THE ALGERIAN DEMOCRATIC AND POPULAR REPUBLIC
MINISTER FOR HIGHER EDUCATION AND RESEARCH
BLIDA -1- UNIVERSITY

Thesis To Obtain The Degree of Doctor
-DLMD-

Awarded by: Aeronautical and Spatial Studies Institute

Specialty: Aeronautics

Defended by: Khiereddine Choutri

k.choutri@gmail.com

Multi-Agents Quadrotors UAVs For Search And Rescue Operations

Defended on 7th July 2019 in front of a jury composed by:

Abderrezak Guessoum	Prof	Blida-1- University	Reviewer
Allel Hadjali	Prof	ENSMA, Poitier	Examiner
Salah Boukraa	Prof	Blida-1- University	President
Mohand Lagha	Prof	Blida-1- University	Director
Laurent Dala	Prof	Northumbria University	Co-Director

Abstract

This thesis addresses the problems of using a multi-agents quadrotors UAVs for search and rescue operations.

In the first part the quadrotors components are modeled, and the interaction between them as long as the physical limitations is taken into consideration. The effects of the environmental constraints represented by a wind gust model were also examined. For the optimal trajectory generation and control problem, a new scheme based on a multi-layer optimal system is proposed. The designed controller aims to improve the robustness, the performances, and energy optimization.

Once the quadrotors is modeled and controlled, the formation control problem is then investigated. A double loop control structure based on a non-linear optimal backstepping controller is applied. The quadrotors UAVs have to follow the desired path, hold their formation, avoid the obstacles and eliminate the effects of the external wind disturbances.

In the other hand, SAR operations using UAVs are often time critical and dangerous missions. The challenge behind this work is that a single quadrotors could track the SAR strategies pattern, search for survivals with high probability of detection and then rescue them.

Finally this thesis presents the development of a quadrotors multi-agents system, in which a swarm of agents coordinate and collaborate their actions in order to search for survivals in a disaster area and rescue them. Furthermore this work proposes new searching strategies that speed up the searching process, and cover the integrity of the searching area in order to detect all the possible targets.

ملخص

تتناول هذه الأطروحة مشكلة استخدام مجموعة الطائرات بدون طيار في عمليات البحث والإنقاذ

في الجزء الأول، يُقترح أسلوب جديد لتصميم المكونات المختلفة للدرون (البطارية ، المحرك ، الدوار ... إلخ). تؤخذ جميع التفاعلات بين مكونات والقيود المادية في الاعتبار. وتناقش أيضا القيود البيئية التي يمثلها عاصفة من نموذج الرياح.

لتوليد مسار والسيطرة على الدرون، تقترح خطة جديدة تقوم على نظام متعدد الطبقات الأمثل. ويتم توليد المسار المنحنى بطريقة مثلى بين نقطة البداية ونقطة النهاية. في حالة وجود عقبات، يعاد إنشاء المسار لتجنب الاصطدام. فيما يتعلق الطيار الآلي، فهو مصمم يستند إلى تحكم الذي يهدف إلى تحسين الأداء، المتانة واقتصاد الطاقة في نفس الوقت. يتم إجراء مقارنة مع وحدات تحكم المثلى أخرى لإثبات فعالية وحدة التحكم المقترحة.

علاوة على ذلك، عمليات البحث والإنقاذ باستخدام الطائرات بدون طيار غالبا ما تكون حرجة وخطيرة. لهذا الغرض، هذا العمل يقدم استراتيجيات البحث والاكتشاف التي تجمع بين طرق البحث التقليدية مع الناجين كشف الاحتمالات. وتهدف الإستراتيجية المقترحة لتسريع عملية البحث وتغطية سلامة منطقة البحث للكشف عن الأهداف المحتملة.

وأخيرا، تقدم هذه الأطروحة تطوير نظام متعدد وكيل مستقل بحتة فيه مجموعة من الطائرات بدون طيار تنسق وتتعاون فيما بينها للبحث عن الناجين في المنطقة المنكوبة ومساعدتهم.

Acknowledgements

In the name of Allah, the most beneficent and most merciful. We thank Allah for all his blessing and strength that he gives us in completing this work. I would like to express my deepest gratitude to my doctorate thesis supervisor, Prof. Mohand Lagha for his invaluable advices and guidance throughout this dissertation. His profound knowledge, ideas and support kept on motivating me to give me all for this work.

Special thanks to the great teacher Prof. Laurent Dala who helped and encouraged me a lot to pursue my work till this step.

I would like to thank my mom and dad for their continuous support, for always pushing me forward and for providing me with the calm environment to work in.

I wish to thank all jury members, my big family, my friends and those who directly or indirectly guided and helped me in this project. The knowledge and support that they shared with me will always be remembered.

To the soul of my grandfather, words cannot express how thankful I am for all what you have done.

Last but not least, I wish to dedicate my appreciation to my wife, I thank her for her infinite patience, love and for always being there for me all these years. Thanks for her unconditional love, encouragement, and support.

Contents

List of Figures

List of Tables

List of Symbols

A	list of N agents
A_g	ground sensing areas
c_D	drag constant
c_T	thrust constant
d	agents inter-distance
e	back EMF
E_b	body frame
E_i	inertial frame
F	external forces
F_t	lift force
F_d	drag force
F_g	gravity force
F_t	The propeller thrust
F^B	body frame
F^I	inertial frame
g	the gravity vector
h	height
i	armature current
I	team of quadrotors
$I_{3\times3}$	identity matrix
J	inertia matrix
J_r	rotor inertia
K_e	electrical motor constant

K_f	friction constant
K_t	torque constant
K_d	drug constant
l	distance from the center of mass to the motor axis of action
L_{xx} , L_{yy} and L_{zz}	turbulence lengths
L_w	winding inductance
L	graph Laplacian matrix
m	mass
n	set of agents
N	Number of agents
p	position
Ω	angular speeds vector
P_i	departure point
P_f	arrival point
$P(T)$	list of N paths
q	quaternion
$\bar{q}$	complex part of q
$q(t)$	scalar part of q
q^*	quaternion conjugate
$q_{13\times}$	skew-symmetric matrix
r	radius of the propeller
R_w	winding resistance
R_a	radius of searching area
$R(t)$	rotational matrix
R_q	rotation matrix
S_w	reference area
t	time
T	list of N topology
T_e	electrical torque
T_L	mechanical load
T_e	electrical torque
u	rotation axis
U	designed control inputs
V	linear speed
V_s	DC voltage source
W_0^*	circular space magnitude of the considered wind
Y	vector of batter performances
α	angle of rotation
α_h	false alarm probability
β_h	missed detection probability
$\delta_{(x,y,z)}(v,z,t)$	stochastic components of the considered wind
Δt	time step
η	Euler angles derivatives vector

G	connected graph
$\widetilde{G}$	normalized interaction matrix
ν	node set
$\widehat{n}$	rotation vector
ω_i	angular velocity
ω	circular space frequency of the considered wind
ϕ	roll
ψ	yaw
ρ	the air density
σ	standard deviations
θ	pitch
τ	torque
τ_{ext}	external torques
τ_u	inputs torques

Acronyms

CLS	Creeping line Search
CS	Contour Search
DAI	Distributed Artificial Intelligence
ESC	Expanding Square Search
IAE	Integral Absolute Error
$IAMSAR$	International Aeronautical and Maritime Search and Rescue
$L-F$	Leader-Follower
LKP	Last Knowing Position
LQR	Linear Quadratic Regulator
$MO-GA$	Multi-Objective Genetic Algorithm
OBS	Optimal Backstepping
PT	Parallel Track
SAR	Search and Rescue
UAV	Unmanned Aerial Vehicle
$VTOL$	Vertical Take-off and Landing

Chapter

1

Introduction

Contents

1.1 General Introduction

Over the past several decades, a growing interest has been shown in robotics. In fact, several industries (automotive, medical, manufacturing, space, ...) require robots to replace men in dangerous, boring or onerous situations. A wide area of this research is dedicated to aerial platforms.

The potential applications of the quadrotors have attracted the attention of researchers in the last decade. The cooperation of multiple quadrotors is promising in order to accomplish complex tasks that are impossible to be completed by a single quadrotor. In this thesis, the cooperation of quadrotors under a formation control is especially considered in the aspect of search and rescue operation.

This thesis presents the development of a quadrotors multi-agents system, in which a swarm of agents coordinate and collaborate their actions in order to search for survivals in a disaster area and rescue them. Furthermore this work proposes new searching strategies that speed up the searching process, and cover the integrity of the searching area in order to detect all the possible targets.

SAR operations using UAVs are often time critical and dangerous missions. The challenge behind this work, is that all the UAVs agents track their search pattern, keep the formation topology and avoid the collusion between them and with the external obstacles. Task allocation and election are also important colloabarative aspects that the swarm should have. This will give them the ability to accomplish the mission in the case of losing one of the agents.

Finally a scenario of SAR mission is studied for fist time with one quadrotors to test the ability of a VTOL UAV to accomplish such king of mission, then with a quadrotors swarm in both centralized and decentralized architecture. The obtained results are compared and the desired goal is reached.

