MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE

NATIONAL AVIATION UNIVERSITY

Faculty of Aeronautics, Electronics and Telecommunications, Department of Aviation Computer-Integrated Complexes

ACCEPT TO PROTECTION
Head of Department
_____ Viktor SINEGLAZOV
“_”_____2023 y.

QUALIFICATION PAPER (EXPLANATORY NOTE)

HIGHER EDUCATION STUDY

“MASTER”

Specialty 151 "Automation and computer-integrated technologies"
Educational and professional program "Information support and engineering of aviation computer systems"

Subject: Drone collision warning system

Performer: student of the group I3-225M Maksym Kobzystyi
Supervisor: Associate professor, Oleg Smirnov

Consultant of the "Environmental Protection" section________ Radomska M.M.
Consultant of the "Occupational safety and health" section________Kazhan.K.I.
Norm control:____________Fylashkin M.K.

KYIV 2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
Факультет аеронавігації, електроніки та телекомунікацій
Кафедра авіаційних комп’ютерно-інтегрованих комплексів

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач випускової кафедри
________Віктор СИНЄГЛАЗОВ
“___”_______________2023 р.

КВАЛІФІКАЦІЙНА РОБОТА
(ПОЯСНЮВАЛЬНА ЗАПИСКА)
ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ

“МАГІСТР”

Спеціальність 151 «Автоматизація та компю’терно-інтегровані технології»
Освітньо-професійна програма «Інформаційне забезпечення та інженерія авіаційних комп’ютерних система»

Тема: Система попередження зіткнень дронів

Виконавець: студент групи ІЗ-225М Кобзистий Максим Макисмович
Керівник: доцент, Смірнов Олег Ігорович

Консультант розділу «Охорона навколишнього середовища»_____ Радомська М.М.Консультант розділу «Охорона праці»__Кажан.К.І.
Нормоконтролер:________Філяшкін М.К.

Київ 2023

NATIONAL AVIATION UNIVERSITY

Faculty of aeronavigation, electronics and telecommunications

Department of Aviation Computer Integrated Complexes

Educational degree: Master

Specialty: 151 "Automation and computer-integrated technologies"

ЗАТВЕРДЖУЮ

Завідувач кафедри

Віктор СИНЄГЛАЗОВ

“ ____ ” __________2023 р.

TASK

For the student's thesis

Kobzystyi Maksym Maksymovich

1. **The topic of the work**: "Drone collision warning system"
2. **Project (work) completion date:** from “_2_” ___10___ 2023. until "_31_"__12____2023.
3. **Initial data for the project (work):** to develop an algorithm that will prevent the collision of drones in formation.
4. **Contents of the explanatory note (list of issues to be developed):** 1. Analysis of the problem of drones being blinded in the air; 2. Analysis of existing methods of avoiding blinding in the air of drones; 3. Algorithm development and efficiency evaluation; 4. Software development.
5. **List of mandatory graphic material:** Mat model. Coordinate system. Block diagram of the algorithm. Simulation results
6. **Calendar plan-schedule**

№ п/п	**Task**	**Execution term**	**Execution mark**
1.	Receiving the task	02.10.2023 – 03.10.2023	
2.	Formation of the goal and main tasks of the research	03.10.2023 – 05.10.2023	
3.	Analysis of existing solutions	07.10.2023 – 15.10.2023	
4.	Theoretical consideration of problem solving	17.10.2023 – 01.11.2023	
5.	Development of an obstacle avoidance system based on Algorithm A*	01.11.2023 – 15.11.2023	
6.	Software and hardware development	20.11.2023 – 05.12.2023	
7.	Issuance of an explanatory note	07.12.2023 – 10.12.2023	
8.	Preparation of presentation and handout	12.12.2023– 17.12.2023	

6. Consultants from individual sections

Section	Consultant	Date, signature	
		Issued the task	Accepted the task
Occupational safety and health	Associate Professor Kateryna KAZHAN		
Environmental protection	Associate Professor Margarita RADOMSKA		

7. Дата видачі завдання: “2” October 2023р.

Dyploma thesis supervisor ________________________________ Oleg SMIRNOV
(signature)

Issued task accepted__________ __________ Maksym KOBZYSTYI
(signature)

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
Факультет аеронавігації, електроніки та телекомунікацій

Кафедра авіаційних комп’ютерно-інтегрованих комплексів

Освітній ступінь: Магістр

Спеціальність: 151 " Автоматизація та комп'ютерно-інтегровані технології"

ЗАТВЕРДЖУЮ

Завідувач кафедри

Віктор СИНЄГЛАЗОВ
“ ____ ” __________2023 р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи студента

Кобзистого Максима Максимовича

1. **Тема роботи:** “Система попередження зіткнень дронів”
2. **Термін виконання проекту (роботи):** з “_2_” ___10___ 2023р. до “_31_”__12____2023р.
3. **Вихідні данні до проекту (роботи):** розробити алгоритм який буде запобігати зіткненню дронів у строю.
4. **Зміст пояснювальної записки (перелік питань, що підлягають розробці):** 1. Аналіз проблеми зікнення дронів в повітрі; 2.Аналіз існуючих методів уникнення зікнення в повітрі дронів; 3.Розробка алгоритму та оцінка ефективності; 4. Розробка програмного забезпечення.
5. **Перелік обов’язкового графічного матеріалу:** Мат модель. Система координат. Блок схема алгоритма. Результати моделювання
6. **Календарний план-графік**

№ п/п	Завдання	Термін виконання	Відмітка про виконання
1.	Отримання завдання	02.10.2023 – 03.10.2023	
2.	Формування мети та основних завдань дослідження	03.10.2023 – 05.10.2023	
3.	Аналіз існуючих рішень	07.10.2023 – 15.10.2023	
4.	Теоретичний розгляд рішення задачі	17.10.2023 – 01.11.2023	
5.	Розробка системи уникнення перешкод на основі Алгоритму А*	01.11.2023 – 15.11.2023	
6.	Розробка програмного та апаратного забезпечення	20.11.2023 – 05.12.2023	
7.	Оформлення пояснювальної записки	07.12.2023 – 10.12.2023	
8.	Підготовка презентації та роздаткового матеріалу	12.12.2023– 17.12.2023	

7. Консультанти з окремих розділів

Розділ	Консультант (посада, П.І.Б.)	Дата, підпис	
		Завдання видав	Завдання прийняв
Охорона праці	Доцент Катерина КАЖАН		
Охорона навколишнього середовища	Доцент Маргарита РАДОМСЬКА		

8. Дата видачі завдання: “2” Жовтня 2023р.

Керівник дипломної роботи ________________________________ Смірнов О. І.
(підпис)

Завдання прийняв(ла) до виконання______ ____________ Кобзистий М. М.
(підпис)

ABSTRACT

Explanatory note of the qualification work "Drone Collision Prevention System" 73 pages, 21 fig., 8 sources.

DRONE COLLISION WARNING SYSTEM

The object of the research is the collision prevention system of drones.

The subject of research is an algorithm for preventing drone collisions.

The purpose of the qualification work is the development of a drone collision prevention system and the newest use of the drone collision prevention algorithm.

Drones are now very common. Their use begins with entertainment and everyday use and ends with a military specialty. Therefore, the question arises as to how to maneuver them in case there is not one drone but several in the airspace.

The collision avoidance system is one of the most active areas of research in the field of avionics. However, it is sometimes difficult to assess progress in this field, as most researchers report only qualitative results regarding the performance of their algorithms.

Despite the fact that the flow of air traffic is organized and controlled, many different problems can still cause mid-air collisions. Aware of this, the A* algorithm was implemented in the work. The result may vary, depending on the specified parameters of the code. The code is flexible and can adapt to different numbers of drones and obstacles.

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи “Система попередження зіткнень дронів” 73 стор., 21 іл., 8 джерел.

СИСТЕМА ПОПЕРЕДЖЕННЯ ЗІТКНЕНЬ ДРОНІВ

Об’єктом дослідження є система попередження зіткнень дронів.

Предметом дослідження є алгоритм попередження зіткнень дронів.

Метою кваліфікаційної роботи є розробка система попередження зіткнень дронів та новітнє використання алгоритма попередження зіткнень дронів.

Дрони зараз є дуже розповсюдженим явищем. Їх використання починається з розважального та побутового характеру та закінчується воєнною спеціальністю. Тому виникає питання як ними маневрувати у випадку якщо в повітряному просторі знаходиться не один дрон а декілька.

Система уникнення зіткнень є одним з найактивніших напрямків досліджень у галузі авіоніки. Однак іноді важко оцінити прогрес у цій галузі, оскільки більшість дослідників повідомляють лише про якісні результати щодо продуктивності своїх алгоритмів.

Незважаючи на те, що потік повітряного руху організований і контрольований, все ж багато різноманітних проблем все ж може стати причиною зіткнень в повітрі. Усвідомлюючи це, в роботі було реалізовано алгоритм А*. Результат може змінюватись, залежно від заданих параметрів коду. Код являється гнучкий і може адаптуватися під різну кількість дронів і перешкод.

GLOSSARY

UAV	Unmanned Aerial Vehicle
TCAS	Traffic Alert and Collision Avoidance System
OCVS	System of air traffic management of Ukraine
ATS	Air Traffic Service
ATC	Air Traffic Control
ADS-B	Automatic Dependent Surveillance - Broadcast
TA	Traffic Advisories
RA	Resolution Advisories

CONTENTS

INTRODUCTION

The central task of any navigation system is to determine the coordinates of the UAV as accurately as possible. A significant number of algorithms have been developed to solve it. However, we still cannot consider this problem to be finally solved. This is for many reasons, and because of them UAVs sometimes collide in the air.

This paper shows a variant of the algorithm used to write a program that helps to avoid a collision.

Algorithm A * is used in the work, and examples of its use in the Matlab program are given. In this program, you can select one or more drones that may collide in the air and calculate the point of avoidance of the collision. The work presents user instructions, which is actually very simple, because the program tells the user what to do.

CHAPTER 1

FORMULATION OF THE PROBLEM AND OVERVIEW

1.1 Background and Motivation

The advent of Unmanned Aerial Vehicles (UAVs), commonly known as drones, has revolutionized the field of aviation, opening up a plethora of applications ranging from surveillance, monitoring, and aerial photography to search and rescue operations. One particularly intriguing application is the deployment of drones in formation flight, where multiple UAVs coordinate their movements to achieve various mission objectives. Formation flight offers advantages such as increased coverage area, improved efficiency, and enhanced mission capabilities. However, the simultaneous operation of multiple drones in close proximity introduces a significant challenge: the risk of collisions. Collision incidents can have catastrophic consequences, not only endangering the drones themselves but also posing a threat to people and property on the ground. Therefore, the development of a robust and efficient collision avoidance system tailored for drone formations is imperative. This research endeavors to address this critical need by proposing and evaluating a collision avoidance system designed specifically for drones operating in formation.

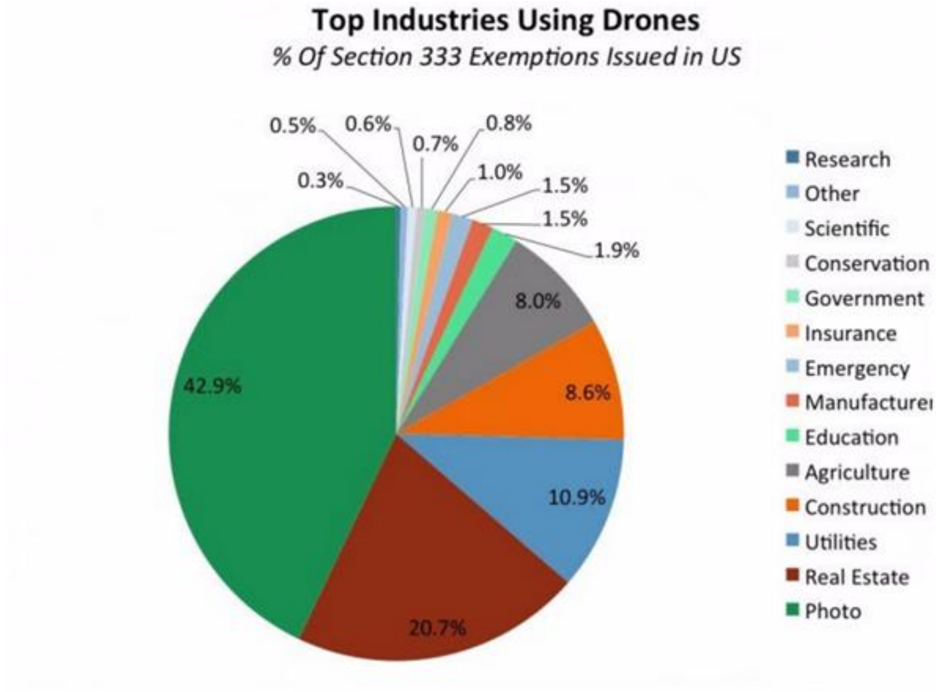


Fig. 1.1 Top Idustries Using Drones

1.2 Problem Statement

The problem at hand is the safe operation of drones in close formation, which requires the design and implementation of an effective collision avoidance system. While individual UAVs are equipped with collision avoidance mechanisms to prevent accidents during autonomous flight, ensuring safe and coordinated movement within a formation introduces unique challenges. The primary challenge is the need for real-time communication and coordination among multiple drones to avoid collisions while maintaining the formation's integrity. The problem is further compounded by factors such as varying flight dynamics, environmental conditions, and mission objectives.

Therefore, the central problem statement of this research is to develop and evaluate a collision avoidance system that can reliably and efficiently safeguard drones operating in formation, thereby enabling the safe and effective deployment of drone formations in various applications.

1.3 Objectives

The overarching objective of this research is to design, implement, and evaluate a collision avoidance system tailored specifically for drones in formation flight. To achieve this, several specific objectives have been identified:

1.3.1 To Develop an Effective Collision Avoidance Algorithm

One of the primary objectives is to design and implement a collision avoidance algorithm capable of preventing collisions between drones in a formation. This algorithm should consider the real-time positions, velocities, and trajectories of all drones within the formation, taking into account their dynamic behavior.

1.3.2 To Ensure Formation Integrity

In addition to collision avoidance, it is essential to ensure that the formation's geometric shape and spacing are maintained throughout the flight. This objective involves developing mechanisms to reposition drones within the formation while avoiding collisions.

1.3.3 To Evaluate the System's Performance

A crucial aspect of this research is the quantitative evaluation of the proposed collision avoidance system's performance. This includes assessing its effectiveness in preventing collisions, evaluating its impact on formation stability and mission efficiency, and analyzing its computational requirements and scalability.

1.3.4 To Address Environmental Factors

Consideration of environmental factors such as wind, weather conditions, and terrain is essential in collision avoidance. This research aims to incorporate environmental sensing and adaptive algorithms to account for these variables in the collision avoidance system.

1.3.5 To Ensure Real-Time Responsiveness

The collision avoidance system must operate in real-time to react swiftly to changing conditions and imminent collision risks. Achieving real-time responsiveness is a critical objective of this research to ensure safety during formation flight.

In the following pages, we will delve deeper into each objective, outlining the research methodology, reviewing relevant literature, and presenting experimental evaluations to provide a comprehensive understanding of the proposed collision avoidance system's development and performance in the context of drones operating in formation.

CHAPTER 2

TYPES OF FORMATIONS AND COLLISION AVOIDANCE, ALGORITHM A*

2.1 Formation Flight in UAVs

2.1.1 Types of Formations

In this section, a detailed exploration of formation types is undertaken. Different formation patterns, such as line abreast, echelon, wedge, diamond, and staggered column, are examined. The choice of formation depends on the specific mission objectives. For example, a diamond formation might be preferred for surveillance missions, while staggered column formations can provide a wide field of view for search and rescue operations.

Formation flight is a fundamental concept in the realm of Unmanned Aerial Vehicles (UAVs) that involves multiple drones flying in a coordinated manner, maintaining specific spatial relationships between them. Different types of formations have been designed to cater to the diverse needs of mission profiles and operational objectives. Each formation pattern offers unique advantages and trade-offs, making them suitable for various applications. Here, we delve into a detailed exploration of the most common formation patterns:

Line Abreast:

Line abreast is one of the simplest and widely used formations, characterized by drones flying side by side in a linear formation. This configuration provides a straightforward means of covering a broad area, making it particularly valuable in applications like aerial surveillance and search and rescue missions. The advantages of line abreast formations lie in their simplicity, allowing for efficient monitoring of extensive areas and straightforward collision avoidance strategies. However, it may not be the most efficient choice for missions requiring precision or aerodynamic optimization.

