required to implement the fusion can be summarised in the following steps. the complete implementation is given in Algorithm 3.

- The first step of the cooperative ESEKF algorithm, is to predict the position, $\hat{p}_{j_b}^{w}{}_{t_{k+1}|t_k}$, of each UAV_j in the swarm using its own ESEKF. This is calculated by using the inertial measurements of each UAV's IMU_j ($a_{jm}(t_k)$ and $\omega_{jm}(t_k)$) to propagate the

estimated states and error states of each UAV$_j$. The propagation of UAV$_j$'s states is performed by Equation 4.15 and 4.16. This process is described in lines 3 through 9 of Algorithm 3.

- The second step corrects the position of UAV$_j$ predicted by the ESEKF ($\hat{p}_{jb}^{w}{}_{t_{k+1}|t_k}$) when new position and attitude estimates from UAV$_j$'s VSLAM algorithm ($z_{jp}(t_k)$, $z_{jq}(t_k)$) are available. The correction of $\hat{p}_{jb}^{w}{}_{t_{k+1}|t_k}$ yields a new position estimate of UAV$_j$ ($\hat{p}_{jb}^{w}{}_{t_{k+1}|t_{k+1}}$). This process is described in lines 11 through 18 of Algorithm 3.

- The third step is to perform a second correction of UAV$_j$'s position by the ESEKF when UAV$_{-j}$ shares its position estimate of UAV$_j$. To achieve this, UAV$_{-j}$ uses its relative localisation system to estimate the position of UAV$_j$ ($z_{jl}(t_k)$). This position estimate is relative to UAV$_{-j}$'s position $\hat{p}_{-jb}^{w}$ at time t_k. The correction of $\hat{p}_{jb}^{w}{}_{t_{k+1}|t_k}$ with the shared position estimate from UAV$_{-j}$ yields a new position estimate of UAV$_j$ ($\hat{p}_{jb}^{w}{}_{t_{k+1}|t_{k+1}}$). This process is described in lines 19 through 26 of Algorithm 3.

This cooperative ESEKF shares sensory measurements available (a_{jm}, ω_{jm}, z_{jp}, z_{jq} and z_{jl}) to the swarm with all its members, as shown in Fig. 5.1. In the general case a UAV could be connected to multiple neighbours to share and receive relative localisation. However, the number of neighbours connected to one UAV is dictated by the formation and the hardware setup. Sharing the relative localisation with multiple UAVs allows the swarm system to cooperatively reduce the positional drift in each UAV$_j$'s VSLAM algorithm position estimate without map sharing or loop closure. Furthermore, this method also allows the swarm to add or reduce its members without modifying the formulation of each ESEKF$_j$, since the ESEKF is performed by each UAV$_j$ for itself.

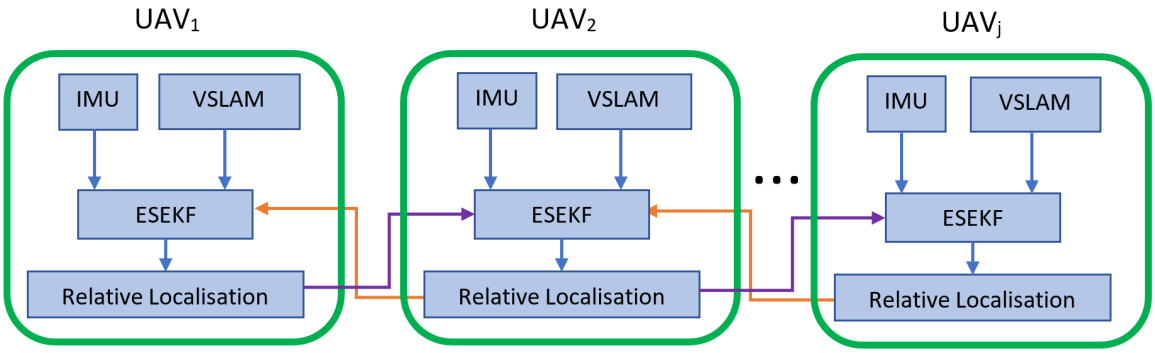


Fig. 5.1 Cooperative ESEKF system architecture.

Algorithm 3 Cooperative Error State Extended Kalman Filter

Input: IMU measurements ($a_{jm}(t_k)$ and $\omega_{jm}(t_k)$), VSLAM algorithm position and attitude estimates ($z_{jp}(t_k), z_{jq}(t_k)$), relative localisation system position estimates ($z_{jl}(t_k)$)

Output: $\hat{p}_{ji}^{w}{}_{t_{k+1}|t_{k+1}}$

loop
  $k \leftarrow 0$
  $t_k \leftarrow 0$
  $\hat{p}_{jb}^{w}{}_{t_{k+1}|t_k} \leftarrow 0$
  $\hat{p}_{jb_{t_{k+1}|t_k}}^{w} \leftarrow \hat{p}_{jb_{t_k|t_k}}^{w} + \dot{\hat{p}}_{jb_{t_k|t_k}}^{w} \Delta t$
  $\hat{v}_{jb_{t_{k+1}|t_k}}^{w} \leftarrow \hat{v}_{jb_{t_k|t_k}}^{w} + \dot{\hat{v}}_{jb_{t_k|t_k}}^{w} \Delta t$
  compute $\hat{q}_{jb}^{w}(t_{k+1})$ using Equation (4.16)
  compute$F_{jd}(t_k)$ using Equation (4.11)
  compute $Q_{jd}(t_k)$ using Equation (4.13)
  $P_{jt_{k+1}|t_k} \leftarrow F_{jd}(t_k) P_{t_k|t_k} F_{jd}^{T}(t_k) + Q_{jd}(t_k)$
  if a new position and attitude estimate from UAV$_j$'s VSLAM algorithm is received or a new measurement from the the relative localisation system in UAV$_{-j}$ is shared **then**
    if a new position and attitude estimate from UAV$_j$'s VSLAM algorithm is received ($z_{jp}(t_k), z_{jq}(t_k)$) **then**
      $\tilde{z}_j \leftarrow z_j - \hat{z}_j$

Algorithm 3 Distributed ESEKF (continued)

14: $S_j \leftarrow H_j P_j {H^T}_j + R_j$ using the observation matrix $H_j = \begin{bmatrix} H_{j\_st_p} \\ H_{j\_st_q} \\ 0 \end{bmatrix}$

15: $K_j \leftarrow P_j {H^T}_j {S^{-1}}_j$

16: $\hat{\tilde{x}}_j \leftarrow K_j \tilde{z}_j$

17: $\hat{x}_{jt_{k+1}|t_{k+1}} \leftarrow \hat{x}_{jt_{k+1}|t_k} + \hat{\tilde{x}}_j$

18: $P_{jt_{k+1}|t_{k+1}} \leftarrow (I_d - K_j H_j) P_{jt_{k+1}|t_k} (I_d - K_j H_j)^T + K_j R_j {K_j}^T$

19: **end if**

20: **if** a new measurement from the the relative localisation system in UAV$_{-j}$ is shared ($z_{jl}(t_k)$) **then**

21: $\tilde{z}_j \leftarrow z_j - \hat{z}_j$

22: $S_j \leftarrow H_j P_j {H^T}_j + R_j$ using the observation matrix $H_j = \begin{bmatrix} 0 \\ 0 \\ H_{j\_l_p} \end{bmatrix}$

23: $K_j \leftarrow P_j {H^T}_j {S^{-1}}_j$

24: $\hat{\tilde{x}}_j \leftarrow K_j \tilde{z}_j$

25: $\hat{x}_{jt_{k+1}|t_{k+1}} \leftarrow \hat{x}_{jt_{k+1}|t_k} + \hat{\tilde{x}}_j$

26: $P_{jt_{k+1}|t_{k+1}} \leftarrow (I_d - K_j H_j) P_{jt_{k+1}|t_k} (I_d - K_j H_j)^T + K_j R_j {K_j}^T$

27: **end if**

28: **else**

29: $P_{jt_{k+1}|t_{k+1}} \leftarrow F_{jd}(t_k) P_{jt_k|t_k} F^T_{jd}(t_k) + Q_{jd}(t_k)$

30: **end if**

31: $k \leftarrow k+1$

32: **end loop**

5.2 Design of Outdoor Experiments

In order to evaluate the accuracy of the proposed cooperative ESEKF, flying experiments with three UAVs (UAV_1, UAV_2, and UAV_3) in two types of outdoor environments (GPS-denied and open field) were run. Each UAV was equipped with the same sensors, as described in Chapter 4. A Real Time Kinematic Differential GPS (RTK-DGPS) receiver was installed on each UAV for ground truth measurements. Importantly the RTK-GPS receiver was only used for error calculations and did not form part of the cooperative ESEKF formulation. The RTK-DGPS had a maximum ground truth accuracy of 2 cm [127]. This is superior to

the standalone GPS accuracy of 5 m [123] and was obtained using a base station, which transmits correction signals to each UAV's RTK-DGPS receiver.

Before the start of all experiments, the GPS base station was placed in a location near where the experiments occurred, with a clear line of sight to the sky. The base station was then left for an hour to average the GPS solutions in order to find the mean of its position. This allowed the transmission of a correction signal to the mobile GPS receivers. The internal clocks of all three UAVs were synchronised by using a network time protocol (NTP) server running on one UAV.

The data collected from these experiments includes: the IMU's linear acceleration and angular velocities, the position estimates of the VSLAM algorithm running on the TX1-GPU, the relative localisation system's position estimates, and the RTK-DGPS's ground truth position measurements. These were captured from all UAVs. The data was used as input for the the proposed cooperative ESEKF, which was coded in Matlab 2017b, running on a Core i7 computer with 8 Gb of RAM. For all experiments, the world frame, w, employed was as a fixed position within the environment and oriented along the navigation path as shown in Fig. 5.2. Therefore, all position estimates were rotated and translated to conform to the world frame. The characteristics of each experiment are now described in turn.

Experiment 1

Experiment 1 was performed under tree canopies, which is a GPS-denied environment, like the model scenario. This type of environment is rich in visual features and therefore, it allowed the VSLAM algorithm to perform better in estimating the position of the UAV compared with an environment that lacks visual features. Because GPS could not reliably

be used as ground truth to determine the accuracy of the position estimates of the ESEKF, the GPS positions at the start and the end of the flights were measured to visually assess the trajectory of the UAVs in a satellite image.

Two flying trials were performed in the GPS-denied environment, in which the three UAVs were flown manually 80m in a forward direction, in a straight line under tree canopies. This is shown in Fig. 5.2a. See also a video of this (https://youtu.be/QIqRG7jf3lQ). The UAVs maintained an average separation of eight metres and an average height of two meters to avoid collisions with trees or other UAVs. In this environment, the onboard relative localisation systems experienced intermittent outages due to the tree line at the centre (see Fig. 5.3).

To determine the measurement noise covariance matrix (R) of the ESEKF for all UAVs in Experiment 1, it was decided at first to use similar values as in Chapter 4 (Equation (4.24)), which yielded smooth trajectory estimates. However, the forward motion performed by the UAVs in the outdoor experiments had to be taken into account when determining the noise covariances of the VSLAM's position estimates. The forward motion enabled the VSLAM algorithm to produce more accurate position estimates in the forward direction because the features in this direction were a lot richer than those in the lateral and height directions.

The results in Chapter 3 showed that the relative localisation system's position estimates in the lateral direction had a higher degree of accuracy than in the longitudinal and height directions. Moreover, in Experiment 1, the UAVs were flown in a forward direction along a straight line with a fixed heading. This meant that the noise covariance of the relative localisation system's position estimates were constant. Also, as the Longitudinal and Lateral axes were approximately aligned with the North and the East axes, the diagonal form of the covariance matrix was used for the relative localisation system's position estimates. To

determine the most adequate R for the environment in Experiment 1, it was decided to test three different combinations of covariance coefficients, and implement them in the two trials.