1.2 Unmanned Aerial Vehicle

An unmanned aerial vehicle (also known as a drone) refers to a pilotless aircraft, a flying machine without an onboard human pilot or passengers. As such, "unmanned" implies total absence of a human who directs and actively pilots the aircraft. Control functions for unmanned aircraft may be either onboard or off-board (remote control). The term UAV or unmanned aerial vehicle has been used for several years to describe unmanned aerial systems. Various definitions have been proposed for this term, like:

Definition 1. *"A reusable aircraft designed to operate without an onboard pilot. It does not carry passengers and can be either remotely piloted or preprogrammed to fly autonomously". Joint Capability Group on Unmanned Aerial Vehicles (2007).*

In the definition above, the characterization reusable is used to differentiate unmanned aircraft from guided weapons and other munition delivery systems.

A few years ago, the U.S. Department of Defense (DoD), followed by the FAA and the European Aviation Safety Agency (EASA), adopted the term UAS or Unmanned Aircraft System. This was meant to signify that UAS are aircraft and as such airworthiness will need to be demonstrated, and they are also systems consisting of ground control stations, communication links, and launch and retrieval systems in addition to the aircraft itself. The FAA has defined an Unmanned Aircraft or UA as:

Definition 2. *"A device used or intended to be used for flight in the air that has no onboard pilot. This includes all classes of airplanes, helicopters, airships, and translational lift aircraft that have no onboard pilot. Unmanned aircraft are understood to include only those aircraft controllable in three axes and therefore, exclude traditional balloons." Federal Aviation Administration (2008)*

From the above definition it's clearly that a UAV is merely one component/subsystem of a distributed system. Thus the correct term when describing the whole system is Unmanned Aerial System (UAS), which has been acknowledged by most organizations and among the scientific community during the last years.

An over simplistic view of an Unmanned Aerial System (UAS) may be regarded as a system that consists of three major components:

1. **Ground**: such as the Ground Control station (GCS), Ground Data Terminal (GDT). Ground Control Station being the most important of this subsystem.
2. **Communication**: such as control, command and Payload data link.
3. **Aerial**: the aerial platform (UAV) and the payloads for its purpose, such as sensors and cameras.

1.3 The Era of UAV

Although there are many who believe that UAVs are a recent invention going back at most two or three decades, unmanned flight has a rich history that goes back all the way to ancient times. Of course, the first systems that can qualify with the modern definition of UAVs are quite recent and mainly involve the reconnaissance drones developed and deployed during the cold war.

In 1917 Dr. Cooper and Elmer Sperry invented the automatic gyroscopic stabilizer, which helps to keep an aircraft flying straight and level. This technology was used to convert a U.S. Navy Curtiss N-9 trainer aircraft into the first radio-controlled Unmanned Aerial Vehicle (UAV). The first UAVs were tested in the US during World War I but never deployed in combat. During World War II, Germany took a serious advantage and demonstrated the potential of UAVs on the battle-

fields. After the two wars, the military recognized the potential of UAVs in combat and started development programs which led, few decades after, to sophisticated systems, especially in the US , like the Predator (General Atomics) or the Pioneer (PUAV).

The first unmanned helicopter was the one built by Forlanini in 1877. It was neither actively stabilized nor steerable. With the outstanding technological advancements after WW II it became possible to build and control unmanned helicopters. The company Gyrodyne of America started the famous DASH program for the navy. The military market of unmanned helicopters became evident. An intensive research effort was deployed and impressive results achieved; like the A160 Hummingbird, a long endurance helicopter able to fly 24 h within a range of 3150 km.

The battlefield of the future would belong to the Unmanned Combat Armed Rotorcraft (see Figure 1.1), if DARPA decides one day to restart this ambitious project. The academic researchers have also shown their interest in the development of autonomous helicopters over the last decades.

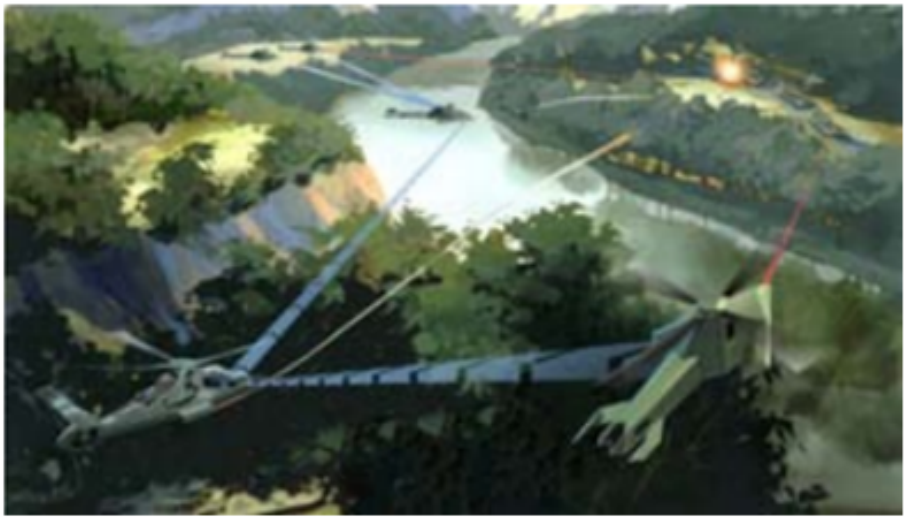

Figure 1.1: UCAR concept

An extensive research effort is being conducted on VTOL UAVs and MAVs, the ductfan design (e.g., the iStar MAV, the Sikorsky Cypher, and the SELEX Galileo Spyball), counterrotating rotors (e.g., the IT-180 and the KOAX X-240), as well as mixed designs like the Eagle Eye (Figure 1.2) and X-50 that combine some of the advantages of fixed-wing and helicopter designs. Finally, the CyberQuad and the AirRobot AR 100-B (Figure 1.3) are both examples of the quad-rotor design that is especially popular in academic environments.

A modern UAV is an unmanned aerial vehicle that is equipped with data processor units, sensors, automatic control, and communication systems and capable of performing autonomous flight missions without any interaction with a human pilot.

Figure 1.2: The Eagle Eye is a tilt-rotor UAV

1.4 UAVs Classification

There is currently a lack of universal classification standard for UAVs and a wide variety of organizations, research communities, governments and industries have employed different terms and scaling. To address this issue, the leading recommendation is that the UAV classifications can be emerged into the already established criteria's and classifications that are set for manned aircraft. For a wider coverage of this subject the reader is referred to [3].

Figure 1.3: The AirRobot AR 100-B

The wide varieties of classification of UAVs extend from grouping the aircrafts into different classes based on one or several of the following characteristics:

1. **Mass/Maximum take-off weight (MTOW)**

2. **Operational altitude**
3. **Range**
4. **Endurance**
5. **Speed**
6. **The maximum kinetic energy**: which is a function of the speed and weight of the aircraft and is essentially a classification based on impact damage.

UAV Category	Range (km)	Flight Atitude (m)	Endurance (hours)	MTOW (kg)
Nan* (NAV)	<1	100	<1	<0.025
Micro (MAV)	<10	250	1	<5
Mini (MUAV)	<10	150-300	<2	<30
Close Range*	10-30	3000	2-4	150
Tacticial (TUAV)	<200	300-5000	5-10	15-500
Medium Altitude-Long Endurance (MALE)	>500	5000-15000	10-48	1500-7000
High Altitude-Long Endurance (MALE)	>20 000	>15000	10-48	4500-15000

Table 1.1: UAVs Classification

Another approach is to group the UAVs into five categories, two more categories has since been added and are marked with a (*) in Table 1.1 These categories are to collect UAVs of common size and operating characteristics into a specific group/class of UAVs as listed in Table 1.1 The operating features in Table 1.1 are compiled from several sources in order to provide an overall view of capabilities of the different UAV classes [4]-[5].

1.5 UAVs Configurations

The conventional UAV platforms are normally classified into three main categories: Fixed Wing, Rotary Wing and Flapping Wing.

In the civilian field, fixed-wing UAVs are most used for long-distance, long-range and high-altitude missions. Generally, they are useful in scientific applications such as meteorological reconnaissance and environmental monitoring. Due to its usage in the military field, it is equipped with similar systems in order to achieve its navigation tasks [2, 6].

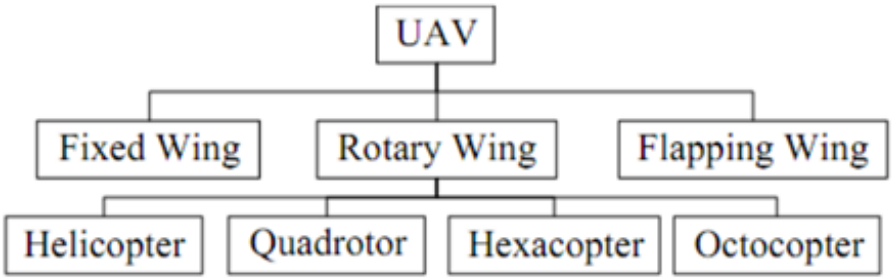

Figure 1.4: UAV Configurations

Flapping-wing UAVs try to duplicate the way birds or insects fly. Most of them are still under development. Belonging to the class of micro UAVs, they have extremely low payload capability and low endurance. Among their interesting characteristics it can be mentioned that the flapping wing UAVs have low power consumption and can perform vertical take-off and landing [4-9].

Rotary wings UAVs, which are also called Vertical Take-off and Landing (VTOL) rotorcrafts, are normally used on missions that require hovering flight. Rotary wing aircrafts are also less susceptible to air turbulance compared with fixed wing aircrafts of similar dimensions [10]. The common example of rotary wing UAVs are helicopters which are a two-rotor aircraft with a main rotor giving the thrust and an anti-torque tail rotor. Another example of rotary wing aircrafts could be quadrotors.