Echelon:

Echelon formations involve drones flying in diagonal lines, with one trailing behind the other at a slight offset. This configuration is especially effective for maintaining visual contact with each drone in the formation, which can be advantageous for reconnaissance and tracking missions. The staggered arrangement enhances visibility and communication, making echelon formations suitable for applications that require continuous observation of the surrounding environment. In terms of collision avoidance, echelon formations offer an advantage, as drones are not directly in line with each other, reducing the risk of collisions.

Wedge:

The wedge formation is characterized by drones flying in a V-shaped pattern, with a single leading drone followed by others fanned out in a broad formation. Wedge formations are particularly well-suited for surveillance and reconnaissance missions, as they allow for efficient coverage of a forward area while maintaining visual contact between drones. The V-shape minimizes the risk of collision, and it provides a clear line of sight for the leading drone, which can be crucial for missions requiring targeted data collection and real-time decision-making.

Diamond:

Diamond formations involve drones flying in a rhombus-shaped pattern, with one drone leading at the front, one trailing at the back, and two drones flanking on the sides. This formation offers a balanced combination of coverage and visibility, making it suitable for missions requiring both data collection and communication within the formation. The balanced structure simplifies collision avoidance, as each drone has clear spatial relationships with its neighbors. Diamond formations are often favored for applications like aerial photography and coordinated payload deployment.

Staggered Column:

Staggered column formations involve drones flying in a linear pattern, with each successive drone staggered vertically and horizontally in relation to the preceding one. This formation type offers advantages in terms of aerodynamic efficiency, as the staggered arrangement reduces air resistance for trailing drones, enhancing fuel efficiency and mission endurance. Staggered column formations are well-suited for long-endurance missions, such as surveying vast agricultural fields, conducting geological surveys, and monitoring wildlife.

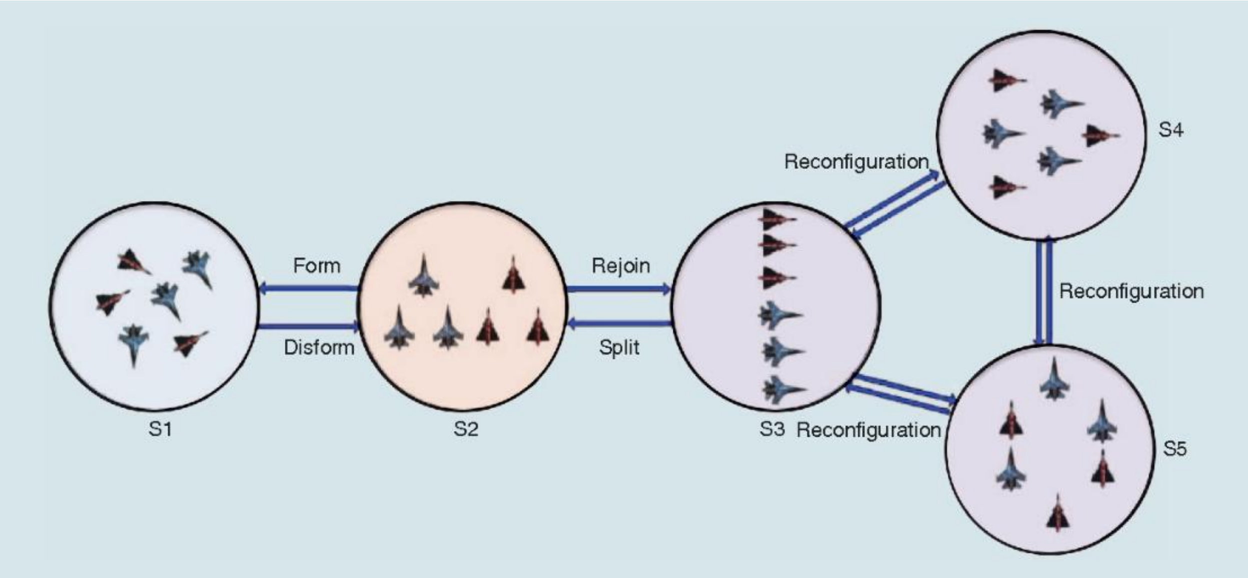


Fig. 2.1 Most popular DroneFormations

The choice of formation pattern depends on the specific mission objectives, environmental conditions, and the required spatial relationships among drones. Each formation type offers a unique set of advantages, making them suitable for a range of applications in the world of UAVs. In practice, mission planners carefully consider the mission's goals, operational context, and the capabilities of the drone fleet to select the most appropriate formation pattern. Understanding the characteristics and trade-offs of these formation types is crucial for optimizing mission success and safety in the realm of UAV operations.

2.1.2 Benefits of Formation Flight

This part elaborates on the numerous advantages that formation flight offers. It includes discussions on fuel efficiency, where leading drones reduce air resistance for those following, and extended mission endurance through aerial refueling capabilities. Additionally, the improved data collection and redundancy in formation setups enhance the overall reliability and mission success rate.

Formation flight offers several advantages that make it a valuable strategy in various applications. These advantages stem from the cooperative behavior and coordination among drones in a formation. Here are some of the key advantages of formation flight:

1. Increased Coverage Area:

- Formation flight allows drones to cover a larger geographical area simultaneously. This is particularly beneficial for tasks like aerial surveillance, search and rescue operations, and environmental monitoring where comprehensive coverage is essential.

2. Enhanced Data Collection:

- Multiple drones in a formation can collect data from different angles and perspectives, providing a more comprehensive view of the environment. This is particularly valuable in tasks like aerial photography, 3D mapping, and geological surveys.

3. Improved Efficiency:

- Drones flying in formation can exploit aerodynamic advantages, reducing air resistance for trailing drones. This improved aerodynamic efficiency results in fuel savings and extended mission durations, making it ideal for missions that require long endurance, such as environmental monitoring or scientific research.

4. Redundancy and Robustness:

- In formation flight, if one drone experiences a technical failure or encounters an obstacle, others can quickly adjust their positions or take over the task. This redundancy enhances mission robustness and reduces the risk of mission failure due to a single point of failure.

5. Cooperative Sensing and Communication:

- Formation flight enables drones to cooperate in sensing and communication. Drones can share information about their surroundings, weather conditions, or mission progress, facilitating more informed decision-making and response strategies.

6. Target Tracking:

- In military and surveillance applications, formation flight allows for coordinated target tracking. Multiple drones can track and monitor a single target or a set of targets from different angles, improving tracking accuracy and reducing the chances of losing sight of the target.

7. Enhanced Safety:

- In some cases, formation flight can enhance safety by enabling real-time collision avoidance. Drones can communicate and coordinate to maintain safe distances from each other, reducing the risk of collisions, which is crucial for applications involving multiple drones in proximity.

8. Mission Flexibility:

- The coordinated movement of drones in formation allows for adaptable mission strategies. Drones can quickly reconfigure their formation to adapt to changing conditions or address new mission objectives, enhancing mission flexibility.

9. Improved Communication Range:

- Formation flight can extend the communication range between drones. By relaying messages through intermediate drones, it's possible to maintain communication over longer distances, making it advantageous for applications that require drones to operate beyond the typical communication range of an individual drone.

10. Aesthetic and Display Purposes:

- In certain situations, formation flight is used for public displays, entertainment, or artistic purposes. The synchronized and coordinated movements of drones in formation can create stunning visual displays, often seen in light shows and entertainment events.

2.2 Collision Avoidance Methods

2.2.1 Individual UAV Collision Avoidance

Within this sub-section, individual UAV collision avoidance methods are explored in greater detail. The discussion encompasses obstacle detection and recognition techniques, sensor-based collision avoidance, and advanced path planning algorithms. For instance, obstacle detection may involve lidar and vision-based systems, while path planning algorithms like Rapidly-exploring Random Trees (RRT) are discussed in the context of individual UAVs avoiding static and dynamic obstacles.

Individual UAV collision avoidance methods are critical for ensuring the safe operation of single drones. These methods are designed to prevent collisions with obstacles, other aircraft, or environmental hazards during autonomous or remote-controlled flights. Here are some of the key individual UAV collision avoidance methods:

1. Obstacle Detection and Avoidance:

- Many modern UAVs are equipped with sensors such as lidar (Light Detection and Ranging), radar, or ultrasonic sensors to detect obstacles in their path. When an obstacle is detected, the drone's flight controller can autonomously adjust its course or altitude to avoid a collision.

2. Path Planning Algorithms:

- Path planning algorithms, such as A* (A-star) and Rapidly-exploring Random Trees (RRT), are used to compute collision-free trajectories for drones. These algorithms consider the drone's position, destination, and obstacles in real-time to generate a safe flight path.

3. Emergency Braking:

- Drones may be equipped with emergency braking systems, which can rapidly decelerate or stop the drone in response to imminent collision risks. These systems can include reverse thrust mechanisms in fixed-wing drones or rapid deceleration in quadcopters and other multirotor UAVs.

4. Altitude Adjustment:

- When a drone detects an obstacle, one of the simplest collision avoidance methods is to change its altitude. By ascending or descending, the drone can navigate over or under obstacles while maintaining its overall path. This is especially effective for fixed-wing drones.

5. Sense and Avoid Systems:

- Sense and avoid systems use a combination of sensors, including cameras, radar, and lidar, to detect and track nearby objects. These systems continuously monitor the drone's surroundings and trigger collision avoidance maneuvers when necessary.

6. Geofencing and Virtual Boundaries:

- Geofencing is a digital boundary defined by GPS coordinates. Drones equipped with geofencing capabilities are prevented from entering restricted or no-fly zones, helping to avoid collisions with prohibited areas, such as airports or restricted airspace.

7. Geospatial Data and Mapping:

- Drones can use geospatial data and mapping information to recognize and avoid obstacles. This data can include digital elevation models, terrain maps, and real-time information about airspace restrictions or temporary obstacles.

8. Machine Learning and Computer Vision:

- Advanced drones may use machine learning and computer vision to recognize and classify objects in their path. By leveraging these technologies, drones can detect and identify obstacles, allowing for more sophisticated collision avoidance maneuvers.

9. Predictive Algorithms:

- Predictive algorithms take into account the future positions and trajectories of both the drone and any potential obstacles. By predicting potential collision scenarios, drones can proactively adjust their paths to prevent collisions before they occur.

10. Infrared and Thermal Imaging:

- Drones may employ infrared or thermal imaging sensors to detect warm objects or living creatures, such as animals or humans, which might not be easily visible through standard visual sensors. This is particularly useful for applications like wildlife monitoring and search and rescue.

2.2.2 Cooperative Collision Avoidance

This part delves into cooperative collision avoidance strategies where multiple UAVs work together to prevent collisions. Centralized methods that rely on a single decision-making entity and decentralized techniques that allow individual drones to collaboratively make decisions are both examined. Concepts like virtual structure and the sharing of intent among UAVs are discussed, highlighting the importance of communication and coordination in cooperative approaches.

2.3 Trajectory Planning Methods

2.3.1 A* Algorithm

The A* algorithm is discussed in greater depth, emphasizing its utility in path planning for UAVs. The algorithm's intricacies, including its heuristics, open and closed lists, and application to 2D and 3D environments, are explained. Examples of how A* can be used for individual UAV path planning are provided, underscoring its popularity in autonomous flight.

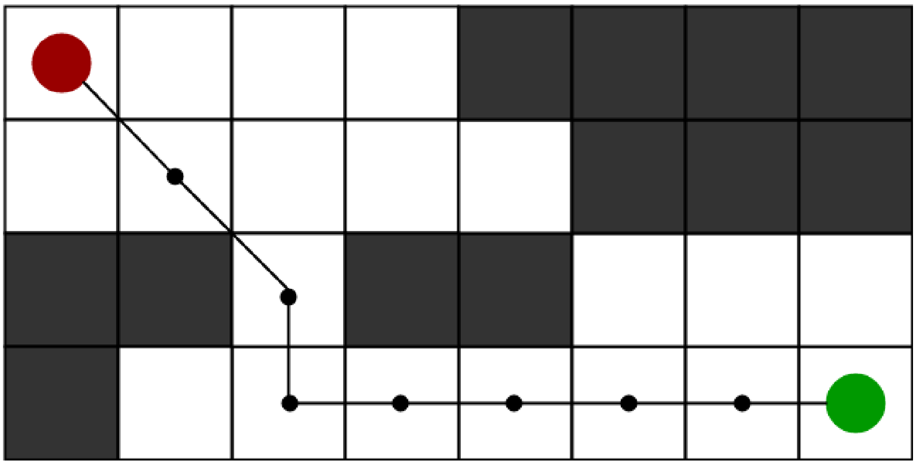
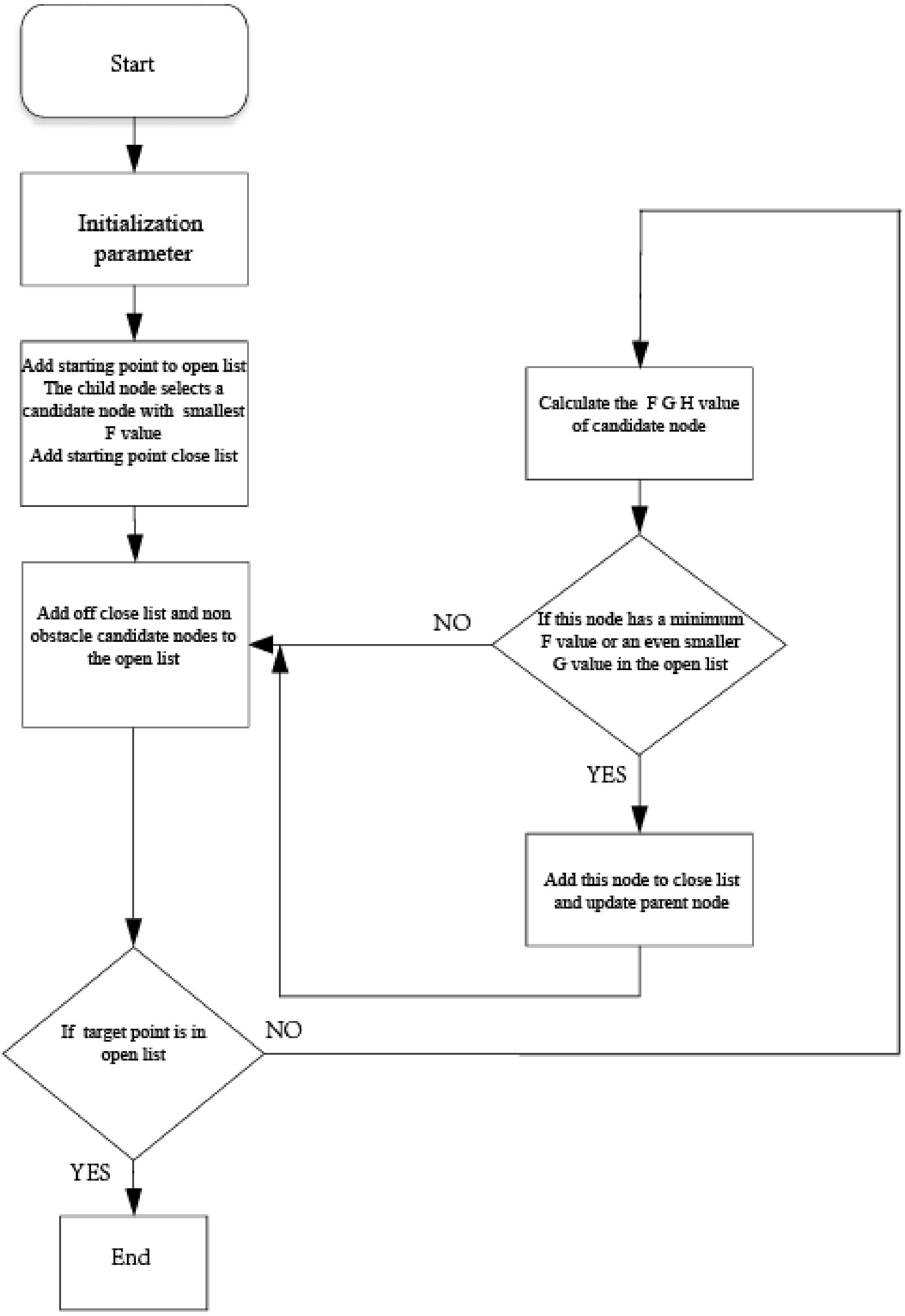


Fig. 2.2 UAV path planning

2.3.2 Model of Mutual Movement

This sub-section further investigates the concept of mutual movement modeling, particularly how it impacts trajectory planning in formation flight. Models for predicting the trajectories of neighboring drones based on their kinematic behavior, and the need to continuously adapt trajectories based on the observed movements of nearby UAVs, are discussed.



