İsa Emre Yıldırım

A Master's Thesis

AGU 2021

RELATIVE LOCALIZATION AND COORDINATION FOR AIR-GROUND ROBOT TEAMS

A THESIS
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND
COMPUTER ENGINEERING
AND THE GRADUATE SCHOOL OF ENGINEERING AND
SCIENCE OF ABDULLAH GUL UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By
İsa Emre Yıldırım
December 2021

RELATIVE LOCALIZATION AND COORDINATION FOR AIR-GROUND ROBOT TEAMS

A THESIS
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE OF ABDULLAH GUL UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By
İsa Emre Yıldırım
December 2021

SCIENTIFIC ETHICS COMPLIANCE

I hereby declare that all information in this document has been obtained in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Name-Surname: İsa Emre Yıldırım
Signature:

REGULATORY COMPLIANCE

M.Sc. thesis titled Relative Localization and Coordination for Air-Ground Robot Teams has been prepared in accordance with the Thesis Writing Guidelines of the Abdullah Gül University, Graduate School of Engineering & Science.

Prepared By	Advisor
İsa Emre Yıldırım	Asst. Prof. Samet Güler
Signature	Signature

Head of the Electrical and Computer Engineering Program

Assoc. Prof. Kutay İçöz

Signature

ACCEPTANCE AND APPROVAL

M.Sc. thesis titled Relative Localization and Coordination For Air-Ground Robot Teams and prepared by İsa Emre Yıldırım has been accepted by the jury in the Electrical and Computer Engineering Graduate Program at Abdullah Gül University, Graduate School of Engineering & Science.

……….. /……….. / ………..

JURY:

Advisor : Asst. Prof. Samet Güler

Member : Assoc. Prof. Günyaz Ablay

Member : Asst. Prof. Erdem Arslan

APPROVAL:

The acceptance of this M.Sc. thesis has been approved by the decision of the Abdullah Gül University, Graduate School of Engineering & Science, Executive Board dated ….. /….. / ……….. and numbered ………….…….. .

……….. /……….. / ………..

(Date)

Graduate School Dean
Prof. Dr. Hakan USTA

ABSTRACT

RELATIVE LOCALIZATION AND COORDINATION FOR AIR-GROUND ROBOT TEAMS

İsa Emre Yıldırım
M.Sc. in Electrical and Computer Engineering
Advisor: Asst. Prof. Samet Güler

December 2021

Recently, autonomous robot teams have been implemented broadly in many social and military applications such as firefighting, agriculture, search and rescue, mapping, target tracking, and docking. A mix of different types of ground robots and aerial vehicles can be employed in a robot team to accomplish tasks efficiently and robustly. Such heterogeneous systems show unparalleled benefits in complex tasks compared to teams composed of identical robot types. In a heterogeneous robot team, precise relative localization, i.e., estimating a robot's position with respect to its neighbor robots, plays a key role. We develop a relative localization system for air-ground robot teams where an aerial vehicle and multiple ground robots work in coordination to perform a reliable relative position estimation. The aerial vehicle is employed to detect special patterns on the ground robots by an onboard monocular camera, while the ground robots perform relative position estimation based on inter-robot distances acquired by ultrawideband sensors and the bearing and heading angles received from the aerial vehicle by communication. Thus, the aerial vehicle serves as an absolute frame provider for the entire team. Notably, each robot in the team uses onboard communication and computation capabilities solely without any need for an external localization infrastructure, making the team realizable in all conditions including GNSS-denied environments. We propose a multi-rate extended Kalman filter algorithm to handle different data rates of the sensor measurements. We carried out an extensive simulation study with a drone and five ground robots in a leader-first follower formation. Simulation results showed a successful estimation performance with an error rate of up to five centimeters in the relative position estimations in both axes.

Keywords: Relative localization, Heterogeneous multi-robot systems, estimation algorithms, ultrawideband sensors

ÖZET

HAVA-YER ROBOT EKİPLERİ İÇİN BAĞIL LOKALİZASYON VE KOORDİNASYON

İsa Emre Yıldırım
Elektrik ve Bilgisayar Mühendisliği Anabilim Dalı Yüksek Lisans
Tez Yöneticisi: Dr. Öğr. Üyesi Samet Güler
Aralık-2021

Son yıllarda, otonom robotlar yangın söndürme, tarım, arama kurtarma, haritalandırma, hedef takibi ve yönelme gibi bir çok sosyal ve askeri uygulamalarda yaygınca kullanılmıştır. Görevleri verimli ve gürbüz bir şekilde tamamlamak için bir robot takımı içerisinde çeşitli tiplerde yer robotları ve hava araçları kullanılabilmektedir. Bu tür heterojen sistemler, karmaşık görevlerin icrasında tek tip robotlardan oluşan takımlara göre benzersiz faydalar sunmaktadır. Heterojen bir robot takımında, hassas bağıl konumlandırma-bir robotun komşularına göre konumunu tahmin etmesi-önemli bir yere sahiptir. Hava-yer robot takımları için, bir hava aracının ve çok sayıda yer robotunun koordineli bir şekilde bağıl konum tahminini gerçekleştirdiği bir bağıl konumlandırma sistemi türetiyoruz. Hava aracı, üzerindeki mono kamerasıyla yer robotları üzerindeki özel örgüleri teşhis ederken, yer robotları da ultrawideband sensörlerden aldıkları robotlar arası mesafeleri ve hava aracından iletişimle aldıkları yönelme açısı ve duruş açısına göre bağıl konum tahmini gerçekleştirmektedirler. Böylece, hava aracı, bütün takım için mutlak düzlem sağlayıcısı olarak görev yapmaktadır. Özellikle, takımdaki her bir robot herhangi bir dış konumlandırma altyapısına ihtiyaç duymadan üzerindeki iletişim ve hesaplama yeteneklerini kullanmaktadır ve bu özellik takımı GNSS olmayan çevreler de dahil olmak üzere her koşulda gerçeklenebilir yapmaktadır. Farklı hızlara sahip sensor ölçümlerini işlemek için bir çok-oranlı genişletilmiş Kalman filtresi önermekteyiz. Bir dron ve lider-ilk takipçi formasyonunda beş yer robotu ile kapsamlı bir simülasyon çalışması gerçekleştirdik. Simülasyon sonuçları her iki eksendeki bağıl konum tahmininde beş santimetreye kadar hata payı olan başarılı bir tahmin performansı göstermiştir.

Anahtar kelimeler: Bağıl lokalizasyon, Heterojen çoklu-robot sistemleri, tahmin algoritmaları, ultrawideband sensörler

Acknowledgements

I would like to express my gratitude to Dr. Samet GÜLER, who helped me to complete this study by giving me a great deal of time and valuable contributions.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

CL	Cooperative Localization
HMRS	Heterogeneous Multi robot Systems
EKF	Extended Kalman Filter
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle

To all my loved ones.

Chapter 1

Introduction

The fact that robots became mobile over time and gained autonomous driving capability with advanced sensors led to the idea that they could work as a team. The concept of using many mobile robots on the same mission has become widespread. Some examples include firefighting [1], search and rescue [2], military operations [3], mapping [4], and target tracking [5]. The field of multi-robot systems (MRS) has become popular with the equipment of the sophisticated sensing mechanisms on mobile robots.

MRS can be grouped into two parts: Homogeneous MRS and heterogeneous MRS. In a homogeneous MRS, the individual robots share the same characteristics and capabilities while in a heterogeneous MRS, team members differ from each other in several aspects. For instance, a team of identical ground robots or a set of identical aerial vehicles compose a homogeneous MRS. To call a system heterogeneous, some robots in the MRS should possess different characteristics and types. A MRS consisting of a set of ground robots and unmanned aerial vehicles (UAV) can be considered a heterogeneous MRS.

The use of heterogeneous MRS allows combining the unique features of ground and aerial robots. For example, although ground robots can move long distances with their large battery capacity, they may not have a highly flexible and agile motion capabilities as aerial robots. While ground robots can easily carry a specified heavy load, it is much more difficult for an aircraft to carry the same load due to its small size and limited battery capacity. Furthermore, if visual sensors are used in an operation, the image features captured by ground and aerial robots differ drastically due to the location and perspectives of the onboard cameras. In homogeneous MRS, aerial imaging operations cannot be used when only ground robots are used. Likewise, ground robots' battery power and carrying capacity are deprived in systems where only aircraft are used. To achieve complex and demanding tasks, employing a heterogeneous MRS might be more useful and efficient.

Although MRS can provide several advantages and improve efficiency in many applications, several underlying issues need to be tackled before utilizing the MRS at full capacity. The lack of a reliable and robust localization algorithm for HMRS remains the most critical limitation. To achieve coordination among team members in a HMRS, every member should be able to have a situational awareness, which requires to implement a localization algorithm. Current localization methods either employ an external infrastructure such as a motion capture system and GPS or are based on a centralized schemes. In a centralized framework, a ground station broadcasts the locations of the robots which are detected by external infrastructures, which makes the entire system dependent to these infrastructures. In contrast, the robots can utilize their onboard sensors to perform localization in a distributed manner.

We address the distributed localization problem for HMRS composed of a UAV and multiple UGVs. We propose a relative localization and coordination framework for the entire system such that the UAV hovers over the UGV team and provide the UGVs with an online stream of quantities which are used in the relative localization of the UGVs. Particularly, the UAV detects the heading and bearing angles of subparts of the UGVs, and the UGVs perform extended Kalman filtering to estimate the relative positions to their neighbor UGVs by utilizing the feedback from the UAV and inter-robot distances acquired from ultrawideband sensors. Our main contribution is the introduction of a localization and coordination framework for air-ground robot teams which operate in environments without external localization infrastructures. We give a comprehensive summary of the recent progress about the localization and coordination works in the literature in the following.

1.1 Related Works

HMRS have been utilized in many applications. For example, Tokekar conducted a study [6] to control the status of agricultural lands. He analyzed images taken by flying over farmland with a UAV to determine the nitrogen level of the soil. It also used the UGV as a charging station for the UAV. In [7], UAVs and UGVs are used together to measure environmental factors such as temperature, humidity, luminosity, and carbon dioxide content of a greenhouse. The UGV carries the UAV on it. When it is deemed necessary, the UAV takes off and collects the data for the necessary measurements. In this way, it aims to extends the UAV’s battery life by taking measurements at the most effective level. In [8], a ground robot and multiple UAVs are employed together to make the agricultural land effective. Similarly, Gonzalez-de-Santos *et. al.* used HMRS for pest control in order to increase product quality in [9]. By detecting weeds over a field, drones locate paths for ground vehicles and direct the ground vehicles to these areas. HMRS has also been widely used in firefighting. Phan and Liu used a UAV for detecting fires and directing UGVs to those areas in [10]. On the other hand, in [11], Quenzel *et. al.* equipped UAVs with thermal cameras and lidars to follow fires and direct UGVs to those locations. Then, the UAV and UGVs collaborated to extinguish the fire. In addition, the unique benefits of HMRS have been exploited in industrial applications as well. In [12], Wang *et. al.* UAVs and UGVs cooperate to save time and improve path planning such that UAVs can tour the storage areas faster and use UGVs as charging stations. Reference [13], on the other hand, considers a HMRS where a UAV guides a group of UGVs by moving in formation to transport materials such as gas masks and tools to people in need. Among the studies carried out with HMRS, those related to search and rescue have an essential place. In [14], different types of ground robots are positioned in different parts of the search and rescue environment. The robots were guided by telemetry, and the area could be scanned to find the person to be rescued.

To compute the a mobile robot’s location, its starting point and velocity at each time interval must be known. One may use proprioceptive or exteroceptive sensors for this purpose. Proprioceptive sensors such as wheel encoders and inertial measurement units (IMUs) are located onboard and provide information about the robot’s linear and angular displacement. The exteroceptive sensors such as GPS, laser range finder, and UWB measure the robot’s displacement or location by utilizing environmental information. Many of these sensors can be used together to obtain more accurate location information.

Self-localization refers to calculation of a robot’s own location with proprioceptive sensors. Inertial sensors convert a moving device’s vibration, inclination, and acceleration into electrical signals to find the force applied to the body. These devices, called Micro Electromechanical System (MEMS), are widely used in many areas due to their low cost and low energy consumption [15]. In particular, these sensors are used to regulate the height and attitude of autonomous aircraft. In addition, MEMS can be used in flight control systems of micro UAVs due to its minimal size and low energy efficiency [16].

Inertial sensors are usually combined in a single unit called IMU. The IMU is an electronic unit that contains a mobile device’s acceleration, angular velocity, and linear velocity information in a single module. It uses a 3-axis accelerometer and a 3-axis gyroscope, and sometimes a magnetometer. The IMU can perform location calculations by knowing the angular and linear velocity values like the odometry sensor. Self-localization with IMU sensors usually result in drifts [17]. For instance, when one of the wheels of the mobile robot drifts, the sensor data will not provide the correct calculation because IMU will detect the robot as rotating or moving. Even insignificant drift movements can cause significant error rates in location estimation with time.

Even if the IMU does not offer a highly reliable calculation, it provides excellent benefits when used with other sensors. For example, IMU sensors can be used together with GPS sensors [18]. The mobile robot determines the location using the IMU in environments where it enters closed areas such as buildings and tunnels and where GPS signals are interrupted. Thus, although it is not reliable, it can be an excellent solution when considering situations such as GPS, where no data can be obtained.

Situational awareness is vital for robots operating as a team. While team members may have to perform tasks spread over an area expressed in kilometers, they may also have to work in small areas. For this reason, each robot has to know the relative positions toward the other robots in the team. The position of a robot with respect to another robot is called its relative location. To keep the team formation under control, avoid collisions, and carry out the mapping operations properly, the robots in a MRS need to know their position relative to each other and develop appropriate mobility. In teams performing the given task in areas spread over square kilometers, locating errors consisting of meters can be acceptable. However, even millimeter error rates can prevent completing the given task in teams working in very small areas. For example, suppose there are robots carrying products in a factory and they have 50 centimeter relative localization error. In that case, these robots are likely to collide with each other. Therefore, the lower the error rate, the higher the task completion rate will be.

The fact that a robot can only calculate its position does not make any sense for robots that act as a team. Because if a robot does not know the location of another robot, it means that they will not be able to direct each other correctly for the execution of the job. For example, imagine a drone and ground robots working together on a project that identifies and rescues an injured person in a disaster area. Drone and ground robots can do self-localization. For an injured person the drone sees on the ground, one of the ground robots has to be directed to that area. For this reason, the drone has to know the position information of the other ground robots. In order to calculate which robot will be sent to that region and with what angle and displacement it will reach that region, they need to know each other's positions. To give a different example, imagine that a land-scanning drone uses a ground robot as a charging station. Assume that the drone is placed on the ground robot at certain time intervals and takes off again as a charge. Likewise, for the drone to be placed on the ground robot, it must know the position of the ground robot relative to itself, with a minimum error rate. For this reason, it is crucial for robots working as a team to know their positions with each other. It is possible to do complex works only if their relative locations are known.

Robots try to detect data obtained from sensors such as sound, Bluetooth, IMU, UWB, acoustics to obtain location information using various algorithms. Location information cannot be obtained with 100% accuracy due to errors from these sensors' production and conditions, such as weather. Although this error rate is reduced day by day thanks to the studies carried out, localization is a significant challenge that needs to be developed for MRS.

For the robots to find their positions with each other, they must be sharing sensor data like Bluetooth, UWB, IMU. The location determination of robots by sharing sensor data is called Cooperative Localization (CL). Calculation by combining sensor data will allow more precise location calculations to be made. We can think of the process of calculating the positions of the robots against each other as three stages. First, each robot must be collecting its sensor data. As a second process, this sensor data should be shared with other robots. As a third process, by adding different types of sensors to different types of robots, diversity should be provided to make more precise measurements.

CL first emerged in Kurazume's 1994 study [19], called cooperative positioning. In this study, the robots were divided into two classes. One group was moving while the other group was standing still and acting as landmarks for the other robots. Later, moving robots and fixed robots were replaced, allowing them to reach goal positions. In 1998, Bison tried to determine the robots' locations by combining the robots' sonar and laser sensor data. He used robots as master-slave in his work [20], where the name cooperative localization was first used. Jenning and Murray used two stereo vision-based mobile robots in their work [21], and the first robot was assigned to find suitable landmarks. The second robot tried to determine its position relative to the first robot by using the landmarks determined by the first robot. The works [22-24] study CL aided by vision sensors. It can be seen that much work has been done using very different types of sensors such as laser range [25-27], GPS and radar [28,29], acoustic [30,31], and also sonar [32]. Also, the studies [33-35] can be given as examples to CL. In [33], Duan tried to implement flocking and formation control operations within a team of UAVs and UGVs. In [34], Tanner developed a robot team application to scan an area to find a vehicle in motion. In [35], Papachristos worked on the navigation operation of many UAVs and UGVs.

CL can be beneficial in tiny robots. Small-sized robots cannot perform large processing capacity since their batteries must also be small. Therefore, instead of using all sensors together in a single robot, it is more advantageous to distribute the sensors to other robots. For example, Grabowski used small robots with dimensions of 7*7*7 in his study [36] and tried to determine a location by placing a separate sensor on each robot.

Positioning is done in two ways. The first is absolute positioning, and the second is relative positioning to perform more sensitive tasks. While absolute location determination is done with GPS sensors, more sensitive sensors should be used for relative position calculations. GPS sensor localization determination is successful up to a certain extent. However, it is not suitable for applications where a more precise ratio should be determined since it may have an error rate of five meters [37]. Especially in applications requiring indoor location information, this error rate may be higher due to signal losses. For example, suppose it is desired to obtain location information in an environment where external signals such as caves, tunnels, and underground cannot be received. In that case, no data can be obtained from GPS [38,39].

The size of this error rate in GPS sensors and the fact that they cannot be used, especially in indoor environments, has led to the development of different sensor types and other methods in location determination.

UWB sensors are radio-based communication sensors that provide high band speed for distance measurements. Since UWB sensors have very high bandwidth, they can reach the receiver in more than one way at the same time. However, the high bandwidth allows different frequencies to be used at different times. Thanks to this, it can be used as a solution against multipath problems and disruptive effects. In addition, UWB sensors consume less energy than other technologies, allowing these sensors to be used for a more extended period. At the same time, UWB sensors also provide more data transmission possibilities than other technologies. So they provide more efficiency due to both energy consumption and more data transmission. UWB sensors generally operate in the 3.1 to 10.6 GHz range. This range ensures that communication tools such as WIFI and Bluetooth operating in the 2.4 GHz frequency band are not affected by their signals. Furthermore, UWB sensors use distance measurement time of flight (ToF) and time of arrival (TaO). With the ToF and TDoA methods, indoor position determination can be performed with an average error of around 30 cm [40].

UWB sensors work with one tag and more than one beacon. The tag location can be attached to the object trying to be detected. Moreover, the location can be determined by measuring the distances to this object with the beacons. At least three beacons are needed to determine the tag's location in the 2d plane [41].