A comparison between flapping-wing, fixed wing and rotary wing are available in Table 1.2 In this comparison characteristics of rotary wings are scored based on the quadrotor type. [11-13]

References	**Rotary wing**	**Fixed wing**	**Flapping wing**
Maneuver	$\oplus$	$\odot$	$\ominus$
Cost	$\ominus$	$\odot$	$\oplus$
Construction and repairing	$\ominus$	$\odot$	$\oplus$
Civilian Application	$\oplus$	$\odot$	$\oplus$
Military Application	$\ominus$	$\ominus$	$\ominus$
Energy Consumption	$\odot$	$\ominus$	$\oplus$
Flight safety	$\ominus$	$\oplus$	$\odot$
Range	$\ominus$	$\oplus$	$\odot$

Table 1.2: Comparison Between UAVs

1.5.1 Why Quadrotors ?

The low cost and simplicity of the quadrotor design provides an excellent testing ground for application of advanced system identification and control techniques which are ideally suited for research institutes and universities with small budgets. Prior to the design and build of the quadrotor, it is essential that a form of system identification is performed in order to ascertain the correct powerplant parameters. Identified powerplant parameters allow for simulations to be carried out, analyzed and improved prior to costly construction.

A quadrotor has apparent advantages over other rotary wing UAVs. It is mechanically simple and is controlled by only changing the speed of rotation for the four motors.

Level	**Decisions/Actions of the Automated system**	**Authority**
10	Makes all descisions, humain operator not informed	Completely autonomous
9	Decides whether to inform the human operator of decisions and actions taken	Completely autonomous
8	Informs the human operator about decisions and actions taken only if requested	Completely autonomous
7	Executes automatically, but the human operator is continuously informed about decisions and actions taken	Completely autonomous
6	Informs the humain operator and executed automatically if the human operator does not disapprove within a restricted time	Autonomous with consent
5	Selects actions, but does not execute until approved by human operator	Semi-Autonomous
4	Suggests one alternative action to the human operator decides whether to reject or approve the syggested alternative	Advisory
3	Narrows down the umber of alternative actions for a fex from which the human operator decides which alternative to be executed	Advisory
2	Complete set of alternative actions is presented from which the human operator decides which alternative is to be executed	Advisory
1	None, the human operator has to make all decisions and actions	None

Table 1.3: UAV Autonomy

The low cost and simplicity means the quadrotor provides an excellent testing ground for the application of advanced control techniques on UAV's. However, the technical challenges for small rotary-wing UAV systems are numerous. High thrust-to-weight ratios are necessary for the propulsion system. An endurance long enough to perform a meaningful mission is the highest priority. In order to achieve this goal a careful matching of aircraft mass, batteries, electric motors, and rotors is imperative. The absence of a method to measure the parameters of interest that allow for a careful pairing of hardware is a problem that needs to be addressed. This is an essential prerequisite for optimal simulation and cost effective testing.

1.6 UAV Autonomy

The autonomy of a UAV may be described by a 10-point scale as illustrated in Table 1.3

The scaling is based on the assistance/workload the automated system requires from a human operator to decide and perform a certain task, i.e. the ability of the system in terms of decision making and authority.

1.7 Application of UAVs

UAVs have predominantly been used for military purposes, however as UAVs have become more affordable so has the interest into the potentials of UAVs for civilian purposes. This has led to a broad set of literature, addressing issues such as the absence of regulations, classification as well as potential civilian applications [20]-[22]. In Figure 1.5 some of the potential applications are displayed in order to present the broad-spectrum of civilian UAV applications.

To this date, UAV applications in "actual situations" have been restricted to single UAVs and most UAV swarm research is conducted by universities and research institutes. The experiments and projects done by researchers are to replicate the "actual situations" either in a smaller scale or more commonly in simulators. The ideal scenarios for application of UAV swarms are however the same as for single-UAVs, above all when a broader area has to be covered during a short period of time. For example search and rescue missions, forest/bushfires and communication setup after a disaster.

1.8 Multi-UAV

1.8.1 Why Multi-UAV?

The advantages of using multiple UAVs when comparing to a single powerful one can be categorized as follows:

Figure 1.5: UAV Applications

1. **Multiple simultaneous interventions**: A single autonomous vehicle is limited at any one time to sense or actuate in a single point. However, the components of a team can simultaneously collect information from multiple locations and exploit the information derived from multiple disparate points to build models that can be used to take decisions. Moreover, multiple UAVs can apply simultaneously forces at different locations to perform actions that could be very difficult for a single UAV.

2. **Greater efficiency**: The execution time of missions such as exploration and searching for targets can be decreased when using simultaneously multiple vehicles.

3. **Complementarities of team members**: Having a team with multiple heterogeneous vehicles offers additional advantages due to the possibility of exploiting their complementarities.

4. **Reliability**: The multi-UAV approach leads to redundant solutions offering greater fault tolerance and flexibility including reconfigurability in case of failures of individual vehicles.

5. **Technology evolution** : The development of small, relatively low-cost UAVs is fuelled by the progress of embedded systems together with the developments on technologies for integration and miniaturization. Furthermore, the progress on communication technologies experienced in the last decade plays an important role in multiple vehicle systems.

6. **Cost**: A single vehicle with the performance required to execute some tasks could be an expensive solution when comparing to several low-cost vehicles performing the same task. This is clear for UAVs and particularly in small size, light, and low-cost versions, where constraints such as power consumption, weight, and size play an important role.

1.8.2 Coordination and Cooperation

In platforms involving multiple vehicles, the concepts of coordination and cooperation play an important role. In general, the coordination deals with the sharing of resources, and both temporal and spatial coordination should be considered. The temporal coordination relies on synchronization among the different vehicles, and it is required in a wide spectrum of applications. For instance, for objects monitoring, several synchronized perceptions of the objects could be required. In addition, spatial coordination of UAVs deals with the sharing of the space among them to ensure that each UAV will be able to perform safely and coherently regarding the plans of the other UAVs and the potential dynamic and/or static obstacles.

Cooperation can be defined as a "joint collaborative behavior that is directed toward some goal in which there is a common interest or reward" (Barnes and Gray 1991). Given some task specified by a designer, a multiple robot system displays cooperative behavior if, due to some underlying mechanism (i.e., the "mechanism of cooperation"), there is an increase in the total utility of the system. The cooperation of heterogeneous vehicles requires the integration of sensing, control, and planning in an appropriated decisional architecture. These architectures can be either centralized or decentralized depending on the assumptions on the knowledge's scope and accessibility of the individual vehicles, their computational power, and the required scalability.

1.8.3 Classification of Multi-UAV Architectures

Multi-UAV systems can be classified from different points of view. One possible classification is based on the coupling between the UAVs (see Figure 1.6):

1.8.3.1 Physical coupling

In this case, the UAVs are connected by physical links and then their motions are constrained by forces that depend on the motion of other UAVs. The lifting and transportation of loads by several UAVs.

1.8.3.2 Formations

The vehicles are not physically coupled, but their relative motions are strongly constrained to keep the formation. Then, the motion planning problem can be also formulated considering the formation as a whole. Regarding the collision avoidance problem within the team, it is possible to embed it in the formation control strategy. Scalability properties to deal with formations of many individuals are relevant, and then, decentralized control architectures are usually preferred.

1.8.3.3 Intentional cooperation

The UAVs of the team move according to trajectories defined by individual tasks that should be allocated to perform a global mission. These UAV trajectories typically are not geometrically related as in the case of the formations.

1.8.3.4 Swarms

UAV swarm is in this context expressed as multiple UAVs collaborating in order to perform a certain task/mission, either controlled by operator(s) or autonomously. The multiple UAVs may be identical or differ in characteristics such as payload (sensors), level of automation or even platform configurations. For this reason, UAV swarms are generally classified as either Homogeneous or Heterogeneous swarms in order to specify the similarities of the UAVs with in the swarm.

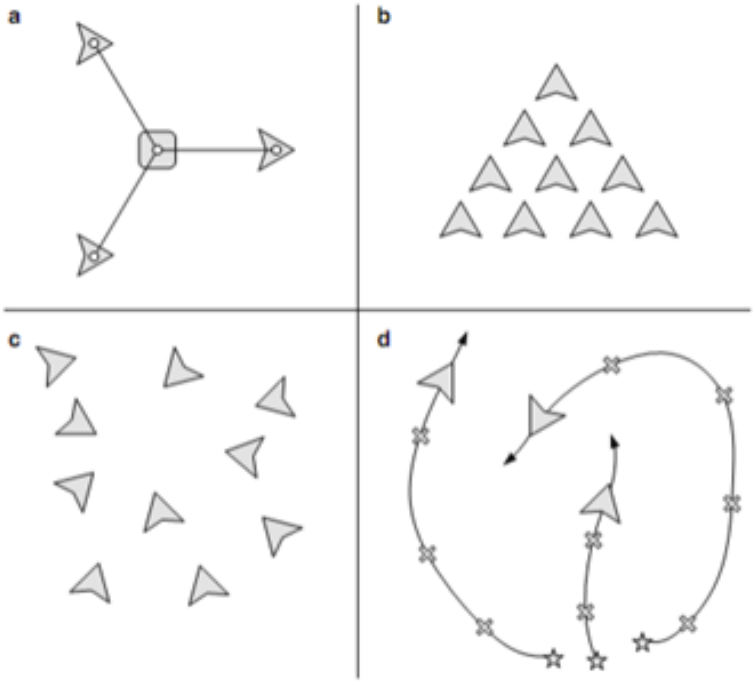

Figure 1.6: Classification of Multi-UAV Architectures

A comprehensive set of literature has been published (such as [11] - [13]) with a wide variety of approaches designed for controlling (path planning and task allocating) a multi-UAV system. These approaches are typically based on either centralized or decentralized control architectures [14]. [15] Provides a detailed description on the differences between these control architectures.

