Ömer Faruk Serin

A Master's Thesis

AGU 2022

AN AUTONOMOUS HETEROGENEOUS MULTI-ROBOT SYSTEM DESIGN FOR EARLY FIRE DETECTION

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
Ömer Faruk SERİN
December 2022

AN AUTONOMOUS HETEROGENEOUS MULTI-ROBOT SYSTEM DESIGN FOR EARLY FIRE DETECTION

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
Ömer Faruk SERİN
December 2022

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: Ömer Faruk SERİN
Signature :

REGULATORY COMPLIANCE

M.Sc. thesis titled "*An Autonomous Heterogeneous Multi-Robot System Design For Early Fire Detection*" has been prepared in accordance with the Thesis Writing Guidelines of the Abdullah Gül University, Graduate School of Engineering & Science.

Prepared By	Advisor
Ömer Faruk SERİN	Asst. Prof. Samet GÜLER

Head of the Electrical and Computer Engineering Program

Assoc. Prof. Zafer AYDIN

Signature

ACCEPTANCE AND APPROVAL

M.Sc. thesis titled “An Autonomous Heterogeneous Multi-Robot System Design For Early Fire Detection” and prepared by Ömer Faruk Serin has been accepted by the jury in the Electrical and Computer Engineering Graduate Program at Abdullah Gül University, Graduate School of Engineering & Science.

13/12/2022

(Thesis Defense Exam Date)

JURY:

Advisor : Asst. Prof. Samet GÜLER

Member : Asst. Prof. Mehmet BOZDAL

Member : Asst. Prof. Serkan SARITAŞ

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. İrfan ALAN

ABSTRACT

AN AUTONOMOUS HETEROGENEOUS MULTI-ROBOT SYSTEM DESIGN FOR EARLY FIRE DETECTION

Ömer Faruk SERİN
MSc. in Electrical and Computer Engineering
Advisor: Asst. Prof. Samet GÜLER

December 2022

The usage of autonomous multi-robot systems for human life-endangering applications is emerging. Early wildfire detection and firefighting are two example applications. In this study, a heterogenous multi-robot system is proposed for both fire detection and response. The system employs an unmanned aerial vehicle for beyond-visual line-of-sight observations and an unmanned ground robot for fire extinguisher carrying. The proposed method uses ultrawideband (UWB) communication and ranging modules for the relative localization of robots during their movements. A specially trained YOLOv7 object detection model is used for robustly detecting forest fires and smoke while a modified version of the Vector Field Histogram Plus (VFH+) algorithm on the ground robot is used for obstacle avoidance while navigating. The structural design of the system requires no odometry or mapping of the environment hence improving the applicability of the system while decreasing system complexity. Additionally, the proposed UWB localization system is shown to be robust in long-lasting operations unlike many odometry-based approaches which accumulate errors with time. Moreover, localization of the UAV is realized with only three independent UWB-based range measurements and the altitude information of the UAV. The system is tested both in a realistic simulation environment and in real experimental setups with multiple runs. Results showed that the proposed system is improvable for better detection and practical to implement even in a dense forest environment without the need for GPS sensors, odometer data, or magnetometer.

Keywords: Multi-robot systems, Ultrawide-band localization, forest fire detection, YOLOv7

ÖZET

ERKEN ORMAN YANGINI TESPİTİ İÇİN OTONOM HETEROJEN ÇOKLU ROBOT SİSTEMİ TASARIMI

Ömer Faruk SERİN
Elektrik ve Bilgisayar Mühendisliği Anabilim Dalı Yüksek Lisans
Tez Yöneticisi: Dr. Öğr. Üyesi Samet GÜLER
Aralık 2022

İnsan sağlığını ya da güvenliğini tehlikeye atan durumlarda otonom mobil robotların kullanımı giderek artmaktadır. Erken orman yangını tespiti ve mücadelesi buna örnek verilebilecek eylemlerdendir. Bu çalışmada, bu iki uygulamada kullanılmak üzere heterojen, çoklu robot sistemi geliştirilmiştir. Sistem görüş alanı dışındaki nesnelerin tespiti için bir insansız hava aracını ve yangın söndürücü malzemelerin olay yerine taşınması için bir yer aracını kullanır. Önerilen yöntem robotların hareketleri esnasında görece lokalizasyonları için ultra geniş bant (UGB) iletişim ve mesafe ölçme modülü kullanır. Bu amaçla, insansız hava aracı yeni nesil bilgisayarlı görme ve nesne tespiti algoritması olan YOLOv7'yi kullanırken insansız kara aracı engellere çarpmamak ve yolunu bulmak için modifiye edilmiş Vektör Alan Histogramı (VFH+) algoritmasını kullanır. Sistemin yapısal tasarımı, odometri sistemlerine ve haritalama işlemlerine ihtiyaç duymayarak sistem karmaşıklığını azaltarak uygulanabilirliğini artırmaktadır. Bunun yanında, zamanla hatayı biriktiren odometri-tabanlı yaklaşımların aksine, önerilen UGB tabanlı yöntemin uzun süreli operasyonlarda gürbüz olduğu gösterilmiştir. Ayrıca, yalnızca üç UGB tabanlı mesafe ölçümü ve hava aracı irtifa bilgilerini kullanarak lokalizasyon işlemleri hesaplanmıştır. Tasarlanan sistem hem gerçekçi simülasyonlar üzerinde hem de gerçek deney sistemi üzerinde test edilmiştir. Sonuçlar, sistemin GPS, odometri veya manyetometre kullanmadan, nesne tespit algoritmalarını ve engel tespit sensörlerini günlük uygulamalar için daha da geliştirilerek gerçek hayatta uygulanabileceğini göstermiştir.

Anahtar kelimeler: Çoklu robotik sistemler, Ultra geniş bant konumlama, orman yangını tespiti, YOLOv7

Acknowledgements

During my MSc journey, in my studies and in difficult times, I would like to thank and express my gratitude to my beloved and to my family. It would be much harder without your endless support. Additionally, I would like to thank my advisor and lab mates for their help and support during research and thesis studies.

This thesis has been produced benefiting from the 2232 International Fellowship for Outstanding Researchers Program of TÜBİTAK (Project No: 118C348). However, the entire responsibility of the thesis belongs to the owner of the thesis. The financial support received from TÜBİTAK does not mean that the content of the publication is approved in a scientific sense by TÜBİTAK.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ANN	Artificial Neural Network
BVLOS	Beyond Visual Line of Sight
CNN	Convolutional Neural Network
CRLB	Cramer-Rao Lower Bound
FOV	Field of View
GPS	Global Positioning System
LIDAR	Light Detection and Ranging
PEB	Position Error Bound
ROS	Robot Operating System
SITL	Software In The Loop
SLAM	Simultaneous Localization And Mapping
TOF	Time of Flight
TWR	Two-Way Ranging
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
UWB	Ultra-Wide Band
VFH	Vector Field Histogram
VIO	Visual Inertial Odometry
VIRO	Visual Inertial Range Odometry
VTOL	Vertical Take Off Landing

To my lifelong beloved Didem

Chapter 1

Introduction

Autonomous mobile robots have been in our life for a long time [1]. Their operational day-night independence and their successes in realizing repetitive tasks make them the best fit for many industrial [2], agricultural [3] or other application areas [4]. Especially, safety-related applications of mobile robots (e.g., tasks that can be harmful to a human to do) are emerging application areas of mobile robots. Autonomously or not, mobile robots are being used in many hazardous areas, for instance, mining areas [5], and search and rescue operations [6], [7]. A similar human safety endangering instance is forest fires. As the average temperature of the earth is increasing due to global warming, climates are being experienced at their limits [8]. In recent years, both the north and south hemispheres witnessed abnormal amounts of forest fire [9]–[11]. During fire extinguishing, many people passed away, huge lands of forest area got damaged and a total of 1.76 billion tons of CO_2 was released into the atmosphere during these fires [12].

When a forest fire starts it needs to be noticed by a forest guard or person and the first person who realizes the fire should inform a competent body by emergency calls. This process can take a lot of time if fire detection is not automatized by sensors. Even in a scenario with an immediate call for firefighters, it can take time for firefighters to arrive in the disaster area. Additionally, the large shape of fire trucks can prevent fighters from arriving, especially in dense forest environments. In the worst-case scenario, late emergency calls, and late realization of obstructed areas to pass can result in enlarged and hard-to-stop forest fires. In this situation, aerial fire extinguishing vehicles should take off if necessary.

As is the case for any natural disaster, a fast reaction to events is vital. Observing and being on the watch for detecting the start of forest fires requires huge amounts of human labor. Even with this forest guard-watching approach, early fire detection with the naked eye might not be possible due to differences in weather conditions and changes in

visibility range. Hence, observation buildings need to be equipped with advanced sensors for early detection. To solve this early detection problem, there are several Internet of Things (IoT) based approaches in the literature [13]–[16]. However, even in the case of early fire detection, the first fire-fighting vehicles might not arrive in place or they can arrive late due to the reasons mentioned before. A proper approach should consider both detection and response to a forest fire.

This simple in nature yet important mission can be done with the help of autonomous vehicles effectively, 7 days, 24 hours. These robots can be placed in different points of a forest and by actively scanning their dedicated area they can both detect and early fight the fire until a fire-fighting team arrives in the area if necessary.

The autonomous vehicle to detect and respond to fire can be an aerial or ground vehicle. By keeping in mind being able to carry a high amount of water or chemicals within the body of the vehicle is advantageous since the probability of extinguishing increases as the payload increases, a ground vehicle is proper for this task. For an aerial vehicle, cathexis will fight both movements towards the target and opposition to gravity. However, a ground vehicle has the advantage of using gravity on behalf of itself.

On the other hand, ground vehicles lack a long-range field of view despite being able to carry high payloads. A dense forest in rugged terrain will limit the sight of the vehicle by meters. Hence a sensor should be used for beyond-visual line-of-sight detection of a forest fire. Such sensors can be thermal cameras, and smoke sensors, however, both sensor types perform poorly if the wind direction is opposite to the vehicle location or a large obstacle exists between the vehicle and the fire location.

An alternative approach to equipping the ground vehicle with advanced sensors would be an easier task if the ground robot would be able to see from the sky with an approach like bird-eye-view. Unfortunately, there is no simple sensor for this purpose. In this case, an aerial vehicle equipped with simple vision sensors can help the ground vehicle for detection and locate the fire.

Hence a team of an unmanned aerial vehicle (UAV) and an unmanned ground vehicle (UGV) is designed in this study. The total system can be briefly summarized in three steps.

1. Detecting fire: Early detection is important before fires get larger. The detection system should be precise and suitable.
2. Locating fire: After the detection of the fire, the relative localization task should be accomplished precisely. Otherwise, it can create hard-to-recover problems.
3. Moving and arriving at the fire location autonomously.

Both aerial and ground vehicles move toward the fire location. In this study, it is assumed that neither aerial nor ground vehicles know the environment and they do not utilize GPS or similar global localization systems.

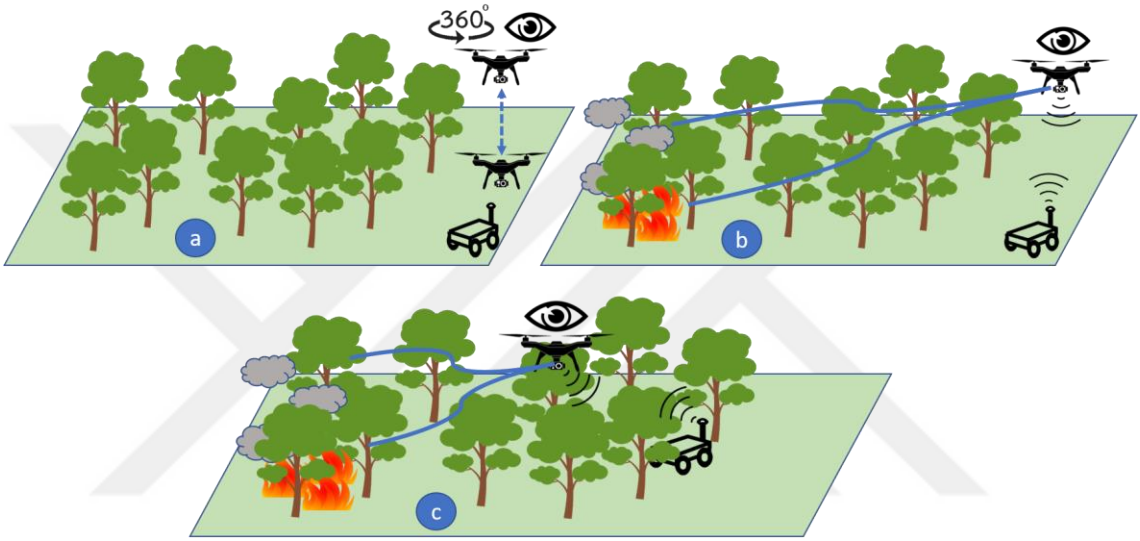

Figure 1.1 Visualization of the overall system working. (a) The UAV periodically takes off and scans the environment. (b) In case of detection, locates fire and informs ground. (c) UAV moves towards the fire location while the UGV locates UAV and follows it.

Such steps (visualized in Figure 1.1) for a team of mobile vehicles require advanced obstacle avoidance, localization and advanced odometry tracking as they move inside the forest. Since it might be difficult to provide strong wireless communication between a station and robot team under different field and weather conditions, a centralized solution to this problem would not be practical. Instead, both vehicles can move autonomously and accomplish their workload independently; hence, the total mission can be finished in a distributed and decentralized manner. All the aforementioned subsystems (odometry, localization, detection, etc.) are further examined in the next section.

1.1 Literature Review

The review of existing algorithms and methods is examined and compared in this section of the study. The literature review is divided into 3 parts as following:

i. Forest fire detection
ii. Multi-robot operations
iii. Path planning and obstacle avoidance algorithms.

1.1.1 Forest Fire detection

There are different approaches in the literature for forest fire detection. These methods can be classified into two categories:

a. Wireless/stationary sensor networks for forest-fire detection.
b. Mobile/moving sensor networks for forest-fire detection.

Different sensor types and network topologies exist in the literature to detect forest fires by employing wireless sensor networks. In [17], [18] authors used gas, particle and temperature sensors for detecting smoke coming out of a fire. In this system, sensors are placed at multiple points in a forest to create a sensor cloud. Sensor placements should be dense enough to observe and locate the coming smoke due to external effects like wind. Other similar methods are present in [19], [20] where instead of gas and temperature sensors, small cameras are placed in a forest, periodically imaging their field of view to detect forest fire and inform the ground station. However, all wireless communication-based systems have the drawback of being connected to a ground station. Even if sensors in the field detect the fire, they might not be able to inform the ground station due to corrupted, noisy messages or wireless communication service outages.

Another vision-based approach exists in [21] to make communication more robust for outages or corruption. In this method, a wireless sensor network forwards messages coming from other sensors which detected the forest fire to the control station. This multi-hopping structure ensures that the maximum communication distance is short and limited to the closest sensor. Additionally, multi-hopping ensures even some of the close proximity network members are disconnected, the ones connected and hearing the messages forward to further network members. Figure 1.2 can be helpful to visualize such a system.

In [22] radio-acoustic sounding system is used for observing the change in the speed of sound using the physical relation between air temperature and the speed of sound. This system is promising for the early detection of forest fires and it has a long range of detection. However, strong winds can add noise to sound speed detection and the system can require a significant amount of fire to observe heat-related sound speed changes on the radar screen.

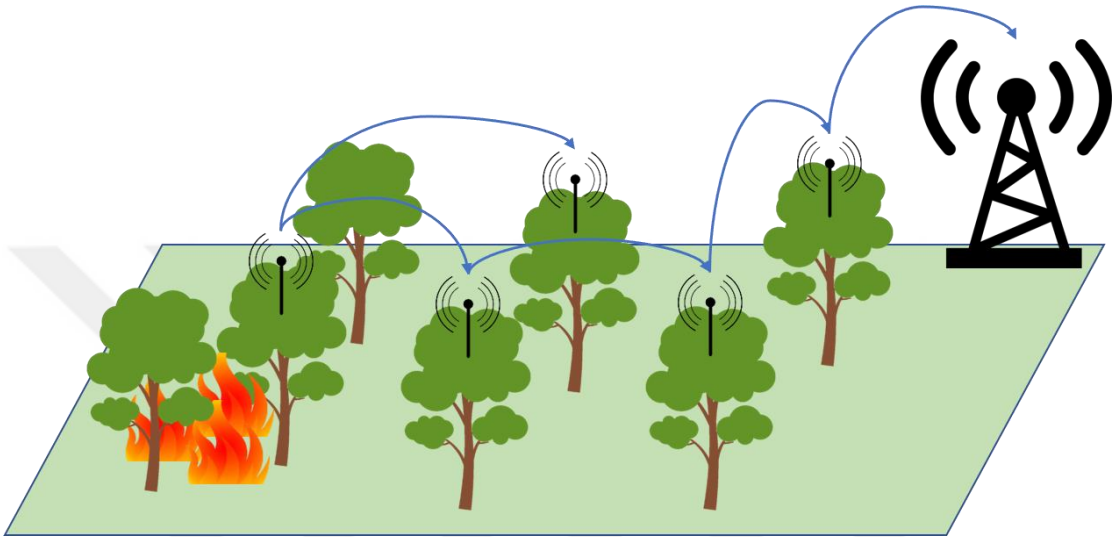

Figure 1.2 Visualization of multi-hopping communication between multiple wireless sensors placed in a forest area.

Authors in [23] fuse many wireless sensing systems with different properties such as optical smoke detectors, microwave, and gas sensors for early detection of forest fires. They additionally suggest using a UAV to check if sensor alarms are correct or not. In this case, an employed UAV with a thermal camera and gas sensors flies to the location of alarming sensors. If there is a forest fire, the UAV is further used as an observer to help firefighters. Furthermore, a blimp is employed with many sensors to check and detect the re-ignition of fires. The overall system covers many aspects of forest fire detection and fighting. However, their solution for false alarms increases system complexity by additionally introducing aerial vehicles to the system. As our study suggests, the overall mission of [23] can be done with only a UAV, without needing additional wireless sensors in the forest.

