Communication-based UAV Swarm Missions

Master thesis

Submitted in Fulfilment of the
Requirements for the Academic Degree
M.Sc.

Dept. of Computer Science
Chair of Computer Engineering

Submitted by: Huan Yang
Student ID.: 445764
Date: 03.01.2023

Supervising Tutors: Prof. Dr. Dr. h. c. Wolfram Hardt
M.Sc. Fatih Kilic

Abstract

Unmanned aerial vehicles have developed rapidly in recent years due to technological advances. UAV technology can be applied to a wide range of applications in surveillance, rescue, agriculture and transport. The problems that can exist in these areas can be mitigated by combining clusters of drones with several technologies. For example, when a swarm of drones is under attack, it may not be able to obtain the position feedback provided by the Global Positioning System (GPS). This poses a new challenge for the UAV swarm to fulfill a specific mission. This thesis intends to use as few sensors as possible on the UAVs and to design the smallest possible information transfer between the UAVs to maintain the shape of the UAV formation in flight and to follow a predetermined trajectory. This thesis presents Extended Kalman Filter methods to navigate autonomously in a GPS-denied environment. The UAV formation control and distributed communication methods are also discussed and given in detail.

Keywords: Machine Learning, Autonomous Navigation, Formation Control, Distributed Communication

Table of Content

List of Figures

List of Tables

List of Abbreviations

APOLI	Automated Power Line Inspection
CNN	Convolutional Neural Networks
DR	Dead Reckoning
EKF	Extended Kalman Filter
GPS	Global Positioning System
HTER	Half Total Error Rate
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
MDP	Markov Decision Process
ML	Machine Learning
PSO	Particle Swarm Optimization
RF	Radio Frequency
RL	Reinforcement Learning
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
SLAM	Simultaneous Localization and Mapping
UAVs	Unmanned Aerial Vehicles
UWB	Ultra-wideband

1 Introduction

A swarm of drones is a collection of autonomous or remotely controlled aircraft in which the drones maintain some form of internal structure among themselves. Swarms of drones have essential uses in many fields and missions, such as surveillance [1], transportation [2], search and rescue [3], agriculture, and defiance [4]. UAVs are low-cost, casualty-free, simple to equip, easy to operate, flexible, and reliable. Usually, a single UAV with advanced control strategies can achieve real-time high-precision attitude control and complete trajectory tracking. The success rate and resistance to emergencies of multiple UAVs flying in formation are higher than that of a single UAV. However, the current level of technology still needs to support the autonomous decision-making function of multi-UAV formation in the complete sense of the word. It is almost impossible to achieve a high degree of intelligent clustered mass cooperative formation flight [5]. Therefore, developing UAV trajectory planning, formation control, and positioning technology is of great interest.

One of the most common methods used in leader-follower formation control. The leader's path is predetermined and both the leader and follower maintain a particular configuration while traveling at the same speed and direction. The leader can be seen as the object to be tracked or as the common interest of the entire multi-intelligence. Desai's team at the University of Pennsylvania [6] has done much theoretical and experimental work on this approach.

In terms of information interaction, a distributed control method is used. Each UAV needs information about its neighbouring UAVs (this communication structure will be reflected in the graph structure). Although distributed control has lower accuracy, it requires less information exchange and has simpler implementation due to reduced computational requirements [7]. In terms of trajectory planning, a general waypoint-switching model was used. No further research has been done.

The primary research in this thesis is on positioning techniques. Under the above approach, the aim is to use a method that copes with GPS-denied environments without incurring the system cost of complex sensor fusion and navigation computations. For a group of UAVs with limited intra-UAV communication and computational resources, there are navigation techniques that can compensate for the lack of GPS-based position feedback. One of the simplest ways to deal with the lack of direct position feedback operation is through heading projection techniques. Applying heading projection methods to a swarm of UAVs introduces a new source of error, namely the rapid accumulation of UAV errors due to environmental biases.

Considering the above methods and premises, in a 3D environment, when one UAV in a formation loses its GPS signal, we use its IMU data to predict the position and, through internal communication, the formation. This UAV will improve its positioning accuracy by combining data predicted through GPS shared by other drones.

1.1 Motivation

The positioning function of UAVs is essential for autonomous navigation. However, it can be challenging to achieve accurate positioning when GNSS/GPS/UWB signals are lost, as the information provided by onboard sensors is still being determined. To address this problem, many researchers have proposed various auxiliary localization methods. In recent years, UAV positioning technologies such as visual and anchor-based positioning have made significant progress in both theoretical and practical research, and some of these technologies have even entered the market stage. They have demonstrated good practicality in unique environments. However, there are still many challenges to achieving safe and reliable navigation and positioning tasks for UAVs in outdoor environments, compared to indoor UAVs or mobile robots. There are numerous limitations to UAV positioning in outdoor environments that need to be overcome in order to achieve reliable and safe navigation.

To maintain flight, the UAV must generate sufficient vertical lift, so it has a limited load. Currently, UAVs can be loaded with instruments, including cameras, IMUs, and rangefinders. An onboard controller is required to accomplish the UAV's autonomous positioning task, which also increases the weight of the load. How to reduce the weight of the UAV load and achieve more accurate autonomous positioning has become a vital issue for visual navigation.

UAV vision localization algorithms require processing image sequences captured by vision sensors, and the image computation is extensive. Even a desktop computer with high computational performance requires significant computation time. Currently, the processing performance of airborne devices is far from that of desktop computers. There are high-performance GPU-enabled companion boards available, like Nvidia Jetson Nano. One common approach is to use the ground station to process the visual localization algorithm and transmit the results to the UAV via communication equipment, allowing the UAV to be localized. The outdoor environment is complex, and the performance of the communication equipment will directly affect the real-time performance of the UAV positioning when a large amount of data needs to be transmitted.

The fastest flight speed of a quadrotor UAV can reach 8 meters per second. If the visual positioning algorithm cannot achieve real-time positioning, and the UAV can obtain delayed position information, then the visual positioning function is invalid. Positioning delays and positioning errors not only do not enable the autonomous positioning function of the UAV but also significantly reduce the safety of the UAV. Positioning accuracy is also a vital issue for UAV autonomous positioning algorithms.

In the field of autonomous navigation for UAVs, it is important to explore methods that utilize minimal onboard instruments and are able to operate in GPS-denied environments. One way to improve positioning accuracy is to utilize a small amount of information exchange between UAVs when they are in formation, as this can assist with positioning and enhance the accuracy of the localization information. An effective formation control method and positioning algorithm can be utilized to achieve this. The main objective of this thesis is to improve the positioning accuracy of UAVs in formation through the use of inter-UAV information sharing and an Extended Kalman Filter, while minimizing the need for onboard instrumentation. By developing a method that relies on minimal onboard instruments and is able to operate in a variety of environments, including those where GPS is not available, it will be possible to improve the autonomous navigation capabilities of UAV swarms.

1.2 Thesis Structure

The primary objective of this thesis is to develop a localization method for use in a swarm of UAVs that relies on distributed communication among the drones in the swarm. This method will be incorporated into the Leader-follower formation control algorithm, which is used to guide the movement of the UAVs and maintain a desired formation. The positioning method utilizes data from the IMUs on board each UAV, as well as GPS/UWB data from neighboring Leader drones, to improve the accuracy of the positioning information. By combining these sources of data, the positioning accuracy is increased, which in turn provides better conditions for the automatic navigation of the UAV swarm. The incorporation of this enhanced positioning method into the Leader-follower formation control algorithm will allow for more effective and efficient navigation of the UAV swarm.

Chapter 2 introduces some of the theoretical knowledge, including the basic mechanisms of a quadrotor UAV, sensors, basic positioning methods, communication structures, and formation control. Chapter 3 presents the current state of the art in UAV swarm formation control. In Chapter 3.2, the current mainstream UAV positioning methods are described. Chapter 3.3 presents the research of the Professorship of Computer Engineering at the Faculty of Computer Science of the Chemnitz University of Technology. Chapter 4 describes the MATLAB package and the positioning methods used in this thesis. The results of the algorithm simulations are given in chapter 5. Finally, Chapters 6 and 7 provide a summary and outlook of the algorithms developed in this thesis.

2 Technical Background

The flight control system can be regarded as the brain of the vehicle. The flight, hovering, and attitude change of the multi-axis vehicle are all done by various sensors that transmit the attitude data of the vehicle itself back to the flight controller. The flight control system calculates and makes decisions, sending instructions to the actuator to execute actions and adjust flight attitude. The control system, similar to the CPU of a UAV, is a crucial component responsible for issuing various commands and processing data received from various parts of the UAV. Swarm control technology for UAVs is the technology that enables UAVs to form formations and keep their shape to fly autonomously. A good UAV swarm control technology can help UAV swarms cope with various particular environments to accomplish challenging missions.

In this chapter, each basic technique of the UAV swarm control system will be described in detail. First, the sensor principles most used by UAVs are described. Next, the theory underlying the UAV positioning method used in this thesis is presented. Then the basic communication architecture of existing UAV swarms, i.e., the direction of information transfer, is explained. Finally, the control principles of individual UAVs and UAV swarms are explained. The above technical background provides the theoretical basis for the autonomous positioning and navigation of UAV swarms described in this thesis.

2.1 Quadcopter

A quadrotor helicopter is a helicopter with four rotors [8]. Quadrotor helicopters have four rotors oriented upwards and evenly spaced from the center of mass in a square formation. These rotors, powered by electric motors, can be adjusted to control the quadrotor. Quadcopters, small UAVs known for their simplicity, are commonly used for various purposes including research. However, larger quadcopters capable of carrying a variety of sensors can be costly. For big groups, smaller quadcopters are more attractive. The reduced size of these vehicles has inspired different system design options compared to the typical setup of smaller numbers of larger vehicles.

The basic structure of a quadrotor helicopter is shown in Figure 2-1 [8] and consists of four rotors each generating angular velocities, torques, and forces. The absolute linear position of the quadcopter is defined in the inertial frame in the x, y, and z axes using ξ. Attitude is defined in the inertial frame by three Euler angles η. The pitch angle θ determines the rotation of the quadcopter about the y-axis. The roll angle φ determines the rotation about the x-axis, and the yaw angle ψ determines the rotation about the x-axis. The vector q contains the linear and angular position vectors.

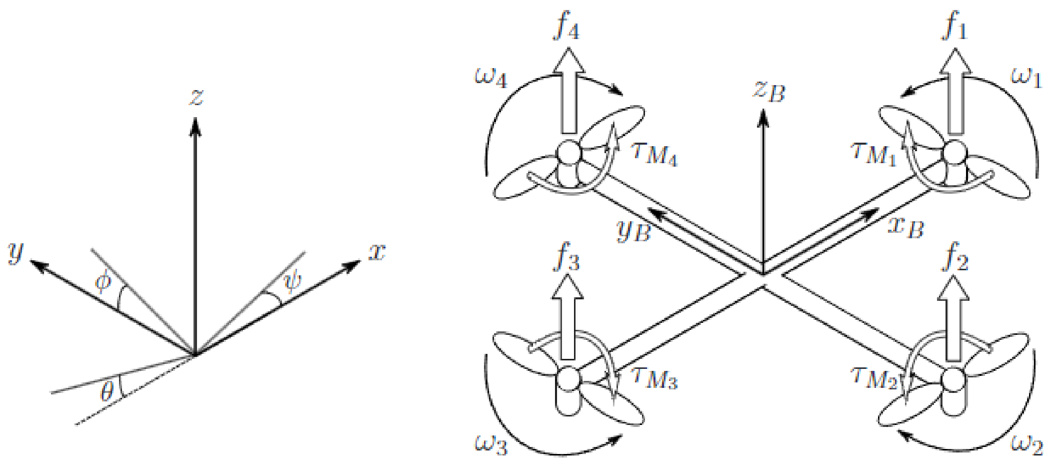

Figure 2-1: The inertial and body frames of a quadcopter [8]

Quadrotor helicopters rely on precise position and attitude measurements from gyroscopes, accelerometers, GPS, sonar, laser sensors, and other measurement devices for control. A PD controller, a simple control technique commonly used for quadcopters, is used to stabilize them. The general form of a PD controller is:

$$e(t) = x_d(t) - x(t) \tag{2.1.1}$$

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau)\, d\tau + K_D \frac{de(t)}{dt} \tag{2.1.2}$$

In this equation, $u(t)$ represents the control input, $\mathrm{e}(t)$ represents the difference between the desired state $x_d(t)$ and the current state $x(t)$, and KP, KI, and KD are the parameters for the proportional, integral, and derivative elements of the PID controller, respectively.

2.2 Sensors

IMU and GPS generate the rawest data for UAV swarm positioning. In this thesis, the simulation model of the UAV toolbox is used to generate the raw data with noise to simulate the natural environment as much as possible. This section briefly introduces the principles of IMU and GPS and the model parameters using the UAV Toolbox.

As this study assumes a swarm of UAVs flying outdoors, it is impossible to set up multiple anchor points and fixed sensors to assist in accurate positioning, as is the case indoors, and only a GPS positioning system was fitted to the Leader UAV. This is due to the significant error of the GPS positioning system for small UAV swarms. If every Follower drone's position data comes from the GPS, keeping the drone information would be difficult and prone to collisions. Therefore, orientation and distance sensors will be installed for each Follower drone. Each drone calculates the relative position of the neighbouring drones by calculating the azimuth and distance to them and the neighbouring drones' global position by using the leader's global position. Finally, the global position of the neighbouring UAV is sent to the neighbouring UAV via internal communication. The neighbouring UAV generates basic UAV control information based on this position parameter. In summary, all the behaviour of the Follower drone is locally referenced to the behaviour of the Leader drone.

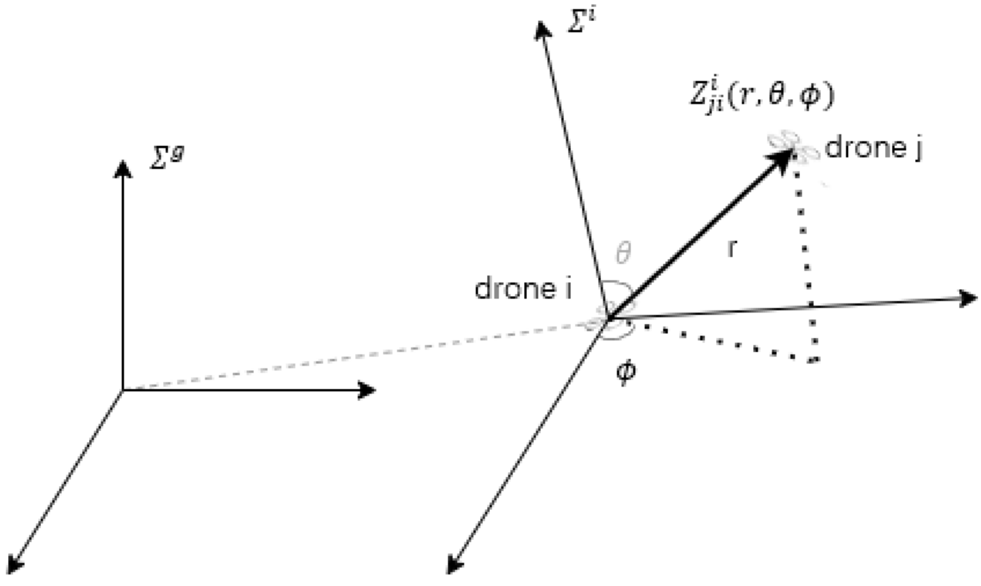

Figure 2-2: Bearing and distance sensors [9]

2.2.1 Bearing and Distance Sensor

As shown in Figure 2-2 [9], each drone is equipped with an orientation and distance sensor [9]. The orientation and distance sensors on drone i measure the distance and angle$Z_{ji}^{i}(r,\theta,\phi)$ of drone j in its local coordinates Σ^{i}.

The local orientation and distance measurement $Z_{ji}^{i}(r,\theta,\phi)$ is essentially a spherical coordinate representation. To simulate the orientation and distance sensors, the global position Z_{ji}^{g} of the UAV j is first calculated from position Z_{j}^{g} and position Z_{i}^{g}.

$$Z_{ji}^{g} = Z_{j}^{g} - Z_{i}^{g} \tag{2.2.1}$$

It is assumed that an onboard attitude estimator, possibly usually an INS sensor, is installed on each UAV. The attitude matrix $\hat{R}_{ig}^{-1}$ is obtained from this sensor, resulting in a local representation of Z_{ji}^{g} in the local coordinate system Σ^{i}.

$$Z_{ji}^{i} = \hat{R}_{ig}^{-1} Z_{ji}^{g} \tag{2.2.2}$$

Then let $Z_{ji}^{i}(x,\ y,\ z)$, the local distance and orientation measurements can be calculated as:

$$r = \sqrt{x^2 + y^2 + z^2} \tag{2.2.3}$$

$$\theta = arccos\frac{z}{\sqrt{x^r + y^2 + z^2}} \tag{2.2.4}$$

$$\phi = arctan\frac{y}{x} \tag{2.2.5}$$

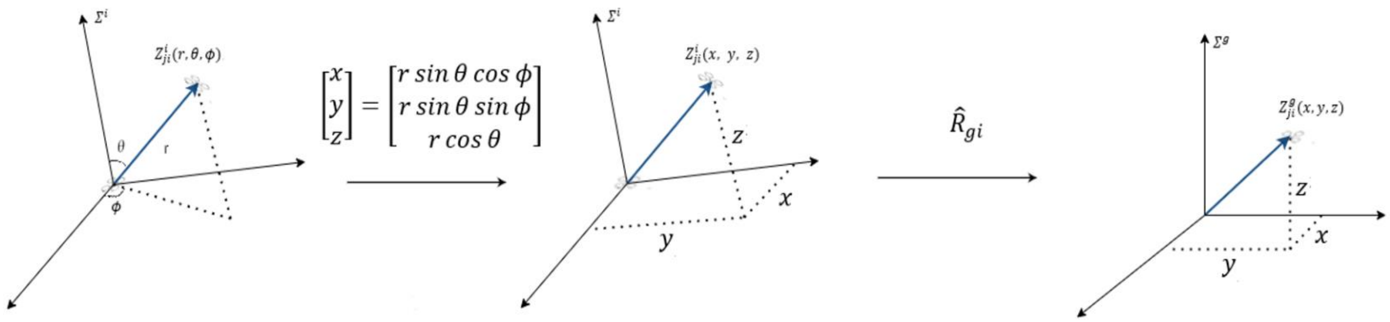

Figure 2-3: Transformation of local bearing and distance measurement in Spherical representationto global Cartesian representation [9]

In order to estimate the global position p_i^g from the local distance and azimuth $Z_{ji}^i(r,\theta,\phi)$ of the neighbouring UAV j, it is necessary to transform the Spherical representation into a global Cartesian representation. The transformation is shown in Figure 2-3 [9]. For the distance and azimuth measurements $Z_{ji}^i(r,\theta,\phi)$ measured in the local coordinate system Σ^i of UAV i, they are first converted to the local Cartesian representation $Z_{ji}^i(x,y,z)$.

$$Z_{ji}^i(x,y,z) = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} r\,sin\,\theta\,cos\,\phi \\ r\,sin\,\theta\,sin\,\phi \\ r\,cos\,\theta \end{bmatrix} \tag{2.2.6}$$

In addition, the local Cartesian representation can be converted to a global Cartesian representation using the pose matrix $\hat{R}_{g_i}$.

$$Z_{ji}^g(x,y,z) = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \hat{R}_{gi} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{2.2.7}$$

The global representation of the relative position $Z_{ji}^g(x,y,z)$ enables the global position of the individual UAVs in the trajectory UAV swarm. The attitude matrix $\hat{R}_{igi}$ plays a vital role in the simulation and preprocessing of the positioning.

2.2.2 IMU Sensor

An inertial measurement unit (IMU) is a device that integrates accelerometers, gyroscopes, and magnetometers to measure and report forces acting on the body, angular rate, and sometimes body orientation. The IMU is used to measure the acceleration and rotational velocity of motion. So, the IMU can measure gravity, just as it can measure acceleration. It is commonly used to manoeuvre modern vehicles, including motorcycles, missiles, aircraft (an attitude and heading reference system), unmanned aerial vehicles (UAVs), etc., and spacecraft, including satellites and landers. IMU allows GPS receivers to operate when GPS signals are unavailable, such as in tunnels, inside buildings, or when electronic interference is present [10].

With the development of microelectronics technology, new inertial sensors, micromechanical gyroscopes, and accelerometers have emerged. MEMS (Micro-Electro-Mechanical System) technology sensors are gradually evolving into the main components of automotive sensors.

Current inertial sensors used in conventional and self-driving cars are usually low to medium, and low-level inertial sensors usually cost only a few dollars. It is characterized by a high update frequency (usually: 1 kHz), which provides real-time position information. However, one major limitation is that its error accumulates over time, so it can only rely on inertial sensors for positioning for a short duration. It is usually used in conjunction with GNSS (Global Navigation Satellite System) in self-driving vehicles.

Inertial sensors are devices that detect physical motion, such as linear displacement or angular rotation, and convert this motion into an electrical signal that is amplified and processed by electronic circuitry. Accelerometers and gyroscopes are the two most common types of MEMS inertial sensors. Accelerometers are sensors that are sensitive to axial acceleration and convert it into an output signal; gyroscopes are sensors that can be sensitive to the angular velocity of motion of a moving body relative to inertial space. Three MEMS accelerometers and three MEMS gyroscopes form a Micro Inertial Measurement Unit (MIMU) that can detect linear acceleration in three directions and angular acceleration in three directions. The inertial microsystem uses three-dimensional heterogeneous integration technology to combine MEMS accelerometers, gyroscopes, pressure sensors, magnetic sensors, signal processing circuits, and other functional parts. The MIMU is an inertial microsystem that uses three-dimensional heterogeneous integration technology to integrate functional parts such as MEMS accelerometers, gyroscopes, pressure sensors, magnetic sensors, and signal processing circuits into a silicon chip with built-in algorithms to achieve chip-level guidance, navigation, and positioning.

The MEMS accelerometer is one of the earliest sensors that began to be studied in the field of MEMS. After years of development, the design and processing technology of MEMS accelerometers has become more and more mature.

It works by the inertia of the movable part in MEMS. Since the middle capacitor plate has a large mass and is a cantilever structure, when the speed change or acceleration reaches large enough, the inertial force on it exceeds the force that fixes or supports it. Movement will cause a change in the distance between the device and the upper and lower capacitor plates, resulting in a change in the upper and lower capacitance. The change in capacitance is proportional to the acceleration. Depending on the measurement range, the strength or elasticity coefficient of the middle capacitor plate cantilever structure can be designed differently. Also, if the acceleration is to be measured in different directions, the MEMS structure will be very different. The capacitance change is converted into a voltage signal by another dedicated chip, sometimes amplified. The voltage signal goes through a digital signal processing process after digitization and is output after zero and sensitivity correction.

The accelerometer also has a self-test function. When powered up, the logic controller sends a command to the self-test circuit. The self-test circuitry generates a DC voltage loaded onto the self-test board of the MEMS chip, and the movable capacitor plate in the middle moves down due to electrostatic attraction. The following process is the same as testing actual acceleration.

Since the 1980s, angular rate-sensitive MEMS gyroscopic angular velocimeters have received increasing attention. According to the performance index, MEMS gyroscopes can be divided into three levels: rate, tactical, and inertial. Rate-grade gyroscopes can be used in consumer electronics, cell phones, digital cameras, game consoles, and wireless mice. Tactical grade gyroscopes are suitable for industrial Control, intelligent cars, trains, steamships, and other fields; inertial grade gyroscopes can be used for navigation, guidance, and Control in satellites and aerospace.

Its working principle is to use the principle of conservation of angular momentum and the Corio effect to measure the angular rate of moving objects. It is primarily a constantly rotating object whose rotational axis points without varying with the rotation of the support that carries it.

Like the accelerometer working principle, the upper active metal of the gyroscope forms a capacitance with the lower metal. When the gyroscope rotates, the distance between him and the capacitor plate below changes, and the upper and lower capacitances change as a result. The capacitance change is proportional to the angular velocity; thus, we can measure the current angular velocity.

Inertial sensors tend to produce erroneous data due to manufacturing processes. One type of error is the offset error, where gyroscopes and accelerometers output non-zero data even when they are not rotating or accelerating. To obtain displacement data, it is necessary to double integrate the output of the accelerometer. After two integrations, even minor offset errors are magnified. As time progresses, displacement errors accumulate, eventually causing us to be no longer able to track the object's position. The second type of error is a proportional error, the ratio between the measured output and the change in the input being detected. Like the offset error, after two integrations, the resulting displacement error accumulates as time advances. The third error is the background white noise, which, if not corrected, can also cause us to be no longer able to track the object's position.

An IMU model is shown in Figure 2-4 [11]. The IMU is made up of individual sensors that provide information about the platform's movement. IMUs combine multiple sensors, including accelerometers, gyroscopes, and magnetometers [12].

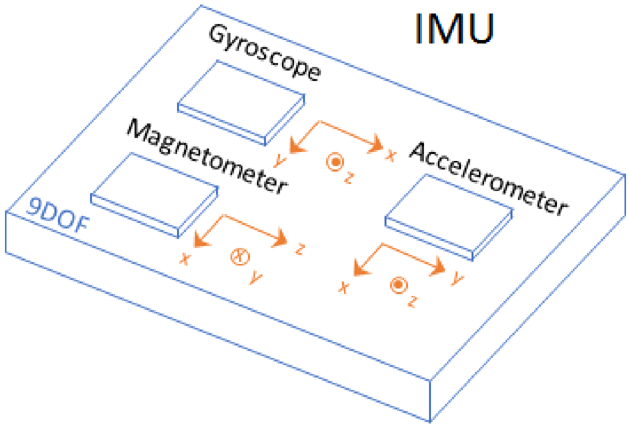

Figure 2-4: IMUs combine multiple sensors, which can include accelerometers, gyroscopes, and magnetometers [11]

In the Matlab toolbox, the measurements returned by the IMU model use the units and coordinate conventions in Table 2-1. Typically, the data returned by the IMU is fused and interpreted as the platform's roll, pitch, and yaw, as shown in Figure 2-5 [11]. Real-world IMU sensors may have different axes for each sensor. The model provided by the Sensor Fusion and Tracking Toolbox assumes that the axes of the individual sensors are aligned.

Output	*Description*	*Units*	*Coordinate System*
Acceleration	Current accelerometer reading	m/s^2	Sensor Body
Angular velocity	Current gyroscope reading	rad/s	Sensor Body
Magnetic field	Current magnetometer reading	μT	Sensor Body

Table 2-1: The units and coordinate conventions [11]

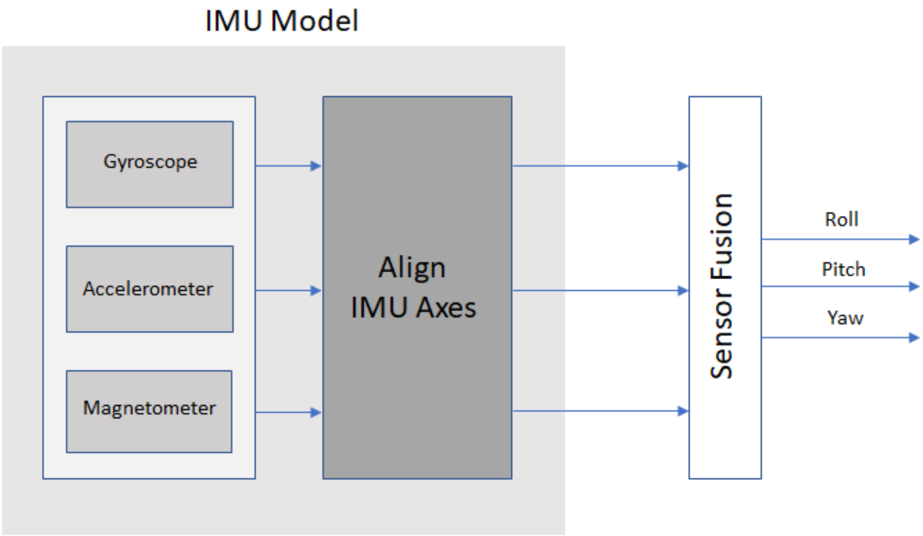

Figure 2-5: The data returned by the IMU is fused and interpreted as the platform's roll, pitch, and yaw [11]

2.2.3 GPS Sensor

The Global Positioning System (GPS) is a satellite-based navigation system developed and operated by the U.S. Department of Defense, specifically the U.S. Pacific Air Force. It is used to provide precise positioning, velocity measurement, and time information to military users around the globe, as well as to most of the Earth's surface (98%). GPS is designed to enable users to determine their 3D position, 3D motion, and time with accuracy and continuity.

GPS implements the concept of "arrival time difference" (time delay) [13]: the arrival time difference from the satellite to the receiver is obtained using the precise position of each GPS satellite and the navigation information generated by the onboard atomic clock sent continuously.