1. **Centralized control architecture** : A centralized control approach permits a low level of autonomy of the UAV system, without any communication between each individual UAV. The operator(s) receives information from each UAV and coordinates and prescribes task assignments to each individual UAV. This approach have the tendency of being simpler theoretically (low level autonomy) and easier to optimize, but less redundant and robust to failure of a UAV or communication.

2. **Decentralized control architecture** : A Decentralized (distributed) control approach requires a high level of autonomy of the UAV system and communication between each individual UAV. Each UAV must be able to communicate, share and receive information and make necessary decisions, thus shifts the role of the operator to a supervisory level. Systems based on decentralized approach has not been utilized until the last years as it is far more complicated and requires a high level of autonomy.

1.9 Multi-Agent Systems

Since its inception in the mid to late 1970s distributed artificial intelligence (DAI) evolved and diversified rapidly. Today it is an established and promising research and application field which brings together and draws on results, concepts, and ideas from many disciplines, including artificial intelligence (AI), computer science, sociology, economics, organization and management science, and philosophy. DAI is the study, construction, and application of multi agent systems, that is, systems in which several interacting, intelligent agents pursue some set of goals or perform some set of tasks. Multi agent systems can differ in the agents themselves, the interactions among the agents, and the environments in which the agents act.

1.9.1 Agents

"Agents" are autonomous, computational entities that can be viewed as perceiving their environment through sensors and acting upon their environment through effectors. To say that agents are computational entities simply means that they physically exist in the form of programs that run on computing devices. To say that they are autonomous means that to some extent they have control over their behavior and Call act without the intervention of humans and other systems. Agents pursue goals or carry out tasks in order to meet their design objectives, and in general these goals and tasks can be supplementary as well as conflicting.

An intelligent agent is one that is capable of flexible autonomous action in order to meet its design objectives, where flexibility means three things :

1. **Reactivity** : intelligent agents are able to perceive their environment, and respond in a timely fashion to changes that occur in it in order to satisfy their design objectives.
2. **Pro-activeness** : intelligent agents are able to exhibit goal-directed behavior by taking the initiative in order to satisfy their design objectives.
3. **Social ability** : intelligent agents are capable of interacting with other agents (and possibly humans) in order to satisfy their design objectives.

1.9.2 Agent Architecture

An agent architecture is essentially a map of the internals of an agent—its data structures, the operations that may be performed on these data structures, and the control flow between these data structures. We will consider four classes of agents:

- **Logic based agents** : in which decision making is realized through logical deduction.
- **Reactive agents** : in which decision making is implemented in some form of direct mapping from situation to action.
- **Belief-desire-intention agents**: in which decision making depends upon the manipulation of data structures representing the beliefs, desires, and intentions of the agent; and finally.
- **Layered architectures**: in which decision making is realized via various software layers, each of which is more-or-less explicitly reasoning about the environment at different levels of abstraction.

The concept of an intelligent agent that interacts allows various degrees of degradation, and is perhaps best viewed as a "guideline" for designing and analyzing systems rather than an "instruction" that allows no variation, or a precise "criterion" that always allows one to determine whether an object does or does not fulfill it.

1.9.3 Advantages of multi agent system

In addition, multi agent systems, as distributed systems, have the capacity to offer several desirable properties:

1. **Speed-up and Efficiency**: Agents can operate asynchronously and in parallel, and this can result in an increased overall speed (provided that the overhead of necessary coordination does not outweigh this gain).

2. **Robustness and Reliability**: The failure of one or several agents does not necessarily make the overall system useless, because other agents already available in the system may take over their part.

3. **Scalability and Flexibility**: The system can be adopted to an increased problem size by adding new agents, and this does not necessarily affect the operationality of the other agents.

4. **Costs**: It may be much more cost-effective than a centralized system, since it could be composed of simple subsystems of low unit cost.

5. **Development and Reusability**: Individual agents can be developed separately by specialists (either from scratch or on the basis of already available hardware and/or software facilities), the overall system can be tested and maintained more easily, and it may be possible to reconfigure and reuse agents in different application scenarios.

1.10 Search and rescue

One area where Unmanned Aerial Vehicles (UAV's) could find a significant application is for the search element of Search and Rescue (SAR) operations. These operations are often tedious, often operated under extreme conditions and when searching for a missing person the sensors have a higher success rate at low altitudes which is too risky for manned flight particularly over water. In this work it is proposed to investigate the utilization of a group of UAVs (UAV swarm) contribution to a search and rescue operation. The reasons for using a UAV swarm is that this greatly reduce the time to fully cover the significant wake region. Also as each UAV will be able to cover a specific area repeatedly over a short period of time, a more precise search is possible reducing the risk of failing to detect the person.

1.10.1 Exploration formation

Approaches to formation generation in robots may be distinguished by their sensing requirements, their method of behavioral integration, and their communication abilities. In general we can consider four formations for the team of multi agents as illustrated in Figure 1.7:

1. **Line**: where the agents travel line–abreast, side by side

2. **Column**: where the agents travel one after the other

3. **Coil**: where the agents travel in a coils (e.g. school of fish)

4. **Wedge**: where the agents travel in a Wedge (e.g. flock of birds) Line and Wedge movement formation provide larger line of sight for the multi–agent system, therefore it leads the system towards faster coverage. In these two formations all agents follow one direction.

However, following these formations and maintaining the implicit communication among all the agents, inside a cluttered environment, is impossible. They are suitable only for open areas (e.g. high in the sky). In contrary to above formations, Column and Coil are able to maintain their implicit communication.

Since agents follow each other, they replace one another. However, during their team movement their coverage is very small (e.g. for column line of sight, for unexplored area, is only one agent). Coil formation (i.e. multi column moving in a line) is like creating a larger agent from several smaller agents, therefore they may not be able to enter narrow pathways. As we are dealing with narrow pathways inside our search fields, coil, wedge, and line are not practical formations for our confined and cluttered maze–like environments. Therefore the best movement formation in these kinds of environments has been selected as column.

Columns have been used for perimeter detection by mobile agents. This technique has been inspired by pack of wolves and their agents are able to track a dynamic perimeter, e.g. an oil spill on the sea surface. Furthermore an outdoor perimeter surveillance over an open large area using a swarm of agents that investigate alarms for intrusion detection sensors.

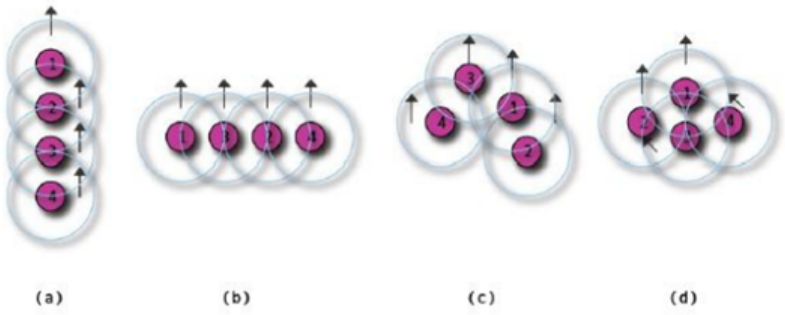

Figure 1.7: Formation for the team of robots

1.10.2 Standard Search Strategies

Several search strategies have been recognized and set as standards by the International Maritime Organization and International Civil Aviation Organization. These are published annually in IAMSAR (International Aeronautical and Maritime Search and Rescue) manual [33]. These are designed to provide simple and effective visual search patterns that can be applied in various situations. The standard search patterns are as follow:

1. Parallel Track and Creeping line Search
2. Expanding Square Search
3. Sector Search
4. Track Line Search
5. Contour Search

1.11 Contributions

This thesis aims to develop a quadrotors multi-agents system, in which a swarm of agents coordinate and collaborate their actions in order to search for survivals in a disaster area and rescue them. The following contributions are considered:

1. New modeling approach: A full quadrotors model is presented by modeling the dynamics using the Newton-Quaternions approach, derive the DC Brushless motors transfer function, and find the relation between the LI-PO battery and the engines speed of rotation, which is so similar to the real functional mode of a quadrotors.
2. The attitude stability and the trajectory tracking are assured using a double loop non-linear optimal backstepping controller. The controller parameters are derived using the Multi-Objective Genetic Algorithm (MO-GA) that optimizes all the cost functions jointly. Furthermore the relation between the input and the output space is given via a differential flatness approach that presents a direct relation between the states and their derivatives, which then simplifies the trajectory generation problem.
3. New formation control scheme that is based only on the attitude control of the swarm agents is proposed, this allow the formation hold with a minimum of a sharing data, and make the controller more robust to any external disturbances.
4. Trajectory tracking and the formation control algorithms are based on a double loop control structure with an LQR and SMC controllers, which then minimize the energy consumption and secure the swarm from any collisions between the different quadrotors.
5. New SAR system that combines the IAMSAR strategies with the probability of detection depending on the UAV flying altitude, and then compares the different strategies performances using a study case mission.
6. An efficient search methodology for employing a UAV swarm is developed in order to efficiently locate intelligent and evading targets within the search region.