During the literature review, it is observed that most of the studies utilize a group of mobile robots as wireless mobile sensor networks. In [24] authors utilize a team of UAVs and UGVs for fire detection and monitoring. This study utilizes a similar multi-robot framework to the work of this thesis for forest monitoring and fire detection. In their

study, the proposed mobile-robot team of UAVs and UGVs move in formation during different periods of the mission. Specifically, UGVs carry UAVs to a region to scan and detect fire if any. When UGVs arrive at the location, UAVs take off and scan the environment while preserving their formation. Additionally, if they detect a fire, one of the UGVs takes sensory data from UAVs and computes new trajectories of UAVs for fire observation. However, most of the study covers only the control model of UAVs and the ground robots during their trajectory following missions, and no real-world or realistic simulation experiments are presented in the study. Although it is not mentioned what kinds of sensory equipment are used for the localization and ranging of robots in [24], in this thesis UAV localization and leader following are simplified thanks to the utilization of ultra-wideband (UWB) sensors.

1.1.2 Multi-Robot Operations

Multi-robot operations have been attracting many researchers for years. Distributing the total workload to many robots, or having different robot types to complete a task by benefiting from each robot's strong sides is more efficient than having a single big robot. Here, we focus on the systems that only utilize UAV-UGV interactions in the literature review, noting that many unique algorithms and systems for this purpose exist in the literature.

In [25] authors designed a hawk-eye structure for ground robots to operate in a target-reaching problem. In their system, the aerial vehicle takes off to an altitude and observes the environment. The target region is marked with a light source whose color is known. Additionally, the ground robots have three light sources on them for the UAV to detect and compute their orientation with respect to the target region. By continuously observing the target region and ground robots' orientations, the UAV gives orientation commands to ground robots to help them find the target region. As the authors claimed in their study, the system is designed as a workbench for further research; however, such light-based marker mechanisms can be a good alternative to QR-code-based markers since QR-based systems cannot be detected in dark without additional lighting.

The study in [26] uses two drones to guide and inform a ground robot. The drones create a formation and fly above the ground robot while taking images of the environment. The collected images and formation information is then used to extract three-dimensional

information about obstacles. Briefly, the ground robot uses these drones as a stereo camera to obtain beyond visual line of sight (BVLOS) obstacle information. Even though the experimental results show the system performance is sufficient in a lab environment, it might be hard to implement in an environment where formation-disrupting winds and dense obstacles exist. However, the idea of using stereo cameras for BVLOS operations of ground robots can be promising in order to optimize the path planning and decision-making operations of the ground robot.

In [27], authors presented a novel visual-inertial-range-odometry (VIRO) system for UAV-UGV or heterogeneous robot systems. Briefly, the system utilizes both visual-inertial odometry and UWB ranging for accurate localization of different robot types for the task of 3D scanning of structures like buildings, trees, etc. As is the case in our study, fusing visual and inertial data and range measurements boosts the accuracy of localization. This is due to, firstly, UWB ranging produces low bias and noise if they are well calibrated. Secondly, any integration operation or any other operation that results in a cumulative increase in error is not needed with UWB ranging. The ranging directly gives the actual distance plus some noise error between two nodes. By using the VIRO method, authors increased the performance of 3D scanning tasks with autonomous robots.

In the study presented in [28], a UAV scans the environment and extracts environmental information to create a map for the ground robots. The UGV uses the created map to optimally compute a path to the target location using an enhanced version of the A* path planning algorithm [29] which is a heuristic-based graph search algorithm. During the operation, the UAV updates its position by checking the progress of the UGV under the current scene provided by the UAV camera. The proposed system needs GPS for robots to localize themselves. In a similar study [30], a UAV collects images of the ground as it passes over and sends images to the ground robot. The ground robot then processes these images to detect obstacles and then creates a map. The resulting map is then used for path planning with the proposed hybrid path planning algorithm. However, one of the main contributions of the study is the effect of applying image denoising on the performance of map creation. The authors improved the performance of the system by additionally processing and correcting images before sending them to the ground robot.

A reinforcement-learning approach is presented in [31] for UAV-UGV path planning. In this system, similar to [26], [28], [30] the UGV takes images of the environment and

computes occluded and non-occluded areas to create a binary map. The UGV uses this map to plan its path to the target location. However, instead of keeping all the information to form a global map of the environment the study considers each observed separated location as a game map and trains the ground robot to act on the given map. This way the robot learns which way to take in a given environment. In a non-changing map, this approach can save memory and additional computations. However, in a dynamic environment, the system's success cannot be guaranteed.

1.1.3 Path Planning and Obstacle Avoidance

Whether a robotic manipulator or a wheeled or legged ground robot, it is almost mandatory to have a path-planning algorithm for successful operation. If the goal is moving from point A to point B in the mission space, path-planning algorithms are used for the optimization of the path-taking task. This way, the target can be reached in less time or the robot can take the shortest possible path. There are many different approaches to path planning operations. However, in general, path-planning algorithms can be divided into two main groups. These are global path planning and local path planning. Global path planning algorithms use the information available about, as the name suggests, a global map of the environment. Briefly, algorithms use three core information, the initial position of the robot, the target position, and the map of the environment.

How these three core pieces of information are handled depends on the algorithm. In the literature, global path planning algorithms are divided into three main categories. These are occupancy grid-based algorithms, evolutionary algorithms, and deterministic algorithms. Occupancy grid maps are widely used in computer games [32]. However, in robotics, during simultaneous localization and mapping (SLAM) operation, the map of the environment is sometimes created as voxels in 3D or occupancy grids in 2D in an 8-connected cell configuration as shown in Figure 1.3. This way grid-based path planning algorithms are directly used on the obtained map.

A well-known A* algorithm [29] is a heuristic search algorithm that searches a series of connected edges from start to goal location. The algorithm efficiently searches and eliminates costly branches with the help of a heuristic function. In robotics, Euclidean distance or L2 distance is commonly used as a heuristic for the path-planning operation. With the help of this function, at each grid, the algorithm compares the neighboring grid

cells and adds the best one to the list. In the end, a series of the best (shortest) list of grids is obtained. The robot then follows the order given in the list. To be able to complete this path planning task, A* search algorithm requires full knowledge of the map. However, in the end, the algorithm provides the optimal result which means the shortest possible path. Additionally, the algorithm is complete which means if there is a path, the algorithm finds it at worst after checking every node.

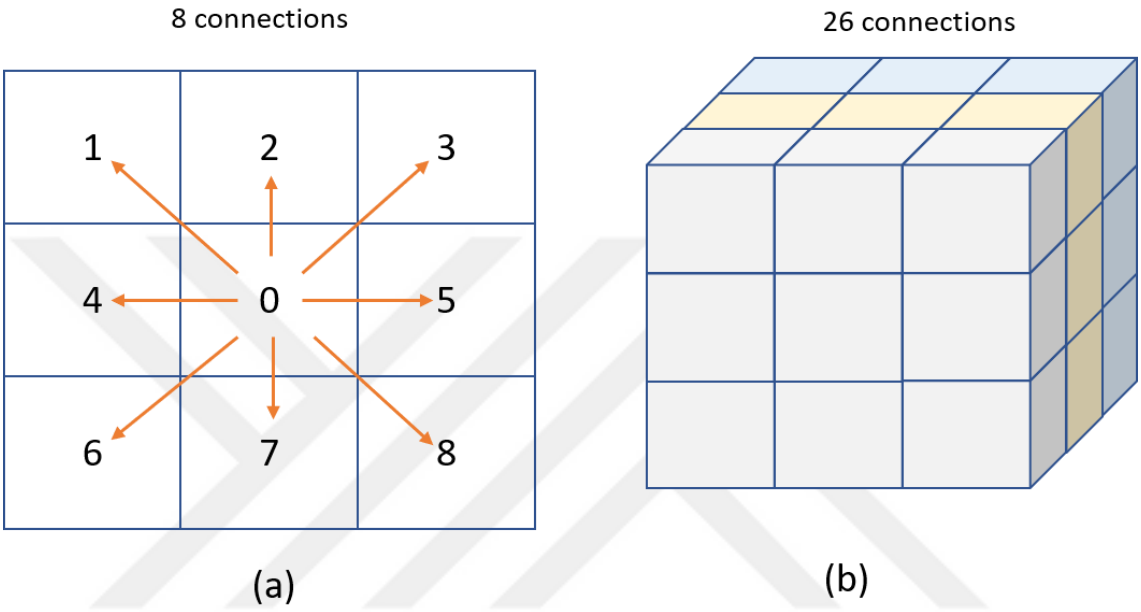

Figure 1.3 Grid connections. (a) 8-connected grids in 2D. (b) 26-connected voxels in 3D.

Different versions of the A* are provided in [33]–[35]. In these algorithms, instead of a global comparison of the edges, the edges local to the position of the robot and its path are only compared with their neighbors. The resulting algorithm runs much faster than the A* algorithm and it can be used in an unknown map. In general, the algorithm is adjusted so that it assumes no obstacle on its path since the robot does not know the map, as the robot moves and discovers obstacles, the algorithm is re-run and updates the path accordingly. This re-run process is executed over the last computed position. This incremental searching decreases the execution time further.

The problem with grid-based path planning algorithms is that the resulting path is only optimal in a grid map not in a continuous environment. Additionally, a robot following the generated path by a grid search algorithm performs inefficient maneuvers. Increasing the map resolution and creating smaller grids can improve the result. However, the increase in the number of grids exponentially affects the runtime performance of the

algorithms. Such a problem is partially solved in [36]. The Field D* algorithm is a modification of the D* Lite algorithm [34]. This algorithm computes the path similar to D* Lite, however, it applies interpolation operations on the generated path during planning. The paths resulting from these interpolations are much shorter in general. Furthermore, interpolation operation smooths the planned path, making robot maneuvers more efficient. The number of grids entered in the interpolation operation depends on the algorithm and environmental information.

In some cases, Field D* cannot produce longer paths with interpolation even though there exist shorter actual paths. This is due to the non-optimal amount of interpolation sizes. Such a case is visualized in Figure 1.4. Interpolation operations and path planning can be used in both 2D and 3D path planning [37]. This makes Field D* applicable for UAV operations as well. However, in this study, the UAV path planning is not included by assuming that the UAV has a high enough altitude not to confront an obstacle. Additionally, it is assumed that the drone takes the fire location on its front by rotating and only follows a straight path to the target location.

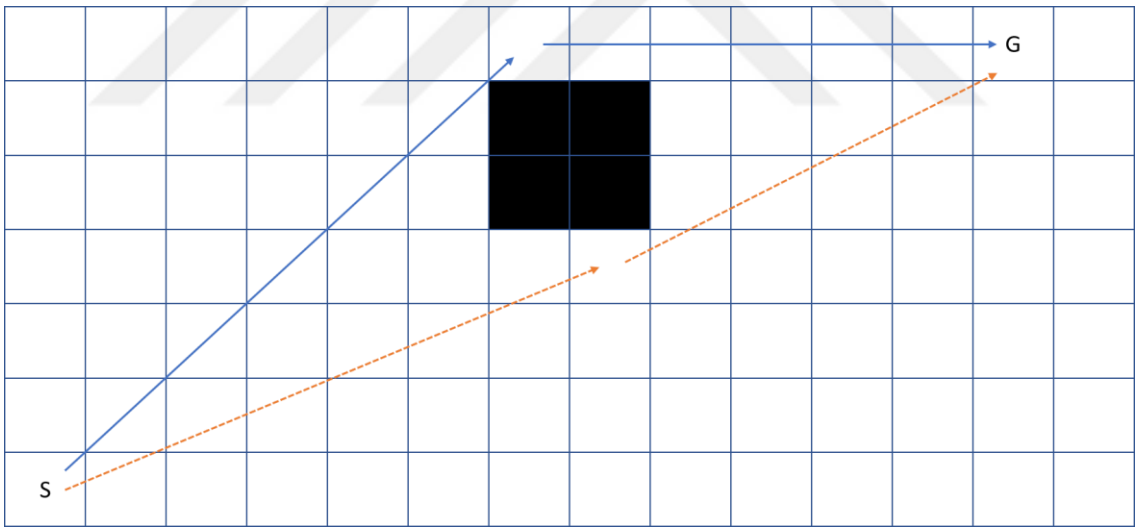

Figure 1.4 An example of Field D* being unable to produce a shorter path due to the disability of handling a large obstacle in interpolation operation and it generates the path shown with blue arrows. (Blue arrows) grid-based shortest path. (Dashed orange arrows) Actual shortest path.

Chapter 2

System Definition

In this study, early forest fire detection and response system is proposed by employing an autonomous multi-robot team. Specifically, the system consists of two robots, one is an autonomous drone and another one is an autonomous ground vehicle. The study covers the early detection of fire, localization, and arrival of robots to the fire location. However, the study does not contain any fire modeling or fire-fighting concepts. The reason behind using multiple robots is to benefit each robot type's strong properties. The drone can fly above trees or any other vegetation, and by using its onboard camera, it can observe far more distance than a camera on the ground vehicle. Additionally, since smoke rises in the air, detection of smoke becomes easier from a drone's perspective. The ground robot can carry payloads and its dynamics and control are less complexes with respect to the drone. Combining these two robots in a team can help detect forest fires and carry fire extinguishers to the fire location efficiently.

The simplified version of the normal operation of the system is schematized in Figure 2.1. The drone periodically takes off to a predefined altitude, scans its environment, and lands again. In the case of a fire or smoke detection, the drone informs the ground vehicle and starts moving toward the fire location at a constant speed. With the assumption of the drone's altitude is high enough so that there will not be an obstacle issue, the drone will be following a straight path to the target location. Self-localization algorithm between drone and ground vehicle constantly measures the ranges between onboard UWB radio frequency sensors. Trilateration of measured distances reveals the two-dimensional location of the drone in the body frame of the ground robot. By keeping their distances lower than a threshold, both robots move toward the fire location.

On the other hand, the ground robot uses an obstacle avoidance algorithm to avoid any crashes. During escaping from obstacles, the bearing angle between the ground robot and

the drone changes and updating the drone location periodically, prevents any failure of the drone-following operation during these baring-angle changes.

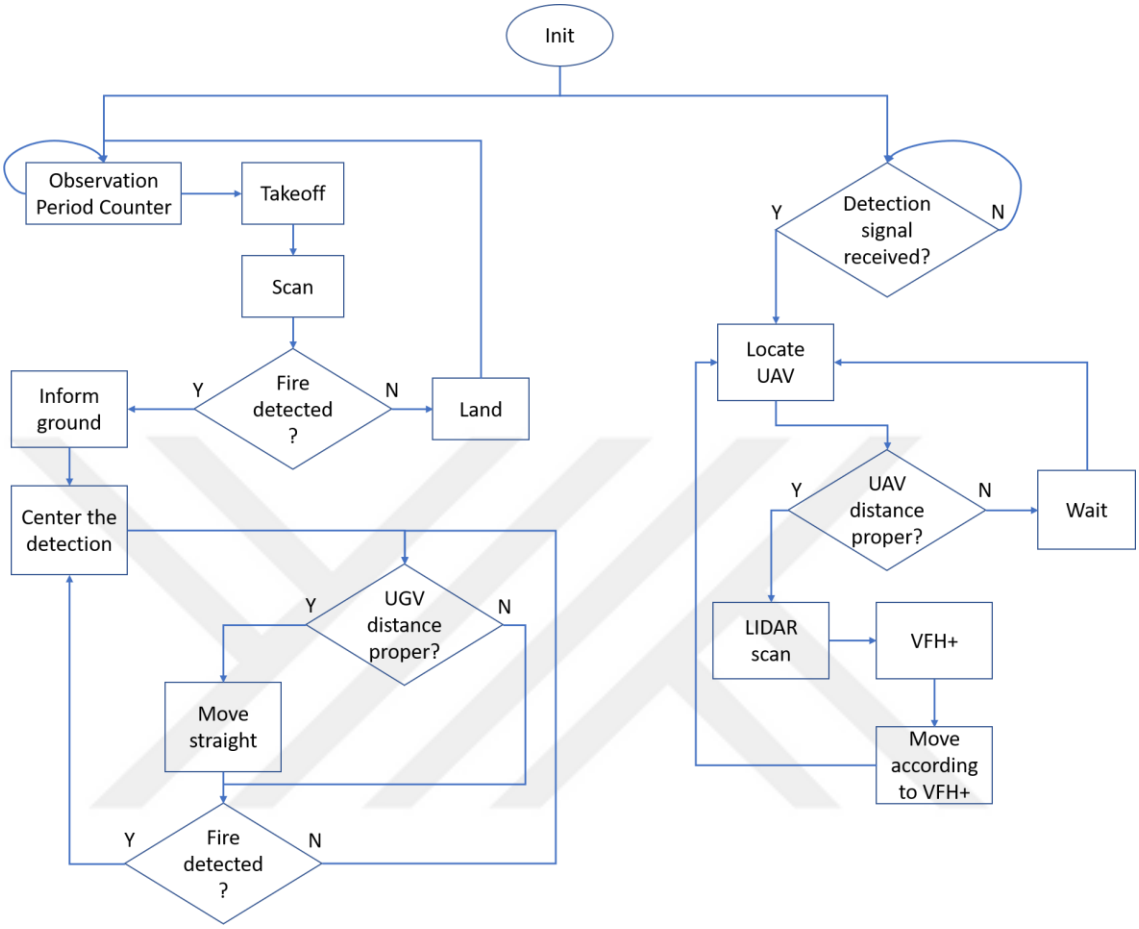

Figure 2.1 Flowchart of the overall system steps.

An important point of fire detection and response is false detection. If the drone detects false-positive fire or smoke, the cost of taking action and moving toward the detection is high. On the other hand, if the drone detects a fire that is true-positive and during the response, the detection algorithm might not be able to detect fire for a period. In this case, the cost of canceling the operation is much higher. To deal with these two cases, the drone collects many images from different angles to be sure if fire or smoke actually exists at the beginning of the mission. In the following subsections, the main parts shown in Figure 2.1 are explained in detail.