GPS satellites continuously send radio signals with time and position information in the air for reception by GPS receivers. Due to the distance factor of transmission, the moment the receiver receives the signal is delayed from when the satellite sends the signal, which is usually called time delay. Thus, distance can also be calculated based on the time delay. Both the satellite and receiver generate the same pseudo-random code simultaneously. Once the two codes are time-synchronized, the receiver can determine the time delay; multiplying the time delay by the speed of light, the distance is obtained.

Computers and navigation information generators on each GPS satellite are precise about its orbital position and system time. The global network of monitoring stations maintains continuous tracking of the satellite's orbital position and system time. The primary control station at Schriever Air Force Base, Colorado, along with its operations control segment, injects calibration data for each GPS satellite at least once daily. The injected data includes orbital position determination and on-satellite clock corrections for each satellite in the constellation. These correction data are calculated based on complex models that can remain valid for several weeks.

The caesium and rubidium frequency scales of the atomic clocks on each satellite maintain GPS time. These clocks are generally accurate within a few nanoseconds of Universal Coordinated Time (UTC). It is maintained by the Naval Observatory's "master clocks," each stable for several 10-13s. The early GPS satellites used two caesium and two rubidium frequency scales but gradually changed to more rubidium frequency scales. Usually, only one frequency marker works on each satellite at any given time [11].

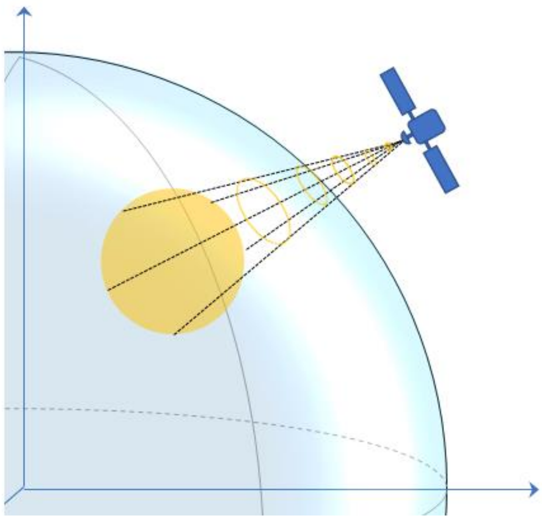

Figure 2-6: Global positioning system [11]

Satellite navigation principle: The distance between the satellite and the user is determined by calculating the difference between the time the satellite's signal was transmitted and the time it is received by the receiver, also known as the pseudo-range. To calculate the user's three-dimensional position and receiver clock deviation, the pseudo-range measurement requires receiving signals from at least four satellites.

GPS is a system of satellites orbiting the Earth that provides 3-D position information to receivers on the Earth's surface, as depicted in Figure 2-7. It is composed of a set of satellites that continuously orbit the Earth. The satellites are maintained in such a

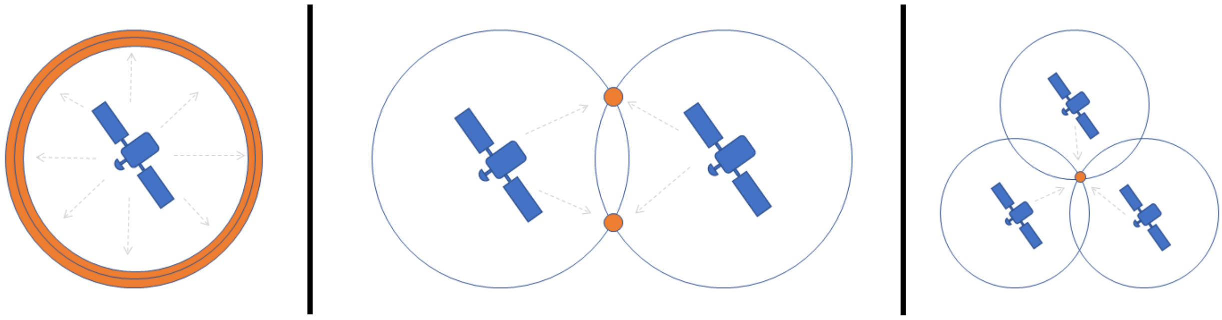

Figure 2-7: Trilateral measurement of the platform's position by measuring the time of flight from the satellite signal to the platform [12]

configuration that the platform is always in the field of view of at least four satellites. The platform's position can be measured trilaterally by measuring the time of flight from the satellite signal to the platform. When received, the satellites timestamp the broadcast signal and compare it to the platform's clock. Three satellites are needed to make a trilateral measurement of a position in three dimensions. A fourth satellite is needed to correct clock synchronization errors between the platform and the satellite. The GPS simulation provided by Matlab Navigation Toolbox models the platform (receiver) data that has been processed and interpreted as altitude, latitude, longitude, velocity, ground speed, and heading.

Output	*Description*	*Units*	*Coordinate System*
LLA	Current global position reading in geodetic coordinates, based on wgs84Ellipsoid Earth model	degrees (latitude), degrees (longitude), meters (altitude)	LLA
Velocity	Current velocity reading from GPS	m/s	local NED
Groundspeed	Current groundspeed reading from GPS	m/s	local NED
Course	Current course reading from GPS	degrees	local NED

Table 2-2: The GPS model returns measurements using the units and coordinate convention [12]

The GPS simulation provided by the MATLAB Navigation Toolbox models platform (receiver) data that has been processed and interpreted as altitude, latitude, longitude, velocity, ground speed, and course as shown in the table Table 2-2.

The GPS model also enables to set off high levels of accuracy and noise parameters, as well as receiver update rates and reference positions.

2.3 Navigation methods

Positioning techniques are essential in UAV formation control. This is because, as described in the previous sections, the real-time position information of a UAV directly affects almost all control parameters of the UAV when tracking its trajectory. Therefore, identifying the exact location of each UAV and minimizing positioning errors and positioning delays is critical for accurately controlling multiple UAVs.

In most missions, as described in chapter 2.2.1, the relative positioning of the UAVs is essential to support safe group control and spatially separated information fusion. Most current UAS primarily use GNSS receivers and RTK (real-time kinematic) technology, offering low cost, high accuracy, and many other advantages. In contrast, GNSS receivers can operate discontinuously or even fail in challenging environments such as interference and rejection due to high-voltage installations and malicious interference, urban canyons, and indoor signal concealment [14].

This thesis uses a combination of GPS, Dead Reckoning and EKF (Extended Kalman Filter). This means that when a UAV in a swarm loses GPS communication, it uses its own IMU data, relative position data, and data projected from the last moments of neighboring UAVs at the time of losing GPS in combination with the EKF to calculate its position. Before describing the method in detail, the following three sections will briefly introduce some current mainstream positioning methods. They are tailored to different scenarios, conditions and uses.

2.3.1 GNSS/INS integrated Navigation System

Low-cost global navigation satellite system (GNSS) receivers are essential for low-cost flight control systems. However, in mega-urbanized cities, multipath effects severely degrade the accuracy of GNSS positioning. The multipath effect cannot be eliminated but mitigated; therefore, combined GNSS/inertial navigation system (INS) navigation is a popular method to reduce this error. Zhang [15] developed an adaptive Kalman filter that adjusts the noise covariance of GNSS measurements based on the positioning accuracy. The adaptive adjustment is based on a supervised machine learning approach that uses an accuracy classification model. The UAV Toolbox includes an interface for this type of localization approach.

A Global Navigation Satellite System (GNSS) is a satellite configuration that provides satellite signals to a GNSS receiver that can be used to calculate velocity, position, and time [16]. An inertial navigation system (INS) utilizes inertial measurement units

(IMUs) which consist of microelectromechanical systems (MEMS) inertial sensors that measure the angular rate and acceleration of the system. These measurements can be fused using advanced Kalman filter techniques to create a GNSS-assisted INS system (GNSS/INS). This combined system has higher accuracy and improved dynamic performance compared to a standalone GNSS or INS system, and can provide estimates for position, velocity, and attitude.

GNSS/INS systems typically include a 3-axis accelerometer, a 3-axis gyroscope, a GNSS receiver, and sometimes a 3-axis magnetometer for estimating the navigation solution. Each of these sensors provides different measurements for the GNSS/INS system.

Both the gyroscope and magnetometer provide the same contribution to the GNSS/INS system as they do to the AHRS. Measuring the angular rate of the gyroscope is integrated into a high update rate attitude solution, while the magnetometer provides a heading reference like a magnetic compass.

The GNSS/INS system accelerometer measures the linear acceleration of the system due to motion and the pseudo-acceleration due to gravity. In order to obtain the linear acceleration due to the system's motion, the pseudo-acceleration due to gravity needs to be subtracted from the accelerometer measurements using an estimate of attitude. The resulting linear acceleration measurements can then be integrated to obtain the system's velocity. It is integrated twice to obtain the position of the system. However, these calculations rely heavily on the INS to maintain an accurate attitude estimate, as any error in attitude will result in an error in the calculated acceleration and, thus, in the integrated position and velocity.

GNSS receivers use navigation messages sent from GNSS satellites and track pseudo-range and Doppler raw observable measurements to provide the receiver's position, velocity, and time (PVT) to the GNSS/INS system. This drift-free PVT solution stabilizes the solution provided by integrating accelerometers and gyroscopes.

Both INS and GNSS track the velocity and position of the system. INS typically reduces errors in the short term but has more significant unbounded errors over extended periods. GNSS generally is noisier in the short term and offers better stability over a more extended period. When the two systems are integrated, GNSS measurements can moderate INS errors and prevent them from growing indefinitely. On the other hand, INS can provide navigation solutions with high output rates, while GNSS navigation solutions are typically updated at rates between 1 Hz and 10 Hz only.

Combining measurements from both systems, INS solutions can bridge the gap between GNSS updates. GNSS/INS systems use a Kalman filter to track the best estimate of the system's position, velocity, and attitude.

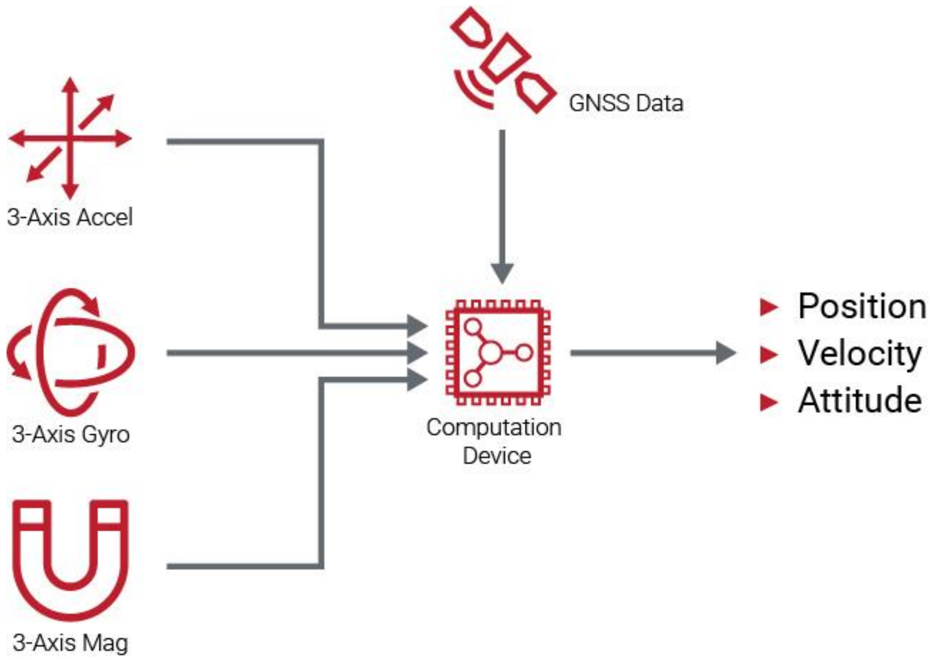

Figure 2-8: GNSS/INS component diagram [16]

Even though combining GNSS and INS can help to overcome many of the limitations of these systems individually, there are still some challenges when using GNSS-assisted INS systems. These challenges include the loss of heading information in static or low dynamic situations, non-Gaussian and non-zero mean GNSS errors, and the potential for GNSS interruptions.

INS fuses inertial sensor data to calculate the platform's position, orientation, and velocity. INS/GPS uses GPS data to calibrate the INS. GPS and INS readings are generally fused using an extended Kalman filter. INS readings are used for the prediction step, and GPS readings for the update step. One use of INS/GPS is for heading projection in the event of unreliable GPS signals.

"INS/GPS" refers to the whole system, including filtering. The INS/GPS simulation provided by Matlab Navigation Toolbox models INS/GPS. Based on live ground motion, it returns the position, velocity, and direction reported by the inertial sensors and GPS receivers as shown in Table 2-3.

Output	Description	Units	Coordinate System
Position	Current position reading from the INS/GPS	meters	local NED
Velocity	Current velocity reading from the INS/GPS	m/s	local NED
Orientation	Current orientation reading from the INS/GPS	quaternion or rotation matrix	N/A

Table 2-3: Measurements returned from the INS/GPS [12]

2.3.2 Ground Control System based Control System

Drones can acquire various information while measuring position and altitude but are relatively limited in indoor environments. GNSS can only work outdoors; a good laser scanner is large and heavy for flying a small indoor drone. In contrast, large UAVs equipped with laser scanners are difficult to control in an indoor environment and may pose a risk. While lightweight ultrasonic sensors can measure distance to some extent, the measurements are not precise. To address this issue, research on computer vision has been applied to UAV technology. Jin [17] proposed a method for estimating indoor location through computer vision techniques.

Computer vision can be utilized for position control of UAVs. Camera-based distance measurement typically uses two cameras, similar to the human eye. The two human eyes observe an object from different positions, creating a difference in the vantage point angles known as parallax, which can be used to estimate the distance between the observed object and the "eye." Stereo vision utilizes this principle.

After measuring the distance, the relative position of the drone can be determined by the camera without the use of GPS or laser sensors. The distance to the origin must be determined in a 3D virtual environment to determine the drone's relative position. Structure from Motion (SfM), which utilizes parallax, a feature of stereo vision with multiple images, can be used to create a 3D reconstruction of the indoor space from 2D images. Multiple images of SfM can be easily acquired using a drone.

In 3D space, since the position is a relative concept, the distance, and direction from the origin in the space where the origin is located in the position. In other words, to locate the drone, the environment map (virtual space) must be constructed first. The origin of the environment map is set to the last position or the average of the camera positions when the spatial configuration image is acquired. 3D space can be constructed from sequential images formed by moving along walls. As shown in Figure 2-9, the spatial configuration is reconstructed by the SfM program, and 360° spatial images are acquired by yaw rotation and circular trajectory at a particular time after the UAV takes off. The amount of yaw rotation of the UAV at the last image acquisition is the forward vector.

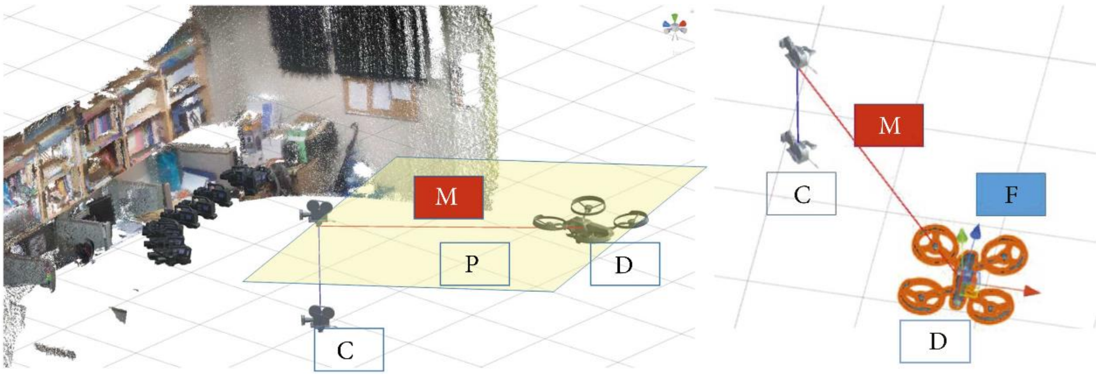

Figure 2-9: Location estimation through virtual camera analysis [17]

The position of the UAV is estimated in two ways. First, the camera's position can be determined by analyzing the sequence images added to the existing sequence. As a result, the camera's position becomes the position of the UAV. In this case, the added sequence should be where the corresponding point can be found in the existing sequence. Another approach is to compare the image captured by the virtual camera in 3D space with the image captured by the current UAV camera, as shown in Figure 2-9 [17], as the UAV moves toward the target point C, approaching in a horizontal, vertical flight. This improves the image comparison accuracy by reducing the image variation instead of flying straight C.

Let D be the current position of the UAV, C be the virtual camera's position, and P be the plane parallel to the UAV. Then the vector M is the orthogonal projection vector of the vector DC to the plane P. At this point, the angle between the forward vector F of the UAV and the vector M can be obtained by:

$$\theta = cos^{-1}\left(\frac{\vec{F}\overrightarrow{M}}{\left\|\vec{F}\overrightarrow{M}\right\|}\right) \tag{2.3.1}$$

is obtained. The image similarity can be estimated if the virtual camera looks at the M vector.

Image similarity can be achieved by template matching, a method to search and check whether a given small template image is present in a large image. The disadvantage of template matching is that it can only show good results at the same scale and orientation. However, since the orientation of the UAV is the same as the virtual camera's, it can produce good results. The similarity of images can be calculated by:

$$R(x,y) = \sum_{x'y'} \left(T(x',y') - I(x+x',y+y')\right)^2 \tag{2.3.2}$$

If the field of view of the virtual camera is the same as the field of view of the UAV camera, the error can be reduced by template matching for position estimation. Again, an image pyramid is constructed for the template images to overcome the error due to image scale. When the UAV reaches the target point, a new sequence of images is formed, and the procedure is repeated to finalize the position. If the current sequence cannot be matched to an existing sequence, it can be estimated. The image captured by the UAV camera is then searched for in the image projected onto the virtual camera, which is oriented in the same direction as the yaw rotation of the current UAV. At this point, the image from the UAV camera becomes a template and the virtual camera is moved in the X-Y plane to place the template area at the center of the virtual camera image and point: 1 of the images scale the z-axis determines one as the position of the UAV.

Ultimately, this approach demonstrates that it is possible to track the position of the UAV in an indoor environment using only a single-view camera.

2.3.3 The Cooperative Positioning Method

The suitable positioning method is a positioning method that has similar positioning accuracy as GNSS and is low cost, low power, small size, and suitable for an extensive range [18]. Its basic principle is a rigid structure.

In the satellite-/ground-based radiolocation system, the receiver calculates the local position and time by ranging or vectoring with multiple anchor points without direct observation between each other, so the relative position is calculated indirectly, i.e., when a point is lost or does not have enough anchor observation parameters, it will lose the relative position at the same time.

Although we can easily find that the point can be located if it can be ranged with other points that have already been located by anchor points. It can also be located because these points can form a specific spatial structure that determines the unknown position.

We define this specific positional relationship as a rigid structure. In effect, if all points could form a rigid structure, their relative positions could be determined directly without using anchor points and a position-by-position process. Although relative coordinates do not have a defined origin, yaw, pitch, and roll, the determined relative position results are sufficient for many control and fusion applications.

The process of forming a rigid structure for a geometric ranging network of more than 4 points, for example, can form a specific spatial structure if each point has more than four observable neighbours not in a plane, according to the spatial degrees of freedom. This assumption has yet to be fully confirmed. However, it is valid every time, meaning that the relative positions of all network points can be calculated using fewer but critical observations.

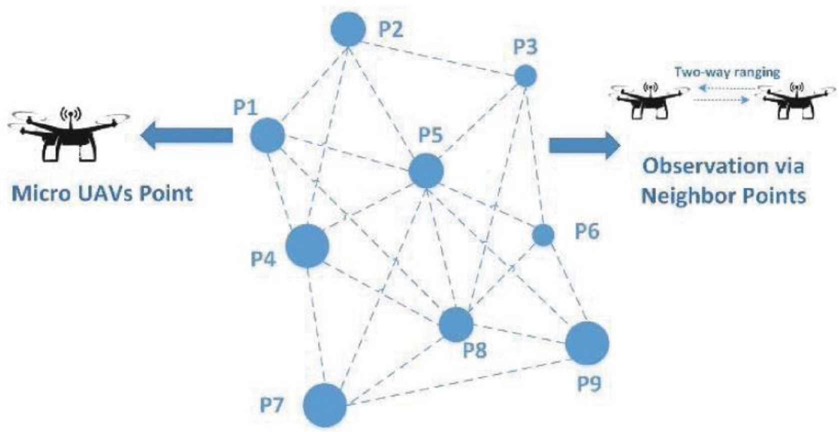

Figure 2-10: UAVs ranging and communication to form a rigid graph [14]

Thus, based on the principle of rigid structure, a micro-UAV ranging communication network and its co-localization method were designed [14]. Its structure is shown in Figure 2-10 [14]. All micro-UAV points in the flight network propagate and receive dedicated signals with high-precision geometric ranges and one-to-many communication capabilities. The time-division chip of each point of the signal is uniformly distributed and reduced to sub-millisecond levels to reduce observation errors caused by UAV movements so that high-quality inertial sensors are no longer needed.

The calculation process has three main parts: data preprocessing, relative position matrix calculation, and position coordinate fitting. This method is like that described in Chapter 2.2.1.

In the data processing part, the range data will be shared and selected to participate in the relative position matrix calculation, including data collection, differentiation/testing, and stiffness checking.

Since the actual range of the points determines the stiffness, only geometric data must be filtered out; otherwise, invalid data may lead to incalculable errors. Data filtering includes both differentiation and feedback testing methods.

- NLOS (Non-Line of Sight) differentiation is a method that uses data time-frequency and space-domain statistical characteristics to distinguish whether a signal is geometrically transmitted or not. A differentiation threshold is required for NLOS and penetration links, which contain complex multipath guidance noise in the tracking loop, which a pre-ranging test can determine.
- Feedback testing is a way to use the localization results to adjust the differentiation threshold and distinguish non-geometric data, which is necessary because uncooperative environments can lead to different statistical features. In contrast, homogeneous medium penetration links sometimes lead to only subtle changes in the specified features.

Each point calculates a local rigidity map of its neighbouring points in the relative position matrix calculation section. Usually, multidimensional scaling can be used for all geometric channel networks. At the same time, observational data in challenging environments are usually insufficient to form a complete one-to-one range matrix, so an iterative approximation method for calculating the position matrix by range is required. The computational procedure is very similar to PSO (Particle Swarm Optimization).

- First, an initial estimated relative position vector of all points of the sub-network is given as the iterative initial value; then, several adjacent position vectors are set, and an estimated range matrix is computed from each vector.
- Then calculate the Euclidean distance of all observable distance errors through the ranging matrix and each estimated ranging matrix and select the estimated position vector with the minor error as the initial value of the iteration.

- The above two processes are repeated until the estimation error converges to be small enough or reaches the iteration's upper limit. Finally, we can obtain the estimated relative position vector in the rigid network.

In the position coordinate fitting part, the relative position results of the sub-networks will be fitted to the exact or absolute coordinates, and interconnections and anchor points will determine the relationship of each point in the whole network. All methods include only rotation and translation of the matrix, with no scaling. The relative position structure and absolute coordinates in the subnetwork may be slightly different for interconnected and anchor points, defining a weight factor for coordinate adhesion and representing the reliability and modifiability of the point or vector.

- Calculating the rotation matrix: The target rotation matrix is constructed by eccentrically collecting and eccentric acting the positions of interconnections and anchor points or their vectors in one relative or absolute coordinate; then, the master matrix is constructed by the relative position results of another subnet and the associated points or vectors of their weights; after that, we can calculate the rotation matrix by the target matrix and the master matrix.
- Calculate the translation matrix: In the above process, we can use the main matrix to rotate the matrix several times to achieve the dispersion matrix; construct the final matrix by the positions of interconnections and anchor points; then subtract the decenter matrix to get the translation matrix.
- Calculate the relative/absolute position vector of the whole network: use the rotation and translation matrices in each pair of subnets to fit one coordinate to another, to stitch the relative position results of two subnets into one; stitch the results of all subnets one by one to reach the relative/absolute position vector of the whole network.

All the above localization processes can be computed at each point or several key points because it is based on a non-differential ranging calculation method instead of anchor localization: even in the extreme case of no anchor and no external collaborative signal. It is still possible to reach the relative results of a cluster continuously.

2.3.4 Ultra-wideband Navigation System

Ultra-wideband (UWB) navigation is a technology that uses UWB signals for positioning and location tracking [19]. UWB is a type of wireless communication that uses a very wide frequency band, typically ranging from 3.1 to 10.6 GHz. This wide frequency range allows UWB signals to transmit a large amount of data over short distances with high accuracy and low power consumption. In navigation applications, UWB can be used to determine the location of a device with high precision, making it useful for applications such as indoor positioning and tracking, location-based services, and asset tracking.

UWB technology has several advantages over other types of wireless technologies that make it well-suited for navigation and positioning applications. One key advantage is its high accuracy. UWB signals can be precisely timed, which allows them to be used to determine the distance between two devices with high accuracy. This makes UWB ideal for applications that require precise positioning, such as autonomous vehicles and robotics.

Another advantage of UWB is its low power consumption. Because UWB signals are transmitted over a very wide frequency band, they can be transmitted at low power levels while still providing good range and accuracy. This makes UWB ideal for battery-powered devices that need to operate for long periods of time without needing to be recharged.

UWB also has a number of other attractive features for navigation and positioning applications, including low interference with other wireless systems, high speed data transmission, and the ability to operate in difficult environments such as urban canyons and dense foliage.

One way that UWB can be used for UAV swarm navigation is by providing a high-precision, low-latency communication and positioning system for the UAVs. UWB signals can be used to accurately determine the distance between UAVs in the swarm, which allows the UAVs to maintain a precise formation and coordinate their actions. This is particularly useful for tasks such as search and rescue, where the UAVs need to cover a large area and work together to locate and assist individuals in need.

UWB signals have a very wide bandwidth, which allows them to carry a large amount of data and to be transmitted over long distances with low power consumption. This makes UWB well-suited for positioning applications, as it allows for the creation of high-resolution maps and the ability to accurately determine the position of objects. In order to use UWB for indoor positioning of UAVs, a network of UWB receivers is typically deployed. These receivers can be used to detect the position of UWB-enabled devices, such as UAVs, by measuring the time it takes for the UWB signal to travel from the device to the receiver. By using multiple receivers, it is possible to triangulate the position of the UAV with high accuracy.

One of the advantages of using UWB for indoor positioning of UAVs is that UWB signals can penetrate through walls and other obstacles, making it possible to accurately track the position of the UAV even when it is inside a building or other structure. In addition, UWB technology is low power, which makes it suitable for use in battery-powered UAVs.

2.4 Communication Architecture

UAVs in formation often must keep their relative position in the formation substantially constant due to mission requirements. The general holding strategy is that each UAV in the formation maintains the same relative position to the agreed point in the formation. This holding strategy is called follow-and-hold when this agreed point is the pilot aircraft. In formation keeping, some disturbances may be caused by some disturbing factors. The conflict prevention strategy is to avoid collisions and blockages in information interaction that may occur under the disturbance. To maintain a specific formation shape, the UAV group must have information interaction between them. Control strategies for information interaction are generally centralized, distributed, and decentralized Control [20], each with its unique definition and advantages.

2.4.1 Centralized

In a centralized scheme, a core processing unit is introduced. It can be a base station on the ground or an agent with high computing power in information. As shown in Figure 2-11 [20], the core unit monitors the coordination of the team to accomplish global tasks based on the information gathered from all remaining agents. All agents

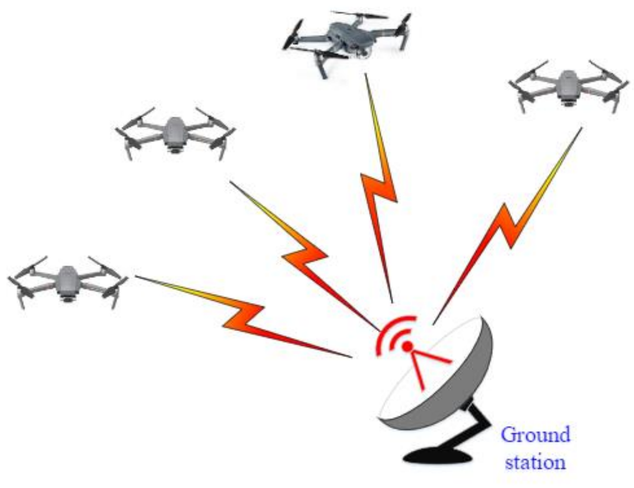

Figure 2-11: The centralized control scheme for multiple UAVs [20]

must remain in contact with the core unit. The centralized solution introduces some disadvantages, such as poor robustness and wasted energy. Due to the core unit's critical function in monitoring the team's global tasks, a failure of the core unit can bring down the entire formation. The computational power of each agent is not utilized, and the connection links required between the core unit and other members burden the communication resources.