7. The search algorithm maximizes the probability of targets' detection and minimizes the expected search time while minimizing the number of UAVs required. Furthermore the UAV failure is faced by reconfiguring the UAVs to optimally continue the mission with the surviving assets.

1.12 Outline of the thesis

This thesis is divided into six chapters. chapter 2 discusses in detail the modeling of the single quadrotors and the wind gust model. In chapter 3, the control and the trajectory generation problems for a single quadrotors are investigated. chapter 4 proposes the formation control strategies as wheel as the control algorithm for the multi-quadrotors system. The coordination and the collaboration between the agents are considered within a multi-agent system. chapter 5 is devoted to the development of SAR strategies using a single quadrotors UAV. The SAR system using Multi-UAV formation control is developed in chapter 6. A full scenario of SAR operation using multi-agents quadrotors is also given. Finally, in chapter 7 this work is ended by some conclusions and propositions of future works.

Chapter

2

Quadrotors Modeling

Contents

2.1 Introduction

A quadrotors is a rotorcraft capable of hover, forward flight and vertical takeoff landing (VTOL); it is emerging as a fundamental research and application platform at present with flexibility, adaptability, and ease of construction. As drawbacks, the nonlinear, underactuated dynamic system and the high energy consumption can be mentioned. The aim of this research is to design a multi-layer navigation system that guaranties the stability, reduces the energy consumption and maximizes the performances.

Before the control issue, a quadrotors dynamics modeling is needed. For this purpose the Newton-Euler approach is often used [33, 41, 47] for its simplicity, but it presents a singularity whenever the pitch angel $\theta = \pm\frac{\pi}{2}$. As an alternative of this approach the most recent researches [8, 15, 16, 75] converge over the modeling with quaternions instead of the Euler angles, to eliminate the gimball lock phenomena, and simplify the modeling algebra.

In the other parts most of the researches are limited on the dynamic modeling, but no one has introduced the rotors dynamics as well as the energy source models, this is important since it affects directly the quadrotors stability.

For this work a full quadrotors model is presented by modeling the dynamics using the Newton-Quaternions approach, derive the DC Brushless motors transfer function, and find the relation between the LI-PO battery and the engines speed of rotation, which is so similar to the real functional mode of a quadrotors.

This chapter is organized as follow: section 2.2 gives the quadrotors dynamic modeling using quaternions, to the and finally the battery model. section 2.3 introduces the wind gust model, while section 2.4 shows the PWM signals generation. section 2.5 discusses the rotors dynamics and section 2.6 the battery model. Finally in section 2.7 the conclusion is given.

2.2 Quadrotors Modeling

2.2.1 Dynamic System Model

As illustrated in Figure 2.1 the quadrotors is a flying robot includes four rotors along a rigid cross frame. Where rotors 2 and 4 are turning clockwise with an angular velocity ω_2, ω_3 and producing a lift force F_2, F_3 respectively. Rotors 1 and 3 are turning counter clockwise with an angular velocity ω_1, ω_4 and producing a lift force F_1, F_4 respectively. Let $E_i = \{x_i, y_i, z_i\}$ denotes the inertial frame fixed with the earth while $E_b = \{x_b, y_b, z_b\}$ denotes the body frame attached to the quadrotors body.

The rotational matrix $R(t) \in SO(3)$ that mapping the vectors expressed in the body frame E_b into the vectors expressed in the inertial frame E_i, by assuming the order of rotation to be roll (ϕ), pitch (θ) then yaw (ψ), is:

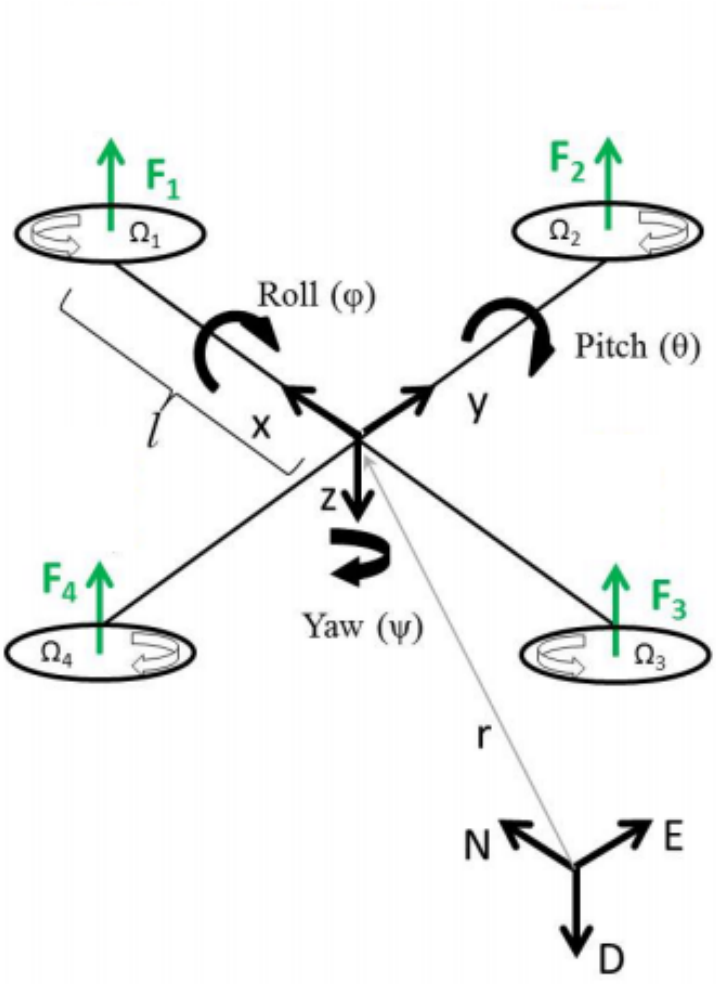

Figure 2.1: Inertial and body-fixed frame of the quadrotors

$$R(t) = \begin{bmatrix} C_\psi C_\theta & S_\phi S_\theta C_\psi - C_\phi S_\psi & C_\phi S_\theta C_\psi - S_\phi S_\psi \\ S_\theta S_\psi & S_\phi S_\theta S_\psi - C_\theta C_\psi & C_\phi S_\theta S_\psi \\ -S_\theta & S_\phi C_\theta & C_\phi C_\theta \end{bmatrix} \tag{2.1}$$

Where $s(.)$ and $c(.)$ are abbreviations for $sin(.)$ and $cos(.)$ respectively.

The rotation matrix R will be used in formulating the dynamics model of the quadrotors, its significance is due to the fact that some states are measured in the body frame (e.g. the thrust forces produced by the propellers) while some others are measured in the inertial frame (e.g. the gravitational forces and the quadrotor's position). Thus, to have a relation between both types of states, a transformation from one frame to the other is needed.

For modeling the physics of the quadrotors two alternative approaches could be followed: a) the full physical model can be derived by the utilization of Newtons' laws of motion and producing a frame dependent model, or b) to use the Euler-Newton equations for translational and rotational dynamics of a rigid body. The utilization of the second approach greatly simplifies the derivation of the model as the only unknown in the derived model is the connection among the control signal and the corresponding torque. In the first place, some assumptions are reasonable and essential shown as follows: [1]

- The structure is supposed rigid.
- The structure is supposed symmetrical.
- The CoG and the body fixed frame origin are assumed to coincide.
- The propellers are supposed rigid.
- Thrust and drag are proportional to the square of propeller's speed.

The Netwon–Euler equations describe the combined translational and rotational dynamics of a rigid body. According to this method, the motion equations of a quadrotors subject to external forces $F \in R$ and torques $\tau \in R$ given by the following Newton-Euler equations with respect to the body coordinate frame [2]

$$\begin{bmatrix} mI_{3\times3} & 0 \\ 0 & J \end{bmatrix} \begin{bmatrix} \dot{V} \\ \dot{\Omega} \end{bmatrix} + \begin{bmatrix} \omega_b \times mV \\ \Omega \times J\Omega \end{bmatrix} = \begin{bmatrix} F \\ \tau \end{bmatrix} \tag{2.2}$$

Where: $I_{3\times3}$, m, V correspond to identity matrix, total mass of the body, linear speed of the CoM, and $\Omega = [p, q, r]^T$ is the angular velocity in the body frame E_b respectively. J introduces the inertia matrix with respect to the body-fixed frame:

$$J = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \tag{2.3}$$

2.2.1.1 Translation Motion

The translation motion can be described using Newton first law as follow:

$$\frac{d(mV)}{dt} = \sum F \tag{2.4}$$

The total forces (listed below) acting on the quadrotors are given by:

$$m.\dot{V} = F_g - F_t + F_d \tag{2.5}$$

Where F_g is the gravity force given by:

$$F_g = Rmg \tag{2.6}$$

With: g is the gravity vector
The propeller thrust F_t is given by the following equation:

$$F_t = c_T \rho \pi r^4 \omega_i^2 = k_T \omega_i^2 \tag{2.7}$$

With ρ is the air density, r the radius of the propeller, c_T is the thrust that depends on blade rotor characteristics, such as number of blades, chord length of the blade, and cube of the rotor blade radius.
Finally the drag force F_d can be obtained by Equation 2.8 :

$$F_d = \frac{1}{2} \rho V^2 S_w c_D \tag{2.8}$$

With : c_D is the drag constant that depends on blade rotor characteristics, such as number of blades, chord length of the blade, and cube of the rotor blade radius, ρ is the air density and S_w is the reference area.