2.1 Periodic Observation and Fire Detection

At the start of every take-off period, the drone is elevated to a predefined altitude. The altitude estimation is obtained using sensor fusion between UWB ranging and onboard

static-pressure sensor. A LIDAR sensor is not used in this study for altitude estimation since LIDAR measurements vary during drone flight due to the trees, rocks, or other objects in a forest environment. The static air pressure information can be converted to the altitude thanks to the effect of gravity on air molecules. As the altitude increases, the number of air molecules decreases in volume, and hence, air pressure decreases. The (2.1) relates air pressure to altitude with respect to sea level where h is the altitude in meters, h_b is sea level height which is 0 meter in this case, T_b is the temperature at sea level, L_b is the temperature lapse rate, P is the measured pressure in atmospheric pressure, P_b is the pressure at sea level, g_0 is the gravity, M is air molar mass in kg, R is the universal gas constant.

$$h = h_b + \frac{T_b}{L_b}\left[\left(\frac{P}{P_b}\right)^{\frac{-R \cdot L_b}{g_0 \cdot M}} - 1\right] \quad \textbf{(2.1)}$$

Fire spreading speed depends on many factors like wind speed, air temperature, humidity, burning material, etc. These factors are not constant and in a practical application, the observation period should be adjusted dynamically by considering these factors. The period should not be too long, in case of a fire starting right after the last landing would not be detected until the next take-off. A too-short observation period can decrease the battery level of the drones quickly hence battery level might not be enough after detecting and moving toward the target. In a scenario where the battery level is well managed and the drone observation period is low enough to detect a fire early, the system uses deep learning algorithms for the recognition of fire or smoke. However, for this system to work, recognition of fire/smoke is not enough since the UAV needs to locate the fire/smoke to reach that place. For this reason, the problem is defined as object detection where recognized objects in an image are additionally marked with their location in the image frame.

Complex object detection problems can be solved with machine-learning approaches. Specifically, for the task of object detection, artificial neural networks can be implemented. Even though artificial neural networks perform well in this kind of problem, a different version of ANNs called convolutional neural networks (CNN) performs quite well in computer-vision tasks.

One of the famous CNN-based approaches for real-time object detection is the well-known YOLO (You Only Look Once) algorithm family. Currently, the YOLO family has 7 major versions for real-time object detection. In this study, YOLO version 7 [38] is used for fire and smoke detection tasks which remains one of the highest-performing real-time object detection algorithms at the time of this study. Additionally, the YOLOv7 model is sufficiently fast in terms of computation time and low in the number of parameters so that it can be loaded to power and payload critical systems such as UAVs.

Figure 2.2 Example images with detections using trained YOLOv7 object detection algorithm.

The YOLOv7 model tries to detect objects in three different scales, so that small, medium, or large versions of the same objects can be recognized and located in the image frame. At the end of detection, the model gives a list of object classes and their coordinates with

the size of enclosing boxes. An example of performed object detection with YOLOv7 is presented in Figure 2.2.

The drone uses information about detected objects and the enclosing boxes to center the image x-axis of the boxes in the image frame. The camera-to-object axis and drone body center-to-object axis are the same in this study. Hence centering an object in the image frame results in fronting the object in the body frame. The drone accomplishes this centering operation by adjusting its yaw angle with a constant angle step.

When the drone detects forest fire, it collects multiple images while hovering. If the detections are predominant in all frames, the drone makes small changes in its perspective and checks if the fire or smoke is still detectable. The predominance of fire or smoke during this procedure eliminates false detection of fire. After detection, false detection filtering and centering of fire or smoke, the drone informs the ground vehicle over UWB communication and ranging link and then starts flying toward the fire location.

2.2 UAV-UGV UWB Communication and Ranging

In the proposed framework, the drone and the ground vehicle communicate continuously and inform each other about their findings and about what will be their next action. Additionally, the two robots measure their distances thanks to the unique features of UWB communication. The ground vehicle is equipped with three UWB sensors and the drone is equipped with one UWB sensor since low energy consumption and payload amounts are critical for the drone.

With this setup, at every communication period, a total of three communication instances are carried out. This way three distance measurements are obtained by the ground vehicle. For each range measurement, the ground robot sends a start message with a time portion to the drone. When the drone receives this message, it adds predefined message reception, CPU computation, and antenna transmission delays to the time portion of the message and transmits this new message to the ground robot. The ground robot then computes the differences in time and computes the range using the time-of-flight (ToF) technique. The overall time of flight (ToF) ranging using UWB is shown in Figure 2.3.

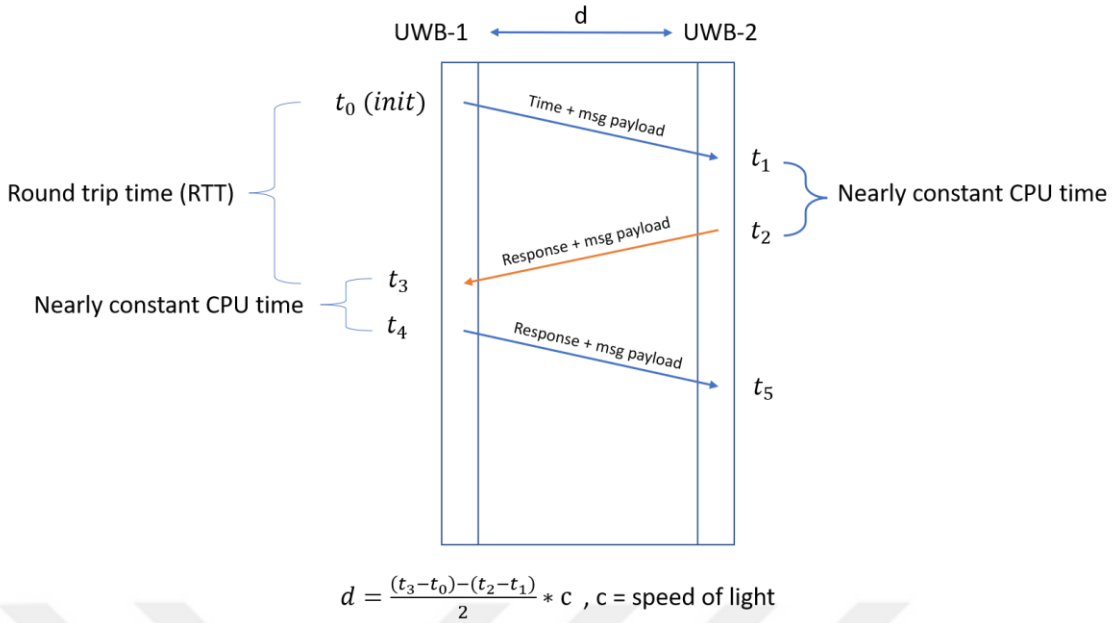

Figure 2.3 Double-sided two-way ranging scheme for UWB communication.

Since the onboard UWB sensors are placed at different points and their locations are known in the body frame of the ground vehicle, the robot can locate the drone in its own body frame using the following series of linear least-square operations **(2.2)** and **(2.3)**.

$$\boldsymbol{A} = \begin{bmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \end{bmatrix},\ \boldsymbol{b} = \begin{bmatrix} \frac{1}{2}(r_1^2 - r_2^2 + d_{21}^2) \\ \frac{1}{2}(r_1^2 - r_3^2 + d_{31}^2 \end{bmatrix}, \tag{2.2}$$

$$\begin{bmatrix} x_{UAV} \\ y_{UAV} \end{bmatrix} = (\boldsymbol{A^T A})^{-1} \boldsymbol{A^T b} + \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}, \tag{2.3}$$

where $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ are locations of UWB modules on UGV, r_1, r_2, r_3 are the measured distances from UAV to UGV, d_{21}, d_{31} are the distances between UWB modules on UGV.

The result of the trilateration is the X and Y coordinates of the drone in the body frame of the ground robot. The position in Z is obtained using the altitude information of the drone. This way computational requirements are decreased for relative localization. Actually, Z information is not crucial since X and Y information is used for target direction assignment for the ground robot. The inaccuracies caused by not measuring and hence not including UAV height in localization computations do not have a significant effect on localization since location information is only used for target direction

assignment. Additionally, the direction estimation is frequently updated and filtered by the ground vehicle to eliminate inaccuracies and measurement noises.

This step of relative localization and formation control between the drone and the ground robot is the main contribution of this study. In a scenario, where a ground robot is not equipped with UWB sensors and localization is performed using only the initial relative states of the two robots and their odometry systems are not easily implementable in a real-world environment.

In such a scenario, first, both robots would be sharing their odometry results frequently to filter and constantly locating themselves will increase communication amount. Additionally, the ground robot must use advanced sensory equipment and heavy computations to track its location precisely. In this situation, system complexity, total sensors used, and system cost will increase drastically. However, in this study, the ground robot can get feedback from UWB range measurements, and the drone gets its feedback from object detections. This way error rates are only limited by single measurement errors and sensor noises at that specific time and this error rate can be decreased by filtering and signal processing. On the other hand, in an odometry scenario, the total error will be increasing cumulatively due to integral or summation operations during Kalman filtering, etc.

2.3 Moving Towards a Target, Local Planner

When the drone detects a forest fire, it starts moving toward the fire location. Since it is assumed that the drone's altitude is sufficiently high, we exclude the probability of collision with obstacles on its way to the target. However, the ground vehicle cannot have such an assumption since a forest environment is full of obstacles for a ground vehicle. In this case, it is mandatory to have an obstacle avoidance algorithm. For this reason, the ground robot utilizes an obstacle avoidance and local planning algorithm titled Vector Field Histogram Plus (VFH+) [39]. This algorithm is chosen due to its direct interaction with LIDAR measurement results. The robot uses LIDAR for detecting and locating obstacles around its body. The measurements coming from the LIDAR sensor, and target direction (i.e., drone relative position) are fed into the VFH+ algorithm and as a result, the algorithm gives a new direction of movement by considering robot dynamics. Therefore, the robot tries to keep its distance from obstacles over a predefined threshold

and follows the target direction while avoiding obstacles. Steps of the VFH+ algorithm are presented in the following Figure 2.4.

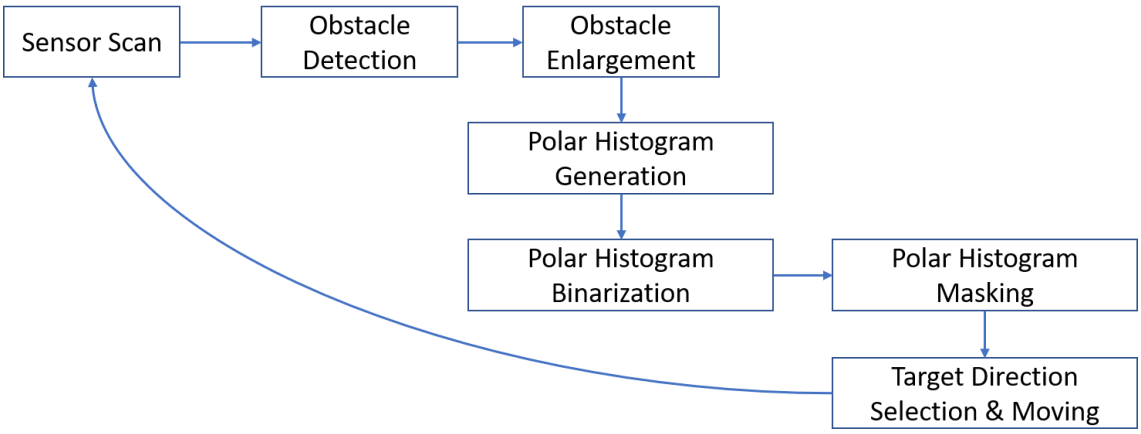

Figure 2.4 General operational processes of VFH+ algorithm.

At first, the LIDAR measurements are truncated according to the maximum obstacle range parameter. This parameter is chosen not to include the obstacles which reside outside a disk with a certain threshold. Secondly, the measurement results are scanned for the detection of obstacles and their edges.

Since the measurement indices of LIDAR scan measurements can be converted to body-frame bearing angles, this information is used to locate obstacles in the body frame. The result of the transformation operation is a polar histogram with a predefined angle resolution. If the target direction is obstacle-free, the robot tries to move there; however, if the target direction is occluded, the ground vehicle picks a new direction according to the polar histogram. Even if the ground robot rotates in a different direction due to an obstacle, periodic range measurements coming from on-body UWB sensors and the relative localization algorithm update the target direction. At each time segment, the UGV local path planner updates the target direction according to the result of the localization algorithm.

With this structure, localization error stays bounded to the UWB measurement errors. In an odometry-based system, the ground robot would have to keep track of its feedback sensors like wheel encoders, inertial measurements, etc. As UWB measurements have noises on them, these odometry sensors have too. However, every time the odometry system measures and integrates sensor measurements to figure out how the vehicle moved, the algorithm additionally integrates sensor noises and biases. In the long run, the

deviation from the actual position to the odometry output will be much larger than the UWB measurement case. This is due to the fact that the UWB measurements are updated at every period and previous measurements are discarded directly or only used for filtering high-frequency noises. This way maximum error is limited to the maximum error of range measurement where the maximum error of the odometry case increases as the operation time goes to infinity.

Chapter 3

Method

In a forest environment $\mathcal{E}$, a vertical take-off and landing (VTOL) capable UAV A and a four-wheeled non-holonomic unmanned ground robot G are employed. The UAV has holonomic kinematic constraints such that it can move in any direction in a three-dimensional (3D) space if there is no obstacle in $\mathcal{E}$. Additionally, the orientation of A can change in any of the three Cartesian axes. Assuming that the motions in the three axes are independent from each other (i.e., these axes are decoupled), the position and orientation of **A** at a time $t + T_s$ with given linear and angular velocity vectors $\mathbf{v_t^A}$ and $\boldsymbol{\omega_t^A}$ can be represented as follows:

$$\mathbf{p}_{t+T_s}^{\mathrm{A}} = \mathbf{p}_t^{\mathrm{A}} + \mathbf{v}_t^{\mathrm{A}} T_s + \boldsymbol{\eta}_t^{p^A}, \tag{3.1}$$

$$\boldsymbol{\varphi}_{t+T_s}^{A} = \boldsymbol{\varphi}_t^{A} + \boldsymbol{\omega}_t^{A} T_s + \eta_t^{\varphi^A}. \tag{3.2}$$

Here, $\mathbf{p}_{\mathrm{t}}^{\mathrm{A}} = [x_t^A \; y_t^A \; z_t^A]^T$ (x, y, z are the positions with respect to a global frame $\mathcal{F}_G$) and $\boldsymbol{\varphi}_{\mathrm{t}}^{\mathrm{A}} = [\alpha_{\mathrm{t}}^{\mathrm{A}} \; \beta_{\mathrm{t}}^{\mathrm{A}} \; \gamma_{\mathrm{t}}^{\mathrm{A}}]$ (α, β, γ are roll, pitch, yaw angles of A). $\boldsymbol{\eta}_t^p \; and \; \boldsymbol{\eta}_t^{\varphi}$ are positional and orientational Gaussian noises which can be defined as $\boldsymbol{\eta}^{\mathbf{A}} \sim \mathrm{N}(0, \Sigma_{\mathrm{p}}),\ \boldsymbol{\eta}^{\boldsymbol{\varphi}} \sim \mathrm{N}(0, \Sigma_{\varphi})$ where $\Sigma_{\mathrm{i}} = \sigma_{\mathrm{i}}^2 \mathbf{I}_3$. It should be noted that, during flight, the roll and pitch axes and the altitude of robot A are controlled by its low-level controller. Thus, it is desired to control the yaw ($\gamma_{\mathrm{t}}^{\mathrm{A}}$) for orientation and $\mathrm{x}_{\mathrm{t}}^{\mathrm{A}}$ $\mathrm{y}_{\mathrm{t}}^{\mathrm{A}}$ for position control. Also, we assume that $\mathrm{z}_{\mathrm{t}}^{\mathrm{A}} = \mathrm{h}$ is constant during flight. Therefore, $\mathbf{v}_{\mathrm{t}}^{\mathrm{A}}$ and $\boldsymbol{\omega}_{\mathrm{t}}^{\mathrm{A}}$ vectors have some indexes with zero value.

Ground robot G has the following forward kinematics equations due to nonholonomic constraints:

$$\mathbf{p}_{t+T_s}^{\mathrm{G}} = \mathbf{p}_t^{\mathrm{G}} + [\cos(\theta_t^G) \quad \sin(\theta_t^G) \quad 0]^{\mathrm{T}}\, \mathbf{v}_t^{\mathrm{G}} T_s + \boldsymbol{\eta}_t^{p^G} \tag{3.3}$$

$$\theta_{t+T_S}^G = \theta_t^G + \omega_t^{\mathrm{G}} T_s + \eta_t^{\theta}, \tag{3.4}$$

where $\boldsymbol{\eta}_{\mathrm{t}}^{\mathrm{p^G}} \sim \mathrm{N}(0, \Sigma_{\mathrm{G}})$ and $\Sigma_{\mathrm{G}} = \sigma_{\mathrm{G}}^2 \mathbf{I}_3$. In other words, its altitude $\mathrm{z}_{\mathrm{t}}^{\mathrm{G}}$ is constant, and the ground robot can change its orientation about $\mathrm{z}_{\mathrm{t}}^{\mathrm{G}}$ by controlling $\theta_{\mathrm{t}}^{\mathrm{G}}$. In addition, robot G has rigidly mounted three UWB sensors as shown in Figure 3.1 at the following locations:

$$\mathbf{p}_t^{\mathbf{u_i}} = \mathbf{p}_t^{\mathrm{G}} + \mathbf{R}(\theta_t^G)\mathrm{r_i}, \tag{3.5}$$

where $\mathrm{r}_1 = [0 \quad \mathrm{m}]^{\mathrm{T}}$, $\mathrm{r}_2 = [0 \quad -\mathrm{m}]^{\mathrm{T}}$, $\mathrm{r}_3 = [\mathrm{m} \quad 0]^{\mathrm{T}}$ and $\mathbf{R}(\theta)$ is the 2x2 rotation matrix. The robot A has a single UWB sensor rigidly mounted to its airframe's center-of-gravity.