2.4.2 Decentralized

In the decentralized scheme, each UAV maintains its close relationship to the agreed points in the queue and does not interact with other UAVs. It has the least effective Control, essentially no interaction of information and the least amount of computation, but the simplest structure.

2.4.3 Distributed

In a distributed scheme, the organization does not need a core unit to be organized. As shown in Figure 2-12 [20], agents in the formation can communicate and share information with other members. The processing unit is available on the agent itself, and decisions are made by the agent based on local observations. Each UAV must interact with information about its position, velocity, attitude, and motion target with the UAVs adjacent to it in the queue. In a distributed control strategy, each UAV needs to know the information of the UAVs adjacent to it. Although the Control is relatively ineffective, there is less information interaction, which significantly reduces the amount

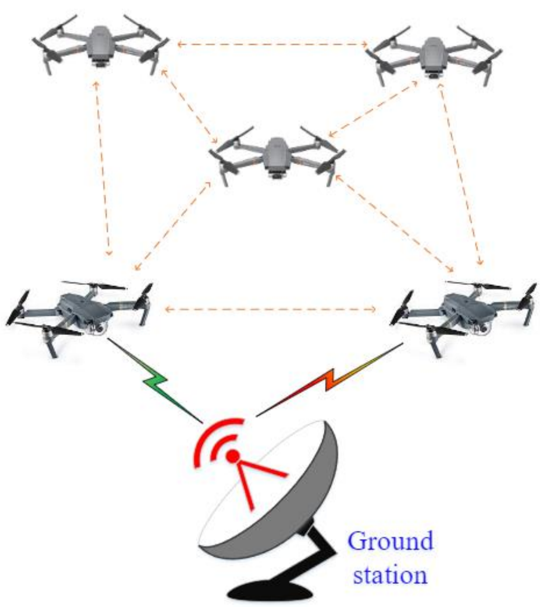

Figure 2-12: The distributed control scheme for multiple UAVs [20]

of computation, the system is relatively simple to implement, and the bottlenecks in computation and communication of the centralized approach are overcome.

In the simulations described in this thesis, a distributed method of communication is used. Although less effective than centralized Control, distributed Control is a simple and reliable control structure with a small amount of information, which makes it easier to avoid information conflicts. From an engineering point of view, such a structure is easy to implement and maintain. In addition, the distributed control strategy is highly adaptable. It has good expandability and fault tolerance, such as when a new UAV is required to join the formation due to a sudden change of mission on the way to a mission. Alternatively, when a UAV cannot continue the mission due to a failure and needs to be removed from the formation and replenished with a new UAV. Since distributed Control can limit the impact of a burst to a localized area, the current research on formation control strategies has shifted from centralized to distributed Control.

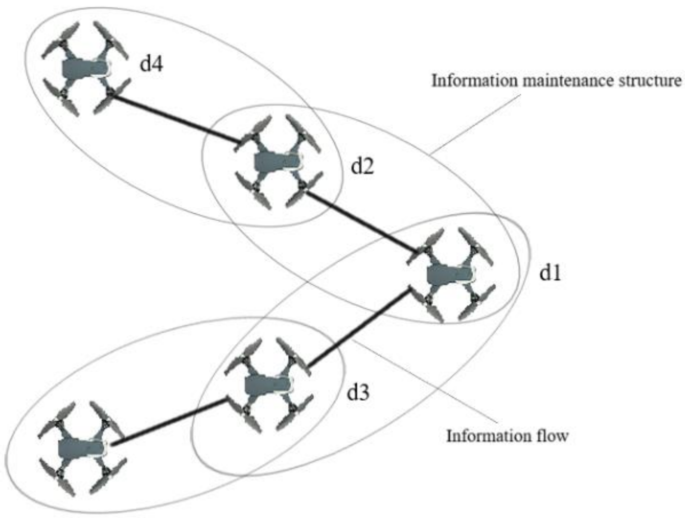

Figure 2-13: The distributed control: based on [20]

The formation profile for distributed Control in this thesis is shown in Figure 2-13 [20]. In this figure, d1 is the first machine, and d2 and d3 follow d1 and maintain their relative positions to d1 to maintain their positions in the queue. d4 can maintain its position in the whole queue if it knows the information of d2 and maintains its relative position to it. The whole queue is made up of several basic two UAVs in one subsequent formation, which is very scalable.

It is assumed that the formation consists of 30 UAVs, moves inwards from different initial position locations, keeps searching for neighbouring UAVs, and needs to keep a certain distance from anyone. After a period, the formation state is completed. Suppose a centralized control strategy is used to complete the formation. In that case, the information interaction is massive, as the complexity of processing this information is geometrically related to the number of UAVs in the formation. If a decentralized control strategy is used, there is no guarantee that no collisions will occur between the UAVs during formation. Only a distributed control strategy can solve the information interaction and collision problems [20].

2.5 Formation Control

In this section, graph theory was first introduced. The flight PID control of quadrotor UAVs will then be presented according to Atheer [21], which is already a relatively mature technique applied as the base control system for each UAV in the simulated systems described in this thesis. Although this thesis focuses not on the control system of each UAV specifically, understanding the latest PID control is very important in understanding the control parameters of the UAV. This is followed by a description of the control method for the entire formation, namely the Leader-follower control method. A description of the structural setup of the formation and a simple reconfiguration of the formation with GPS available follows this. Finally, the method of trajectory optimization used in the simulation is described.

2.5.1 Graph Theory

The content of this subsection is summarized by Jia [9] and Bondy [22]. Its methods and ideas will be applied to the initial setup of the UAV formation. The graph theory allows to store information about the formation and the direction of communication between UAVs in a simple and efficient way.

Undirected graph An undirected graph is a tuple $G = (V, E)$, where V is a list of nodes and E is a set of unordered nodes all over. An edge of unordered nodes is the set of two nodes $\{i, j\}$. $\forall i, j \in V$ and $i \neq j$. If $\{i, j\} \in E$, then i, j are called neighbours. N_i denotes the set of neighbours of i.

Directed graph A directed graph is a tuple $G = (V, and\ E)$. V is a node list. E is a set of edges of ordered pairs of nodes. An ordered pair (i, j) denotes an edge from i to j. i is called the in-neighbour of j, and j is called the out-neighbour of i. Let N_i^{Out} denote the set of out-neighbour of i.

Path A path of an undirected graph is an ordered sequence of nodes such that any pair of consecutive nodes is an edge of the graph.

Directed path A directed path of a digraph is an ordered sequence of nodes such that any pair of consecutive nodes is an edge of the digraph.

Connectivity An undirected graph is connected if, between any pair of nodes, there exists a path. A digraph is strongly connected if a directed path exists between any pair of nodes.

Weighted digraph A weighted digraph is a triplet $G = (V, E, \{a_e\}_{e \in E})$, where (V, E) is a digraph and a_e is a strictly positive scalar of an edge $e \in E$. A weighted digraph is undirected if, for any edge $(i, j) \in E$, there exists an edge $(i, j) \in E$ and $a_{(i,j)} = a_{(i,j)}$.

Adjacency matrix The adjacency matrix is a square matrix used to represent a finite graph. Each element of the matrix represents whether an edge connects the points. In a particular case, the adjacency matrix of a simple graph is a (0,1) matrix, and the diagonal elements are all 0. The adjacency matrix of an undirected graph is symmetric. Spectral graph theory investigates the relationship between the eigenvectors and eigenvalues of a graph and its adjacency matrix. The association matrix of a graph needs to be distinguished from the adjacency matrix. It is another matrix representation of the graph whose elements indicate whether the individual node-edge pairs are related.

Given a weighted diagraph $G = (V, E, \{a_e\}_{e \in E})$, the adjacency matrix A is defined as follows:

$$a_{ij} = \begin{cases} a_{(i,j)}, & (i,j) \in E \\ 0, & otherwise \end{cases} \tag{2.5.1}$$

The weighted out-degree matrix D_{out} is defined by:

$$D_{out} = diag(A1) \tag{2.5.2}$$

2.5.2 PID Control System

A quadrotor robot is a small flying vehicle with four propellers around the main body. The main body includes the power source and the control hardware. The four rotors are used to control the vehicle. The rotation speed of the four rotors is independent. Due to this independence, it is possible to control the vehicle's pitch, roll, and yaw attitudes. Its displacement is then generated by the total thrust of the four rotors, the direction of which varies according to the attitude of the four rotors. The motion of the vehicle can therefore be controlled [21].

Quadrotor flying robots have four input forces and six output coordinates, allowing them to carry larger payloads compared to conventional helicopters. They are able to change direction by adjusting the speed of individual rotors, eliminating the need for cyclic and collective pitch control, resulting in a simpler mechanical design and lower production cost. However, there are some disadvantages. The space requirements and power consumption of four rotors are the main drawbacks of this design. A cross-frame configuration with four rotors consumes more space than a conventional UAV with only one rotor. The high-power consumption is due to using four motors as actuators for the flying robot. Although minimal cross-coupling simplifies the quadrotor vehicle, the dynamics of the quadrotor, especially its low-rate damping, can make it difficult to control. For a low-cost vehicle, controlling it can be even more difficult.

A quadrotor is an underdriven aircraft with a fixed pitch angle of four rotors, as shown in Figure 2-1. Modelling a quadrotor aircraft is difficult due to its complex structure. The aim is to develop a vehicle model as realistically as possible.

In a quadrotor, four fixed-angle rotors represent four input forces, essentially the thrust gained by each propeller, as shown in Figure 2-1. The input u_1 is the sum of the thrust forces of each motorPitch motion is achieved by increasing or decreasing the speed of the rear motor and decreasing or increasing the speed of the front motor, respectively. Similarly, rolling motion is obtained by adjusting the speed of the corresponding motor and decreasing or increasing the speed of the left motor. Yaw motion is produced by decreasing or increasing the speed of the front and rear motors and decreasing or increasing the speed of the lateral motors. This should be done while keeping the total thrust constant. Each controller input affects some aspect of the quadrotor model; here, u_2 affects the roll angle rotation, while u_3 affects the pitch angle, u_4 controls the yaw angle during flight, and u_1 affects the altitude (z-axis) of this model.

Each rotor generates moments as well as vertical forces. Experimentally, these moments are linearly related to the forces at low speeds: four inputs and six outputs $(x, y, z, \theta, \psi, \varphi)$. The quadrotor is, therefore, an underdriven system. As in Figure 2-1, two rotors rotate clockwise, while the other two rotate counterclockwise to balance the moments and produce yaw movement as required.

Compensation of this torque at the centre of gravity is established thanks to rotors 1 to 3 and 2 to 4, which rotate counterclockwise. Rotors 2 and 4 rotate counterclockwise, while rotors 1 and 3 rotate clockwise. To move the quadrotor model from a point to a fixed point in space, the mathematical design should depend on the directional cosine matrix, as in the equation:

$$R_{zxy} = \begin{bmatrix} C_\varphi C_\theta & C_\varphi S_\theta S_\psi - S_\varphi C_\psi & C_\varphi S_\theta C_\psi - S_\varphi S_\psi \\ C_\varphi S_\theta & S_\varphi S_\theta S_\psi - S_\varphi C_\psi & S_\varphi S_\theta C_\psi - C_\varphi S_\psi \\ -S_\theta & C_\theta S_\psi & C_\theta C_\psi \end{bmatrix} \quad (2.5.3)$$

where $C_\theta = Cos(\theta)$, $S_\psi = Cos(\psi)$, and R is the pose matrix.

The LaGrange method can obtain the dynamic model of a quadrotor helicopter. The motion can be written in terms of force and moment balance:

$$\left.\begin{aligned} \ddot{x} &= u_l(Cos\varphi Sin\theta Cos\psi + Sin\varphi Sin\psi) - \frac{K_2\dot{x}}{m} \\ \ddot{y} &= u_l(Sin\varphi Sin\psi Cos\psi + Cos\varphi Sin\psi) - \frac{K_2\dot{y}}{m} \\ \ddot{z} &= u_l(Cos\varphi Cos\psi) - g - \frac{K_3\dot{z}}{m} \end{aligned}\right\} \quad (2.5.4)$$

The K_i given above is the drag coefficient. In the following, the drag is assumed to be zero, as drag is negligible at low speeds. As the centre of gravity is moved up (or down) by d units, the angular acceleration reduces less sensitivity to force, and therefore the stability increases. Stability can be improved by shifting the rotor force towards the centre. This will reduce the roll and pitch moments and the total vertical thrust. For convenience, we define the input as shown in Equation:

$$\left.\begin{aligned} U_1 &= \frac{Th_1 + Th_2 + Th_3 + Th_4}{m} \\ U_2 &= \frac{l(-Th_1 - Th_2 + Th_3 + Th_4)}{I_1} \\ U_3 &= \frac{l(-Th_1 + Th_2 + Th_3 - Th_4)}{I_2} \\ U_4 &= \frac{C(Th_1 + Th_2 + Th_3 + Th_4)}{I_3} \end{aligned}\right\} \quad (2.5.5)$$

In this equation, Th_i represents the thrust forces produced by the four rotors and can be considered as the actual control inputs to the system, C is the coefficient of proportionality of the forces to the moments, and I_i is the moment of inertia around the shaft. The Euler angle equation is as follows:

$$\left.\begin{aligned} \ddot{\theta} &= u_2 - \frac{lK_4\dot{\theta}}{I_1} \\ \ddot{\psi} &= u_3 - \frac{lK_5\dot{\psi}}{I_2} \\ \ddot{\varphi} &= u_1 - \frac{K_6\dot{\varphi}}{I_3} \end{aligned}\right\} \tag{2.5.6}$$

Where (x, y, z) are the three positions. (θ, ψ, φ) three Euler angles represent pitch, roll, and yaw, respectively. g is the acceleration of gravity. I is the half-length of the helicopter. m is the total mass of the helicopter. I_i is the moment of inertia relative to the axis, and K_i is the drag coefficient.

This quadrotor model has six outputs $(x, y, z, \theta, \psi, \varphi)$ and only four independent inputs, so the quadrotor is an under-stimulated system. We are not able to control all states simultaneously. The possible combinations of the control outputs can be (x, y, z). To track the desired position, move to a random heading and stabilize the other two angles. A suitable controller should be able to achieve the desired position and yaw angle while keeping the roll and pitch angles constant.

2.5.3 Control Modelling

In this thesis, the quadrotor responds quickly to the PID controller. Using this method as a recursive algorithm for synthesizing control rules, all calculation stages concerning tracking errors are simplified.

Another aspect of controller selection depends on the control method of the UAV. It can be pattern-based or non-pattern-based. Each state requires a separate controller for model-based controllers, while higher-level controllers determine how these controllers interact. In contrast, a non-mode-based controller uses a single controller to control all states simultaneously.

However, the control strategy employed is summarized as two subsystems of Control; the first is related to position control, while the second is attitude control.

The quadrotor model can be divided into two subsystems. One fully dynamic subsystem includes the dynamics of the vertical position z and yaw angle (Z and ψ). Gyroscopic effects can be ignored to make it possible to design multiple PID controllers for this system, thus eliminating any cross-coupling between parameters.

$$\begin{bmatrix}\ddot{z}\\ \ddot{\varphi}\end{bmatrix} = \begin{bmatrix}u_1 Cos\varphi Cos\psi - g\\ u_4\end{bmatrix} + \begin{bmatrix}\frac{-K_3\dot{z}}{m}\\ \frac{-K_4\dot{\varphi}}{I_3}\end{bmatrix} \qquad (2.5.7)$$

An underdriven subsystem represents the underdriven subsystem, which gives the dynamics of the horizontal position (x, y) about the pitch and roll angles as shown in Equations:

$$\begin{bmatrix}\ddot{x}\\ \ddot{y}\end{bmatrix} = \begin{bmatrix}u_1 Cos\varphi & u_1 Sin\varphi\\ u_1 Sin\varphi & u_1 Cos\varphi\end{bmatrix}\begin{bmatrix}Sin\theta Cos\psi\\ Sin\psi\end{bmatrix} + \begin{bmatrix}\frac{-K_1\dot{x}}{m}\\ \frac{-K_2\dot{y}}{m}\end{bmatrix} \qquad (2.5.8)$$

and

$$\begin{bmatrix}\ddot{x}\\ \ddot{y}\end{bmatrix} = \begin{bmatrix}u_1 Cos\varphi & u_1 Sin\varphi\\ u_1 Sin\varphi & u_1 Cos\varphi\end{bmatrix}\begin{bmatrix}Sin\theta Cos\psi\\ Sin\psi\end{bmatrix} + \begin{bmatrix}\frac{-K_1\dot{x}}{m}\\ \frac{-K_2\dot{y}}{m}\end{bmatrix} \qquad (2.5.9)$$

PID control applies to the above equations with inputs U_1, U_2, U_3, $and\ U_4$ and outputs ϕ, θ, ψ, and Z_d. We can construct a rate-limited PID controller for the moving subsystem that moves the states $(z, \phi, \psi,\ and\ Z_d\ )$to their desired values. The control

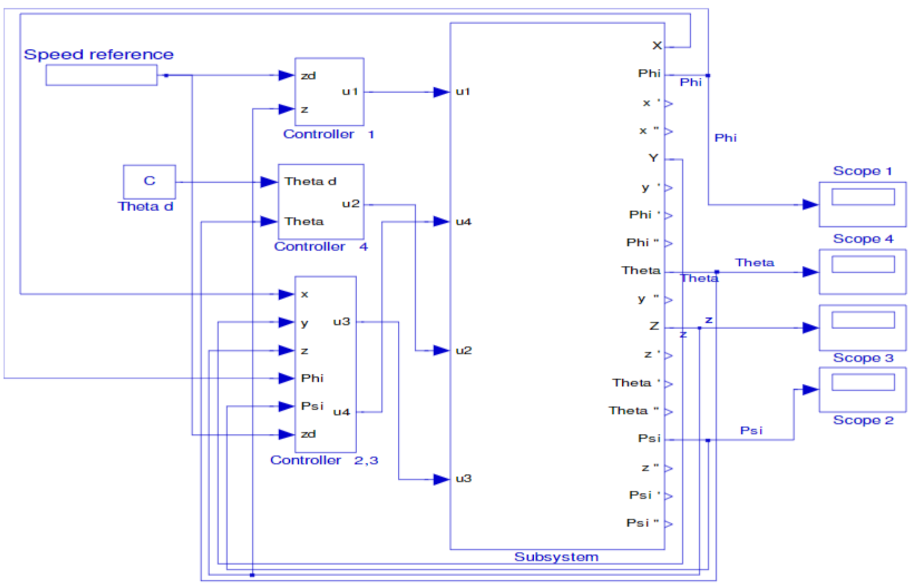

Figure 2-14: The simulation model with the PID controllers for the quadrotor [21]

algorithm, shown in Figure 2-14 [21], consists of all the intrinsic relationships between the controller, the inputs, the velocity reference, and the thrust.

2.5.4 Trajectory Tracking Control

This thesis uses a trajectory tracking controller from Mercado [23]. This controller is based on a translational dynamics controller and produces an output that specifies the desired direction as input control for attitude stabilization to ensure trajectory tracking. This trajectory tracking control is used for Leader-follower based Control to reach the desired position of the UAV swarm. As this method also uses Leader-follower formation control and requires automatic navigation. It differs from the method in this thesis in that it uses a virtual Leader but is well suited to this thesis in terms of trajectory tracking for a natural leader.

A quadrotor can be represented as a rigid body m with mass in space and an inertia matrix J subject to gravity and aerodynamic forces. Let us consider an inertial coordinate system $I = \{X\ Y\ Z\}$, a body-fixed coordinate system $B = \{e_1, e_2, e_3\}$. The UAV's position is represented by a vector $\xi = [x, y, z]^T$, and its roll, pitch, and yaw in the inertial coordinate system are represented by a vector $\Phi = [\varphi, \theta, \psi]^T$. Then the equations of motion in the inertial coordinate system are given by the Newton-Euler equations.

$$m\ddot{\xi} = T\hat{R}e_3 - mge_3 \tag{2.5.10}$$

$$J\dot{\Omega} = -\Omega_x J\Omega + \Gamma \tag{2.5.11}$$

where $T \in \mathbb{R}^+$ is the total thrust, g is the gravitational constant, $\Gamma \in \mathbb{R}^3$is the control moment in the body coordinate system B, and R is the rotation matrix. $\Omega = [p\ q\ r]^T$denotes the angular velocity in the coordinate system B. Ω_x denotes the skew-symmetric matrix such that $\Omega_x v = \Omega x v$. The kinematic relations $\Phi = (\dot{\varphi}, \dot{\theta}, \dot{\psi})$ and the angular velocity Ω are denoted as:

$$\Omega = Q\dot{\Phi} \tag{2.5.12}$$

and:

$$Q = \begin{bmatrix} 1 & 0 & -Sin\theta \\ 0 & Cos\phi & Cos\theta Sin\theta \\ 0 & -Sin\phi & Cos\theta Cos\theta \end{bmatrix} \tag{2.5.13}$$

This is where we define the error $\bar{\xi} = \xi - \xi_d$ and substitute it into Eq. (2.5.10) to obtain:

$$m\ddot{\bar{\xi}} = \left(T\hat{R}e_3\right)_d - mge_3 - m\ddot{\xi}_d \qquad (2.5.14)$$

Then the so-called switching function is:

$$\sigma_1 = k_1\bar{\xi} + k_2\int \bar{\xi}\,dt + \dot{\bar{\xi}} \qquad (2.5.15)$$

Where k_1 and k_2 are constant control parameters. The system is expected to remain on the defined surface by $\sigma_1 = 0$ because of the dynamic error on this surface:

$$k_1\dot{\bar{\xi}} + k_2\bar{\xi} + \ddot{\bar{\xi}} = 0 \qquad (2.5.16)$$

It is guaranteed to have an asymptotic convergence of $\bar{\xi} \to 0$ by choosing appropriate values of k_1 and k_2. The equivalent control u_{eq} that guarantees the system's dynamics to stay in the surface $\sigma_1 = 0$ is obtained from $\dot{\sigma}_1 = 0$. Substituting Eq. (2.5.14) in this last equation leads to the following:

$$k_1\dot{\bar{\xi}} + k_2\bar{\xi} + \frac{1}{m}\left(TR_{e_3}\right)_d - ge_3 - \ddot{\xi}_d = 0 \qquad (2.5.17)$$

Where $(TR_{e_3})_d$ is the control input. The equivalent control u_{eq} is given by:

$$u_{eq} = \left[\left(TR_{e_3}\right)_d\right]_{eq} = m\left(ge_3 + \ddot{\xi}_d - k_1\dot{\bar{\xi}} - k_2\bar{\xi}\right) \qquad (2.5.18)$$

Attracting the system dynamics to the surface $\sigma_1 = 0$ and keeping it there adds discontinuities despite the uncertainties and perturbations. Let us consider the function $sgn(x)$ defined as:

$$sgn(x) = \begin{cases} 1 & x > 0 \\ -1 & x < 0 \end{cases} \qquad (2.5.19)$$

and the vector:

$$Sgn(\sigma_1) = \begin{bmatrix} sgn(\sigma_{11}) \\ sgn(\sigma_{12}) \\ sgn(\sigma_{13}) \end{bmatrix} \qquad (2.5.20)$$

where σ_{11} , σ_{12} , σ_{13} are the components of the vector σ_1 . Then by making the assignment:

$$\dot{\sigma}_1 = k_1\dot{\bar{\xi}} + k_2\bar{\xi} + \frac{1}{m}(TRe_3)_d - ge_3 - \ddot{\xi}_d = -L_\xi Sgn(\sigma_1) \quad (2.5.21)$$

With l_ξ a natural, positive constant different from zero, it is possible to attract the system trajectories to the surface $\sigma_1 = 0$ in a finite time. The discontinuity control obtained from Eq. (2.5.21) is given by:

$$\left(TR_{e_3}\right)_d = u_{eq} - mL_\xi Sgn(\sigma_1) \quad (2.5.22)$$

To analyze the robustness of the control scheme, a bounded uncertainty model is considered:

$$m\ddot{\bar{\xi}} = \left(TR_{e_3}\right)_d - mge_3 - m\ddot{\xi}_d + \Delta f(\xi) \quad (2.5.23)$$

Where it is supposed that $\|\Delta f(\xi)\| < \iota$ with τ a positive constant. Let us consider Lyapunov's candidate function:

$$V = \frac{1}{2}\sigma_1^T\sigma_1 \quad (2.5.24)$$

Differentiating Eq. (2.5.24) concerning time leads to:

$$\dot{V} = \sigma_1^T\dot{\sigma}_1 \quad (2.5.25)$$

or equivalently:

$$\dot{V} = \sigma_1^T\left(k_1\dot{\bar{\xi}} + k_2\bar{\xi} + \ddot{\bar{\xi}}\right) \quad (2.5.26)$$

Substituting Eq. (2.5.24) in this last expression allows us to write Eq. (2.5.26) as:

$$\dot{V} = \sigma_1^T\left(k_1\dot{\bar{\xi}} + k_2\bar{\xi} + \frac{1}{m}(TR_{e_3})_d - ge_3 - \ddot{\xi}_d + \frac{1}{m}\Delta f(\xi)\right) \quad (2.5.27)$$

Using the control law (2.5.10) – (2.5.13), $\dot{V}$ takes the form:

$$\dot{V} = \sigma_1^T\left(-L_\xi Sgn(\sigma_1) + \frac{1}{m}\Delta f(\xi)\right) \quad (2.5.28)$$

From which:

$$\dot{V} \leq \|\sigma_1\| \left(-L_\xi + \frac{1}{m}\tau\right) \quad (2.5.29)$$

Moreover, $\dot{V}$ will be negatively defined when $-L_\xi \geq \frac{1}{m}\tau$.

So, the sliding mode control law Eq. (2.5.23) – Eq. (2.5.29) solves the trajectory tracking problem. It is important to note that:

$$R_d e_3 = \begin{bmatrix} R_{dx} \\ R_{dy} \\ R_{dz} \end{bmatrix} = \frac{(TRe_3)_d}{T_d} \quad (2.5.30)$$

With $T_d = \|(TRe_3)_d\|$. Besides, if ψ_d is constant, it is possible to write ϕ_d and θ_d explicitly as:

$$\phi_d = arcsin\left(-\frac{R_{dy} - R_{dx}tan\psi_d}{sin\psi_d tan\psi_d + cos\psi_d}\right) \quad (2.5.31)$$

$$\theta_d = arcsin\left(\frac{R_{dx} - sin\phi_d sin\psi_d}{cos\psi_d + cos\psi_d}\right) \quad (2.5.32)$$

For the attitude stabilization control, a proportional derivative controller is proposed that acts on the orientation error defined by $\bar{\Phi} = \Phi - \Phi_d$; this is:

$$\Gamma = -k_{do}\bar{\Phi} - \dot{k}_{po}\bar{\Phi} \quad (2.5.33)$$

where k_{do}, $k_{po} \in \mathbb{R}^+$.

2.5.5 Leader-follower based Control

The Leader-follower based control method is used in this thesis for the follower with no set path, while the main UAV formation control methods currently available are Leader-follower based Control [24] , Behavior-based control method [25], and Virtual structure method [25]. This section explains the reasons for using Leader-follower based Control by comparing the advantages and disadvantages of each method.

The Leader-Follower control method is a strategy that is often used in the control of multi-agent systems, such as a formation of unmanned aerial vehicles (UAVs). In this method, one intelligence is designated as the leader and the rest are followers that track the leader's movement. The followers are able to track the position and direction of the leader using parameters such as distance or speed. Within a multi-intelligence system, there can be one or multiple navigators, but only one navigator is responsible for controlling the shape of the group formation. By setting different position relationships between the navigator and the following intelligence, different network topologies, or formation shapes, can be achieved. The key feature of this method is that the collaboration between the members of the intelligence group is achieved through the sharing of information about the state of the leader intelligence.

One advantage of the Leader-Follower control method is that the leader, as the dominant player in controlling the movement of the entire system, can control the behavior of the entire group of intelligence with a given trajectory, which greatly simplifies the control process. This can be particularly useful in scenarios where the leader is able to accurately follow a predefined path and the followers are able to adjust their flight data based on the leader's position, heading, distance, velocity, and angular velocity. However, a disadvantage of this approach is the lack of direct feedback control in the system. If the leader misbehaves, it can directly lead to a disruption in the behavior of the followers, potentially causing the entire system to collapse. For example, if the leader moves faster than the followers can track, the followers will fall out of line. Additionally, if the system is too large, the volume of information can potentially overload the leader, negatively impacting the efficiency and stability of the system. In order to address these limitations, some researchers have suggested introducing feedback technology into the Leader-Follower method to achieve more stable and effective formation control. This can involve incorporating sensors and algorithms that allow the followers to adjust their behavior in real-time based on the performance of the leader, enabling them to more closely track the leader's movement and maintain the desired formation shape.