It can be seen that UWB sensors are used in many areas in positioning applications. In indoor positioning applications, the beacons (base stations) are fixed to the environment. At the same time, the tag moves, and operations are carried out. For example, in this study [42], Zhang placed the tag on the ground robot, fixed the beacons with tripods, tried to find the location of the ground robot using a Bayesian filter. He managed to reduce the error rate up to a few centimeters. In this [43] study, Prorok performed a location determination application using Monte-Carlo Localization (MCL) algorithm with four fixed beacons in a 3x3 area. In this [44] study, Feng developed a location determination application by fixing the beacons indoors and using the IMU and UWB sensor data in the Kalman Filter algorithm. Similarly, Gonzalez combined GPS and UWB sensor data with the Monte Carlo algorithm using fixed anchors in his study [45] and applied location determination. On the other hand, in this study [46], Hyun performed the location determination application by placing 16 UWB anchors in different places and floors to find a drone's location in an area that he called a disaster environment.

We can say several methods make that distance measurement. For example, we can mention it as a robot-to-beacon method in [47-53] applications. It is possible to say that these [48, 53-55] studies as beacon-to-beacon.

Beacons in an environment can be either fixed or mobile. For example, in this [56] study, Ortiz removed the beacons from stationary by placing UWB sensors on a single UAV. At the same time, he stated that they got the best results when the distance between the anchors was 40 centimeters. Based on this, Hepp placed 1 Decawave 1000 UWB sensors on a moving human and four on a UAV in this [57] study. The UAV is intended to follow a moving human being at a certain distance and omni-directionally. In this study, the UWB sensor on the tag is assigned as slave. One of the sensors on the UAV is assigned as master, and the others are assigned as listener. The other three sensors on the UAV were used as listeners. Distance measurement was made with TOF. To track the target slave and master have to communicate with each other. Also, master communicates with listeners. An iterated EKF algorithm is used when the measurement is obtained, and the position is found. Hepp has achieved an error accuracy between 10cm to 22cm for target tracking with this work.

Like Hepp's work, Güler placed three Decawave UWB sensors on a hexacopter as an anchor and one Decawave UWB sensor on a quadrotor as a tag in his work [58]. Unlike Hepp, Güler used a dual MCL algorithm to compute the agile maneuvers of the robot and calculate the relative position. He compared dual MCL with standard Kalman filter and particle filter algorithms in both experiments conducted separately as indoor and outdoor. As a result, he achieved better results with dual MCL.

Similarly, in this study by Xianjia, the UAV is equipped with two UWB transceivers, and a camera performs VIO estimation. The robot is an outdoor platform equipped with four UWB responder transceivers for cooperative positioning. He extracts the UAV position from the lidar's point cloud and utilizes this as a reference. Although he achieved a good result at first with this study, as the UAV gains altitude, the UAV loses its reference, and the error rate increases.

UWB sensors are widely used in relative localization calculations in HMRS systems. UWB sensors are not used alone but more commonly in applications, especially in conjunction with VIO odometry. Sometimes, it can be the robots that are tried to be detected with the camera or the markers placed on the robots. For example, In this paper [59], path planning, mapping, and localization are done with a ground robot and two MAVs. They have UWB sensors on them and depth cameras. The UGV is set to rotate. Then, move within a limited space to maintain the MAV within its FoV and LoS. The ground robot is equipped with a Livox Horizon lidar. A TF Mini Lidar is utilized for height estimation on the MAV, also equipped with a depth camera for sensing and 3D reconstruction. Both robots also use cameras for VIO-based ego-motion estimation. The distance between each pair of robots is estimated using Decawave DWM1001 UWB transceivers. They consider three-dimensional space and use a graph realization method to do the collaborative localization. The proposed algorithm effectively ensures MAVs are always within the UGVs FoV. Realtime experiments show that the localization framework is accurate enough to align point clouds even at the raw data level.

In such studies, if there are no UWB sensors that detect distance, the application must be confined to a minimal area since aircraft will lose their position values when they cannot follow the ground robots with their cameras. For example, Cognetti tried to determine the identities and locations of the robots on the ground by keeping the UAV on the ground robots in this study [60]. However, the UAV has to see the ground robots with its camera constantly. When it cannot take pictures, the algorithm becomes disfunctional.

Again, we can see the same problem in Stegagno's [61] study. To avoid this problem, many studies were carried out with UWB sensors.

In his study [62], Nguyen used a UGV and a UAV for relative positioning and target tracking. There are 2 UWB sensors on the UGV and 2 UWB sensors on the UAV. The robots' inertial measurement unit (IMU) fused the UWB sensor data with the altimeters and optical flow. It made the relative localization calculation with the EKF algorithm. In his following study, [63] Nguyen tried to do docking by placing 4 UWB sensors on the ground robot this time. He considers the autonomous docking problem of UAVs onto a moving Unmanned Ground Vehicle (UGV) from a long distance by interacting Ultra-Wide Band (UWB) and vision sensors to provide accurate and reliable relative localization. He used An UAV with a UWB sensor and a moving platform with 4 UWB sensors, and a QR code platform. Both UAV and UGV find their displacement on themselves. UGV also finds its orientation. UAVs do this using EKF. UGV sends the replacement information to UAV, and finally, after getting the replacements and orientation information, a recursive least square optimization algorithm is employed to estimate relative position. A velocity command is calculated and sent to the onboard flight controller to drive the UAV towards the target based on this estimate. From 50 m away, all distances reduce to 1.25 m in the end, reflecting the successful docking task. UAV can move to the target and land on it with its one UWB sensor. In his study [64], Petrillo tried to determine the UAV's location using the UGV but also tried to find the best trajectory by mapping its environment. For this, he used a fusion of fish-eye camera, 3d lidar, UWB sensors, and laser altimeter data. An absolute distance to the ground is measured from a sensor mounted in fixed orientation on the UAV's body for the altimeter. The UAV's pitch and roll are estimated from the UAV's IMU during the belief-state propagation process. Moreover, the distance between UAV and UGV is taken via UWB sensors. To estimate the UAV position EKf algorithm is used for fusing the sensor values.

Detection of ground vehicles of aircraft in HMRS is of great importance. For example, for an aircraft to land at a moving ground station, it needs sensors that determine the landing place with the camera sensor we have applied. They also need sensors that calculate the distance. In such applications, features such as object identification are used to detect the ground target of the aircraft. Recently, instead of defining the whole object, this identification has become more accessible with a QR code or marker placed on the object. Considering that more than one ground robot of the same type is acting together in a team, it will be challenging for the aircraft to identify them. However, when a different marker is placed on each robot, identification will be much easier. It will help to get rid of an extra identification algorithm burden. For this reason, the use of markers has become widespread in many applications.

We can see the use of QR codes in many studies in HMRS applications. For example, Dubois has placed QR codes on both the top and back of the ground robots to perform localization and mapping in this study [65]. The upper QR enables the UAV to detect the identity of the ground robot, and the QR at the back enables the identification of other ground robots. For this, both the ground robot and the UAV have camera sensors. Dubois 3 placed different QR codes on the ground robot and performed the mapping process with Multi-Robot SLAM algorithm. In this [66] study, Cocchioni placed a small black dot on the ground robot and a marker with a small arrow showing this black dot. Thanks to this black dot and arrow, the UAV can detect the landing direction on the platform. Localization was performed with fastSLAM, and it was able to reduce the landing process on an area of 200 cm square to 2 cm error rate in 200 cm. In his study [67], Hood placed a marker on both the UGV and the UAV for relative position estimation, allowing both robots to detect each other. In this [68] study by Mueggler, markers were placed on the obstacles, target position, and ground robot. The UAV first localizes the UGV. A different vision-based approach is described. The UAV maps an area of interest based on given waypoints. It then guides a UGV through obstacles with collected camera data. Niu aimed to charge the drone by landing at the mobile ground station in his study [69]. Niu transformed the ground robot into a platform, placed a QR code on it, and tried to detect this QR code and the platform's location with the drone's camera. Localization is of great importance in such studies. If Niu had not been able to find the location of the ground robot with a small error, it could lead to a drone damage event. The drone would land incorrectly on the station and cause it to be damaged. We can also give examples of [70-73] applications to HMRS studies using cameras and markers.

1.2 Organization

The rest of the thesis is organized as follows. In Chapter 2, we provide a general description of the heterogeneous multi-robot system. We explain the types, sensor characteristics, and motion models of the individual robots. Namely, we combine the ground and aerial robots in a unified system. Then, we define two objectives that are studied in the work.

Chapter 3 gives the general localization framework designed for the ground robots. We start by designing a Bayesian estimation algorithm for a two-robot system. Particularly, we apply a commonly preferred extended Kalman filter to estimate the relative positions between the ground robots after deriving a unique state-space model with four states for each robot. Afterwards, we give the formation graph among the entire UGV team and explain the roles of the individual robots in the team. Furthermore, we give the estimation and control algorithms for the drone.

In Chapter 4, we provide the results of an extensive simulation study. We show the estimation results together with their ground truth values. We analyze the results deeply, discussing the practical aspects for a smooth transition to real-life experiments.

In Chapter 5, we demonstrate our experimental studies. We carried out a set of experiments with a drone and three nonholonomic ground robots in an indoor environment. We describe the robots, the hardware and software used, and the operational procedures. Then, we illustrate the results of the experiments in detail and discuss the advantages and drawbacks of our framework. Finally, the last chapter concludes the work and opens a new avenue for future studies.

Chapter 2

System Definition

We consider a team of mobile robots which consists of a drone and N ground robots, forming a heterogeneous MRS of ground and aerial vehicles (Figure 2.1). The robots are designed to move in coordination to achieve a given task such as search and rescue, border patrolling, and precision agriculture. Accordingly, the system operates based on a predefined path planning procedure. In this section, we will define the system in detail, including the robot types and kinematic models and the formation structure among the team, and then describe the problem of interest.

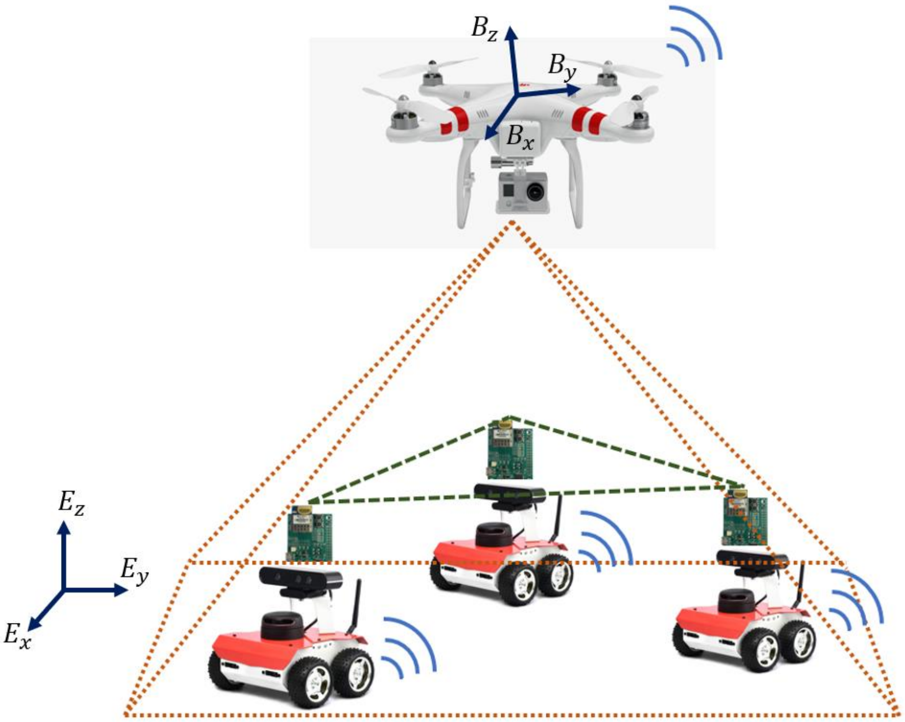


Figure 2.1 The proposed heterogeneous system of four robots.

2.1 Multi-Robot System Model

Our multi-robot system (MRS) comprises $N + 1$ robots, an aerial vehicle and N ground robots. The aerial vehicle is designed as a vertical take-off and landing (VTOL) vehicle, commonly referred to as a drone. It has the capability of moving on a plane at a given altitude with a desired velocity. Thus, it is intended to provide an aerial view of the ground robots to facilitate their estimation mechanisms. We describe the motion models and characteristics of the heterogeneous robotic system in the following.

2.1.1 Motion Models of the Ground Robots

There are a variety of ground robot motion types. Motivated by the widespread use of the two-wheeled ground robots (together with a castor wheel) such as the commercial RosBot and Turtlebot, we assume that the ground robots have non-holonomic kinematics. The term holonomic refers to the relationship between a robot's controllable and total degree-of-freedom (DOF). If the total DOF is more than the controllable DOF in a robot, the robot is called non-holonomic. A mobile ground robot has three DOFs, namely its position in two axes and its direction. In a non-holonomic ground robot, there are only two DOFs that can be controlled, which are acceleration and steering wheel turning angle. On the other hand, in a holonomic robot, the controllable DOF are equal to the total DOF. A robot built on caster wheels or omni-wheels is a good example of holonomic driving. These caster wheels robots can move freely in any direction.

The kinematic model of a robot examines the relationship between the geometric connections that build the system, the control variables, and the system behavior in the state space, without considering the dynamical effects such as force and torque. We define the state variables of a differential drive robot as:

$$\mathbf{x} = [x, y, \theta]^{\top} \tag{2.1}$$

where x, y denote the 2D planar coordinates of the robot, and θ the orientation of the robot. As shown in Figure 2.1, denote the linear velocities of the left and right wheels of the robot by v_L and v_R, and the angular velocity by ω. Let the distance between the wheels of the robot and the center of curvature radius be given by D and CCR, respectively. Accordingly, the speed of the robot relative to the center point is calculated as:

$$v_c = (v_L + v_R)/2 \tag{2.2}$$

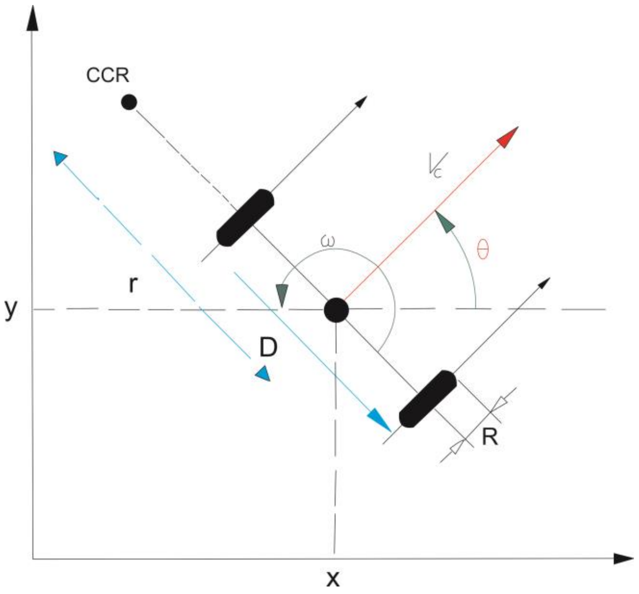


Figure 2.2 Representation of a nonholonomic mobile robot in the Euclidean plane.

The change in speed and orientation angle of the robot's center point with respect to the planar coordinates is expressed by:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_c \\ \omega \end{bmatrix} \tag{2.3}$$

The robot has three state variables x, y, θ and two control variables, making it non-holonomic. The control variables in this system are the angular speeds of the DC motors driving the left and right wheels, w_L, w_L, and the output of the system is the planar coordinates x, y and the heading angle θ of the robot.

2.1.2 Kinematic Model of the Drone

Before obtaining the kinematic model of the drone, we first define the coordinate frames to be used. We show these coordinates in Figure 2.1 as ground coordinate frame (reference frame) $\mathcal{E} = \{E_x, E_y, E_z\}$ and the body coordinate frame $\mathcal{B} = \{B_x, B_y, B_z\}$. The ground coordinate frame is fixed and provides a reference for the motion, while the body coordinate frame is attached to the drone and translates and rotates with the drone. We define the relative position between the ground $\mathcal{E}$ and body $\mathcal{B}$ coordinate frames as $\mathbf{r} = [x, y, z]^T$.

The drone’s orientation relative to the ground axis is defined by the transformation matrix R_B^E. This transformation can be represented by using the three Euler angles or quaternion. For instance, the transformation with the Euler angles is represented as three rotations in the yaw, pitch, and roll axes as follows:

$$R_i^b = R(\phi, x) * R(\theta, y) * R(\psi, z) \quad = \tag{2.4}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\phi) & sin(\phi) \\ 0 & sin(\phi) & sin(\phi) \end{bmatrix} \begin{bmatrix} cos(\theta) & 0 & sin(\theta) \\ 0 & 1 & 0 \\ -sin(\theta) & cos(\theta) & 0 \end{bmatrix} \begin{bmatrix} cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} cos(\psi)cos(\theta) & cos(\psi)sin(\theta)sin(\phi) - sin(\psi)cos(\phi) & cos(\psi)sin(\theta)cos(\phi) + sin(\psi)sin(\phi) \\ sin(\psi)cos(\theta) & sin(\psi)sin(\theta)sin(\phi) + cos(\psi)cos(\phi) & sin(\psi)sin(\theta)cos(\phi) - sin(\phi)cos(\psi) \\ -sin(\theta) & cos(\theta)sin(\phi) & cos(\theta)cos(\phi) \end{bmatrix} \tag{2.5}$$

2.1.3 Dynamic Models of the Robots

The dynamic model of a system is created by considering all the forces affecting that system. Many methods are used to construct dynamic equations of non-holonomic systems. The most commonly used of these are the Euler-Lagrange equations. The dynamic equations of the mobile robot can be express as:

$$M(q)\ddot{q} = C(q, \dot{q})\dot{q} + G(q) = B(q)\tau + J^T(q)\lambda \tag{2.6}$$

Then, we can write nonholonomic constraint equation:

$$J(q)\dot{q} = 0 \tag{2.7}$$

In the (2.7) equations, q is the state variables vector of the robot. $M(q)$ is a positive definite symmetric matrix of size nxn. $C(q, \dot{q})$ denotes an n dimensional vector showing central torque and coriolis (swing) moments. $G(q)$ is an n dimensional vector of the moments of gravity. $B(q)$ represents the nxr dimensional input transformation matrix $(r < n)$, τ the r dimensional input vector, and λ the Lagrangian multipliers of the constraint conditions.

If we rearrange the equation assuming that $G(q)$ and $C(q, \dot{q})$ are 0 we can get the (2.8**)** equation:

$$\begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \end{bmatrix} = \frac{1}{R} \begin{bmatrix} cos(\theta) & cos(\theta) \\ sin(\theta) & sin(\theta) \\ L & -L \\ \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \tag{2.8}$$

In the (2.8) equation τ_1 and τ_2 indicates the torques of the left and right motors, m and I indicate the mass and inertia of the robot. R indicates the radius of the wheels, and L indicates the distance between the wheels. After these assumptions are made, the equations are solved by adding the nonholonomic constraint.

2.1.4 Dynamic Model of the Drone

Obtaining the dynamic model of the quadrotor is examined under two main parts. The first one is rotational motion equations, and the second one is transformation motion equations. The rotational motion equations are the equations of motion derived from the B axes using the Newton-Euler method. The moment expression acting on the B axes is written as follows:

$$M_b = J\dot{\omega} + \omega x J\omega + \omega x \begin{bmatrix} 0 \\ 0 \\ J_r\Omega_r \end{bmatrix} \tag{2.9}$$

M_b moment acting on B axes, J is the diagonal inertia matrix, ω is the angular velocity vector, $\dot{\omega}$ is the angular acceleration vector of the drone. J_r the rotor inertia, ω_r the rotational imbalance of the drone in the z-axis.