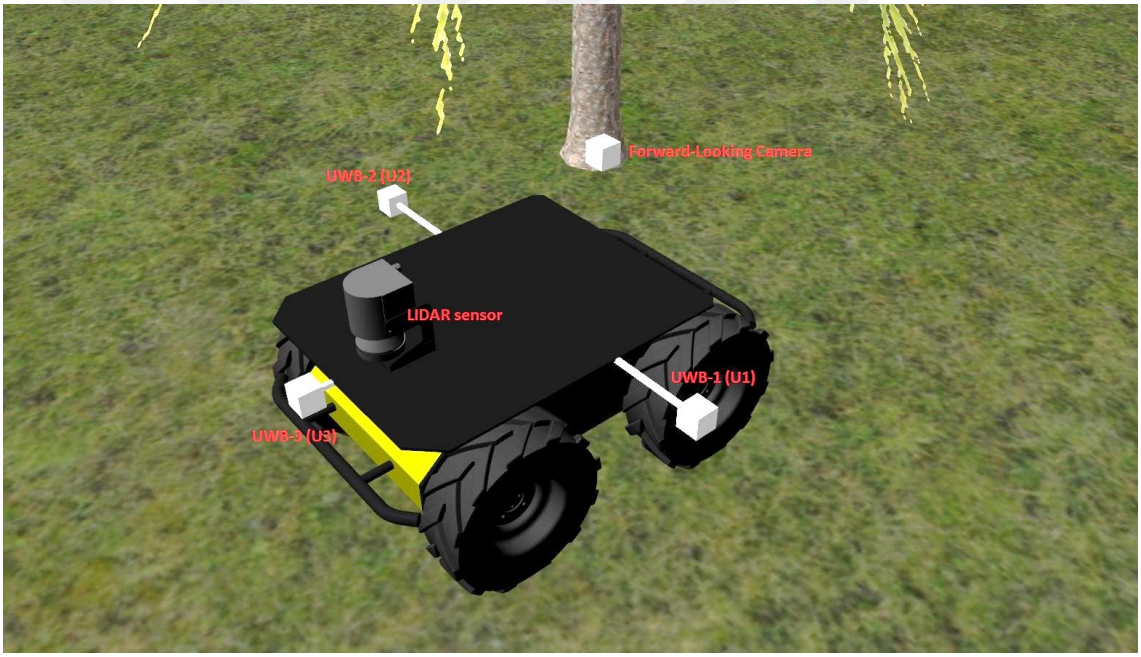

Figure 3.1 UWB locations on the ground robot. The forward-looking camera of the UGV is only used for debugging purposes.

The three distances between UWB sensors on G to UWB sensor on A can be defined as:

$$d_t^i = ||\mathbf{p}_t^{\mathrm{u_i}} - \mathbf{p}_t^{\mathrm{A}}|| + \delta_t^i, \qquad i \in \{1,2,3\}, \tag{3.6}$$

where, $\delta_{\mathrm{t}}^{\mathrm{i}} \sim \mathrm{N}(0, \sigma_\delta^2)$, and $\mathbf{p}_t^{\mathbf{u_i}}$ is the position vector of UWB sensors on G. With these given kinematics and models, the robots A and G have two main objectives:

Objective 1: Fire detection using onboard resources and target (fire-location) reaching.

Objective 2: UAV localization and following while avoiding obstacles.

In the following sections, we study these objectives independently.

3.1 Objective 1: Fire Detection and Target Reaching

For the first objective, robot A uses specially trained YOLOv7 CNN model. For an image frame x, the model f(**X**) can be defined as:

$$f(\boldsymbol{X}) = \ y = \begin{cases} [\ ]\ (empty\ list), & n = 0 \\ [\boldsymbol{c}, \boldsymbol{x_1}, \boldsymbol{y_1}, \boldsymbol{x_2}, \boldsymbol{y_2}], & n > 0 \end{cases} \quad \textbf{(3.7)}$$

where *n* is the number of detections, *c* is a vector of classes (fire, smoke or 0, 1) $\mathbf{x_1}, \mathbf{y_1}$ are the vector of x- and y-coordinates of the top-left corner of bounding boxes in the image frame, $\mathbf{x_2}, \mathbf{y_2}$ are the bottom-right corner coordinates of bounding boxes in the image frame. Assume a 3 by 640 by 640 RGB image frame X_i taken at time t and that $f(X_i)$ detects a single object fire or smoke. Given $\mathbf{x} = [x_1, y_1, x_2, y_2]$ coordinates of the bounding box, robot A uses the following equation to center the image detection in the image frame:

$$g(\boldsymbol{x}) = \begin{cases} -q, \ x_1 + \frac{x_2 - x_1}{2} < \frac{640(pixel)}{2} - \varepsilon \\ q, \ x_1 + \frac{x_2 - x_1}{2} > \frac{640(pixel)}{2} - \varepsilon \\ 0, \quad otherwise \end{cases}, \quad \textbf{(3.8)}$$

Where the design constant q is the yaw angle increment amount, and ε is an allowed error rate in pixels. Robot A uses the result of g(x) for orientation correction. By controlling its yaw angle by q or -q amount, robot A faces the detected fire or smoke. When the detection is centered in the image frame, robot A follows a straight path to the detected place. However, for this algorithm to work, robot A must use a forward-looking camera. To better explain the centering and path following operation, consider the following scenario: For $\varepsilon = 10$ pixels, a forward-looking camera with a horizontal field-of-view (FOV) angle of 60 degrees sees a smoke cloud at a distance of 100 meters with a diameter of 5 meters (assuming that the smoke cloud is disk-shaped) in the center of the image frame. If the camera has a 640 by 640 pixels resolution, the horizontal length of the cloud in image pixels would be (discarding any nonlinearities, 3D projection errors, etc.):

$$2 * \left(\frac{2.5m * 320px}{\tan\left(\frac{\pi}{6}\right) * 100m} \right) \cong 28\ pixels \quad \textbf{(3.9)}$$

This shows that for the 5-meter horizontal length of smoke at 100 meters away, the ε error rate will be less than the object size of 5 meters. If the distance from the camera to the smoke decreases to 80 meters, smoke would take around 34 pixels and around 55 pixels at 50 meters away. Even if the accuracy of the centering operation is low at far distances, as the drone gets closer to the detected object, the result of object centering will be more accurate and robot A will reach the fire location by repeating centering and straight-path-following operations.

During robot A's motion toward the target region, the distance between A and G can increase to a level where UWB ranging and communication is not possible anymore. To avoid such a situation, robot A limits its maximum distance to robot G to the predefined level of l_{AG}. If the distance between them exceeds l_{AG}, A hovers and only tries centering the detections in the image frame.

3.2 Objective 2: UAV Localization and Following

The second objective is the main task of robot G. Here, after robot A informs robot G about fire detection, robot G waits for A to move away for a predefined distance limit l_{GA}. Separating robots A and G by a certain distance is required for improving the performance of UWB-based localization. In our system, A has an elevation over the ground plane during flight. To locate A in a 3D space, G must have at least 4 UWB sensors (4 independent ranges). However, A can track its altitude with respect to the ground plane using onboard resources such as a barometer and/or a downward-facing laser sensor, and can share this information with G. With the altitude information, G can locate A with 3 range measurements since the system can be reduced into 2D space by extracting information related to third dimension. Indeed, for the system to work, no altitude information is required since the least square operation has only 2 dimensions, and the distance error caused by not considering drone altitude cancels out during the calculation of vector **b** in **(3.10)**. At any time t, by employing the measured distances between UWB sensors d_t^i, $i \in \{1,2,3\}$, we reach the following set of equations:

$$\mathbf{A} = \begin{bmatrix} \boldsymbol{r_2 - r_1} \\ \boldsymbol{r_3 - r_1} \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} \frac{1}{2}(d_1^2 - d_2^2 + m^2) \\ \frac{1}{2}(d_1^2 - d_3^2 + m^2 \end{bmatrix}. \tag{3.10}$$

We apply the following least-squares algorithm at each time instant t to generate robot A position:

$$\boldsymbol{X_t} = \begin{bmatrix} \bar{x}_t \\ \bar{y}_t \end{bmatrix} = (\mathbf{A^T A})^{-1}\mathbf{A^T b} + \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}. \tag{3.11}$$

Here, $\bar{x}_t, \bar{y}_t$ are the result of trilateration, and the bearing angle from G to A is $\tau_t = \tan^{-1}(\bar{y}_t/\bar{x}_t)$. Deviations in τ due to measurement noises and biases become maximum if projection of A falls over G (close to or inside of the triangle which the three UWB modules create). Additionally, high sensor noises can result with imaginary results due to square root operations. For these reasons, a large enough minimum distance l_{GA} is required for stable bearing direction calculations. To improve the precision, a weighted least-squares approximation can be used as follows:

$$\mathbf{X_t} = \begin{bmatrix} \bar{x}_t \\ \bar{y}_t \end{bmatrix} = (\mathbf{A^T \Omega A})^{-1}\mathbf{A^T \Omega b} + \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}, \qquad \boldsymbol{\Omega} = \begin{bmatrix} \frac{1}{\sigma_\delta^2} & 0 \\ 0 & \frac{1}{\sigma_\delta^2} \end{bmatrix}. \tag{3.12}$$

Another method for filtering can be using weighted moving average filters for the measurements. $\mathbf{k} = [X_t, \dots, X_{t-nT_s}]$, where $n + 1$ is the filter length and $\boldsymbol{W} = [w_0 \dots w_n]$, w_i is being the filter coefficient. The filtered result would be:

$$\bar{X}_t = \sum_{i=0}^{n} k_i * W_i \tag{3.13}$$

The weights of the least-squares method and the filter size and coefficients depend on the system and measurement noise levels.

For the UAV following task of G, the robot uses bearing angle information τ_t and range measurements to keep at least l_{GA} amount of distance. Here, a modification of VFH+ is used for obstacle avoidance. For a LIDAR scan s_t^i, $i \in \{1,2, \dots n\}$ of size n at time t, robot safety radius $r_{safety} \geq r_{robot}$, and obstacle handling radius $r_{threshold}$, VFH+ algorithm creates a polar histogram of the environment and guides the robot according to the polar histogram and target direction (τ_t in this case). To simplify computations, detected

obstacle edges are enlarged by r_{safety} amount so that robot G is handled as a point object in the X-Y plane. Enlargement operation is done with **(3.14)**.

$$\bar{\imath} = \left\lceil \frac{\sin^{-1}\frac{r_{safety}}{s_t^i}}{\mu} \right\rceil, \quad s_t^{\bar{\imath}:i} = s_t^i \tag{3.14}$$

where μ is the angle increment of each LIDAR beam, $\bar{\imath}$ represents the enlargement amount as indices and from i to $\bar{\imath}$, all indices are assigned with the original obstacle distance. As a second step, $\boldsymbol{s}_t^i$ is binarized:

$$s_t^i = \begin{cases} 1, & s_t^i < r_{threshold} \text{ (Obstacle)} \\ 0, & s_t^i \geq r_{threshold} \text{ (No obstacle)} \end{cases} \quad i \in \{1,2,\dots n\}. \tag{3.15}$$

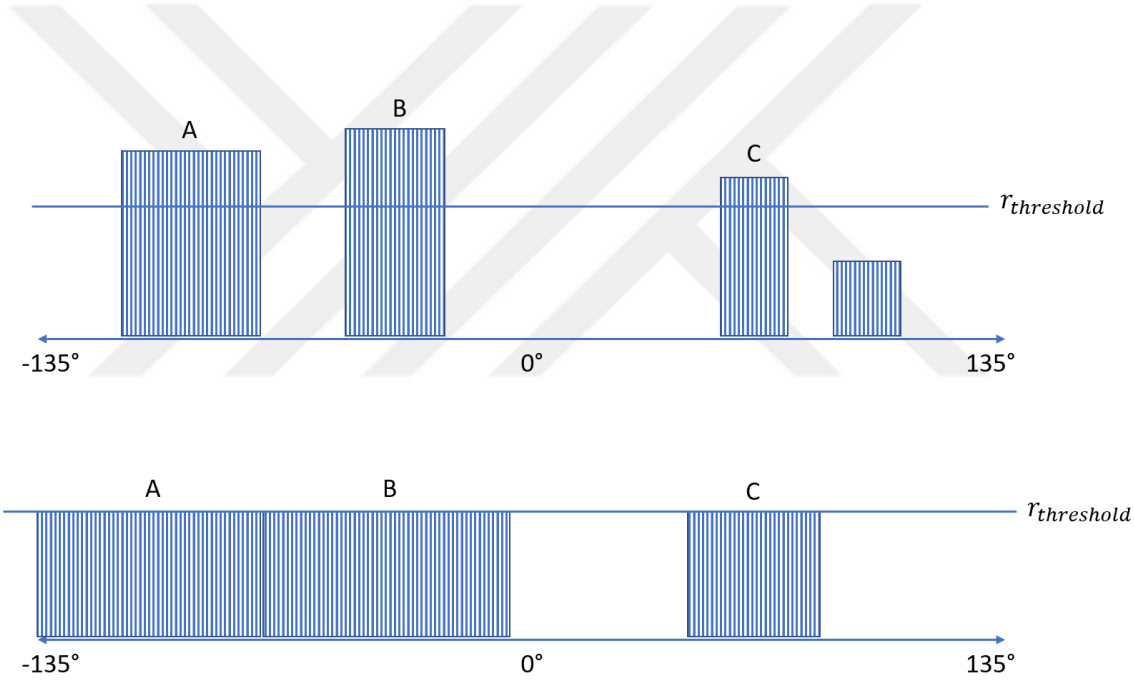

Figure 3.2 Example polar histogram. (a) Histogram with different range magnitudes. (b) Binarized and enlarged polar histogram.

An example of a binary histogram for a LIDAR with 270-degree FOV is presented in Figure 3.2. Additionally, indices (angles) that are out of G's dynamic range (G cannot turn that sharp) are marked as occluded even though the portions are obstacle free. With r_{left}, r_{right} being the left and right minimum turning radiuses respectively, an obstacle detected with s_t^i beam blocks the left or right turning path if and only if:

1. $d_l^2 < r_{robot} + r_{safety}$ *where* $d_l^2 = \left(s_t^i\right)^2 + r_{left}^2 - 2s_t^i r_{left} \cos\left(\frac{\pi}{2} - \frac{i}{\mu}\right)$

2. $d_r^2 < r_{robot} + r_{safety}\ where\ d_r^2 = \left(s_t^i\right)^2 + r_{right}^2 - 2s_t^i r_{right} \cos\left(\frac{\pi}{2} - \frac{i}{\mu}\right)$

Calling the final processed scan result as $\bar{\boldsymbol{s}}_t$, robot G takes a path obeying following rules:

1. If $\left\lfloor\frac{\tau_t}{\mu}\right\rfloor \in \{1,2,\dots n\}\ and\ \bar{s}_t^{\lfloor\tau_t\rfloor} = 0$ take $\left\lfloor\frac{\tau_t}{\mu}\right\rfloor$ as target direction.
2. If $\left\lfloor\frac{\tau_t}{\mu}\right\rfloor \in \{1,2,\dots n\}\ and\ \bar{s}_t^{\lfloor\tau_t\rfloor} = 1$ take closest i to $\left\lfloor\frac{\tau_t}{\mu}\right\rfloor$ which $\bar{s}_t^i = 0$ as target direction.
3. If $\left\lfloor\frac{\tau_t}{\mu}\right\rfloor \notin \{1,2,\dots n\}$ take closest i to $\left\lfloor\frac{\tau_t}{\mu}\right\rfloor$ which $\bar{s}_t^i = 0$ as target direction.

In practice, LIDAR measurements, the assumed approximations, and other nonlinearities can cause collisions if r_{safety} is taken very close or equal to r_{robot}. Hence, a safety distance should be selected by considering these factors. Additionally, a robot with non-holonomic constraints cannot take the selected direction instantly instead it takes a curved path until the orientation and target direction are met. The selection of r_{safety}, $r_{threshold}$ and VFH+ update rate plays an important role in avoiding crashes during rotations too.

Another important aspect of VFH+ implementation is the robot shape. If robot's shape is not circular or its width-to-length ratio is significantly bigger or smaller than one, the VFH+ algorithm should be implemented differently.

3.3 UWB Communication and Distance Calculation

A simple double-sided two-way ranging scheme of two UWB sensors is visualized in Figure 2.3 in Chapter 2. A UWB node (or module) u_i at position $\mathbf{p}_t^{u_i}$, $i \in \{1, \dots, n\}$ where $n = 4$ is the total number of nodes, Node 1 or u_1 sends an initial message at time $t_{initial}$ with an abstract time value (generally the time passed from sensor start) as a payload in the sent message. If the message is successfully received by the sensor u_2 in its own timeframe at t_2, the signal is processed in a predefined (experimentally computed) time and the new message is prepared in a duration of t_{cpu}. Before submission of the message to the antenna, the received abstract time value is summed with the processing time of t_{cpu}, antenna reception delay of t_{rx}, and antenna transmission delay of t_{tx} and transmits the response message at time t_3. These values are experimentally calculated and can be found in the particular UWB sensor's datasheet. Finally, the sensor u_2 responds with the newly generated message. The received message is noted by u_1 as t_{final}. By processing

passed time information and using the constant speed of light in air, u_1 calculates the distance to the u_2 as follows:

$$d_t^{1\leftrightarrow 2} = \frac{\left(t_{final} - t_{initial}\right) - (t_3 - t_2)}{2} * c_{air} \quad \textbf{(3.16)}$$

Here, it should be noted that t_2 and t_3 are in the time frame of u_2 while t_{final} and $t_{initial}$ are in the time frame of u_1. The two nodes do not need to be synchronized or do not need time-frame conversion since only the differences between time values are required for the distance computation and the first three time-stamps ($t_{initial}, t_2, t_3$) are always transmitted in the messages while t_{final} tracked by the distance calculating node. Moreover, the additions of antenna transmission delay and message processing durations are required for the correct measurement of the distance between two nodes. Otherwise, the delay introduced during processing and transmission adds biases to the range measurements and the exact knowledge of the time passing during these processes improves the quality and correctness of the measured distances.

After the above ranging procedure, the computed distance is only known by u_1. However, to inform u_2 about its distance to u_1 there are two possible methods:

i. u_2 loads its own $t_{initial}$ as a 4th label and u_1 responds to message of u_2 by adding its own t_2 and t_3 time stamps and finally u_2 recomputes the distance.
ii. u_1 shares the computed distance information as a simple 3rd massage.

From a practical perspective, method (ii) is simpler and straightforward. However, if any of the sensors are moving so fast that frequent updates to distance measurements are required, method (i) can be utilized since it measures the distance 2 times in 3 messaging sequences while method (ii) only measures 1 time and shares this information between nodes.

In our system, there are 4 UWB modules in total. To make the communication in order and simple, all ranging sequences are started by robot A since it has only one sensor and no initiation sharing between nodes is required in this way. Robot A sends an initial message from the node u_4 with messaging and ranging payloads to u_1 and u_1 respond with time stamps and messaging payloads. Finally, the u_4 transmits computed distance information to u_1 as in method 2. This sequence of messaging repeated between $u_4 \leftrightarrow u_1, u_4 \leftrightarrow u_2, u_4 \leftrightarrow u_3$ in order. The such messaging scheme is visualized in Figure 3.3.