Despite the potential disadvantages of the Leader-Follower control method, it can still be an effective approach for the control of multi-agent systems in certain scenarios. For example, it may be particularly useful in situations where the leader is able to accurately follow a predetermined path and the followers are able to adjust their flight data based on the leader's position, heading, distance, velocity, and angular velocity. Additionally, the Leader-Follower control method may be more efficient than other approaches that require more complex communication and coordination between the agents. Overall, the effectiveness of the Leader-Follower control method will depend on the specific requirements and constraints of the application, and it may be necessary to combine it with other control strategies in order to achieve the desired performance.

One potential way to address the limitations of the Leader-Follower control method is to incorporate additional layers of control or to use a hybrid control approach that combines multiple control strategies. For example, it may be beneficial to include a supervisory layer that is able to monitor the performance of the system and adjust the behavior of the leader or the followers as needed. This could involve using advanced algorithms to detect and correct for deviations from the desired path or formation shape, or to adapt to changes in the environment or external disturbances. Additionally, it may be useful to incorporate other control strategies, such as consensus-based approaches or decentralized control, in order to enhance the robustness and adaptability of the system. By combining the strengths of different control approaches, it may be possible to achieve improved performance and more flexible and reliable control of multi-agent systems.

The main idea of the behaviour-based control method is to design the individual behavioural rules and local control schemes of each intelligent body in advance. This is based on the overall behavioral pattern expected to be produced by the control effect on the intelligent body system. It is a movement control method for the first effect and causes. Usually, the behavioural pattern of each intelligent body is stored in the formation controller like a 'library function.' When the system is running, the corresponding behaviour is executed according to the environmental information and control instructions, such as avoiding obstacles, forming a particular formation, changing to another formation, moving in the direction of the target, etc. For example, in an obstacle environment, the formation intelligence has to avoid collision with obstacles and other intelligence during their movement. The intelligent body system uses its sensing system to detect changes in the external environment and selects the desired behavior from the behavioral pattern "library function" based on the current system input. This is then used as the system response and output. Unlike the pilot-follower approach, the collaborative role in this method is achieved through sharing information, such as position and state input values, between the intelligence, for which an efficient and stable communication system is indispensable.

The idea of the virtual body method is to view the multi-intelligent body formation as a single rigid body structure. The intelligence points at certain fixed positions on this rigid body structure, using their positions in the structure's coordinate system as a reference, and when the intelligence move, i.e., when the multi-intelligent body formation moves, as long as the individual intelligence track their corresponding points in the rigid body structure. In other words, the coordinates of the intelligence in the reference coordinate system remain the same, which means that the relative positions of the intelligence remain unchanged, and the whole multi-intelligence system is always moving in a particular formation.

It is important to note that although the method provides explicit Control over the position information of the intelligence, it does not constrain their orientation information, which means that there is a great deal of freedom in controlling the orientation of each intelligence if it maintains the same reference position. Because of this capability, this method is commonly used in the formation control of artificial Earth satellites in the space industry to enable them to follow a given orbit and formation while adjusting their direction to gather necessary information to complete their tasks. In addition, if the intelligent body in the system is made to follow points other than its fixed point on the rigid structure, then different formation shapes and switching between them can be achieved. The advantages and disadvantages of the three methods are summarized in Table 2-4.

Formation control method	Advantages	Disadvantages
Leader-follower based	• Conservation energy • Reduce communication costs • Enhance group communication • Ensure group orientation direction	• Over-reliance on reliance on a single target
Behavior based	• Adaptable to sensor errors	• Over-reliance on pre-determined information and trigger conditions
Visual structure based	• The problem of interference with leader is avoided • High control accuracy	• Requires high communication quality • Requires high computing power

Table 2-4: Formation control method

In order to achieve the objectives of this thesis, it is necessary to use an approach that requires minimal computation and reduces the amount of communication data as much as possible, while also addressing the need for better queueing. The leader-follower approach is the best choice for this purpose, as it is able to meet these requirements effectively. Additionally, the method described in this thesis is designed to handle the situation where the leader of the formation is lost or becomes corrupted during flight. In such cases, the method is able to assign a new Leader to the formation, which requires storing a preset path in each UAV. This ensures that the formation is able to continue functioning even in the event of the loss of the original leader, which is an important consideration in order to maintain the stability and reliability of the system. Overall, the leader-follower approach is an effective method for enabling autonomous positioning and navigation of UAV swarms, and the method described in this thesis is able to address the challenges and limitations of this approach in order to achieve optimal performance.

2.5.6 Structures of UAV-Formation

In this thesis, several different formations are designed for testing based on Leader-follower based Control. In practice, different formations are used depending on different environments [26].

There are two main configurations of UAV formations in the leader-follower model, i.e., follower formation and diamond formation. The schematic diagrams of the two types of stratigraphic configurations are shown in Table 2-4. The development of these two typical configurations forms a variety of other complex stratigraphic formations.

Each of these two formation configurations has advantages and disadvantages. The follow-on formation covers the smallest area of land, thus reducing the probability of detection by reconnaissance forces or enemy ground radar and increasing the formation's survivability but finding less information. The diamond formation is advantageous because it enables a large area to be covered and ensures that each team member has good visibility, potentially increasing the likelihood of detecting the enemy. The two basic formation configurations are diamond formation and following

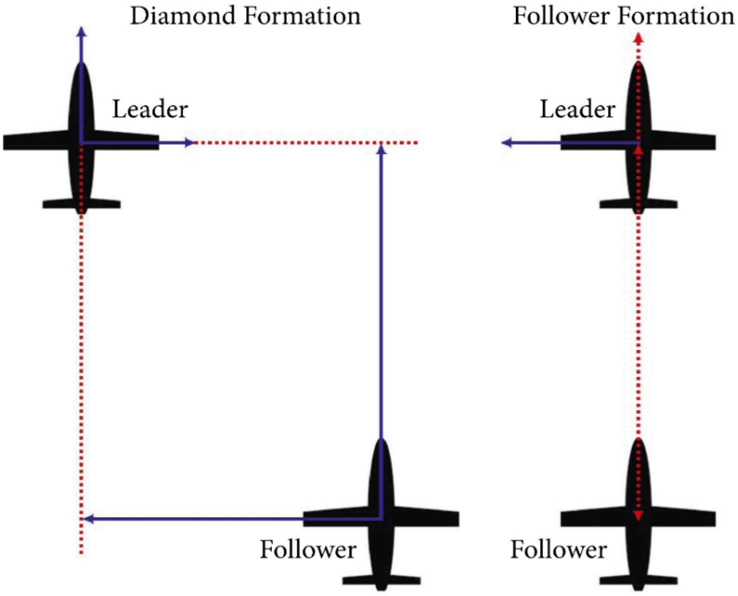

Figure 2-15: Formation structure of different UAVs [26]

formation, which can evolve into many common UAV formations, such as the "plus" and "arrow" shapes. Common stratigraphic structures are shown in Figure 2-16 [26].

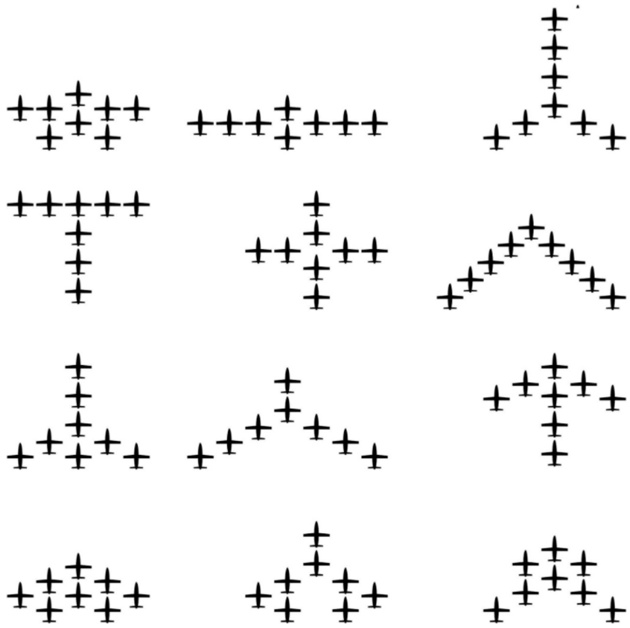

Figure 2-16: Typical configuration of UAV formations [26]

Since UAV formation dramatically impacts the overall performance of the whole formation, some new formation structures are gaining more and more attention to improve formation efficiency, reduce energy loss, and avoid danger, as shown in Figure 2-17 [26].

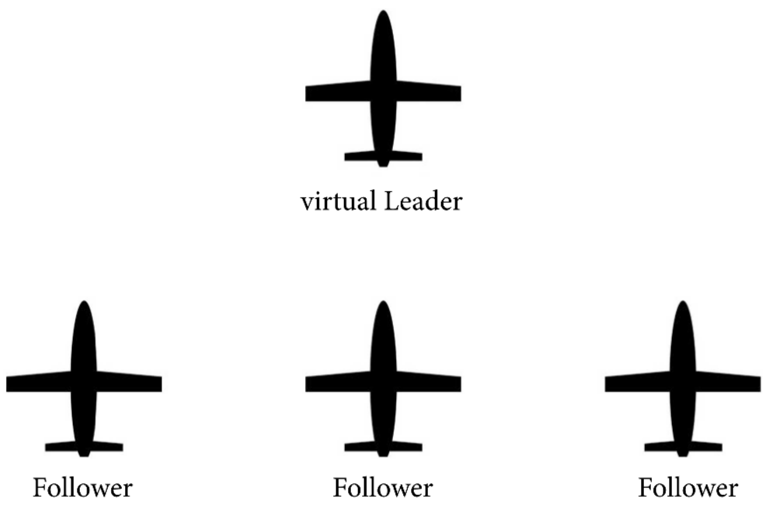

Figure 2-17: Virtual leader formation structure [26]

The formation of a virtual leader structure within the formation can significantly reduce communication latency between formation members and decrease the computational load on the onboard computers. This can be particularly useful in situations where there are a large number of UAVs in the formation, as it helps to alleviate the burden on the communication and computing resources. As a result, it is possible to increase the number of formations building blocks, which can further enhance the overall performance of the formation. Other new stratigraphic structures also have their own unique advantages and may be more suitable for certain missions or environments.

In practical applications, it is important to design the formation rationally in order to achieve the best mission efficiency based on the specific mission requirements, the number of formation members, the hardware communication capabilities, the processing power of the airborne processor, and the logistics of the UAV swarm transition. It is necessary to carefully consider these factors in order to optimize the formation for the specific mission at hand. The formation should be designed to maximize efficiency and effectiveness while also taking into account the limitations and constraints of the system.

2.6 Summary

The flight control system is mainly used for flight attitude control and navigation. The first thing to know for flight control is the vehicle's current state. As shown in Table 2-5, UAV motion requires basic parameters expressed as 16-element vectors.

Data	***Element***	***Units***	***Original Source***
Positions	[x y z]	m	GPS
Velocities	[vx vy vz]	m/s	GPS
Accelerations	[ax ay az]	m/s	IMU
Orientation	[qw qx qy qz]		IMU
Angular Velocities	[wx wy wz]	r/s	IMU

Table 2-5: Basic parameters of UAV motion requirements: based on [12]

As a multi-rotor aircraft is an inherently unstable system, the power of each motor must be continuously regulated and distributed at an ultra-high frequency in order to achieve stable hovering and flight. For this reason, even the simplest release of the rocker aircraft for independent hovering action requires continuous monitoring of the 16 flight control parameters. A series of "cascade controls" are used to achieve a stable hover.

While this may seem simple, the operation of the flight control system is actually quite complex and involves a range of technical challenges.

One of the most fundamental and difficult technical challenges of the flight control system is accurately sensing the various states of the aircraft. If there are problems or errors with the perception data, it can lead to abnormal actions by the drone.

Due to the limitations of current sensor technology, the data measured by these sensors is subject to specific errors and may be affected by the environment, which can impact the accuracy of state estimation. To ensure good flight performance, it is necessary to fully utilize the data from each sensor and fuse high-confidence states through techniques in electronic signal processing, such as combined navigation technology. This technology combines the strengths and weaknesses of GPS, IMU, barometer, and geomagnetic compass to obtain more accurate state measurements by fusing data from multiple sensors. Some basic methods, such as the INS navigation system, are also described in this chapter.

In general, the technical background outlined above provides the theoretical foundation for the autonomous positioning and navigation of UAV swarms described in this thesis. In the following chapter, we will delve into the current technologies related to UAV swarms in more detail and provide a comprehensive overview of current UAV swarm positioning techniques. Despite the significant progress that has been made in this area, there are still many challenges to be addressed in order to achieve reliable and safe navigation and positioning for UAV swarms, particularly in outdoor environments.

3 State of the Art

3.1 Preliminary Research

A search of the current literature on UAV swarm flocking strategies and internal coordination, as well as a study of mainly survey-based literature, yielded relevant challenges for UAV swarms involving various aspects, as shown in Figure 3-1 [27]. While among the challenges of designing physical locations include determining the optimal deployment of drones, physical coordination of drones and trajectory optimisation. In this section, several recent studies are summarised, which suggest the use of ML methods to address location-related challenges.

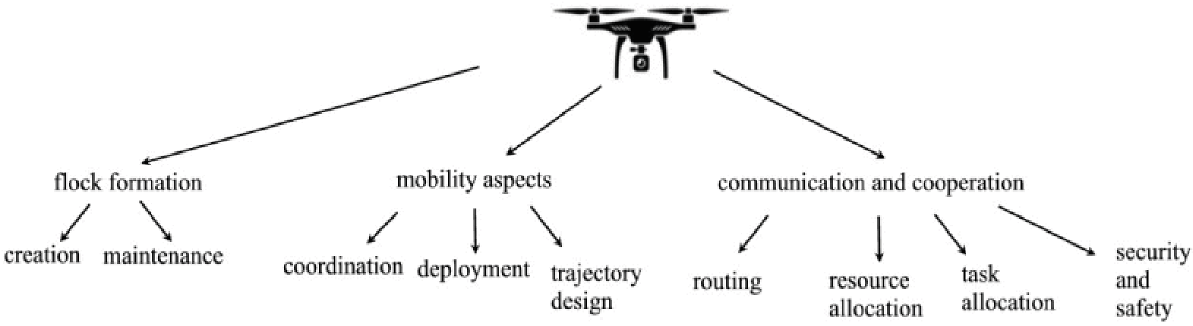

Figure 3-1: Challenges in UAV flocking [27]

Given a flock of flying objects, flock members must coordinate to avoid cohesion. Ragi [28] defines the fundamental swarming challenge for any flock of flying objects as follows: UAVs try to stick together and avoid collisions with each other and other objects in the surrounding environment. The three basic rules for maintaining flocking behaviour are Separation: avoiding collisions with nearby flock mates; Alignment: trying to match speed with nearby flock mates; and Cohesion: trying to stay close to nearby flocks. Like the clustering behaviour observed in animals and insects, autonomous drone coordination should be ensured.

Quintero [29] developed a novel algorithm that enables multiple UAVs to position themselves in a cluster to distribute a given sensing task among group members, assuming a leader-follower network topology. They concentrate on the control strategy of the followers and formulate a cost function that considers the distance and heading relative to the leader, as well as a stochastic kinematic model to support the cluster. After that, they used dynamic programming to minimise the expected cost of each follower. They assume that there is a predefined flock leader, known to the entire flock, and their goal is to optimise the distance and head of the followers relative to the leader.

Hung and Givigi [30] proposed using a model-free RL approach to enhance autonomous coordination among drones in a flock. They used a leader-follower policy where Peng's Q (λ) [31] followers with variable learning rates were used to learn control strategies that facilitate clustering. The problem was constructed as a Markov decision process (MDP) in which agents were modelled as drones that experience stochasticity due to disturbances such as wind and control noise and weight and balance problems. The learned strategy is compared with a dynamic programming approach using stochastic optimal control solutions. Simulation results demonstrate the feasibility of the proposed learning method, enabling the agent to learn how to aggregate in a leader-follower topology while operating in a non-stationary stochastic environment.

Tsai [32] focused on vision-based collision avoidance for UAVs. First, images from UAV cameras are fused based on a deep CNN. After that, recurrent neural networks (RNNs) are constructed to obtain high-level image features for target tracking and extract low-level image features for noise reduction. The system distributes the computations across multiple UAVs to efficiently perform target detection, tracking and collision avoidance. To achieve high overall system performance, the study used a semi-total error rate (HTER) accuracy measure based on the error rejection and acceptance rates.

Wu [33] studied the issue of path conflicts in UAV clusters and developed a method to calculate the probability of UAV collisions under the constraints of the mission space and the number of UAVs. In cluster flight mode, automatic tracking and predicting UAV cluster trajectories were achieved to avoid intra-cluster path conflicts. A Kalman algorithm-based state estimation method is proposed for the inconsistency problem caused by noise. Cluster state prediction and collision probability are calculated to prevent the formation of UAV clusters from conflicting on the path during flight. Simulation results are used to validate the effectiveness and validity of the method in multi-UAV formation flight planning.

Based on a priori information subject to inter-UAV distance constraints and target probabilities, Wang [34] proposes a collaborative search algorithm for secure communication. The results show that the algorithm can effectively overcome the communication distance constraint to complete the co-movement target search.

Ruby [35] presents an algorithm for estimating the state of UAVs. The algorithm combines measurements from air data systems to correctly correlate predicted attitude information with aircraft speed information. Their experimental results show that the algorithm effectively estimates roll, pitch, yaw and Heading angles and provides smooth estimates of the angle of attack and sideslip. This new attitude estimation method can effectively distinguish between the yaw and heading angles of the aircraft, with the attitude estimate appropriately adjusted by air data system measurements.

3.2 Positioning Technologies

In this chapter, some of the technologies directly related to this thesis are presented, namely the current state of the art of positioning UAV swarms. UAV positioning has various technical details depending on various environments and conditions. As shown in Table 3-1 below, this thesis lists some common cases where swarms of UAVs are positioned indoors or outdoors via internal communication or vision sensors, some of which also incorporate machine learning methods to improve positioning accuracy.

3.2.1 Inter-Communication-based Positioning

Reference	***Positioning techniques***	***Optimisation Methods***	***Data***
	Inter- communication-based Positioning		
Xiaoqiang Qi. (2021) [36]	IMU	Extended Kalman Filter	noisy and delayed position data
Zhimin Han et al. (2018) [37]	distance position measurements -based scheme	complex Laplacian-based formation control scheme	distance, velocity, all agents Share: Orientation
Zhiyun Lin et al. (2015) [18]	relative position measurements-based scheme	fully distributed localisation algorithm	Known anchor node position, other nodes unknown anchor node positions in a typical global coordinate frame and relative position measurements in local coordinate frames between node pairs
Kexin Guo et al. (2017) [38]	distance measurements-based scheme	Kalman Filter	distance, self-displacements
William Power et al. (2020) [39]	dead reckoning	Multi-Target Gaussian Conditional Random Field (MT-GCRF)	share: predicted global position
Xiaoyang Liu et al. (2018) [40]	dead reckoning, distance measurements-based scheme,	traditional federal Kalman filter	the relative navigation system provides the distance and angle information between the leader and the follower, and the dead reckoning system provides

	bearing measurements-based scheme		information on the position coordinates and speed of each UAV
Mario Coppola et al. (2018) [41]	distance measurements-based approach	discrete-time Extended Kalman Filter (EKF)	the MAVs exchange on-board states (height, velocity orientation) while the signal strength indicates the range
Che Lin et al. (2016) [42]	bearing measurements-based scheme	distributed source localisation	bearing angle
Vision-based Positioning			
Fabian Schilling et al. (2022) [43]	the relative visual localisation	attractive/repulsive flocking algorithm	/
MARTIN SASKA ET AL. (2017) [44]	the relative visual localisation	algorithm (details described in Krajník et al. (2014))	information from an on-board camera, data from IMU, the altitude and velocities from the intelligent sensor

Table 3-1: Status of Art for inter-communication-based localisation and vision-based localisation: based on [18, 26, 37, 38, 39, 40, 41, 42, 43, 44]

Intra-communication-based positioning usually refers to the relative positioning between UAVs in a swarm by sharing information in real-time and by simple on-board sensors. Thus, the UAV's actual position, velocity, and other information are estimated. As shown in Table 3-1, this thesis summarises the latest technologies that are directly relevant. They all use internal communication and combine some other methods to improve positioning accuracy.

Qi [36] proposes a state estimator for estimating the total ground UAV state in cluttered environments. Where only noise and delayed position data are available, they implement an extended Kalman filter to fuse the position data with the inertial measurement unit (IMU) data, which is also used to compensate for delays in the position data. Simulation results demonstrate the performance of the method.

Han [37] proposes a usage method if the UAVs can measure their velocity and the rate of change of distance from neighbouring UAVs and that all UAVs share a common direction. Based on such a setup, a relative positioning scheme is developed for each UAV. Later, a formation control scheme with relative localisation and a leader-follower network is introduced by combining the adapted scheme with a complex Laplace operator-based formation control scheme.

Lin [42] strictly considers the problem of self-localisation, i.e., determining the location of each sensor by the sensor node itself rather than by other nodes. Once the sub-location problem is solved, a distributed verification algorithm is developed to determine whether a given sensor network is self-locating. The point is that instead, the paper develops a distributed localisation algorithm that computes the position of each sensor node based on its local measurements, exchanging some information with its neighbours. The proposed localisation algorithm is of iterative and linear form with guaranteed global convergence.

Guo [38] proposes an indirect collaborative relative positioning method. The position of the UAV relative to its neighbours is estimated based only on distance and self-displacement measurements in a GPS-rejected environment. The method approach consists of two phases. Initially, assuming no knowledge of its own and its neighbours' states and subject to environmental or mission constraints, each UAV solves a functional 2D relative localisation problem to obtain an estimate of its initial position relative to the statically hovering quadcopter. An extended Kalman filter subsequently refines this to account for noise in the distance and displacement measurements. The second stage applies the extended Kalman filter strategy to the case where all UAVs move simultaneously, beginning with a fine initial relative positioning guess. In this phase, each UAV is co-located by the inter-UAV distance given by ultra-wideband and by exchanging the self-displacement of neighbouring UAVs. Extensive simulations and flight experiments are presented to confirm the proposed relative localisation strategy and algorithm's effectiveness.

Power [39] proposes a structure learning approach to optimise error accumulation in heading projections. It is believed that this dataset can be used to train a model that will enhance the accuracy of heading prediction in GPS-denied environments by forecasting future deviations based on changes in the structure of the swarm network. It is assumed that such a network collects some information about environmental deviations that introduce cumulative errors in heading projections. By using an appropriate structural machine learning tool, this information can be used for the correction of the estimated position of each UAV in the swarm.

Liu [40] proposes a cooperative UAV navigation algorithm based on a joint filtering structure. This addresses the problem of rapidly increasing UAV navigation errors for each low-precision navigation device in the UAV formation, as well as the high computational burden of centralized Kalman filtering in cooperative navigation. The state model for each sub-filter is established and the measurement model of the sub-filter is constructed based on the output of the navigation system and related navigation information. The measurement equation for the central filter is then constructed using the error correction of the UAV navigation system. The Kalman filter is improved through the use of a federal filter structure to estimate the errors of each UAV navigation system, resulting in a low computational effort and good error tolerance.

What the above methods have in common is that they all use an information-sharing mechanism between drones. The flight data is collected to optimise the positioning accuracy of UAVs for better navigation of UAV swarms for autonomous flight and obstacle avoidance.

3.2.2 Vision-based Positioning

Vision-based positioning is also the current mainstream, with numerous advantages, such as high relative localisation accuracy and better obstacle avoidance capability. A few of the latest methods in recent years are listed below.

Schilling [43] proposes a visual neighbour selection model. The model addresses visibility limitations caused by occlusion from the robot's perspective by providing a perceptually reasonable alternative to neighbour selection for the prevalent but impractical metric of assuming that an agent can perceive any neighbour within a given radius regardless of occlusion. The paper simulates vision-based populations of up to one thousand point-quality agents and implements a simple attraction/repulsion cluster algorithm to enable them to perform collective waypoint navigation. The results show that the agents perform poorly for all visible neighbours, especially at high densities and population sizes exceeding tens of agents.

Saska [44] developed a position stabilisation mechanism that uses data from visual relative localisation units in control feedback. The paper shows three examples of group motion planning. As the third level of control is designed for the navigation of the entire MAV group and its stability in the desired shape, motion planning can be changed dynamically. These methods employ the concept of adaptively evolving group behaviours that are built to reduce uncertainty in relative positioning. The group motion planning approach proposed in the paper uses a model of the localization system that is derived from a theoretical analysis of the visual system and an experimental evaluation of the system's performance in real-world scenarios.

Vision-based positioning has better accuracy and adaptability than just localisation with internal communication. However, it requires higher cost and computational effort. Unlike this thesis, the research direction is to optimise the localisation capability of low-cost UAVs, i.e., using only internal communication and low-cost sensors.

3.3 Research at Computer Engineering Professorship

In particular, the Professorship of Computer Engineering at the Faculty of Computer Science of the Chemnitz University of Technology carries out a research project aiming to develop a UAS capable of autonomously inspecting power lines and detecting faults. This is the Automated Power Line Inspection [52]-[55] project, referred to as APOLI. The UAS inspects power lines by navigating the UAV around the power poles and evaluating the video footage captured by a camera mounted on the UAV's gimbal. The UAV must stay within the power lines themselves. In the APOLI project, this collision avoidance problem, however, has not been tackled yet and a procedure for this has yet to be implemented. However, the collision avoidance problem is a much more tractable task if an environment map is available.

3.4 Summary

More directly relevant to this thesis is the study of UAV swarms achieving accurate self-orientation in GPS-rejected environments. Saska [44] used an airborne vision sensor to locate UAVs, and the vision sensor was used to observe the visual indicators of no UAVs. The group orientation was combined with observed landmarks to achieve an excellent simultaneous localisation and mapping (SLAM) framework. Lockspeiser [45] investigated RF ranging. This can provide a method for inter-drone localisation within a swarm by observing the received signal strength of RF signals from neighbouring UAVs.

When position feedback is lost, aerial dead reckoning techniques can be used. Dead reckoning is an old and standard method of localisation. The position at the current moment is known, and then sensors such as IMU are used to estimate the position at the next moment. This type of control assumes that the UAV position's current position estimate is correct and flies accordingly. Such a system is prone to rapid accumulation of errors when used in isolation. There have been several efforts to improve the estimation of dead reckoning. For example, the combination of dead reckoning and the Kalman filter improves trajectory tracking [46]. The dead reckoning can also be combined with other methods to improve accuracy. This thesis uses a combination of GPS data by neighbouring UAVs with dead reckoning and uses Extended Kalman Filter to achieve accurate positioning.

Generally, in a two-dimensional scene, distance measurements are required for three non-common-line nodes with known positions to determine the position of a node uniquely. Flip-flop ambiguity may occur when distance measurements are available for only two nodes whose positions are known.

This work focuses on the problem of estimating the location of a UAV that has temporarily lost GPS connectivity. In this case, it may be possible to use the UAV's known location before the GPS loss for flip-flop ambiguity resolution. That is, the location of the UAV prior to the GPS disruption may help determine on which side of two known nodes the UAV is located. Thus, distance measurements of two UAVs with known locations may be sufficient to uniquely determine the location of the UAV that temporarily lost its GPS connection. However, it is essential to note that distance measurements for only two UAVs with known locations may still have flip-flop ambiguity issues in some exceptional cases. When this happens, more information must be considered, such as the maximum angular speed of the drone.

In summary, a 2D distance measurement of two UAVs with GPS may be sufficient to estimate the position of the UAV that temporarily loses GPS connectivity based on the position measurement of the UAV before it loses GPS and the geometric relationship between the initial formation and the UAV that temporarily loses GPS connectivity.

4 Methodology

This chapter consists of four parts, starting with an introduction to the control architecture of the UAV swarm. Then the software information for implementing the method is described. Then the control model of the UAV swarm is described in detail. Finally, the most critical positioning method of this thesis is presented.

4.1 Overview

This thesis presents a control scheme for a formation of UAVs. The UAVs use distributed communication, meaning each UAV communicates with and considers one neighboring UAV to be a local leader. For formation control, a leader-follower approach is employed. In this approach, the leader UAV follows a predefined path and the other UAVs in the formation determine their flight data based on the leader's position, heading, distance, speed, and angular velocity. Two types of positioning systems are used in the formation: UWB positioning for indoor environments, and INS positioning for outdoor environments. While UWB positioning tends to have good accuracy, INS positioning can be less accurate due to the influence of wind and the limitations of GPS positioning. If most of the UAVs in the formation lose GPS or UWB capability, they can still receive positioning information from UAVs that have not lost these capabilities. By combining this relative position information with their own IMU data, they can reduce positioning errors in GPS-deficient environments using an EKF. The proposed control scheme is demonstrated through simulations involving different formation shapes and path complexities.