The thrust force produced by the rotors performs the rotation effect and thus the moment effect occurs as a result of the product of the force and the lever arm. The force and moment equations produced by the rotors are expressed as follows:

$$\begin{gathered} F_i = b\Omega_i^2 \\ M_i = bl\Omega_i^2 \end{gathered} \tag{2.10}$$

F_i is the thrust of the ith rotor, b is the aerodynamic force constant, Ω is the angular velocity of the ith rotor, M_i is the torque of the ith rotor. l is the distance from the center of the drone to the rotor. If the effects of moments and forces produced by the rotors are examined on the drone, the moments on the x, y, z axes are written as follows.

$$M_B = \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} lb(-\Omega_2^2 + \Omega_4^2) \\ lb(\Omega_2^2 + \Omega_2^2) \\ d(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \end{bmatrix} \tag{2.11}$$

M_x can be expressed as the moment acting on the x-axis, M_y as the y axis, M_z as the z-axis, and d as the rotational unbalance moment constant. We can write the transformation equation of motion to the fixed axes based on Newton's second law, as (2.12) equation.

$$m\ddot{r} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + RF_B \tag{2.12}$$

Here F_B is the total thrust of the quadrotor rotors independent of gravity, m the mass of the quadrotor, $\ddot{r}$ the acceleration in Newton's second law, R the transformation matrix. Than, we can write the total thrust force as:

$$F_B = \begin{bmatrix} 0 \\ 0 \\ -b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \end{bmatrix} \tag{2.13}$$

2.2 Problem Definition

The heterogeneous MRS defined above can serve in various tasks efficiently and robustly by exploiting the distinct characteristics of the individual robots. To operate such a system in unknown environments with minimal sensing capabilities remains an open problem in the robotics literature. Resilient solutions to this problem should combine the localization, mapping, and control efforts in a robust and efficient manner. Such solutions can be successful only if the individual group members are aware of each other during an operation, which demands for a practical localization algorithm.

Our main objective is to design a robust localization algorithm for the system components such that each robot will be able to estimate or receive by communication the relative quantities required for a coordinated motion. For this purpose, the drone and the UGVs benefit from each other by means of an efficient sensing and communication mechanism. We define the localization objective as follows. The drone should provide a perspective view of the sub-groups of the UGV team while the UGVs should estimate the relative positions to their neighbors in an underlying constraint graph defined among the entire UGV team. We focus on the estimation of the relative positions between the ground robots. Denote by $\mathbf{r}_i$ the relative position between robots R_0 and R_i in the local coordinate frame $\mathcal{F}_i$ of robot R_i. Notably, most leader-based formation control algorithms require the acquisition of $\mathbf{r}^i$ for efficient coordination mechanisms. Simultaneously, drone D_0

moves with the ground robots in coordination by providing the subgroups of the UGVs with absolute measurements to be used in the relative position estimations.

In our framework, the motion model of robot $\mathrm{R_i}$ is represented as the usual non-holonomic kinematics as given in (2.1). Each ground robot $\mathrm{R_i}$ is equipped with UWB sensors which acquire inter-robot distance measurements. Also, each ground robot possesses the proprioceptive sensors IMU and wheel encoders for low-level motion control, and a low-cost computational board to perform estimation algorithms.

The drone motion is assumed to obey the first-order kinematics as given in Section 1. The drone is equipped with a downward-facing monocular camera to acquire a perspective view of the ground robots. As a common practice, the drone is also equipped with a flight control unit (FCU) and proprioceptive sensors such as IMU and magnetometer. The drone always moves on a plane at a constant altitude h from the ground. Also, we assume that the motion in the roll and pitch axes are stabilized by the low-level controller of the FCU.

We seek for solutions to the following objectives:

Objective 1. Design a Bayesian localization algorithm for the UGVs to estimate the relative positions between themselves by employing onboard sensors only.

Objective 2. Design a coordination algorithm for the entire MRS which controls the individual robots by using onboard sensors only, without the aid of an external infrastructure.

Essentially, Objective 1 requires a derivation of the system model based on the relative quantities between the UGVs. In addition, Objective 2 requires to combine the estimation results with the formation control algorithm for the UGVs. On the other hand, the drone control based on the perception of the UGV team requires a precise low-level control algorithm. In the following chapter, we address these issues and derive our estimation and control algorithms accordingly.

Chapter 3

Relative Localization and Coordination

We design a robust localization algorithm for the entire MRS under the sensing and communication constraints described in the previous chapter. The heterogeneity of the MRS inherently provides advantages for the observability of the relative states. Motivated by this fact, we aim at minimizing the number of the sensing and communication units onboard of the robots in the design process. In our framework, the drone observes the ground robots from above and sends them the feedback signal required for their reliable localization. On the other hand, the ground robots receive the inter-robot distance and bearing information by communication to estimate the relative positions to their neighbors. This mutual relationship improves the localization performance and results in a coordinated team behavior.

In this chapter, we describe the general perception-decision framework, including the formation graph of the UGVs, derive a state-space system for the relative quantities between the UGVs, and design a state estimation algorithm.

3.1 Formation Graph

The UGVs can be assigned various tasks to perform a variety of applications. In many tasks, the robots must move in coordination. To provide an efficient coordination mechanism among the robots, each robot can be assigned a specific rule which usually depends on the other robots. A particular example of such applications is formation control where a team of robots move from an initial configuration to a final configuration by achieving some rules. A typical rule is to maintain a certain distance, bearing, or relative position to a sub-group of other robots in the team.

In abstract terms, we can define a formation graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ with a set of nodes $\mathcal{V}$ and a set of edges $\mathcal{E}$ which contain pairs of the nodes in $\mathcal{V}$. A typical formation graph is the leader-first follower graph where each of the $n - 1$ nodes $v_1, \ldots, v_{n-1}$ in $\mathcal{V}$ are paired with the first node v_0, the leader, such that $\mathcal{E} = \{e_1, \ldots, e_{n-1}\} = \{(v_0, v_1), \ldots, (v_0, v_{n-1})\}$. A UGV team can be associated with the graph $\mathcal{G}$ such that each UGV can be cast as a node v_i in the graph and the sensing/constraint mechanism between the robots can follow the rules based on the edge set $\mathcal{E}$. In such a scheme, a UGV can be assigned as the leader to move based on a predefined path in the workspace, while the other robots can follow the leader robot by maintaining the constraints defined by the edge set. Particularly, every follower robot can try to maintain the relative position between itself and the leader robot. Such a graph is called a leader-first follower, and if all follower robots can successfully maintain the desired relative positions, then the UGV team will move by maintaining the formation structure. Such formations are usually useful when only one robot can be equipped with expensive sensors such as a GPS or 3D LIDAR and act as the leader.

In the proposed heterogeneous MRS, the drone is designed to estimate its self position with respect to the UGV team while guiding them for their reliable navigation. To achieve the synergy between the drone and the UGV team, the drone should follow the robots on the two-dimensional (2D) horizontal plane at its operational altitude. We design the new formation such that the drone follows two robots, the leader and a follower, at any time (Figure 2.1). In this scheme, we allow the drone to switch between the midpoints of the robot pairs defined by (v_0, v_i) periodically. Therefore, the drone follows two UGVs, a leader and a first-follower, and thus becomes a second-follower vehicle. The aim of the second follower robot is to follow distances or bearing angles to its two leader robots.

Other formation control graphs can be defined for our framework as well, including the virtual leader approach, minimally persistent graphs, and undirected graphs. For instance, in an undirected graph, if node v_i senses node v_j, then node v_j also senses node v_i. In this thesis, we focus on the leader-first follower formation graph, leaving the application on other graph types as a future work.

3.2 System Framework

The robots can be designed to meet the requirements of the formation graph defined in the previous section. To achieve the coordination mechanism, we design each robot's perception and decision mechanisms in a distributed way. Figure 3.1 demonstrates the block diagram of the entire system, including the sensor outputs and the estimation and control algorithms.

The leader UGV executes exogenous control commands to achieve a set of certain path planning requirements. The main objective for the follower UGVs is to estimate their relative positions toward the leader UGV, and if desired, control the relative position. On the other hand, the drone's task is to measure and broadcast the heading and bearing angles of sub-groups of the UGVs. While doing so, the drone inherently controls its position with respect to the UGV team.

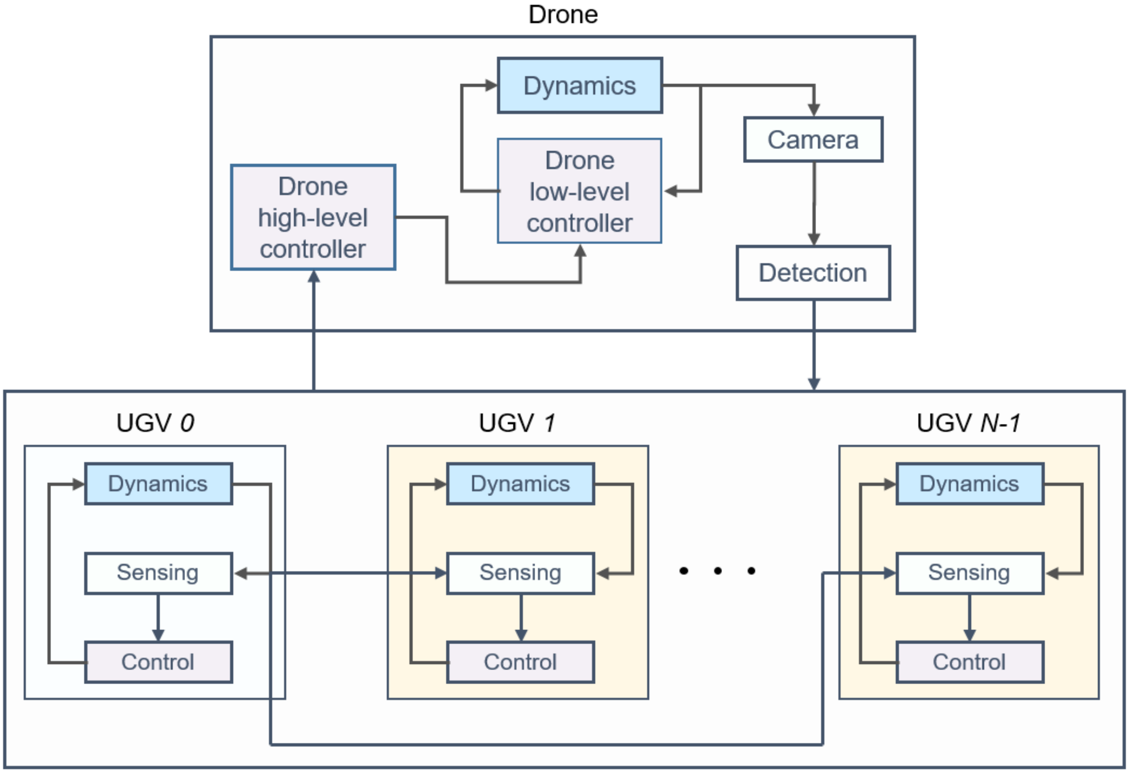


Figure 3.1 Block diagram of the proposed system.

The drone's low-level controller runs on the FCU and possesses the IMU and magnetometer sensors. We assume that the FCU can control well the low-level dynamics of the drone and omit the design of advanced low-level control algorithms. Additionally, a downward-facing mono camera connected to a commercial computational board detects the patterns on the UGVs and measures the heading angles of each detected UGV. Furthermore, the board calculates the bearing angles between R_0 and R_i whenever these two robots are detected by the camera. The calculated quantities are broadcast to the UGVs for the relative position estimations. For this purpose, we mount specific patterns on the UGVs to make them visible and detectable by the drone's downward facing camera. Particularly, we use QR codes which can possess the unique identity of every robot. Whenever the drone detects a pattern on the ground robot, it sends the related quantities (e.g., the heading of the specific UGV and the bearing angle between the leader and the specific UGV) to the UGVs by local communication. We exploit the heterogeneous structure of the MRS and employ the drone to calculate the bearing angles ϕ_i and the heading angles θ_i. During an operation, the drone flies with a fixed heading, i.e., $\psi_k = \bar{\psi}$ for some arbitrary constant $\bar{\psi}$, and serves as the absolute coordinate frame provider for the entire MRS. For convenience, the drone is assumed to fly at a sufficiently high altitude so that it can detect a pair of robots $\{\mathrm{R}_0, \mathrm{R}_\mathrm{i}\}$ with its camera at any time.

To measure the distances between the UGVs, we employ the off-the-shelf UWB sensors which can detect the omnidirectional distances based on a time-of-flight (TOF) measurement model. UWB sensors can be used in several schemes such as multiple anchors-one tag, multiple anchors-multiple tags, and custom designed schemes. In the traditional method, a UWB sensor can be set as an anchor and initiate the communication with the other responder UWBs. In our framework, one cas use a common multiple anchor-one tag model on the UGVs and set the UWB on the leader UGV as the tag and the UWBs on the follower UGVs as anchors. Thus, the distances between the leader and follower UGVs can be obtained. To obtain the distances between the follower UGVs, one can use another custom designed UWB set which produce all-to-all distances based on the TOF method. Alternatively, one can use a set of custom designed UWBs with a mesh network to obtain distances between all UGVs simultaneously.

The UGVs run their motion control algorithms based on either externally commanded signals or a control law. Each follower UGV R_i obtains the heading and bearing values (if available) and its distance to the leader UGV. These sensor data are fused in an estimation algorithm to calculate the relative position $\mathbf{r}_i$ toward the leader UGV.

Our drone travels on ground robots and tries to make bearing and heading measurements in the system we have established. Since each time the leader and a ground robot are flying on it, it can calculate the relative localization of these two robots. So three different situations arise due to the flying of the drone between these robots. First, the drone can detect the leader and a trailing ground robot simultaneously and measure the bearing and heading values. Secondly, when the drone starts flying between another robot and the leader, it does not get any bearing and heading values. In the proposed method, we can measure distance with ultrawideband (UWB) sensors. Therefore, in the third case, if we know the distances of the other robots and the bearing angle value of one, we can find the other bearings to the leader robot. With these bearing and distance values, using the cosine rule, we can calculate the bearing angle values of other robots. Therefore, the values we have are constantly changing, which causes the algorithm we use to differ. For this reason, we have developed three different EKF algorithms. We perform relative localization operations by using these algorithms according to the status of the incoming values.

We emphasize that the proposed framework remains distributed because each robot in the team (UAV or UGV) operates its computational unit by acquiring the measurement values through proprioceptive and exteroceptive sensors. Therefore, the system does not possess a central computational unit. We also note that we focus on the base relative localization and coordination of the entire team and leave the further modifications such as obstacle/collision avoidance to future works.

3.3 Multi-Rate EKF Design

Based on the availability of the heading and bearing values of the UGVs, we have three measurement models to be employed in the estimation algorithm. The multi-rate EKF allows to process distinct sensor readings at each time step and thus is a suitable estimation algorithm for our framework. Here, we design a multi-rate extended Kalman filter (EKF) with different measurement models.

3.3.1 Motion Model

We first derive the motion model of our system. As mentioned in previous chapter, we are interested in a distributed and decentralized system. Therefore, we aim at designing a separate model for each of the robot pairs $\{R_0, R_i\}$ where $i \in \{1, \dots, N\}$. We emphasize that provided that the motion characteristics of all follower UGVs are the same, these models would be identical. Assuming that each UGV is modeled as the nonholonomic kinematics, we propose the following state-space model:

$$\mathbf{x}^i[k+1] = \mathbf{x}^i[k] + \mathrm{g}(\mathbf{x}^i[k], \mathbf{u}[k])T_s \tag{3.14}$$

$$= \mathbf{x}^i[k] + \begin{bmatrix} \mathbf{u}_2[k]\cos(\mathbf{x}_4[k]) - \mathbf{u}_1[k]\cos(\mathbf{x}_3[k]) \\ \mathbf{u}_2[k]\sin(\mathbf{x}_4[k]) - \mathbf{u}_1[k]\sin(\mathbf{x}_3[k]) \\ \mathbf{u}_3[k] \\ \mathbf{u}_4[k] \end{bmatrix} T_s, \tag{3.15}$$

where

$$\mathbf{x}^i = \left[(\mathbf{r}^i)^\top, \theta_1, \theta_0\right]^\top, \tag{3.16}$$

$$\mathbf{u} = [v_i, v_0, \omega_i, \omega_0]^\top, \tag{3.17}$$

and $\mathbf{r}^i = \left[r_x^i, r_y^i\right]^\top \in \Re^2$ is the relative position seen by $\mathrm{R_i}$. Notably, we treat each two-robot system $\{\mathrm{R_0}, \mathrm{R_i}\}$ as one entity with the stacked input vector $\mathbf{u}$. Accordingly, linearizing ((**3.15**) around its equilibrium point, the predicted covariance is updated based on the following linearized motion model:

$$G[k] = \begin{bmatrix} 1 & 0 & \mathbf{u}_1 \sin(\mathbf{x}_3[k])\, T_s & -\mathbf{u}_2 \sin(\mathbf{x}_4[k])\, T_s \\ 0 & 1 & -\mathbf{u}_1 \cos(\mathbf{x}_3[k])\, T_s & \mathbf{u}_2 \cos(\mathbf{x}_4[k])\, T_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.18}$$

Thus, the multi-rate EKF algorithm uses (3.15)-(3.18) in the prediction step. The UGVs obtain the linear and angular velocities from the encoders on the wheels, and the control signals are generated by a control algorithm.

3.3.2 Measurement Models

Each follower R_i can obtain a subgroup of the four sensor readings: UWB distance to the leader ($d_{0\mathrm{i}}$), self heading angle (θ_i), the leader's heading angle (θ_0), and the bearing toward the leader (ϕ_i). Therefore, a complete observation model consists of the distance and bearing angle between $\mathrm{R_0}, \mathrm{R_i}$ together with the heading angle measurements, as follows:

$$\mathbf{y}^{i,1} = [d_{0\mathrm{i}}, \phi_\mathrm{i}, \theta_i, \theta_0]^\top, \tag{3.19}$$

where $d_{0\mathrm{i}} = \| \mathbf{r}^i \|$ and $\phi_\mathrm{i} = \operatorname{atan}(r_y^i, r_x^i)$.