In these three test cases, the noise covariance of the position and rotation estimations of VSLAM algorithm remained constant. The noise covariance coefficients of the relative localisation system's position estimates were modified as follows: Case 1. The noise covariance coefficients of the relative localisation system's position estimates were set to a higher level than those of the VSLAM's position estimates. This means that the position estimates of the relative localisation system were trusted less than the position estimates from the VSLAM algorithm.

$$\text{Case 1} \quad \begin{array}{lll} \sigma^2_{z_{px}} = 3.53 & ; \quad \sigma^2_{z_{py}} = 5.23 & ; \quad \sigma^2_{z_{pz}} = 6.45 \\ \sigma^2_{z_{qx}} = 0.32 & ; \quad \sigma^2_{z_{qy}} = 0.42 & ; \quad \sigma^2_{z_{qz}} = 0.52 \\ \sigma^2_{z_{l_x}} = 18.91 & ; \quad \sigma^2_{z_{l_y}} = 10.26 & ; \quad \sigma^2_{z_{l_z}} = 20.91 \end{array}$$

Case 2. The noise covariance coefficients of the relative localisation system's position estimates were set to a lower level than those of the VSLAM's position estimates.This meant that the relative localisation system was trusted more than VSLAM.

$$\text{Case 2} \quad \sigma^2_{z_{l_x}} = 1.15 \quad ; \quad \sigma^2_{z_{l_y}} = 0.91 \quad ; \quad \sigma^2_{z_{l_z}} = 3.23$$

Case 3. The noise covariance coefficient of the relative localisation system's position estimate in the Lateral direction was set to a lower level than the one of the VSLAM's position estimate in the same direction. The noise covariance coefficients of the relative localisation system's position estimates in the longitudinal and height directions were set to a higher level than those of the of the VSLAM's position estimate in the same direction. This meant that the relative localisation system was trusted more than VSLAM, but only in the lateral direction.

$$\text{Case3} \qquad \sigma^2_{z_{l_x}} = 4.91 \quad ; \quad \sigma^2_{z_{l_y}} = 2.45 \quad ; \quad \sigma^2_{z_{l_z}} = 8.85$$

Experiment 2

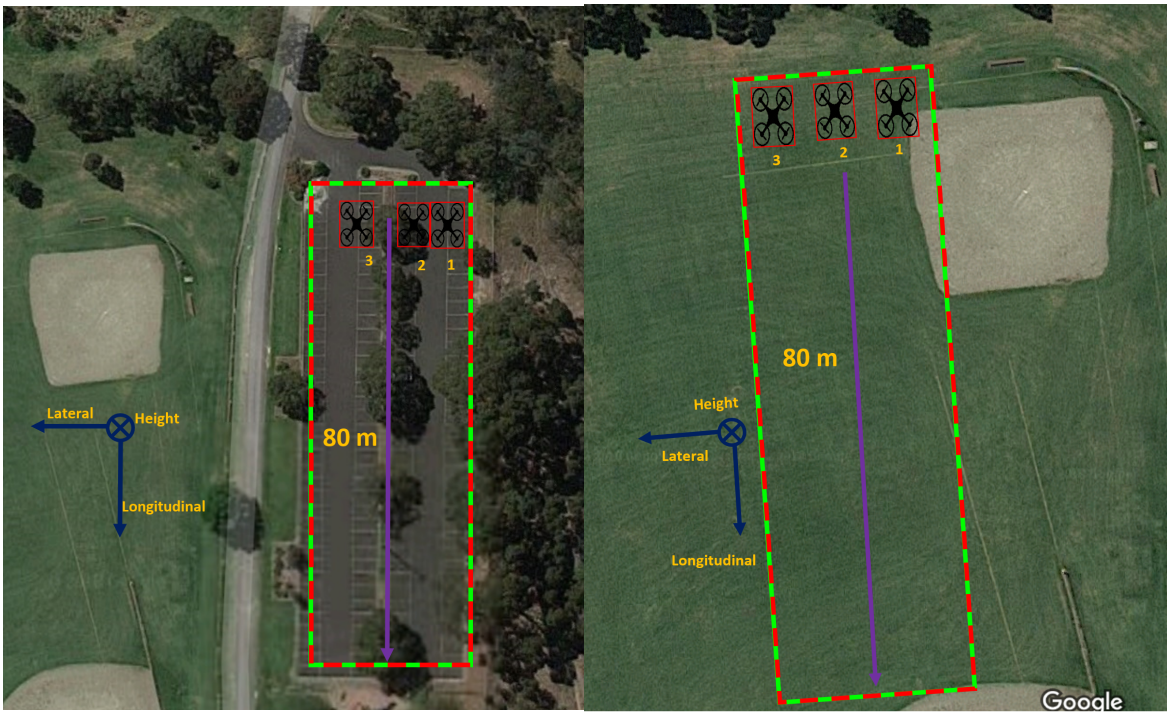
Experiment 2, was performed in an open field with a clear line of sight to the sky. This meant that the high-accuracy RTK-DGPS measurements could be recorded. This enabled the RMSE calculation over the entire trajectory of the three UAVs in Experiment 2. However, this environment lacked visual features, which degraded the performance of the VSLAM algorithm. This meant that in this environment the position and attitude estimates of the VSLAM algorithm were less accurate than those made in the GPS-denied environment, which had richer visual features.

Two trials were performed where three UAVs were flown manually 80 m in forward direction along a straight line at the height of about two meters.The average separation between UAVs of about 8 meters to avoid collisions between UAVs (see Fig. 5.2b). In this environment, the visual line of sight of the 180° FoV camera was not occluded at any point (see Fig. 5.4). These meant that the position estimates of the relative localisation systems were performed without any interruption.

The lack of visual features had to be taken in to account in order to estimate the noise covariance matrix for VSLAM position and rotation estimates. Therefore, noise covariance coefficients of the VSLAM position and rotation estimates were increased to a higher degree compared with Experiment 1. Because the UAVs followed a similar trajectory as in

Experiment 1 and the axes were also approximately aligned with the NED frame, similar coefficients to Case 3 of Experiment 1 were used for the relative localisation. The coefficients used in the two trials of Experiment 2 are shown below.

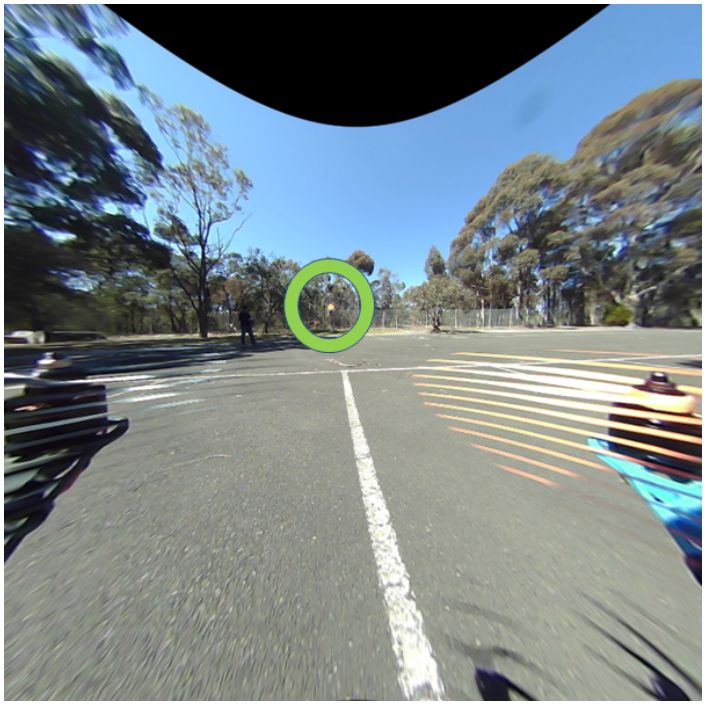
$$\sigma^2_{z_{p_x}} = 25.58 \quad ; \quad \sigma^2_{z_{p_y}} = 20.83 \quad ; \quad \sigma^2_{z_{p_z}} = 10.85$$
$$\sigma^2_{z_{q_x}} = 0.82 \quad ; \quad \sigma^2_{z_{q_y}} = 0.72 \quad ; \quad \sigma^2_{z_{q_z}} = 0.62$$
$$\sigma^2_{z_{l_x}} = 3.45 \quad ; \quad \sigma^2_{z_{l_y}} = 1.67 \quad ; \quad \sigma^2_{z_{l_z}} = 7.65$$



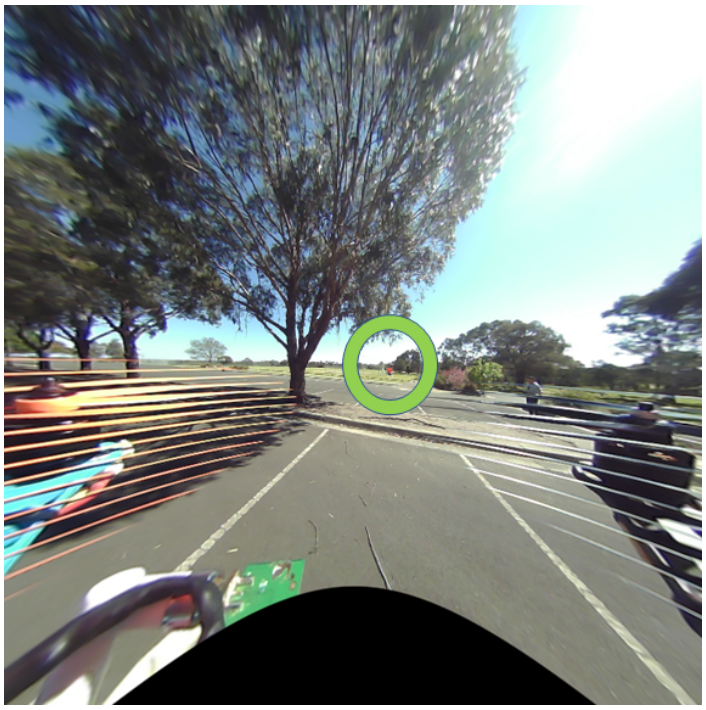
(a) GPS-denied environment

(b) Open field environment

Fig. 5.2 Satellite image of the outdoor areas where the experiments were performed a) Outdoor GPS-denied area (parking lot) and the starting positions of the UAVs with the orientation of the world frame. b) Outdoor open field area (oval) and the starting positions of the UAVs with the orientation of the world frame. In both experiments, the UAVs moved 80 m forward in a straight line while maintaining a linear formation.

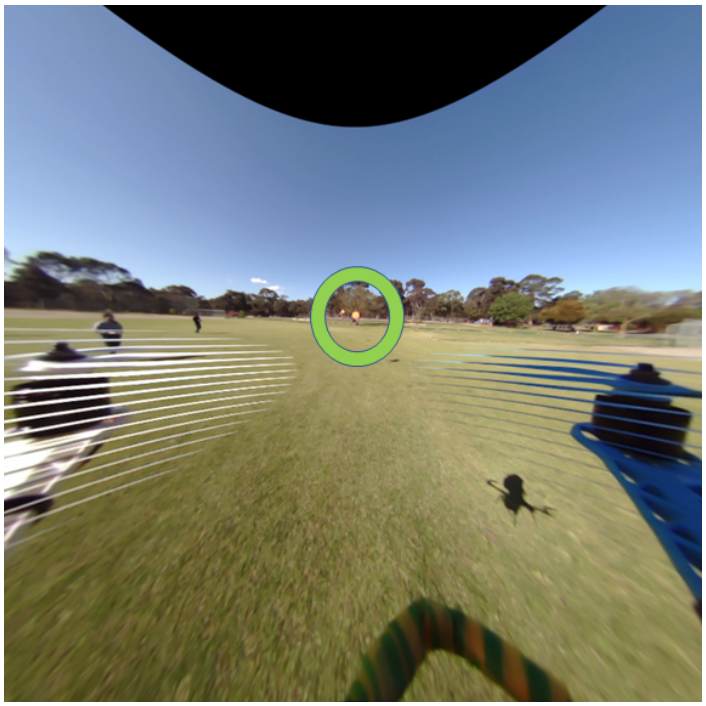


(a) UAV_2 left side camera

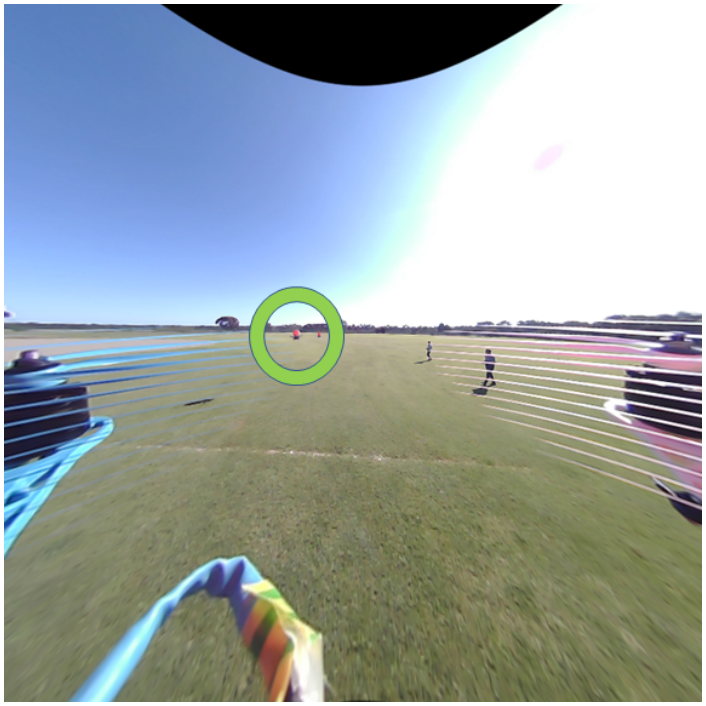


(b) UAV_2 right side camera

Fig. 5.3 Images from the side looking cameras of the neighbouring UAVs during the GPS-denied test. The green circle shows the tracked marker. a) UAV_2 left side looking camera. b) UAV_2 right side looking camera