4.2 Software Information

As shown in Figure 4-1, the project developed several modules using the Matlab UAV toolbox for simulating swarms of UAVs. The system parameterises the UAV's formation shape, trajectory, and positioning accuracy. Machine learning methods facilitate the comparison and testing different formation types and positioning methods.

The Formation module defines the formation using a directed graph, i.e., it defines the relative positions of neighbouring drones in the formation and specifies the direction of information transfer from neighbouring drones. The trajectory defines the waypoints and flight times of the flight, i.e., waypoints one second apart. The environment module accepts the formation and trajectory parameters and defines by default the size of the scene and the type of drone. The control module, a 16-element vector, controls the drone. Accuracy defines the positioning accuracy of the drone, i.e., the accuracy of the Inertial navigation system and GNSS/GPS. The Kalman Filter is then used to reduce the positioning errors. Finally, the flight data is collected and compared in the Simulation module. In the following sections, the system setup is described in detail.

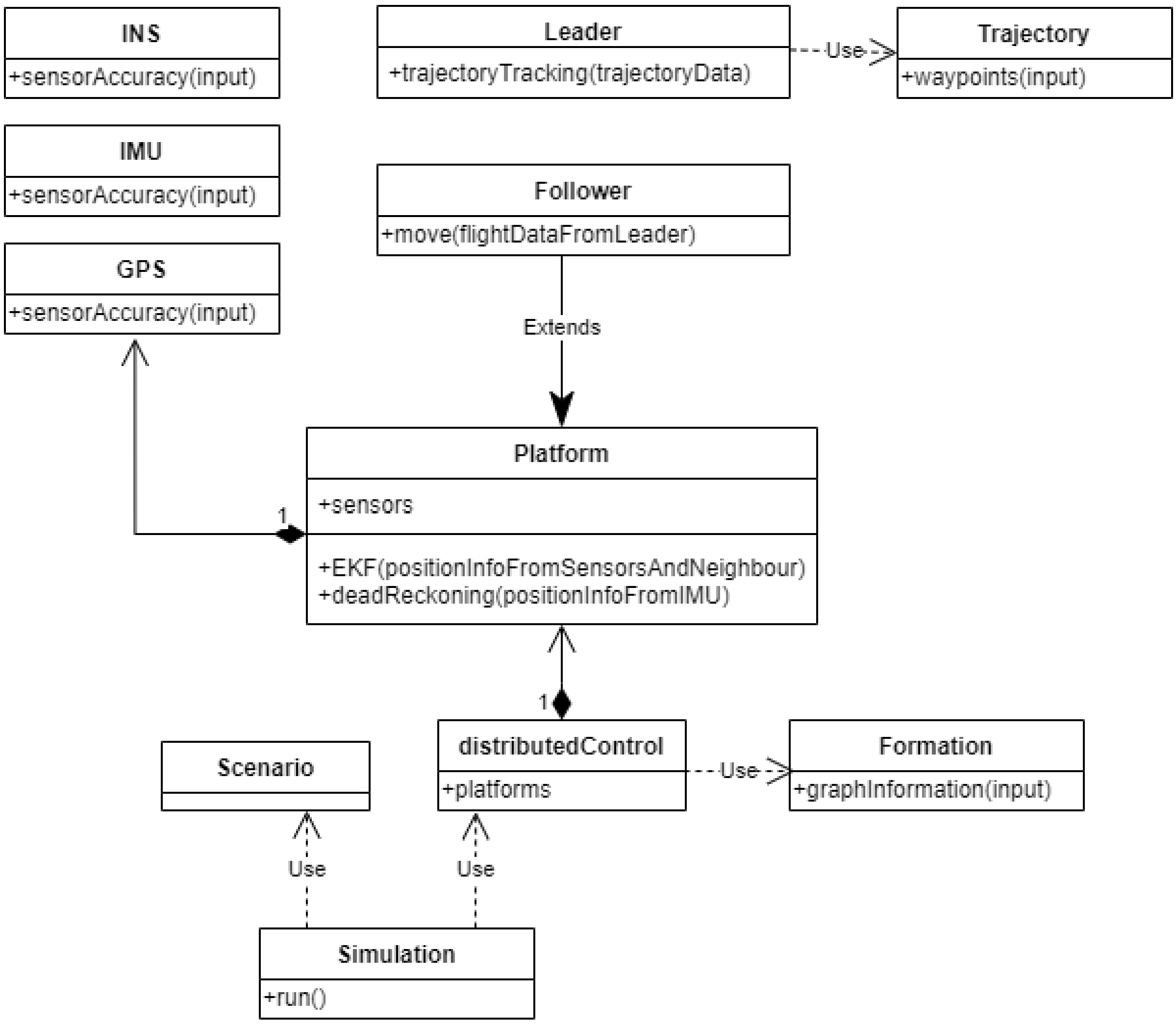

Figure 4-1: Simplified UML class diagram of the software

This project uses the UAV toolbox [47] to simulate UAV formations. The toolbox provides a powerful and intuitive simulation environment that can be used to design, simulate, test, and deploy tools and reference applications for UAV and UAV applications. Users can design autonomous flight algorithms, UAV missions and flight controllers. The UAV Toolbox allows users to design and test autonomous flight algorithms and controllers, as well as simulate UAV missions in a 3D or 2.5D environment. It also provides the ability to simulate the outputs of various sensors, including cameras, LIDAR, IMU, and GPS.

The Flight Log Analyzer application included in the toolkit allows for interactive analysis of 3D flight paths, telemetry data, and sensor readings from standard flight log formats. The toolkit also provides rich reference examples, such as parcel delivery using multi-rotor drones. The toolkit includes support for generating C/C++ code for rapid prototyping, hardware-in-the-loop testing, and standalone deployment to hardware platforms such as the Pixhawk® Autopilot.

The toolkit contains modules for Flight Log Analysis, Scenario Simulation, Planning and Control, Data Processing and Visualization and MavLink Support. The first three modules are the main ones used in this project. The last two modules were not used due to the focus of the study.

4.3 Dead Reckoning

Dead reckoning is a prevalent positioning method [46]. After knowing the position at the current moment, the position at the next moment is then estimated by sensors such as IMU. Assume that the GPS provides 10Hz positioning information at the time of UAV positioning. The vehicle position will also move a lot within the 0.1-second interval of each GPS information coming. It is necessary to determine the direction and distance of the vehicle. Therefore, DR is the most basic and necessary algorithm for autonomous driving.

DR relies on using a known position and extrapolating from that position to a new position by considering speed and heading. Sailors and pilots have historically used this technique, and many navigation systems still rely on heading projection to some extent, but this technique has many drawbacks. The navigator using heading projection, determines the current position based on known speed, Heading, and the last confirmed position. In a heading projection, the navigator starts with a known position fix obtained by observation and other tools. This position has an indicator on the chart indicating that it is a fixed position fix and not an extrapolation method. The next time the navigator wants to estimate the position, the elapsed time is considered together with the recorded travel speed and heading. The DR can also include adjustments for currents and winds, which can cause the ship to drift off course due to these factors. A new position is entered on the chart to reflect the calculation results. The aircraft can use heading projection techniques to navigate if the positioning instruments begin to fail. A severe problem with DR is the risk of cumulative error. Recording position information relative to a previous position (rather than using new data) has the potential to compound the error. Even if the initial position is correctly located, subsequent positions may be wrong and worsen over time.

Imagine that the GPS receives the position and attitude of the UAV at 10hz. The drone flies at 33m/s; then, the vehicle acquires data every 3.3m of movement. Such an interval is difficult for the UAV to control in an environment with obstacles. So, GPS must be combined with IMU sensors, as mentioned in Chapter 2.3.1. IMU update frequency is usually measured in kHz. For example, the update frequency of ps-imu-3000a is 30Khz. Assuming the use of this IMU, the position update speed will change from 3.3m/time to 0.011m/time during the UAV flight at 33m/s. It should be noted that GPS can give position and attitude. However, IMU can't, so IMU must determine the present position and attitude by accumulating the change amount of each axis.

Moreover, we want to combine GPS and dead reckoning because the projection is based on IMU; due to only relying on the IMU to provide each axis velocity, acceleration, and each axis rotation speed, acceleration, the sensor has a small error, which will lead to the projected position error will increase more and more over time. Such as when it is necessary to project the angle ϕ. where ϕ_0 is the initial angle. $\dot{\phi}$ is the angular velocity at a particular moment, obtained from the sensor and regarded as a known parameter. η is the angular velocity measurement noise at a particular moment. The distribution of the noise $\sim N(0, \sigma^2)$.

$$\phi_1 = \phi_0 + \left(\dot{\phi}_0 + \eta_0\right)\Delta t \tag{4.3.1}$$

$$= \phi_0 + \dot{\phi}_0 \Delta t + \eta_0 \Delta t$$

where $\eta_0 \Delta t$ is sensor noise $\sim N(0, \sigma^2 \Delta t^2)$.

$$\phi_2 = \phi_1 + \left(\dot{\phi}_1 + \eta_1\right)\Delta t \tag{4.3.2}$$

$$= \phi_1 + \dot{\phi}_1 \Delta t + \eta_1 \Delta t$$

$$= \phi_0 + (\dot{\phi}_0 + \eta_0)\Delta t + \dot{\phi}_1 \Delta t + \eta_1 \Delta t$$

$$= \phi_0 + \dot{\phi}_0 \Delta t + \eta_0 \Delta t + \dot{\phi}_1 \Delta t + \eta_1 \Delta t$$

$$= \phi_0 + (\dot{\phi}_0 + \dot{\phi}_1)\Delta t + (\eta_0 + \eta_1)\Delta t$$

where $(\eta_0 + \eta_1)\Delta t$ is sensor noise $\sim N(0, 2\sigma^2 \Delta t^2)$. For the same reason:

$$\phi_3 = \phi_2 + \left(\dot{\phi}_2 + \eta_2\right)\Delta t \tag{4.3.3}$$

$$= \phi_0 + (\dot{\phi}_0 + \dot{\phi}_1 + \dot{\phi}_2)\Delta t + (\eta_0 + \eta_1 + \eta_2)\Delta t$$

where $(\eta_0 + \eta_1 + \eta_2)\Delta t$ is sensor noise $\sim N(0, 3\sigma^2 \Delta t^2)$.

Eventually, the magnitude of the noise accumulates with Δt. In the case of using IMU only for projection, the error of projection will slowly grow with the accumulation of time.

The combination of IMU and GPS makes up for the long-range error accumulation of IMU and the short-range data reception interval of GPS.

For the accumulation of angular errors, a similar principle is used in Eq. (4.3.2) for replacing the angles by the positions $[x, y, z]$.

$$x_3 = x_2 + \left(\dot{x}_2 + \eta_{2,x}\right)\Delta t \tag{4.3.4}$$

$$y_3 = y_2 + \left(\dot{y}_2 + \eta_{2,y}\right)\Delta t \tag{4.3.5}$$

$$z_3 = z_2 + \left(\dot{z}_2 + \eta_{2,z}\right)\Delta t \tag{4.3.6}$$

It is important to note here that the measurements given by the IMU are based on their coordinate system. Attitude information is used to determine the position of the UAV in the global coordinate system. The calculation process is described in Chapter 3.1.1.

An IMU typically consists of three single-axis accelerometers and three single-axis gyroscopes. The gyroscope measures the angular velocity and acceleration of the object in three dimensions relative to the coordinate navigation system, while the accelerometer detects the acceleration of the object in a vehicle-based coordinate system and uses this information to determine the attitude of the object.

IMU is the output of each axis's velocity, acceleration, and angular acceleration derived around its body coordinate system. Then, to observe the position and attitude in the global coordinate system, it is necessary to rotate each axis concerning the inertial coordinate system based on the angular velocity accumulated. Since the transformation being performed is from a body-fixed coordinate system to an inertial coordinate system, no translation is required.

The body frame refers to the frame (coordinate) of the IMU fixed on the unmanned vehicle or UAV. Moreover, the IMU output of each axis of rotation speed, and acceleration, through these data, can be integrated to get the corresponding angle. To measure the angle or angular acceleration, there is a reference axis. This reference axis is the inertial frame, and the inertial coordinate system is created to simplify the transformation from the world coordinate system to the inertial coordinate system. The origin of the inertial frame is coincident with the origin of the object coordinate system, and the axes of the inertial frame are parallel to those of the world coordinate system. With the introduction of the inertial coordinate system, the conversion from a coordinate object system to an inertial coordinate system requires only rotation. From the inertial coordinate system to the world coordinate system requires only translation.

4.4 Extended Kalman Filter

In this thesis, we use a combination of relative position measurements with an extended Kalman filter to localise a UAV that has temporarily lost its GPS signal. Specifically, we consider a set of UAVs that may have different functions and cooperate to accomplish trajectory tracking. At a certain point, one of the UAVs loses its GPS signal. The drone without a GPS signal will use the EKF to estimate its position. If the UAV that lost its GPS signal can measure its relative position by Chapter 2.2.1, we fuse that position with the IMU-derived position using the EKF. We want to arrive at a more accurate position.

To facilitate the simplification of the equation, only the motion of the UAV in the horizontal plane is considered here, i.e., only the two-dimensional localisation problem is considered. Moreover, the assumption is that the UAV can be decoupled in vertical and horizontal motions. This assumption is widely used to simplify the modelling of aircraft dynamics [48]. Moreover, this thesis uses the three-dimensional positioning method in the actual simulation.

The following continuous time equations give the dynamic simulation of the UAV:

$$\dot{x}(t) = v(t)\, cos\, \eta(t) \tag{4.4.1}$$

$$\dot{y}(t) = v(t) \sin \eta(t) \tag{4.4.2}$$

$$\dot{\eta}(t) = \omega(t) \tag{4.4.3}$$

$$\dot{\omega}(t) = \varepsilon_{\omega}(t) \tag{4.4.4}$$

$$\dot{v}(t) = \varepsilon_{v}(t) \tag{4.4.5}$$

Where $\{x(t), y(t)\}$ are the water plane coordinates of the UAV at time t and $\eta(t)$ is the Heading, $\omega(t)$ is the angular speed, and $v(t)$ is the ground speed. $\varepsilon_{\omega}(\cdot)$ and $\varepsilon_{v}(\cdot)$ are supposed to be stationary, zero mean, Gaussian, independent, white noise processes, with covariances $E[\varepsilon_{v}(t)\varepsilon_{v}(s)]$ and $E[\varepsilon_{\omega}(t)\varepsilon_{\omega}(s)]$ provided by and $Q_{v}\delta(t-s)$ and $Q_{\omega}\delta(t-s)$ respectively. δ is the Kronecker delta function. They simulate the acceleration caused by wind and control manoeuvres, etc. The values of Q_{ω} and Q_{v} are taken from the maximum angular speed and the experience of the potential changes in the linear speed of the UAV.

A discrete-time equation set corresponding to Eq. (4.4.1) through Eq. (4.4.2) will be an approximation given the nonlinearities in the continuous-time model and due to the treatment of noise terms. The simplest approximation is one based on Euler's equation.

This is a famous procedure for certainty equations and the foundation for approximating stochastic equations. The discrete-time dynamic model is shown in the following:

$$x_{k+1} = x_k + v_k \Delta t_k \cos \eta_k \tag{4.4.6}$$

$$y_{k+1} = y_k + v_k \Delta t_k \sin \eta_k \tag{4.4.7}$$

$$\eta_{k+1} = \eta_k + \omega_k \Delta t_k \tag{4.4.8}$$

$$\omega_{k+1} = \omega_k + \varepsilon_{\omega,k} \tag{4.4.9}$$

$$v_{k+1} = v_k + \varepsilon j_{\omega,k} \tag{4.4.10}$$

The discrete-time state vector is $\theta = [x_k y_k \eta_k \omega_k v_k]'$. It should be approximated by the value of the continuous-time state vector at the time of k-th distance measurement; call it t_k. Δt_k is the interval between the k-th and $(k+1)$-th distance measurement updates. The sequences $\{\varepsilon_{\omega,k}\}$ and $\{\varepsilon_{v,k}\}$ are fixed sequences of zero-averaged white Gaussian random variables, independent of each other. The first of these discrete-time series is:

$$E\left[\varepsilon_{\omega,k}\varepsilon_{\omega,j}\right] = Q_\omega \Delta t_k \delta_{kj} \tag{4.4.11}$$

Similarly, the covariance for $\{\varepsilon_{v,k}\}$ the procedure is:

$$E\left[\varepsilon_{v,k}\varepsilon_{v,j}\right] = Q_v \Delta t_k \delta_{kj} \tag{4.4.12}$$

This means that one can instead Eq. 9 and Eq. 10 by:

$$\omega_{k+1} = \omega_k + \gamma_{\omega,k}\sqrt{\Delta t_k} \tag{4.4.13}$$

$$v_{k+1} = v_k + \gamma_{v,k}\sqrt{\Delta t_k} \tag{4.4.14}$$

Where $\gamma_{\omega,k}$ and $\gamma_{v,k}$ are stationary, mutually independent zeroth mean white Gaussian sequence of random variables, The variance is Q_ω and Q_v. The advantage of this form is that it displays the dependence of discrete-time noise on the interval between successive updates.

Therefore, the discrete-time dynamic model of the UAV is:

$$\theta_k + 1 = \begin{pmatrix} 1 & 0 & 0 & 0 & \Delta t_k \cos\eta_k \\ 0 & 1 & 0 & 0 & \Delta t_k \sin\eta_k \\ 0 & 0 & 1 & \delta t_k & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \theta_k + \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ \sqrt{\Delta t_k} & 0 \\ 0 & \sqrt{\Delta t_k} \end{pmatrix} \begin{pmatrix} \gamma_{\omega,k} \\ \gamma_{v,k} \end{pmatrix} \tag{4.4.15}$$

The measurement equation is:

$$d_k = \sqrt{(x_k - x_0)^2 - (y_k - y_0)^2} + \varepsilon_{d,k} \tag{4.4.16}$$

Where x_0, y_0 are the coordinates received by k-th for the distance measurement, and $\{\varepsilon_{d,k}\}$ is a stationary zero-averaged white Gaussian sequence of random variables used to model the distance measurement error. The sequence is taken to be independent of $\{\varepsilon_{v,k}\}$ and $\{\varepsilon_{\omega,k}\}$. In real-world applications, receiving more than one distance measurement from multiple UAVs at around the same time is possible. In such a case, the measurement equation can be extended to include more than one equation, as in (6.3.16). In this thesis, we only consider the more general case where only one distance measurement is received at any given time.

Using the Extended Kalman Filter, an estimate of the UAV position $\{x_k, y_k\}$ at time step k can be obtained based on the non-linear dynamic equation shown in Eq. (4.4.15) and the non-linear measurement equation shown in Eq. (4.4.16). Both the dynamic and the measurement equations are linearised by retaining the first-order terms in the Taylor series expansion and neglecting the higher-order terms. The procedure of the EKF (Extended Kalman Filter) [49] [50] is shown in the following for completeness.

State prediction:

$$\hat{\theta}_{k+1|k} = f\left[\hat{\theta}_{k+1|k}\right] \tag{4.4.17}$$

Where f is defined in (6.3.15).

The state prediction covariance matrix is calculated as follows:

$$P_{k+1|k} = f_x\left[\hat{\theta}_{k|k}\right] P_{k|k} f_x'\left[\hat{\theta}_{k|k}\right] + U_k Q U_k' \tag{4.4.18}$$

where U_k is defined in Eq. (4.4.15) and $f_x[\hat{\theta}_{k|k}]$ is the Jacobian of f as evaluated at $\hat{\theta}_{k|k}$. $Q = diag\{Q_\omega\ Q_V\}$ is a covariance matrix of the error items $\gamma_{\omega,k}$ and $\gamma_{\omega,k}$, and is assumed to be known.

The predicted measured values are calculated as follows:

$$\hat{d}_{k+1|k} = h[\hat{\theta}_{k+1|k}] \tag{4.4.19}$$

Where the equation h is defined in Eq. (4.4.16).

The measurement prediction covariance was calculated as follows:

$$S_{k+1} = h_x[\hat{\theta}_{k+1|k}]P_{k+1|k}h'_x[\hat{\theta}_{k+1|k}] + R \tag{4.4.20}$$

$h_x[\hat{\theta}_{k+1|k}]$ is the Jacobian of h evaluated at $\hat{\theta}_{k+1|k}$. R is the variance of $\varepsilon_{d,k}$ and is assumed to be a known value.

The filtering gain is calculated as follows:

$$W_{k+1} = P_{k+1|k}h_{\hat{\theta}_{k+1|k}}S_{k+1}^{-1} \tag{4.4.21}$$

The status estimates are updated as follows:

$$\hat{\theta}_{k+1|k+1} = \hat{\theta}_{k+1|k} + W_{k+1}\lambda_{k+1} \tag{4.4.22}$$

Where λ_{k+1} is called the innovation and is as defined by:

$$\lambda_{k+1} = d_{k+1} - \hat{d}_{k+1|k} \tag{4.4.23}$$

The state covariance matrix is updated as follows:

$$\hat{P}_{k+1|k+1} = P_{k+1|k} - W_{k+1}S_{k+i}W'_{k+1} \tag{4.4.24}$$

The initial values $\hat{\theta}_{0|0}$, $P_{0|0}$ can be chosen from experience or the UAV's state before the GPS's loss. The influence of the initial values on the state estimate is usually at an exponential rate.

4.5 Summary

This chapter illustrates the essential methods used in this thesis by describing the control architecture of the UAV swarm used, the simulation software information and the positioning algorithm. The method is designed to provide more accurate positioning of the UAV swarm. The method was successfully run-in specific simulations and effectively provided automatic navigation for the UAV swarm. In chapter 5, the results and comparison of the operation of the method are shown.

5 Results

In this chapter, the results of the simulations are shown. The effectiveness of the overall scheme of this thesis is verified, i.e., a scheme using a combination of a distributed communication approach, a leader-follower control approach, and an extended Kalman filter localisation approach.

Most of the simulation results shown are the trajectory of a UAV randomly selected in a v-shaped formation in a straight path of 350 meters, to facilitate visual comparison of trajectories. Other preset paths are also shown at the end of the results. Each UAV is equipped with a GPS and an IMU sensor, and it is assumed that they generate information at the same frequency of 240 Hz. In practice, the GPS usually generates information at a slower frequency than the IMU, but since this thesis aims to simulate GPS loss, the same frequency is set here. Each UAV is also equipped with a simple intra-group communication device, i.e., neighbouring UAVs can communicate according to a distributed approach.

The results of the simulations presented in this chapter demonstrate the effectiveness of the control scheme proposed in this thesis in maintaining the formation of UAVs. The combination of distributed communication, leader-follower control, and EKF localization enables the UAVs to effectively follow a predefined path and maintain their positions relative to one another, even in the presence of GPS loss or high location noise. The results show that the EKF is able to reduce the positioning error and improve the formation's ability to follow the desired path, particularly when at least one UAV has not lost GPS or UWB capability. However, even when all UAVs have lost these capabilities, the EKF is still able to provide better performance than the dead reckoning method alone. The results also demonstrate the effectiveness of the proposed method in adapting to the loss of the leader UAV, with the formation able to smoothly transition to a new leader and continue following the desired path. The results for the different preset paths show that the proposed method is flexible and able to handle paths of varying complexity. Overall, the results of the simulations support the conclusion that the control scheme proposed in this thesis is effective in maintaining the formation of UAVs in a variety of scenarios and environments.

For the initial accuracy settings as shown in Table 5-1, the accuracy of GPS can be improved, but it is also affected by many factors, including the location and orientation of the antenna, the environment (e.g., buildings, forests, etc.), the quality and functionality of the receiver, and the orbit and signal of the GPS satellite. In general, the accuracy of GPS is around 5 meters. However, by using more advanced receivers, the accuracy of GPS can be improved to about 1 meter. The accuracy of an IMU depends on a variety of factors, including the quality, accuracy, and performance of the sensor, the quality and efficiency of the signal processing algorithm, and the characteristics of the environment in which it is used. In general, IMUs can achieve accuracies in the range of a few centimeters to tens of centimeters. High accuracy IMUs can achieve accuracies of several centimeters, but typically require the use of higher quality sensors. Compared with the actual accuracy, the following accuracy settings are for better simulation and are usually more ideal.

In summary, the following results are to demonstrate that in a GPS-denied environment, the use of EKF is effective in improving the positioning accuracy and reducing the error accumulation of dead reckoning for UAV formations with different preset paths and different positioning accuracies (the parameters are set as in Table 5-1). And the UAV formation can also effectively solve the situation of losing the leader UAV.

Data	***Units***	***Indoor***	***outdoor***
Roll, Pitch, Yaw	*deg*	[0.2 0.2 0.2]	[0.2 0.2 0.2]
Position	*m*	[0.15 0.15 0.15]	[1.0 1.0 1.0]
Velocity	*m/s*	0.02	0.05
Acceleration	m/s^2	0.02	0.05
Angular velocity	*deg/s*	0.02	0.05

Table 5-1: Positioning accuracy setting values for UWB and INS for indoor and outdoor conditions: based on [12]

5.1 Results of Indoor Simulation

Figure 5-2 and Figure 5-1 shows the preset trajectory (red) and the flight trajectory of the UAV in a coordinate system measured in meters with INS positioning (green), GPS/UWB positioning (blue) and IMU (yellow) combined with Dead Reckoning indoors. INS fuses its own GPS/UWB and IMU, so it usually has better accuracy. Compared to INS positioning, there is a more significant error when using only

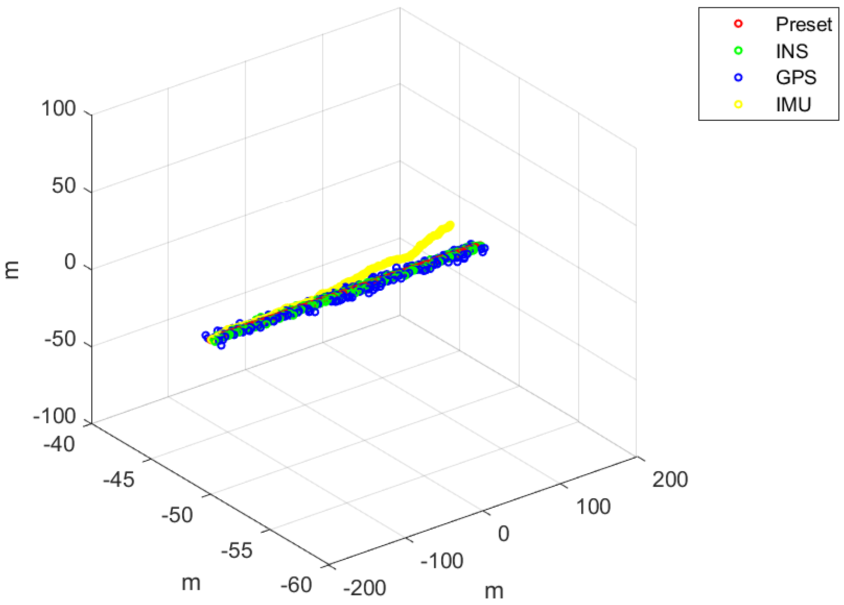

Figure 5-2: Preset true path, INS positioning, GPS positioning, IMU positioning of the trajectory (overhead view)

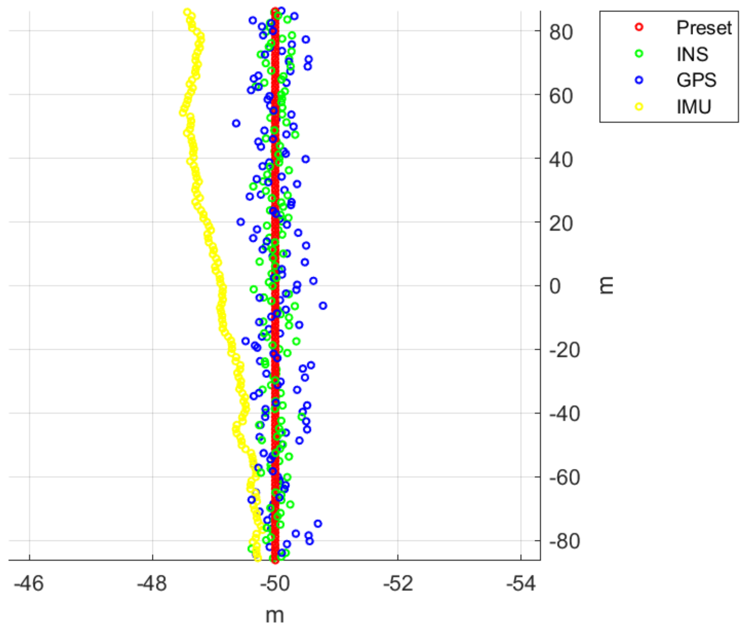

Figure 5-1 : Preset true path, INS positioning, GPS positioning, IMU positioning of the trajectory positioning

GPS/UWB or Dead Reckoning, and Dead Reckoning will accumulate the error over time.