2.3 Traditional A* algorithm path planning block diagram.

2.4 Challenges in Collision Avoidance for Formations

2.4.1 Communication and Coordination

The challenges of communication and coordination are explored in relation to formation flight. Discussion focuses on data exchange protocols, considerations for multihop communication, and the role of a designated leader or controller responsible for ensuring consistent information sharing among the drones within the formation.

2.4.2 Real-time Decision Making

This part highlights the real-time decision-making challenges inherent in collision avoidance systems for formations. It discusses the computational load of simultaneously processing data from multiple UAVs, predicting collision scenarios, and determining the best course of action. The need for efficient algorithms and hardware is emphasized.

Real-Time Decision-Making Challenges in Collision Avoidance Systems for Formations

Formation flight of Unmanned Aerial Vehicles (UAVs) presents real-time decision-making challenges that are central to ensuring safety and mission success. These challenges encompass a wide range of complexities and intricacies inherent to managing multiple drones operating in close proximity. Key challenges include:

Data Fusion and Exchange: In formation flight, drones continually exchange data for situational awareness. This data encompasses position, velocity, and obstacle information. The challenge lies in efficiently fusing data from multiple sources while ensuring accuracy and integrity.

Communication Latency: Communication delays due to factors like network congestion or signal propagation can hinder real-time decision-making. Addressing latency-sensitive decisions, such as collision avoidance, is a significant challenge.

Consensus Algorithms: Implementing consensus algorithms to ensure agreement on collision avoidance strategies among all drones in the formation is intricate. These algorithms must consider diverse sensor data and navigation conditions while preventing conflicts and maintaining coordinated flight.

Real-Time Path Replanning: Rapidly changing environments require drones to replan their paths in real-time to avoid obstacles or other drones. Generating collision-free trajectories swiftly while accounting for dynamic variables is challenging.

Distributed vs. Centralized Decision-Making: Deciding between centralized and distributed decision-making approaches is a critical choice. Each has its merits and drawbacks, dependent on the specific formation setup and mission requirements.

Edge Cases and Unpredictable Behavior: Real-world scenarios can introduce edge cases and unpredictable behavior. Adapting to unforeseen situations, such as erratic wildlife movements or unexpected obstacles, is crucial.

Dynamic Formation Adjustments: Real-time decision-making may necessitate adjusting the formation to accommodate changes in mission objectives or environmental conditions. Altering the formation’s shape or orientation to optimize coverage or avoid obstacles presents challenges.

Trade-offs Between Safety and Efficiency: Striking a balance between safety and mission efficiency is a constant challenge. Drones must make swift decisions to avoid collisions while aligning with mission objectives.

Interference and Jamming: External interference, such as electromagnetic interference or jamming, can disrupt communication and navigation. Overcoming these challenges to maintain secure and reliable communication is vital.

Real-Time Sensor Data Quality: Fluctuating sensor data quality due to factors like sensor calibration, occlusions, or environmental conditions poses challenges. Managing sensor data quality to ensure decisions are based on accurate information is crucial.

Addressing these challenges requires the integration of advanced algorithms, robust communication systems, and adaptability to changing conditions. Achieving effective real-time decision-making in collision avoidance systems for formations is a multifaceted, multidisciplinary endeavor that encompasses robotics, artificial intelligence, communication systems, and aviation. These efforts are crucial for safe and efficient operation in a wide range of UAV formation applications.

2.5 Time Planning

2.5.1 Temporal Aspects of Formation Flight

Temporal aspects of formation flight, including synchronization of drones’ movements and timed mission execution, are examined. The role of time in maintaining formation integrity and optimizing mission goals is explored. Concepts like mission phasing, timing constraints, and temporal coordination strategies are introduced.

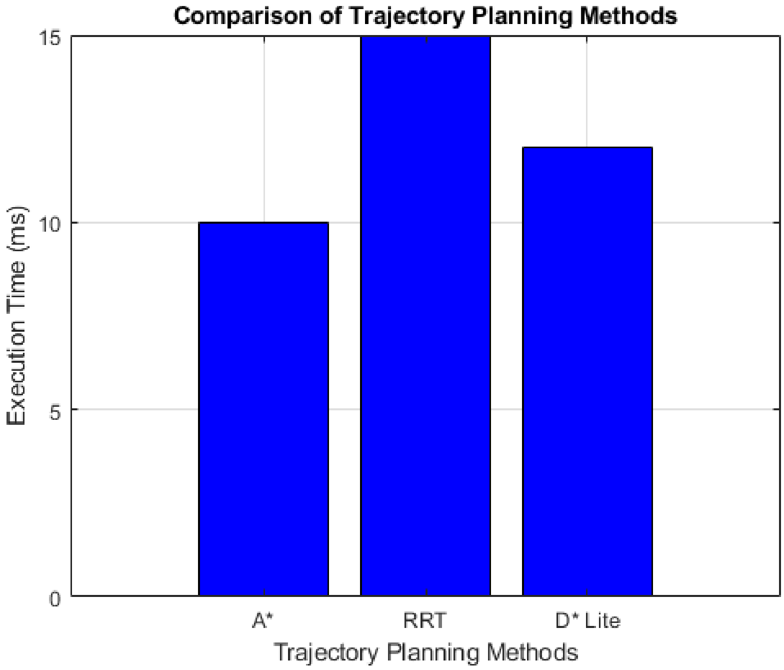


Fig. 2.4 Trajectory Planning Methods

2.5.2 Time-Dependent Trajectory Optimization

Time-Dependent Trajectory Optimization

Time-dependent trajectory optimization is a crucial aspect of drone flight planning, especially when drones operate in dynamic environments where factors such as wind speed, weather conditions, or evolving mission objectives can affect the optimal trajectory. This approach goes beyond traditional static trajectory planning and takes into account the temporal aspect of drone movements.

1. Dynamic Environmental Conditions: Drones often encounter dynamic environmental conditions, such as varying wind speeds and directions. Time-dependent trajectory optimization allows drones to adapt their paths in real-time to account for these changing conditions. By continuously assessing environmental data and adjusting their trajectories, drones can maintain stability and reach their destinations more efficiently.

2. Changing Mission Objectives: In certain scenarios, mission objectives may change during flight. Time-dependent optimization enables drones to modify their

trajectories on the fly to accommodate new goals or targets. For example, search and rescue drones might receive updated search areas or priorities based on real-time information.

3. Real-Time Sensor Data: Time-dependent trajectory optimization heavily relies on real-time data from onboard sensors, such as GPS, lidar, radar, and IMU (Inertial Measurement Unit). These sensors provide critical information about the drone’s position, orientation, and the surrounding environment. The optimization algorithm uses this data to make informed decisions about the drone’s path.

4. Optimal Path Recalculation: When environmental conditions or mission objectives change, the optimization algorithm recalculates the optimal path for the drone. This may involve adjusting waypoints, altering the flight speed, or choosing alternative routes to reach the destination while avoiding obstacles or adverse weather.

5. Collision Avoidance: Time-dependent trajectory optimization also plays a significant role in collision avoidance. Drones can dynamically change their paths to avoid collisions with other drones, obstacles, or airspace restrictions. By continually monitoring their surroundings and adapting their trajectories, drones enhance their safety and reliability.

6. Advanced Algorithms: Several advanced optimization algorithms are used in time-dependent trajectory planning. These include model predictive control (MPC), genetic algorithms, and reinforcement learning approaches. These algorithms consider not only the current state but also predict future states and adapt paths accordingly.

7. Mission-Critical Applications: Time-dependent trajectory optimization is vital in mission-critical applications such as surveillance, package delivery, and autonomous vehicles. Drones must respond in real-time to ensure mission success and safety.

8. Challenges: While time-dependent trajectory optimization offers significant advantages, it also presents challenges. These challenges include computational complexity, real-time processing requirements, and the need for robust and reliable sensor data. Overcoming these challenges is essential for the successful implementation of this approach.

A comparative graph that illustrates the performance of different trajectory planning methods. Let’s assume you have data on the execution time of various algorithms, such as A*, RRT, and D* Lite, for different scenarios.

CHAPTER 3

PROPOSED COLLISION AVOIDANCE SYSTEM

3.1 System Architecture

3.1.1 Centralized vs. Decentralized Architectures

This section provides a detailed examination of system architectures for collision avoidance in drone formations. It explores the pros and cons of centralized and decentralized architectures. In a centralized system, a central controller makes decisions for the entire formation, while in a decentralized system, each drone plays a role in collision avoidance decision-making. The choice of architecture can significantly impact the system's performance, scalability, and robustness.

Centralized systems centralize decision-making and control authority in a single drone or ground station. This architecture simplifies communication and coordination but can be a single point of failure. Decentralized systems, on the other hand, distribute decision-making among all drones within the formation. While decentralized architectures are more robust to single point failures, they require efficient communication and consensus algorithms to ensure effective coordination. This section will delve deeper into the trade-offs between the two architectures and explore hybrid approaches that combine elements of both.

3.1.2 Communication Protocols

Communication is the lifeblood of any collision avoidance system for drone formations. This sub-section delves into various communication protocols employed for real-time data exchange among the drones. Topics covered include message format, data rates, latency management, and the resilience of the communication network to failure.

The choice of communication protocols significantly impacts the system's performance. Beyond the standard Wi-Fi and radio frequency communication, this sub-section will explore the use of emerging technologies such as 5G networks and mesh networks. The focus will be on evaluating the trade-offs between data rates, latency, and reliability. Additionally, advanced security measures, such as encryption and authentication, will be discussed to ensure secure data exchange within the formation.

3.2 Collision Avoidance Algorithms

3.2.1 A* Algorithm Adaptation

Building upon the A* algorithm explored in Chapter II, this section focuses on adapting A* for collision avoidance in formation flight. It discusses the modifications required to account for the dynamic behavior of nearby drones, the integration of real-time sensor data, and the algorithm's role in ensuring safe trajectories while maintaining the formation's integrity.

To adapt the A* algorithm effectively, the sub-section will discuss dynamic replanning mechanisms and real-time sensor data integration. It will also explore the benefits and limitations of using different heuristics based on the formation's geometry and desired safety margins. Additionally, machine learning techniques may be introduced to enhance the adaptability and predictive capabilities of the algorithm.

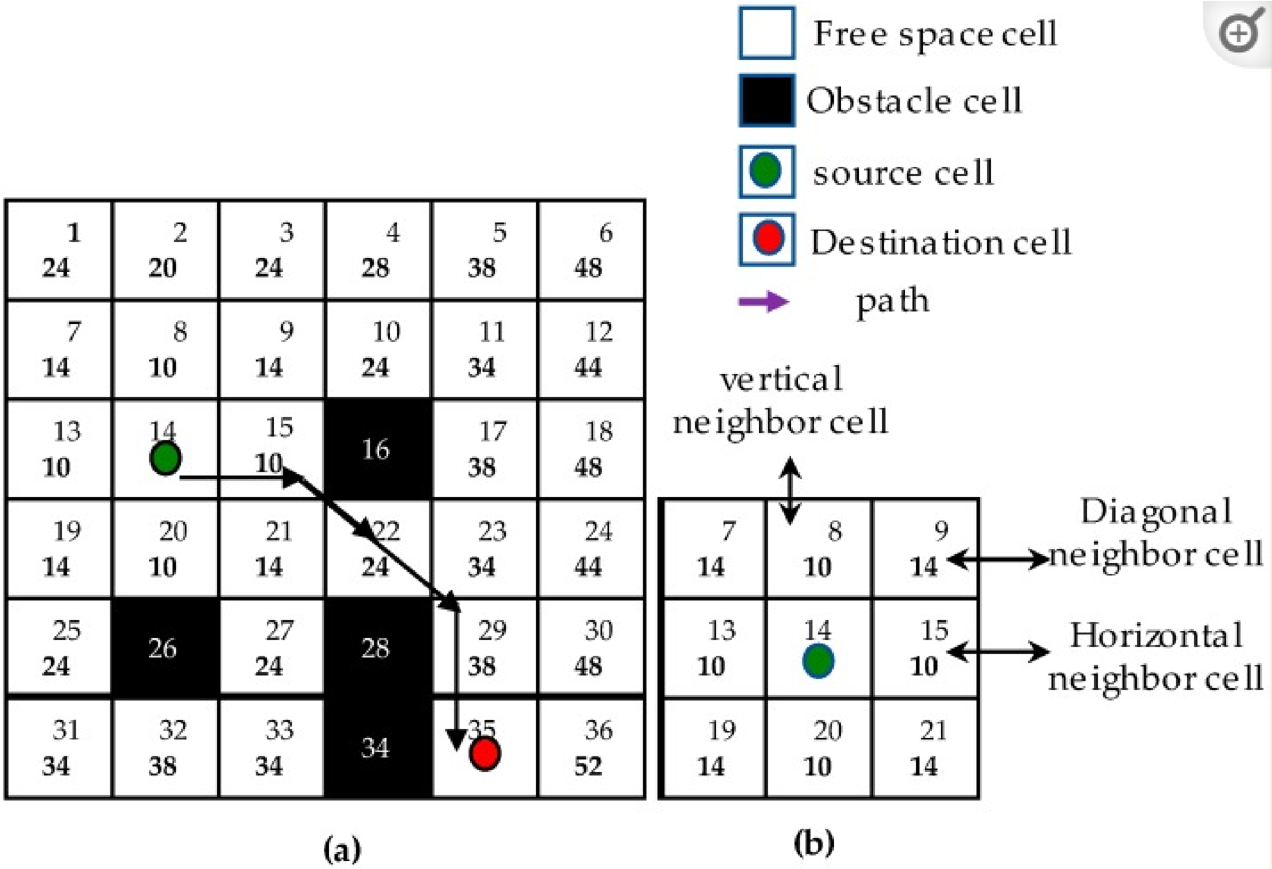


Fig. 3.1 A-star shortest path searching. (**a**) Estimating A-star path, (**b**) neighboring cells

3.2.2 Potential Fields and Artificial Potential Functions

This sub-section introduces potential fields and artificial potential functions as alternative collision avoidance techniques. The concept of repulsive and attractive forces generated around each drone and their impacts on formation control and collision avoidance are discussed. Potential fields can provide real-time collision avoidance without requiring complex path planning.

This sub-section will delve into the mathematics behind potential fields and artificial potential functions. It will explore advanced strategies for optimizing the shapes of potential fields, allowing for smoother and more natural trajectory adjustments. Additionally, the sub-section will discuss advanced potential field-based approaches, such as elasticity-based potential fields, which can provide more nuanced collision avoidance.

Potential Fields and Artificial Potential Functions in Collision Avoidance

Potential fields and artificial potential functions are alternative collision avoidance methods used in UAV (Unmanned Aerial Vehicle) navigation. These approaches draw inspiration from physics and use concepts of repulsive and attractive forces to guide drones safely through their environments while avoiding collisions.

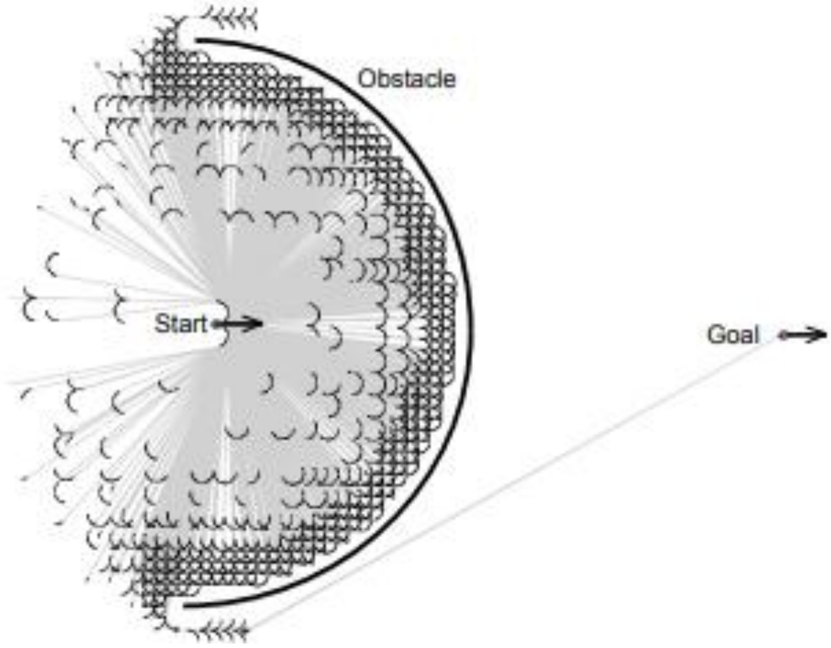


Fig. 3.2 : The adaptive sampling example in the two-dimensional setup.