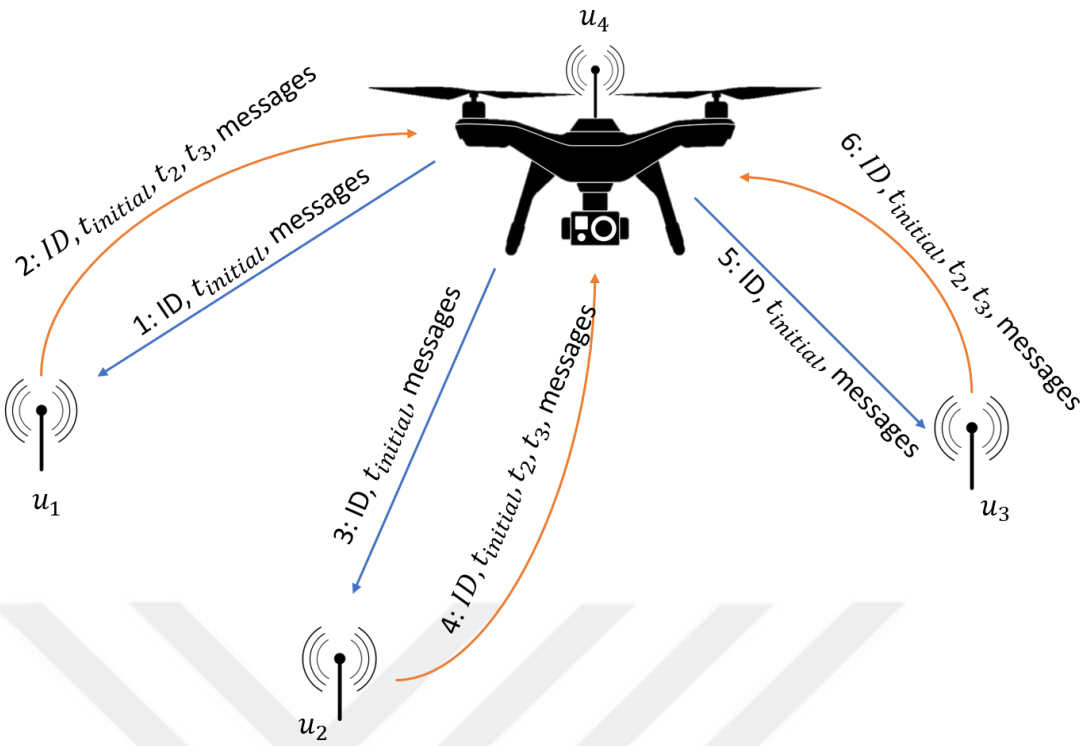

Figure 3.3 UWB communication and ranging sequence between nodes. All ranging sequences are initiated by the UAV node.

The nodes that are not receiving messages to their unique IDs wait for their turn to come. By continuously repeating these processes and following the sequence, the two robots know their updated positions during the operation.

3.4 Position Error Bounds for Unbiased Measurement Noise

In a practical application, for the successful operation of the system, localization errors caused by noises from UWB range measurements should be limited and well-filtered to prevent undesired sudden changes in target angles at any time. These frequent changes in bearing angle can introduce unwanted oscillations on the ground robot, and the robot might fail in obstacle avoidance. Since l_{AG}, l_{GA} and $\mathbf{p}_t^{u_i}, i \in \{1,2,3\}$ are three important parameters for stable operation of the localization system, threshold distances and placement of UWB modules on the ground robot can be optimized.

Assuming well-calibrated UWB modules with no biases in LOS operations, one can model the lower bound standard deviation for position estimation as in [40]. Since the measurement noises rather than the measurement biases oscillate the robot and cause the sudden changes in bearing angle, high noise and zero bias are assumed for lower-bound computations here. As in **(3.6)**, with a constant standard deviation of σ_δ in range

measurements at a distance, [40] computes the position error bound (PEB) by using the Fisher information matrix and Cramer-Rao lower bound (CRLB). Moreover, overall PEB formulation can be simplified to **(3.17)** for our system ((18) in [40]) when there is no bias in measurements and the standard deviation is constant.

$$\mathrm{PEB}(\sigma, \theta_{u_i}) = \sqrt{\frac{3\sigma^2}{\left(\sum_{i=1}^{3} \cos^2(\theta_{u_i})\right)\left(\sum_{i=1}^{3} \sin^2(\theta_{u_i})\right) - \left(\sum_{i=1}^{3} \cos(\theta_{u_i}) \sin(\theta_{u_i})\right)^2}} \quad \textbf{(3.17)}$$

where σ is the standard deviation and θ_{u_i} is the bearing angle from each UWB module on G to A. Here, since the only varying parameters are the bearing angles of UWB modules, as the distance increases, differences in bearing angles decreases (assuming the bearing angle to the center of G does not change) and the denominator inside of the square-root operation decreases hence PEB value increases. In Figure 3.4 an example PEB model is presented for a robot at $[\mathrm{x}, \mathrm{y}] = [0, 0]$ and UWB sensors placed as in **(3.5)** with $\mathrm{m} = 0.5\,\mathrm{meter}$ for $\mathrm{r}_{1,2,3}$, with $\sigma = 0.15$ meters. No filters are used for noise suppression during the computations of this figure.

For better visualization, Figure 3.5 shows the localization of A for 4900 samples in a setup where G is placed at the origin and A is placed at $[\mathrm{x}, \mathrm{y}, \mathrm{z}] = [10, 10, 7]$ and $\sigma = 0.15$. Additionally, for Figure 3.5, a Gaussian weighted moving average filter with length 12 is used to further smooth generated noisy data.

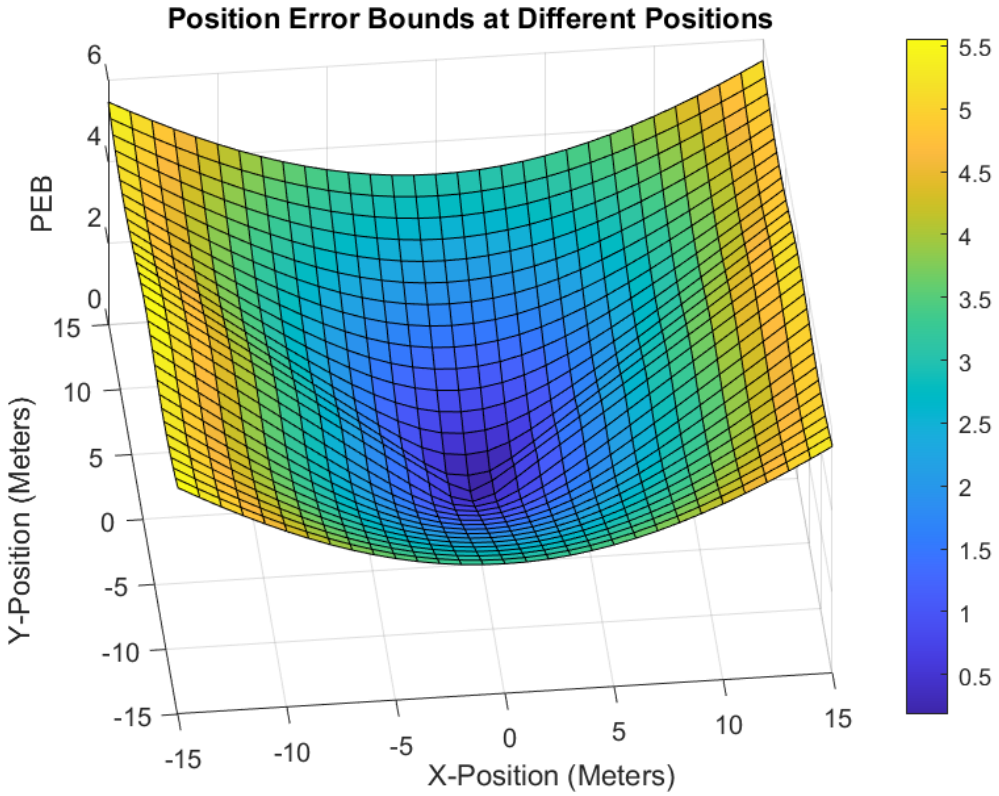

Figure 3.4 Position error bounds at different locations around the ground robot. σ = 0.15.

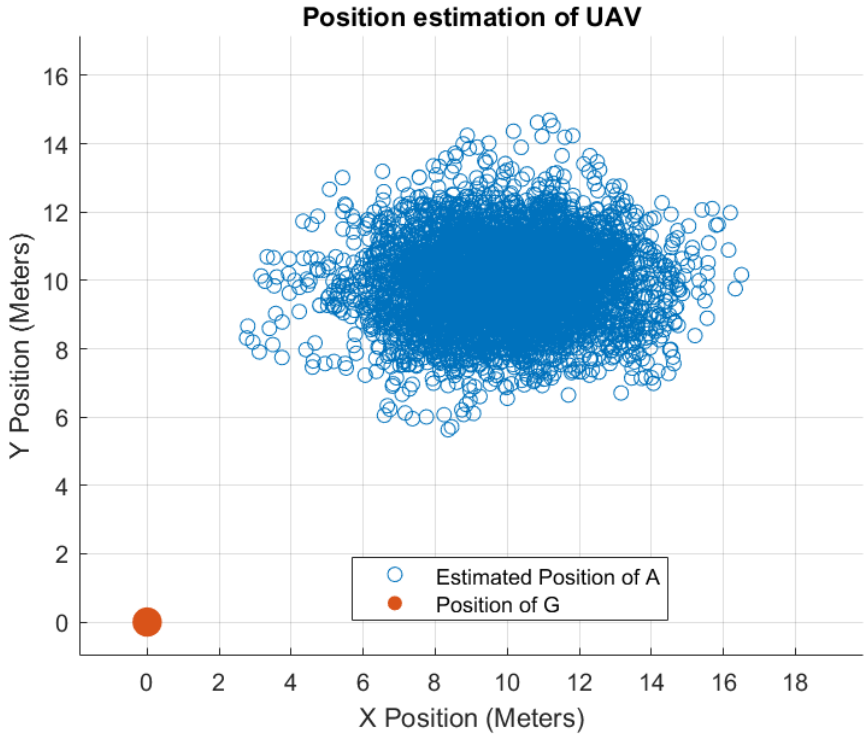

Figure 3.5 Scatter plot of position estimation for UAV from generated 4900 different noisy range measurements with σ = 0.15.

Finally, from Figure 3.5, the amount of noise might be seen high however, it should be noted that most of the points are close to the actual position of the UAV and since these samples are collected in order, for a 10Hz updating system, 4900 samples would take 490 seconds or 8 minutes. Hence, with a good noise filtering, the localization system can work successfully. Additionally, increasing distances between UWB modules on G (while preserving the same triangle shape) increases the performance of the system since differences in bearing angles of UWB modules to robot A increases this way. The effect of increasing distances between UWB nodes can be seen in Figure 3.6.

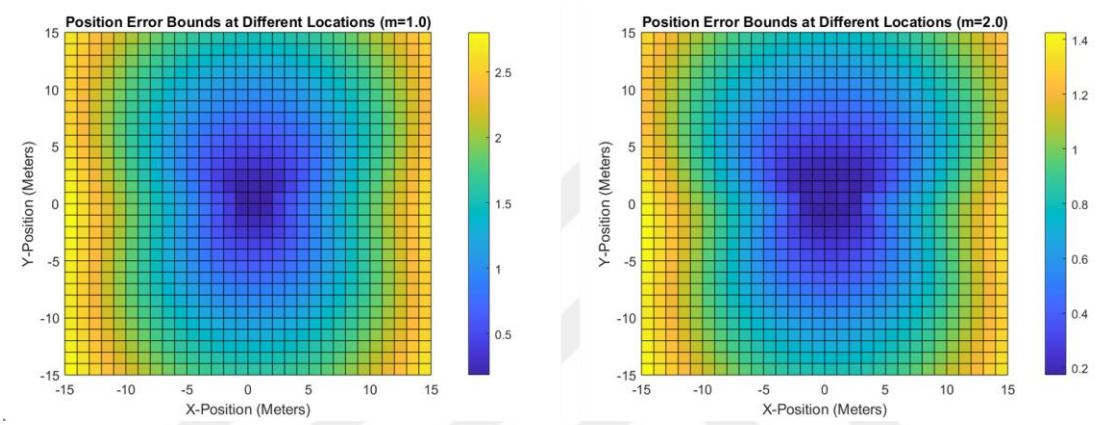

Figure 3.6 Position error bounds for different onboard UWB distances on robot G. (Left: m=1, right m=2)

However, it should not be forgotten that while increasing distances between UWB modules, symmetry of the robot G should be preserved otherwise modification of VFH+ algorithm is required for obstacle avoidance.

To calculate the bounds for the bearing angle error, we analyze the PEB in the neighborhood of the desired drone location. Consider Figure 3.6 where the anchors are located at the positions $\mathbf{r_1} = [0.5 \quad 0]^{\mathrm{T}}$, $\mathbf{r_2} = [-0.5 \quad 0]^{\mathrm{T}}$, $\mathbf{r_3} = [0 \quad 0.5]^{\mathrm{T}}$, and the standard deviation of the UWB distance measurement noises are chosen $\sigma = 0.1$ m. The three black circles centered at $\mathbf{c_1} = [0 \quad 2]^{\mathrm{T}}, \mathbf{c_2} = [0 \quad 5]^{\mathrm{T}}, \mathbf{c_3} = [0 \quad 8]^{\mathrm{T}}$ m, with radius being the PEB for these center locations denote the maximum position errors a least-squares algorithm can result in if robot A is located at $\mathbf{c_i}$. Since the bearing angle between robots G and A are calculated from the estimated location of robot A, the maximum bearing angle errors can be represented by the two black lines drawn on the right and left sides of the circles. We found that this error $e_{\mathrm{bear,max}} = 0.141$ rad is the same for all three locations $\mathbf{c_i}$. Notably, although the PEB increases as the planar distance between

robots G and A increases, the bearing angle error remains bounded, which demonstrates the efficiency of the proposed approach. That is, robot G can utilize the three UWB distances to calculate the bearing angle towards robot A with a bounded error to be implemented in its navigation module.

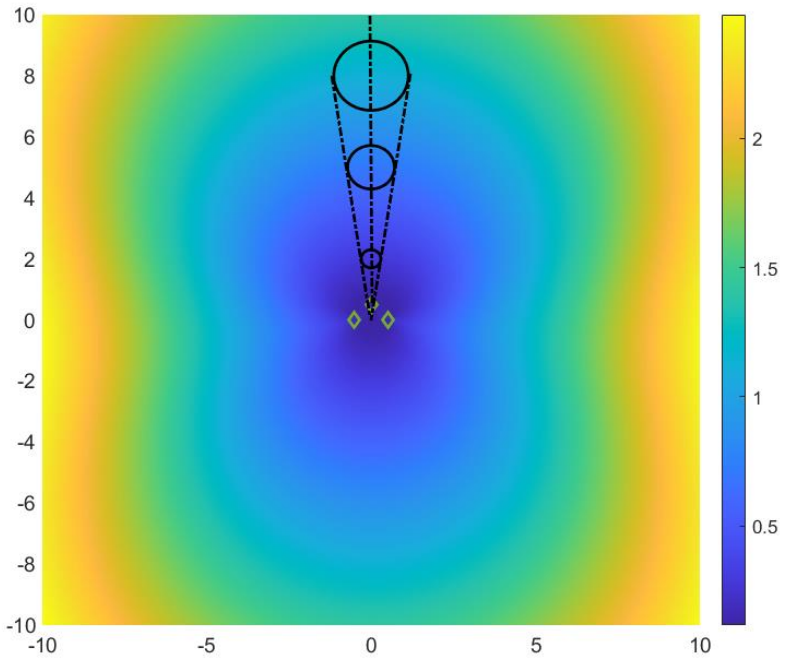

Figure 3.7 Standard deviations in bearing angles at different distances. Three diamonds are UWB locations while circles have the radius of calculated PEB at that location (circle centers).

Chapter 4

Results

The study presented in Chapters 2 and 3 is verified with realistic simulations and laboratory experiments. To evaluate the system, a simulation environment is created with trees, a grass plane, forest fire images, normal forest images in the environment, a wheeled ground robot, and a VTOL quadrotor drone. Furthermore, before running the simulation, other submodules of the overall system like the object detection model, obstacle avoidance system, and communication and ranging system are tested.

As discussed in the previous chapters, the drone used in the simulations should take off, scan the environment by slowly rotating in the yaw direction and detect the presence of a forest fire. This is also the first objective of the system as mentioned in the method Chapter 3. Thus, the object detection algorithm plays a critical role in the system. The trained model must be robust against noises and image distortions. Additionally, model performance should be high enough to not miss any forest fire instances. Moreover, the trained model should continue to detect fire until both robots arrive at the fire location, in other words, the detection system should perform sufficiently from the beginning to the end of the operation. On the other hand, in a practical application, the UAV should have an obstacle avoidance system for take-off and landing. Here, it is assumed that there are no nearby obstacles in the take-off and landing location, and the UAV takes off to an altitude which is higher than any other obstacles around. We remind that this assumption can be easily relaxed in the real-world implementation by choosing a suitable location for the drone base.

To test the second objective mentioned in the previous section, many trees (as obstacles) are put in the environment. The VFH+ algorithm is used for obstacle avoidance observed during simulations to detect any instance of crashes. Additionally, the VFH+ needs a target direction to produce an obstacle-free direction accordingly. This makes UAV localization another important factor for the model performance. During simulations,

Gaussian noise with zero mean and 15cm standard deviation is added to distance measurements to see if localization and target following system acts robustly even with noisy measurements.

The following sections explain object detection model training, simulation environment, and system simulations in detail.

4.1 Model Training

The original YOLOv7 model comes with pre-trained weights. The algorithm is originally trained on the COCO dataset [41] which mainly includes 80 different classes related to traffic signs**,** some objects which can be seen in city traffic**,** some animals (like horses, cats, sheep, etc.) some types of furniture, foods and daily used objects. Unfortunately, pre-trained YOLOv7 models cannot recognize fire or smoke. For this reason, both convolutional layers and decision layers of the YOLOv7 are trained with the fire and smoke images.