Based on the drone's perception, a follower may or may not obtain all four measurements defined in (3.19). Particularly, there are three cases regarding the acquisition of the measurements. The first case corresponds to acquisition of all measurements $d_{0\mathrm{i}}, \phi_{\mathrm{i}}, \theta_i, \theta_0$. This case occurs when the drone hovers on top of the midpoint between the leader R_0 and the follower R_i. While we have these data, we calculate relative localization by running the following EKF algorithm. Hence, the measurement model for the first case is identical to (3.19). The linearized measurement model in this case can be obtained by linearizing (3.19) around its equilibrium as follows:

$$H = \begin{bmatrix} \frac{r_x^i}{d_{0\mathrm{i}}} & \frac{r_y^i}{d_{0\mathrm{i}}} & 0 & 0 \\ -\frac{r_y^i}{d_{0\mathrm{i}}^2} & \frac{r_x^i}{d_{0\mathrm{i}}^2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.20}$$

The second case for R_i occurs when a drone hovers around the midpoint between the leader R_0 and another follower R_j. We can calculate the other robots' bearing to the leader robot using the cosine rule if we have one bearing data. However, the drone cannot measure the UGVs heading because the drone cannot see other robots' QR codes. So in that situation, we have to write another EKF measurement model to measure relative locations. Therefore our state vector and motion model is the same equation with (3.16) and (3.17). And here is the measurement model:

$$H = \begin{bmatrix} rx/d & ry/d & 0 & 0 \\ -ry/d_sqr & rx/d_sqr & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.21}$$

Measurement Model:

$$y_i(t) = \begin{bmatrix} \sqrt{(R_{ix} - R_{ox})^2 + (R_{ix} - R_{ox}^2)} \\ \mathrm{atan2}(R_{iy}, R_{ix}) \end{bmatrix} \tag{3.22}$$

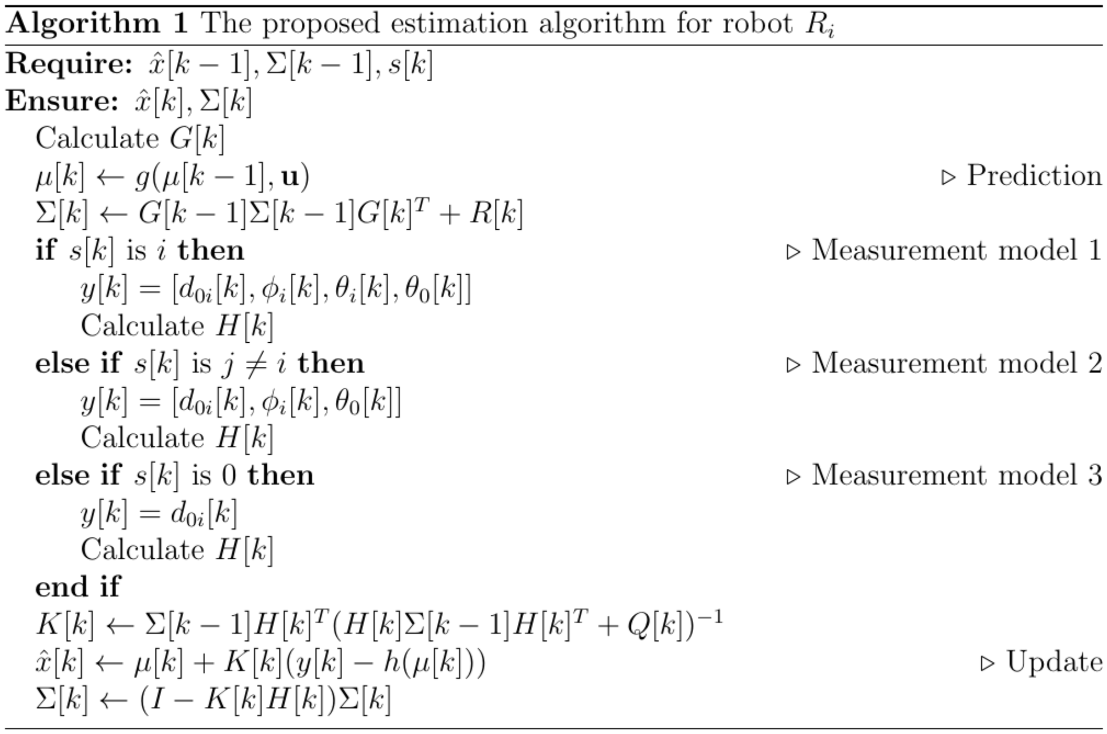
In order to find the relative localization of the leader and other robots, the drone must constantly fly over other robots for a specific time. The drone's inability to detect barcodes during the displacement movement causes it not to receive heading and bearing data. For this reason, we have to use a different EKF algorithm during this time. Although we do not have bearing and heading data, we obtained a distance from UWB sensors. For this reason, we only use distance data in our EKF algorithm this time. Therefore, the third measurement model of EKF becomes:

$$y_i(t) = \left[\sqrt{(R_{ix} - R_{ox})^2 + (R_{ix} - R_{ox}^2)}\right] \tag{3.23}$$

$$H = \begin{bmatrix} \frac{r_x^i}{d_{0\mathrm{i}}} & \frac{r_x^y}{d_{0\mathrm{i}}} & 0 & 0 \end{bmatrix} \tag{3.24}$$

We now give the pseudocode for the multirate EKF algorithm in Table 3.1. The algorithm requires the state estimate $\hat{x}_i[k-1]$ and the switching signal $s[k]$ denoting the measurement model at time instant k. If the drone detects the leader R_0 and the follower R_i at time k, we have $s[k] = i$, denoting that the current heading θ_i, θ_0 and the bearing ϕ_i values are available and broadcast by the drone. Then, the other follower robots $R_j, j \neq i$, uses the second measurement model where the bearing ϕ_j is calculated using geometric rules. Notably, in the second measurement model, the heading θ_j is not available to robot R_j. Finally, if $s[k] = 0$, all follower robots run the third measurement model where the only available measurement is the distance d_{0i}. Remarkably, the drone needs to calculate the matrix H according to the drone's sensor measurement and the switching signal. We also show the prediction and update steps of the algorithm in Table 3.1.

Table 3.1 The proposed estimation algorithm for robot Ri



```
Algorithm 1 The proposed estimation algorithm for robot $R_i$
Require: $\hat{x}[k-1], \Sigma[k-1], s[k]$
Ensure: $\hat{x}[k], \Sigma[k]$
  Calculate $G[k]$
  $\mu[k] \leftarrow g(\mu[k-1], \mathbf{u})$                          ▷ Prediction
  $\Sigma[k] \leftarrow G[k-1]\Sigma[k-1]G[k]^T + R[k]$
  if $s[k]$ is $i$ then                                     ▷ Measurement model 1
      $y[k] = [d_{0i}[k], \phi_i[k], \theta_i[k], \theta_0[k]]$
      Calculate $H[k]$
  else if $s[k]$ is $j \neq i$ then                         ▷ Measurement model 2
      $y[k] = [d_{0i}[k], \phi_i[k], \theta_0[k]]$
      Calculate $H[k]$
  else if $s[k]$ is $0$ then                                ▷ Measurement model 3
      $y[k] = d_{0i}[k]$
      Calculate $H[k]$
  end if
  $K[k] \leftarrow \Sigma[k-1]H[k]^T(H[k]\Sigma[k-1]H[k]^T + Q[k])^{-1}$
  $\hat{x}[k] \leftarrow \mu[k] + K[k](y[k] - h(\mu[k]))$              ▷ Update
  $\Sigma[k] \leftarrow (I - K[k]H[k])\Sigma[k]$
```

3.3.3 Distance-based Bearing Calculation

When the drone flew between two of the leader and follower robots, we could find the bearing angle of the follower robot to the leader using barcodes. We have determined that other robots do not need to be seen with a drone to calculate the bearing angle values to the leader robot. We can measure distances between robots with UWB sensors. In the figure Figure 3.2, the following robots move to form a triangle with the leading robot. For this reason, we can determine other bearing angle values in this triangle where we know their lengths and one bearing angle value. We used the Cosine Rule that can be presented with Figure 3.3 A triangle with sides and angles., a trigonometry rule, for the calculation..

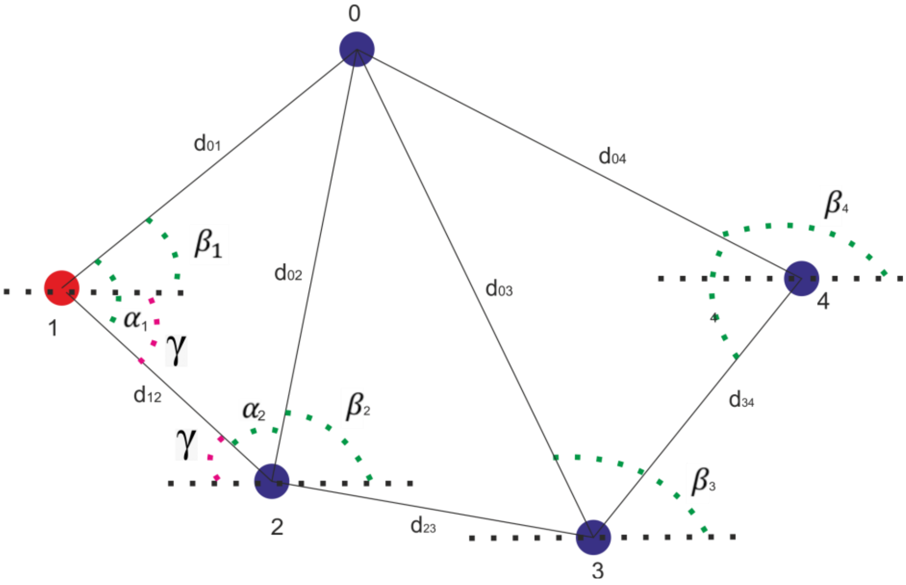


Figure 3.2 Bearing angles of robot team.

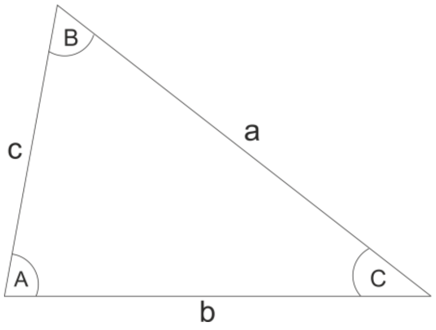


Figure 3.3 A triangle with sides and angles.

For finding sides with cosine rule:

$$a^2 = b^2 + c^2 - 2bc\cos(A) \tag{3.25}$$

For finding angles:

$$\cos(A) = \frac{b^2 + c^2 - a^2}{2bc\cos(A)} \tag{3.26}$$

We have positions d_{01}, d_{02}, d_{12} between ground robots. Using the equation (3.25) we can find α_1. We already have the bearing angle of the first follower robot R_1 to leader robot R_0.

$$cos(\alpha_1) = \frac{d_{01}^2 + d_{12}^2 - d_{02}^2}{2d_{01}d_{12}} \tag{3.27}$$

From these equation we can get $alpha_1$ and $alpha_2$ as:

$$\alpha_1 = arccos\left(\frac{d_{01}^2 + d_{12}^2 - d_{02}^2}{2d_{01}d_{12}}\right) \tag{3.28}$$

$$\alpha_2 = arccos\left(\frac{d_{02}^2 + d_{23}^2 - d_{03}^2}{2d_{02}d_{23}}\right) \tag{3.29}$$

$$\gamma = \alpha_1 - \beta_1 \tag{3.30}$$

So we can find the second follower robot’s bearing to the leader with (3.31) equation:

$$\beta_2 = \pi - (\alpha_2 + \gamma) \tag{3.31}$$

Chapter 4

Simulation

We performed the simulation part of our study using Robot Operating System (ROS) and Gazebo environment. We used Rosbot for the simulation of ground robots and iris drone models as aircraft. In this chapter, we explain the simulation environment, the robotic system setup, operational procedures, and the simulation results with a discussion.

4.1 Simulation Environment

ROS (Robot Operating System) is a Linux-based licensed software system that allows communication of several robot components. ROS aims at developing software that can work on different robots by making minor changes to the code. Thanks to ROS, a standard algorithm written for a standard robot will be run at load-and-play speed. A ROS system consists of nodes which run based on the publish/subscribe messaging model. Many nodes can exist within a ROS system. Nodes are operations that can perform computation. For instance, a node can receive the image from a camera sensor, another node can process the image, and another node can display the image. A ROS master manages the nodes so that they can communicate with each other.

We used an Iris drone to simulate a UAV in the Gazebo environment powered with ROS and a nonholonomic ground robot Rosbot for UGV simulation. We integrated an RGB camera with the size of 800*800 pixels to the drone, facing downward. On the Rosbot UGVs, we placed two barcodes with different content detected from the drone camera. We have ensured that the barcodes on each robot are different. This difference will also help us identify and determine direction angles. The UGV with barcodes and the UAV with a camera can be seen in the simulation in Figure 4.1.

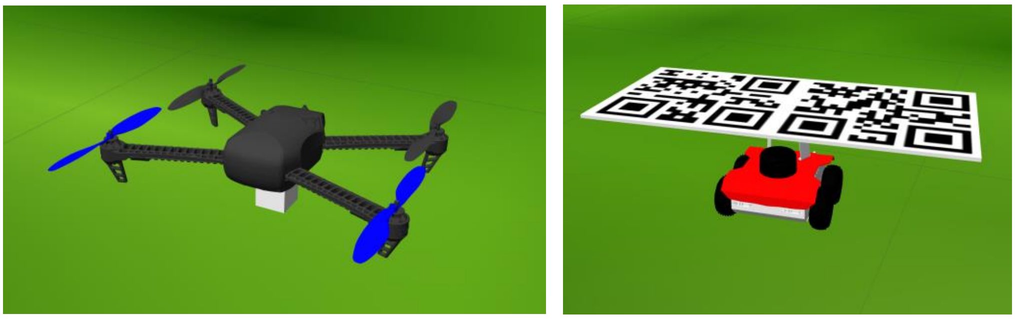
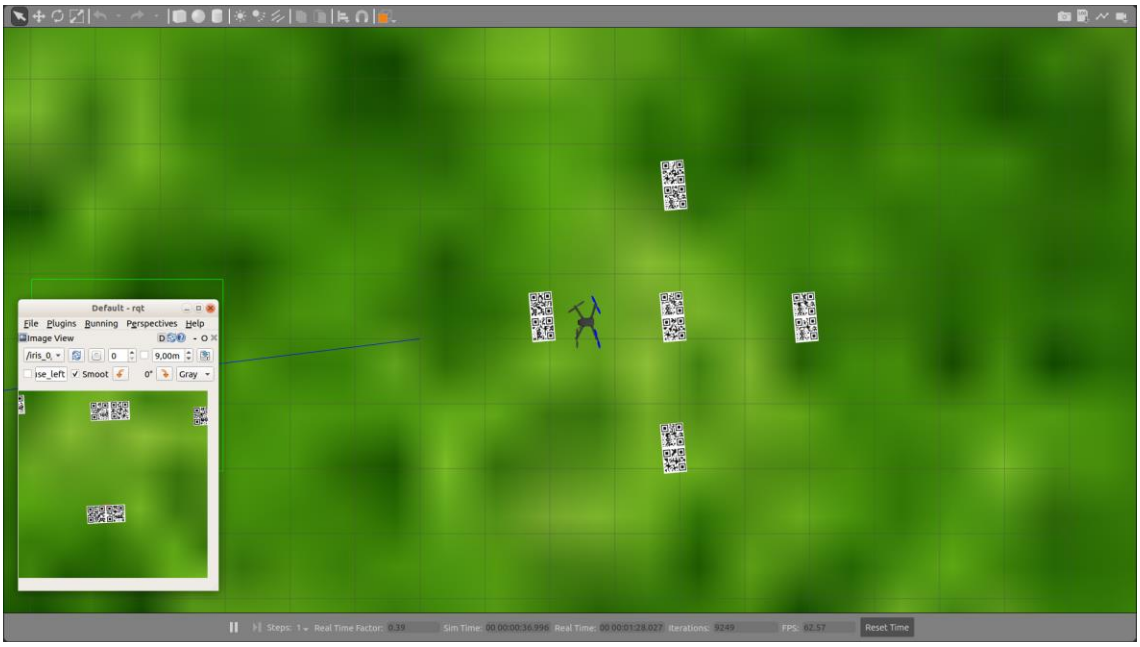


Figure 4.1 Iris drone model (left) and Rosbot model (right) in Gazebo. We mount specific patterns on the ground robots for detection by the drone's camera.

4.2 Formation Model

In our study, we organized five robots in the leader-first follower formation shape (Figure 4.3). The first robot R_0 is chosen as the leader ground robot. The other four robots are followers. We manually sent speed commands to the leading robot. A sample image from the simulation environment with five UGVs and a drone are shown in Figure 4.2 below.



Steps: 1 Real Time Factor: 0.39 Sim Time: 00 00:00:36.996 Real Time: 00 00:01:28.027 Iterations: 9249 FPS: 62.57

Figure 4.2 A top view of an example simulation in ROS Gazebo. The drone's camera view is shown in the bottom-left corner.

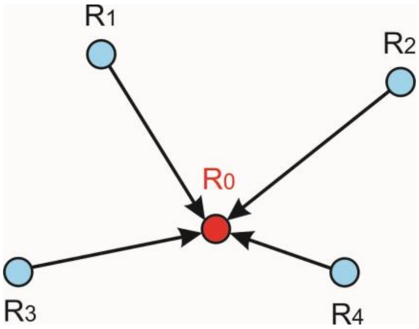


Figure 4.3 Leader-first follower graph.

4.3 Pattern Detection

We placed a QR code on the ground robots and gave unique identity to each code. We used the Pyzbar library for the detection of QR codes by the drone camera, which allowed to detect multiple code patterns simultaneously. The Pyzbar library allows to find the x, y coordinates, width, and height values in the camera frame by drawing a square around the codes.

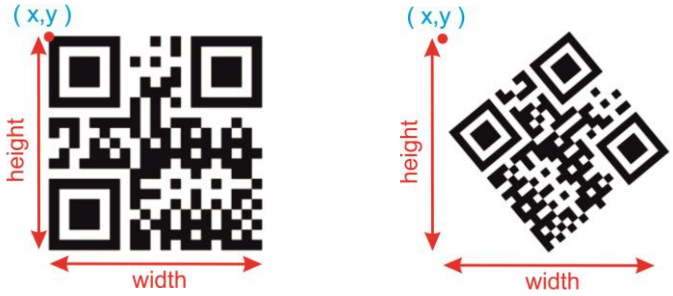


Figure 4.4 QR code detection: (Left) Not rotated; (right) rotated.

Figure 4.4-left shows a QR code and the determined x, y, width, height values expressed on the figure. As shown in Figure 4.4-left, if the QR code is not rotated, we can find its exact height and width. However, as the ground robot moves, the orientation of the attached QR code changes. For this reason, the x, y, width, and height values of a rotated QR code must also be determined. Oriented QR codes can be determined by the Pyzbar library as well (Figure 4.4-right). However, the x and y coordinates here are not exactly the coordinates of the upper left corner of the QR code. Similarly, the height and width values do not represent the actual height and width values of the QR code.

We have previously stated that our study is built on the orientations of ground robots. For this reason, we cannot detect whether the ground robot has changed direction with a single barcode. Therefore, we placed two barcodes on every ground robot. Since we can detect barcodes with different contents simultaneously, we can detect two QR codes and their coordinates, height, and width information separately. By detecting two codes, we were able to calculate the center points of both QR codes. Furthermore, thanks to the angles of these calculated center points to each other, we were able to find the robot's orientation with the camera of the drone.

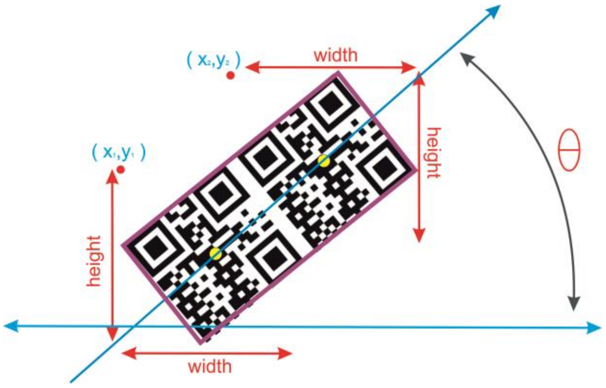


Figure 4.5 QR codes detection with rotated state.