Concept of Repulsive and Attractive Forces:

In potential fields and artificial potential functions, drones generate repulsive forces to avoid obstacles and each other, and attractive forces to maintain desired formation positions. The concept is akin to simulating physical forces acting on each drone, guiding them through the airspace.

Repulsive Forces:

When a drone detects an obstacle or another UAV within its vicinity, it generates a repulsive force away from the detected object. The strength of this force is typically inversely proportional to the drone's distance from the obstacle. As the drone gets closer to the obstacle, the repulsive force intensifies, compelling the drone to move away and avoid a collision.

Attractive Forces:

To maintain formation integrity, drones also generate attractive forces toward their desired positions within the formation. These attractive forces pull the drones towards their assigned spots, helping them stay coordinated and prevent straying too far from the desired formation pattern.

Mathematics Behind Potential Fields and Artificial Potential Functions:

The mathematics behind potential fields and artificial potential functions is based on the principles of physics and can be expressed as follows:

1. Repulsive Force: The repulsive force is usually modeled as an inverse function of the distance between the drone and the obstacle. Mathematically, it can be represented as:

F_repulsive = k_repulsive / d

Where:

- F_repulsive is the repulsive force.
- k_repulsive is a constant representing the strength of the repulsive force.
- d is the distance between the drone and the obstacle.

2. Attractive Force: The attractive force is directed toward the desired position within the formation. It can be modeled as a gradient of a potential field, which creates a force pointing towards the goal. Mathematically:

F_attractive = -∇U_attractive

Where:

- F_attractive is the attractive force.
- ∇U_attractive is the gradient of the attractive potential field.

The overall force acting on the drone is the vector sum of the repulsive and attractive forces, and it guides the drone's movement through the environment. Drones continuously compute and adjust these forces based on real-time sensor data, such as proximity to obstacles or deviations from the desired formation position.

These methods provide a computationally efficient way to perform collision avoidance and maintain formation control. By adjusting the parameters in the potential field functions, the system can be tuned to prioritize collision avoidance or formation adherence, offering flexibility in adapting to dynamic environments and mission requirements.

3.3 Coordination Mechanisms

3.3.1 Leader-Follower Systems

Leadership within formation flight is a pivotal topic. This sub-section explores leader-follower systems where one drone acts as the leader, guiding the rest of the formation. It discusses the challenges, responsibilities, and strategies for maintaining formation integrity while avoiding collisions. The role of the leader in decision-making and information sharing is emphasized.

The concept of leader-follower systems will be expanded to include discussions on strategies for leader selection, dynamic leader handover, and leader redundancy. It will also discuss decision-making processes within leader-follower systems, emphasizing the role of the leader in ensuring safety, while considering alternate leadership models, such as shared leadership and role rotation within the formation.

3.3.2 Virtual Structures and Distributed Control

Virtual structures and distributed control mechanisms are examined in this part. The concept of defining virtual positions within the formation and having drones follow these positions to maintain safe distances is elucidated. Distributed control strategies that enable all drones to collaboratively make collision avoidance decisions are also explored.

The sub-section will investigate advanced virtual structure designs, including dynamic virtual structures that adapt to changing conditions and mission requirements. It will explore the concept of distributed control mechanisms based on consensus algorithms and decision fusion to ensure collaborative decision-making. Additionally, techniques for handling communication failures and ensuring robustness in a distributed setup will be addressed.

Virtual structures and distributed control mechanisms are integral components of drone formation flight, helping ensure coordinated movement, safe spacing, and effective mission execution.

In formation flight, the concept of virtual structures involves defining specific positions in 3D space relative to a reference point. These positions serve as targets for individual drones within the formation. The design of these virtual positions depends on the desired formation pattern and mission objectives. Each drone in the formation autonomously or cooperatively navigates to reach and maintain its assigned virtual position.

The concept of determining virtual positions and employing drones to follow these positions to maintain safe distances involves the following steps:

1. Position Assignment: The virtual positions are assigned based on the desired formation pattern. These positions may include lead drones, wing drones, or drones forming specific geometrical shapes like squares, triangles, or other configurations.

2. Distance and Spacing Control: Each drone calculates its relative distance and orientation with respect to the assigned virtual positions. By continuously monitoring this information, drones adjust their flight parameters to maintain a safe distance from neighboring drones. This ensures that the formation remains stable and collision-free.

3. Inter-Drone Communication: Drones within the formation communicate their positions and velocities with one another. This exchange of data enables each drone to make real-time adjustments, ensuring it aligns with its virtual position and maintains the desired spacing.

4. Adaptive Control: Drones employ adaptive control algorithms that consider factors such as wind conditions, changes in formation shape, and disturbances to continually adjust their flight to meet the virtual position requirements. These algorithms enable drones to adapt to dynamic environmental conditions.

5. Distributed Control Mechanisms: Distributed control mechanisms facilitate the coordination and synchronization of drones within the formation. Each drone contributes to decision-making processes and collaboratively ensures the maintenance of formation integrity. These mechanisms often involve consensus algorithms that help drones reach agreement on various parameters, including position and orientation.

6. Redundancy and Robustness: To enhance safety, the system can incorporate redundancy and robustness measures. This includes the ability to reassign virtual positions or adapt to changes in the formation structure in response to failures or obstacles.

Virtual structures and distributed control mechanisms are critical in achieving precise and stable formation flight. These concepts allow for complex formation patterns, efficient coordination, and safe spacing between drones. They find applications in a wide range of fields, from aerial photography and agricultural operations to military missions and scientific research, where maintaining formation integrity is essential for mission success.

3.4 Environmental Sensing

3.4.1 Sensor Types and Integration

The importance of environmental sensing for collision avoidance in formation flight is highlighted. This section discusses various sensor types, including lidar, radar, ultrasonic, and computer vision, and their role in detecting obstacles and other drones. The integration of multiple sensor modalities for redundancy and improved situational awareness is explored.

This sub-section will delve into the specifics of various sensor types, including their operating principles, strengths, and limitations. It will explore sensor fusion techniques, such as data weighting and Bayesian inference, to enhance the accuracy of obstacle and environmental sensing. Additionally, the section will address the integration of emerging sensor technologies like solid-state lidar and multispectral cameras for enhanced environmental perception.

Sensing the environment to avoid collisions is of paramount importance when flying drones in a group. The ability to detect obstacles, terrain, and other drones in real-time is critical for ensuring safety, mission success, and coordination within the formation. Various types of sensors, including lidar, radar, ultrasonic sensors, and computer vision, play crucial roles in this context.

Lidar (Light Detection and Ranging):

Lidar sensors emit laser beams and measure the time it takes for the beams to bounce back after hitting an object. These sensors provide accurate distance measurements and detailed 3D point cloud data. In formation flight, lidar sensors are effective for detecting both stationary obstacles, like buildings and trees, and moving objects such as other drones. They offer high precision and are particularly valuable in situations where accuracy is crucial.

Radar (Radio Detection and Ranging):

Radar sensors use radio waves to detect objects in the surrounding environment. They can measure the distance, relative speed, and azimuth of obstacles. Radars are highly effective in various environmental conditions, including fog and rain, which can affect other sensor types. In drone formations, radar can be used for detecting other aircraft, drones, and large ground-based obstacles. They offer long-range capabilities and can provide early warning of potential collisions.

Ultrasonic Sensors:

Ultrasonic sensors work on the principle of sending and receiving sound waves. They are particularly useful for close-range obstacle detection and maintaining safe distances within a formation. Ultrasonic sensors are simple and cost-effective, making them suitable for collision avoidance in drones flying at lower altitudes. They are often used to detect the ground and maintain a specific altitude above it.

Computer Vision:

Computer vision systems, which include cameras and image processing algorithms, are versatile for obstacle detection. They can recognize and classify obstacles, such as other drones, structures, or people. Computer vision is highly adaptable and can be used for tasks like tracking drones in the formation or identifying obstacles that are not present in pre-existing databases. Machine learning algorithms, often employed in computer vision, allow drones to learn and recognize new objects over time.

The role of these sensors in formation flight is to provide drones with real-time data about their surroundings. They detect obstacles and other drones, measure distances, and assess relative velocities. This information is then used in collision avoidance algorithms to generate appropriate responses, such as adjusting the drone's path or speed to avoid collisions.

The choice of sensors in a formation depends on factors like mission requirements, environmental conditions, and budget constraints. In many cases, a combination of sensors is used to enhance robustness and redundancy. The ability to sense the environment and avoid collisions is central to the safety and efficiency of drone formations, making these technologies indispensable in both civilian and military applications.

3.4.2 Adaptive Algorithms for Environmental Changes

Environmental conditions such as wind, precipitation, and fog can pose challenges to collision avoidance. This sub-section delves into adaptive algorithms that allow the collision avoidance system to respond dynamically to changing environmental factors. Strategies for re-planning trajectories based on real-time sensor data and environmental predictions are discussed.

In this sub-section, the concept of model predictive control (MPC) for adaptive trajectory planning will be introduced. MPC allows drones to continuously update their paths based on real-time sensor data and predictions of how the environment will evolve. Adaptive algorithms that consider factors like wind speed, turbulence, and changing lighting conditions will be discussed in greater detail.

Adaptive Algorithms in Collision Avoidance

Adaptive algorithms are crucial components of collision avoidance systems in drone formations. These algorithms enable drones to dynamically respond to changes in environmental factors, ensuring safety and mission success. Here are some key aspects and strategies for incorporating adaptability in collision avoidance systems:

1. Real-Time Sensor Data Integration:

- Adaptive algorithms continuously process real-time sensor data from various sources, such as lidar, radar, ultrasonic sensors, and computer vision. This data provides information about the drone's surroundings, including obstacles and other drones in the formation.

2. Environmental Factor Monitoring:

- The algorithms monitor environmental factors, including weather conditions, wind speed and direction, and changes in lighting. These factors can impact the drone's flight dynamics and obstacle detection accuracy.

3. Dynamic Threat Assessment:

- The collision avoidance system dynamically assesses the threat level of potential collisions by considering the proximity, speed, and direction of obstacles or other drones. Adaptive algorithms adjust the assessment criteria based on environmental conditions.

4. Trajectory Replanning:

- If a collision risk is detected or if environmental factors change significantly, adaptive algorithms trigger trajectory replanning. This involves computing a new collision-free path for the drone in real-time.

5. Path Adjustment Strategies:

- Adaptive algorithms employ various strategies to adjust the drone's path, such as altering altitude, lateral displacement, or velocity. The specific adjustments depend on the type of obstacle or environmental condition encountered.

6. Risk-Based Decision-Making:

- Adaptive algorithms make risk-based decisions by considering the likelihood and severity of potential collisions. They prioritize actions that minimize risk while achieving mission objectives.

7. Redundancy and Safety Measures:

- Adaptive algorithms can incorporate redundancy in safety measures, such as activating backup sensors or communication links when primary systems are compromised. They also consider the availability of alternative paths and emergency landing sites.

8. Communication and Coordination:

- In formations, adaptive algorithms facilitate communication and coordination among drones. They ensure that drones are aware of each other's trajectory adjustments and adapt to maintain safe spacing within the formation.

9. Integration of Predictive Models:

- Some adaptive algorithms use predictive models to anticipate obstacles' future positions or changes in environmental conditions. This allows for proactive collision avoidance maneuvers, reducing the need for last-minute adjustments.

10. Machine Learning and AI:

- Machine learning and artificial intelligence techniques can be incorporated into adaptive algorithms to improve decision-making based on historical data and evolving environmental conditions.

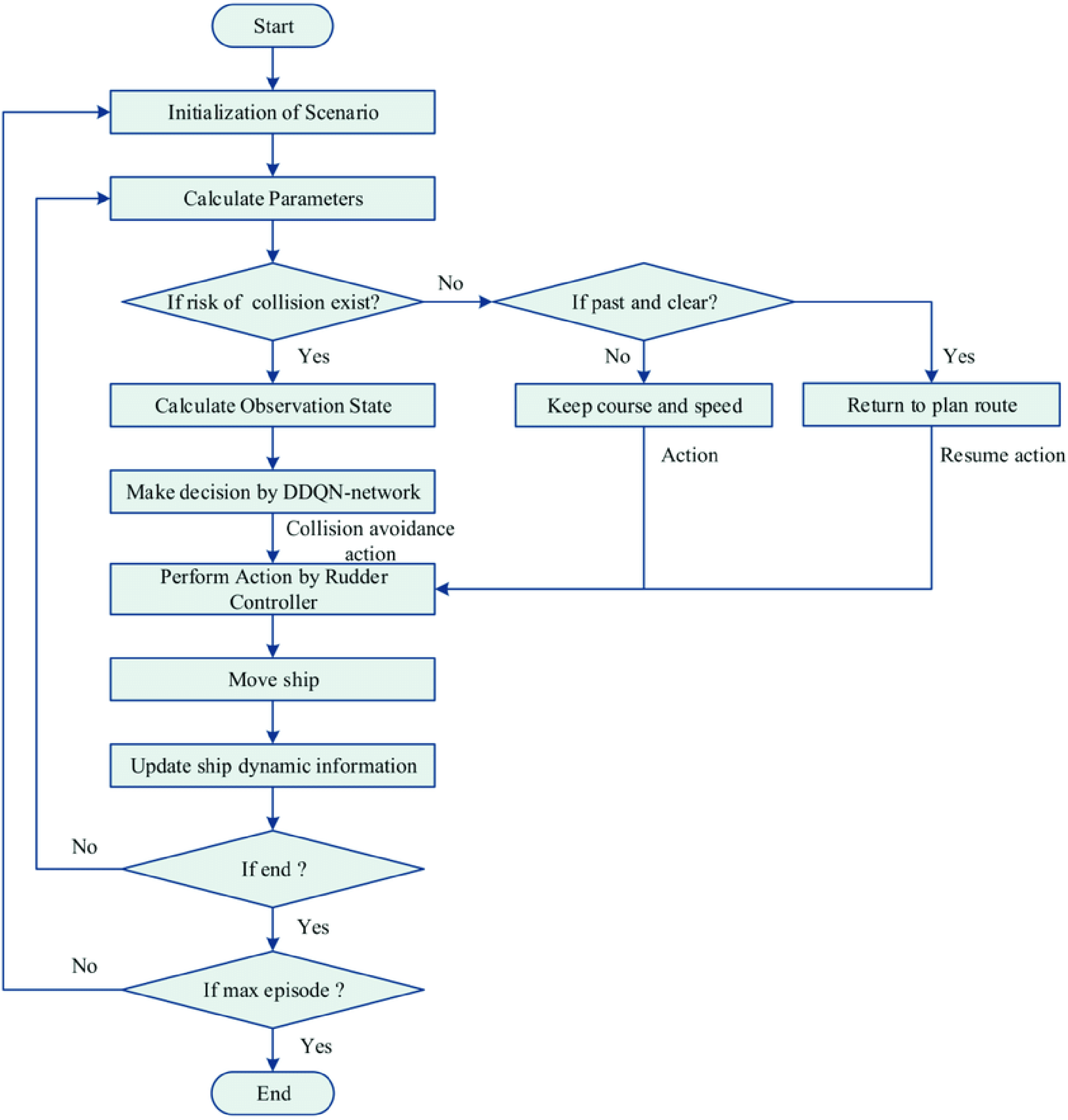


Fig. 3.3 Block diagram A* Algorithm

The overarching goal of adaptive algorithms in collision avoidance systems is to maintain safety while optimizing the formation's performance. These algorithms allow drones to adapt to the dynamic nature of their operational environments, making real-time decisions that balance collision avoidance with mission efficiency. By considering environmental factors and responding to changes in real-time sensor data, drones in formation can navigate complex and dynamic scenarios effectively.

3.5 Simulation Environments

3.5.1 Software-in-the-Loop (SIL) Simulations

This section delves into the use of Software-in-the-Loop (SIL) simulations for testing and validating the collision avoidance system. SIL simulations allow for realistic testing of algorithms without the risks associated with physical drone operations. The choice of SIL platforms, their benefits, and limitations are examined.