2.2.1.2 Rotation Motion

Using the Newton's law about the rotation motion, the sum of moments is given by:

$$\begin{aligned} \dot{R} &= R * S(\Omega) \\ J\dot{\Omega} &= -S(\Omega) * J\Omega + \tau(t) \end{aligned} \tag{2.9}$$

With:

$$S(\Omega) = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \tag{2.10}$$

And: $\tau(t) = \tau_u(t) + \tau_{ext}(t)$,

$$\tau_u = c_D \rho \pi r^5 \omega_i^2 + J_r \dot{\omega}_l = k_D \omega_i^2, i = 1:4 \tag{2.11}$$

With J_r is the rotor inertia.
The external torques τ_{ext} applied on the aerial vehicle in the body-fixed frame are:

- **Thrust moment**

$$\tau_f = \begin{bmatrix} l(F_4 - F_2) \\ l(F_3 - F_1) \\ k_D(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{bmatrix} \tag{2.12}$$

- **Gyroscopic Moment**

  The gyroscopic moment of a rotor is a physical effect in which gyroscopic torques or moments attempt to align the spin axis of the rotor along the inertial z-axis

$$\tau_g = \sum_{i=1}^{4} \Omega \wedge J_r \begin{bmatrix} 0 \\ 0 \\ (-1)^i \omega_i \end{bmatrix} \tag{2.13}$$

The vector $\Theta = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T$ describes the orientation of the body frame with respect to the inertial frame called the Euler angles. The relation between the angular speeds vector $\begin{bmatrix} p & q & r \end{bmatrix}^T$ and Euler angles derivatives vector η is:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & sin(\phi)tan(\theta) & cos(\phi)tan(\theta) \\ 0 & cos(\phi) & -sin(\phi) \\ 0 & sin(\phi)sec(\theta) & cos(\phi)sec(\theta) \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{2.14}$$

The designed control inputs are given by Equation 2.15:

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} k_T(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \\ lk_T(\omega_1^2 - \omega_2^2 - \omega_3^2 + \omega_4^2) \\ lk_T(\omega_1^2 + \omega_2^2 - \omega_3^2 - \omega_4^2) \\ k_D(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{bmatrix} \tag{2.15}$$

Where l is the distance from the center of mass to the motor axis of action. Consequently the complete dynamic model is as follows:

$$
\begin{aligned}
\ddot{\phi} &= \frac{I_y - I_z}{I_x}\dot{\theta}\dot{\psi} + \frac{J_r}{I_x}\dot{\theta}\Omega + \frac{l}{I_x}U_2 \\
\ddot{\theta} &= \frac{I_z - I_x}{I_y}\dot{\phi}\dot{\psi} - \frac{J_r}{I_y}\dot{\phi}\Omega + \frac{l}{I_y}U_3 \\
\ddot{\psi} &= \frac{I_x - I_y}{I_z}\dot{\phi}\dot{\theta} + \frac{U_2}{I_z} \\
\ddot{x} &= \frac{1}{m}(C_\phi S_\theta C_\psi + S_\phi S_\psi)U_1 \\
\ddot{y} &= \frac{1}{m}(C_\phi S_\theta C_\psi - S_\phi S_\psi)U_1 \\
\ddot{z} &= \frac{1}{m}(C_\phi S_\theta)U_1 - g
\end{aligned} \tag{2.16}
$$

2.2.2 Modeling with quaternions

According to Equation 2.14 The problem of singularity emerges whenever $\theta = \pm\frac{\pi}{2}$;To avoid this problem other methods are used to represent the quadrotors attitude such as the direction cosine matrix (DCM) and the Quaternions. For simplicity quaternions are often used.

A quaternion, which belong to the quaternion space H, is a hyper complex number of $\mathbb{H}$ rank 4, described by:

$$
\begin{aligned}
q(t) \in \mathbb{H}, \bar{q}(t) \in \mathbb{R}^3, q_0(t) \in \mathbb{R} \\
q(t) = q_0(t) + \bar{q}(t) = q_0(t) + [q_1(t), q_2(t), q_3(t)]^T
\end{aligned} \tag{2.17}
$$

With $q_0(t)$ is a given quaternion, $\bar{q}(t)$ is the complex and $q_0(t)$ is the scalar parts of $q(t)$.

The quaternion product $\otimes$ of tow quaternion vectors q and r is given as follow:

$$
\begin{aligned}
r(t) \in \mathbb{H}, \bar{r}(t) \in \mathbb{R}^3, r_0(t) \in \mathbb{R} \\
q \otimes r = (q_0 r_0 - \bar{q}\bar{r}) + (r_0\bar{q} + q_0\bar{r} + \bar{q} \times \bar{r})
\end{aligned} \tag{2.18}
$$

The quaternion conjugate q^* is defined as: $q^* = q_0 - \bar{q}$.

In this thesis only the unit quaternions vectors are used, thus, $q^{-1} = q^*$. And $||q||$ is done by:

$$
||q|| = \sqrt{q \otimes q^*} = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} \tag{2.19}
$$

The three-dimensional rotation of any vector is given as a quaternion multiplication on the left by the unit quaternion q and on the right by its conjugate q^*. This mathematical operation can be rewritten as a multiplication of the matrix R_q and the above mentioned vector. Because we are rotating a vector, only the vector part from the rotation matrix is extracted, as can be seen in Equation 2.20, where $q_{13\times}$ stands for a skew-symmetric matrix. The rotation matrix R_q is orthogonal; therefore the expression $R_q^{-1} = R_q^T$ is true. [3]

$$\begin{aligned} R_q &= (q_0 I + q_{13x})^2 + \bar{q}\bar{q}^T \\ &= (q_0^2 - \bar{q}^T\bar{q})I + 2q_0 q_{13x} + 2\bar{q}\bar{q}^T \\ &= \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_3q_0) & 2(q_1q_3 + q_2q_0) \\ 2(q_1q_2 + q_3q_0) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_1q_0) \\ 2(q_1q_3 - q_2q_0) & 2(q_2q_3 + q_1q_0) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \end{aligned} \tag{2.20}$$

The rotation can also be represented using a rotation vector as denoted in Equation 2.21 where u is the rotation axis (unit vector) and α is the angle of rotation. Using this notation can have many benefits when creating an error or specifying a reference as it has a direct physical connection. [3]

$$q = cos\frac{\alpha}{2} + usin\frac{\alpha}{2} \tag{2.21}$$

The derivative of a quaternion is given by the quaternion multiplication of the quaternion q and the angular velocity of the system ω, which in this case is the quadrotor. [3]

$$\dot{q} = \frac{1}{2} q \otimes \Omega \tag{2.22}$$

Finally, for representing quaternion rotations in a more intuitive manner, the conversion from Euler angles to quaternion and from quaternion to Euler angle can be performed by utilizing the following two equations respectively. This property is very useful in case that the aim is to represent an orientation in angles, while retaining the overall dynamics of the system in a quaternion form. [4]

$$\begin{bmatrix} cos\frac{\phi}{2}cos\frac{\theta}{2}cos\frac{\psi}{2} + sin\frac{\phi}{2}sin\frac{\theta}{2}sin\frac{\psi}{2} \\ sin\frac{\phi}{2}cos\frac{\theta}{2}cos\frac{\psi}{2} - cos\frac{\phi}{2}sin\frac{\theta}{2}sin\frac{\psi}{2} \\ cos\frac{\phi}{2}sin\frac{\theta}{2}cos\frac{\psi}{2} + sin\frac{\phi}{2}cos\frac{\theta}{2}sin\frac{\psi}{2} \\ cos\frac{\phi}{2}cos\frac{\theta}{2}sin\frac{\psi}{2} - sin\frac{\phi}{2}sin\frac{\theta}{2}cos\frac{\psi}{2} \end{bmatrix} \tag{2.23}$$

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} atan2(2(q_0q_1 - q_2q_3), q_0^2 - q_1^2 - q_2^2 + q_3^2) \\ asin2(2(q_0q_2 - q_3q_1)) \\ atan2(2(q_0q_3 + q_1q_2), q_0^2 + q_1^2 - q_2^2 - q_3^2) \end{bmatrix} \tag{2.24}$$

The derivative translation motion according to the quaternion approach will concern only the thrust force, since it is the only force expressed in the body frame.