(a) UAV_3 left side camera



(b) UAV_1 right side camera

Fig. 5.4 Images from the side looking cameras of the neighbouring UAVs during the open field test. The green circle shows the tracked marker. a) UAV_3 left side looking camera. b) UAV_1 right side looking camera

5.3 Results and Discussion

Experiment 1

For Case 1: Fig. 5.5 shows the estimated trajectories of each UAV using VSLAM alone (red) and cooperative ESEKF (green) for Trial 1. In this environment, the RTK-DGPS trajectory measurements of each UAV are intermittent. Thus, the RMSE of the whole trajectories of each UAV could not be calculated. It can be observed from Fig. 5.5 that only the position estimates inside the red boxes can be used to calculate the RMSE, as this when the GPS measurement was reliable. Thus, RMSE was only calculated over this parts of the UAV's trajectory. Table 5.1 and 5.2 show the calculated RMSE of the ESEKF and the VSLAM trajectory estimates for Trial 1 and Trial 2 respectively. In these tables, it can be seen that ESEKF in each UAV reduces the error of the trajectories estimated by VSLAM in all axes. This reduction of the error in both trials is approximately 22%, 78% and 48% in the Longitudinal, Lateral and Height axes respectively. Even though the RMSE of the UAVs trajectory estimates can only be partially calculated in this experiment, the accuracy of the entire estimated trajectories from the cooperative ESEKF and VSLAM can be visually assessed by the GPS positions at the start and end of both trials. Fig. 5.6 shows the estimated trajectories of each UAV using the cooperative ESEKF (green) and VSLAM alone (red) on a satellite image of the environment as well as the measured start and end points of the three UAVs from Trial 1.

The results from Case 1 in both trials show that the estimated trajectories of the cooperative ESEKF have higher accuracy than VSLAM alone in all three axes. It can be seen in Fig. 5.6 that the estimated trajectories of the cooperative ESEKF follow the true path of the UAVs in Trial 1. This figure also shows that the cooperative ESEKF yields smooth trajectories in all axes. From the results of both trials, the estimated trajectories of UAV_2's

ESEKF are the most accurate because UAV_2 corrects its position estimates with the shared position estimates from UAV_1 and UAV_3. It can also be noted from the figures and tables of Case 1 that the trajectories estimated by the cooperative ESEKF in the Longitudinal axis do not have the same degree of accuracy as in the Lateral and Height axes. Fig. 5.6 shows the error of the estimated trajectories of the ESEKF in the longitudinal axis. This error is the difference in distance between the measured GPS position and the estimated position of the ESEKF at the end of Trial 1. This error is approximately UAV_1= 10 m, UAV_2= 6 m, and UAV_3= 5 m.

For Case 2: Fig. 5.7 shows the estimated trajectories of each UAV using VSLAM alone (red) and cooperative ESEKF (green) for Trial 1. This figure shows that the UAV's estimated trajectories using the ESEKF are more accurate in the longitudinal direction than the estimated trajectories of Case 1 in the same direction. This increase in accuracy is because the relative localisation system's position estimates are trusted more than the position estimates of the VSLAM algorithm. The error in distance is approximately UAV_1= 8 m, UAV_2= 4 m, and UAV_3= 3 m. in the longitudinal axis of Trial 1. However, in Fig. 5.7, it is also shown that the UAV trajectories estimated by the ESEKF oscillated to a higher degree than the for Case 1. This is because the correction form the relative localisation system is trusted more but evaluated less frequently, leading to a more aggresive correction than for case 1.

Case 3: Fig. 5.8 shows the estimated trajectories of each UAV using VSLAM alone (red) and cooperative ESEKF (green) for Trial 1. This figure shows a similar improvement in accuracy

of the estimated trajectories in the Longitudinal axis as in Case 2. The error in distance is approximately UAV_1= 9 m, UAV_2= 5 m, and UAV_3= 4 m in the longitudinal axis. Moreover, Fig. 5.8 also shows that the estimated trajectories are smoother than the for Case2. However, the estimated trajectories are not as smooth as for Case 1. This es because they still undergo a relatively aggressive correction.

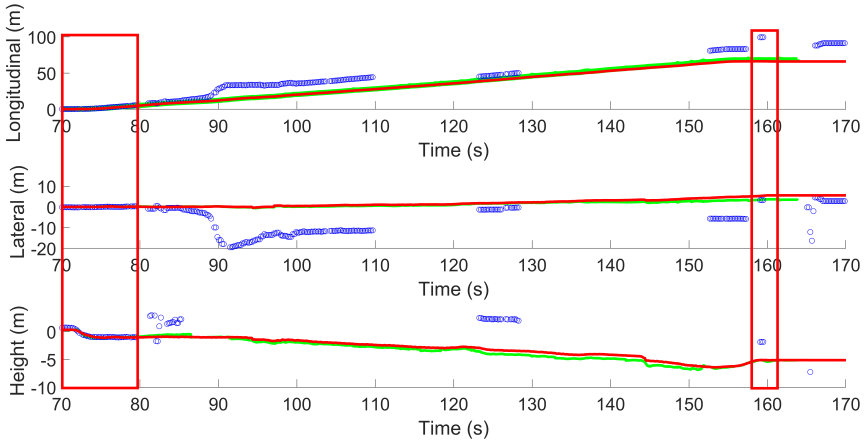
Experiment 2

Fig.5.9 shows the estimated trajectories of each UAV using VSLAM alone (red), cooperative ESEKF (green) and GPS ground truth measurements (blue) for Trial 1. This figure confirms that the trajectories estimated by VSLAM in Experiment 2 are less accurate than those in Experiment 1 due to the lack of visual features in the environment. Fig. 5.10 shows the error between the trajectories estimated by ESEKF (green) and VSLAM (red) and each UAV's GPS ground truth trajectories for Trial 1. It can be seen in this figure that the trajectory estimated by VSLAM for UAV_3 in the Longitudinal axis has a higher accuracy than for UAV_1 and UAV_2. This is because was closer to some visual features such as a fence than the other two UAVs. However, this figure also shows that the trajectory estimated by ESEKF for UAV_3 in the Longitudinal axis is less accurate than trajectory estimated by VSLAM on the same axis. This is confirmed by Table 5.3 and Table 5.4, which show the RMSE of the trajectories estimated by ESEKF and VSLAM in each axis for Trial 1 and Trial 2 respectively. From these tables, it can be seen that the trajectory of UAV_3 estimated by ESEKF in the Longitudinal axis has an approximate increase of 0.7 m when compared with the RMSE of the trajectory estimated by VSLAM in the same axis.

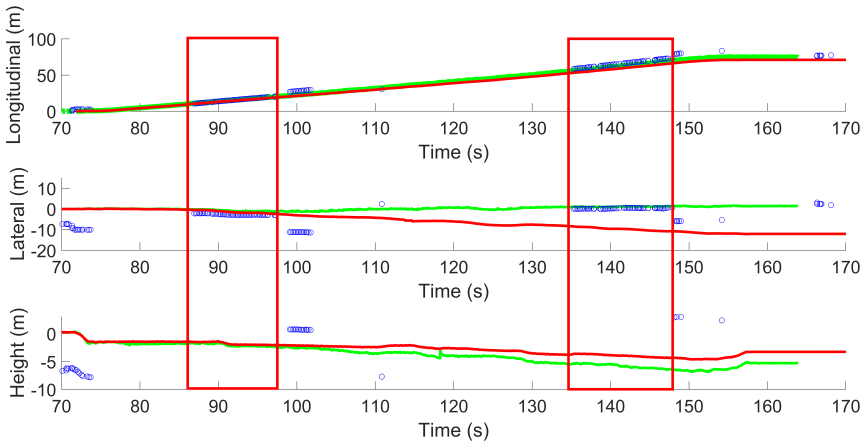
The propagated errors of the ESEKF position estimates for UAV_1 and UAV_2 are transmitted to UAV_3 by the relative localisation system's position estimates. Moreover, the shared

position estimates are also bounded the linear formation. This means that the trajectory estimates of UAV_3 are affected by the error of UAV_1 and UAV_2 as well as the linear formation followed by the UAVs. Also, the oscillations that are present in Fig.5.9 and Fig. 5.10 are because the selected noise covariance matrix trusted the relative localisation measurement more than the VSLAM position estimate. This forced an aggressive correction of the position estimates at a rate of approximately 4 Hz, while a mild correction was performed at a rate of 20 Hz and the prediction at a rate of 100 Hz. The different rates of the corrections and the difference in the amount of correction performed generated the oscillations.

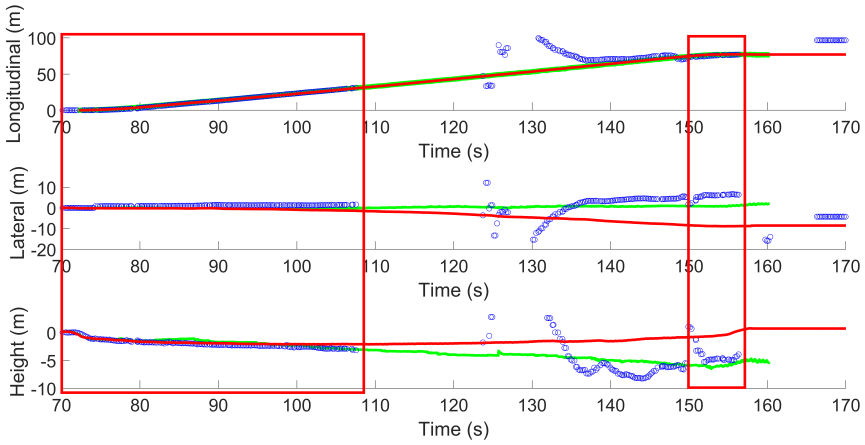
The results of experiment 2 also showed that proposed ESEKF can correct the position estimates of the UAVs, even when the VSLAM algorithm lacks sufficient features to determine the position of the UAV at an acceptable level. However, the proposed method heavily relies on the position estimates of the VSLAM algorithm in order to correct the position of the UAVs.



(a) UAV_1



(b) UAV_2



(c) UAV_3

Fig. 5.5 ESEKF's etimated trajectories in the GPS-denied environment of Trial 1 Case 1, where the red squares show when the RTK-float solution is available. Blue: GPS trajectory estimate, Red: VSLAM algorithm's trajectory estimate, and Green: ESEKF's position estimate.

UAV	Axis	RMSE VSLAM (m)	RMSE ESEKF (m)
1	Longitudinal	15.00	11.13
	Lateral	7.70	2.43
	Height	27.38	15.34
2	Longitudinal	9.40	5.79
	Lateral	11.06	1.49
	Height	9.07	6.08
3	Longitudinal	6.27	4.95
	Lateral	9.49	2.35
	Height	29.74	10.93

Table 5.1 GPS-denied environment: Trail 1 Case 1 RMSE of the trajectories estimated by VSLAM alone and the ESEKF per UAV.