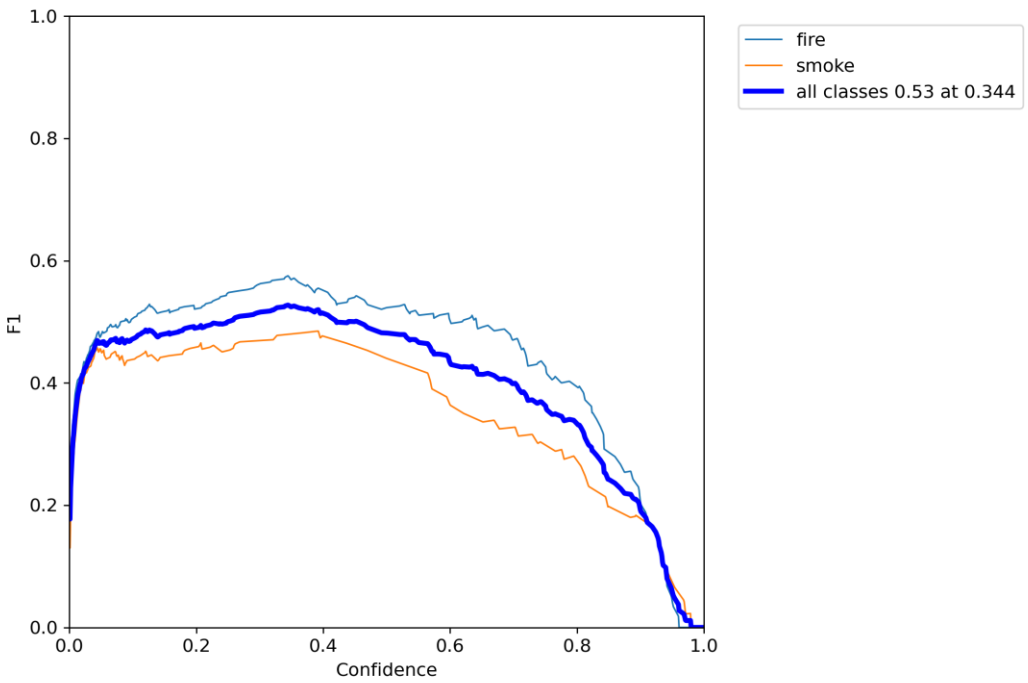

Figure 4.1 F1 score to confidence graph obtained from 200 epoch training of the YOLOv7-tiny model.

Usually, the transfer learning [42] approach is used in the case of new object recognition with a pre-trained network. In transfer learning, especially for convolutional neural networks, if the object(s) to be recognized has similar features to the original train objects,

most of the pre-trained weights of the convolutional layers are kept constant during training. However, weights of the decision and/or deep neural network layers are trained (or fine-tuned) for the new objects to be detected. As a result, the new system makes use of pre-trained feature recognition of the convolutional layer to recognize the new object and the deep layers and/or the decision layer configured to make final judgments for the newly introduced objects.

However, in the case of fire and smoke detection, since the features of these objects vastly differ from the original train objects in terms of human perception, both convolutional and deep layers have been re-trained. Figure 4.1 shows the F_1 score [43] of the trained object detection model. Basically, the F_1 score indicates the model performance in terms of recall and precision. It is calculated with the following formula given in **(4.1)**.

$$F_1 = 2\frac{Precision \times Recall}{Precision + Recall} = \frac{2 \times TP}{2 \times TP + FP + FN} \quad \textbf{(4.1)}$$

where TP is the true-positive prediction ratio, FP is the false-positive prediction ratio and FN is the false-negative prediction ratio of the model.

For fire and smoke detection, the YOLOv7-tiny model structure is used. The dataset used for model training is created by collecting many publicly available forest fire images on the internet. Labeling of the images is handled by using online tools (i.e., makesense.ai). The model was trained with the parameters listed in Table 4.1 using a 2 NVIDIA Tesla T4 GPU in parallel. Train results and some of the test images can be found in Figure 4.2 and Figure 4.3 respectively. The performance of the model was lower than the expected; however, most of the inaccuracies were caused by incorrect sizing of boxes or insufficient detection of all occurrences of objects in the images.

Table 4.1 Hyperparameters that are used for the model training.

Batch Size	Epoch	Learning Rate	Flip-LR (0-1)	Mosaic (0-1)	Image Size
16+16	200	0.01	0.5	1.0	640x640

Both fire and smoke have unstructured, undefinable shapes. The performance of the model for detecting single occurrences was observed as sufficient, however, multiple occurrences or defining the bounds of a big fire or smoke is a hard task even for complex CNN models. Considering these factors, model performance in the simulation was adequate for detecting and bound-boxing fires or smokes for the task of target detection

and reaching. Additionally, multiple sampling of scenes from different perspectives helped reduce false detections and moving toward falsely detected places.

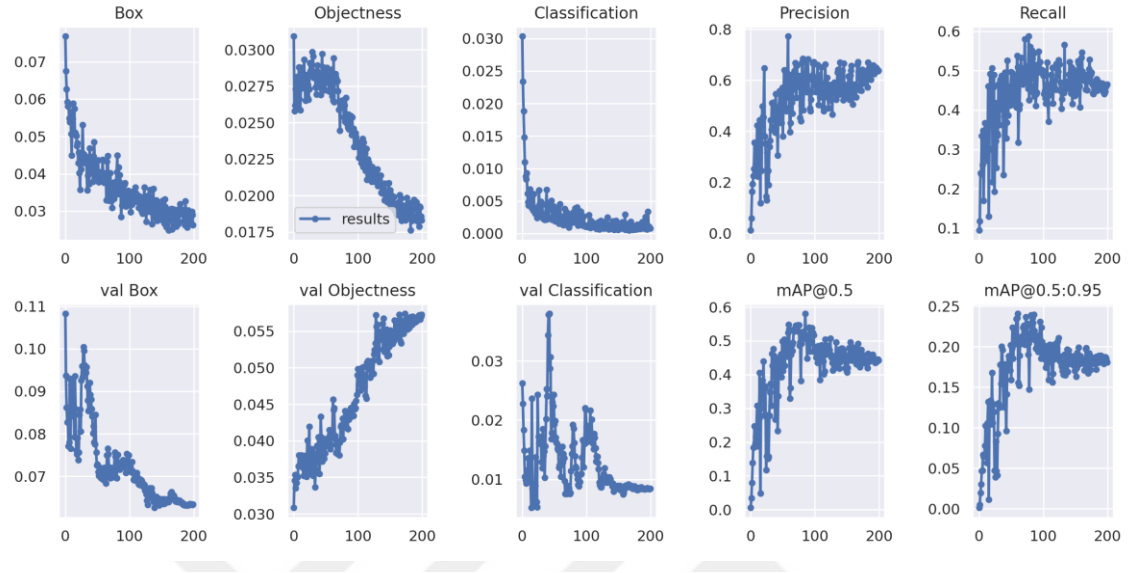

Figure 4.2 Results obtained during training process. Best results obtained at epoch 90s.

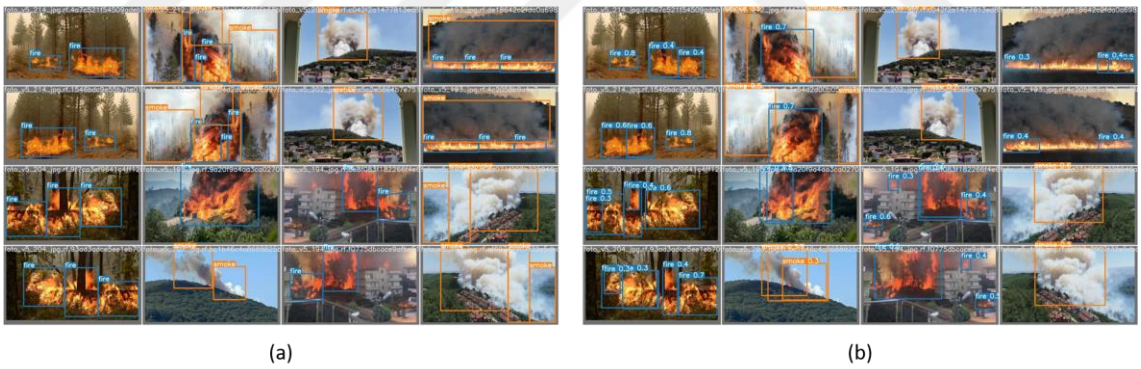

Figure 4.3 Comparison between true labeling and model predictions. (a) True labels, (b) model predictions.

4.2 Simulation Results

The performance of the proposed method is validated with computer simulations. A simulation environment is created with a realistic simulation program called GAZEBO. Version 9 of the program is used during experiments. The two robots are placed in the created simulation environment where there are multiple trees and a forest fire image for the drone to detect. Figure 4.4 shows the top view of the simulation world. As a model

ground vehicle, the Husky robot is used. The original model is modified and 3 UWB imitating boxes are placed on the robot with $m = 0.5$ meters for $\mathbf{r_1}, \mathbf{r_2}, \mathbf{r_3}$.

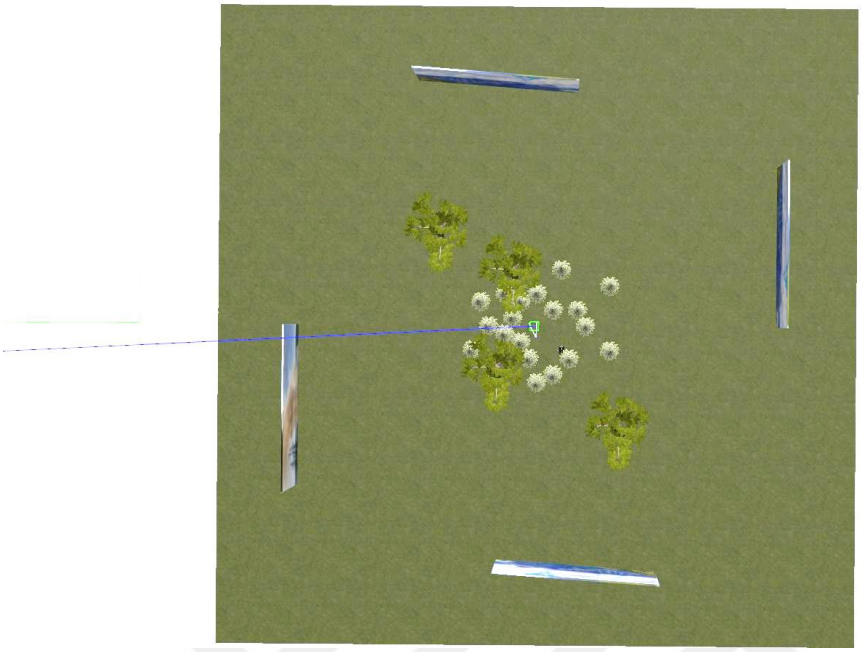

Figure 4.4 Simulation environment view from the top. Trees and robots in the middle, images with fire or no-fire around.

The drone model used is the iris drone of the PX4-Autopilot system [44]. Pixhawk software-in-the-loop (SITL) system is used along with the Mavlink communication line. Both vehicles are controlled and commanded over ROS-Melodic middleware running on UBUNTU version 18.2 LTS. During the simulation, UWB ranges are computed according to ground truth obtained from the rostopic "/gazebo/model_states". Zero-mean random Gaussian noise with a standard deviation of 15 cm is added to the range measurements between the drone center and ground vehicle UWB simulating boxes at every measurement. The results are published with the rostopic "/uwb" at 10Hz and both vehicles received the results online by subscribing to this topic.

Relative localization, local planner, and control/command algorithms are implemented on Python version 2.17 since the ROS-Melodic is supporting this version of the Python programming language. However, YOLOv7 is implemented for Python version 3.6 or more. To make both control and detection systems compatible with each other, "rospkg" is installed for Python version 3.7, and some functions of OpenCV-ROS are re-implemented for Python version 3.7. This way programs on both python versions make use of generated topics and they can publish to the required topics.

Figure 4.5 UAV taking off to a predefined altitude and scanning environment. Left (a) take off, right (b) scanning. Red circle: UAV front-looking camera footage; blue circle: UAV; yellow circle: UGV.

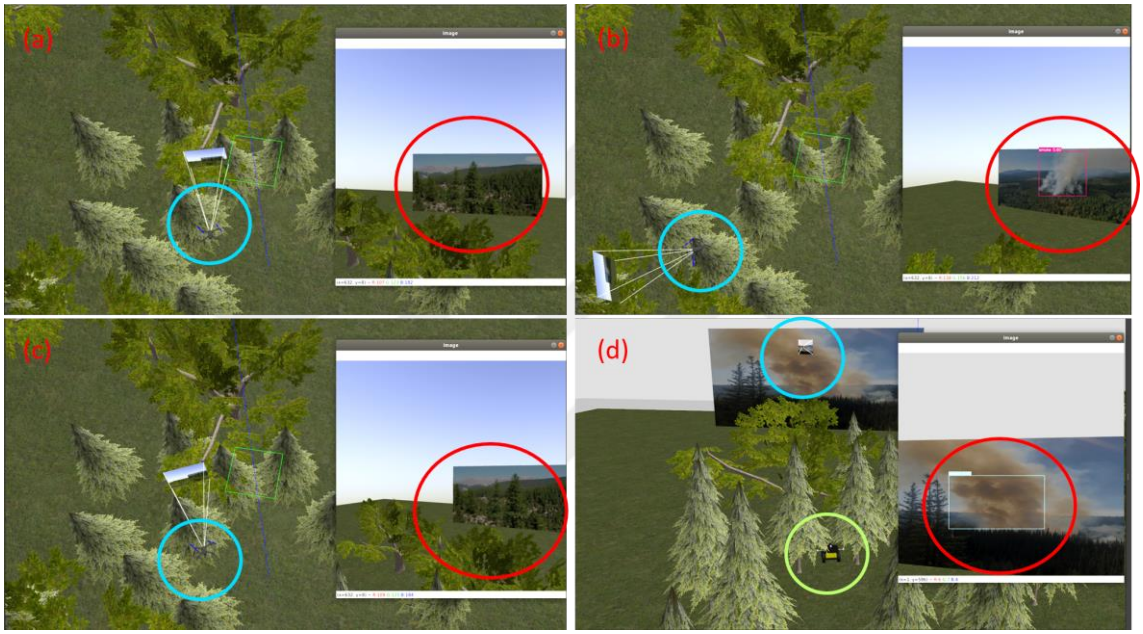

Figure 4.6 Different steps during simulations. (a) UAV scanning environment, it is seeing a no-fire image. (b) UAV moving towards detected smoke. (c) UAV rotating to scan the environment. (d) UAV getting close to the smoke region while UGV following UAV.

In Figure 4.5 the drone took off to 7.5 meters altitude to scan and detect its environment. A forest fire image is placed in an unknown location. As the drone rotates about its z-axis, its camera sees the image and the trained YOLOv7 model detects both smoke and fire. In Figure 4.6, rotation of the drone during the scan, smoke detection, and target reaching of UAV, and UAV following operation of UGV can be seen. During the movement of both vehicles, if the distance between UAV and UGV increases to a limit, UAV waits for the UGV to come closer. Such behavior is required in a practical application since practical radio frequency sensors have limited range and in general ranging quality of UWB modules decreases with increasing distance. During the

simulations, images used were selected from the ones that are not used in the training dataset.

In Figure 4.7, the plot of the x-y axis movements of both ground and air robots can be seen. As it can be seen from Figure 4.7a and Figure 4.7b, the ground robot only moves when the distance between the drone and the UGV is bigger than a threshold. This ensures higher accuracy localization and bearing computation of the drone. On the other hand, the drone waits for UGV to get closer if the distance between them exceeds a threshold (Figure 4.7c). To show that, in simulation 2, the UGV started lately after the UAV took off and started to approach the target location. It is clear from Figure 4.7c that UAV waits for the ground robot to close the gap between them. This is necessary for a real-life application since communication signals and the link can be corrupted by environmental sources and keeping the distance low enough ensures high-quality communication between nodes. From the paths-plot created from UGV and UAV movements, it can be clearly seen that the proposed system can detect fire and approach the fire location without colliding with trees or other obstacles.

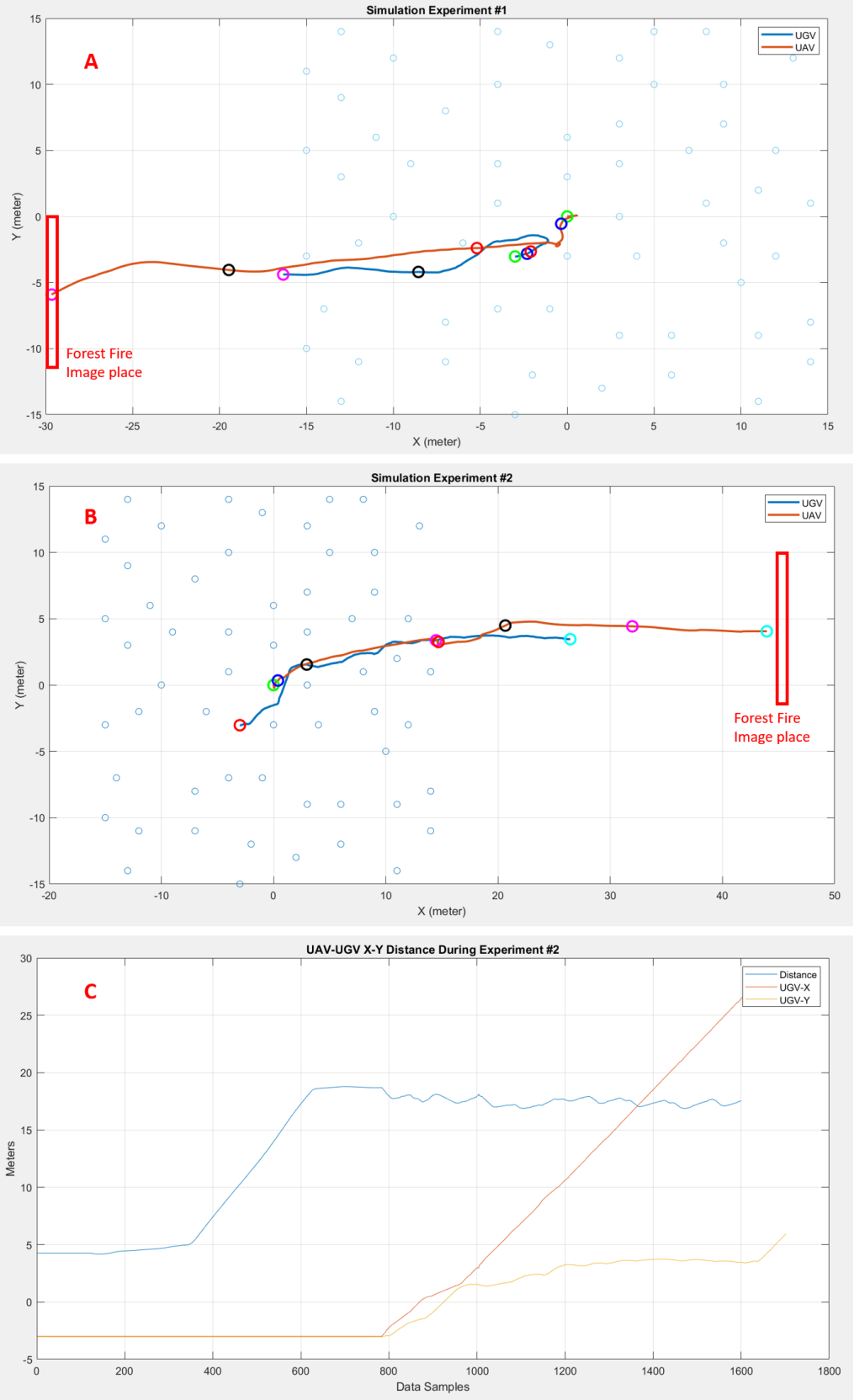

Figure 4.7 Path created during two different simulation experiments. Blue lines UGV, orange lines UAV, small blue circles obstacles (trees). (A) simulation 1, (B) simulation 2, (C) distance change during simulation 2.