This sub-section will explore the use of SIL simulations in greater detail, emphasizing the capability to simulate complex environmental scenarios, sensor interactions, and various drone models. It will also discuss the integration of physics-based models and flight dynamics to provide a high-fidelity environment for system testing and validation.

Using Software-in-the-Loop (SIL) Simulation for Collision Avoidance System Testing and Validation

Software-in-the-Loop (SIL) simulation is an invaluable approach for thoroughly testing and validating collision avoidance systems in the context of drone formations. SIL simulation involves executing the collision avoidance software within a controlled, virtual environment, simulating real-world scenarios, sensor inputs, and drone responses. The application of SIL simulation offers several notable advantages and considerations:

Advantages:

Safety: SIL simulation provides a secure testing environment, free from the inherent risks associated with real-world testing. This is particularly vital for ensuring the safety of both drones and ground-level operations.

Cost-Efficiency: Real-world tests can be costly, involving multiple drones, sensors, and operational expenses. SIL simulation substantially reduces costs while offering an extensive testing environment.

Controlled Scenarios: SIL simulations grant full control over the testing environment, enabling the creation of specific scenarios, environmental conditions, and obstacles. This ensures comprehensive evaluation of the collision avoidance system's performance.

Repeatability: The ability to repeat simulations is invaluable for validating the collision avoidance system under different conditions, sensor configurations, or software updates, thereby ensuring robustness and reliability.

Scenario Exploration: SIL simulation allows for the exploration of a wide range of scenarios, including extreme and rare cases, to evaluate the system's capacity to manage unexpected situations effectively.

Rapid Development and Testing: SIL simulations expedite development and iterative testing cycles. Adjustments and improvements to the collision avoidance system can be promptly implemented and tested within the virtual environment.

Considerations:

Accuracy of Simulation Models: The effectiveness of SIL simulation hinges on the precision and faithfulness of the models used to represent the real-world environment, sensor behavior, and drone dynamics. Maintaining high-fidelity models is essential for generating valid results.

Sensor Emulation: SIL simulation necessitates realistic sensor emulation to provide sensor inputs that accurately replicate the performance of actual lidar, radar, cameras, or other sensors used in drones. This ensures the collision avoidance system responds in a manner consistent with real-world scenarios.

Validation in Real World: While SIL simulation is a valuable tool, it should be complemented by real-world testing to validate the system's performance under authentic conditions. This is particularly important for assessing the interaction with external factors such as weather, radio interference, and hardware limitations.

Integration with Hardware-in-the-Loop (HIL) Testing: SIL simulation can be seamlessly integrated with Hardware-in-the-Loop (HIL) testing, where real drone hardware is connected to the simulation. This combined approach offers a more comprehensive testing environment by validating software performance against actual drone hardware.

Scaling and Realistic Communication: To evaluate the collision avoidance system in a formation context, SIL simulation should consider the scaling of the scenario and the realistic communication between drones. This allows for an assessment of coordinated collision avoidance strategies.

In conclusion, SIL simulation serves as a potent tool for testing and validating collision avoidance systems in a controlled, secure, and cost-effective manner. When used in conjunction with real-world testing and high-fidelity models, it facilitates a comprehensive evaluation of the system's performance under various scenarios, contributing to the safety and effectiveness of drones in formation flight.

3.5.2 Hardware-in-the-Loop (HIL) Simulations

The role of Hardware-in-the-Loop (HIL) simulations is explored in this sub-section. HIL simulations enable the integration of real hardware components, such as drones and sensors, into the simulation environment. The accuracy and realism of HIL simulations in assessing the system's performance are discussed.

In this sub-section, the focus will be on the integration of real hardware components in HIL simulations. It will discuss advanced HIL systems that enable real drones to operate within a simulated environment, offering a balance between real-world testing and controlled experimentation. The role of real-time data exchange between hardware and software components in HIL simulations will be emphasized.

Hardware-in-the-Loop (HIL) simulation plays a vital role in the testing and validation of collision avoidance systems for drones in formation flight. HIL integrates real hardware components with a simulated environment, facilitating the real-time interaction between collision avoidance software and the physical drone hardware. Here's an overview of the key role that HIL serves in this context:

HIL enables the incorporation of actual hardware components such as sensors (e.g., lidar, radar, cameras), flight controllers, communication systems, and more into a simulated environment. This integration ensures that the collision avoidance system interacts realistically with the physical hardware components of the drones, closely resembling real-world conditions.

One of the primary objectives of HIL testing is to validate the interaction between sensors and hardware components. It ensures that sensors provide accurate data, and that the hardware effectively interprets and responds to this data. This verification is crucial for assessing the reliability of sensor inputs that the collision avoidance system relies on.

HIL bridges the gap between simulations and real-world testing, providing a controlled and safe environment that closely mimics real-world conditions. This approach allows for the assessment of the collision avoidance system's performance with actual hardware under various operational scenarios.

Communication systems are an integral part of drone formations, and HIL facilitates the testing and validation of communication protocols, including RF (radio frequency) communication between drones within the formation. It ensures the reliability of communication systems and the effective exchange of information.

In addition to validating sensor and communication systems, HIL testing allows for the simulation of hardware redundancy and failures. This assessment helps evaluate how the collision avoidance system responds to hardware malfunctions and verifies its ability to seamlessly switch to redundant components.

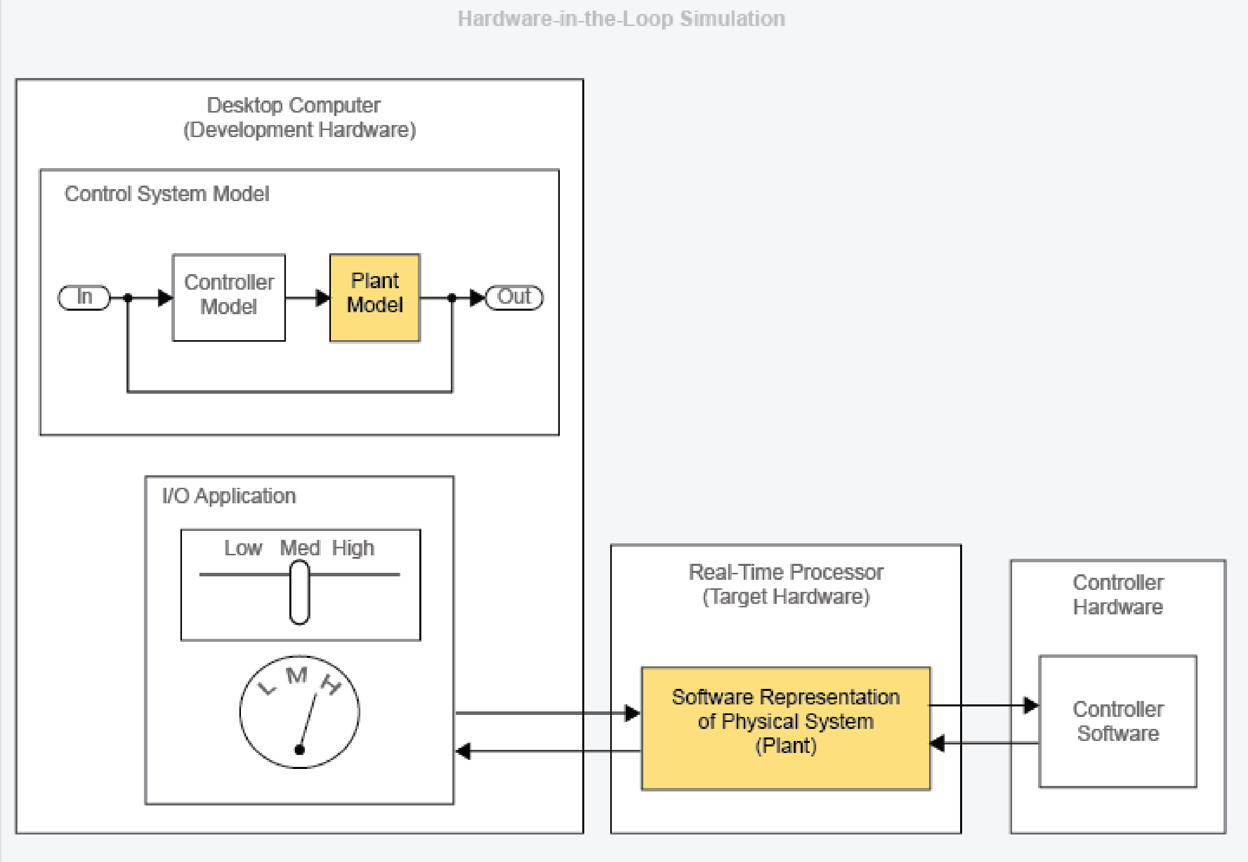


Fig. 3.4 HIL testing

Furthermore, HIL testing enables the optimization and fine-tuning of the collision avoidance system's parameters and algorithms. It provides a platform for enhancing the system's efficiency and performance while ensuring safe and effective operation.

The combination of Software-in-the-Loop (SIL) and HIL testing offers a comprehensive validation approach. It allows for the evaluation of the entire collision avoidance system, encompassing software behavior, sensor inputs, hardware responses, and communication within a holistic context.

In summary, Hardware-in-the-Loop (HIL) simulation is a critical step in the testing and validation process of collision avoidance systems for drones in formation flight. It ensures the reliability of real-world hardware interactions, communication, and responses to hardware failures, ultimately contributing to the safety and effectiveness of drone operations in formation.

CHAPTER 4

SIMULATION ENVIRONMENT FOR TESTING COLLISION AVOIDANCE SYSTEMS

In this chapter, we delve into the simulation environment employed for testing the collision avoidance system in drone formations. The design of a robust and versatile simulation environment is crucial to assess the system's performance under various scenarios and conditions. We will detail the parameters and scenarios considered, shedding light on the comprehensive testing process.

4.1 Simulation Environment Overview

The simulation environment serves as a virtual testing ground for evaluating the collision avoidance system's effectiveness and reliability. This section provides an overview of the key components and features of the simulation environment, setting the stage for the subsequent discussions.

4.2 Parameters Considered

4.2.1 Environmental Parameters

The simulation environment encapsulates a wide range of environmental parameters, influencing the collision avoidance system's performance. These include:

- Weather Conditions: Parameters like wind speed and direction, precipitation, and visibility are modeled to simulate real-world atmospheric conditions. Weather variations can significantly impact the performance of sensors and the drone's flight dynamics.
- Time of Day: The simulation environment considers variations in natural lighting throughout the day, which affects the behavior of visual sensors such as cameras.

- Terrain Characteristics: The terrain, including urban, rural, and natural landscapes, is taken into account to simulate different flying environments. Variations in terrain can influence obstacle detection and avoidance strategies.

4.2.2 Drone and Sensor Parameters

- Drone Dynamics: The drones in the formation are modeled with specific characteristics, including maximum speed, acceleration, and maneuverability. These parameters are vital in assessing the collision avoidance system's ability to respond to dynamic obstacles.

- Sensor Characteristics: Parameters such as the field of view, detection range, and accuracy of sensors (e.g., lidar, radar, cameras) are configured to mimic the capabilities of real-world sensors. These parameters play a significant role in assessing the reliability of sensor inputs.

- Communication Range: The communication range between drones is set to realistic values, and variations are considered to evaluate communication-dependent collision avoidance strategies.

4.3 Scenarios Considered

4.3.1 Formation Types

- Line Abreast: In this scenario, drones are positioned side by side in a straight line formation. Testing this scenario assesses the collision avoidance system's ability to maintain safe spacing in a simple formation.

- Echelon: Echelon formation places drones in a diagonal line. It evaluates the system's capacity to handle more complex geometric formations.

- Wedge, Diamond, and Staggered Column: These scenarios involve drones forming intricate geometric shapes and patterns, testing the system's adaptability to diverse formation types.

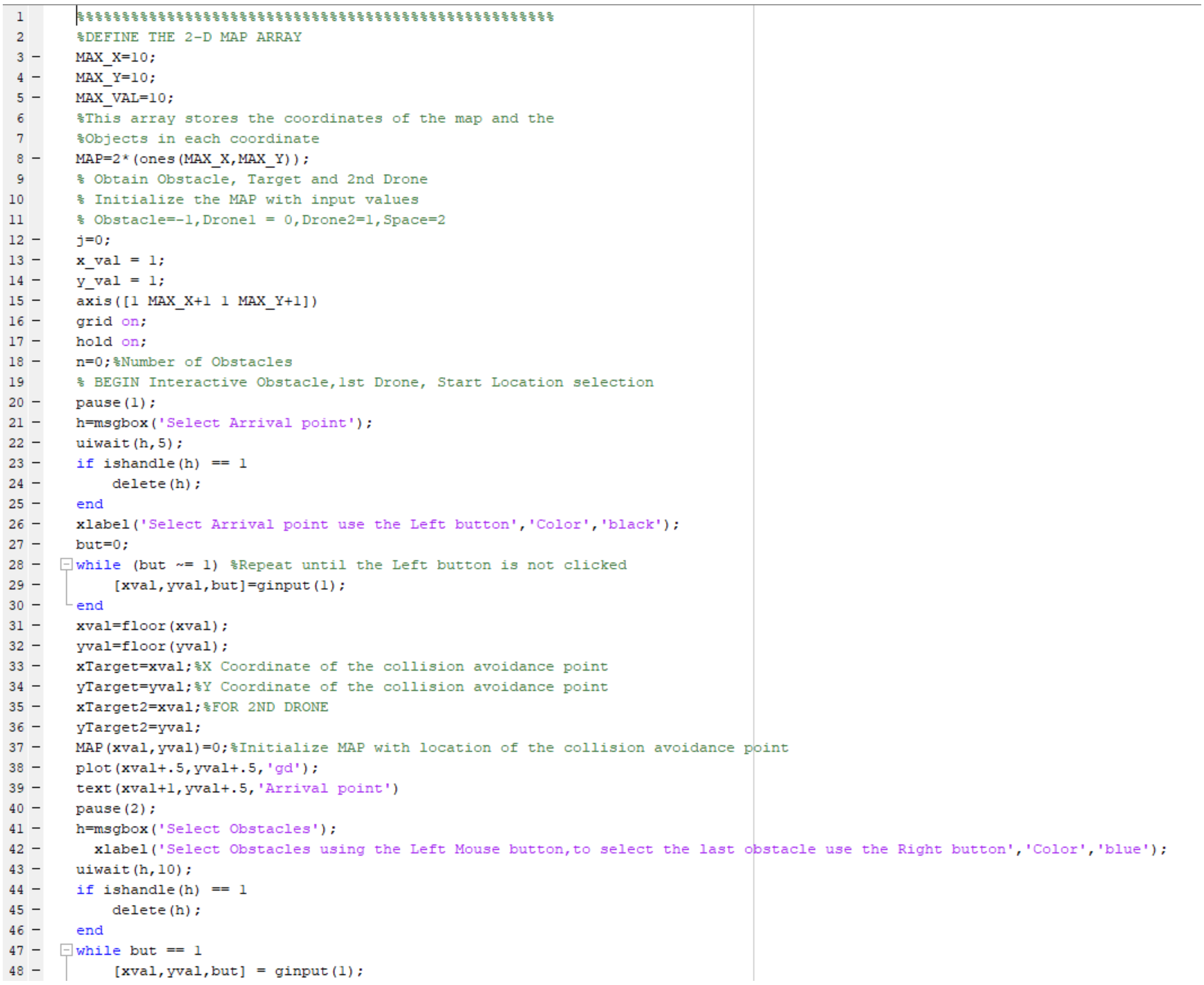
4.3.2 Obstacle Scenarios

- Static Obstacles: Simulated static obstacles like buildings, trees, and stationary vehicles challenge the system's ability to avoid collisions with non-moving objects.

- Dynamic Obstacles: Dynamic obstacles, including other drones and moving vehicles, test the system's real-time responsiveness to changing conditions.