We found the center points of both of these QR codes on the ground robot. At the same time, two robots are viewed simultaneously with a drone camera. The midpoint of these two center points gives us the center point of the robot. This method allows us to detect more than one robot's orientation and center points with the drone camera. Using these central points, we can easily calculate the bearing angle value.

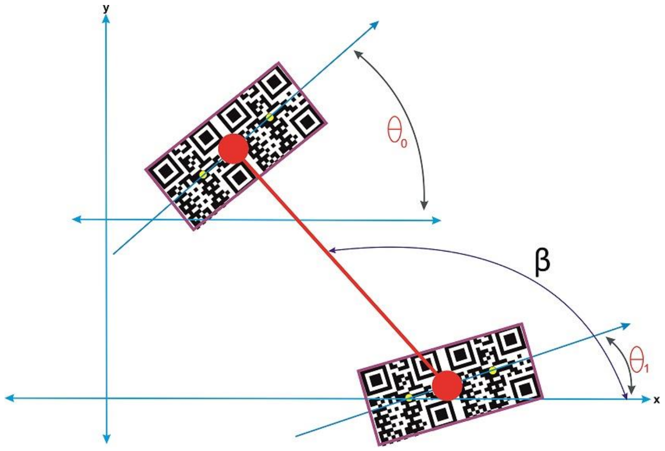


Figure 4.6 Camera frame and center of two ground robot. θ_0, θ_1 are heading, β is the bearing angle.

We find the bearing and heading angle values according to the camera frame of the drone. The drone moves on the ground robots so that the yaw angle is fixed and constantly facing north. Therefore, we assume that the directions of the x and y coordinates of the camera frame coordinate plane do not change. Moreover, according to this coordinate plane, we calculate the drones' heading and bearing angle values.

In order to find the bearing angle values of the two ground robots, we fly the drone between the leading robot and one of the following robots for a certain period. The drone is set at the height of 2 meters to read QR codes better. If we reduce the height below 2 meters, we prevent us from seeing the ground robots simultaneously. In this way, we cannot detect the bearing angle value. When we climbed to heights of more than 2 meters, we saw that the detection rate of QR codes decreased. Therefore, we ideally used a height of 2 meters. We assumed the midpoint of the camera frame as $x = 0$, $y = 0$ points. In order for the drone to follow both ground robots and the frame to come to this point (0,0), we moved the drone so that the yaw angle remains constant in the x and y coordinate planes. We calculate the amount of this movement by measuring the distance from the center of the frame to the midpoints of the robots. We convert the x and y distances from the robots to the midpoint of the frame into velocity values. In this way, the drone manages to get the angle values in the determined time by going towards the middle point of the robots.

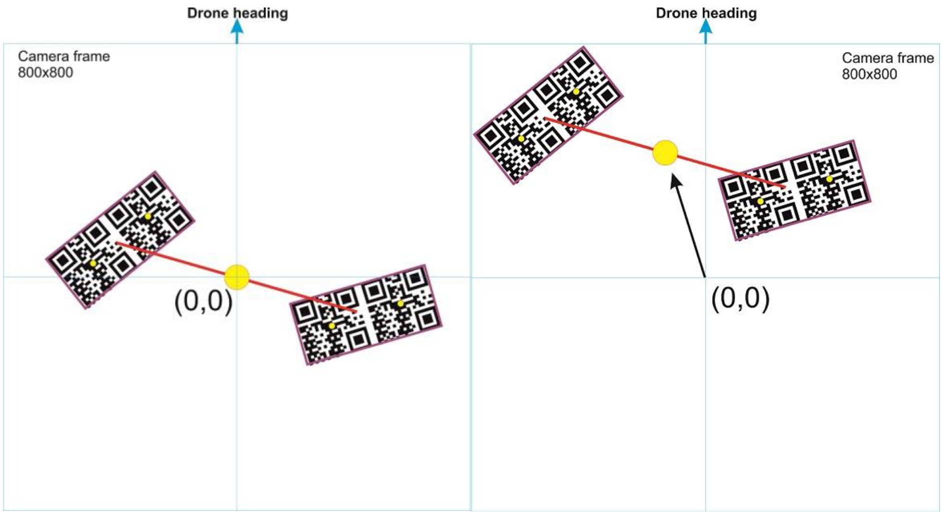


Figure 4.7 Movement of the drone towards the center point of ground robots.

To calculate the speed of the drone, we need to convert the pixel distance to linear speed. If the distance to the center point is d_x and d_y, then our velocity in the direction x is $v_x = d_x/310$, and our speed in the direction y is $v_y = d_y/310$.

It was not possible to use UWB sensors in the simulation environment. For this reason, we obtain the distance between the robots in the simulation environment with the topic of gazebo/modelstates. In this service, the position x, y, z information of the objects in the simulation environment is presented. After obtaining the position information of the ground robots, we can easily find the distance between them.

After detecting the leader and a follower of the ground robots with the drone camera, we can find the bearing angle value of the follower robot to the leader robot. After knowing the distances between the robots, we can determine the bearing values of the leader in the remaining robots by using the cosine rule. In Figure 3.2, we can see the arrangement of the ground robots and the drawing of the angles.

4.4 Simulation Results

Our simulation experiments consisted of the following stages:
Placing the QR codes on the ground robots,

- Reading the QR codes from the drone camera,
- Measuring the heading angles of the robots,
- Arranging the movement of the drone on the mobile robots,
- Measure the bearing angle values for each pair of leader and follower robot,
- Performing the relative localization operations with three different EKF algorithms,
- Collecting and analyzing the data.

After passing many stages, we will explain the results we obtained from the data collected in the simulation study.

In order to analyze the results we obtained, a graph showing the x and y positions come from EKF algorithms and graphs showing the heading and beta angles were plotted. In Figure 4.8, we see the displacement of five ground robots and one drone in approximately three minutes in the simulation. Ground robots take action at a certain speed. The figure in dark blue represents the robot that is the leader. The other four ground robots in red, yellow, purple, and green are the following robots. The robot, which can be seen moving in different positions in light blue, shows the aircraft. As can be seen from the figure, the ground robots follow the leader robot at an equal speed and angle. We see that the aircraft is constantly changing its place. We mentioned earlier that the drone should see a leading robot and the following robot simultaneously and read their barcodes. For this reason, the drone always sees the leading robot and moves in a way that takes the following robots in line of sight. In the simulation we made between the leader and the following robot, the drone flies for 20 seconds. It calculates the bearing values in the meantime. It then moves towards another follower robot. That is why we programmed the drone to fly so that it constantly changes places by flying over the robots.

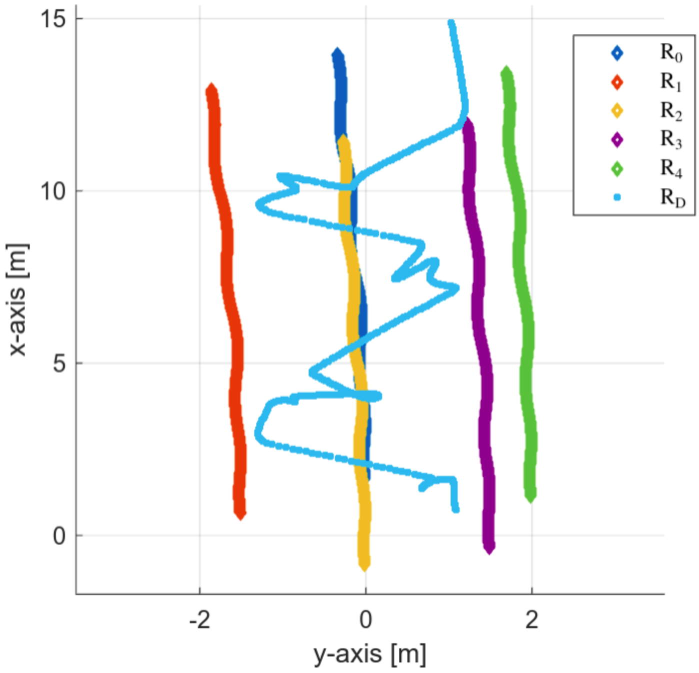


Figure 4.8 Ground robots and drone pozitions in simulation.

In Figure 4.9Figure 4.9, we see the real relative locations of the ground robots and the relative position x values calculated with our EKF algorithms. r_{40} represents the actual value of x distance between robots R_0 and R_4. As can be seen in the graph, r_{40}^{est} refers to the first pair of leading followers that the drone runs an EKF algorithm using the bearing, distance, and heading values. Therefore, it is much more stable and closer to the true value than other r_{10}^{est}, r_{20}^{est}, r_{30}^{est} estimations. Afterward, we can see that the drone flew between the R_0 and R_3 robots and measured the relative x position. It can be seen that the calculation is made for the other two robots and estimation is much more stable than before. We see that the drone then stabilizes the calculations by flying between R_0 and R_2, then R_0, R_1. In our EKF algorithm, the accuracy rate of the estimation is in the range of +-5 centimeters for the relative x position. The EKF algorithm becomes more stable when the drone comes between the robot pairs due to the variety of values used in the algorithm. We calculated with distance only, distance and bearing only, and distance, bearing, and heading values in our three different EKF algorithms. The values we use in our EKF algorithm increase; we can see from the graph that the stability in the calculations increases.

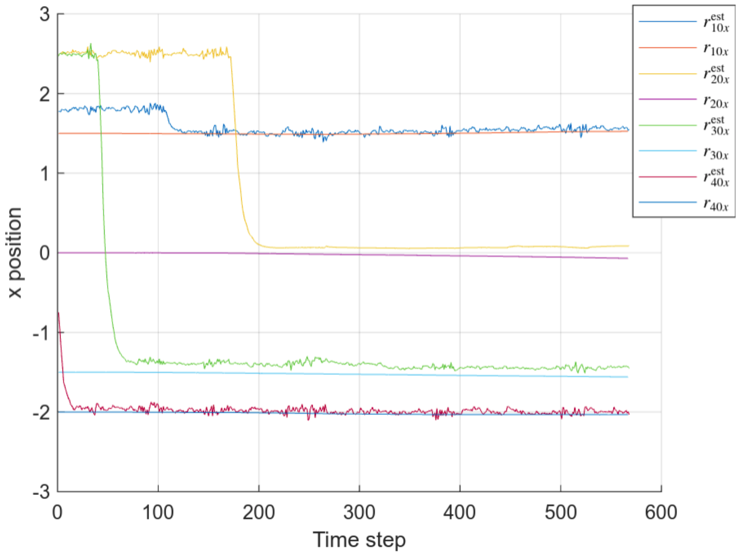


Figure 4.9 Exact relative x positions, and EKF estimation for x relative positions of the ground robots.

In Figure 4.10, we see the relative y position calculated by the EKF algorithm and their actual values plotted. The drone first flies over the R_0 and R_4 robots and uses the EKF algorithm, including distance, bearing, and heading values. In this way, we can see that the position values become stable faster than the others. Then we perform relative localization operations by flying over R_0 and R_3 , R_0 and R_2 , R_0 and R_1 . After a short while, we can see from Figure 4.10 that all calculated values approach nearly the real values. It can be seen that the error rate calculated for the relative y location is +-5 centimeters, almost at the same rate as that calculated for the same x relative position estimation error.

In our application, we also tried to calculate the heading angles of the ground robots. The drone flies constantly facing north. We explained whether the ground robots changed their direction or not by finding the center points of the double QR code on them. In this way, we used the heading values from the drone in our EKF algorithm. In Figure 4.11, we can see the comparison of the actual heading values and the calculated heading values. Again, we can see the same estimation order in the relative x and y values. As can be seen in Figure 4.11, our EKF algorithm achieves excellent success by calculating almost close to actual values.

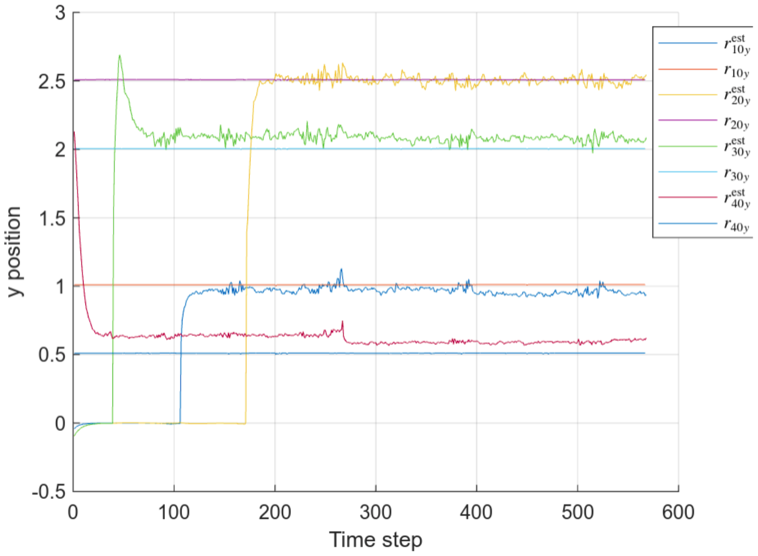


Figure 4.10 Exact relative y positions, and EKF estimation for y relative positions of the ground robots.

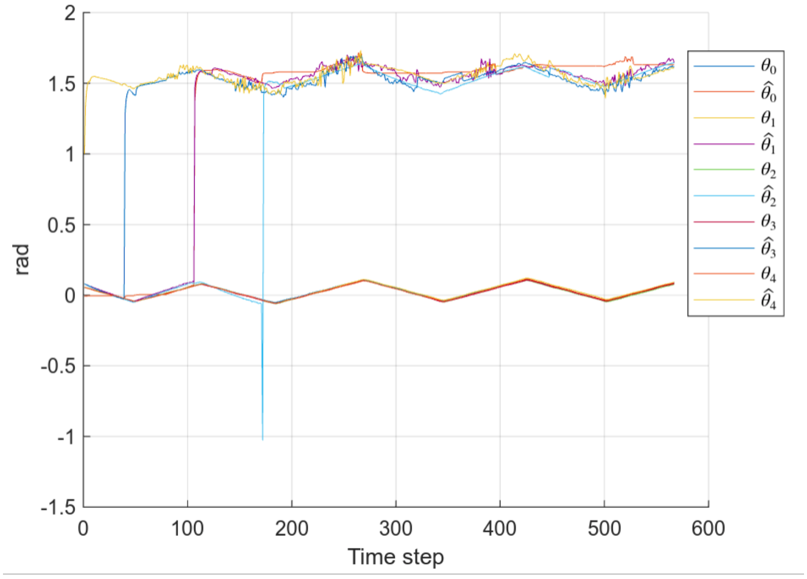


Figure 4.11 Exact heading values and EKF estimation headings of the ground robots.

We stated that when we calculate the bearing value of a follower robot and a leading robot, we can calculate the beta values of the others with the cosine rule. at eq (3.26), we see the beta values we calculated with this method.

We also moved our robots in the form of an arc in the simulation. In Figure 4.12, we see the actual positions of the robots in the simulation environment. In Figure 4.13, we see the x position values obtained by EKF algorithms and the absolute x values together. Likewise, in Figure 4.14, we can see the comparison of the y values calculated by the EKF algorithms and the actual y values of the robots. The comparison of the heading values can be seen in Figure 4.15.

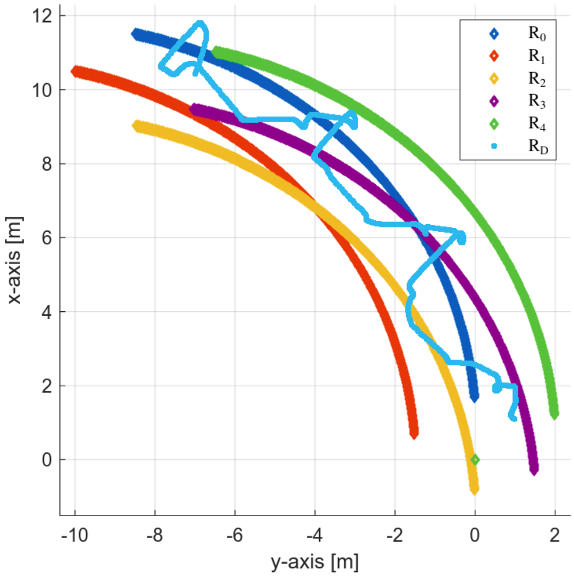


Figure 4.12 Arc movement real robot states.

We can see from the figures that when we move the robot team in the form of an arc, we get results very close to the results we get when we move it straight.

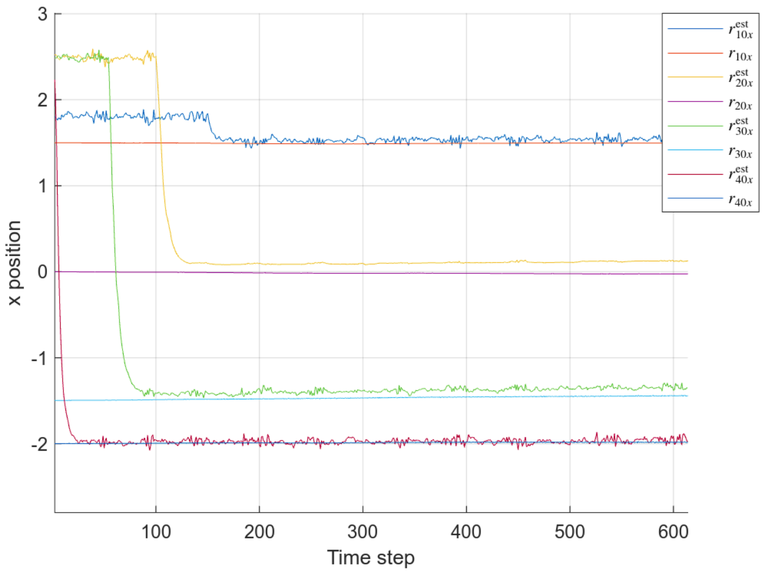


Figure 4.13 Arc movement x position estimation and ground truths.

In another simulation we did the same simulation with two drones. In Figure 4.16 we can see the simulation environment. The first drone is programmed to fly over the first and second robots with the leader robot. Our second drone was programmed to fly above the third and fourth robots. Unlike the simulations we made with a single drone, we aimed to obtain more heading and bearing angle values here. In the simulations using a single drone, we did not have a second alternative to obtain an angle in cases where we could not detect the QR code at times. However, thanks to two drone s, we are able to obtain more angle values by transferring information between drones.

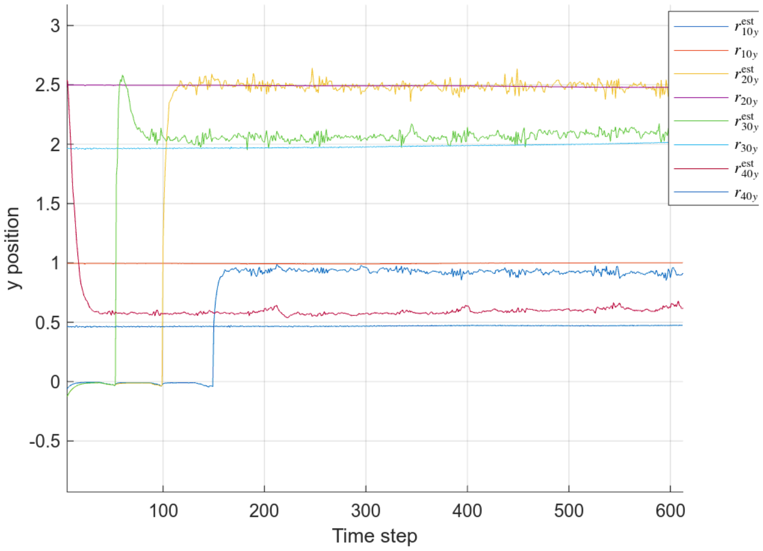


Figure 4.14 Arc movement *y* position estimation and ground truths.