UAV	Axis	RMSE VSLAM (m)	RMSE ESEKF (m)
1	Longitudinal	16.83	12.29
	Lateral	6.38	3.48
	Height	26.51	13.50
2	Longitudinal	8.89	4.98
	Lateral	13.32	2.27
	Height	8.07	6.68
3	Longitudinal	5.87	3.56
	Lateral	8.96	1.88
	Height	26.94	9.39

Table 5.2 GPS-denied environment: Trail 2 Case 1 RMSE of the trajectories estimated by VSLAM alone and the ESEKF per UAV.

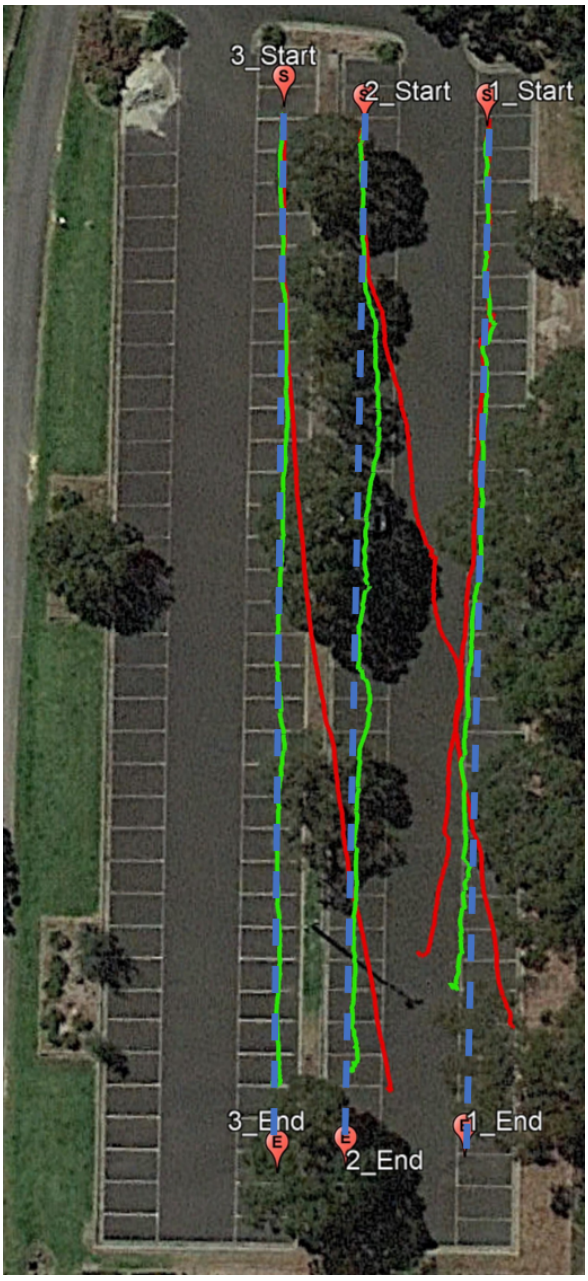


Fig. 5.6 Satellite image (sourced from Google maps) for Trial 1 Case 1 trajectories estimated by the ESEKF in the GPS-denied environment showing the starting point and end point in the trial. Green: UAV_1, UAV_2, and UAV_3 ESEKF's trajectory estimates, RED: UAV_1, UAV_2, and UAV_3 VSLAM algorithm's position, and BLUE dotted line: expected trajectory.

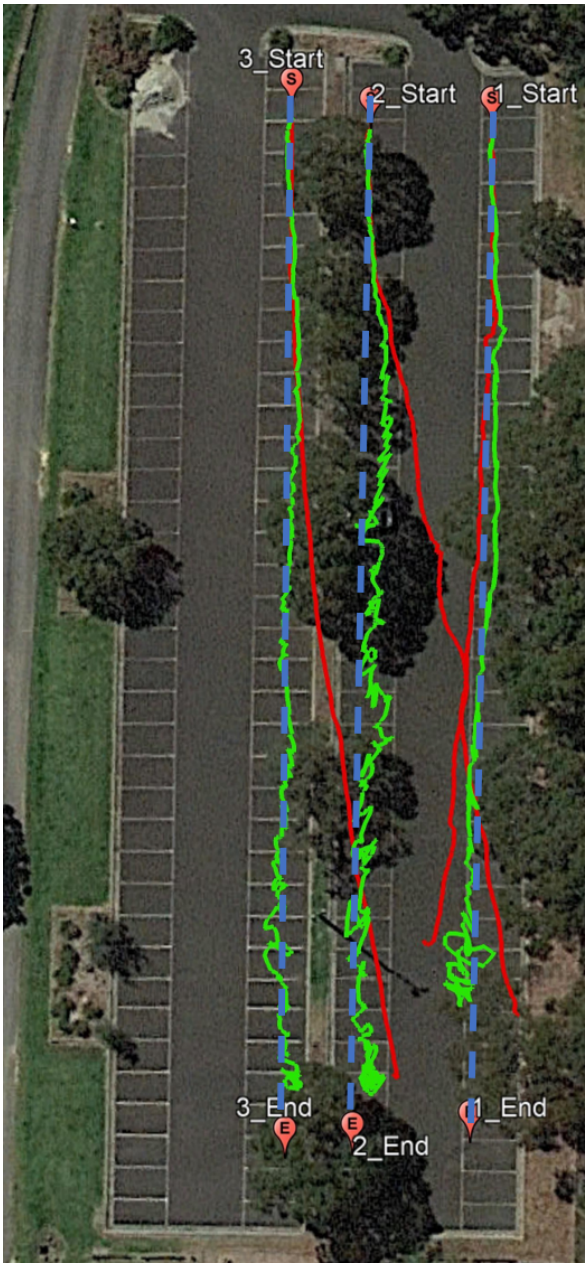


Fig. 5.7 Satellite image (sourced from Google maps) for Trial 1 Case 2 trajectories estimated by the ESEKF in the GPS-denied environment showing the starting point and end point in the trial. Green: UAV_1, UAV_2, and UAV_3 ESEKF's trajectory estimates, RED: UAV_1, UAV_2, and UAV_3 VSLAM algorithm's position, and BLUE dotted line: expected trajectory.

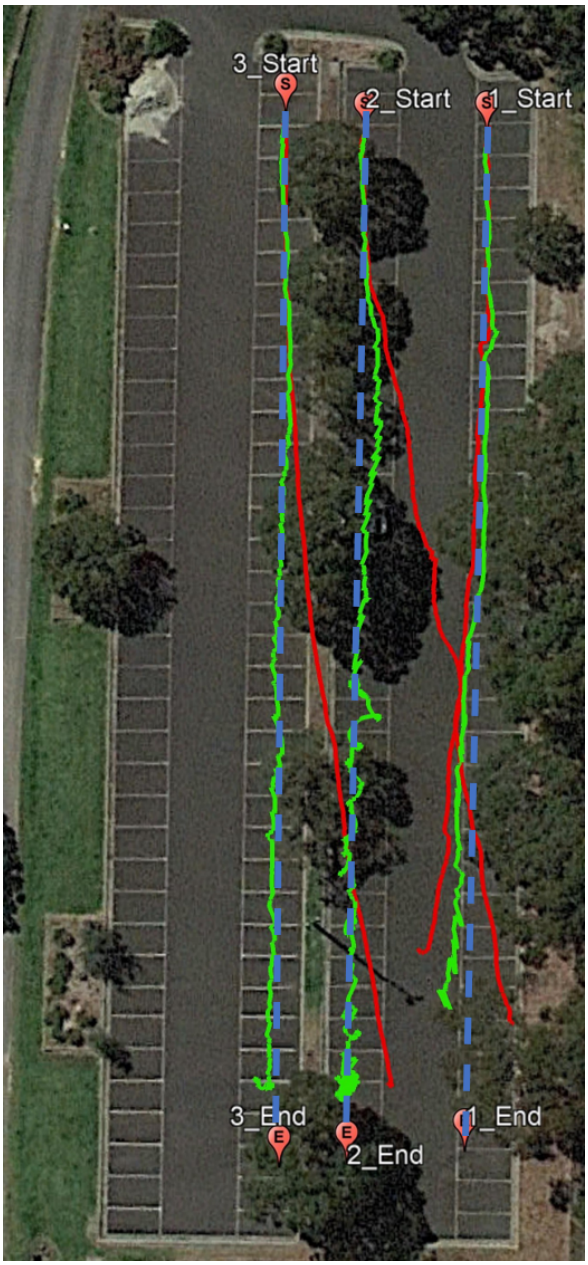
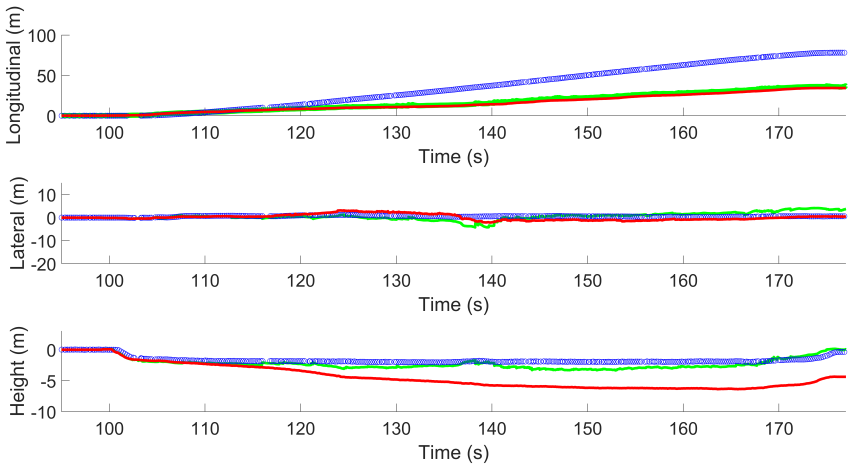
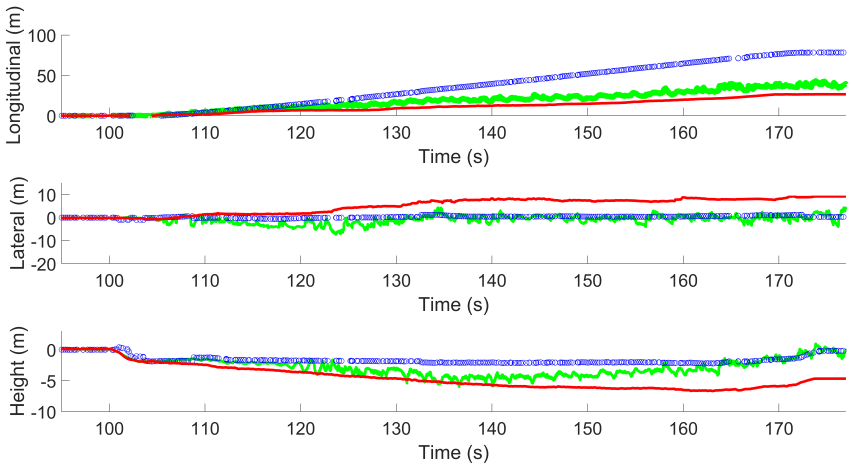


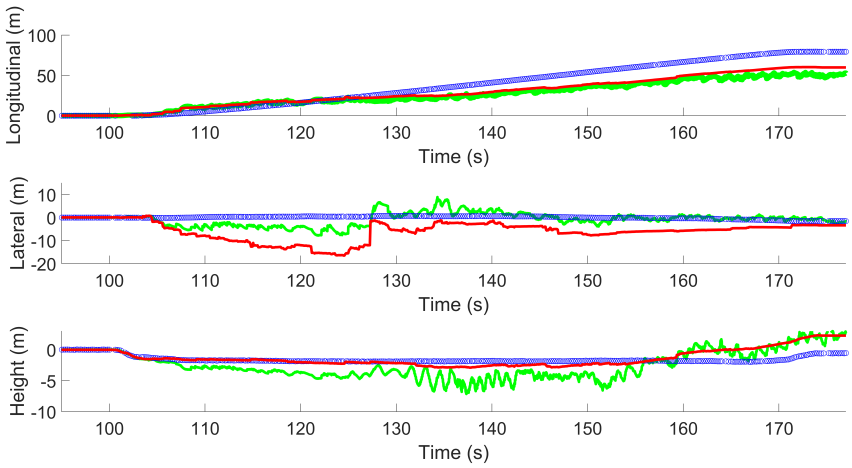
Fig. 5.8 Satellite image (sourced from Google maps) for Trial 1 Case 3 trajectories estimated by the ESEKF in the GPS-denied environment showing the starting point and end point in the trial. Green: UAV_1, UAV_2, and UAV_3 ESEKF's trajectory estimates, RED: UAV_1, UAV_2, and UAV_3 VSLAM algorithm's position, and BLUE dotted line: expected trajectory.