4.3 Experiments

To demonstrate and test the performance of the designed system, two different experiments are conducted. For the first experiment, in a motion capture room, the performance of the UAV following system of the UGV is tested. As a ground robot, ROSbot 2 is used. The robot has a Rockchip RK3288 (Quad-core ARM Cortex-A17 32-bit) processor running at 1.8GHz clock frequency, ARM Mali-T764 MP2 type GPU and 2GB of random-access memory. Additionally, the robot has accelerometer and gyro sensors for inertial measurements. On top of the robot, 3 UWB sensors are placed by using 35 cm long plastic rods and connected to the robot's USB port for powering the modules and communication. However, no LIDAR system was installed on the robot since the only purpose of this experimental setup is to check if UAV following system works in a laboratory experiment. Additionally, UBUNTU version 18.2 LTS along with ROS-Melodic is used for ground robot control and commanding. The commanded angular and linear velocities over ROS-Melodic middleware are turned into actuations by using the drivers provided by the robot producer company (HUSARION). Incoming UWB measurements are collected from the USB port connected to one of the UWB modules. As a UAV system, DJI Mavic Pro is used. A power source and a UWB sensor are placed on top of the drone. UWB modules used on both UGV and UAV is the Decawave DWM1001 UWB module. The designed UWB communication system communicates at 20Hz. In this experiment, the drone is controlled using its remote controller.

In Figure 4.8 images of the running system can be seen. It is clear from the snapshots that the ground robot reacts to changing UAV position and by following the bearing angle direction to the UAV, it follows the aerial robot. Additionally, since the UWB on the UAV is placed on top of the drone, a non-line-of-sight situation exists during the experiment. This shows that the designed system can generate correct bearing angles under NLOS conditions where communication is still possible. Position information collected from the conducted two repeats of the first type experiment is plotted using MATLAB in Figure 4.9. On the plot, some of the position data that are collected at the same time are marked for ease of tracking the followed paths.

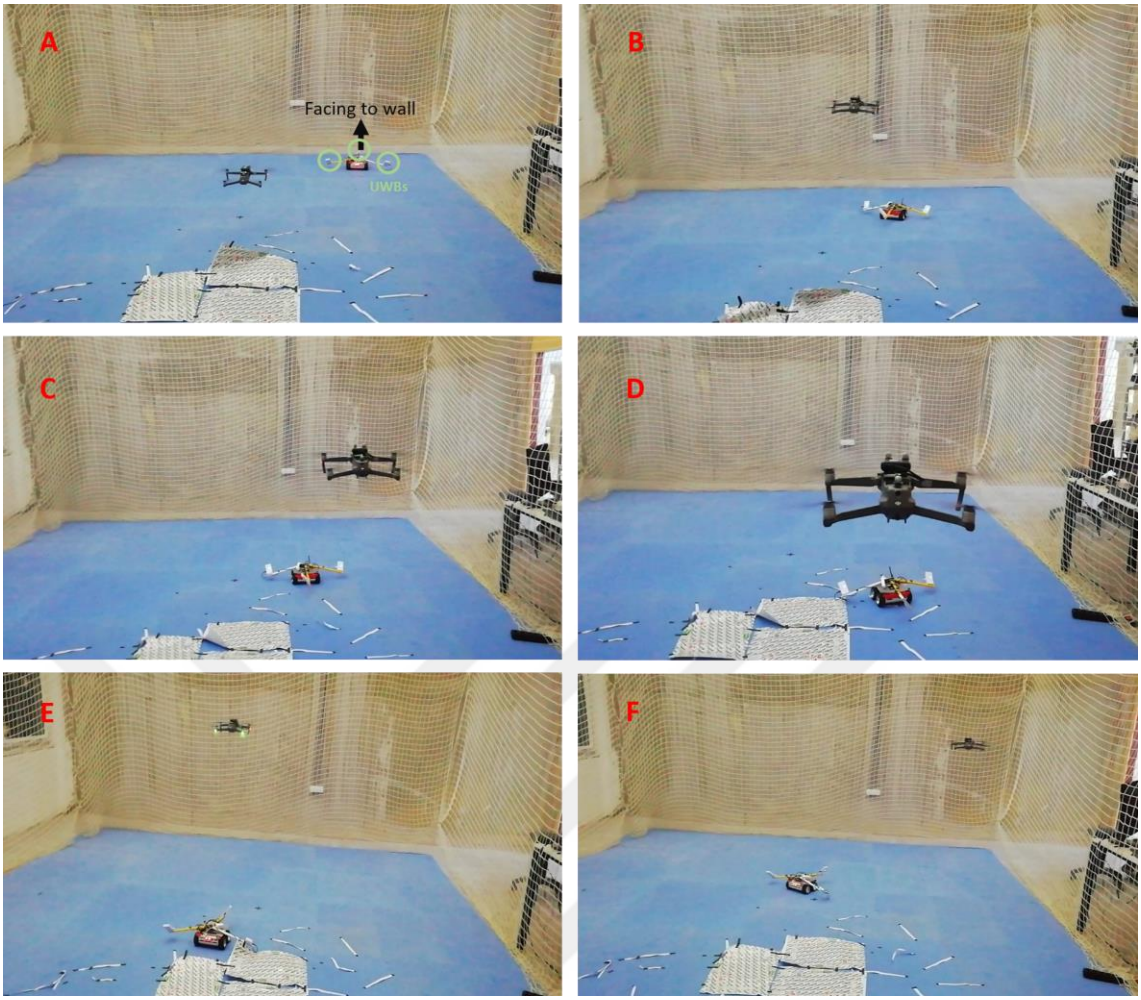

Figure 4.8 Experiment snapshots (A-F in order).

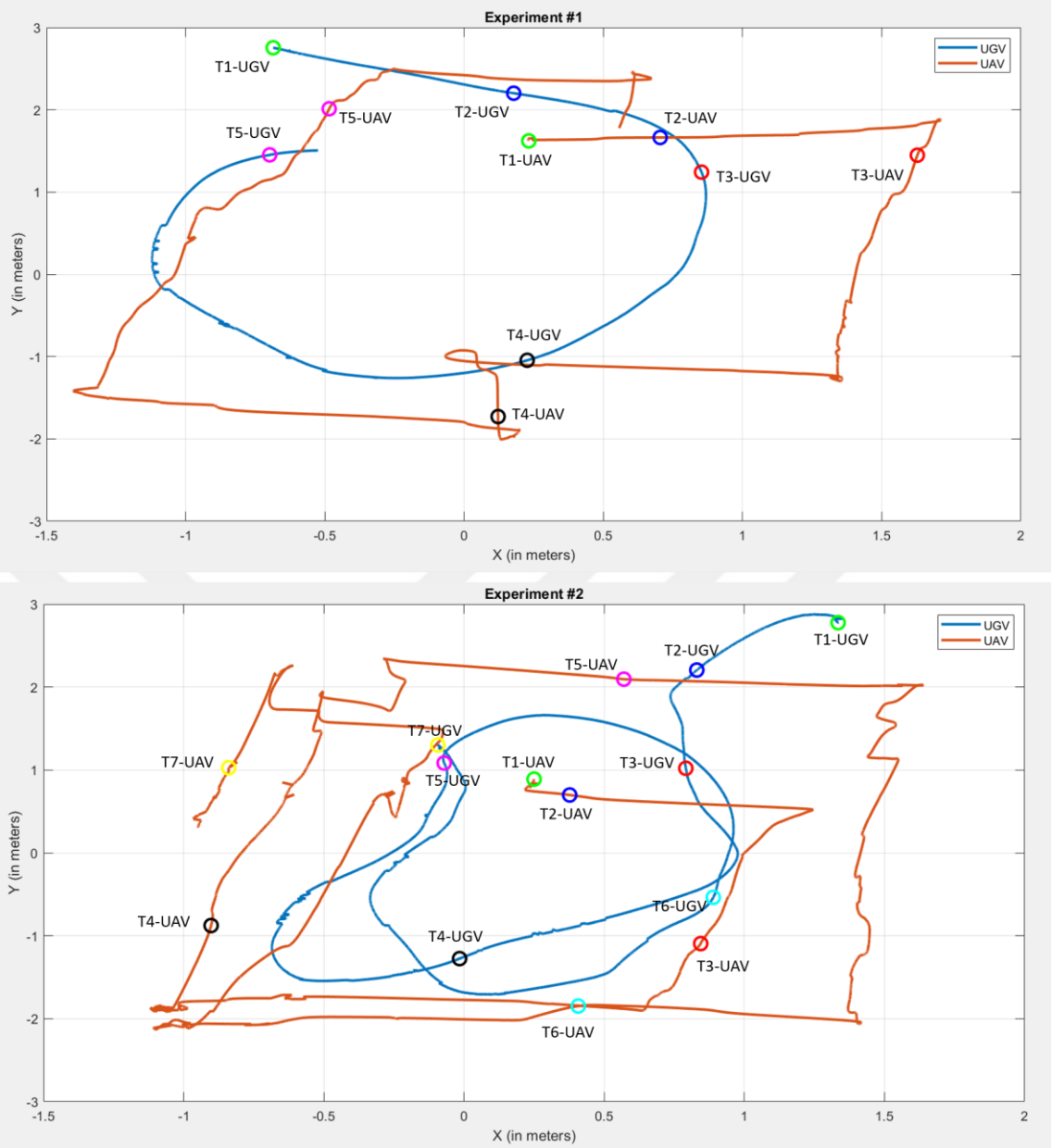

Figure 4.9 The path created by UGV (blue) and UAV (orange) during two different experiments. Times for UAV and UGV are marked with circles and named Tx-UAV for UAV and Tx-UGV for the ground robot for better tracking the positions of the two robots.

For the second experiment, in a corridor environment, UAV following performance and obstacle avoidance system performance is tested together. Different from the previous experiment, as UGV, Rover Zero 3 robot with 4-wheel configuration is used. This robot uses an Intel NUCi3 computer to control its actuators. Same operating system and same ROS version are used for the computer. However, since actuators, actuator sizes and the robot itself is different from the robot in experiment 1, drivers provided by the producer company is used for actuator control.

To place 3 UWB modules, 3 plastic rods with lengths of 55cm each were used, and sensor placement was similar to the previous experiment and simulations except that this time plastic rods are attached to a platform approximately 50cm above the robot top plate. Additionally, a 2D LIDAR sensor is placed on top of the platform installed on the robot to detect obstacles and run the VFH+ algorithm. The LIDAR sensor has 360-degree vision and 1 degree angle increment hence produces an array of 360 range measurements. Update rate of the sensor is 5Hz. The overall robot is present in Figure 4.10. Since this experiment is conducted in a corridor environment and no motion capture system is used, overall performance is evaluated visually in terms of moving towards the UAV and avoiding obstacles during the operation.

For the first part of the second experiment, in a narrow corridor, the UAV is manually controlled and follows a straight path. Here, a person walked in front of the UGV to represent an obstacle. Additionally, the drone kept at hovering position for multiple times to check if the ground robot can protect the predefined minimum distance between the drone and itself. Snapshots of the experiment can be found in Figure 4.11. All the inspections for following, obstacle avoidance, and distance protection were successful.

The second part is conducted in a wider corridor where 3 different boxes are placed on the path of the UGV, and a drone again manually controlled following a straight path in the middle of the corridor. Both experiments are concluded without any collision and the UGV followed the drone successfully. Snapshots of the second part can be found in Figure 4.12.

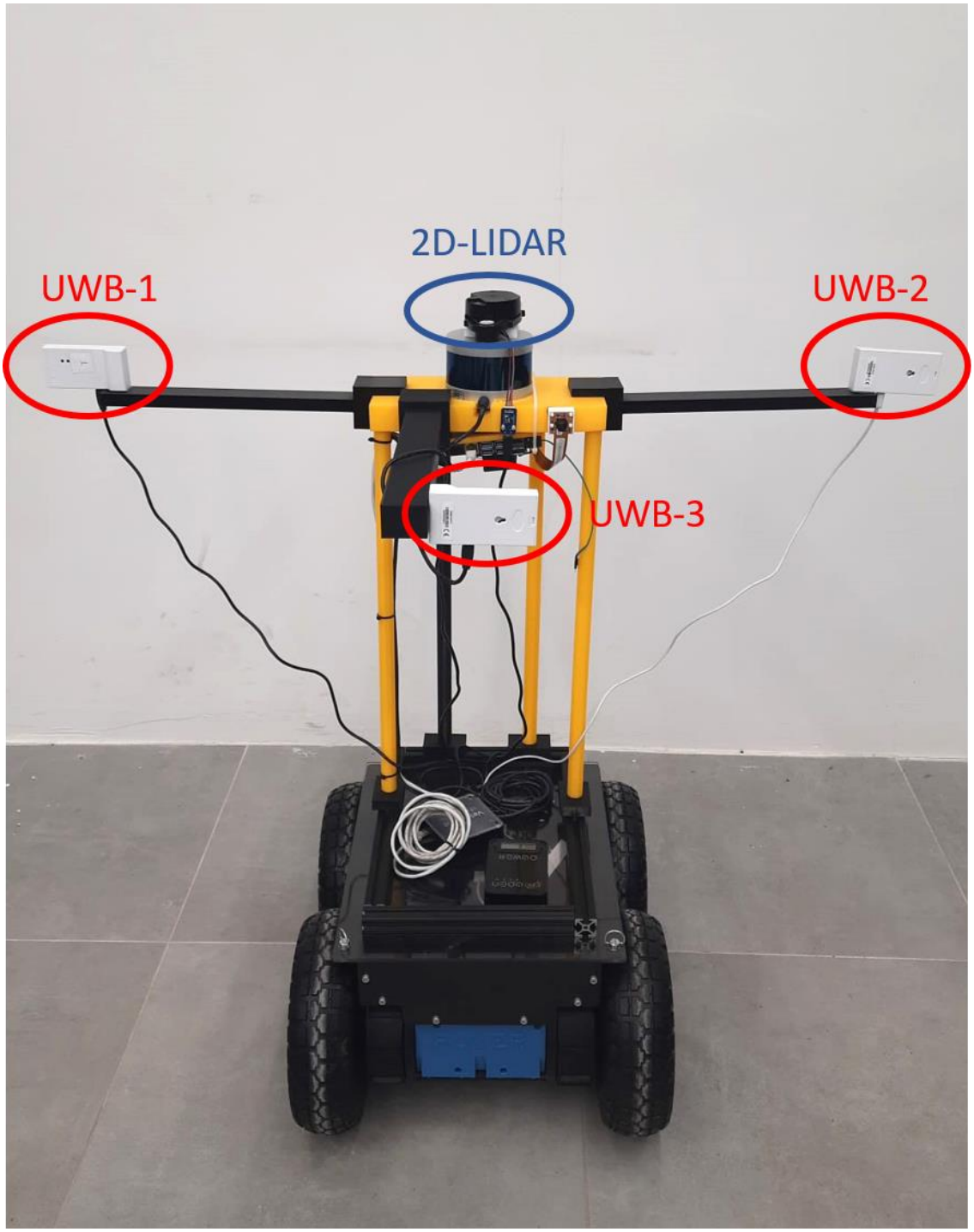

Figure 4.10 Ground robot setup in experiment 2.

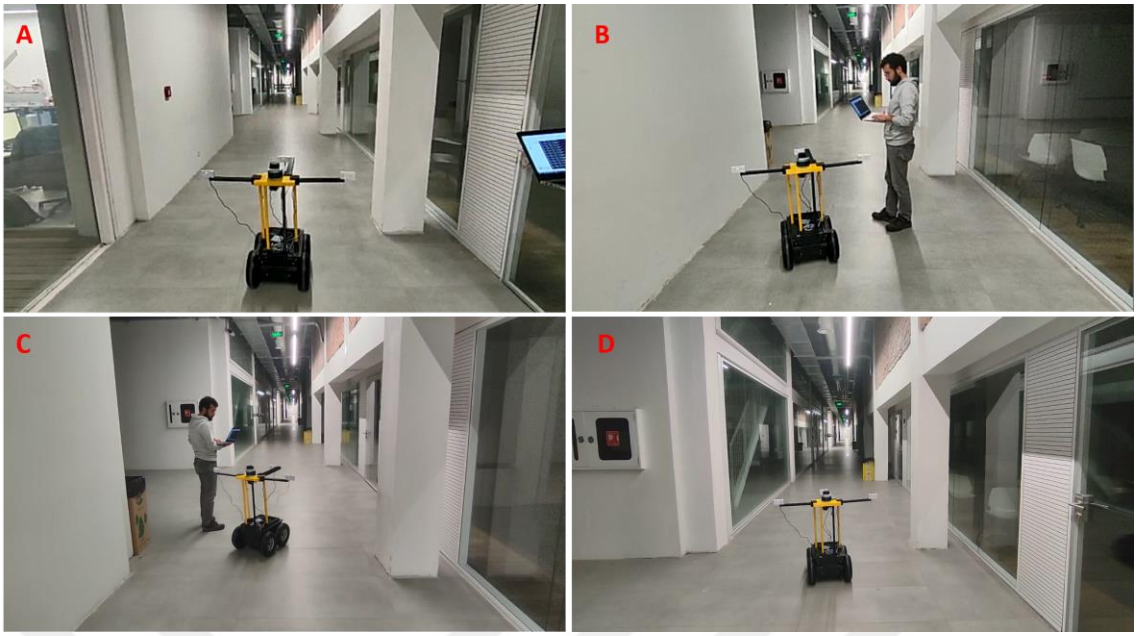

Figure 4.11 Part 1 of experiment 2. (A) UGV following UAV. (B-C) obstacle avoidance. (D) UGV following UAV.