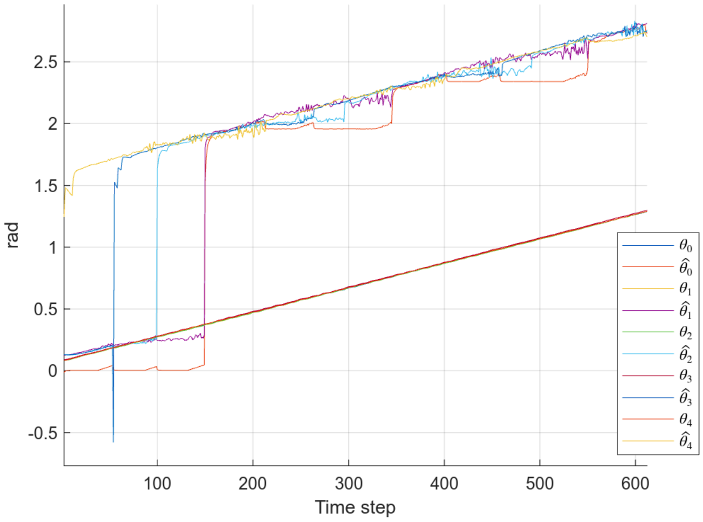


Figure 4.15 Arc movement heading estimation.

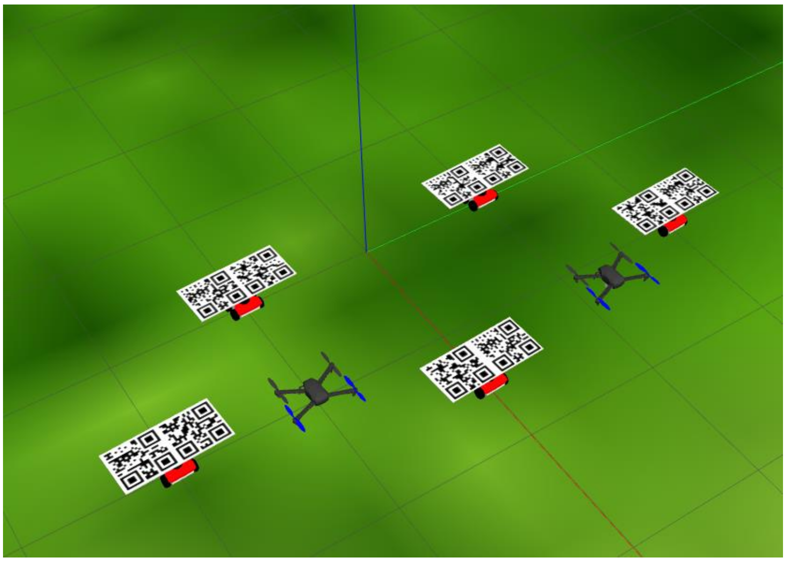


Figure 4.16 Simulation environment for two drone and five ground robots.

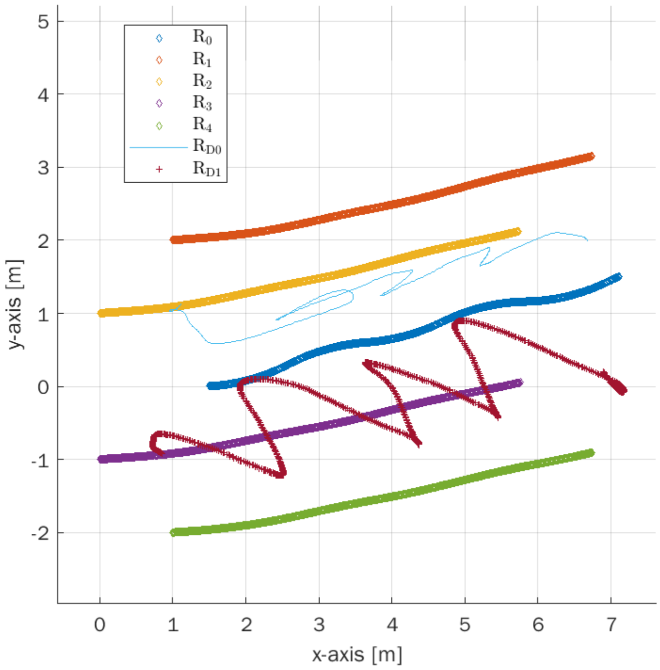


Figure 4.17 Ground truth for robot team in S-path movement simulation.

In Figure 4.17 we can see the ground truth oh the robot team. Figure 4.18 shows first and second following robots estimated values and thein ground truths. And Figure 4.19 shows the same results for third and fourth robots. As a result of using two drones, we can see that we estimate position values that are close to real values with a very small error rate.

In Figure 4.20 shows the estimation results of headings of the ground robots. We can see that the values sometimes diverge from the real values. This is because drones cannot always detect heading values. However, when we look at the graph, it can be seen that our error rate in the calculation is extremely small.

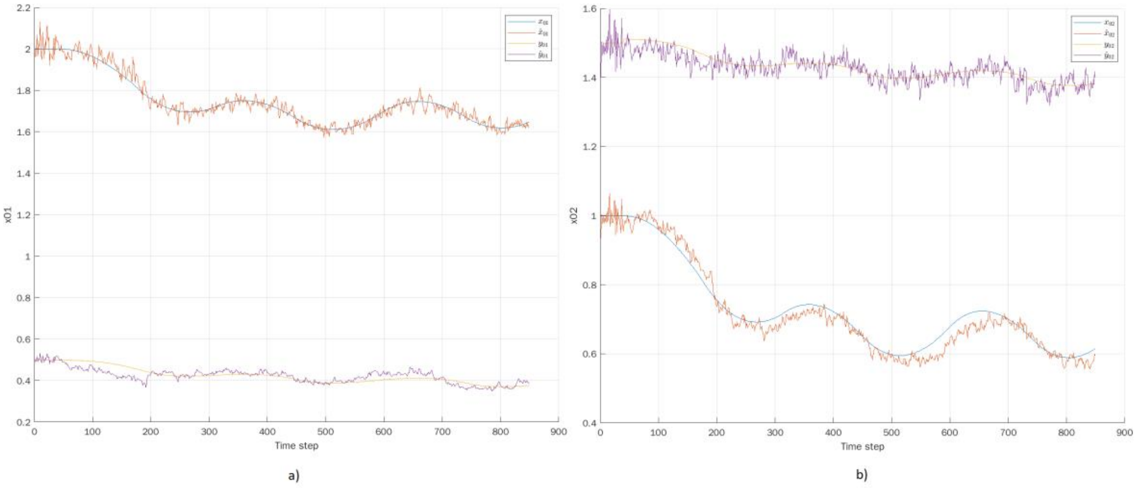


Figure 4.18 First and second robots relative x and y positions to the leader.

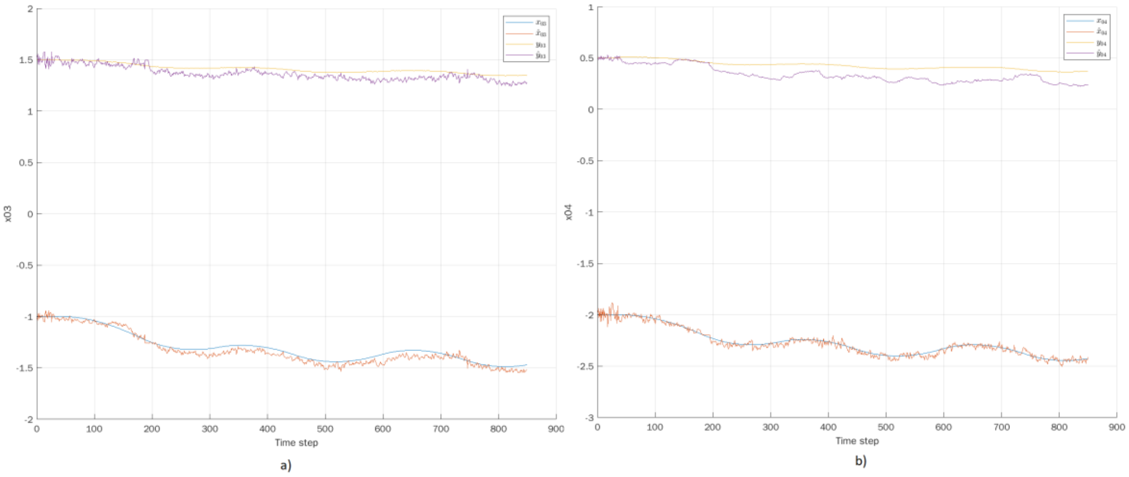


Figure 4.19 Third and fourth robots relative x and y positions to the leader.

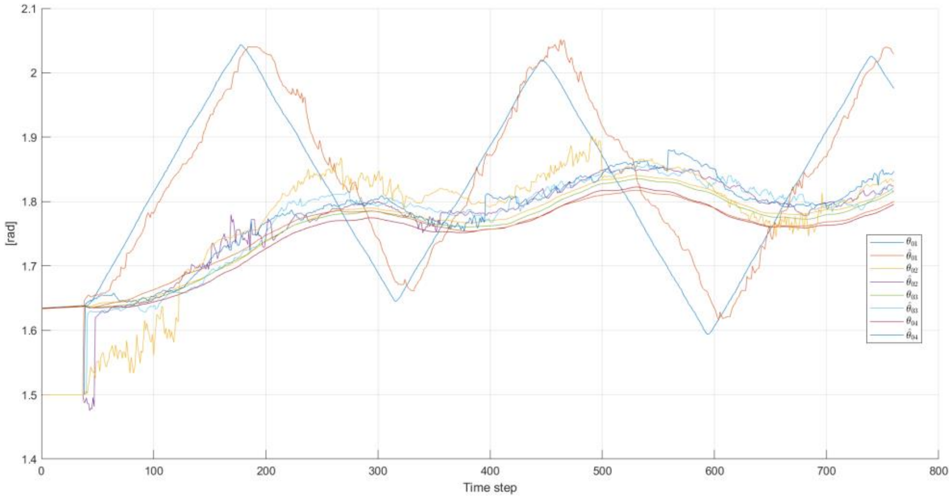


Figure 4.20 Heading estimations of ground robots and ground truths.

Chapter 5

Experiment

5.1 Experiment Setup

We performed a series of tests in our laboratory environment. We used Hexsoon f450 Frame as the drone, as shown in Figure 5.1. As the flight control card, we preferred the Pixhawk Cube 2.1, which is in Figure 5.2. The Pixhawk flight controller is based on "APM", an open-source project from the rapidly developing and refined Ardupilot mega or 3DR robotics. This flight controller is a fixed-wing, rotary-wing, or enables it to transform multi-wing vehicles (even boats and cars) into a fully autonomous vehicle. It is able to maintain autonomous stabilization, waypoint-based navigation, radio telemetry modules, and two-way telemetry. Pixhawk Cube is a modern autopilot system with a simple hardware interface, and Pixhawk Cube provides all the expected functionality. The Cube has the 80pin DF17 connector. It allows the Cube to be seamlessly connected to an existing array of carrier boards. It separates the interfaces into a standalone plug-and-play autopilot system. It can also be used by OEMs to design carrier boards for integrated drone solutions. In addition to its advanced features, Pixhawk has been our choice due to its easy installation and high processing power.

We used Raspberry pi 4 as the on-board computer (Figure 5.3). Raspberry Pi 4 is a small DIY computer with 28nm based 1.5G Quad-Core CPU and 4Gb DRR4 RAM. To accomplish PC-like capabilities, Raspberry Pi 4 features 4K Micro HDMI, USB 3.0, BLE Bluetooth 5.0, dual-band 2.4/5.0 GHz Wireless LAN, a USB-C power jack, and True Gigabit Ethernet compatible with PoE. In order to detect QR codes on ground robots, we placed a Raspicam v2.1 camera on the drone. The Raspberry Pi high-resolution camera is compatible with all models with a CSI connector on it. This camera can also be used for photography and HD resolution video capturing or as a webcam. Our project used the Raspicam camera with a frame width of 600 * 480 to detect QR codes. And the final drone model which we created can be seen in Figure 5.1.

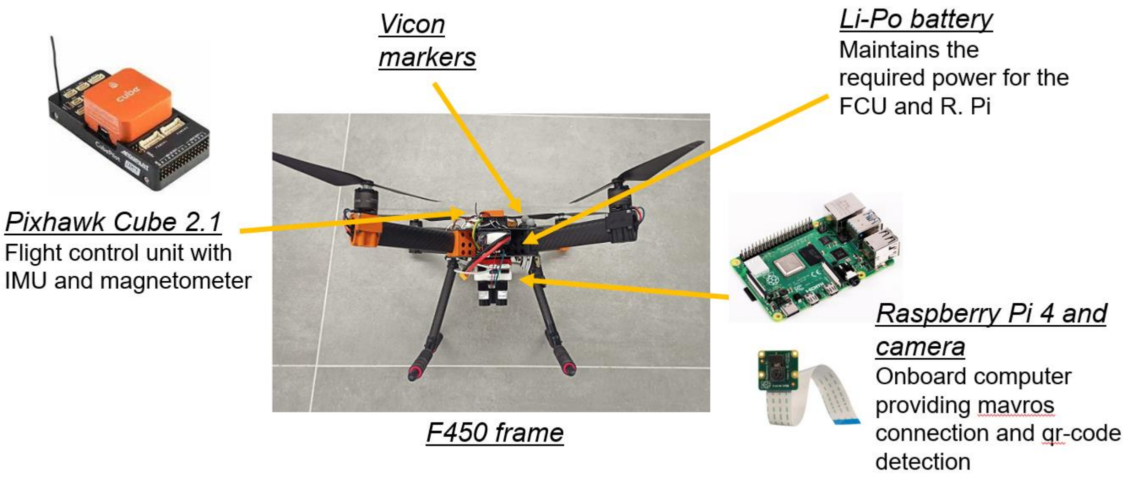

Vicon markers
Li-Po battery
Maintains the required power for the FCU and R. Pi
Pixhawk Cube 2.1
Flight control unit with IMU and magnetometer
Raspberry Pi 4 and camera
Onboard computer providing mavros connection and qr-code detection
F450 frame

Figure 5.1 The drone frame (F450) and onboard components.

We can see the drone model we created by combining the equipment mentioned in Figure 5.1. The grey markers on the frame are used to detect the drone by seven VICON motion capture cameras. When at least three cameras send the detected marker information to a ground station, a custom software on the ground station calculate the robot’s position and broadcast through wi-fi with more than 100 Hz data rate. Then, the drone uses this location information in its low-level position control. We also installed markers on the ground robots and used this location information as the ground truth data to analyze the performance of our algorithms.

We carried out our experiments with three ground robots: Two Rosbots and one Turtlebot. Both Rosbot and Turtlebot are ROS-based mobile platforms (with Turtlebot being a cheaper option) that can be used for many purposes by adding different sensors (motion, temperature, color, contact). The TurtleBot3 burger has 2D LIDAR to implement SLAM (simultaneous mapping and localization) and autonomous navigation algorithms and is controlled via Raspberry Pi. When required, their mechanical parts can be replaced, e.g., a high-speed computational board can be installed on them instead of the Raspberry Pi board (Figure 5.1). On the other hand, Rosbot is an autonomous, non-holonomic robot with bigger and more powerful actuators and tires. It can consist of a LIDAR and an RGB-D camera besides its IMU, encoders, distance sensors (Figure 5.3). We tried to explain our robot team, which we use in our seas, with its essential features. In Figure 5.4, we can see our robot team together.

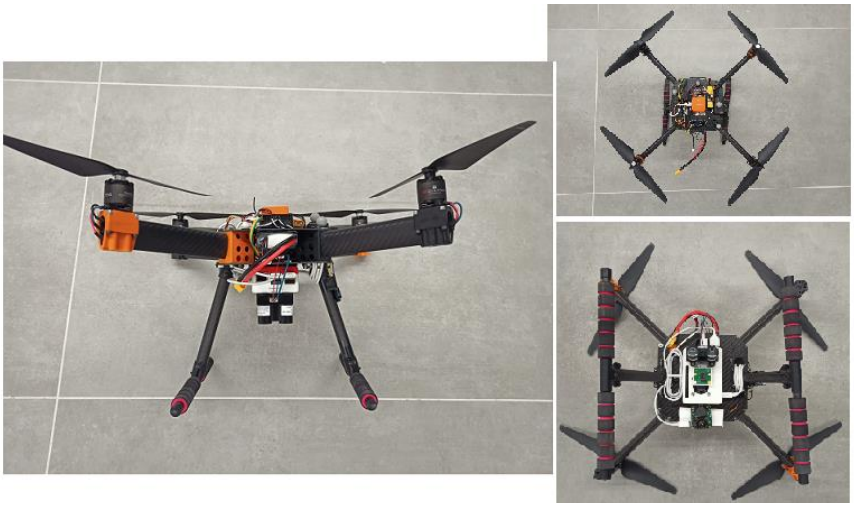


Figure 5.2 Drone model

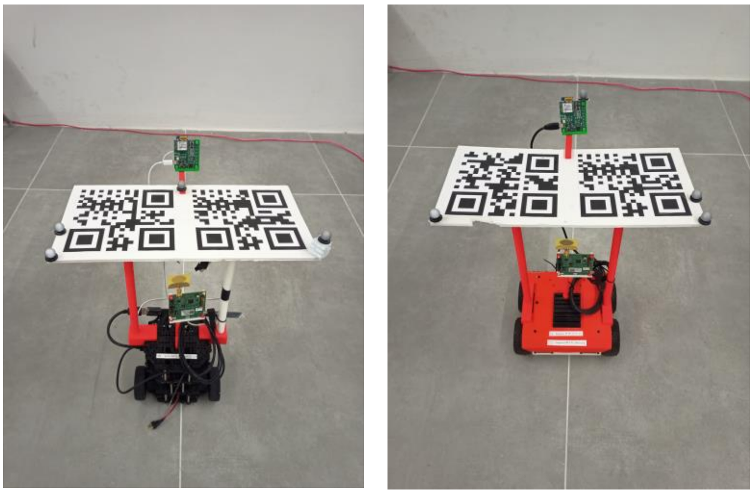


Figure 5.3 Turtlebot Burger and Rosbot 2.0

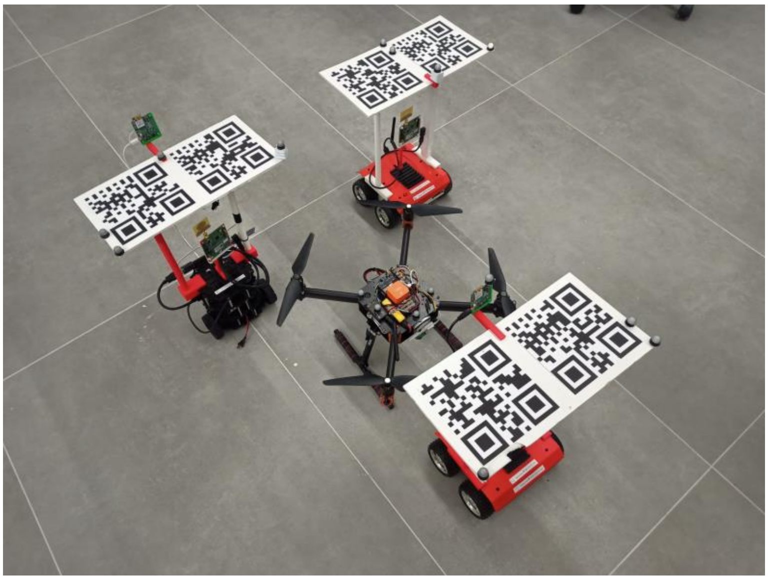


Figure 5.4 Robot team consist of one dron two Rosbot and one Turtlebot

5.2 Results

In the first part of our work, we moved the robots straight, and set the angular velocity to zero. We assigned a Rosbot robot as the leader and set it as the ROS master which provided the Ros core module for the entire system. We controlled the other two robots to follow the leader robot at the same speed. Unlike the simulation, we reduced the drone's tracking time for ground robots to four seconds (note that this was set to 20 seconds in simulations). In this way, the drone moved much faster between the leading and following binary robots and detected QR codes in more time. Thanks to this, we have used our EKF algorithm, which we operate with distance, bearing, and heading, for a more extended period.