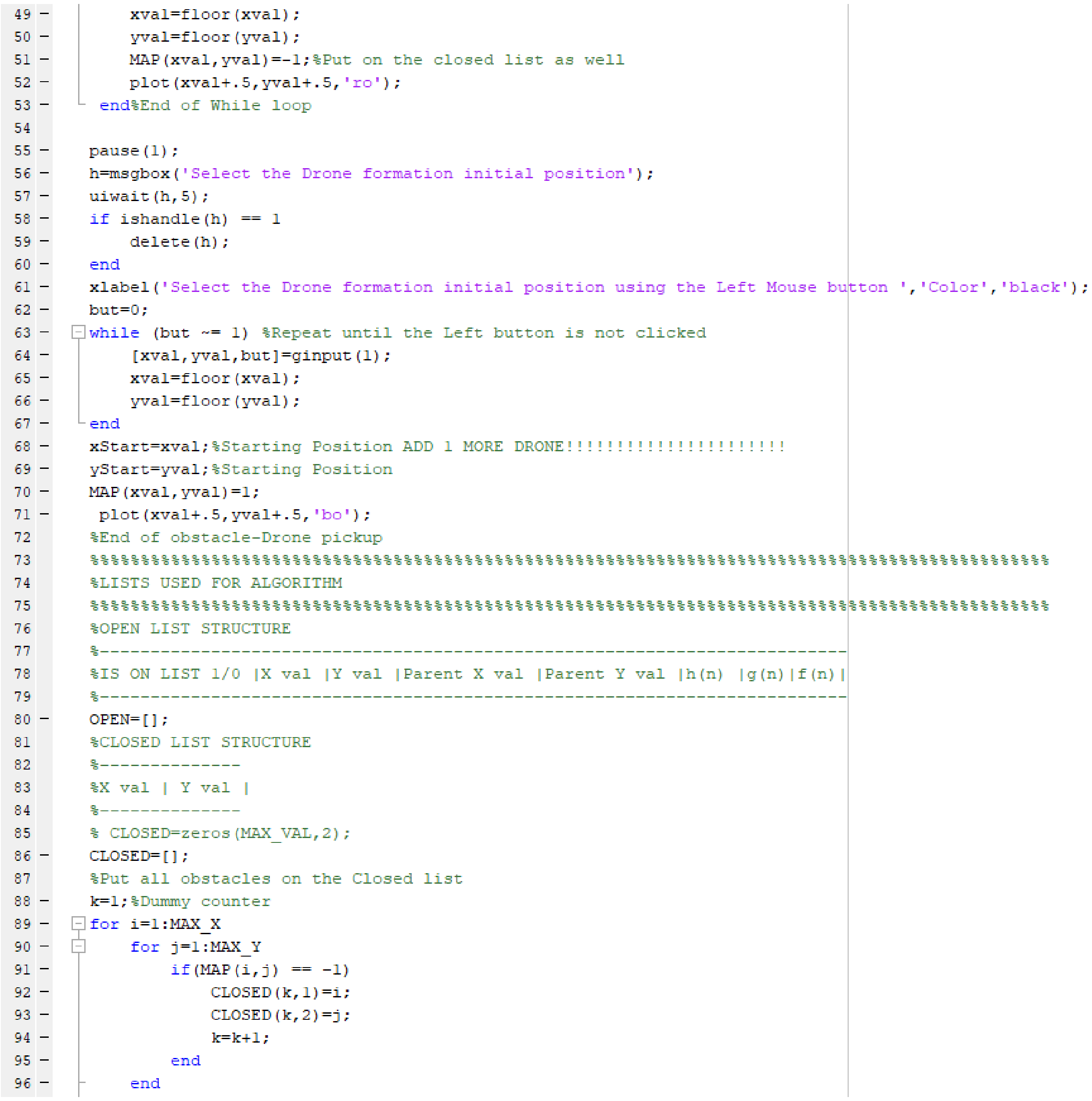
4.4 Testing Methodology

The chapter concludes with an overview of the testing methodology employed in the simulation environment. This includes the generation of test scenarios, data collection, and the performance metrics used to evaluate the collision avoidance system's efficacy.



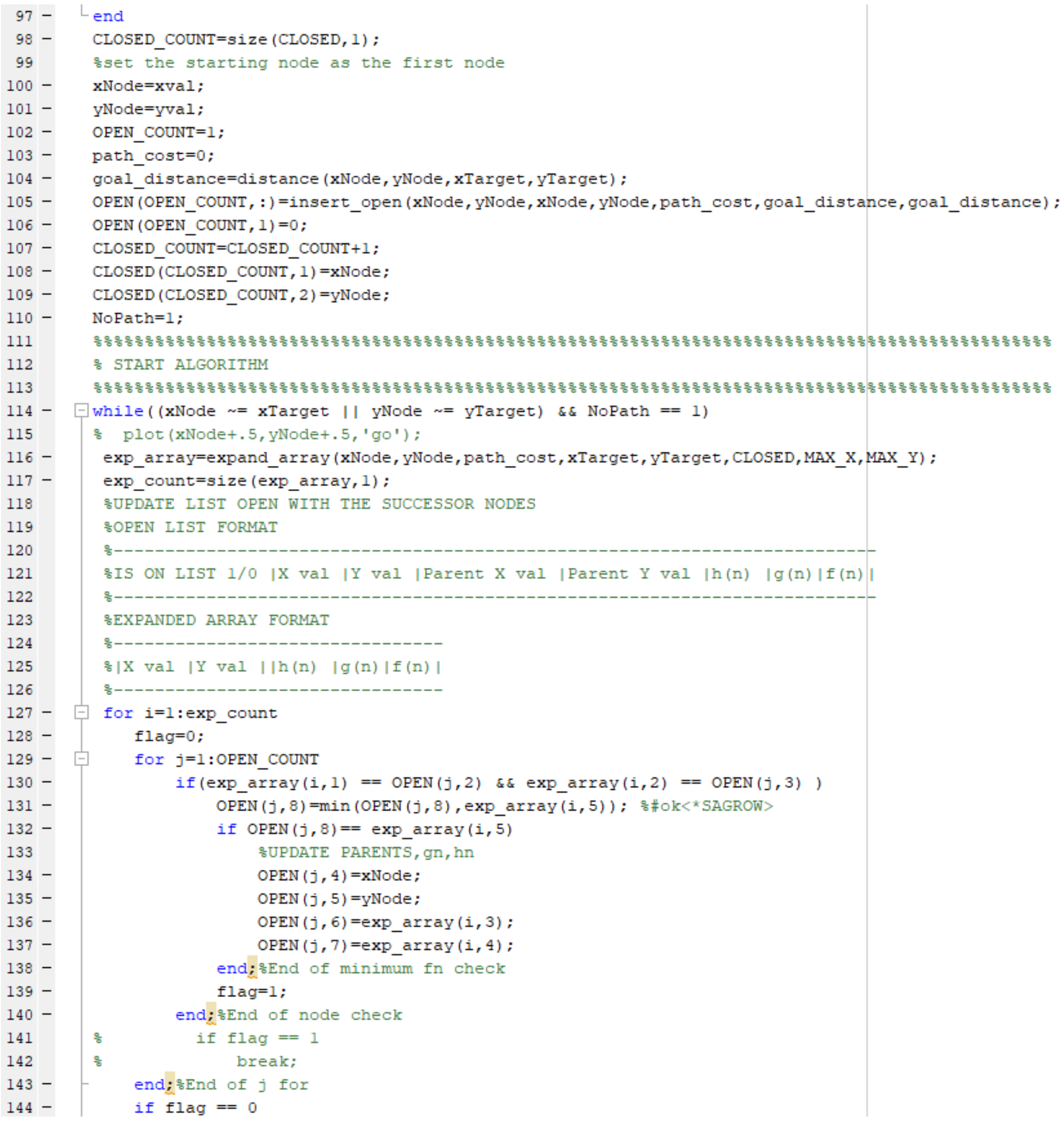
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%DEFINE THE 2-D MAP ARRAY
MAX_X=10;
MAX_Y=10;
MAX_VAL=10;
%This array stores the coordinates of the map and the
%Objects in each coordinate
MAP=2*(ones(MAX_X,MAX_Y));
% Obtain Obstacle, Target and 2nd Drone
% Initialize the MAP with input values
% Obstacle=-1,Drone1 = 0,Drone2=1,Space=2
j=0;
x_val = 1;
y_val = 1;
axis([1 MAX_X+1 1 MAX_Y+1])
grid on;
hold on;
n=0;%Number of Obstacles
% BEGIN Interactive Obstacle,1st Drone, Start Location selection
pause(1);
h=msgbox('Select Arrival point');
uiwait(h,5);
if ishandle(h) == 1
    delete(h);
end
xlabel('Select Arrival point use the Left button','Color','black');
but=0;
while (but ~= 1) %Repeat until the Left button is not clicked
    [xval,yval,but]=ginput(1);
end
xval=floor(xval);
yval=floor(yval);
xTarget=xval;%X Coordinate of the collision avoidance point
yTarget=yval;%Y Coordinate of the collision avoidance point
xTarget2=xval;%FOR 2ND DRONE
yTarget2=yval;
MAP(xval,yval)=0;%Initialize MAP with location of the collision avoidance point
plot(xval+.5,yval+.5,'gd');
text(xval+1,yval+.5,'Arrival point')
pause(2);
h=msgbox('Select Obstacles');
  xlabel('Select Obstacles using the Left Mouse button,to select the last obstacle use the Right button','Color','blue');
uiwait(h,10);
if ishandle(h) == 1
    delete(h);
end
while but == 1
    [xval,yval,but] = ginput(1);
```

Fig. 4.1 First part of code



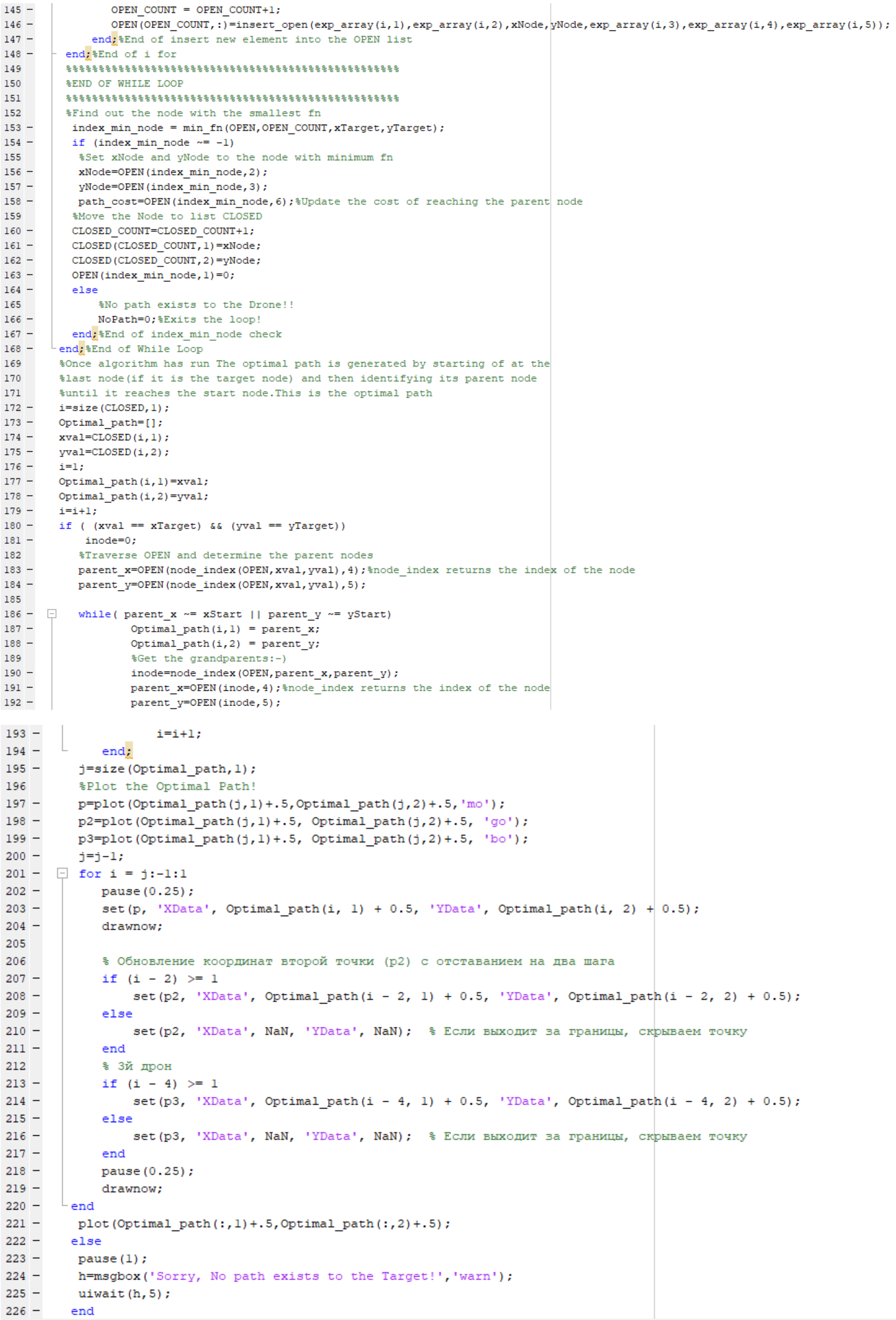
```
    xval=floor(xval);
    yval=floor(yval);
    MAP(xval,yval)=-1;%Put on the closed list as well
    plot(xval+.5,yval+.5,'ro');
 end%End of While loop

pause(1);
h=msgbox('Select the Drone formation initial position');
uiwait(h,5);
if ishandle(h) == 1
    delete(h);
end
xlabel('Select the Drone formation initial position using the Left Mouse button ','Color','black');
but=0;
while (but ~= 1) %Repeat until the Left button is not clicked
    [xval,yval,but]=ginput(1);
    xval=floor(xval);
    yval=floor(yval);
end
xStart=xval;%Starting Position ADD 1 MORE DRONE!!!!!!!!!!!!!!!!!!!!!!
yStart=yval;%Starting Position
MAP(xval,yval)=1;
 plot(xval+.5,yval+.5,'bo');
%End of obstacle-Drone pickup
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%LISTS USED FOR ALGORITHM
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%OPEN LIST STRUCTURE
%--------------------------------------------------------------------------
%IS ON LIST 1/0 |X val |Y val |Parent X val |Parent Y val |h(n) |g(n)|f(n)|
%--------------------------------------------------------------------------
OPEN=[];
%CLOSED LIST STRUCTURE
%--------------
%X val | Y val |
%--------------
% CLOSED=zeros(MAX_VAL,2);
CLOSED=[];
%Put all obstacles on the Closed list
k=1;%Dummy counter
for i=1:MAX_X
    for j=1:MAX_Y
        if(MAP(i,j) == -1)
            CLOSED(k,1)=i;
            CLOSED(k,2)=j;
            k=k+1;
        end
    end
```

Fig. 4.2 Second part of code