Figure 5-3 and Figure 5-4 shows the indoor results of the UAV projecting its position with the help of GPS/UWB positions of other UAVs in the UAV cluster after losing GPS/UWB (blue), and the results show that the predicted results have relatively minor

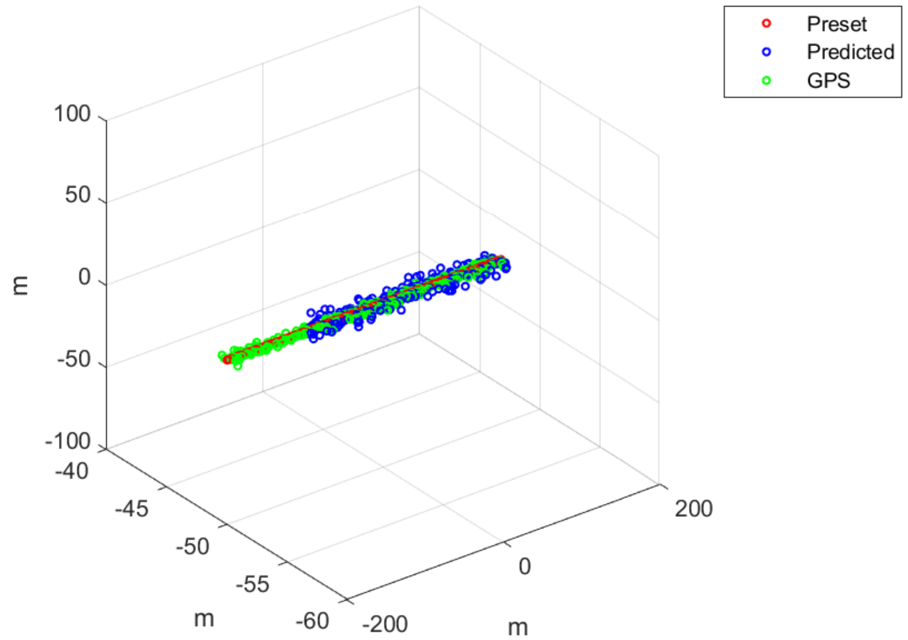

Figure 5-4 : Preset true path, predictive positioning, GPS positioning of the trajectory and its overhead view positioning

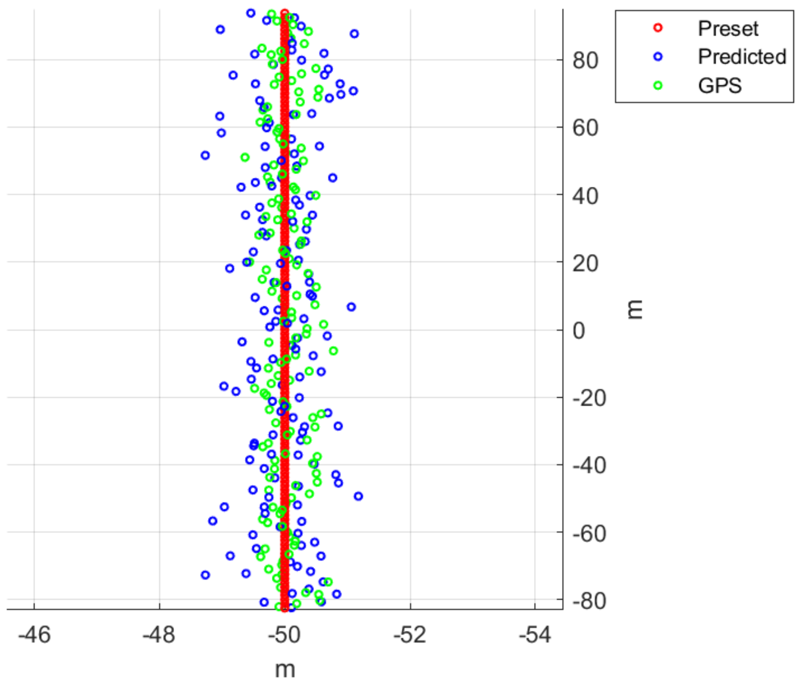

Figure 5-3 : Preset true path, predictive positioning, GPS positioning of the trajectory (overhead view)

deviations from the actual GPS/UWB position of the UAV (green), which is within the acceptable range.

Figure 5-5 and Figure 5-6 show the indoor results of fusion of the UAV trajectory predicted by IMU (yellow) and the UAV trajectory generated by the adjacent UAV

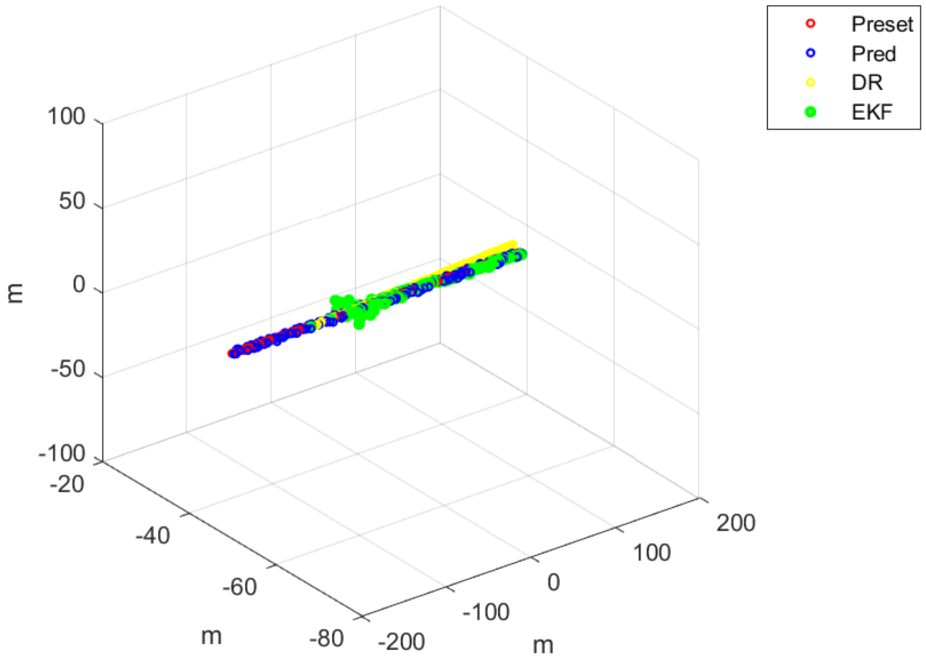

Figure 5-6 : Preset true path, predictive positioning by other UAVs, IMU positioning (DR), and fusion positioning by the predictive, IMU positioning

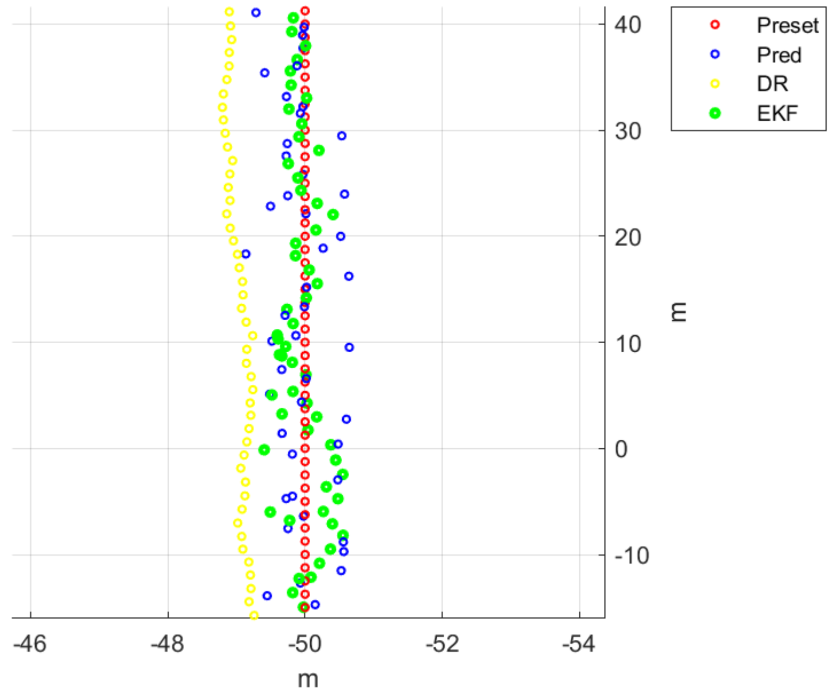

Figure 5-5: Preset true path, predictive positioning by other UAVs, IMU positioning (DR), and fusion positioning by the predictive, IMU positioning (overhead view)

GPS/UWB prediction (blue) using the Extended Kalman Filter (green) after the UAV lost the GPS/UWB. The results show that the fused trajectory is more accurate than both the original trajectories.

Figure 5-7 shows the indoor noise values for the set GPS/UWB for neighbour (yellow), IMU positioning (blue) and the EKF fusion methods (red). The average error in noise for the fusion method is reduced by approximately 30% and 60% on the x-axis compared to GPS/UWB and IMU, respectively. In the y-axis, the reduction is approximately 30% and 50% compared to GPS/UWB and IMU, respectively. In the z-axis, the reduction is approximately 30% and 30% compared to GPS/UWB and IMU, respectively.

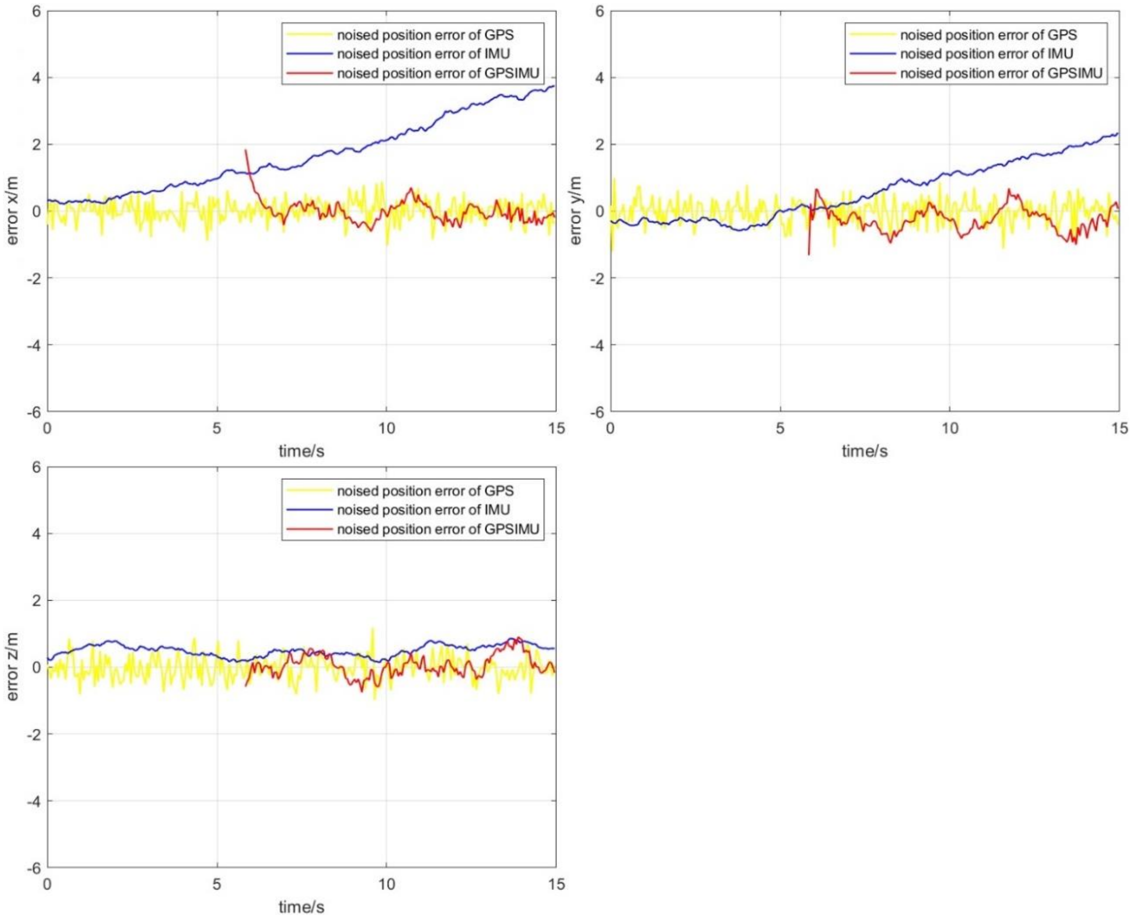

Figure 5-7: Indoor noised position error of GPS, IMU and their fusion (GPS-IMU)

Figure 5-8 shows the RMSE (root mean square error) between GPS positioning (red), IMU positioning (yellow), INS positioning (green) and the positioning generated by the EKF method used in this thesis (blue) after 500 simulations. The RMSD works by aggregating the magnitude of prediction errors from different data points into a single measure of predictive power. RMSD is a measure of accuracy used to compare the prediction errors of different models for a given data set rather than comparing data sets. The RMSE of EKF is relatively large in the initial stage, but it can quickly converge to a lower level, which is better than both IMU and GPS localisation. INS naturally performs better because it uses its own IMU and GPS fused together, avoiding the noise of relative position measurements.

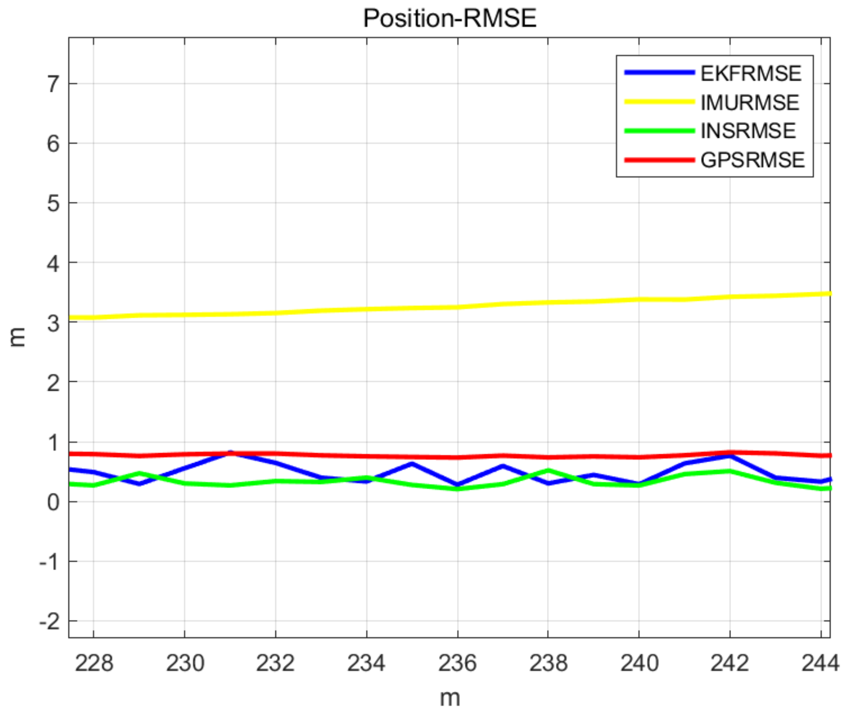

Figure 5-8: RMSE between GPS positioning, IMU positioning, INS positioning and the positioning generated by the method used in this thesis after 500 simulations (Indoor)

In addition to this, the simulation results for the indoor scenario, when the UAV swarm completely loses GPS/UWB, are shown in Figure 5-9.

In this case, the position information of the neighboring UAVs (blue) is no longer accurate. This is because the accuracy of different UAV IMUs is also susceptible to environmental effects, so the accumulation of navigation data errors varies from UAV to UAV. In half of the 500 simulations, the navigation data with better accuracy can also correct the data with poorer accuracy.

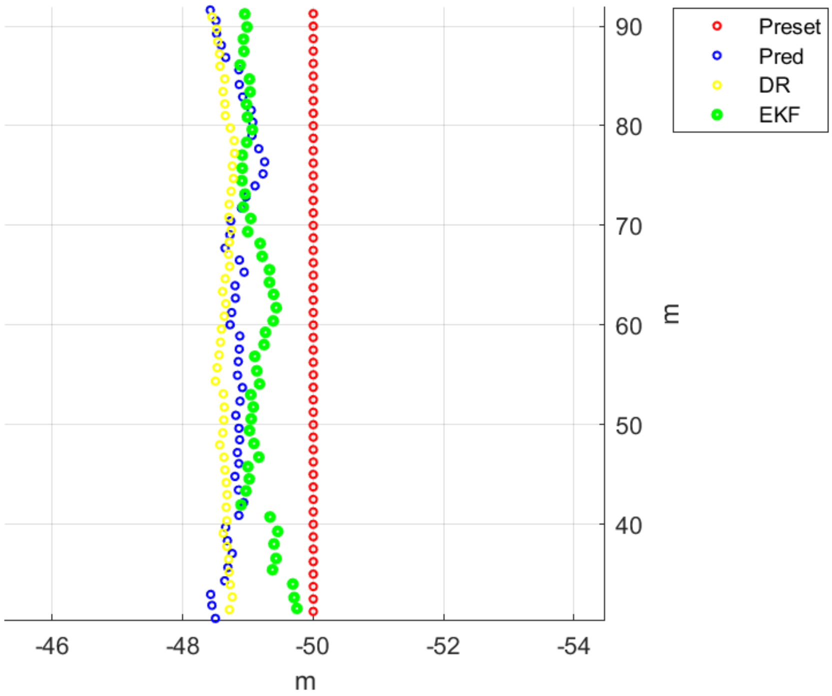

Figure 5-9: Preset true path, predictive positioning by other UAVs, IMU positioning, and fusion positioning by the predictive, IMU positioning when the UAV swarm completely loses GPS/UWB (overhead view)

In this thesis, the case of losing the leader of a UAV swarm is also simulated. As shown in Figure 5-10 and Figure 5-11, the situation is simulated in this thesis using trajectory 3 (Figure 5-13) and a v-shape formation. When UAV1 is removed, the formation will designate UAV2 as the new leader according to the pre-stored graph information, and the original UAV3 will change the following target to UAV2, while UAV4 and UAV5 follow the target unchanged.

The results show that after losing the leader of the whole team, the formation can effectively assign a new leader and complete the mission based on the pre-stored graph information.

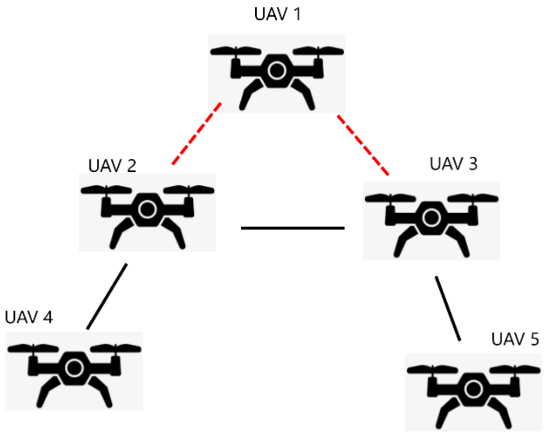

Figure 5-10: The swarm loses Leader and assigns a new leader

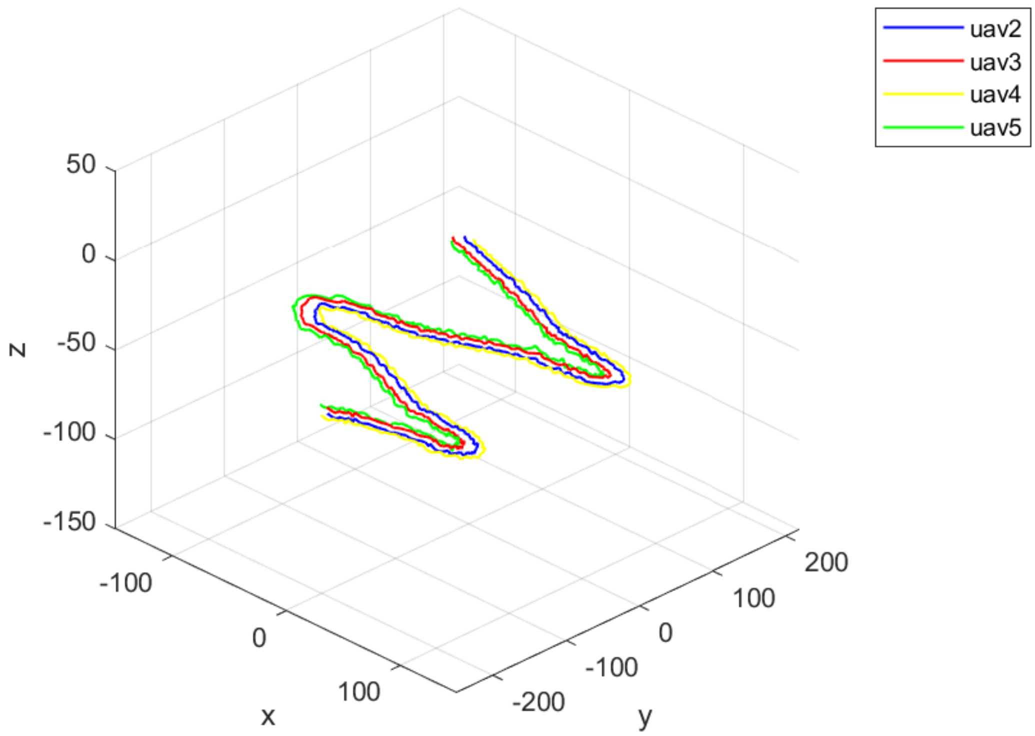

Figure 5-11: The swarm loses Leader and assigns a new leader with a v-shape formation

In the other, three kinds of paths are implemented in this thesis. In addition to the simple straight path shown above, the other two trajectories are simulated as shown in Figure 5-13 and Figure 5-12. The figures also show the simulation results of the indoor EKF with a v-shape formation and at least one drone with GPS/UWB. With the results of this simulation and the analysis of the theoretical noise, the formation can maintain the formation shape to fit various complex paths without colliding with each other, while keeping the distance between UAVs above 1.8 meters indoors and above 5 meters outdoors.

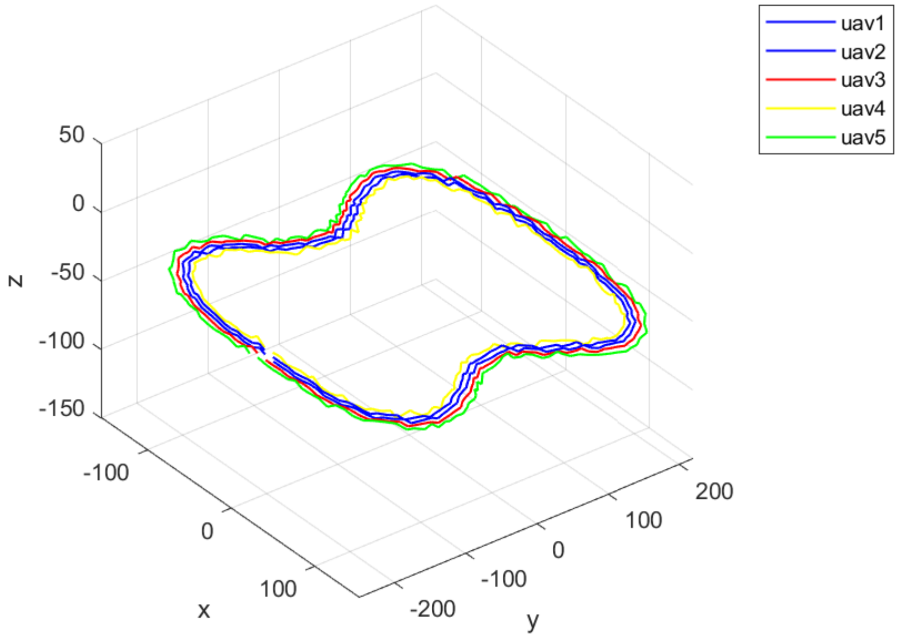

Figure 5-12: Trajectory 2

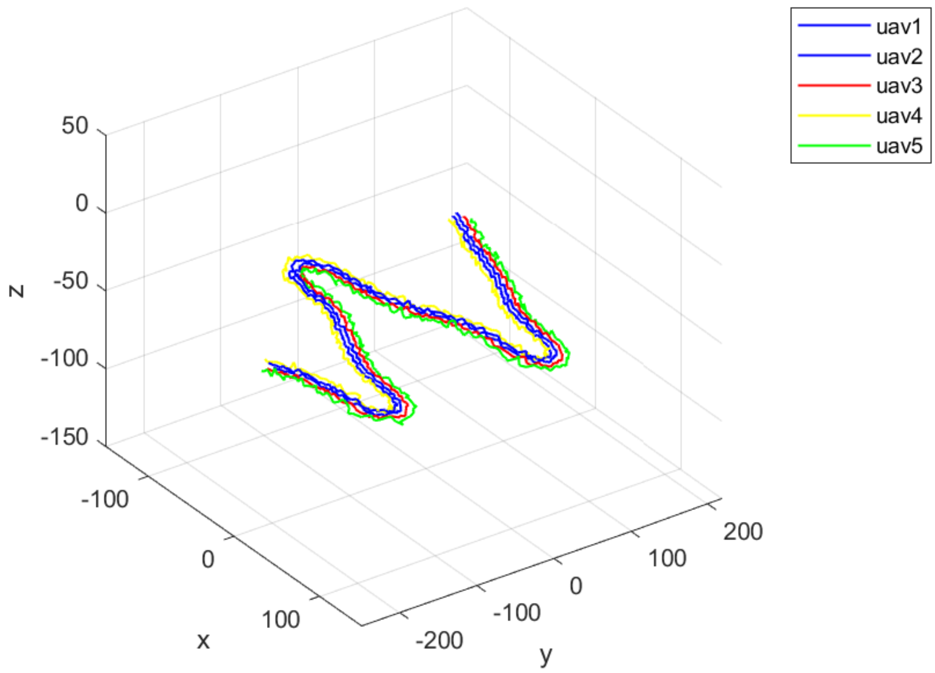

Figure 5-13: Trajectory 3

5.2 Results of Outdoor Simulation

Similar to the indoor case (Figure 5-6 and Figure 5-5), the simulation results of outdoor are shown in Figure 5-14 and Figure 5-15. It can be seen that the EKF can also effectively fuse GPS position data with higher noise levels.

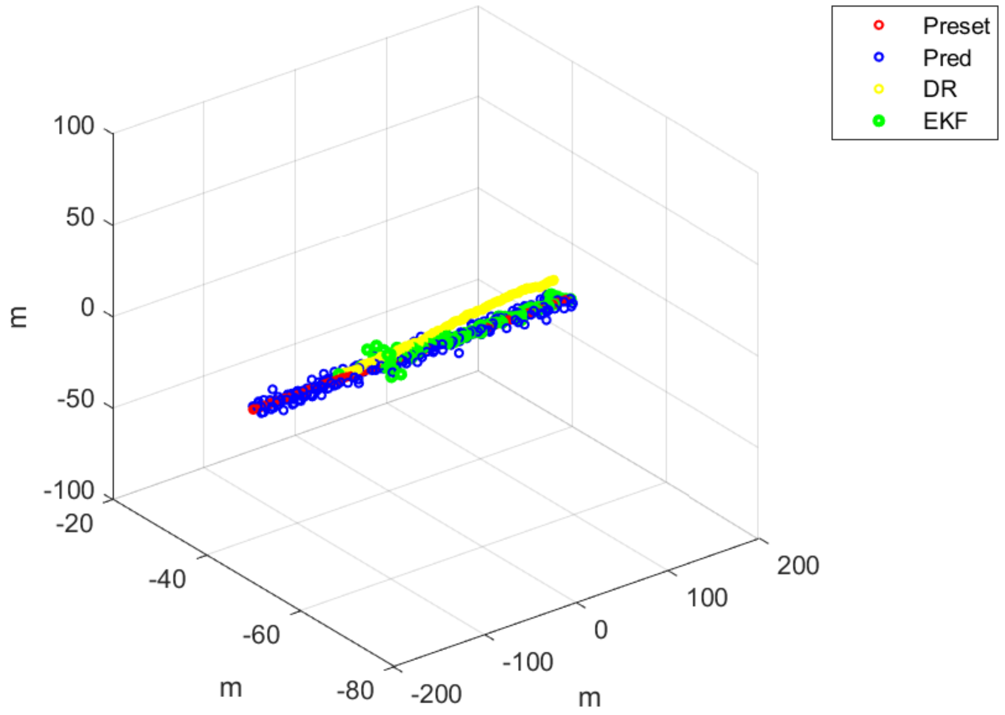

Figure 5-14: Preset true path, predictive positioning by other UAVs, IMU positioning, and fusion positioning by the predictive, IMU positioning (DR) in the outdoors

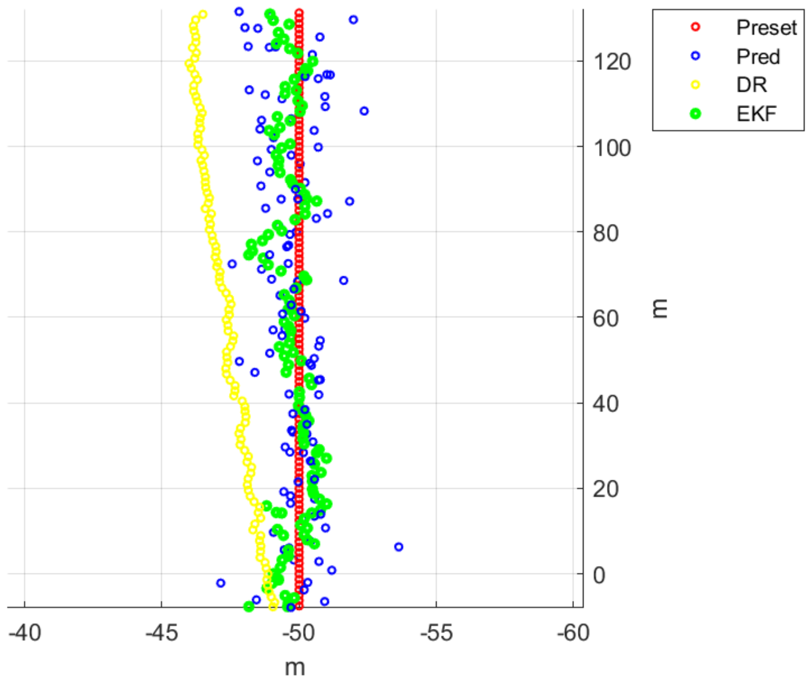

Figure 5-15: Preset true path, predictive positioning by other UAVs, IMU positioning, and fusion positioning by the predictive, IMU positioning (DR) in the outdoors (overhead view)

Figure 5-16 illustrates the outdoor noise values for the set GPS/UWB for the neighbor (yellow), IMU positioning (blue), and the EKF fusion method (red). On the x-axis, the average error in noise for the fusion method is reduced by approximately 40% and 50% compared to GPS/UWB and IMU, respectively. On the y-axis, the reduction is approximately 40% and 50% compared to GPS/UWB and IMU, respectively. On the z-axis, the reduction is approximately 30% and 50% compared to GPS/UWB and IMU, respectively.