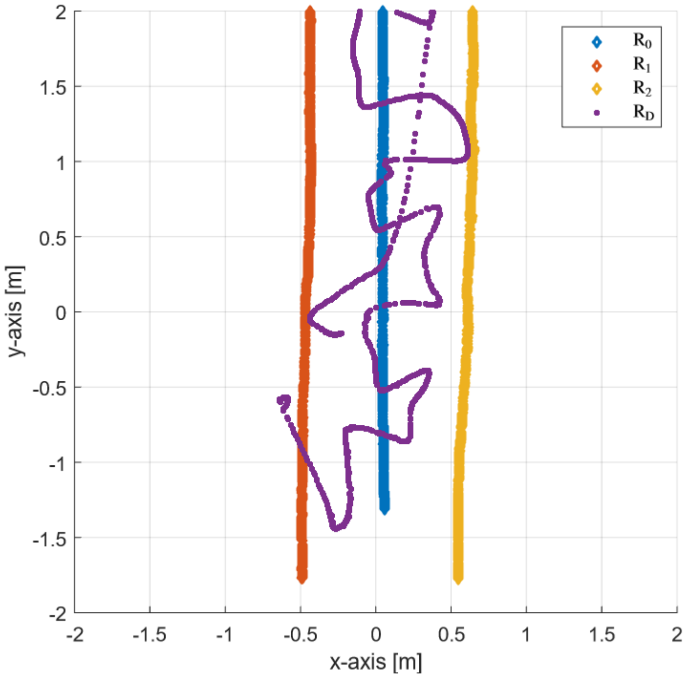


Figure 5.5 Ground truth of drone and ground robots

In an experiment we conducted in Figure 5.5, we can see the robots' movements in graphic form. We said that, unlike our simulation work, we moved the drone between pairs of robots in much shorter times. In Figure 5.5, we can see that the drone makes more frequent transitions than simulation tests.

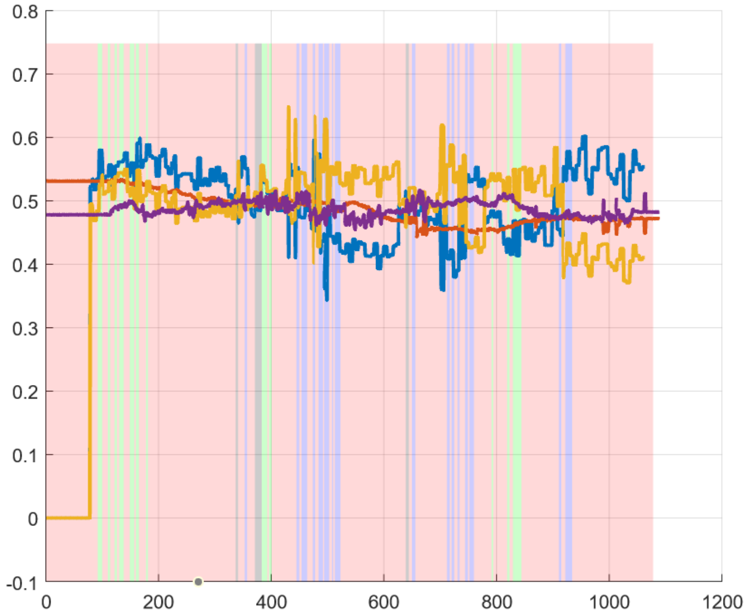


Figure 5.6 x and y positions estimation and ground truths of first ground robot

In Figure 5.6, we can see the actual x and y position values of the leading robot and the first following robot and the values calculated by our EKF algorithm. We observed that sometimes we get very close to the absolute values, and sometimes we move away from these values. The reason for this can be explained as follows. When the drone hovers over these two robots, our EKF algorithm can calculate much more decisively because it uses distance, bearing, and heading angles. In the other case, when the drone does not give a heading and bearing angles, we can calculate relative positions in EKF with only the distance value from the UWB sensors. Therefore, the calculated values are not better than the previous ones.

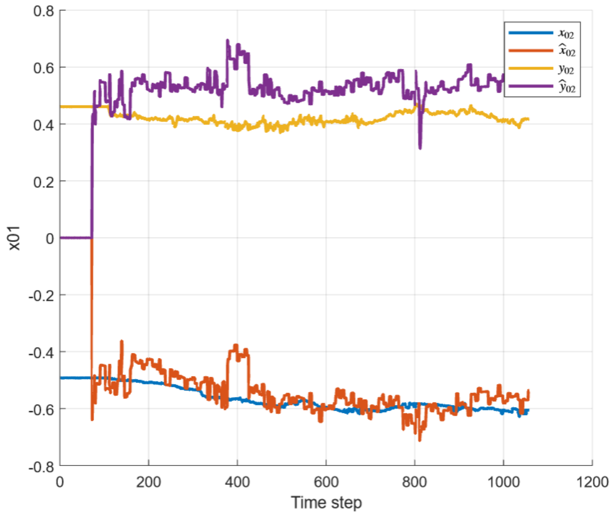


Figure 5.7 x and y positions estimation and ground truths of second ground robot.

By looking at Figure 5.7, we can see that our EKF algorithm results for the first robot set and the second robot pair are almost identical. When we look at the calculated relative position x and y values of the leader robot and the following robot, we can see deviations simultaneously. So, it becomes stable in the same situation. The reason for this is that the drone's heading and bearing angles are used together with the distance value in our EKF algorithm.

In Figure 5.8, we see the absolute positions of the robots in another test. Again, in this test, as in the other test, we quickly moved our drone between the robots. In this test, we can see that the drone switches between robots more than the previous one.

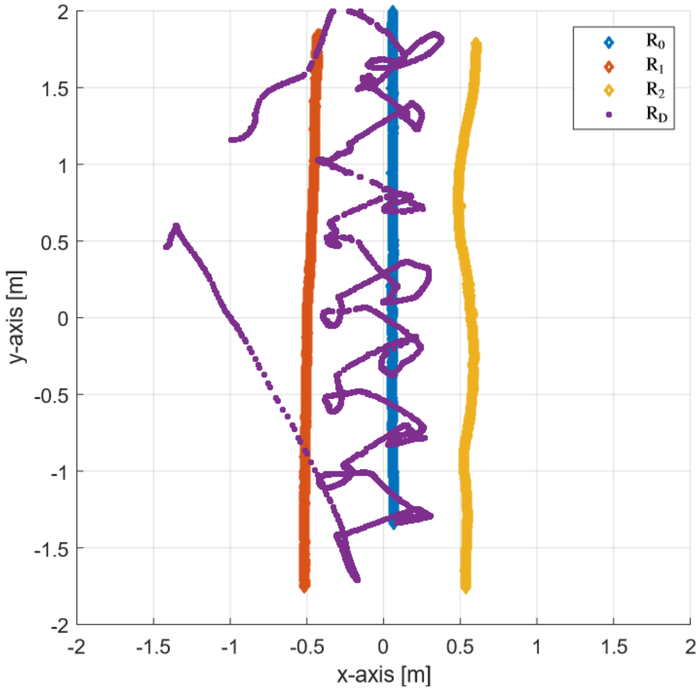


Figure 5.8 Ground robots and drone pozitions in a real experiment.

For this test, we can compare the actual and calculated relative position values between the leader and the first following robot in Figure 5.9. Again, when our drone cannot see the robots, it cannot get heading and bearing values. We cannot stably do the EKF calculation and gets away from the actual values. When the drone takes the heading and bearing values, we can see that our EKF calculation gives very close results to the absolute values.

In Figure 5.10, we can see the actual and calculated heading values of this test. In all our simulations and actual tests, the drone always sees the leading robot. The drone shifts and flies on other follower robots. Meanwhile, the drone never stops following the leader robot, and the changing robot is always the follower. As we can see in Figure 5.10, the heading angle value of the leader robot started to be calculated as soon as the test started. Then, it can be seen that the heading values of other robots are calculated. As a result of our test, we can see in Figure 5.10 that we have obtained very close results to the actual heading values. These results show the success of the method we have developed.

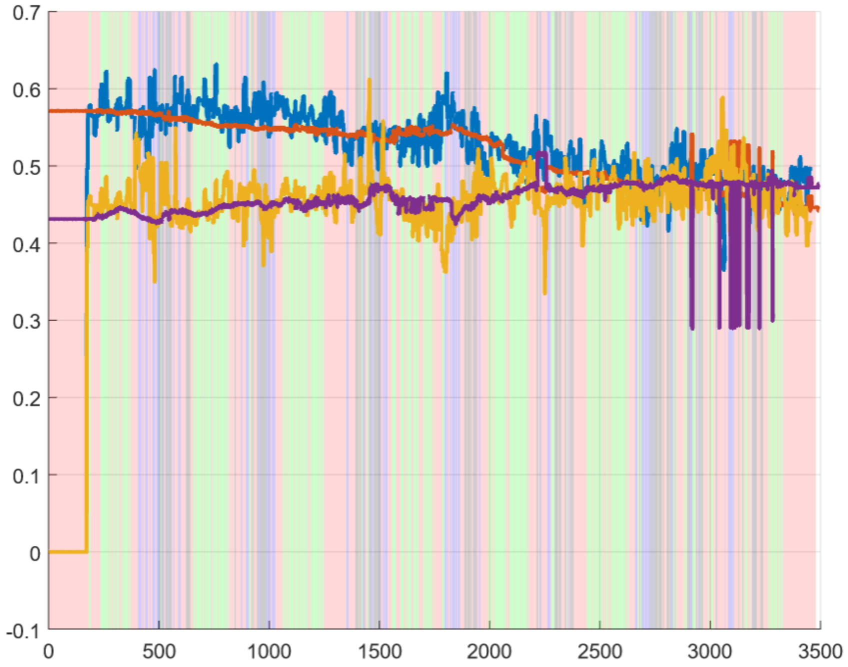


Figure 5.9 x and y positions estimation and ground truths of first ground robot.

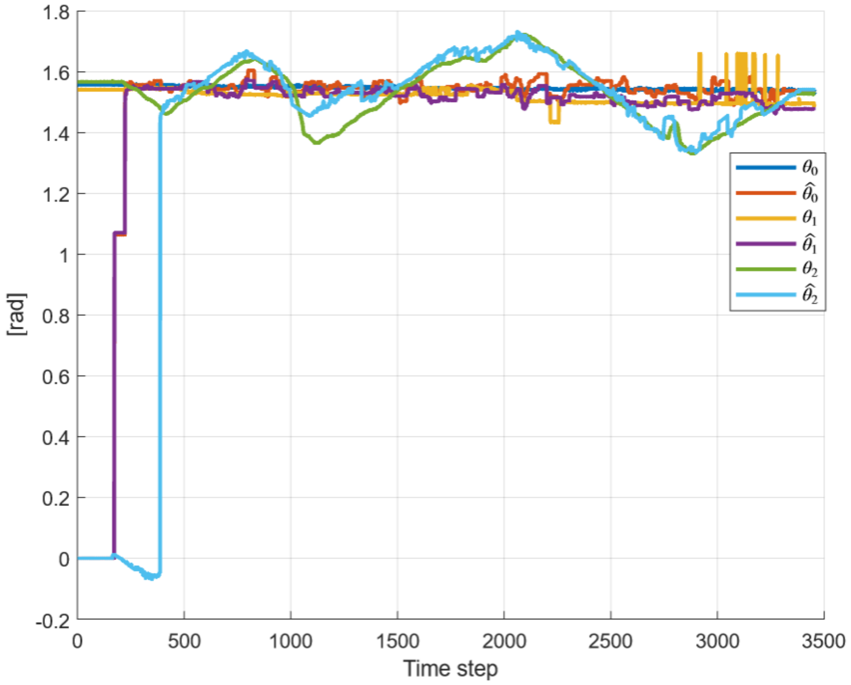


Figure 5.10 Heading angle estimation and absolute heading angle values of ground robots.

Chapter 6

Conclusions and Future Prospects

6.1 Conclusions

Heterogeneous multi-robot systems have attracted significant attention in the recent years due to their flexibility, robustness, and resilience in complex tasks in unknown environments. A reliable coordination between air and ground entities in an air-ground robot team requires to solve the perception and decision problems simultaneously and remains a key challenge in the robotics field. We have proposed a complete localization solution for a team of an aerial vehicle and multiple ground robots which use sensors with distinct characteristics. Particularly, the drone has a mono camera that provides the ground robots with a perspective view from above, and the ground robots use ultrawideband distance sensors which provide omnidirectional distance measurements regardless of the field-of-view constraint. The drone is employed as an absolute frame provider for the entire team, removing the requirement for an external infrastructure such as a GPS sensor or a motion capture system. We have derived an extended Kalman filter with three measurement models which correspond to the availability of the drone's perception. The drone is allowed to hover on top of different pairs of the ground robots to provide each ground robot with their heading and bearing angle measurements for a sufficient amount of time. For a reliable detection of such angles from the drone's camera, we suggest the use of special patterns such as QR code. Simulation results have showed that the ground robots can perform a reliable localization under the leader-follower formation scheme.

The main advantages of the proposed scheme lie in the heterogeneous structure among the team members. First, since the drone provides a perspective view of the ground robots, the ground robots do not need an absolute sensor such as absolute encoder, magnetometer, and GPS. In our framework, the ground robots used a low-cost computational board and driver, and an ultrawideband distance sensor. Furthermore, the team do not make use of an external infrastructure such as a motion capture system. Since the drone can always hover on the ground robots, it can use onboard sensors and computational units solely for a precise flight. Hence, the entire system coordinates as a team in which every member contributes to the localization and control of its neighbors.

6.2 Societal Impact and Contribution to Global Sustainability

Heterogeneous robot teams have been implemented broadly in many social and military applications such as firefighting, agriculture, search and rescue, mapping, target tracking, and docking. In the future, it is expected that drones will play an important role in agricultural applications. Teaming a drone with ground robots extends the drone’s impact and reduces the time and human effort in many applications. Particularly, such heterogeneous teams are expected to improve efficiency in agricultural applications. For instance, currently, detection of anomalies in crops is performed by humans manually and inefficiently. An autonomous air-ground team can be implemented in such a task, and the ground robots can use a camera to detect the anomalies in the crops while maintaining a formation. Meanwhile, the drone can hover on the ground robots and improve the precision of the anomaly detection by providing a bird-eye view of the crops. One cannot trust the GPS data in such a heterogeneous formation with narrow edge lengths because GPS sensors may not produce such precision in most cases. Therefore, our framework can prove useful in such applications by reducing the time spent for detecting anomalies or counting products in large farms.

Furthermore, border surveillance by military entities require to drive tons of hours across borders and cause a huge carbon dioxide (CO2) emission. The proposed framework can be implemented in border surveillance with a suitable path planning strategy. Also, since one can use different types of sensors on the ground and aerial vehicles, the quality of the surveillance can be improved by a heterogeneous robot team compared to the usual human surveillance. Such an implementation can greatly reduce the human effort and time and the need for the observation of remote areas by traditional vehicles. Since the CO2 emission of the drone and ground robots would be much less than that of usual vehicles, border surveillance by an air-ground team can significantly improve efficiency and contribute to the sustainability.

6.3 Future Prospects

The proposed framework can be improved in several ways to meet the demands of real-life applications. For a smooth transition from simulations to real experiments, one needs to consider the tuning of the drone motion. Also, the drone should be equipped with sensors which provide precise flight in GNSS denied environments such as optical flow camera and a single beam laser distance sensor.

Furthermore, combination of the proposed localization algorithm with a suitable path planning and formation control framework is a promising direction. For instance, the ground robots can apply a control procedure in a leader-follower formation to maintain the relative position toward the leader ground robot. Simultaneously, the drone can explore the field and find collision-free paths for the entire team.

BIBLIOGRAPHY

[1] J. Gancet, E. Motard, A. Naghsh, C. Roast, M. M. Arancon, and L. Marques, "User interfaces for human robot interactions with a swarm of robots in support to firefighters," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2010, pp. 2846–2851, doi: 10.1109/ROBOT.2010.5509890.

[2] E. T. Alotaibi, S. S. Alqefari, and A. Koubaa, "LSAR: Multi-UAV Collaboration for Search and Rescue Missions," *IEEE Access*, vol. 7, pp. 55817–55832, 2019, doi: 10.1109/ACCESS.2019.2912306.

[3] C. J. R. McCook and J. M. Esposito, "Flocking for heterogeneous robot swarms: A military convoy scenario," in *Proceedings of the Annual Southeastern Symposium on System Theory*, 2007, pp. 26–31, doi: 10.1109/SSST.2007.352311.

[4] A. Ravankar, A. Ravankar, T. Emaru, and Y. Kobayashi, "Multi-Robot Mapping and Navigation Using Topological Features," *Proceedings*, vol. 42, no. 1, p. 68, Nov. 2019, doi: 10.3390/ecsa-6-06580.

[5] L. Jin, S. Li, H. M. La, X. Zhang, and B. Hu, "Dynamic task allocation in multi-robot coordination for moving target tracking: A distributed approach," *Automatica*, vol. 100, pp. 75–81, Feb. 2019, doi: 10.1016/J.AUTOMATICA.2018.11.001.

[6] P. Tokekar, J. Vander Hook, D. Mulla, and V. Isler, "Sensor Planning for a Symbiotic UAV and UGV System for Precision Agriculture," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1498–1511, Dec. 2016, doi: 10.1109/TRO.2016.2603528.

[7] J. J. Roldán, P. Garcia-Aunon, M. Garzón, J. De León, J. Del Cerro, and A. Barrientos, "Heterogeneous Multi-Robot System for Mapping Environmental Variables of Greenhouses," *Sensors 2016, Vol. 16, Page 1018*, vol. 16, no. 7, p. 1018, Jul. 2016, doi: 10.3390/S16071018.

[8] C. Ju and H. Il Son, "Modeling and Control of Heterogeneous Agricultural Field Robots Based on Ramadge-Wonham Theory," *IEEE Robot. Autom. Lett.*, vol. 5, no. 1, pp. 48–55, Jan. 2020, doi: 10.1109/LRA.2019.2941178.