(a) UAV_1

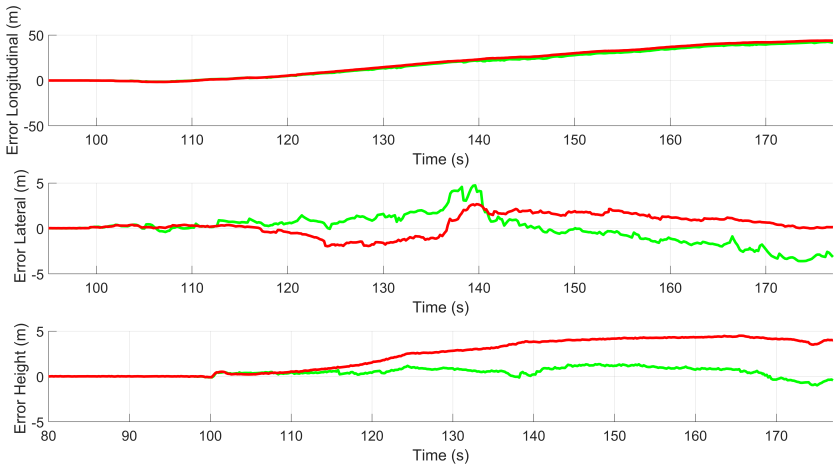


(b) UAV_2

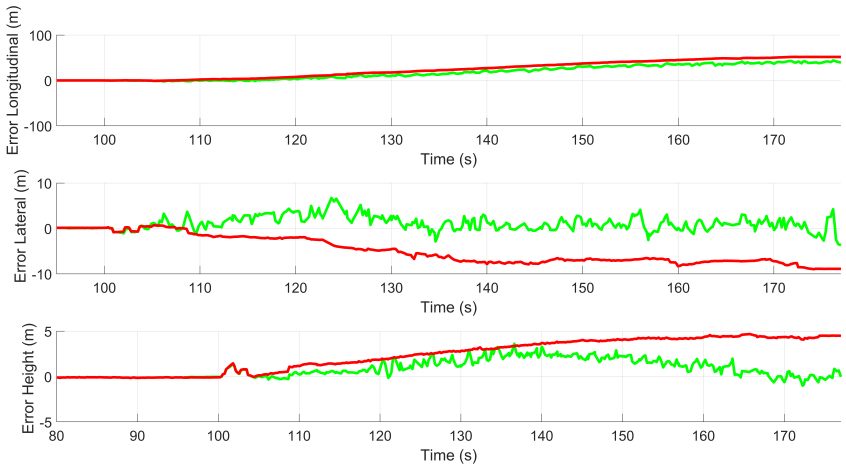


(c) UAV_3

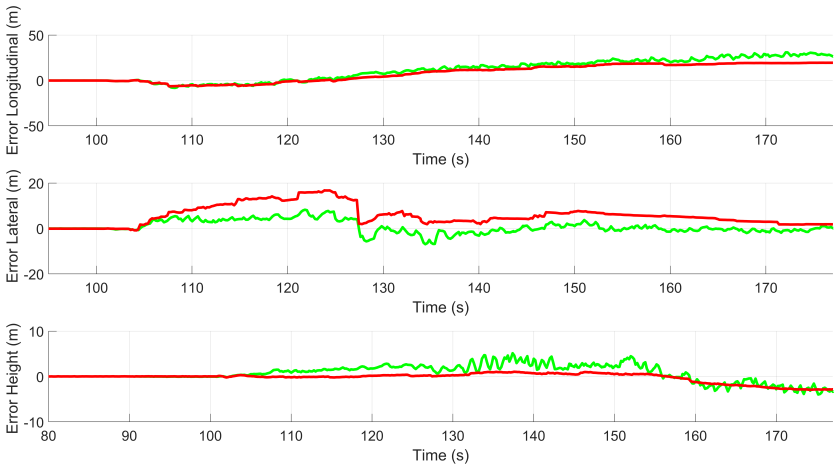
Fig. 5.9 trajectories estimated by the ESEKF in the open field for Trial 1. Blue: GPS trajectory estimate. Red: VSLAM algorithm's trajectory estimate, and Green: ESEKF's position estimate.



(a) UAV_1



(b) UAV_2



(c) UAV_3

Fig. 5.10 Error plot trajectories estimated by the ESEKF in the open field for Trial 1. Red: VSLAM algorithm's trajectory estimate, and Green: ESEKF's position estimate.

UAV	Axis	RMSE VSLAM (m)	RMSE ESEKF (m)
1	Longitudinal	24.88	18.01
	Lateral	1.03	1.02
	Height	2.96	1.28
2	Longitudinal	30.95	24.49
	Lateral	5.44	1.96
	Height	2.96	1.45
3	Longitudinal	12.45	13.19
	Lateral	5.72	4.07
	Height	1.46	1.00

Table 5.3 Open field environment: Trail 1 RMSE of the trajectories estimated by VSLAM alone and the ESEKF per UAV

UAV	Axis	RMSE VSLAM (m)	RMSE ESEKF (m)
1	Longitudinal	27.49	17.18
	Lateral	1.91	1.77
	Height	3.96	1.70
2	Longitudinal	33.52	25.61
	Lateral	6.24	2.85
	Height	3.28	1.65
3	Longitudinal	15.52	16.12
	Lateral	6.59	3.28
	Height	1.15	0.91

Table 5.4 Open field environment: Trail 2 RMSE of the trajectories estimated by VSLAM alone and the ESEKF per UAV.

5.4 Summary

This chapter has presented the implementation and outdoor testing of the proposed cooperative ESEKF. The filter aims to reduce the positional drift of VSLAM's position estimates of every member of a swarm in real time. The cooperative ESEKF fuses the IMU measurements (linear acceleration and angular velocity), the stereo camera's position and rotation estimates, and the relative localisation system's position estimates in order to reduce the drift

of the VSLAM's position estimates. Two flying experiments of two repetitions each were performed in different outdoor environments using three UAVs to determine the accuracy of the cooperative ESEKF's position estimates.

The first experiment was conducted in a GPS-denied environment where the three UAVs flew in a forward direction while maintaining a linear formation. This experiment was used to determine the accuracy of the cooperative ESEKF's trajectory estimates as well as to tune the noise covariance matrix for the GPS-denied environment using three cases. The results show that the ESEKF reduces the error of the VSLAM position estimates by approximately 22%, 78% and 48% in the Longitudinal, Lateral and Height axes respectively. For the tuning of the noise covariance matrix, the results show that trusting the relative localisation system's position estimates more in lateral axis and the VSLAM algorithm in the Longitudinal and Height axes yield accurate trajectory estimates with oscillations.

The second experiment was performed in an open field environment where the three UAVs flew in a forward direction while maintaining a linear formation. This experiment was used to asses the RMSE of the entire estimated trajectories of the cooperative ESEKF in all axes. The results for the experiment show that the cooperative ESEKF reduces the error of the VSLAM algorithm's trajectory estimates. However, for the UAV_3 trajectory estimated by the ESEKF in the Longitudinal axis has a greater error than VSLAM in the same axis. This is because the trajectory estimates of UAV_3 are affected by the error of UAV_1 and UAV_2 as well as the linear formation followed by the UAVs.

The experimental results presented in this chapter have shown that the proposed cooperative ESEKF improves the accuracy of the VSLAM position estimates in visually rich GPS-denied environments. The open field experiments have shown that the benefits of

ESEKF are reduced in visually poor environments when the VSLAM is less effective for position estimates. However, it is most likely that this visually poor environments would have access to GPS. Using the system developed in this chapter, a swarm of UAVs could perform cooperative SLAM without map sharing, loop closure or multi-session SLAM. This would enable a decentralised swarm of UAVs with limited resources to navigate a GPS-denied unstructured environment.

6. Conclusion

6.1 Thesis Summary and Research Contributions

The research presented in this thesis has developed methods and technologies that enable a swarm of UAVs with limited resources to perform cooperative simultaneous localisation and mapping (SLAM) efficiently in unstructured outdoor environments when GPS is unavailable. This research was motivated by the difficulty of accurately mapping under tree canopies using sensor technologies, but could be applied to many other GPS-occluded environments.

The objective of this research was to use a swarm of UAVs to cooperatively reduce the drift of each UAV's SLAM position estimate in order to map an unstructured outdoor area faster, more accurately and more robustly than a single UAV could. Such a system has been realised by addressing the following specific research objectives:

- Creating an accurate, computationally efficient and lightweight relative localisation system that can be mounted on UAVs to estimate the position of its immediate neighbours and share this position.
- Designing an Error Sate Extended Kalman Filter that fuses the visual SLAM's position estimates with the IMU measurements for each UAV and the shared relative localisation system's position estimates from other UAVs.

- Creating a decentralised Error Sate Extended Kalman Filter that can cooperatively reduce the drift in the position estimate of each UAV's visual SLAM algorithm in the swarm.

In Chapter 3, this research presented a novel, computationally efficient relative localisation sensor based on a 180° FoV camera, a Mulle-UWB ranging sensor and an IMU placed on a measuring UAV. A distinctive coloured marker and Mulle-UWB were placed on a second target UAV whose position was to be estimated. The measuring UAV used the Camshift algorithm for colour tracking algorithm to determine the position of the marker on an image. To estimate the position of the UAV in 3D space the geometric pinhole camera model was used. This model allows the triangulation of the position based on the ranging measurements from the Mulle-UWB and the position of the marker on the image. When the line of sight between UAVs is temporarily broken, a Kalman Filter was implemented in the Camshift algorithm. This enabled to predict the position of the marker until the line of sight was restored. In order to share the position estimates from the relative localisation system, these were translated the world frame.

To determine the accuracy of the relative localisation system, two types of experiments with 15 repetitions each were performed in an indoor environment while the ground truth was measured by the motion capture system. The first experiment comprised of the measuring UAV estimating the position of the target UAV without any occlusions. The second experiment explores the case when an obstacle temporarily broke the visual line of sight between UAVs. The results from these experiments show that the relative localisation system estimates the trajectory even when the visual line of sight is temporarily broken. The estimated trajectories of the relative localisation system have a greater error in Longitudinal and Height axes than in the Lateral axis. Therefore, relative localisation system is a viable

option when applied to a swarm of UAVs that fly in a linear formation. This is because the estimation of the position in the Lateral axis is the important in order to avoid collisions between members. However, this is only the case when the relative localisation frame and is aligned with the NED frame.

In Chapter 4, an Error State Extended Kalman Filter (ESEKF) was developed in order to reduce the drift of the VSLAM's position estimates. This was achieved by fusing the IMU's linear acceleration and angular velocity measurements with the VSLAM position an rotation estimates and the shared relative position estimate from a neighbouring UAV. The filter predicted the position of the UAV based on the IMU measurements and corrected this prediction every time the VSLAM algorithm computed a position estimation or when a relative position was shared with the UAV.

To assess whether the position yielded by the ESKEF reduced the drift of the VSLAM algorithm and to determine whether it could be implemented in a swarm of UAVs, a set of indoor tests with 15 repetitions was performed. The RMSE between the VSLAM and the ground truth measured trajectories from the motion capture system was calculated, as well as the RMSE between the ESKEF estimated trajectories and the ground truth. The tests used two UAVs, where UAV_1 remained stationary and oriented with the world frame while measuring the position of UAV_2. UAV_2 determined its position within the environment via the VSLAM algorithm. Comparing the errors in position estimates given by each of these systems, it was shown that using the ESKEF trajectory estimates resulted in a reduction in drift compared VSLAM estimated trajectories by Longitudinal $= 24\%$, Lateral $= 75\%$, and Height $= 11\%$. These results show that with the help of a neighbouring UAV_1 and its relative localisation system, the implemented ESEKF reduces the drift of the VSLAM algorithm from UAV_2 in all axes, without the necessity of sharing a map of the environment.

In Chapter 5, a novel approach to cooperative SLAM for a swarm of UAVs was developed. the system did not require sharing parts of the maps produced by each UAV to improve the position estimation of single VSLAM by an individual UAV. To achieve this, for each UAV the ESEKF and relative localisation system were used to determine the position of a neighbouring UAV. This position estimate was then shared with the neighbour, to be used as an input into the neighbour's ESEKF. This resulted in an approach that reduced the drift in the single VSLAM position estimates of every UAV compared with VSLAM alone. This approach also made the swarm scalable, because the number of UAVs could be increased without increasing the computational complexity of the individual ESKEF in each UAV.