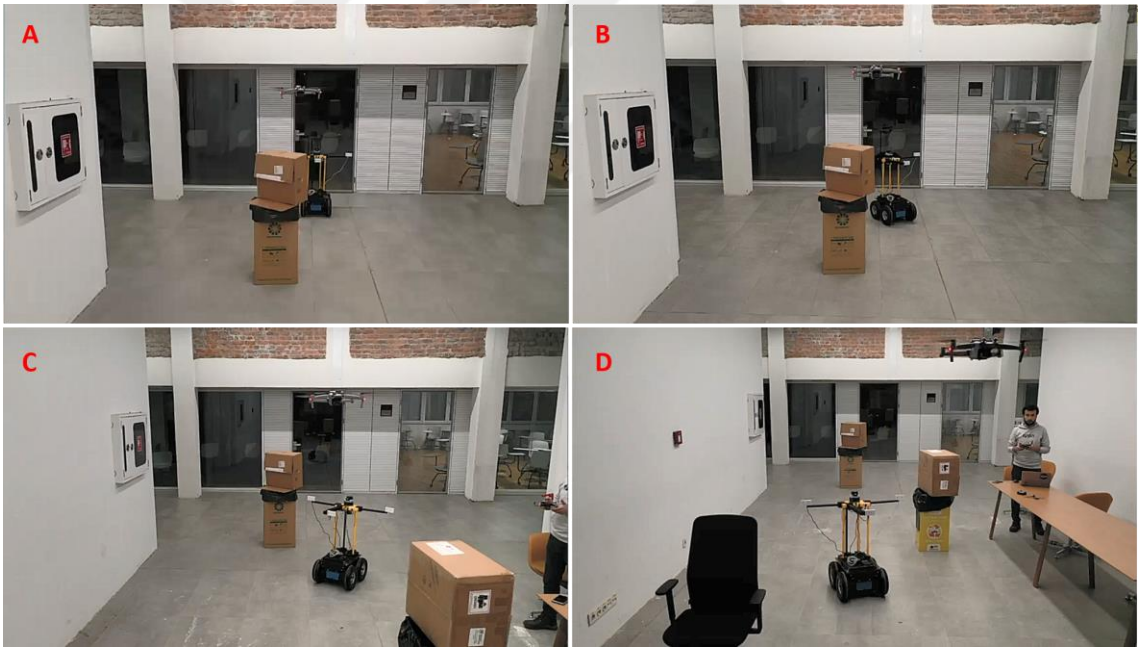

Figure 4.12 Part 2 of experiment 2. (A, B, C, D) UGV maneuvering to avoid from different obstacles while following UAV.

Chapter 5

Conclusions and Future Prospects

5.1 Conclusions

In this study, a heterogenous robot team with a UAV and UGV is designed. The UAV is responsible for forest fire detection and ground robot guidance while the ground vehicle is responsible for following the UAV without any collision with obstacles in a forest environment. Two robots use ultra-wideband communication and ranging sensors to locate themselves for a successful follow-up operation. Simulation results showed that the designed system is robust against ranging errors and can outperform only-odometry-based systems since the UWB-based ranging system has a bounded error level. However, further research must be conducted to prove this proposition. The localization and following system performed with small errors in the simulations, and the trained computer vision model performed quite well.

5.2 Societal Impact and Contribution to Global Sustainability

The effects of global warming are at a level where it can be felt obviously. As the consequences increase day by day, we have no time to lose or spend without considering the results. Unfortunately, forest fires are only one of the consequences and it has a boosting effect on the crises. As the seasons lived on their edges, the summers have become extremely hot. For this reason, the number of forest fires increased rapidly. Hence, an early forest fire detection and response system became mandatory. The presented in this study can be a good start for fighting against forest fires and protecting our diminishing forests.

According to recent studies [9]–[11], a huge amount of CO_2 has been released into the atmosphere in the last three years due to forest fires globally. In fact, the problem here is not only the carbon release. Carbon capture property of the burned forest lands disappears

so that the consequences double. As the number of fires increases, a precaution must be taken to stop forest fires before they become a bigger problem for all creatures. Changing climates harm food supply and agricultural processes. As time passes, we experience this fact more and more. However, forests are one of the most critical environments to protect biodiversity and as a weapon against global warming. The number of studies on protection or forest-supporting research should increase to protect and fortify our forests. It is hard to measure the direct impact of this study on global sustainability since the study is at the basic experimental level yet. However, by using the robust architecture presented in this study, novel robotic systems can be created to be practically used in forests. This way the number of forest fires will decrease rapidly. By protecting our forests, we will be protecting our farmlands, clean water resources, forest-related industries, and most importantly the ecosystem it has.

As we do not have any extra time for taking precautions against global warming, we also do not have any plants or trees to lose. As engineers, we are great at automating machines and finding new solutions. I believe, enough attraction to this topic will produce a good solution for sure.

5.3 Future Prospects

This thesis study examines a heterogeneous multi-robot system for early forest fire detection. Proposed methods and the overall system are shown to be applicable to implementation in real by presenting successful simulations and basic experimental results. However, even the most realistic simulations cannot model noises or any other uncertainties in a practical application. For this reason, real implementation of the system and field experiments require a full test of the system's performance and its robustness under uncertainties.

Besides further experimental setups, new aspects can be added to the system. An example can be using a second UAV or an additional downward-looking camera on UAV to observe the environment and extract obstacle information on behalf of the UGV. This way the ground robot can compute optimal and shorter paths toward the target place.

In the study, UAV obstacle avoidance is not included in the system by assuming UAV altitude is high enough so that the UAV does not confront any obstacle. However, this

assumption might not be practical for a real-world application. The assumption can hold if the UAV altitude is higher than any tree, grid, or any other structures in the field of activity. However, this may require an advanced camera and lens systems to record images of far objects. Especially if the main goal is to detect fire or smoke when they are not so big. Additionally, for this assumption to hold, the UAV takes-off and landing area should be cleared of obstacles. To improve this aspect of the study, the UAV to be used should be equipped with suitable obstacle detection sensors or it can even use computer vision algorithms to detect and avoid obstacles. This way, the UAV can adjust its height during the mission to optimize image capturing of fire location. However, the addition of multiple sensors and implementing obstacle avoidance for a 6-degrees-of-freedom vehicle introduces high complexity to the system.

BIBLIOGRAPHY

[1] Walter W. G., "An imitation of life," *Sci Am*, vol. 182, no. 5, pp. 42–45, May 1950.

[2] K. Kaltsoukalas, S. Makris, and G. Chryssolouris, "On generating the motion of industrial robot manipulators," *Robot Comput Integr Manuf*, vol. 32, pp. 65–71, Apr. 2015.

[3] R. Sparrow and M. Howard, "Robots in agriculture: Prospects, impacts, ethics, and policy," *Precis Agric*, vol. 22, no. 3, pp. 818–833, Oct. 2020.

[4] Y. Khan Arshia and Anwar, "Robots in Healthcare: A Survey," in *Advances in Computer Vision*, 2020, pp. 280–292.

[5] A. and N. E. and S. S. Marshall Joshua A. and Bonchis, "Robotics in Mining," in *Springer Handbook of Robotics*, O. Siciliano Bruno and Khatib, Ed. Cham: Springer International Publishing, 2016, pp. 1549–1576. doi: 10.1007/978-3-319-32552-1_59.

[6] F. Matsuno and S. Tadokoro, "Rescue Robots and Systems in Japan," in *2004 IEEE International Conference on Robotics and Biomimetics*, 2004, pp. 12–20. doi: 10.1109/ROBIO.2004.1521744.

[7] R. R. Murphy, J. Kravitz, S. L. Stover, and R. Shoureshi, "Mobile robots in mine rescue and recovery," *IEEE Robot Autom Mag*, vol. 16, no. 2, pp. 91–103, 2009, doi: 10.1109/MRA.2009.932521.

[8] A. AghaKouchak, L. Cheng, O. Mazdiyasni, and A. Farahmand, "Global warming and changes in risk of concurrent climate extremes: Insights from the 2014 California drought," *Geophys Res Lett*, vol. 41, no. 24, pp. 8847–8852, 2014, doi: https://doi.org/10.1002/2014GL062308.

[9] J. G. Canadell *et al.*, "Multi-decadal increase of forest burned area in Australia is linked to climate change," *Nat Commun*, vol. 12, no. 1, p. 6921, 2021, doi: 10.1038/s41467-021-27225-4.

[10] J. Paudel, "Environmental Disasters and Property Values: Evidence from Nepal's Forest Fires," *Land Econ*, vol. 98, pp. 110519-0159R2, Jan. 2021, doi: 10.3368/le.98.1.110519-0159R2.

[11] M. Alizadeh, J. Abatzoglou, C. Luce, J. Adamowski, A. Farid, and M. Sadegh, "Warming enabled upslope advance in western US forest fires," *Proceedings of the National Academy of Sciences*, vol. 118, p. e2009717118, Jan. 2021, doi: 10.1073/pnas.2009717118.

[12] K. Abnett, "This is how much carbon wildfires have emitted this year," *World Economic Forum*, Dec. 10, 2021. https://www.weforum.org/agenda/2021/12/siberia-america-wildfires-emissions-records-2021/ (accessed Nov. 03, 2022).

[13] K. Mehta, S. Sharma, and D. Mishra, "Internet-of-Things Enabled Forest Fire Detection System," in *2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 2021, pp. 20–23. doi: 10.1109/I-SMAC52330.2021.9640900.

[14] P. and C. N. Dubey Vinay and Kumar, "Forest Fire Detection System Using IoT and Artificial Neural Network," in *International Conference on Innovative Computing and Communications*, 2019, pp. 323–337.

[15] A. Tehseen, N. A. Zafar, T. Ali, F. Jameel, and E. H. Alkhammash, "Formal Modeling of IoT and Drone-Based Forest Fire Detection and Counteraction

System," *Electronics (Basel)*, vol. 11, no. 1, 2022, doi: 10.3390/electronics11010128.
[16] A. Divya, T. Kavithanjali, and P. Dharshini, "IoT Enabled Forest Fire Detection and Early Warning System," in *2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN)*, 2019, pp. 1–5. doi: 10.1109/ICSCAN.2019.8878808.
[17] W. Benzekri, A. el Moussati, O. Moussaoui, and M. Berrajaa, "Early Forest Fire Detection System using Wireless Sensor Network and Deep Learning," *International Journal of Advanced Computer Science and Applications*, vol. 11, p. 496, Jan. 2020, doi: 10.14569/IJACSA.2020.0110564.
[18] T. Miriyala, R. Karthik, J. Mahitha, and V. Reddy, "IoT based forest fire detection system," *International Journal of Engineering & Technology*, vol. 7, p. 124, Jan. 2018, doi: 10.14419/ijet.v7i2.7.10277.
[19] J. Lloret, M. Garcia, D. Bri, and S. Sendra, "A Wireless Sensor Network Deployment for Rural and Forest Fire Detection and Verification," *Sensors*, vol. 9, no. 11, pp. 8722–8747, 2009, doi: 10.3390/s91108722.
[20] Z. Ateeq and M. Momani, "Wireless Sensor Networks using image processing for fire detection," in *2020 5th International Conference on Innovative Technologies in Intelligent Systems and Industrial Applications (CITISIA)*, 2020, pp. 1–10. doi: 10.1109/CITISIA50690.2020.9371798.
[21] J. Fernández-Berni, R. Carmona-Galán, J. F. Martinez-Carmona, and Á. Rodríguez-Vázquez, "Early forest fire detection by vision-enabled wireless sensor networks," *Int J Wildland Fire*, vol. 21, pp. 938–949, 2012.
[22] Y. Sahin and T. Ince, "Early Forest Fire Detection Using Radio-Acoustic Sounding System," *Sensors (Basel)*, vol. 9, pp. 1485–1498, Jan. 2009, doi: 10.3390/s90301485.
[23] W. Krüll, R. Tobera, I. Willms, H. Essen, and N. von Wahl, "Early Forest Fire Detection and Verification using Optical Smoke, Gas and Microwave Sensors," *Procedia Eng*, vol. 45, pp. 584–594, 2012, doi: https://doi.org/10.1016/j.proeng.2012.08.208.
[24] K. A. Ghamry, M. A. Kamel, and Y. Zhang, "Cooperative forest monitoring and fire detection using a team of UAVs-UGVs," in *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2016, pp. 1206–1211. doi: 10.1109/ICUAS.2016.7502585.
[25] Y. Wei, H. Qiu, Y. Liu, J. Du, and M.-O. Pun, "Unmanned aerial vehicle (UAV)-assisted unmanned ground vehicle (UGV) systems design, implementation and optimization," in *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*, 2017, pp. 2797–2801. doi: 10.1109/CompComm.2017.8323042.
[26] J. H. Kim, J.-W. Kwon, and J. Seo, "Multi-UAV-based stereo vision system without GPS for ground obstacle mapping to assist path planning of UGV," *Electron Lett*, vol. 50, no. 20, pp. 1431–1432, 2014, doi: https://doi.org/10.1049/el.2014.2227.
[27] J. P. Queralta, L. Qingqing, F. Schiano, and T. Westerlund, "VIO-UWB-Based Collaborative Localization and Dense Scene Reconstruction within Heterogeneous Multi-Robot Systems," *CoRR*, vol. abs/2011.00830, 2020, [Online]. Available: https://arxiv.org/abs/2011.00830
[28] A. Lakas, B. Belkhouche, O. Benkraouda, A. Shuaib, and H. J. Alasmawi, "A Framework for a Cooperative UAV-UGV System for Path Discovery and

Planning," in *2018 International Conference on Innovations in Information Technology (IIT)*, 2018, pp. 42–46. doi: 10.1109/INNOVATIONS.2018.8606028.
[29] P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968, doi: 10.1109/TSSC.1968.300136.
[30] J. Li, G. Deng, C. Luo, Q. Lin, Q. Yan, and Z. Ming, "A Hybrid Path Planning Method in Unmanned Air/Ground Vehicle (UAV/UGV) Cooperative Systems," *IEEE Trans Veh Technol*, vol. 65, no. 12, pp. 9585–9596, 2016, doi: 10.1109/TVT.2016.2623666.
[31] C. Chen, Y. Wan, B. Li, C. Wang, G. Xie, and H. Jiang, "Motion Planning for Heterogeneous Unmanned Systems under Partial Observation from UAV," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 1474–1479. doi: 10.1109/IROS45743.2020.9341326.
[32] O. Souissi, R. Benatitallah, D. Duvivier, A. Artiba, N. Belanger, and P. Feyzeau, "Path planning: A 2013 survey," in *Proceedings of 2013 International Conference on Industrial Engineering and Systems Management (IESM)*, 2013, pp. 1–8.
[33] A. Stentz and M. Hebert, "A complete navigation system for goal acquisition in unknown environments," in *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, 1995, vol. 1, pp. 425–432 vol.1. doi: 10.1109/IROS.1995.525831.
[34] S. Koenig and M. Likhachev, "D*lite," in *Eighteenth National Conference on Artificial Intelligence*, 2002, pp. 476–483.
[35] M. Przybylski and B. Putz, "D* Extra Lite: A Dynamic A* With Search–Tree Cutting and Frontier–Gap Repairing," *International Journal of Applied Mathematics and Computer Science*, vol. 27, no. 2, pp. 273–290, 2017, doi: doi:10.1515/amcs-2017-0020.
[36] D. Ferguson and A. Stentz, "Using interpolation to improve path planning: The Field D* algorithm," *J Field Robot*, vol. 23, no. 2, pp. 79–101, 2006, doi: https://doi.org/10.1002/rob.20109.
[37] J. Carsten, D. Ferguson, and A. Stentz, "3D Field D: Improved Path Planning and Replanning in Three Dimensions," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 3381–3386. doi: 10.1109/IROS.2006.282516.
[38] C.-Y. Wang, A. Bochkovskiy, and H. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors." Jan. 2022. doi: 10.48550/arXiv.2207.02696.
[39] I. Ulrich and J. Borenstein, "VFH+: reliable obstacle avoidance for fast mobile robots," in *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, 1998, vol. 2, pp. 1572–1577 vol.2. doi: 10.1109/ROBOT.1998.677362.
[40] D. B. Jourdan, D. Dardari, and M. Z. Win, "Position Error Bound for UWB Localization in Dense Cluttered Environments," in *2006 IEEE International Conference on Communications*, 2006, vol. 8, pp. 3705–3710. doi: 10.1109/ICC.2006.255648.
[41] M. and B. S. and H. J. and P. P. and R. D. and D. P. and Z. C. L. Lin Tsung-Yi and Maire, "Microsoft COCO: Common Objects in Context," in *Computer Vision – ECCV 2014*, 2014, pp. 740–755.

[42] A. Brodzicki, M. Piekarski, D. Kucharski, J. Jaworek-Korjakowska, and M. Gorgon, “Transfer Learning Methods as a New Approach in Computer Vision Tasks with Small Datasets,” *Foundations of Computing and Decision Sciences*, vol. 45, no. 3, pp. 179–193, 2020, doi: doi:10.2478/fcds-2020-0010.

[43] N. Chinchor, “MUC-4 Evaluation Metrics,” in *Proceedings of the 4th Conference on Message Understanding*, 1992, pp. 22–29. doi: 10.3115/1072064.1072067.

[44] L. Meier, D. Honegger, and M. Pollefeys, “PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 6235–6240. doi: 10.1109/ICRA.2015.7140074.

CURRICULUM VITAE

2016 – 2019	B.Sc., Electrical and Electronics Engineering, Abdullah Gül University, Kayseri, TURKEY
2019 – Present	Embedded System Engineer, Valentis Biotech, Kayseri, TURKEY