Using Equation 2.6 the obtained thrust force in the inertial frame is expressed as follow:

$$F_t = q \otimes F_t \otimes q^* \tag{2.25}$$

The complete translation motion using quaternion approach can be expressed as follow:

$$\begin{aligned}
\ddot{x} &= \frac{1}{m}((2q_1q_3 + 2q_2q_4))\sum_{i=1}^{4} F_i \\
\ddot{y} &= \frac{1}{m}((2q_4q_3 - 2q_2q_1))\sum_{i=1}^{4} F_i \\
\ddot{x} &= \frac{1}{m}((q_0^2 - q_1^2 - q_2^2 + q_3^2))\sum_{i=1}^{4} F_i - mg
\end{aligned} \tag{2.26}$$

Finally the dynamic model of a quadrotor using Newton-Euler equations with quaternions is described as follows:

$$\ddot{p} = q \otimes \frac{F_t}{m} \otimes q^* + \overline{g} \tag{2.27}$$

$$\dot{q} = \frac{1}{2} q \otimes \omega \tag{2.28}$$

$$\dot{\Omega} = J^{-1}(\tau - \Omega \times J\Omega) \tag{2.29}$$

The system describes by the equations Equation 2.30, Equation 2.31 and Equation 2.32 is then linearized around an hovering point where: $\dot{p} = [0,0,0]^T$, $\omega = [0,0,0]^T$ and $q = [1,0,0,0]^T$, thus:

$$\ddot{p} = \begin{bmatrix} g(2q_0q_1) \\ -g(2q_0q_2) \\ \frac{F_t}{m} - g \end{bmatrix} \tag{2.30}$$

$$J\dot{\Omega} = \tau \tag{2.31}$$

$$\dot{q} = \frac{1}{2} q\Omega \tag{2.32}$$

This linear model is designed to be used later with the LQR controller.

2.3 Adopted Wind Model

In this study, because of the symmetrical assumption of the quadrotor structure underlined previously, longitudinal, lateral, and vertical wind components are expressed here according to as follows:

$$w_x = W_x(x, z, t) = W_x(z) + \delta_x(v, z, t) \tag{2.33}$$

$$w_y = W_y(y, z, t) = W_y(z) + \delta_y(v, z, t) \tag{2.34}$$

$$w_z = W_z(x, y, z, t) = \delta_z(v, z, t) \tag{2.35}$$

Where: $W_x(z)$, $W_y(z)$, and $\delta_{(x,y,z)}(v, z, t)$ represent the deterministic and stochastic components of the considered wind, respectively.

The deterministic horizontal wind speed components are functions of the altitude z according to a reference altitude z_0, and they are expressed as follows:

$$W_x(z) = W_0(z) \ln(\frac{z}{z_0}) \tag{2.36}$$

$$W_y(z) = W_0(z) \ln(\frac{z}{z_0}) \tag{2.37}$$

$$W_z(z) = W_0^* cos(\omega z + \phi_0) \tag{2.38}$$

Where ω and W_0^* denote the circular space frequency and magnitude of the considered wind component.

The stochastic wind components adopt the Dryden spectrum model generated from two normalized white Gaussian noise processes through linear filters, such as:

$$H_{\delta_{x,y}} = \sigma_{x,y} \sqrt{\frac{2L_{xx,yy}}{v}} \sqrt{\frac{1}{1 + \frac{L_{xx,yy}}{v} s}} \tag{2.39}$$

And:

$$H_{\delta_z} = \sigma_z \sqrt{\frac{2L_{zz}}{v}} \frac{1 + \sqrt{3}\frac{L_{zz}}{v} s}{(1 + \frac{L_{zz}}{v} s)^2} \tag{2.40}$$

Here, L_{xx} , L_{yy} and L_{zz} are shape parameters (turbulence lengths), such as the following:

For z≤305 m,

$$L_{xx,yy} = \frac{z}{(0.177 + 0.027z)^{1.2}} \tag{2.41}$$

$$L_{zz} = z \tag{2.42}$$

For z>305 m,

$$L_{xx} = L_{yy} = L_{zz} = 305m \tag{2.43}$$

Where $\sigma_{(x,y)}$ and σ_z represent standard deviations of independent processes, such as:

$$\sigma_z = 0.1W_{20} \tag{2.44}$$

And W_{20} is the horizontal wind speed at 20 ft above ground level. For z≤305 m,

$$\sigma_{x,y} = \frac{\sigma_z}{(0.177 + 0.027z)^{0.4}} \tag{2.45}$$

For z>305 m,

$$\sigma_{x,y} = \sigma_z \tag{2.46}$$

The time and spatial derivatives of the wind components are then given by the following:

$$\begin{bmatrix} \dot{W}_x \\ \dot{W}_y \end{bmatrix} = \begin{bmatrix} W_{xx} & W_{xy} & W_{xz} \\ W_{yx} & W_{yy} & W_{yz} \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} + \begin{bmatrix} W_{xt} \\ W_{yt} \end{bmatrix} \tag{2.47}$$

With:

$$W_{xx} = \frac{\partial W_x}{\partial x} \quad W_{yy} = \frac{\partial W_y}{\partial y} \quad W_{xy} = \frac{\partial W_x}{\partial y} \tag{2.48}$$

$$W_{yx} = \frac{\partial W_y}{\partial x} \quad W_{xz,yz} = \frac{\partial W_{x,y}}{\partial z} \quad W_{xt,yt} = \frac{\partial W_{x,y}}{\partial t} \tag{2.49}$$

And:

$$W_z = W_{zx}\dot{x} + W_{zy}\dot{y} + W_{zz}\dot{z} + W_{zt} \tag{2.50}$$

With:

$$W_{zx} = \frac{\partial W_z}{\partial x} \quad W_{zy} = \frac{\partial W_z}{\partial y} \quad W_{zz} = \frac{\partial W_z}{\partial z} \quad W_{zt} = \frac{\partial W_z}{\partial t} \tag{2.51}$$

2.4 Motor Mixer

As in a real application the control inputs are interpreted from the radio channels like a PWM (Pulse Width Modulation) signals between values of 1000 μs and 2000 μs. The ESC (Engine Speed Controller) receives the PWM signal every single time step Δt which correspond to the model simulation frequency. The following formulate is used to mix the signals of roll, pitch, yaw and thrust such that the appropriate motors deliver an increase or decrease in thrust.

$$\begin{aligned}
PWM1 &= (\frac{(\tau_{u_x} + \tau_{u_y} - \tau_{u_z}) * F_1}{2} + F_1) * 1000 + idle_{PWM} \\
PWM2 &= (\frac{(-\tau_{u_x} + \tau_{u_y} + \tau_{u_z}) * F_2}{2} + F_2) * 1000 + idle_{PWM} \\
PWM3 &= (\frac{(-\tau_{u_x} - \tau_{u_y} - \tau_{u_z}) * F_3}{2} + F_3) * 1000 + idle_{PWM} \\
PWM4 &= (\frac{(\tau_{u_x} - \tau_{u_y} + \tau_{u_z}) * F_4}{2} + F_4) * 1000 + idle_{PWM}
\end{aligned} \tag{2.52}$$

The $idle_{PWM}$ is seted to 1000 μs that always unsure the minimum generated thrust. Attitude commands are interpreted from the input PWM radio signal and scaled to range in magnitude from -1 to 1 to define the imposed 'load' over the engines.

2.5 Rotors Dynamics

In the quadrotors the propellers are driven by the brushless motors. A typical dc motor equivalent electrical circuit is illustrated in Figure 2.2:

Using the Kirchehoff's voltage law the following equation is obtained:

$$V_s = Ri_a + L_w\frac{di_a}{dt} + e \tag{2.53}$$

The back EMF (Electro Magnitic Force), e is derived as follow:

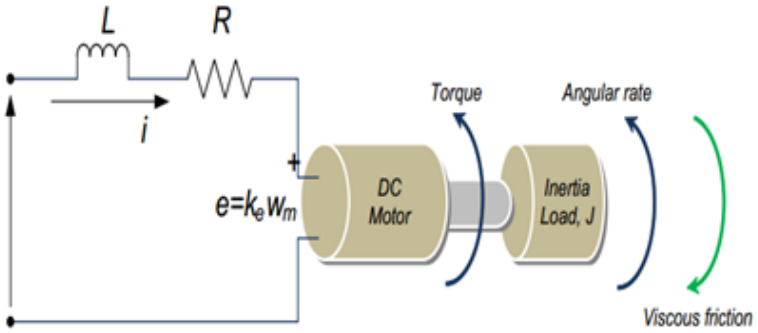

Figure 2.2: DC motor equivalent electrical circuit

$$e = -R_w i_a - L_w \frac{di_a}{dt} + V_s \tag{2.54}$$

Where: R_w : the winding resistance, L_w is the winding inductance, V_s is the DC voltage source, And i_a is the armature current.

Similarly, and using the Newton's second law, the mechanical properties related to the torque of the system would be:

$$J_r \frac{d\omega}{dt} = \sum T_i \tag{2.55}$$

$$T_e = k_f \omega + J_r \frac{d\omega}{dt} + T_L \tag{2.56}$$

Where: T_e is the electrical torque, K_f is the friction constant, J_r is the rotor inertia, ω is the angular velocity, And T_L is supposed to be a mechanical load.

The electrical torque T_e and the back EMF, e, can be written as: $e = k_e \omega_m$ and $T_e = k_t \omega_m$ Where: K_e is the electrical motor constant, and K_t is the torque constant.