```
end
CLOSED_COUNT=size(CLOSED,1);
%set the starting node as the first node
xNode=xval;
yNode=yval;
OPEN_COUNT=1;
path_cost=0;
goal_distance=distance(xNode,yNode,xTarget,yTarget);
OPEN(OPEN_COUNT,:)=insert_open(xNode,yNode,xNode,yNode,path_cost,goal_distance,goal_distance);
OPEN(OPEN_COUNT,1)=0;
CLOSED_COUNT=CLOSED_COUNT+1;
CLOSED(CLOSED_COUNT,1)=xNode;
CLOSED(CLOSED_COUNT,2)=yNode;
NoPath=1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% START ALGORITHM
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
while((xNode ~= xTarget || yNode ~= yTarget) && NoPath == 1)
%  plot(xNode+.5,yNode+.5,'go');
 exp_array=expand_array(xNode,yNode,path_cost,xTarget,yTarget,CLOSED,MAX_X,MAX_Y);
 exp_count=size(exp_array,1);
 %UPDATE LIST OPEN WITH THE SUCCESSOR NODES
 %OPEN LIST FORMAT
 %--------------------------------------------------------------------------
 %IS ON LIST 1/0 |X val |Y val |Parent X val |Parent Y val |h(n) |g(n)|f(n)|
 %--------------------------------------------------------------------------
 %EXPANDED ARRAY FORMAT
 %--------------------------------
 %|X val |Y val ||h(n) |g(n)|f(n)|
 %--------------------------------
 for i=1:exp_count
    flag=0;
    for j=1:OPEN_COUNT
        if(exp_array(i,1) == OPEN(j,2) && exp_array(i,2) == OPEN(j,3) )
            OPEN(j,8)=min(OPEN(j,8),exp_array(i,5)); %#ok<*SAGROW>
            if OPEN(j,8)== exp_array(i,5)
                %UPDATE PARENTS,gn,hn
                OPEN(j,4)=xNode;
                OPEN(j,5)=yNode;
                OPEN(j,6)=exp_array(i,3);
                OPEN(j,7)=exp_array(i,4);
            end;%End of minimum fn check
            flag=1;
        end;%End of node check
%          if flag == 1
%              break;
    end;%End of j for
    if flag == 0
```

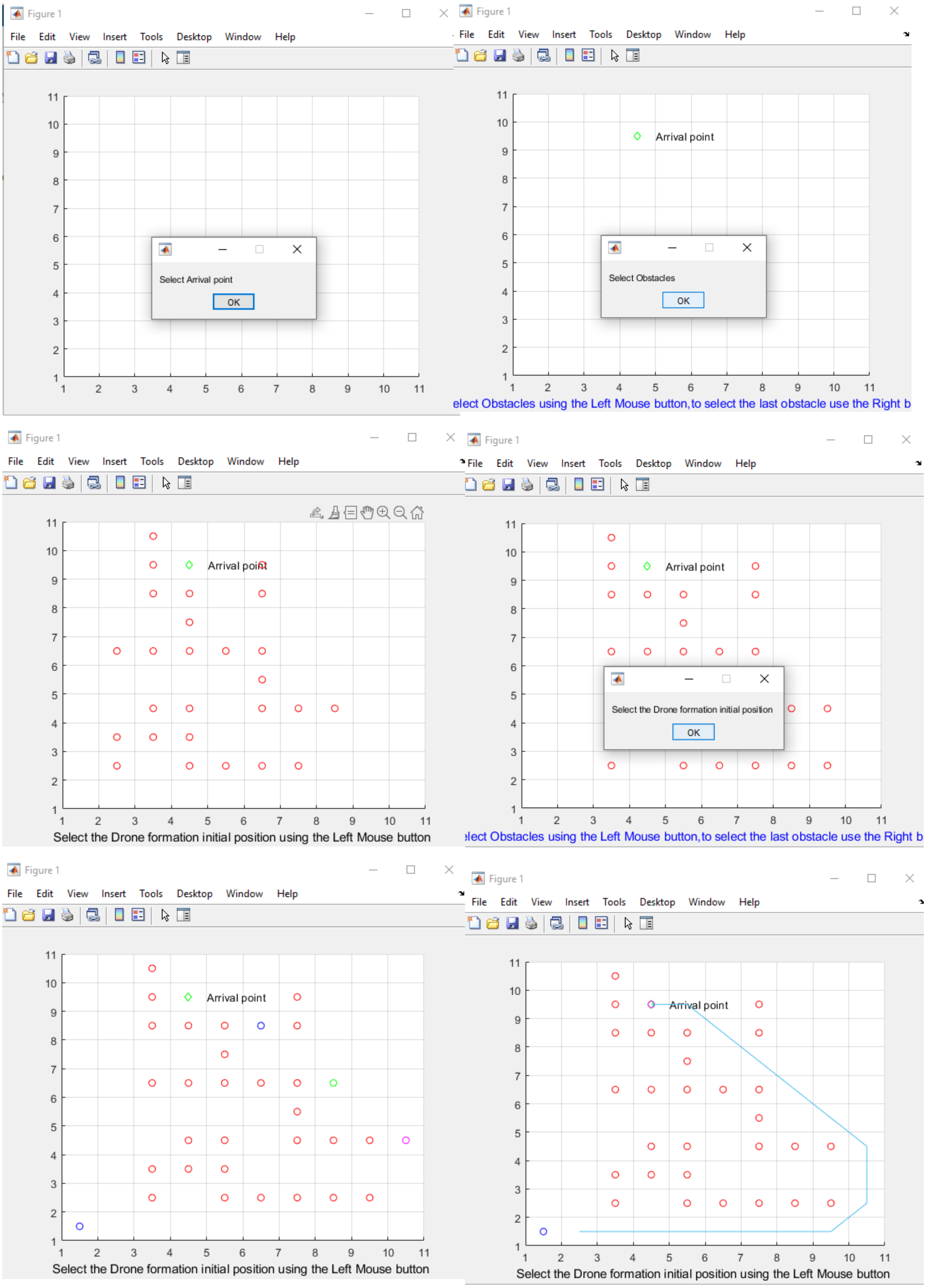
Fig. 4.3 Third part of code



```
        OPEN_COUNT = OPEN_COUNT+1;
        OPEN(OPEN_COUNT,:)=insert_open(exp_array(i,1),exp_array(i,2),xNode,yNode,exp_array(i,3),exp_array(i,4),exp_array(i,5));
     end;%End of insert new element into the OPEN list
 end;%End of i for
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %END OF WHILE LOOP
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %Find out the node with the smallest fn
  index_min_node = min_fn(OPEN,OPEN_COUNT,xTarget,yTarget);
  if (index_min_node ~= -1)
   %Set xNode and yNode to the node with minimum fn
   xNode=OPEN(index_min_node,2);
   yNode=OPEN(index_min_node,3);
   path_cost=OPEN(index_min_node,6);%Update the cost of reaching the parent node
  %Move the Node to list CLOSED
  CLOSED_COUNT=CLOSED_COUNT+1;
  CLOSED(CLOSED_COUNT,1)=xNode;
  CLOSED(CLOSED_COUNT,2)=yNode;
  OPEN(index_min_node,1)=0;
  else
      %No path exists to the Drone!!
      NoPath=0;%Exits the loop!
  end;%End of index_min_node check
end;%End of While Loop
%Once algorithm has run The optimal path is generated by starting of at the
%last node(if it is the target node) and then identifying its parent node
%until it reaches the start node.This is the optimal path
i=size(CLOSED,1);
Optimal_path=[];
xval=CLOSED(i,1);
yval=CLOSED(i,2);
i=1;
Optimal_path(i,1)=xval;
Optimal_path(i,2)=yval;
i=i+1;
if ( (xval == xTarget) && (yval == yTarget))
    inode=0;
   %Traverse OPEN and determine the parent nodes
   parent_x=OPEN(node_index(OPEN,xval,yval),4);%node_index returns the index of the node
   parent_y=OPEN(node_index(OPEN,xval,yval),5);

   while( parent_x ~= xStart || parent_y ~= yStart)
           Optimal_path(i,1) = parent_x;
           Optimal_path(i,2) = parent_y;
           %Get the grandparents:-)
           inode=node_index(OPEN,parent_x,parent_y);
           parent_x=OPEN(inode,4);%node_index returns the index of the node
           parent_y=OPEN(inode,5);
           i=i+1;
    end;
 j=size(Optimal_path,1);
 %Plot the Optimal Path!
 p=plot(Optimal_path(j,1)+.5,Optimal_path(j,2)+.5,'mo');
 p2=plot(Optimal_path(j,1)+.5, Optimal_path(j,2)+.5, 'go');
 p3=plot(Optimal_path(j,1)+.5, Optimal_path(j,2)+.5, 'bo');
 j=j-1;
 for i = j:-1:1
    pause(0.25);
    set(p, 'XData', Optimal_path(i, 1) + 0.5, 'YData', Optimal_path(i, 2) + 0.5);
    drawnow;