The experimental results show that, in a setting where the UAVs only move forward, the position estimate from the VSLAM alone will drift and, without correction, this may cause collisions within the swarm or with trees and other obstacles. Therefore, correction using the distributed ESEKF is an effective method for a UAV navigation system to reduce position estimate error. For the trials conducted in this research, the average reduction in error was approximately Longitudinal $= 39\%$, Lateral $= 74\%$, Height $= 48\%$. These results show that in a GPS-denied environment the standalone VSLAM algorithm will drift without loop closure even if the area to be mapped is rich in features. Therefore, without this distributed swarm system the UAVs might collide with obstacles such as trees or other members of the swarm. The approach implemented corrected the position estimation in all UAVs when GPS was occluded without map sharing. Drift in position estimation was reduced even when the VSLAM had reduced effectiveness due to the lack of features, although this benefit was reduced. This system is also scalable as the increase in of UAVs into the system will not impact the formulation of the individual filters.

In summary, the main research contributions of this thesis are:

1. A novel distributed approach for cooperative SLAM to be implemented in a swarm of UAVs; this approach improves the estimation of the single SLAM output of every member of the SRS and allows the system to be scalable.

2. The design of an ESEKF that fuses the position output from a stereo camera, an IMU and the shared relative position.

3. The development and implementation of an relative localisation system that is computationally lightweight and robust to the temporary loss of the line of sight with other members of the swarm.

6.2 Limitations and Future Research

During the experimentation of the proposed cooperative ESEKF for a swarm of UAVs, it was noted that the accuracy of the position estimates is affected by three main factors: the alignment of the world frame, the trajectory followed by the UAVs, and the environment that the swarm of UAVs is exploring. In the proposed system, a UAVs' longitudinal and lateral navigational trajectories have to be aligned with the North and the East axes. This is to allow the use of the diagonal form of the covariance matrix for the relative localisation system's position estimates in the ESEKF formulation. Moreover if the longitudinal and lateral navigational trajectories of the UAVs change during flight, the covariance matrix for the relative localisation system's position estimates has to be re-estimated when the change occurs. This will also make the use of diagonal form of the covariance matrix invalid in the ESEKF formulation. Therefore it is necessary to orient the swarm system's trajectory to the NED frame to use the current model.

Also, during experimentation, it was noticed that the sunlight affected the cameras used for the relative localisation system when they were facing towards the sun. This caused the images to become overexposed, which meant that the relative localisation system lost track of the UAVs. One possible solution is to use an algorithm to adjust the exposure of the camera to adapt to environmental conditions. In general, the relative localisation system and the stereo cameras used for VSLAM are affected by the environmental light. Therefore, when performing experiments, one has to be mindful that these systems may lose accuracy in such conditions.

The accuracy of the VSLAM algorithm's trajectory estimates is affected by the visual features available in the environment. This means that the measurement noise covariance matrix has to be tuned accordingly to reflect the accuracy of the VSLAM algorithm in different environments. Moreover because the error is propagated between the UAVs, the accuracy of the cooperative ESEKF's trajectory estimates are affected by the accuracy of the VSLAM algorithm. To guarantee accurate trajectory estimates, the UAVs have to be placed in an environment rich in visual features.

This PhD research has built a framework for improving the position estimation of VSLAM by using cooperative ESEKF in swarm of UAVs. The tests performed show that the cooperative system can reduce the drift in the individual VSLAM position estimations. However, drift could potentially be reduced even further if the number of members was increased or the formation adopted by the swarm. To explore this more fully, the next stage in this research is to implement the cooperative ESEKF in a larger number of UAVs in order to create an autonomous swarms of varying sizes and configurations to test this hypothesis.

References

[1] ABBAS, R., AND QINGHE, W. Formation tracking for multiple quadrotor based on sliding mode and fixed communication topology. In *Intelligent Human-Machine Systems and Cybernetics (IHMSC), (2013) 5th International Conference on*, vol. 2, pp. 233–238.

[2] ACHTELIK, M., BACHRACH, A., HE, R., PRENTICE, S., AND ROY, N. Stereo vision and laser odometry for autonomous helicopters in GPS-denied indoor environments. (2002). vol. 7332, p. 733219.

[3] ADAMS, M. SLAM — algorithmic advances, loop closing, measurement classification and outdoor implementations. *Robotics and Autonomous Systems 55*, 1 (2007), 1–2.

[4] AHMAD, A., AND LIMA, P. Multi-robot cooperative spherical-object tracking in 3D space based on particle filters. *Robotics and Autonomous Systems 61*, 10 (2013), 1084–1093.

[5] AHMADI, M., KHAYATIAN, A., AND KARIMAGHAEE, P. Orientation estimation by error-state extended kalman filter in quaternion vector space. In *SICE, (2007) Annual Conference*, IEEE, pp. 60–67.

[6] AILIPU TECHNOLOGY COLTD. ELP. http://www.webcamerausb.com/180degree-fisheye-lens-1080p-wide-angle-pc-web-usb-camerausb-camera-module-for-android-windows-p-85.html. Accessed: 22-11-2018.

[7] ALLEN, J. G., XU, R. Y., AND JIN, J. S. Object tracking using camshift algorithm and multiple quantized feature spaces. In *Proceedings of the Pan-Sydney area workshop on Visual information processing* (2004), Australian Computer Society, Inc., pp. 3–7.

[8] ALMURIB, H. A. F., NATHAN, P. T., AND KUMAR, T. N. Control and path planning of quadrotor aerial vehicles for search and rescue. In *SICE Annual Conference (SICE), Proceedings of (2011)*, pp. 700–705.

[9] ANDERSONE, I. The characteristics of the map merging methods: A survey. *Scientific Journal of Riga Technical University 43* (2010), 113–121.

[10] ASCORTI, L. An application of the extended kalman filter to the attitude control of a quadrotor. (2013).

[11] ASMAR, D. C., ZELEK, J. S., AND ABDALLAH, S. M. Tree trunks as landmarks for outdoor vision SLAM. In *Computer Vision and Pattern Recognition Workshop, (2006). CVPRW-2006. Conference on*, pp. 196–196.

[12] BAILEY, T., AND DURRANT-WHYTE, H. Simultaneous localization and mapping (SLAM): part ii. *Robotics Automation Magazine, IEEE 13*, 3 (2006), 108–117.

[13] BAROOAH, P., RUSSELL, W. J., AND HESPANHA, J. P. *Approximate Distributed Kalman Filtering for Cooperative Multi-agent Localization*. (2010), pp. 102–115.

[14] BEN-ARI, M., AND MONDADA, F. *Elements of Robotics*. Springer, (2018).

[15] BINAZZI, G., CHISCI, L., CHITI, F., FANTACCI, R., AND MENCI, S. Localization of a swarm of mobile agents via unscented kalman filtering. In *IEEE ICC (2009) proceedings*.

[16] BONIN-FONT, F., BELTRAN, J.-P., AND OLIVER, G. Multisensor aided inertial navigation in 6DOF AUVs using a multiplicative error state kalman filter. In *OCEANS-Bergen, (2013) MTS/IEEE*, IEEE, pp. 1–7.

[17] CADENA, C., AND NEIRA, J. SLAM in with the combined kalman-information filter. *Robotics and Autonomous Systems 58*, 11 (2010), 1207–1219.

[18] CARLONE, L., KAOUK NG, M., DU, J., BONA, B., AND INDRI, M. Simultaneous localization and mapping using rao-blackwellized particle filters in multi robot systems. *Journal of Intelligent Robotic Systems 63*, 2 (2011), 283–307.

[19] CARLONE, L., MACCHIA, V., TIBALDI, F., AND BONA, B. Quaternion-based EKF-SLAM from relative pose measurements: observability analysis and applications. *Robotica 33*, 06 (2015), 1250–1280.

[20] CARLONE, L., NG, M. K., JINGJING, D., BONA, B., AND INDRI, M. Rao-Blackwellized particle filters multi robot SLAM with unknown initial correspondences and limited communication. In *Robotics and Automation (ICRA), IEEE International Conference on (2010)*, pp. 243–249.

[21] CHAI, T., AND DRAXLER, R. R. Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature. *Geoscientific model development 7*, 3 (2014), 1247–1250.

[22] CHAMBERS, A., SCHERER, S., YODER, L., JAIN, S., NUSKE, S., AND SINGH, S. Robust multi-sensor fusion for micro aerial vehicle navigation in GPS-degraded/denied environments. In *American Control Conference (ACC), (2014)*, IEEE, pp. 1892–1899.

[23] CHOUDHARY, S., CARLONE, L., NIETO, C., ROGERS, J., CHRISTENSEN, H. I., AND DELLAERT, F. Distributed mapping with privacy and communication constraints: Lightweight algorithms and object-based models. *The International Journal of Robotics Research 36*, 12 (2017), 1286–1311.

[24] CHROBOTICS. *UM7 DATASHEET*, (2014). Rev. 1.3.

[25] CIHAN, U., AND HAKAN, T. 3D multi-layered normal distribution transform for fast and long range scan matching. *Springer Science+Business Media B.V.* (2012).

[26] CONGDAO, H., ZHIYU, X., JILIN, L., AND ERYONG, W. Stereo vision based slam in outdoor environments. In *Robotics and Biomimetics, (2007). ROBIO 2007. IEEE International Conference on*, pp. 1653–1658.

[27] CORNEJO, A., AND NAGPAL, R. Long-lived distributed relative localization of robot swarms. *cs.RO* (2013).

[28] COUCEIRO, M. S., ROCHA, R. P., FERREIRA, N. M. F., AND VARGAS, P. A. Darwinian robotic swarms for exploration with minimal communication. In *Evolutionary Computation (CEC), IEEE Congress on (2013)*, pp. 127–134.

[29] CRASSIDIS, J. L. Sigma-point kalman filtering for integrated gps and inertial navigation. *IEEE Transactions on Aerospace and Electronic Systems 42*, 2 (2006), 750–756.

[30] DAS, A. K., FIERRO, R., KUMAR, V., OSTROWSKI, J. P., SPLETZER, J., AND TAYLOR, C. J. A vision-based formation control framework. *Robotics and Automation, IEEE Transactions on 18*, 5 (2002), 813–825.

[31] DAVISON, A. J., AND KITA, N. Active visual localisation for cooperating inspection robots. In *Intelligent Robots and Systems.(IROS 2000). Proceedings. IEEE/RSJ International Conference on (2000)*, vol. 3, IEEE, pp. 1709–1715.

[32] DECAWAVE. *Ultra Wideband (UWB) Transceiver*, (2015). Rev. 2.07.

[33] DEL BIMBO, A., AND DINI, F. Particle filter-based visual tracking with a first order dynamic model and uncertainty adaptation. *Computer Vision and Image Understanding 115*, 6 (2011), 771–786.

[34] DUGAS, O., GIGUERE, P., AND REKLEITIS, I. 6DoF cooperative localization for mutually observing robots.(2013).

[35] DURRANT-WHYTE, H., AND BAILEY, T. Simultaneous localization and mapping: part i. *Robotics Automation Magazine, IEEE 13*, 2 (2006), 99–110.

[36] DURRANT-WHYTE, H., AND HENDERSON, T. C. *Multisensor data fusion*. Springer, (2008), pp. 585–610.

[37] EISTEC AB. Mulle wireless sensor platform. http://www.eistec.se/mulle/. Accessed: 22-11-2018.

[38] ENGEL, J., SCHÖPS, T., AND CREMERS, D. *LSD-SLAM: Large-Scale Direct Monocular SLAM*, vol. 8690 of *Lecture Notes in Computer Science*. Springer International Publishing, (2014), book section 54, pp. 834–849.

[39] ERYONG, W., WENHUI, Z., GUOJUN, D., AND QICONG, W. Monocular vision SLAM for large scale outdoor environment. In *Mechatronics and Automation. ICMA (2009). International Conference on*, pp. 2037–2041.

[40] FORSMAN, P., AND HALME, A. 3D mapping of natural environments with trees by means of mobile perception. *Robotics, IEEE Transactions on 21*, 3 (2005), 482–490.