[9] P. Gonzalez-de-Santos *et al.*, "Fleets of robots for environmentally-safe pest control in agriculture," *Precis. Agric. 2016 184*, vol. 18, no. 4, pp. 574–614, Oct. 2016, doi: 10.1007/S11119-016-9476-3.

[10] C. Phan and H. H. T. Liu, "A cooperative UAV/UGV platform for wildfire

detection and fighting,” in *2008 Asia Simulation Conference - 7th International Conference on System Simulation and Scientific Computing, ICSC 2008*, 2008, pp. 494–498, doi: 10.1109/ASC-ICSC.2008.4675411.

[11] J. Quenzel *et al.*, “Autonomous Fire Fighting with a UAV-UGV Team at MBZIRC 2020.”

[12] Q. Wang, H. Chen, L. Qiao, J. Tian, and Y. Su, “Path planning for UAV/UGV collaborative systems in intelligent manufacturing,” *IET Intell. Transp. Syst.*, vol. 14, no. 11, pp. 1475–1483, Nov. 2020, doi: 10.1049/IET-ITS.2019.0688.

[13] E. H. C. Harik, F. Guerin, F. Guinand, J. F. Brethe, and H. Pelvillain, “UAV-UGV cooperation for objects transportation in an industrial area,” *Proc. IEEE Int. Conf. Ind. Technol.*, vol. 2015-June, no. June, pp. 547–552, Jun. 2015, doi: 10.1109/ICIT.2015.7125156.

[14] I. R. Nourbakhsh, K. Sycara, M. Koes, M. Yong, M. Lewis, and S. Burion, “Human-robot teaming for Search and Rescue,” *IEEE Pervasive Computing*, vol. 4, no. 1. Institute of Electrical and Electronics Engineers Inc., pp. 72–77, 2005, doi: 10.1109/MPRV.2005.13.

[15] Y. N. Bezkorovainyi and O. A. Sushchenko, “Improvement of UAV Positioning by Information of Inertial Sensors,” in *2018 IEEE 5th International Conference on Methods and Systems of Navigation and Motion Control, MSNMC 2018 - Proceedings*, Dec. 2018, pp. 151–155, doi: 10.1109/MSNMC.2018.8576307.

[16] P. Clausen and J. Skaloud, “On the calibration aspects of MEMS-IMUs used in micro UAVs for sensor orientation,” in *2020 IEEE/ION Position, Location and Navigation Symposium, PLANS 2020*, Apr. 2020, pp. 1457–1466, doi: 10.1109/PLANS46316.2020.9110160.

[17] I. Y. Bar-Itzhack and R. R. Harman, “Optimized TRIAD algorithm for attitude determination,” *J. Guid. Control. Dyn.*, vol. 20, no. 1, pp. 208–211, May 1997, doi: 10.2514/2.4025.

[18] S. K. Hong and S. Park, “Minimal-drift heading measurement using a MEMS gyro for indoor mobile robots,” *Sensors*, vol. 8, no. 11, pp. 7287–7299, Nov. 2008, doi: 10.3390/s8117287.

[19] R. Kurazume, S. Nagata, and S. Hirose, “Cooperative positioning with multiple robots,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 1994, no. pt 2, pp. 1250–1256, doi: 10.1109/robot.1994.351315.

[20] P. Bison, G. Chemello, C. Sossai, and G. Trainito, “Cooperative Localization

Using Possibilistic Sensor Fusion," *IFAC Proc. Vol.*, vol. 31, no. 3, pp. 177–182, Mar. 1998, doi: 10.1016/s1474-6670(17)44081-x.

[21] C. Jennings, D. Murray, and J. J. Little, "Cooperative robot localization with vision-based mapping," *Proc. - IEEE Int. Conf. Robot. Autom.*, vol. 4, 1999, doi: 10.1109/robot.1999.773999.

[22] R. Madhavan, K. Fregene, and L. E. Parker, "Distributed cooperative outdoor multirobot localization and mapping," *Auton. Robots*, vol. 17, no. 1, pp. 23–39, Jul. 2004, doi: 10.1023/B:AURO.0000032936.24187.41.

[23] S. Vemprala and S. Saripalli, "Monocular Vision based Collaborative Localization for Micro Aerial Vehicle Swarms," in *2018 International Conference on Unmanned Aircraft Systems, ICUAS 2018*, Aug. 2018, pp. 315–323, doi: 10.1109/ICUAS.2018.8453412.

[24] S. I. Roumeliotis and G. A. Bekey, "Distributed multirobot localization," *IEEE Trans. Robot. Autom.*, vol. 18, no. 5, pp. 781–795, Oct. 2002, doi: 10.1109/TRA.2002.803461.

[25] Y. Zhuang, G. U. Mingwei, W. Wang, and Y. U. Haiyang, "Multi-robot cooperative localization based on autonomous motion state estimation and laser data interaction," *Sci China Inf Sci*, vol. 53, no. 11, pp. 2240–2250, 2010, doi: 10.1007/s11432-010-4096-4.

[26] A. Wąsik, R. Ventura, J. N. Pereira, P. U. Lima, and A. Martinoli, "Lidar-based relative position estimation and tracking for multi-robot systems," in *Advances in Intelligent Systems and Computing*, 2016, vol. 417, pp. 3–16, doi: 10.1007/978-3-319-27146-0_1.

[27] R. Dube, A. Gawel, H. Sommer, J. Nieto, R. Siegwart, and C. Cadena, "An online multi-robot SLAM system for 3D LiDARs," in *IEEE International Conference on Intelligent Robots and Systems*, Dec. 2017, vol. 2017-Septe, pp. 1004–1011, doi: 10.1109/IROS.2017.8202268.

[28] X. Zhao, G. Wang, and S. S. Ge, "Cooperative Localization Based on Robust GPS and Radar Fusion for Multiple Aerial Vehicles," *Int. J. Control. Autom. Syst.*, vol. 15, no. 2, pp. 732–742, 2017, doi: 10.1007/s12555-014-0246-6.

[29] F. D. P. Muller, E. M. Diaz, and I. Rashdan, "Cooperative positioning and radar sensor fusion for relative localization of vehicles," in *IEEE Intelligent Vehicles Symposium, Proceedings*, Aug. 2016, vol. 2016-August, pp. 1060–1065, doi: 10.1109/IVS.2016.7535520.

[30] J. Kim, “Optimal motion controllers for an unmanned surface vehicle to track a maneuvering underwater target based on coarse range-bearing measurements,” *Ocean Eng.*, vol. 216, p. 107973, Nov. 2020, doi: 10.1016/j.oceaneng.2020.107973.

[31] Y. Fan, Y. Zhang, G. Wang, X. Wang, and N. Li, “Maximum correntropy based unscented particle filter for cooperative navigation with heavy-tailed measurement noises,” *Sensors (Switzerland)*, vol. 18, no. 10, p. 3183, Oct. 2018, doi: 10.3390/s18103183.

[32] H. Takai, M. Miyake, and K. Tachibana, “Multi-robot cooperative localization and mapping using multi-sensor information fusion,” 2008.

[33] H. Duan and S. Liu, “Unmanned air/ground vehicles heterogeneous cooperative techniques: Current status and prospects,” *Sci. China Technol. Sci. 2010 535*, vol. 53, no. 5, pp. 1349–1355, Apr. 2010, doi: 10.1007/S11431-010-0122-4.

[34] H. G. Tanner, “Switched UAV-UGV cooperation scheme for target detection,” *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 3457–3462, 2007, doi: 10.1109/ROBOT.2007.364007.

[35] C. Papachristos and A. Tzes, “The power-tethered UAV-UGV team: A collaborative strategy for navigation in partially-mapped environments,” *2014 22nd Mediterr. Conf. Control Autom. MED 2014*, pp. 1153–1158, Nov. 2014, doi: 10.1109/MED.2014.6961531.

[36] R. Grabowski, L. E. Navarro-Serment, C. J. J. Paredis, and P. K. Khosla, “Heterogeneous Teams of Modular Robots for Mapping and Exploration,” 2000.

[37] L. Jurisica, A. Vitko, F. Duchon, and D. Kastan, “Statistical approach to GPS positioning of mobile robot,” *Control Eng. Appl. Informatics*, vol. 12, no. 2, pp. 44–51, 2010, [Online]. Available: file:///C:/Users/SAEMRE~1/AppData/Local/Temp/927-895-1-PB.pdf.

[38] R. Gonzalez, F. Rodriguez, J. L. Guzman, C. Pradalier, and R. Siegwart, “Combined visual odometry and visual compass for off-road mobile robots localization,” *Robotica*, vol. 30, no. 6, pp. 865–878, Oct. 2012, doi: 10.1017/S026357471100110X.

[39] F. Caron, E. Duflos, D. Pomorski, and P. Vanheeghe, “GPS/IMU data fusion using multisensor Kalman filtering: Introduction of contextual aspects,” *Inf. Fusion*, vol. 7, no. 2, pp. 221–230, Jun. 2006, doi: 10.1016/j.inffus.2004.07.002.

[40] “Infrastructureless indoor mapping using a mobile antenna array | IEEE

Conference Publication | IEEE Xplore." https://ieeexplore.ieee.org/abstract/document/6208151 (accessed Jul. 29, 2021).

[41] L. Giarre, F. Pascucci, S. Morosi, and A. Martinelli, "Improved PDR Localization via UWB-Anchor Based on-Line Calibration," *IEEE 4th Int. Forum Res. Technol. Soc. Ind. RTSI 2018 - Proc.*, Nov. 2018, doi: 10.1109/RTSI.2018.8548377.

[42] S. Zhang, R. Han, W. Huang, S. Wang, and Q. Hao, "Linear Bayesian Filter Based Low-Cost UWB Systems for Indoor Mobile Robot Localization," in *Proceedings of IEEE Sensors*, Dec. 2018, vol. 2018-Octob, doi: 10.1109/ICSENS.2018.8589829.

[43] A. Prorok, L. Gonon, and A. Martinoli, "Online model estimation of ultra-wideband TDOA measurements for mobile robot localization," *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 807–814, 2012, doi: 10.1109/ICRA.2012.6224869.

[44] D. Feng, C. Wang, C. He, Y. Zhuang, and X. G. Xia, "Kalman-Filter-Based Integration of IMU and UWB for High-Accuracy Indoor Positioning and Navigation," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3133–3146, Apr. 2020, doi: 10.1109/JIOT.2020.2965115.

[45] J. González *et al.*, "Combination of UWB and GPS for indoor-outdoor vehicle localization," 2007, doi: 10.1109/WISP.2007.4447550.

[46] J. Hyun, T. Oh, H. Lim, and H. Myung, "UWB-based Indoor Localization Using Ray-tracing Algorithm," in *2019 16th International Conference on Ubiquitous Robots, UR 2019*, Jun. 2019, pp. 98–101, doi: 10.1109/URAI.2019.8768568.

[47] D. B. Jourdan, J. J. Deyst, M. Z. Win, and N. Roy, "Monte Carlo localization in dense multipath environments using UWB ranging," *ICU 2005 2005 IEEE Int. Conf. Ultra-Wideband, Conf. Proc.*, vol. 2005, pp. 314–319, 2005, doi: 10.1109/ICU.2005.1570005.

[48] J. Djugash, S. Singh, G. Kantor, and W. Zhang, "Range-only SLAM for robots operating cooperatively with sensor networks," *Proc. - IEEE Int. Conf. Robot. Autom.*, vol. 2006, pp. 2078–2084, 2006, doi: 10.1109/ROBOT.2006.1642011.

[49] C. Gentner, M. Ulmschneider, and T. Jost, "Cooperative Simultaneous Localization and Mapping for Pedestrians using Low-Cost Ultra-Wideband System and Gyroscope." pp. 1197–1205, Apr. 26, 2018, Accessed: Aug. 09, 2021. [Online]. Available: http://www.ion.org/publications/abstract.cfm?jp=p&articleID=15769.

[50] L. Geneve, O. Kermorgant, and E. Laroche, "A composite beacon initialization for

EKF range-only SLAM," *IEEE Int. Conf. Intell. Robot. Syst.*, vol. 2015-Decem, pp. 1342–1348, Dec. 2015, doi: 10.1109/IROS.2015.7353542.

[51] G. Vallicrosa, P. Ridao, D. Ribas, and A. Palomer, "Active Range-Only beacon localization for AUV homing," *IEEE Int. Conf. Intell. Robot. Syst.*, pp. 2286–2291, Oct. 2014, doi: 10.1109/IROS.2014.6942871.

[52] T. Deißler and J. Thielecke, "UWB SLAM with rao-blackwellized Monte Carlo data association," *2010 Int. Conf. Indoor Position. Indoor Navig. IPIN 2010 - Conf. Proc.*, 2010, doi: 10.1109/IPIN.2010.5647596.

[53] J. L. Blanco, J. González, and J. A. Fernández-Madrigal, "A pure probabilistic approach to range-only SLAM," *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 1436–1441, 2008, doi: 10.1109/ROBOT.2008.4543404.

[54] E. Menegatti, A. Zanella, S. Zilli, F. Zorzi, and E. Pagello, "Range-only slam with a mobile robot and a wireless sensor networks," *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 8–14, 2009, doi: 10.1109/ROBOT.2009.5152449.

[55] J. Djugash and S. Singh, "A Robust Method of Localization and Mapping Using Only Range," *Springer Tracts Adv. Robot.*, vol. 54, pp. 341–351, 2009, doi: 10.1007/978-3-642-00196-3_40.

[56] G. Ortiz, F. Treven, L. Svensson, P. Larsson-Edefors, and S. Johansson-Mauricio, "A framework for a relative real-time tracking system based on ultra-wideband technology," *2017 14th Work. Positioning, Navig. Commun. WPNC 2017*, vol. 2018-Janua, pp. 1–6, 2018, doi: 10.1109/WPNC.2017.8250067.

[57] B. Hepp, T. Nägeli, and O. Hilliges, "Omni-directional person tracking on a flying robot using occlusion-robust ultra-wideband signals," *IEEE Int. Conf. Intell. Robot. Syst.*, vol. 2016-Novem, pp. 189–194, 2016, doi: 10.1109/IROS.2016.7759054.

[58] S. Guler, J. Jiang, A. A. Alghamdi, R. I. Masoud, and J. S. Shamma, "Real Time Onboard Ultrawideband Localization Scheme for an Autonomous Two-robot System," in *2018 IEEE Conference on Control Technology and Applications, CCTA 2018*, Oct. 2018, pp. 1151–1158, doi: 10.1109/CCTA.2018.8511568.

[59] J. P. Queralta, L. Qingqing, F. Schiano, and T. Westerlund, "VIO-UWB-Based Collaborative Localization and Dense Scene Reconstruction within Heterogeneous Multi-Robot Systems," 2020, [Online]. Available: http://arxiv.org/abs/2011.00830.

[60] M. Cognetti, G. Oriolo, P. Peliti, L. Rosa, and P. Stegagno, "Cooperative control

of a heterogeneous multi-robot system based on relative localization," in *IEEE International Conference on Intelligent Robots and Systems*, Oct. 2014, pp. 350–356, doi: 10.1109/IROS.2014.6942583.

[61] P. Stegagno, M. Cognetti, L. Rosa, P. Peliti, and G. Oriolo, "Relative localization and identification in a heterogeneous multi-robot system," *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 1857–1864, 2013, doi: 10.1109/ICRA.2013.6630822.

[62] T. M. Nguyen, A. H. Zaini, C. Wang, K. Guo, and L. Xie, "Robust Target-Relative Localization with Ultra-Wideband Ranging and Communication," in *Proceedings - IEEE International Conference on Robotics and Automation*, Sep. 2018, pp. 2312–2319, doi: 10.1109/ICRA.2018.8460844.

[63] T. M. Nguyen, T. H. Nguyen, M. Cao, Z. Qiu, and L. Xie, "Integrated UWB-vision approach for autonomous docking of UAVS in GPS-denied environments," *Proc. - IEEE Int. Conf. Robot. Autom.*, vol. 2019-May, pp. 9603–9609, 2019, doi: 10.1109/ICRA.2019.8793851.

[64] M. De Petrillo, J. Beard, Y. Gu, and J. N. Gross, "Search Planning of a UAV/UGV Team with Localization Uncertainty in a Subterranean Environment," Feb. 2021, Accessed: Mar. 11, 2021. [Online]. Available: http://arxiv.org/abs/2102.06069.

[65] R. Dubois, A. Eudes, and V. Fremont, "AirMuseum: a heterogeneous multi-robot dataset for stereo-visual and inertial Simultaneous Localization and Mapping," *IEEE Int. Conf. Multisens. Fusion Integr. Intell. Syst.*, vol. 2020-Septe, pp. 166–172, 2020, doi: 10.1109/MFI49285.2020.9235257.

[66] F. Cocchioni *et al.*, "Unmanned ground and aerial vehicles in extended range indoor and outdoor missions," *2014 Int. Conf. Unmanned Aircr. Syst. ICUAS 2014 - Conf. Proc.*, pp. 374–382, 2014, doi: 10.1109/ICUAS.2014.6842276.

[67] S. Hood, K. Benson, P. Hamod, D. Madison, J. M. O'Kane, and I. Rekleitis, "Bird's eye view: Cooperative exploration by UGV and UAV," *2017 Int. Conf. Unmanned Aircr. Syst. ICUAS 2017*, pp. 247–255, 2017, doi: 10.1109/ICUAS.2017.7991513.

[68] E. Mueggler, M. Faessler, F. Fontana, and D. Scaramuzza, "Aerial-guided navigation of a ground robot among movable obstacles," *12th IEEE Int. Symp. Safety, Secur. Rescue Robot. SSRR 2014 - Symp. Proc.*, Jan. 2014, doi: 10.1109/SSRR.2014.7017662.

[69] G. Niu, Q. Yang, Y. Gao, and M.-O. Pun, "Vision-based Autonomous Landing for Unmanned Aerial and Mobile Ground Vehicles Cooperative Systems," *IEEE*

Robot. Autom. Lett., pp. 1–1, 2021, doi: 10.1109/LRA.2021.3101882.

[70] S. M. Nogar, "Autonomous Landing of a UAV on a Moving Ground Vehicle in a GPS Denied Environment," *2020 IEEE Int. Symp. Safety, Secur. Rescue Robot. SSRR 2020*, pp. 77–83, 2020, doi: 10.1109/SSRR50563.2020.9292607.

[71] A. Dewan, A. Mahendran, N. Soni, and K. M. Krishna, "Heterogeneous UGV-MAV exploration using integer programming," *IEEE Int. Conf. Intell. Robot. Syst.*, pp. 5742–5749, 2013, doi: 10.1109/IROS.2013.6697188.

[72] C. Li *et al.*, "ReLoc 2.0: UHF-RFID Relative Localization for Drone-Based Inventory Management," *IEEE Trans. Instrum. Meas.*, vol. 70, 2021, doi: 10.1109/TIM.2021.3069377.

[73] W. Li, T. Zhang, and K. Kühnlenz, "A vision-guided autonomous quadrotor in an air-ground multi-robot system," *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 2980–2985, 2011, doi: 10.1109/ICRA.2011.5979579.

CURRICULUM VITAE

2004 – 2009	Computer Engineering, İstanbul University, İstanbul, TURKEY
2019 – Present	M.Sc., Electrical and Computer Engineering, Abdullah GÜL University, Kayseri, TURKEY

SELECTED PUBLICATIONS AND PRESENTATIONS

There is no publication and presentation