    % Обновление координат второй точки (p2) с отставанием на два шага
    if (i - 2) >= 1
        set(p2, 'XData', Optimal_path(i - 2, 1) + 0.5, 'YData', Optimal_path(i - 2, 2) + 0.5);
    else
        set(p2, 'XData', NaN, 'YData', NaN);  % Если выходит за границы, скрываем точку
    end
    % 3й дрон
    if (i - 4) >= 1
        set(p3, 'XData', Optimal_path(i - 4, 1) + 0.5, 'YData', Optimal_path(i - 4, 2) + 0.5);
    else
        set(p3, 'XData', NaN, 'YData', NaN);  % Если выходит за границы, скрываем точку
    end
    pause(0.25);
    drawnow;
end
 plot(Optimal_path(:,1)+.5,Optimal_path(:,2)+.5);
else
 pause(1);
 h=msgbox('Sorry, No path exists to the Target!','warn');
 uiwait(h,5);
end
```

Fig. 4.4 Fourth part of code

4.5 Practical demonstration



Select Arrival point

Select Obstacles

elect Obstacles using the Left Mouse button,to select the last obstacle use the Right b

Select the Drone formation initial position using the Left Mouse button

Select the Drone formation initial position

ılect Obstacles using the Left Mouse button,to select the last obstacle use the Right b

Select the Drone formation initial position using the Left Mouse button

Select the Drone formation initial position using the Left Mouse button

Fig. 4.5 Practical demonstration

4.6 Summary

This chapter provides an in-depth exploration of the simulation environment used for testing the collision avoidance system in drone formations. It outlines the key parameters, scenarios, and testing methodology, setting the stage for a thorough evaluation of the system's capabilities in complex and dynamic environments. Also, the shown code is flexible, it can be integrated into various drone designs. It is working and can change according to requirements.

CHAPTER 5

ENVIRONMENTAL PROTECTION THROUGH LIFE CYCLE ASSESSMENT OF PROJECT EQUIPMENT

5.1 Introduction: Understanding Life Cycle Assessment (LCA)

Environmental sustainability is a critical consideration in contemporary technological developments. Life Cycle Assessment (LCA) serves as a systematic method to evaluate the environmental impacts associated with a product, process, or system throughout its life cycle. In the context of the proposed project, which aims to prevent collisions of drones in formation, a comprehensive LCA of the essential equipment—Matlab, drones, and the PC—is imperative to ensure the overall environmental friendliness of the system.

In the contemporary landscape of technological innovation, the intersection of progress and environmental preservation stands as a critical challenge. As we navigate the complex realm of developing solutions, such as the drone collision avoidance system using Matlab, drones, and PCs, it becomes imperative to underscore the severity of environmental protection as a pressing concern.

Environmental Crisis Overview:

The planet is facing an unprecedented environmental crisis marked by climate change, biodiversity loss, and resource depletion. Human activities, including industrial processes and technological advancements, have accelerated these challenges, posing a direct threat to the delicate balance of ecosystems. The urgency of addressing environmental concerns has never been more pronounced.

Technological Contributions to Environmental Challenges:

While technological innovations have undeniably propelled societies forward, they have also been implicated in contributing to environmental degradation. The life cycle of electronic devices, such as those at the core of our drone collision avoidance system, involves resource extraction, manufacturing processes, energy consumption, and end-of-life disposal—all of which leave an ecological footprint.

Rising Environmental Consciousness:

In recent years, there has been a noticeable shift in global consciousness towards environmental sustainability. Stakeholders, including governments, businesses, and individuals, are increasingly recognizing the need for responsible practices that minimize the impact of technological advancements on the environment. Originality in project development must be coupled with an acute awareness of the broader environmental context.

The Crucial Role of Life Cycle Assessment:

Life Cycle Assessment (LCA) emerges as a powerful tool in the pursuit of sustainable technological solutions. It allows us to systematically analyze and quantify the environmental impact of a product or system throughout its entire life cycle. The inclusion of LCA in our project is not merely a checkbox for compliance; it signifies a commitment to acknowledging and addressing the profound environmental challenges that accompany technological progress.

Call to Action:

As developers and innovators, it is our responsibility to contribute to solutions that not only advance our technological capabilities but also safeguard the planet. The severity of the environmental problem demands that every project, including the drone collision avoidance system, be scrutinized through the lens of environmental protection. By doing so, we not only meet the criteria of originality but also fulfill our duty to create a future where technological progress harmonizes with ecological preservation. This chapter delves into the heart of this commitment, dissecting the environmental implications of each component in our system and providing actionable recommendations for a more sustainable future.

5.2 Equipment Description: Materials and Composition

The primary equipment for the drone collision avoidance system involves Matlab, drones, and the PC. In terms of materials, composite materials and aluminum alloys are commonly used. The choice of these materials is influenced by their lightweight nature, durability, and recyclability, aligning with environmental consciousness.

The fundamental equipment for the drone collision avoidance system—Matlab, drones, and the PC—each possesses unique characteristics in terms of materials and composition, contributing to their functionality and environmental impact.

Matlab:

Materials Used:

Matlab, being a software application, does not have a physical composition in the traditional sense. However, its operation depends on the underlying hardware, typically composed of metals, silicon-based components, and various electronic materials. The choice of these materials is guided by considerations such as conductivity, heat resistance, and overall durability.

Composition and Environmental Implications:

The hardware components supporting Matlab often consist of composite materials or aluminum alloys. The use of composite materials allows for a balance between structural integrity and reduced weight, contributing to energy efficiency during operation. However, the extraction and processing of metals for these components, such as aluminum, can result in notable environmental impacts. The production of aluminum, for instance, involves energy-intensive processes, contributing to greenhouse gas emissions.

Drones:

Materials Used:

Drones are intricate systems comprised of various materials. The frame is usually constructed from lightweight yet durable materials such as aluminum alloys or composite materials. Electronic components, including sensors, processors, and batteries, are integral to their composition. Plastics may also be used for non-structural elements to reduce weight.

Composition and Environmental Implications:

The use of aluminum alloys contributes to the lightweight design of drones, enhancing their energy efficiency during flight. Composite materials further aid in achieving a balance between strength and weight. However, the extraction of metals and the production of plastics have well-documented environmental impacts. Additionally, the incorporation of electronic components raises concerns about e-waste and the responsible disposal of batteries.

PC:

Materials Used:

Personal computers consist of a variety of materials, including metals for the chassis and internal components, plastics for the casing, and silicon-based materials for processors and memory.

Composition and Environmental Implications:

The choice of materials in PC construction considers factors such as conductivity, heat dissipation, and overall durability. Aluminum and steel are commonly used for structural components, contributing to the computer's robustness. The use of plastics helps reduce weight and manufacturing costs. However, the environmental impact arises during the extraction of metals and the production of plastics, contributing to resource depletion and emissions. The potential for e-waste, particularly with rapid advancements in technology, underscores the importance of recycling and responsible disposal practices.

In the quest for originality and environmental consciousness, exploring alternative materials with lower environmental footprints becomes essential. The careful selection of materials and composition aligns with the project's commitment to sustainability, ensuring that technological advancements are not achieved at the expense of environmental degradation.

5.3 Environmental Impact at Each Life Cycle Stage

Matlab:

- Raw Material Extraction:

  - Description: The initial stage involves the extraction of raw materials such as metals, silicon, and other components used in the production of hardware for Matlab. The mining process contributes to environmental disturbances, habitat destruction, and energy-intensive extraction methods.

- Environmental Impact: Resource depletion, soil erosion, water pollution, and energy consumption during extraction.

- Manufacturing:

- Description: The manufacturing phase encompasses the production of electronic components, circuit boards, and the assembly of hardware. It involves various energy-intensive processes such as molding, soldering, and etching.

- Environmental Impact: Greenhouse gas emissions, air pollution, and chemical waste generation during manufacturing.

- Use Phase:

- Description: The use phase for Matlab involves running algorithms and computations on a computer. The energy consumption during this phase depends on the complexity of the computations and the efficiency of the computer hardware.

- Environmental Impact: Carbon footprint associated with electricity consumption, contributing to greenhouse gas emissions.

- End-of-Life:

- Description: The end-of-life phase refers to the disposal or recycling of Matlab-related electronic components. E-waste management practices significantly influence the environmental impact at this stage.

- Environmental Impact: Improper disposal can lead to soil and water contamination due to hazardous materials in electronic components. Recycling helps mitigate these impacts by recovering valuable materials and reducing the need for new resource extraction.

Drones:

- Raw Material Extraction:

- Description: Drones are composed of various materials, including metals (aluminum, titanium), plastics, and electronic components. The extraction of these raw materials involves mining and processing.

- Environmental Impact: Similar to Matlab, the environmental impact includes habitat disruption, resource depletion, and energy consumption during extraction.

- Manufacturing:

- Description: The manufacturing process for drones includes assembly, fabrication of components, and integration of electronic systems. It contributes to energy consumption, waste generation, and emissions.

- Environmental Impact: Greenhouse gas emissions, air and water pollution, and waste generation during manufacturing.

- Use Phase:

- Description: During the use phase, drones consume energy for flight operations. This phase also considers the environmental impact of battery production and disposal.

- Environmental Impact: Carbon footprint from energy consumption, potential air pollution from battery components.

- End-of-Life:

- Description: Proper disposal and recycling of drones at the end of their life cycle are crucial. Recycling helps recover valuable materials and reduces the environmental impact of electronic waste.

- Environmental Impact: Inadequate disposal can lead to soil and water contamination. Proper recycling reduces the demand for new raw materials.

PC:

- Raw Material Extraction:

- Description: The production of computers involves the extraction of metals (copper, gold, aluminum) and other materials like plastics. Mining and processing are part of this stage.

- Environmental Impact: Similar to Matlab and drones, impacts include habitat disruption, resource depletion, and energy consumption during extraction.

- Manufacturing:

- Description: Manufacturing computers involves assembling components, soldering, and integrating electronic systems. This phase contributes to energy consumption, emissions, and waste.

- Environmental Impact: Greenhouse gas emissions, air and water pollution, and waste generation during manufacturing.

- Use Phase:

  - Description: The use phase for PCs includes energy consumption during data processing, algorithm execution, and general operation.
  - Environmental Impact: Carbon footprint associated with electricity consumption.

- End-of-Life:

  - Description: Responsible disposal or recycling of PCs is crucial to mitigate environmental impact. E-waste management practices significantly influence the overall impact at this stage.
  - Environmental Impact: Inadequate disposal can lead to soil and water contamination. Recycling reduces the demand for new raw materials and minimizes environmental pollution.

5.4 Comparison with Similar Products

A comparative analysis of the proposed equipment with similar products in terms of environmental impact reveals potential areas for improvement. This analysis considers factors such as energy efficiency, recyclability, and manufacturing processes. By identifying and addressing the environmental hotspots, the project can contribute to more sustainable technological advancements.

5.5 Recommendations for Limiting Exposure

To minimize the environmental impact of the drone collision avoidance system, several recommendations can be implemented:

- Material Selection: Explore alternative materials with lower environmental footprints.
- Energy Efficiency: Optimize algorithms and software to reduce energy consumption during operation.
- Recycling Programs: Establish mechanisms for the responsible disposal and recycling of equipment components.

- Continuous Improvement: Regularly update the system to incorporate the latest environmentally friendly technologies.

5.6 Conclusions

In conclusion, a thorough life cycle assessment of the essential equipment for the drone collision avoidance system highlights the environmental considerations at each stage. By understanding the materials, manufacturing processes, and usage patterns, this assessment provides a foundation for making informed decisions towards a more environmentally sustainable project. The comparison with similar products and the subsequent recommendations for limiting exposure underscore the commitment to originality and environmental responsibility in the development and implementation of the proposed system.

CHAPTER 6

LABOR PROTECTION

Introduction:

Labor protection is a crucial aspect in any workplace, ensuring the well-being and safety of workers. In the context of a worker involved in manufacturing drones, it's essential to understand and address potential harmful and hazardous factors to create a safe working environment.

6.1 Harmful and Hazardous Working Factors:

In the production of drones, workers may be exposed to various factors that can lead to health issues or accidents. These factors include:

- Chemical Exposure: Workers might encounter chemicals during the production process. It is essential to identify and list the specific chemicals used, emphasizing any potential health risks. Legislative requirements regarding the handling and storage of these chemicals should be outlined.
- Noise and Vibration: The manufacturing process of drones may involve noisy machinery or equipment that generates vibrations. Standards for permissible noise levels and protective measures, such as the use of personal protective equipment (PPE) like earplugs, should be detailed.
- Ergonomic Hazards: Prolonged periods of sitting or repetitive movements can lead to ergonomic issues. The employer must provide ergonomic workstations, and workers should adhere to recommended posture guidelines. Break schedules and exercises can be implemented to reduce the risk of musculoskeletal problems.

6.2 Analysis of Working Conditions and Development of Protective Measures:

Let's take chemical exposure as an example:

- Chemical Exposure:
  - Assessment of Levels: Identify the specific chemicals used, assess their concentration levels in the workplace, and determine potential exposure routes (inhalation, skin contact, etc.).

- Possible Harm: Detail potential health hazards associated with each chemical, considering acute and chronic effects. Provide information on symptoms and long-term consequences of exposure.
- Comparison with Normative Recommendations: Compare the observed chemical concentrations with established limits and Maximum Allowable Concentrations (MAC). Highlight any disparities and emphasize the importance of staying within recommended limits.
- Protective Measures:
    - Personal Protective Equipment (PPE): Specify the required PPE, such as gloves, goggles, or respirators, based on the nature of the chemicals.
    - Engineering Controls: Implement measures like proper ventilation systems or enclosed workspaces to minimize chemical exposure.
    - Training: Ensure that workers are adequately trained on handling chemicals, emergency procedures, and the proper use of protective equipment.

This approach ensures a comprehensive understanding of potential hazards, compliance with legislative requirements, and the development of effective protective measures in the context of drone manufacturing.

References

1. Kai Chang, Yuanqing Xia, Kaoli Huang. UAV formation control design with obstacle avoidance in dynamic three-dimensional environment // SpringerPlus 5. 2016. – № 1124. – Pp. 1- 16.

2. Jeonghoon Kwak, Yunsick Sung. Autonomous UAV Flight Control for GPS-Based Navigation // in IEEE Access. 2018. – Volume 6. – Pp. 37947- 37955.

3. Chin E. Lin, Ya-Hsien Lai. Quasi-ADS-B Based UAV Conflict Detection and Resolution to Manned Aircraft // Journal of Electrical and Computer Engineering. 2015. – Vol. 2015(3) – Pp. 1-12.

4. Vivek Agarwal, Saket Thakare, Akshay Jaiswal. Instant aircraft recovery during disaster using ADS-B technology // International Journal of Scientific & Engineering Research. 2015. – Volume 6, issue 8. – Pp. 1769-1772.

5. Constantino Gonc¸ Alves Ribeiro, Luciano Constantin Raptopoulos, Max Suell Dutra. A platform for autonomous path control of unmanned air-ship // Journal of Brazilian Society of Mechanical Sciences and Engineering. 2017. – Pp. 215–228.

6. Bin Fang, Tefang Chen. Research on UAV Collision Avoidance Strategy Considering Threat Levels // Practical Applications of Intelligent Systems. 2014. – Pp. 887-897. 20 10. Rafael G. Braga, Roberto C. da Silva, Alexandre C.B. Ramos, Felix Mora-Camino. Collision Avoidance Based on Reynolds Rules: A Case Study Using Quadrotors // In Information Technology – New Generations, Advances in Intelligent Systems and Computing, S. Latifi, Eds. Springer Interna-tional Publishing AG. 2018. – Pp. 773-780.

7. Antonov V. O., Gurchinsky M. M., Petrenko V. I., Tebueva F. B. Metod planirovaniya trayektorii dvizheniya tochki v prostranstve s prepyatstviyem na osnove iterativnoy kusochnolineynoy approksimatsii [A method for planning the trajectory of a point in space with an obstacle based on iterative piecewise linear approximation] [Electronic resource] // Sistemy upravleniya, svyazi i bezopasnosti [Control, communication and security systems], 2018. Available at:

http://sccs.intelgr.com/archive/2018-01/09-Antonov.pdf, free (Access date: September 22, 2019).

8. Azade Fotouhi, Ming Ding, Mahbub Hassan. DroneCells: Improving 5G Spectral Efficiency using Drone-mounted Flying Base Stations // Journal of Transactions on mobile computing. 2017. – Pp. 1-14

APPENDIX A

A_Star1.m

There is an explanation of the tapes in the comments.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%DEFINE THE 2-D MAP ARRAY
MAX_X=10;
MAX_Y=10;
MAX_VAL=10;
%This array stores the coordinates of the map and the
%Objects in each coordinate
MAP=2*(ones(MAX_X,MAX_Y));
% Obtain Obstacle, Target and 2nd Drone
% Initialize the MAP with input values
% Obstacle=-1,Drone1 = 0,Drone2=1,Space=2
j=0;
x_val = 1;
y_val = 1;
axis([1 MAX_X+1 1 MAX_Y+1])
grid on;
hold on;
n=0;%Number of Obstacles
% BEGIN Interactive Obstacle,1st Drone, Start Location selection
pause(1);
h=msgbox('Select Arrival point');
uiwait(h,5);
if ishandle(h) == 1
    delete(h);
end
```

```
xlabel('Select Arrival point use the Left button','Color','black');
but=0;
while (but ~= 1) %Repeat until the Left button is not clicked
    [xval,yval,but]=ginput(1);
end
xval=floor(xval);
yval=floor(yval);
xTarget=xval;%X Coordinate of the collision avoidance point
yTarget=yval;%Y Coordinate of the collision avoidance point
xTarget2=xval;%FOR 2ND DRONE
yTarget2=yval;
MAP(xval,yval)=0;%Initialize MAP with location of the collision avoidance point
plot(xval+.5,yval+.5,'gd');
text(xval+1,yval+.5,'Arrival point')
pause(2);
h=msgbox('Select Obstacles');
  xlabel('Select Obstacles using the Left Mouse button,to select the last obstacle use the Right button','Color','blue');
uiwait(h,10);
if ishandle(h) == 1
    delete(h);
end
while but == 1
    [xval,yval,but] = ginput(1);
    xval=floor(xval);
    yval=floor(yval);
    MAP(xval,yval)=-1;%Put on the closed list as well
    plot(xval+.5,yval+.5,'ro');
 end%End of While loop

pause(1);
h=msgbox('Select the Drone formation initial position');
uiwait(h,5);
if ishandle(h) == 1
    delete(h);
```

```
end
xlabel('Select the Drone formation initial position using the Left Mouse button ','Color','black');
but=0;
while (but ~= 1) %Repeat until the Left button is not clicked
    [xval,yval,but]=ginput(1);
    xval=floor(xval);
    yval=floor(yval);
end
xStart=xval;%Starting Position ADD 1 MORE DRONE!!!!!!!!!!!!!!!!!!!!!!
yStart=yval;%Starting Position
MAP(xval,yval)=1;
 plot(xval+.5,yval+.5,'bo');
%End of obstacle-Drone pickup
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%LISTS USED FOR ALGORITHM
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%OPEN LIST STRUCTURE
%--------------------------------------------------------------------------
%IS ON LIST 1/0 |X val |Y val |Parent X val |Parent Y val |h(n) |g(n)|f(n)|
%--------------------------------------------------------------------------
OPEN=[];
%CLOSED LIST STRUCTURE
%--------------
%X val | Y val |
%--------------
% CLOSED=zeros(MAX_VAL,2);
CLOSED=[];
%Put all obstacles on the Closed list
k=1;%Dummy counter
for i=1:MAX_X
    for j=1:MAX_Y
        if(MAP(i,j) == -1)
            CLOSED(k,1)=i;
```

```
      CLOSED(k,2)=j;
      k=k+1;
    end
  end
end
CLOSED_COUNT=size(CLOSED,1);
%set the starting node as the first node
xNode=xval;
yNode=yval;
OPEN_COUNT=1;
path_cost=0;
goal_distance=distance(xNode,yNode,xTarget,yTarget);
OPEN(OPEN_COUNT,:)=insert_open(xNode,yNode,xNode,yNode,path_cost,goal_distance,goal_distance);
OPEN(OPEN_COUNT,1)=0;
CLOSED_COUNT=CLOSED_COUNT+1;
CLOSED(CLOSED_COUNT,1)=xNode;
CLOSED(CLOSED_COUNT,2)=yNode;
NoPath=1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% START ALGORITHM
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
while((xNode ~= xTarget || yNode ~= yTarget) && NoPath == 1)
%  plot(xNode+.5,yNode+.5,'go');
 exp_array=expand_array(xNode,yNode,path_cost,xTarget,yTarget,CLOSED,MAX_X,MAX_Y);
 exp_count=size(exp_array,1);
 %UPDATE LIST OPEN WITH THE SUCCESSOR NODES
 %OPEN LIST FORMAT
 %-----------------------------------------------------------------------
 %IS ON LIST 1/0 |X val |Y val |Parent X val |Parent Y val |h(n) |g(n)|f(n)|
 %-----------------------------------------------------------------------
 %EXPANDED ARRAY FORMAT
 %-------------------------------
```

```
            %|X val |Y val ||h(n) |g(n)|f(n)|
            %-------------------------------
            for i=1:exp_count
               flag=0;
               for j=1:OPEN_COUNT
                   if(exp_array(i,1) == OPEN(j,2) && exp_array(i,2) == OPEN(j,3) )
                       OPEN(j,8)=min(OPEN(j,8),exp_array(i,5)); %#ok<*SAGROW>
                       if OPEN(j,8)== exp_array(i,5)
                           %UPDATE PARENTS,gn,hn
                           OPEN(j,4)=xNode;
                           OPEN(j,5)=yNode;
                           OPEN(j,6)=exp_array(i,3);
                           OPEN(j,7)=exp_array(i,4);
                       end;%End of minimum fn check
                       flag=1;
                   end;%End of node check
           %        if flag == 1
           %            break;
               end;%End of j for
               if flag == 0
                   OPEN_COUNT = OPEN_COUNT+1;

OPEN(OPEN_COUNT,:)=insert_open(exp_array(i,1),exp_array(i,2),xNode,yNode,exp_array(i,3),exp_ar
ray(i,4),exp_array(i,5));
                end;%End of insert new element into the OPEN list
            end;%End of i for

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            %END OF WHILE LOOP

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            %Find out the node with the smallest fn
             index_min_node = min_fn(OPEN,OPEN_COUNT,xTarget,yTarget);
             if (index_min_node ~= -1)
              %Set xNode and yNode to the node with minimum fn
```

```
   xNode=OPEN(index_min_node,2);
   yNode=OPEN(index_min_node,3);
   path_cost=OPEN(index_min_node,6);%Update the cost of reaching the parent node
  %Move the Node to list CLOSED
  CLOSED_COUNT=CLOSED_COUNT+1;
  CLOSED(CLOSED_COUNT,1)=xNode;
  CLOSED(CLOSED_COUNT,2)=yNode;
  OPEN(index_min_node,1)=0;
  else
      %No path exists to the Drone!!
      NoPath=0;%Exits the loop!
  end;%End of index_min_node check
end;%End of While Loop
%Once algorithm has run The optimal path is generated by starting of at the
%last node(if it is the target node) and then identifying its parent node
%until it reaches the start node.This is the optimal path
i=size(CLOSED,1);
Optimal_path=[];
xval=CLOSED(i,1);
yval=CLOSED(i,2);
i=1;
Optimal_path(i,1)=xval;
Optimal_path(i,2)=yval;
i=i+1;
if ( (xval == xTarget) && (yval == yTarget))
    inode=0;
   %Traverse OPEN and determine the parent nodes
   parent_x=OPEN(node_index(OPEN,xval,yval),4);%node_index returns the index of the node
   parent_y=OPEN(node_index(OPEN,xval,yval),5);

   while( parent_x ~= xStart || parent_y ~= yStart)
           Optimal_path(i,1) = parent_x;
           Optimal_path(i,2) = parent_y;
           %Get the grandparents:-)
           inode=node_index(OPEN,parent_x,parent_y);
```

```
        parent_x=OPEN(inode,4);%node_index returns the index of the node
        parent_y=OPEN(inode,5);
        i=i+1;
   end;
 j=size(Optimal_path,1);
 %Plot the Optimal Path!
 p=plot(Optimal_path(j,1)+.5,Optimal_path(j,2)+.5,'mo');
 p2=plot(Optimal_path(j,1)+.5, Optimal_path(j,2)+.5, 'go');
 p3=plot(Optimal_path(j,1)+.5, Optimal_path(j,2)+.5, 'bo');
 j=j-1;
 for i = j:-1:1
   pause(0.25);
   set(p, 'XData', Optimal_path(i, 1) + 0.5, 'YData', Optimal_path(i, 2) + 0.5);
   drawnow;

   % Обновление координат второй точки (p2) с отставанием на два шага
   if (i - 2) >= 1
      set(p2, 'XData', Optimal_path(i - 2, 1) + 0.5, 'YData', Optimal_path(i - 2, 2) + 0.5);
   else
      set(p2, 'XData', NaN, 'YData', NaN);  % Если выходит за границы, скрываем точку
   end
   % 3й дрон
   if (i - 4) >= 1
      set(p3, 'XData', Optimal_path(i - 4, 1) + 0.5, 'YData', Optimal_path(i - 4, 2) + 0.5);
   else
      set(p3, 'XData', NaN, 'YData', NaN);  % Если выходит за границы, скрываем точку
   end
   pause(0.25);
   drawnow;
end
 plot(Optimal_path(:,1)+.5,Optimal_path(:,2)+.5);
else
 pause(1);
 h=msgbox('Sorry, No path exists to the Target!','warn');
 uiwait(h,5);
```

```
end
```