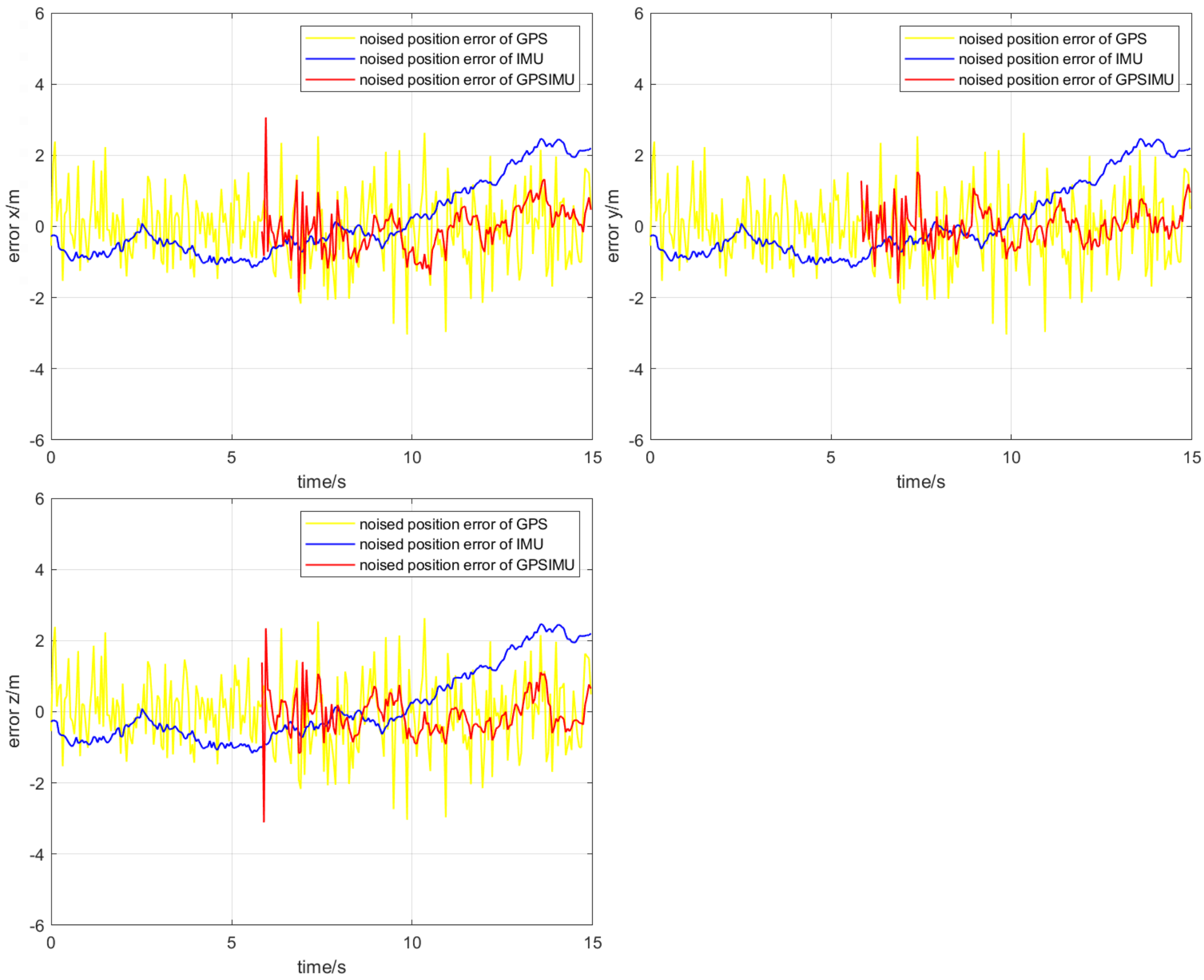

Figure 5-16: Outdoor noised position error of GPS, IMU and their fusion (GPS-IMU)

As shown in Figure 5-17, similar to indoor after 500 outdoor simulations, the RMSE between GPS localization (red), IMU localization (yellow), INS localization (green), and the localization produced by the EKF method (blue) was measured. The results indicate that the EKF method quickly converges to a lower level of error, outperforming both IMU and GPS positioning. INS performed the best, as it combines IMU and GPS measurements to reduce the impact of noise on relative position estimates.

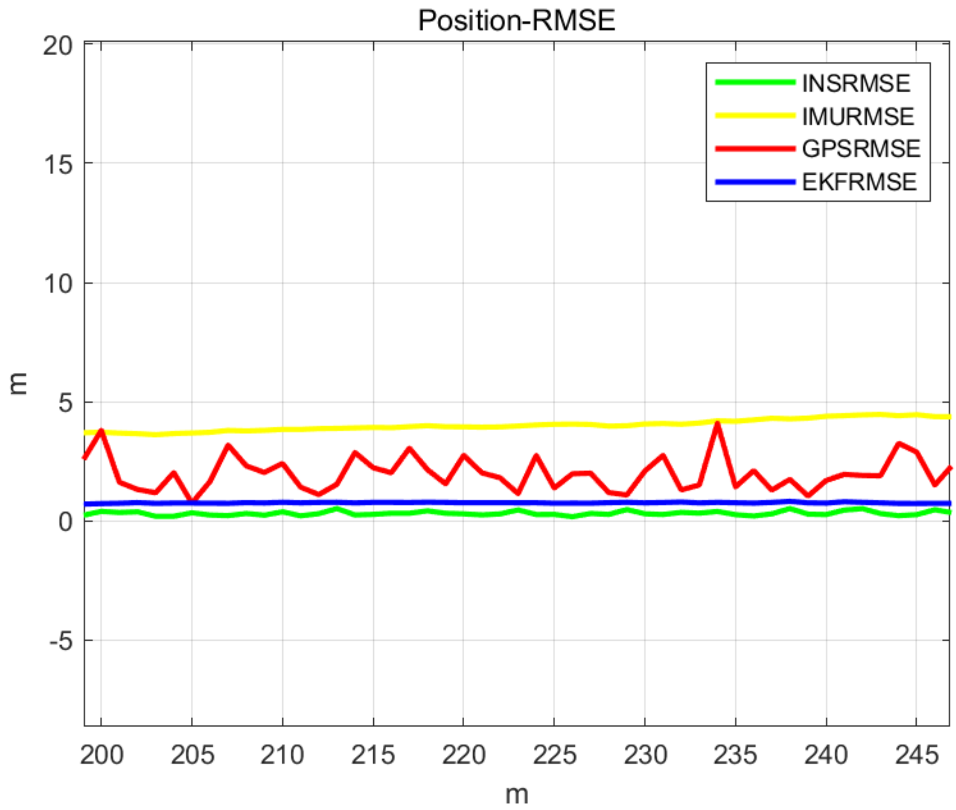

Figure 5-17: RMSE between GPS positioning, IMU positioning, INS positioning and the positioning generated by the method used in this thesis after 500 simulations (Outdoor)

6 Conclusion

In this thesis, a control scheme is proposed for the formation of UAVs. The scheme utilizes distributed communication, where each UAV communicates with neighboring UAVs and treats them as local leaders, and leader-follower control, where the leader's UAV stores a predetermined path and the other followers adjust their flight data according to the leader's position, direction, distance, speed, and angular velocity. The formation can operate in both indoor and outdoor environments, using ultra-wideband positioning, which typically has good positioning accuracy, in indoor situations and inertial navigation systems in outdoor situations. However, the accuracy of INS positioning can be low due to the limitations of GPS positioning accuracy and the effect of wind on the UAV. In this thesis, to improve the positioning accuracy indoors and outdoors, we propose that UAVs in formation can receive navigation information from other UAVs that have not lost GPS or UWB capability and use extended Kalman filtering in combination with data from their own inertial measurement units to effectively reduce positioning errors in GPS-denied environments. The final simulation results demonstrate that this scheme can be very effective in reducing the error accumulation caused by its own DR positioning and improving the positioning accuracy when at least one UAV in the UAV fleet has not lost GPS/UWB navigation information. Even in the case that the UAV formation completely loses GPS or UWB, the method in this thesis can improve the positioning accuracy in more than half of the cases. This thesis also conducts simulations using predefined paths of different complexity and different formation shapes to demonstrate the effectiveness of the method.

7 Future Considerations

In this thesis, the indoor environment is localized by measuring the distance data from the anchor point using UWB. In the outdoor environment, the positioning is done by GPS. Based on these data, we can simulate the loss of GPS/UWB connection and the acquisition of distance measurements between UAVs by assume a specific time series of synthetic GPS interruption intervals. During these intervals, the distances between UAVs are synthesized at discrete moments. To generate synthetic inter-UAV distances, actual GPS or distance measurements from an anchor point are taken and the corresponding inter-UAV distances are calculated. Gaussian random variables with zero mean and standard deviation are then added to these values, and the resulting values are passed to the algorithm as the synthetic inter-UAV distances. The Kalman Filter is then run using this data. However, this approach can be affected by the accuracy of the actual ranging measurements.

In future research, it will be important to incorporate more realistic wind fields and noise into the simulations. Since there are numerous parameters to be adjusted in flight control, it is necessary to verify the effectiveness of these inaccurate flights through additional experiments. These experiments will also help to evaluate the optimality and robustness of the EKF. In addition, the effectiveness of good obstacle avoidance algorithms as a swarm solution for UAVs should be further investigated. Finally, the method's effectiveness could be improved if low-cost range and orientation sensors were available.

In addition to this, it is important to consider the limitations of the proposed control scheme. One potential limitation is the assumption of a fixed communication range between UAVs, which may not hold in all scenarios. Another limitation is the assumption of a fixed communication topology, which may change in dynamic environments. Additionally, the proposed method relies on the accuracy of the IMU and EKF for localization, which may be affected by biases or other errors. It is also important to consider the impact of external factors, such as wind or other disturbances, on the formation's ability to follow the desired path. Future work could address these limitations by incorporating more sophisticated communication and localization methods, as well as by considering the impact of external factors on the formation's performance. Overall, the proposed control scheme shows promise as a means of maintaining the formation of UAVs, but further research is needed to fully understand its capabilities and limitations.

Apart from this, the implementation of the algorithm into hardware is also necessary to validate the effectiveness of the solution. This requires the use of MATLAB Simulink to redesign the model. It can automatically generate C, C++, etc. code and deploy it directly to the embedded system [51] [39].

References

[1] S. Waharte, N. Trigoni, and S. Julier, “Coordinated Search with a Swarm of UAVs,” in *2009 6th IEEE Annual Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks Workshops*, Rome, Italy, Jun. 2009, pp. 1–3. doi: 10.1109/SAHCNW.2009.5172925.

[2] B. Shirani, M. Najafi, and I. Izadi, “Cooperative load transportation using multiple UAVs,” *Aerospace Science and Technology*, vol. 84, pp. 158–169, Jan. 2019, doi: 10.1016/j.ast.2018.10.027.

[3] D. W. Casbeer, Sai-Ming Li, R. W. Beard, R. K. Mehra, and T. W. McLain, “Forest fire monitoring with multiple small UAVs,” in *Proceedings of the 2005, American Control Conference, 2005.*, Portland, OR, USA, 2005, pp. 3530–3535. doi: 10.1109/ACC.2005.1470520.

[4] D. H. A. Maithripala and S. Jayasuriya, “Radar deception through phantom track generation,” in *Proceedings of the 2005, American Control Conference, 2005.*, Portland, OR, USA, 2005, pp. 4102–4106. doi: 10.1109/ACC.2005.1470620.

[5] X. Wang *et al.*, “Coordinated flight control of miniature fixed-wing UAV swarms: methods and experiments,” *Sci. China Inf. Sci.*, vol. 62, no. 11, p. 212204, Nov. 2019, doi: 10.1007/s11432-018-9887-5.

[6] J. P. Desai, J. Ostrowski, and V. Kumar, “Controlling formations of multiple mobile robots,” in *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, Leuven, Belgium, 1998, vol. 4, pp. 2864–2869. doi: 10.1109/ROBOT.1998.680621.

[7] X. Fu, J. Pan, H. Wang, and X. Gao, “A formation maintenance and reconstruction method of UAV swarm based on distributed control,” *Aerospace Science and Technology*, vol. 104, p. 105981, Sep. 2020, doi: 10.1016/j.ast.2020.105981.

[8] P. Wang, Z. Man, Z. Cao, J. Zheng, and Y. Zhao, “Dynamics modelling and linear control of quadcopter,” in *2016 International Conference on Advanced Mechatronic Systems (ICAMechS)*, Melbourne, Australia, Nov. 2016, pp. 498–503. doi: 10.1109/ICAMechS.2016.7813499.

[9] Z. Jia, M. Hamer, and R. D’Andrea, “Distributed Estimation, Control and Coordination of Quadcopter Swarm Robots.” arXiv, Feb. 14, 2021. Accessed: Dec. 28, 2022. [Online]. Available: http://arxiv.org/abs/2102.07107

[10] N. Ahmad, R. A. R. Ghazilla, N. M. Khairi, and V. Kasi, “Reviews on Various Inertial Measurement Unit (IMU) Sensor Applications,” *IJSPS*, pp. 256–262, 2013, doi: 10.12720/ijsps.1.2.256-262.

[11] “Model IMU, GPS, and INS/GPS - MATLAB & Simulink - MathWorks China.” https://ww2.mathworks.cn/help/fusion/gs/model-imu-gps-and-insgps.html?#mw_1cddef41-123c-40dd-9f52-60b87914592c (accessed Dec. 28, 2022).

[12] “IMU simulation model - MATLAB - MathWorks China.” https://ww2.mathworks.cn/help/nav/ref/imusensor-system-object.html (accessed Dec. 28, 2022).

[13] E. D. Kaplan and C. Hegarty, *Understanding GPS/GNSS: Principles and Applications, Third Edition*. Artech House, 2017.

[14] K. Li, C. Wang, S. Huang, G. Liang, X. Wu, and Y. Liao, “Self-positioning for UAV indoor navigation based on 3D laser scanner, UWB and INS,” in *2016 IEEE International Conference on Information and Automation (ICIA)*, Ningbo, China, Aug. 2016, pp. 498–503. doi: 10.1109/ICInfA.2016.7831874.

[15] G. Zhang and L.-T. Hsu, "Intelligent GNSS/INS integrated navigation system for a commercial UAV flight control system," *Aerospace Science and Technology*, vol. 80, pp. 368–380, Sep. 2018, doi: 10.1016/j.ast.2018.07.026.
[16] "Learn how a GNSS-Aided Inertial Navigation System (GNSS/INS) works · VectorNav." https://www.vectornav.com/resources/inertial-navigation-primer/theory-of-operation/theory-gpsins (accessed Dec. 28, 2022).
[17] Y.-H. Jin, K.-W. Ko, and W.-H. Lee, "An Indoor Location-Based Positioning System Using Stereo Vision with the Drone Camera," *Mobile Information Systems*, vol. 2018, pp. 1–13, Oct. 2018, doi: 10.1155/2018/5160543.
[18] Z. Lin, M. Fu, and Y. Diao, "Distributed Self Localization for Relative Position Sensing Networks in 2D Space," *IEEE Trans. Signal Process.*, vol. 63, no. 14, pp. 3751–3761, Jul. 2015, doi: 10.1109/TSP.2015.2432739.
[19] J. Tiemann, F. Schweikowski, and C. Wietfeld, "Design of an UWB indoor-positioning system for UAV navigation in GNSS-denied environments," in *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Banff, AB, Canada, Oct. 2015, pp. 1–7. doi: 10.1109/IPIN.2015.7346960.
[20] H. Do *et al.*, "Formation Control Algorithms for Multiple-UAVs: A Comprehensive Survey," *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems*, vol. 8, no. 27, p. 170230, Jun. 2021, doi: 10.4108/eai.10-6-2021.170230.
[21] A. L. Salih, M. Moghavvemi, H. A. F. Mohamed, and K. S. Gaeid, "Flight PID controller design for a UAV quadrotor".
[22] C. Vasudev, *Graph Theory with Applications*. New Age International, 2006.
[23] D. A. Mercado, R. Castro, and R. Lozano, "Quadrotors flight formation control using a leader-follower approach," in *2013 European Control Conference (ECC)*, Zurich, Jul. 2013, pp. 3858–3863. doi: 10.23919/ECC.2013.6669637.
[24] A. M. de Souza Neto and R. A. F. Romero, "A Decentralized Approach to Drone Formation Based on Leader-Follower Technique," in *2019 Latin American Robotics Symposium (LARS), 2019 Brazilian Symposium on Robotics (SBR) and 2019 Workshop on Robotics in Education (WRE)*, Rio Grande, Brazil, Oct. 2019, pp. 358–362. doi: 10.1109/LARS-SBR-WRE48964.2019.00069.
[25] K.-K. Oh, M.-C. Park, and H.-S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440, Mar. 2015, doi: 10.1016/j.automatica.2014.10.022.
[26] Z. A. Ali, A. Israr, E. H. Alkhammash, and M. Hadjouni, "A Leader-Follower Formation Control of Multi-UAVs via an Adaptive Hybrid Controller," *Complexity*, vol. 2021, pp. 1–16, Nov. 2021, doi: 10.1155/2021/9231636.
[27] R. Azoulay, Y. Haddad, and S. Reches, "Machine Learning Methods for UAV Flocks Management-A Survey," *IEEE Access*, vol. 9, pp. 139146–139175, 2021, doi: 10.1109/ACCESS.2021.3117451.
[28] S. Ragi and E. K. P. Chong, "UAV Path Planning in a Dynamic Environment via Partially Observable Markov Decision Process," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 49, no. 4, pp. 2397–2412, Oct. 2013, doi: 10.1109/TAES.2013.6621824.
[29] S. A. P. Quintero, G. E. Collins, and J. P. Hespanha, "Flocking with fixed-wing UAVs for distributed sensing: A stochastic optimal control approach," in *2013 American Control Conference*, Washington, DC, Jun. 2013, pp. 2025–2031. doi: 10.1109/ACC.2013.6580133.
[30] S.-M. Hung and S. N. Givigi, "A Q-Learning Approach to Flocking With UAVs in a Stochastic Environment," *IEEE Trans. Cybern.*, vol. 47, no. 1, pp. 186–197, Jan. 2017, doi: 10.1109/TCYB.2015.2509646.
[31] J. Peng and R. J. Williams, "Incremental Multi-Step Q-Learning," in *Machine*

Learning Proceedings 1994, W. W. Cohen and H. Hirsh, Eds. San Francisco (CA): Morgan Kaufmann, 1994, pp. 226–232. doi: 10.1016/B978-1-55860-335-6.50035-0.
[32] Y.-H. Tsai, "Vision-Based Collision Avoidance for Unmanned Aerial Vehicles by Recurrent Neural Networks," *International Journal of Computer and Information Engineering*, vol. 13, no. 4, pp. 196–200, Mar. 2019.
[33] Z. Wu, J. Li, J. Zuo, and S. Li, "Path Planning of UAVs Based on Collision Probability and Kalman Filter," *IEEE Access*, vol. 6, pp. 34237–34245, 2018, doi: 10.1109/ACCESS.2018.2817648.
[34] N. Wang, Z. Li, X. Liang, Y. Li, and F. Zhao, "Cooperative Target Search of UAV Swarm with Communication Distance Constraint," *Mathematical Problems in Engineering*, vol. 2021, pp. 1–14, Sep. 2021, doi: 10.1155/2021/3794329.
[35] "UAV Attitude, Heading, and Wind Estimation Using GPS/INS and an Air Data System," *Guidance, Navigation, and Control and Co-located Conferences*. https://arc.aiaa.org/doi/10.2514/6.2013-5201 (accessed Dec. 28, 2022).
[36] X. Qi, J. Li, and W. Dong, "A state estimator of UAV using time-delayed position and IMU data," in *The 4th International Conference on Electronics, Communications and Control Engineering*, Seoul Republic of Korea, Apr. 2021, pp. 105–110. doi: 10.1145/3462676.3462693.
[37] Z. Han, K. Guo, L. Xie, and Z. Lin, "Integrated Relative Localization and Leader–Follower Formation Control," *IEEE Trans. Automat. Contr.*, vol. 64, no. 1, pp. 20–34, Jan. 2019, doi: 10.1109/TAC.2018.2800790.
[38] K. Guo *et al.*, "Ultra-Wideband-Based Localization for Quadcopter Navigation," *Un. Sys.*, vol. 04, no. 01, pp. 23–34, Jan. 2016, doi: 10.1142/S2301385016400033.
[39] J. Antoniou and A. Grabowski, "Autonomous UAV Development and Evaluation with MATLAB and Simulink".
[40] X. Liu and S. Xu, "Multi-UAV Cooperative Navigation Algorithm Based on Federated Filtering Structure," in *2018 IEEE CSAA Guidance, Navigation and Control Conference (CGNCC)*, Xiamen, China, Aug. 2018, pp. 1–5. doi: 10.1109/GNCC42960.2018.9018684.
[41] M. Coppola, K. N. McGuire, K. Y. W. Scheper, and G. C. H. E. de Croon, "On-board communication-based relative localization for collision avoidance in Micro Air Vehicle teams," *Auton Robot*, vol. 42, no. 8, pp. 1787–1805, Dec. 2018, doi: 10.1007/s10514-018-9760-3.
[42] C. Lin, Z. Lin, R. Zheng, G. Yan, and G. Mao, "Distributed Source Localization of Multi-Agent Systems With Bearing Angle Measurements," *IEEE Trans. Automat. Contr.*, vol. 61, no. 4, pp. 1105–1110, Apr. 2016, doi: 10.1109/TAC.2015.2457112.
[43] F. Schilling, E. Soria, and D. Floreano, "On the Scalability of Vision-Based Drone Swarms in the Presence of Occlusions," *IEEE Access*, vol. 10, pp. 28133–28146, 2022, doi: 10.1109/ACCESS.2022.3158758.
[44] M. Saska *et al.*, "System for deployment of groups of unmanned micro aerial vehicles in GPS-denied environments using onboard visual relative localization," *Auton Robot*, vol. 41, no. 4, pp. 919–944, Apr. 2017, doi: 10.1007/s10514-016-9567-z.
[45] J. R. Lockspeiser, M. L. Don, and M. Hamaoui, "Radio Frequency Ranging for Swarm Relative Localization," 2017.
[46] Q.-L. Zhou, Y. Zhang, Y.-H. Qu, and C.-A. Rabbath, "Dead reckoning and Kalman filter design for trajectory tracking of a quadrotor UAV," in *Proceedings of 2010 IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications*, QingDao, China, Jul. 2010, pp. 119–124. doi:

10.1109/MESA.2010.5552088.
[47] “UAV Toolbox Documentation - MathWorks China.” https://ww2.mathworks.cn/help/uav/index.html (accessed Dec. 28, 2022).
[48] G. Mao, S. Drake, and B. D. O. Anderson, “Design of an Extended Kalman Filter for UAV Localization,” in *2007 Information, Decision and Control*, Adelaide, Australia, Feb. 2007, pp. 224–229. doi: 10.1109/IDC.2007.374554.
[49] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Courier Corporation, 2012.
[50] K. A. Ghamry and Y. Zhang, “Formation control of multiple quadrotors based on leader-follower method,” in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, Denver, CO, USA, Jun. 2015, pp. 1037–1042. doi: 10.1109/ICUAS.2015.7152394.
[51] “UAV Toolbox Support Package for PX4 Autopilots Documentation - MathWorks China.” https://ww2.mathworks.cn/help/supportpkg/px4/index.html (accessed Dec. 28, 2022).

References of Professorship of Computer Engineering

[52] U. Tudevdagva, B. Battseren, W. Hardt, S. Blokzyl, and M. Lippmann, “UAV-based Fully Automated Inspection System for High Voltage Trans mission Lines.”

[53] B. Battseren, U. Tudevdagva, and W. Hardt, “A Finite State Machine Ba sed Adaptive Mission Control of Mini Aerial Vehicle,” *Embedded Selforg anizing Systems (ESS)*, vol. 5, no. 1, pp. 14–18, 2018, doi: 10.14464/es s51225.

[54] H. Ariane, H. Reda, B. Battseren, and W. Hardt, “AREIOM: Adaptive Re search Multicopter Platform,” *IFOST*, pp. 219–223, 2019.

[55] M. Stephan, B. Battseren, and U. Tudevdagva, “Autonomous Unmanne d Aerial Vehicle Development: MAVLink Abstraction Layer,” *Internationa l Symposium on Computer Science, Computer Engineering and Educat ional Technology*, vol. 10, pp. 45–49, 2020, [Online]. Available: https://w ww.ibs-laubusch.de/en/publications/.

Appendix

In this appendix, the structure of the software is described in more detail. Although some implementation details are described in chapter 4, which includes the specific software structure and usage, the specific folders are described here.