[41] FORSTER, C., CARLONE, L., DELLAERT, F., AND SCARAMUZZA, D. IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation. Georgia Institute of Technology. (2015).

[42] FORSTER, C., LYNEN, S., KNEIP, L., AND SCARAMUZZA, D. Collaborative monocular SLAM with multiple micro aerial vehicles. In *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on (2013)*, pp. 3962–3970.

[43] FORSTER, C., PIZZOLI, M., AND SCARAMUZZA, D. SVO: Fast semi-direct monocular visual odometry. In *Robotics and Automation (ICRA), IEEE International Conference on (2014)*, pp. 15–22.

[44] FORSYTH, D. A., AND PONCE, J. Computer vision: A modern approach. *Computer vision: A modern approach* (2003), 88–101.

[45] FRANCHI, A., ORIOLO, G., AND STEGAGNO, P. Mutual localization in multi-robot systems using anonymous relative measurements. *The International Journal of Robotics Research* (2013).

[46] FRAUNDORFER, F., ENGELS, C., AND NISTER, D. Topological mapping, localization and navigation using image collections. In *Intelligent Robots and Systems. IROS (2007). IEEE/RSJ International Conference on*, pp. 3872–3877.

[47] GAMAGE, R., AND TUCERYAN, M. An experimental distributed framework for distributed simultaneous localization and mapping. (2016).

[48] GAYATHRI, T., ANEESH, R., AND NAYAR, G. R. Feature based simultaneous localisation and mapping. In *Circuits and Systems (ICCS), IEEE International Conference on* (2017), IEEE, pp. 419–422.

[49] GENTNER, C., AND ULMSCHNEIDER, M. Simultaneous localization and mapping for pedestrians using low-cost ultra-wideband system and gyroscope. In *Indoor Positioning and Indoor Navigation (IPIN), (2017) International Conference on*, IEEE, pp. 1–8.

[50] GIL, A., REINOSO, S., BALLESTA, M., AND JULIÁ, M. Multi-robot visual SLAM using a rao-blackwellized particle filter. *Robotics and Autonomous Systems 58*, 1 (2010), 68–80.

[51] GUERRA, E., MUNGUIA, R., AND GRAU, A. Monocular SLAM for autonomous robots with enhanced features initialization. *Sensors (Basel, Switzerland) 14*, 4 (2014), 6317–6337.

[52] GUIVANT, J. E., MASSON, F. R., AND NEBOT, E. M. Simultaneous localization and map building using natural features and absolute information. *Robotics and Autonomous Systems 40*, 2–3 (2002), 79–90.

[53] GUNEY, M. A., AND UNEL, M. Formation control of a group of micro aerial vehicles (MAVs). In *Systems, Man, and Cybernetics (SMC), IEEE International Conference on (2013)*, pp. 929–934.

[54] GUTIÉRREZ, L., CAMPO, A., DORIGO, M., AMOR, D., MAGDALENA, L., AND FÉLIX, M.-H. An open localization and local communication embodied sensor. *Sensors (Basel, Switzerland) 8*, 11 (2008), 7545–7563.

[55] HARDKERNEL. *ODROID-XU4*, (2017). Rev. 2170310.

[56] IACONO, M., AND SGORBISSA, A. Path following and obstacle avoidance for an autonomous UAV using a depth camera. *Robotics and Autonomous Systems 106* (2018), 38–46.

[57] JENNINGS, C., MURRAY, D., AND LITTLE, J. J. Cooperative robot localization with vision-based mapping. In *Robotics and Automation, (1999). Proceedings. 1999 IEEE International Conference on*, vol. 4, IEEE, pp. 2659–2665.

[58] JEONG, J., YOON, T. S., AND PARK, J. B. Multimodal sensor-based semantic 3D mapping for a large-scale environment. *Expert Systems with Applications 105* (2018), 1–10.

[59] JIE, Z., XIANGGUO, S., AND JIHONG, Y. A novel strategy for distributed multi-robot coordination in area exploration. In *Measuring Technology and Mechatronics Automation, (2009). ICMTMA '09. International Conference on*, vol. 2, pp. 24–27.

[60] KAMALI, C., AND JAIN, S. Multiplicative error state kalman filter vs nonlinear complimentary filter for a high performance aircraft attitude estimation. *Defence Science Journal 66*, 6 (2016).

[61] KANELLAKIS, C., AND NIKOLAKOPOULOS, G. Survey on computer vision for uavs: Current developments and trends. *Journal of Intelligent Robotic Systems. 87*, 1 (2017), 141–168.

[62] KARLSSON, N., GONCALVES, L., MUNICH, M., AND PIRJANIAN, P. The VSLAM algorithm for navigation in natural environments. *Korean Robotics Society*, 2 (2005), 51–67.

[63] KIRCHMAIER, U., HAWE, S., AND DIEPOLD, K. Dynamical information fusion of heterogeneous sensors for 3D tracking using particle swarm optimization. *Information Fusion 12*, 4 (2011), 275–283.

[64] KNUTH, J., AND BAROOAH, P. Distributed collaborative 3D pose estimation of robots from heterogeneous relative measurements: an optimization on manifold approach. (2014).

[65] KWON, H., AHMAD YOUSEF, K. M., AND KAK, A. C. Building 3D visual maps of interior space with a new hierarchical sensor fusion architecture. *Robotics and Autonomous Systems 61*, 8 (2013), 749–767.

[66] LE, C., AND LI, X. Sparse3D: A new global model for matching sparse RGB-D dataset with small inter-frame overlap. *Computer-Aided Design 102* (2018), 33–43.

[67] LEE, Y.-H., ZHU, C., GIORGI, G., AND GÜNTHER, C. Fusion of monocular vision and radio-based ranging for global scale estimation and drift mitigation. *arXiv preprint arXiv:1810.01346* (2018).

[68] LEMAIRE, T., BERGER, C., JUNG, I.-K., AND LACROIX, S. Vision-Based SLAM: Stereo and monocular approaches. *International Journal of Computer Vision 74*, 3 (2007), 343–364.

[69] LEÓN, A., BAREA, R., BERGASA, L., LÓPEZ, E., OCAÑA, M., AND SCHLEICHER, D. Multi-robot SLAM and map merging. (2008). In *IX Workshop of Physical Agents (WAF 08)*, pp. 171–176.

[70] LEUNG, K. Y., BARFOOT, T. D., AND LIU, H. H. Decentralized cooperative slam for sparsely-communicating robot networks: A centralized-equivalent approach. *Journal of Intelligent Robotic Systems 66*, 3 (2012), 321–342.

[71] LEWIS, F. L., ZHANG, H., HENGSTER-MOVRIC, K., AND DAS, A. *Cooperative control of multi-agent systems: optimal and adaptive design approaches*. Springer Science Business Media, (2013).

[72] LEWIS, M. A., AND TAN, K.-H. High precision formation control of mobile robots using virtual structures. *Autonomous Robots 4*, 4 (Oct 1997), 387–403.

[73] LI, J., WEI, X., AND ZHANG, G. An extended kalman filter-based attitude tracking algorithm for star sensors. *Sensors 17*, 8 (2017), 1921.

[74] LI, M., AND MOURIKIS, A. I. Improving the accuracy of EKF-based visual-inertial odometry. (2006). In *ICRA*, Citeseer, pp. 828–835.

[75] LIGORIO, G., AND SABATINI, A. M. Extended kalman filter-based methods for pose estimation using visual, inertial and magnetic sensors: Comparative analysis and performance evaluation. *Sensors 13*, 2 (2013), 1919–1941.

[76] LIU, C. B., CHEN, C. C., AND LI, X. Object tracking system in dynamic scene based on improved camshift algorithm and kalman filter. (2014). In *Applied Mechanics and Materials*, vol. 602, Trans Tech Publ, pp. 2061–2064.

[77] LUFT, L., SCHUBERT, T., ROUMELIOTIS, S. I., AND BURGARD, W. Recursive decentralized localization for multi-robot systems with asynchronous pairwise communication. *The International Journal of Robotics Research* (2018), 0278364918760698.

[78] M., I., REKLEITIS, G., DUDEK, E., AND MILIOS, E. On the positional uncertainty of multi-robot cooperative localization. In *(2002) NRL Workshop on Multi-Robot Systems: From Swarms to Intelligent Automata*, L. E. P. Alan C. Schultz, Ed., Springer Netherlands.

[79] MACHADO SANTOS, J., COUCEIRO, M., PORTUGAL, D., AND ROCHA, R. A sensor fusion layer to cope with reduced visibility in SLAM. *Journal of Intelligent and Robotic Systems (JINT), Special Issue on Autonomous Robot Systems, Springer, London* (2015).

[80] MADHAVAN, R., FREGENE, K., AND PARKER, L. E. Distributed cooperative outdoor multirobot localization and mapping. *Autonomous Robots 17*, 1 (2004), 23–39.

[81] MAE, Y., CHOI, J., TAKAHASHI, H., OHARA, K., TAKUBO, T., AND ARAI, T. Interoperable vision component for object detection and 3D pose estimation for modularized robot control. *Mechatronics 21*, 6 (2011), 983–992.

[82] MARKLEY, F. L., CHENG, Y., CRASSIDIS, J. L., AND OSHMAN, Y. Quaternion averaging. (2007).

[83] MASAR, M. A biologically inspired swarm robot coordination algorithm for exploration and surveilance. In *Intelligent Engineering Systems (INES), IEEE 17th International Conference on (2013)*, pp. 271–275.

[84] MAYBECK, P. S., AND SIOURIS, G. M. Stochastic models, estimation, and control, volume i. *IEEE Transactions on Systems, Man, and Cybernetics 10*, 5 (1980), 282–282.

[85] MCDONALD, J., KAESS, M., CADENA, C., NEIRA, J., AND LEONARD, J. J. Real-time 6-DOF multi-session visual SLAM over large-scale environments. *Robotics and Autonomous Systems 61*, 10 (2013), 1144–1158.

[86] MENG, Z., LIN, Z., AND REN, W. Leader–follower swarm tracking for networked lagrange systems. *Systems & Control Letters 61*, 1 (2012), 117–126.

[87] MITCHELL, H. B. *Multi-sensor data fusion: an introduction*. Springer Science Business Media, (2007).

[88] MLADENOVIC, D., JOVANOVIC, D., AND DENIC, N. Open source solutions in the development of military unmanned aerial systems. *Scientific Technical Review 63*, 1 (2013), 36–46.

[89] MOURIKIS, A. I., AND ROUMELIOTIS, S. I. On the treatment of relative-pose measurements for mobile robot localization. In *Robotics and Automation, ICRA 2006. Proceedings (2006) IEEE International Conference on*, IEEE, pp. 2277–2284.

[90] MULLANE, J. S., VO, B.-N., ADAMS, M. D., AND VO, B.-T. *Random Finite Sets for Robot Mapping SLAM: New Concepts in Autonomous Robotic Map Representations*, vol. 72. Springer Science Business Media, (2011).

[91] NALPANTIDIS, L., SIRAKOULIS, G. C., CARBONE, A., AND GASTERATOS, A. Computationally effective stereovision SLAM. In *Imaging Systems and Techniques (IST), IEEE International Conference on (2010)*, pp. 458–463.

[92] NVIDIA. *JETSON TX1 DEVELOPER KIT*, (2016). 24.1 Release.

[93] OPENCV TEAM. Open Source Computer Vision Library. https://opencv.org/. Accessed: 22-11-2018.

[94] OPTITRACK. *FLEX 13*, (2017). 402.1207.

[95] OZKUCUR, N. E., KURT, B., AND AKIN, H. L. *A Collaborative Multi-robot Localization Method without Robot Identification*. Springer Berlin Heidelberg, (2009), pp. 189–199.

[96] PARKER, L. E., KANNAN, B., XIAOQUAN, F., AND YIFAN, T. Heterogeneous mobile sensor net deployment using robot herding and line-of-sight formations. In *Intelligent Robots and Systems. (IROS 2003). Proceedings. IEEE/RSJ International Conference on (2013)*, vol. 3, pp. 2488–2493 vol.3.

[97] PARKER, L. E., SCHNEIDER, F. E., AND SCHULTZ, A. C. *Multi-robot systems. From swarms to intelligent automata*, vol. 111. Springer, (2005).

[98] PAULL, L., SETO, M., LEONARD, J. J., AND LI, H. Probabilistic cooperative mobile robot area coverage and its application to autonomous seabed mapping. *The International Journal of Robotics Research 37*, 1 (2018), 21–45.

[99] PILZ, U., POPOV, A. P., AND WERNER, H. Robust controller design for formation flight of quad-rotor helicopters. In *Decision and Control, 2009 held jointly with the 28th Chinese Control Conference. CDC/CCC (2009). Proceedings of the 48th IEEE Conference on*, pp. 8322–8327.

[100] PINTO, L. R. *Aerial Multi-hop Sensor Networks*. PhD Thesis, Carnegie Mellon University, (2018).

[101] PIZARRO, D., MAZO, M., SANTISO, E., MARRON, M., JIMENEZ, D., COBRECES, S., AND LOSADA, C. Localization of mobile robots using odometry and an external vision sensor. *Sensors (Basel, Switzerland) 10*, 4 (2010), 3655–3680.

[102] POST, T. *Precise localization of a UAV using visual odometry*. PhD thesis, University of Twente, (2015).

[103] PRADEEP, V., MEDIONI, G., AND WEILAND, J. Visual loop closing using multi-resolution SIFT grids in metric-topological SLAM. In *Computer Vision and Pattern Recognition, (2009). CVPR 2009. IEEE Conference on*, pp. 1438–1445.

[104] PUGH, J., RAEMY, X., FAVRE, C., FALCONI, R., AND MARTINOLI, A. A fast on-board relative positioning module for multirobot systems. *Mechatronics, IEEE/ASME Transactions on 14*, 2 (2009), 151–162.

[105] PX4 DEV TEAM. PixhawkSeries. https://docs.px4.io/en/flight_controller/pixhawk_series.html. Accessed: 22-11-2018.

[106] RAMIREZ, B., CHUNG, H., DERHAMY, H., ELIASSON, J., AND BARCA, J. C. Relative localization with computer vision and UWB range for flying robot formation control. In *Control, Automation, Robotics and Vision (ICARCV), 14th International Conference on* (2016), IEEE, pp. 1–6.

[107] RAMOS, F. T., NIETO, J., AND DURRANT-WHYTE, H. F. Recognising and modelling landmarks to close loops in outdoor SLAM. In *Robotics and Automation, (2007) IEEE International Conference on*, pp. 2036–2041.

[108] RASHID, A. T., FRASCA, M., ALI, A. A., RIZZO, A., AND FORTUNA, L. Multi-robot localization and orientation estimation using robotic cluster matching algorithm. *Robotics and Autonomous Systems 63, Part 1*, 0 (2015), 108–121.

[109] RIBAS, D., RIDAO, P., AND NEIRA, J. *Underwater SLAM for structured environments using an imaging sonar*, vol. 65. Springer, (2010).

[110] ROUMELIOTIS, S., AND BEKEY, G. A. Distributed multirobot localization. *Transactions on Robotics and Automation* (2002), 781 – 795.

[111] ROUMELIOTIS, S. I., AND BEKEY, G. A. Collective localization: a distributed kalman filter approach to localization of groups of mobile robots. In *Robotics and Automation, (2000). Proceedings. ICRA '00. IEEE International Conference on*, vol. 3, pp. 2958–2965 vol.3.

[112] SANTAMARIA-NAVARRO, A., LOIANNO, G., SOLÀ, J., KUMAR, V., AND ANDRADE-CETTO, J. Autonomous navigation of micro aerial vehicles using high-rate and low-cost sensors. *Autonomous Robots* (2017), 1–18.

[113] SARKAR, A., SRIVASTAVA, S., AND MANOJ, B. S. Elevation mapping using stereo vision enabled heterogenous multi-agent robotic network. In *Global Humanitarian Technology Conference: South Asia Satellite (GHTC-SAS), IEEE (2013)*, pp. 340–345.

[114] SCARAMUZZA, D., ACHTELIK, M. C., DOITSIDIS, L., FRIEDRICH, F., KOSMATOPOULOS, E., MARTINELLI, A., ACHTELIK, M. W., CHLI, M., CHATZICHRISTOFIS, S., KNEIP, L., GURDAN, D., HENG, L., GIM HEE, L., LYNEN, S., POLLEFEYS, M., RENZAGLIA, A., SIEGWART, R., STUMPF, J. C., TANSKANEN, P., TROIANI, C., WEISS, S., AND MEIER, L. Vision-Controlled micro flying robots: From system design to autonomous navigation and mapping in GPS-Denied environments. *Robotics Automation Magazine, IEEE 21*, 3 (2014), 26–40.

[115] SEN, Z., LIHUA, X., AND ADAMS, M. Gradient model based feature extraction for simultaneous localization and mapping in outdoor applications. In *Control, Automation, Robotics and Vision Conference, (2004). ICARCV (2004) 8th*, vol. 1, pp. 431–436 Vol. 1.

[116] SÁEZ, J. M., AND ESCOLANO, F. 6DOF entropy minimization SLAM for stereo-based wearable devices. *Computer Vision and Image Understanding 115*, 2 (2011), 270–285.

[117] SHEIJANI, M. S., GHOLAMI, A., DAVARI, N., AND EMAMI, M. Implementation and performance comparison of indirect Kalman filtering approaches for AUV integrated navigation system using low cost IMU. In *21st Iranian Conference on Electrical Engineering (ICEE),* (2013), pp. 1–6.

[118] SIAGIAN, C., AND ITTI, L. Biologically-inspired robotics vision monte-carlo localization in the outdoor environment. In *Intelligent Robots and Systems,(2007). IROS 2007. IEEE/RSJ International Conference on*, pp. 1723–1730.

[119] SICILIANO, B., AND KHATIB, O. *Springer handbook of robotics*. Springer, 2016.

[120] SIRTKAYA, S., SEYMEN, B., AND ALATAN, A. A. Loosely coupled kalman filtering for fusion of visual odometry and inertial navigation. In *Information Fusion (FUSION), 16th International Conference on (2013)*, IEEE, pp. 219–226.

[121] SOLA, J. Quaternion kinematics for the error-state kalman filter. *arXiv preprint arXiv:1711.02508* (2017).

[122] SOLANAS, A., AND GARCIA, M. A. Coordinated multi-robot exploration through unsupervised clustering of unknown space. In *International Conference on Intelligent Robots and Systems. (2004).*, vol. 1, IEEE, pp. 717 – 721.

[123] SONG, Y., NUSKE, S., AND SCHERER, S. A multi-sensor fusion mav state estimation from long-range stereo, IMU, GPS and barometric sensors. *Sensors 17*, 1 (2016), 11.

[124] SPERATI, V., TRIANNI, V., AND NOLFI, S. Self-organised path formation in a swarm of robots. *Swarm Intelligence 5*, 2 (2011), 97–119.

[125] STEGAGNO, P., COGNETTI, M., ROSA, L., PELITI, P., AND ORIOLO, G. Relative localization and identification in a heterogeneous multi-robot system. In *Robotics and Automation (ICRA), IEEE International Conference on (2013)*, IEEE, pp. 1857–1864.

[126] STEREOLABS INC. Stereo labs. https://www.stereolabs.com/docs/getting-started/. Accessed: 22-11-2018.

[127] SWIFT NAVIGATION. *PIKSI MULTI*, (2017). Version 1.0.7.

[128] TESSIER, C., DEBAIN, C., CHAPUIS, R., AND CHAUSSE, F. Map aided localization and vehicle guidance using an active landmark search. *Information Fusion 11*, 3 (2010), 283–296.

[129] TODA, Y., AND KUBOTA, N. Self-localization based on multiresolution map for remote control of multiple mobile robots. *Industrial Informatics, IEEE Transactions on 9*, 3 (2013), 1772–1781.

[130] TOMATIS, N., NOURBAKHSH, I., AND SIEGWART, R. Hybrid simultaneous localization and map building: a natural integration of topological and metric. *Robotics and Autonomous Systems 44*, 1 (2003), 3–14.

[131] TRAWNY, N., AND ROUMELIOTIS, S. I. Indirect kalman filter for 3D attitude estimation. *University of Minnesota, Dept. of Comp. Sci. Eng., Tech. Rep 2* (2005), 2–5.

[132] UMAMAGESWARI, A., IGNATIOUS, J. J., AND VINODHA, R. A comparitive study of kalman filter, extended kalman filter and unscented kalman filter for harmonic analysis of the non-stationary signals. *International Journal of Scientific Engineering Research 3*, 7 (2012), 1–9.

[133] VÁSÁRHELYI, G., VIRÁGH, C., SOMORJAI, G., TARCAI, N., SZÖRÉNYI, T., NEPUSZ, T., AND VICSEK, T. Outdoor flocking and formation flight with autonomous aerial robots. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on* (2014), IEEE, pp. 3866–3873.

[134] WALCOTT-BRYANT, A., KAESS, M., JOHANNSSON, H., AND LEONARD, J. J. Dynamic pose graph SLAM: Long-term mapping in low dynamic environments. In *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on (2012)*, pp. 1871–1878.

[135] WANASINGHE, T., I. MANN, G., AND GOSINE, R. Relative localization approach for combined aerial and ground robotic system. *Journal of Intelligent Robotic Systems 77*, 1 (2015), 113–133.

[136] WEI, T., HAOMIN, L., ZILONG, D., GUOFENG, Z., AND HUJUN, B. Robust monocular SLAM in dynamic environments. In *Mixed and Augmented Reality (ISMAR), IEEE International Symposium on (2013)*, pp. 209–218.

[137] WEISS, S., ACHTELIK, M. W., CHLI, M., AND SIEGWART, R. Versatile distributed pose estimation and sensor self-calibration for an autonomous MAV. In *Robotics and Automation (ICRA), IEEE International Conference on (2012)*, pp. 31–38.

[138] WEISS, S. M. *Vision based navigation for micro helicopters*. PhD Thesis, ETH Zurich, (2012).

[139] WENDEL, A., IRSCHARA, A., AND BISCHOF, H. Natural landmark-based monocular localization for MAVs. In *Robotics and Automation (ICRA), IEEE International Conference on (2011)*, pp. 5792–5799.

[140] WU, E.-Y., LI, G.-Y., XIANG, Z.-Y., AND LIU, J.-L. Stereo vision based SLAM using rao-blackwellised particle filter. *Journal of Zhejiang University SCIENCE A 9*, 4 (2008), 500–509.

[141] YANMAZ, E., YAHYANEJAD, S., RINNER, B., HELLWAGNER, H., AND BETTSTETTER, C. Drone networks: Communications, coordination, and sensing. *Ad Hoc Networks 68* (2018), 1–15.

[142] YOUNG-CHEOL, C., AND HYO-SUNG, A. Formation control of quad-rotors in three dimension based on euclidean distance dynamics matrix. In *Control, Automation and Systems (ICCAS), 11th International Conference on (2011)*, pp. 1168–1173.

[143] YUFENG, L., AND THRUN, S. Results for outdoor-SLAM using sparse extended information filters. In *Robotics and Automation, (2003). Proceedings. ICRA '03. IEEE International Conference on*, vol. 1, pp. 1227–1233 vol.1.

[144] ZETIK, R., JOVANOSKA, S., AND THOMÄ, R. Simple method for localisation of multiple tag-free targets using UWB sensor network. In *Ultra-Wideband (ICUWB), IEEE International Conference on* (2011), IEEE, pp. 268–272.

[145] ZHAN, W., SHOUDONG, H., AND DISSANAYAKE, G. Multi-robot simultaneous localization and mapping using D-SLAM framework. In *Intelligent Sensors, Sensor Networks and Information,(2007). ISSNIP 2007. 3rd International Conference on*, pp. 317–322.

[146] ZHU, C., GIORGI, G., LEE, Y.-H., AND GÜNTHER, C. Enhancing accuracy in visual SLAM by tightly coupling sparse ranging measurements between two rovers. In *Position, Location and Navigation Symposium (PLANS), (2018) IEEE/ION*, IEEE, pp. 440–446.

[147] ZHUANG, Y., WANG, Z., YU, H., WANG, W., AND LAURIA, S. A robust extended filtering approach to multi-robot cooperative localization in dynamic indoor environments. *Control Engineering Practice 21*, 7 (2013), 953–961.

[148] ZIHAJEHZADEH, S., YOON, P. K., KANG, B.-S., AND PARK, E. J. Uwb-aided inertial motion capture for lower body 3D dynamic activity and trajectory tracking. *IEEE Transactions on Instrumentation and Measurement 64*, 12 (2015), 3577–3587.