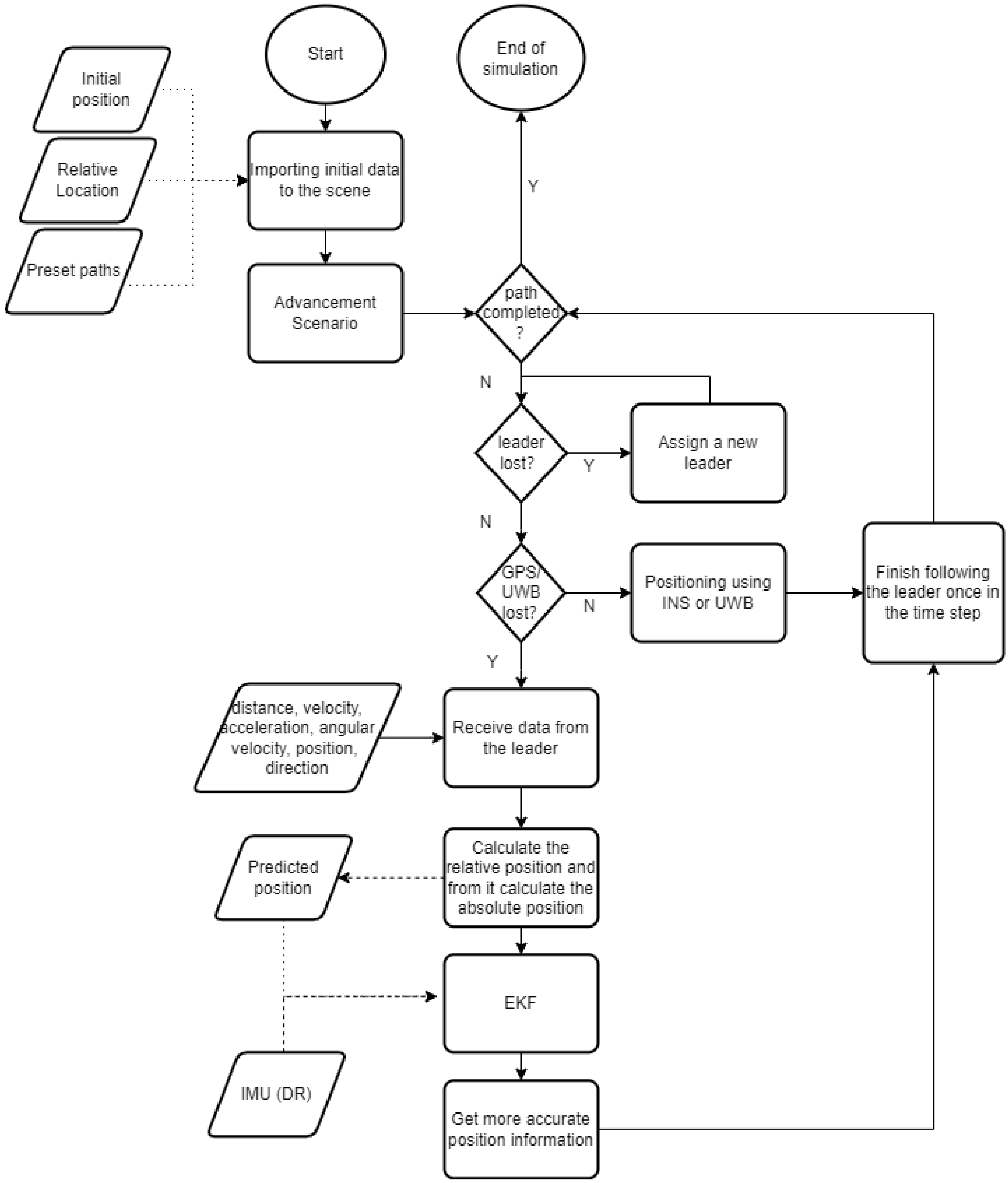

Figure 1: Flowchart of the software

The basic running of the program will start in the root directory from:

`runDistributedLeaderFollowerIndoor.m`,
`runDistributedLeaderFollowerOutdoor.m`,
`runDistributedLeaderFollowerLosingLeaderIntdoor.m`
`runDistributedLeaderFollowerLosingLeaderOutdoor.m`

Although the four startup scenarios are different, the basic process is shown in Figure 1.

First the initial and relative positions of the formation can be viewed in `formation` folder. Then in `run.m` the preset paths can be selected. The flight speed of the drone formation is determined by the number of path points imported. Then in `run.m` the initial GNSS/UWB positioning accuracy can be also defined. And import them into the scenario. After advancing one time step, the scenario determines whether the preset path is complete, and if so, ends the simulation; if not, determines whether the global leader is lost; if so, assigns a new leader; if not, each follower drone determines whether it has lost GPS/UWB; if not, uses INS/UWB positioning, then completes one time step, and then If GPS/UWB is lost, it accepts distance, velocity, acceleration, angular velocity, position and direction information from neighboring leaders. The absolute position of itself is calculated from this information. The EKF then fuses this information with the DR information to obtain more accurate position information. The position information is then used to follow the neighboring leader once. Then it returns to determine if the path is complete.

In the following sections, the specific function and content of each folder will be described in detail.

- ## Control

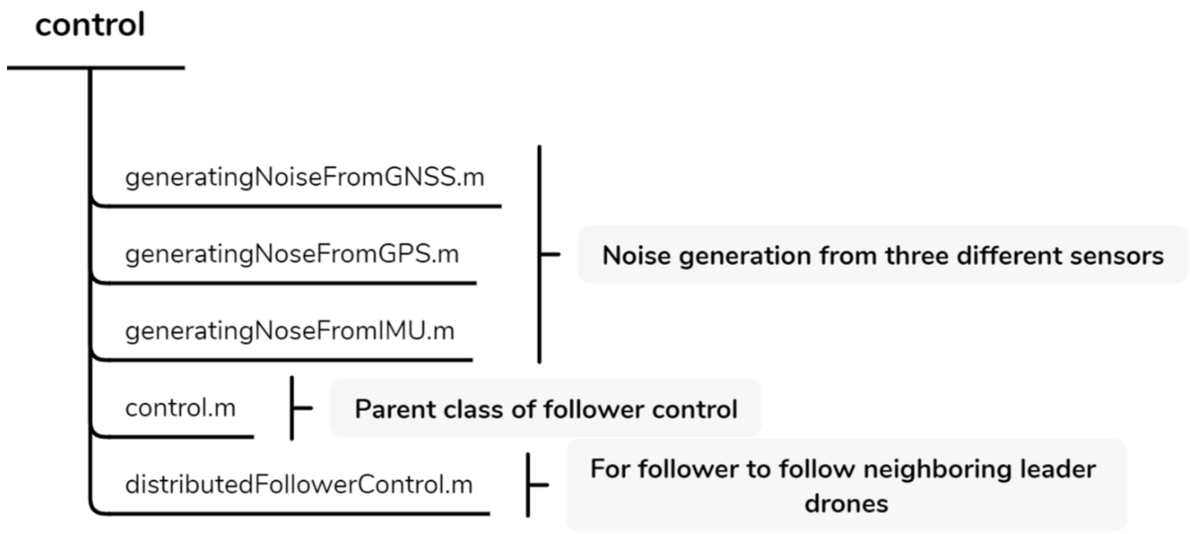

Figure 1: Structure of the folder control

In the folder `control`, the control system of follower is implemented. In the formation, except for the leader of the whole team which stores the preset trajectory, all other UAVs are controlled directly or indirectly according to the flight data of the leader at the current moment. Therefore, the control modes of leader and follower are somewhat different.

Three methods are implemented in the files `generatingNoiseFromGNSS.m`, `generatingNoiseFromGPS.m` and `generatingNoiseFromIMU.m`, which set the base parameters for GNSS (INS), GPS and IMU sensors, respectively. Each UAV will use all three sensors to generate data. The next `control.m` is a parent class and `distributedFollowerControl.m` is a subclass of `control.m`. In distributedFollowerControl.m, the function `folTruePos = followerMove(obj)` uses the `move` function of the UAV Toolbox to achieve follower following control at a time step. And return the true position for recording.

- ## Environment

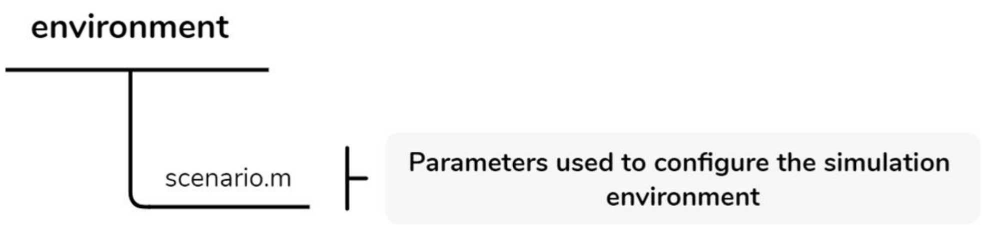

Figure 2: Structure of the folder environment

In the folder `environment`, the flight environment was simulated mainly using Matlab UAV Toolbox. In the file scenario.m parameters such as update rate, initial frame, preset trajectory, initial position, etc can be set. Funktions `uavScenario`, `waypointTrajectory`, `updateMesh` and `uavPlatform` are used to initialize the scene, set the cruise path, update the scene and initialize the UAV, respectively. For the user, this class can be used to import different leader flight paths and formation shapes before running.

- **Formation**

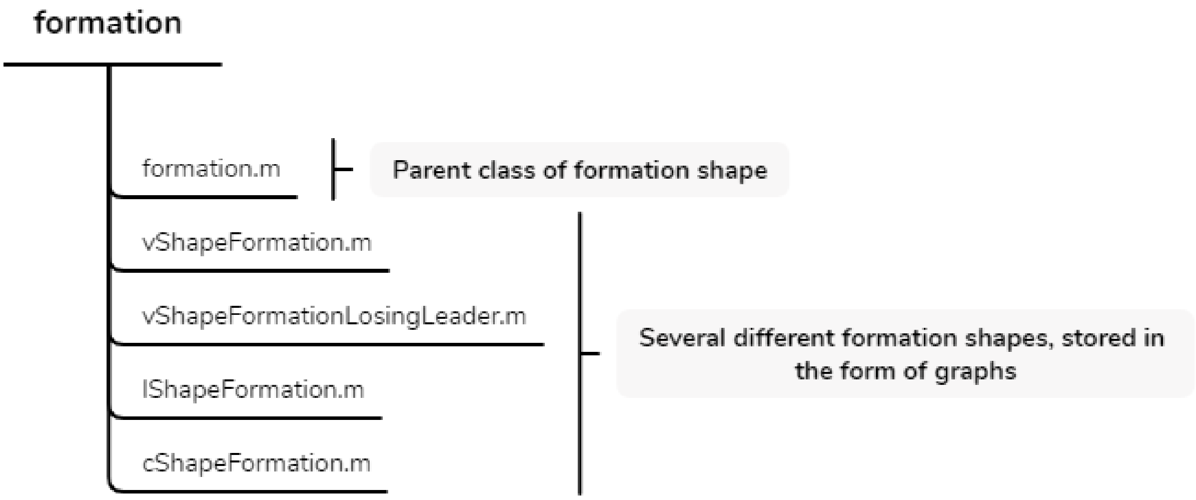

Figure 3: Structure of the folder formation

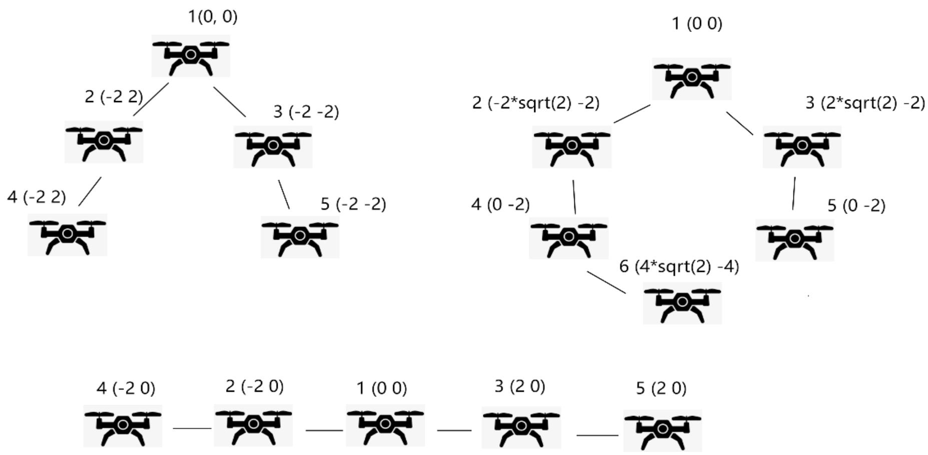

Figure 4: Three simple formation settings

In the folder `formation`, the different formation shapes and maintain the same flight altitude are defined using the form of the graph. Where `formation.m` is a parent class containing the parameters number of drones, initial position, relative position, and graph. The other files define the parameters of the different formations separately. They are v-shape, v-shape without leader, linear-shape and c-shape. In each formation, first the number of formation drones is defined. The default coordinates of the leader are (0, 0, -50). Then the relative position relationship between the drones is defined using the graph. For example, when s = [1 1 2 3], t = [2 3 4 5], it means that drone #2 and drone #3 follow drone #1, drone #4 follows drone #2, and drone #5 follows drone #3. The relative position definition is also relative to the leader of each drone. As shown in Figure 4, they are v-shape, linear-shape, and c-shape. Their relative positions have the same altitude, so a two-dimensional coordinate representation is used.

- **Method**

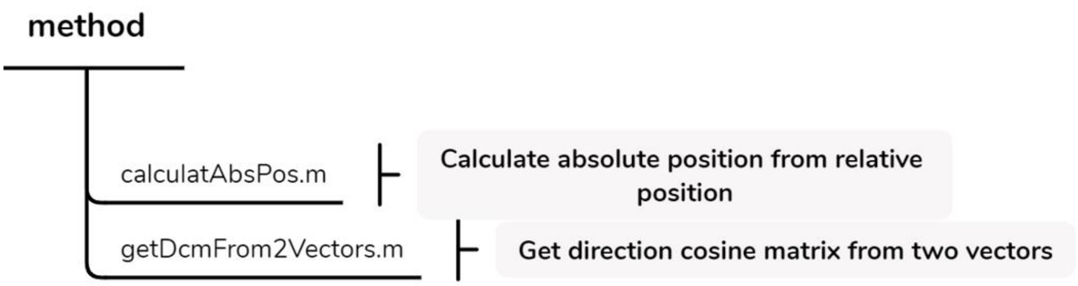

Figure 5: Structure of the folder method

In the folder `method`, two simple mathematical methods are implemented. `calculatAbsPos.m` implements the calculation of the absolute position of a neighboring drone from its own absolute position and the relative position of the neighboring drone. The specific method is described in Chapter 2.2.1. And the method `getDcmFrom2Vectors.m` calculates direction cosine matrix from two vectors. Its ultimate purpose is also to calculate the relative position.

- ## Positioning

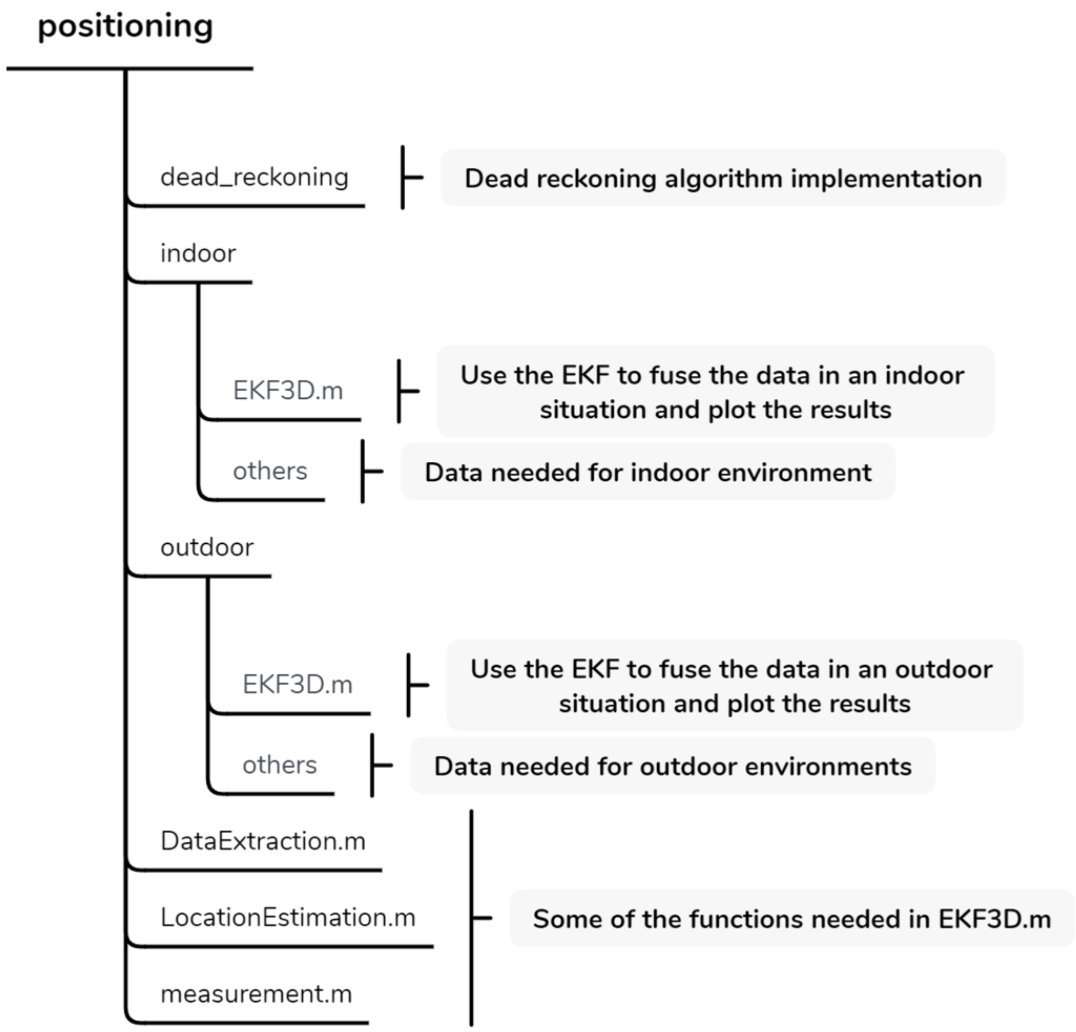

Figure 6: Structure of the folder positioning

The folder `positioning` implements the EKF positioning algorithm at the core of this thesis. That is, after the UAV loses GPS/UWB communication during flight, the EKF is used to fuse the positioning data of neighboring UAVs and its own IMU data to improve the positioning accuracy.

First, a simple DR algorithm is implemented in the folder `dead_reckoning`. That is, the position, velocity and acceleration from the IMU sensor at the previous moment are used to derive the current position. In the folder `indoor`, `EKF3D.m` implements the fusion of indoor position data. `EKF3D.m` is a runnable program that shows a comparison of the simulation results after running. Some of these results are shown in Chapter 5. Similarly, running `EKF3D.m` in the folder `outdoor` shows the results of the simulation with lower accuracy of the raw position data.

The other functions `DataExtraction.m`, `LocationEstimation.m` and `measurement.m` are required in EKF3D.m.

- **Simdata**

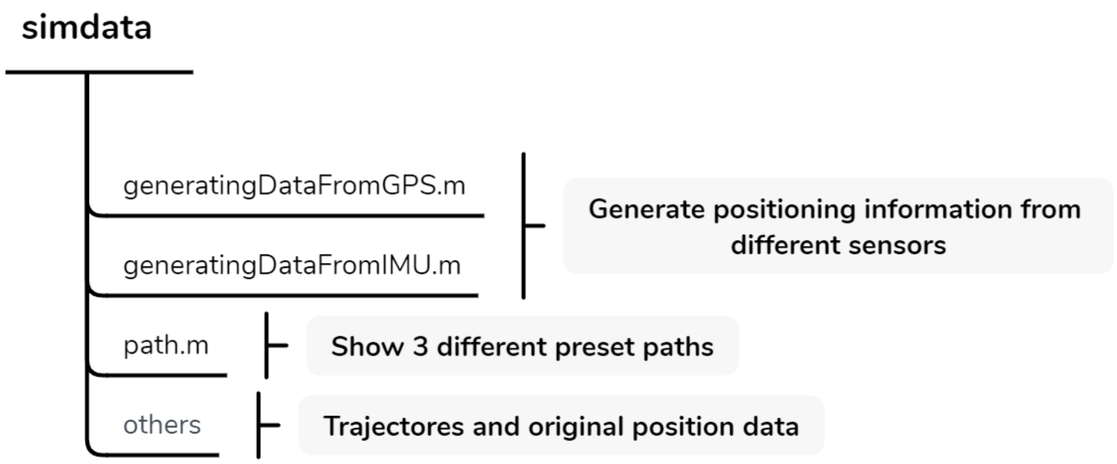

Figure 7: Structure of the folder simdata

In the folder `simdata`, `generatingDataFromGPS.m` and `generatingDataFromIMU.m` call `generatingNoseFromGPS.m` and `generatingNoseFromIMU.m` in the `control` folder to generate the flight data based on a straight-line preset path flight data for the drone formation. For each drone, the data is a 16-element vector:

- [x y z] — Positions in xyz-axes in meters
- [vx vy vz] — Velocities in xyz-directions in meters per second
- [ax ay az] — Accelerations in xyz-directions in meters per second
- [qw qx qy qz] — Quaternion vector for orientation
- [wx wy wz] — Angular velocities in radians per second

In addition to this `path.m` plots three simple predefined paths. Other files store the generated data.

- ## Simulation

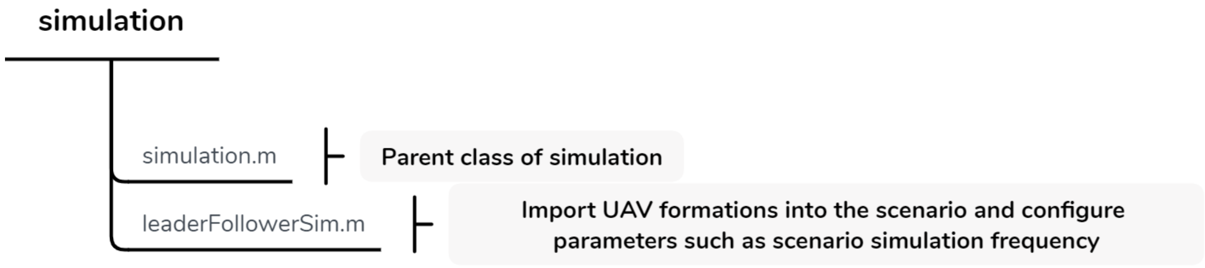

Figure 8: Structure of the folder simulation

In the folder `simulation`, first `simlualtion.m` is a parent class. `leaderFollowerSim.m` is its subclass. This subclass reads the information of the formation and the scene, implements the settings of the whole formation and imports it into the scene.

- ## Root directory

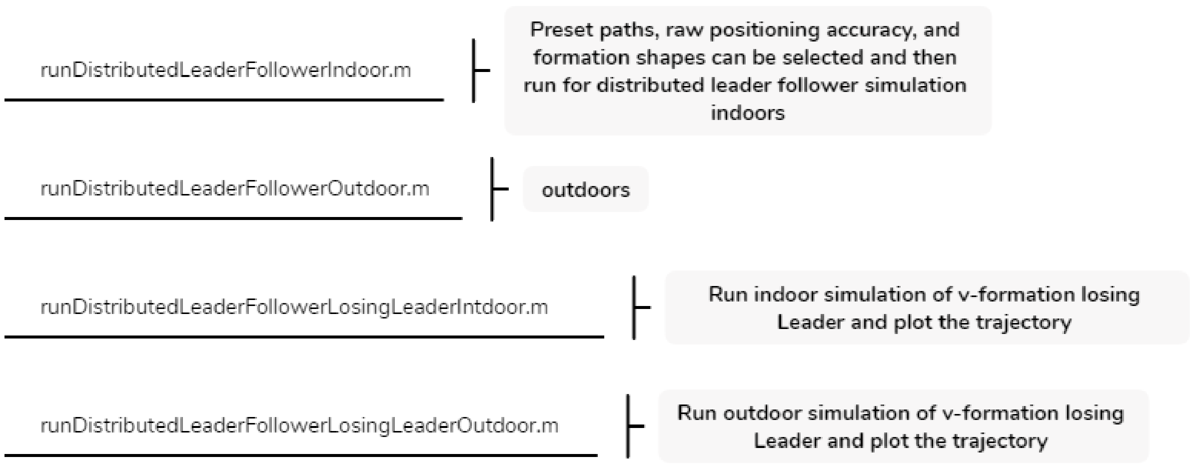

Figure 9: Thesis and run files

In the root directory of the program, there are three runnable files. One of them, `runDistributedLeaderFollowerIndoor.m`, starts a simulation of a UAV swarm in which at least one UAV has not lost its UWB position. And there are three preset paths (as shown in Figure 10) and three formation shapes (in the folder `formation`) to choose from. The user can also set the raw positioning accuracy. `runDistributedLeaderFollowerOutdoor.m` started an outdoor simulation. Like the indoor one, only with the basic accuracy of positioning toned down, i.e., using GPS as the original positioning location method.

`runDistributedLeaderFollowerLosingLeaderIndoor.m` launches to simulate the case when a UAV loses its full leader and then assigns a new one. The case is also run under indoor conditions.

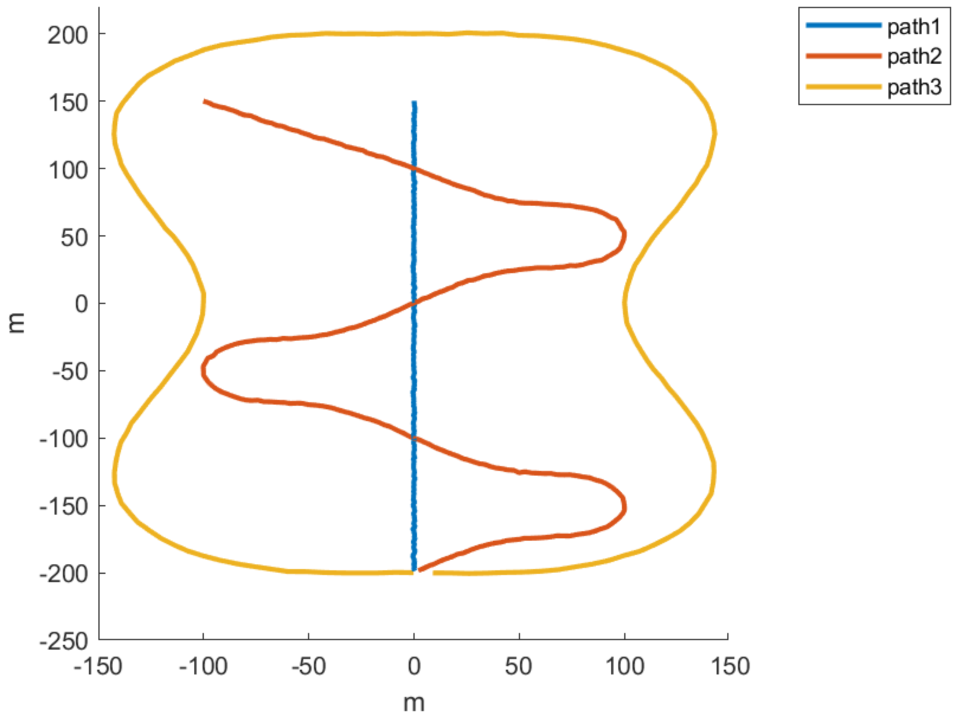

Figure 10: Three different paths

- ## Deploy on PX4 Autopilot

To deploy the algorithm to PX4 Autopilot, the UAV Toolbox Support Package for PX4 Autopilot is required. Autopilot peripherals can be accessed from MATLAB and Simulink. Using Embedded Coder, it is possible to automatically generate C++ code and customize algorithms for Pixhawk and Pixracer using the PX4 toolchain. In addition to this, various sensor data can be integrated.

Here is a general outline of the steps can be followed:

- In Simulink, use the Model Configuration Parameters to set up code generation options for the model.
- Use the Embedded Coder app to create a new build configuration for the model.
- In the build configuration, specify the target hardware and any other required options, such as the C compiler to use.
- Use the build command to generate code from the model and build an executable file.

- Connect to the drone's hardware using a suitable method (e.g., via USB or over a network).

- Use a tool such as xc16-objcopy to convert the generated executable file into a format that can be loaded onto the drone's hardware.

- Use a tool such as xc16-gdb to load the converted executable onto the drone's hardware and run it.

Note that this is just a general outline and the specific steps may vary depending on the specific hardware and software environment used.

Content in CD

The CD contains two main folders:

- `simulation`
- `thesis document`

Folder `simulation` contains the MATLAB code for the simulation. Four scenarios are provided in the root of this folder. They simulate each of the four scenarios as described in the file name:

`runDistributedLeaderFollowerIndoor.m`,

`runDistributedLeaderFollowerOutdoor.m`,

`runDistributedLeaderFollowerLosingLeaderIntdoor.m`

`runDistributedLeaderFollowerLosingLeaderOutdoor.m`

Folder `thesis document` contains thesis report in pdf and MS word file.

Selbständigkeitserklärung

TECHNISCHE UNIVERSITÄT CHEMNITZ

Zentrales Prüfungsamt
Selbstständigkeitserklärung

Name: Yang Vorname: Huan geb. am: 27.12.1991 Matr.-Nr.: 445764	Bitte beachten: 1. Bitte binden Sie dieses Blatt am Ende Ihrer Arbeit ein.

Selbstständigkeitserklärung*

Ich erkläre gegenüber der Technischen Universität Chemnitz, dass ich die vorliegende Masterarbeit selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe.

Die vorliegende Arbeit ist frei von Plagiaten. Alle Ausführungen, die wörtlich oder inhaltlich aus anderen Schriften entnommen sind, habe ich als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch nicht als Prüfungsleistung eingereicht und ist auch noch nicht veröffentlicht.

Datum: 03.01.2023 Unterschrift: Huan Yang

* Statement of Authorship

I hereby certify to the Technische Universität Chemnitz that this thesis is all my own work and uses no external material other than that acknowledged in the text.

This work contains no plagiarism and all sentences or passages directly quoted from other people's work or including content derived from such work have been specifically credited to the authors and sources.

This paper has neither been submitted in the same or a similar form to any other examiner nor for the award of any other degree, nor has it previously been published.