Therefore rewritten Equation 2.53, Equation 2.54, Equation 2.55 and Equation 2.56 can be obtained:

$$\frac{di_a}{dt} = -i_a \frac{R_w}{L_w} - \frac{k_e}{L_w} \omega_m + \frac{1}{L_w} V_s \tag{2.57}$$

$$\frac{d\omega}{dt} = i_a \frac{k_t}{J_r} - \frac{k_f}{J_r} \omega_m + \frac{1}{J_r} T_L \tag{2.58}$$

Using Laplace transform, and by considering no load ($T_L = 0$):

$$si_a = -i_a \frac{R_w}{L_w} - \frac{k_e}{L_w}\omega + \frac{1}{L_w}V_s \tag{2.59}$$

$$s\omega = i_a \frac{k_t}{J_r} - \frac{k_f}{J_r}\omega \tag{2.60}$$

By driven i from Equation 2.60, and substituting it into Equation 2.59:

$$(\frac{s^2 J_r}{k_t} + \frac{sk_f}{k_t} + \frac{sR_w J_r}{k_t L_w} + \frac{k_f R_w}{k_t L_w} + \frac{k_e}{L_w})\omega = \frac{1}{L_w}V_s \tag{2.61}$$

The transfer function that relates the angular velocity to source voltage is:

$$G(s) = \frac{\omega}{V_s} = \frac{k_t}{s^2 J_r L_w + (R_w J_r + k_f L_w)s + k_f R_w + k_e k_t} \tag{2.62}$$

Finally, and with some simplification assumptions, $G(s)$ is given as in Equation 2.63 :

$$G(s) = \frac{\omega}{V_s} = \frac{\frac{1}{k_e}}{\tau_m \tau_e s^2 + \tau_m s + 1} \tag{2.63}$$

With: $\tau_m = \frac{R_w J_r}{k_e k_t}$ and $\tau_e = \frac{L_w}{R_w}$

2.6 Battery Model

As mentioned before the battery generates a voltage Vs supplied to the engines. The voltage formula can be obtained by an approximation of the battery discharge curve illustrated in Figure 2.3:

Depending on the consumed current over the engines ($Eng-C$) and the embedded electronic devices ($Dev-C$), the battery cells voltage ($Cell-V$) can be approximated as a function of discharge capacity (DiC). Vs is then calculated depending on the battery cells voltage, the number of cells ($N - cell$) and the battery capacity ($capacity$). algorithm 2.1 explains the different steps.

It's important to mention that the effect of temperature has been neglected since we often work in the ambient temperature. The interaction between the quadrotors model components is illustrated in Figure 2.4

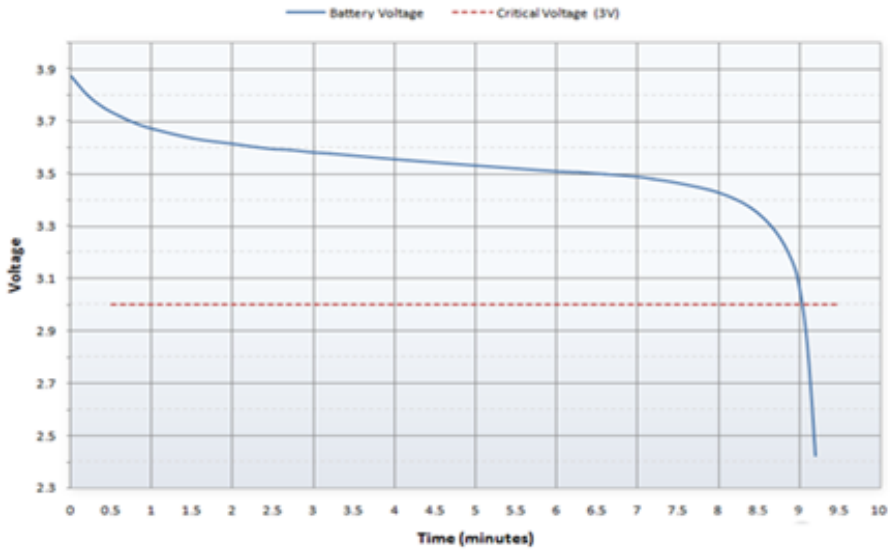

Figure 2.3: LIPO battery discharge curve

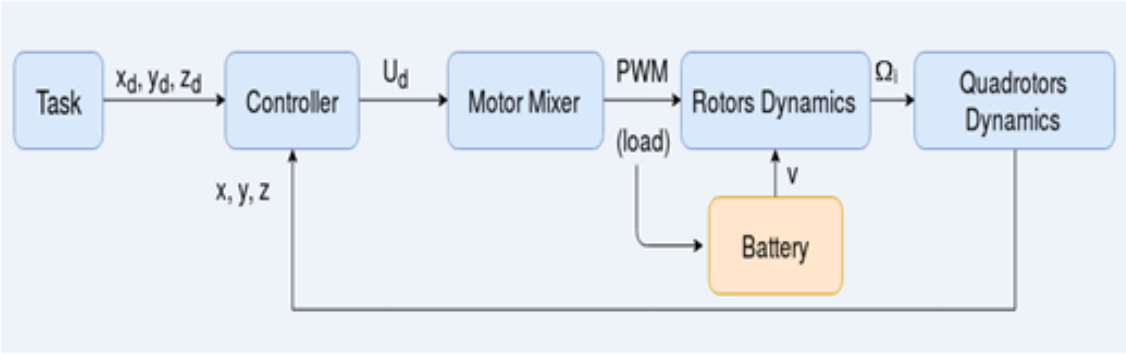

Figure 2.4: Quadrotors Full Model

2.7 Conclusion

In this chapter the quadrotors modeling problem was investigated using a new approach that model all the quadrotors components, the interaction between them and their physical limitations.

In the first section the quadrotors dynamics was modeled using the Newton-quaternion approach, this approach was used to eliminate the gimball lock phenomena, and simplify the modeling algebra.

The second part was concerned by the modeling of the wind gust perturbation, to simulate the disturbances over the controller.

The third part was about the brushless motor dynamics, while part four was concerned by the battery discharge algorithm.

Algorithm 2.1: Battery Discharge Algorithm

begin
  $current = Eng - C * load2 + Dev - C$;
  $discharge = current * \frac{\Delta t}{3600} + discharge$;
  $DiC = \frac{discharge}{capacity}$;
  if $(0 < DiC <= 0.20)$ **then**
    $Cell - V = -14.029 * DiC^3 + 16.975 * Dic^2 - 5.3339 * DiC + 4.2$;
    if $(0.20 < DiC < 0.70)$ **then**
      $Cell - V = -0.2 * DiC + 3.74$;
    else
      $Cell - V = -48 * DiC^3 + 89.6 * DiC^2 - 55.08 * DiC + 14.716$;
    end
    $Supp - V = N - Cell * Cell - V$;
    $V_s = Supp - V * load$;
  end
end

Chapter

3 Quadrotors Control and Trajectory Generation

Contents

3.1 Introduction

The aim of this research is to design a multi-layer navigation system that guaranties the stability, reduces the energy consumption and maximizes the performances.

Many works have been published on the quadrotors attitude stability [1,29,33,60], while others such as [3,15,17,37,69,73] treat the trajectory generation and tracking using both non-linear and feed-back linearization control techniques. For the sake of stabilization, regulation and trajectory tracking many types of controllers are used such as PID, linear quadratic LQR algorithm [8,44,65], H∞ loop forming method, sliding mode variable structure control, feedback linearization, backstepping [39,48], optimal control [35,71], and even intelligent control using neural networks and fuzzy logic [72,73]. A comparison of the different linear and non-linear controllers using the quaternion approach can be found in [47]. Overall all the above cited works aim to improve the robustness, the performances, and energy optimization, but they may fail to raise them jointly.

Thought this chapter the attitude stability and the trajectory tracking are assured using a double loop non-linear optimal backstepping controller. The controller parameters are derived using the Multi-Objective Genetic Algorithm (MO-GA) that optimizes all the cost functions jointly. Furthermore the relation between the input and the output space is given via a differential flatness approach that presents a direct relation between the states and their derivatives, which then simplifies the trajectory generation problem.

In the other hands an optimal trajectory generation algorithm with the obstacle avoidance ability is also introduced to secure the quadrotors from any collision.

This chapter is organized as follow: section 3.2 introduces a presentation of the designed system. section 3.3 introduces the differential flatness method with the LQR/OBS controllers and their application to the quadrotors trajectory tracking and control, while section 3.4 show the trajectory generation and the obstacle avoidance algorithms.section 3.5 discusses the simulation results of many scenarios and the controllers comparison. Finally, in section 3.6 the conclusion as well as the future recommendations is given.

3.2 System Design

The aim of this work is to concept an optimal navigation system that increases the performances and minimizes the energy consumption of a quadrotors UAV.

This system is designed as multi-layer so that the optimization can occur at several stages from the trajectory generation to the control architecture. As shown in Figure 3.1 the system components can be divided into three main levels:

1. **Optimal Trajectory Generation:** This layer is concerned by the generation of the quadrotors trajectory from a departure point P_i to the arrival point P_f by choosing the optimal reference path, and avoid any presented obstacles. This first layer provide the second layer by the desired path (x_d, y_d, z_d) to track.

2. **Optimal Controller:** This layer aims to design an optimal controller able to track the desired bath generated by the first layer, two controllers are designed and compared, a linear LQR (Linear Quadratic Regulator) controller and a non-linear Optimal Backstepping (OBS) to allow the quadrotors to track the generated trajectory with a high accuracy and a minimum of energy. The OBS controller parameters are derived using the Multi-Objective Genetic Algorithms (MO-GA) which optimize the energy via several cost functions from several levels of the controller.

3. **The Control Architecture:** The quadrotors control architecture is based on a double loop control strategy, an inner loop for the attitude control and outer loop for the position control. An optimization over the control space is reached by using a differential flatness- quaternion based equations.

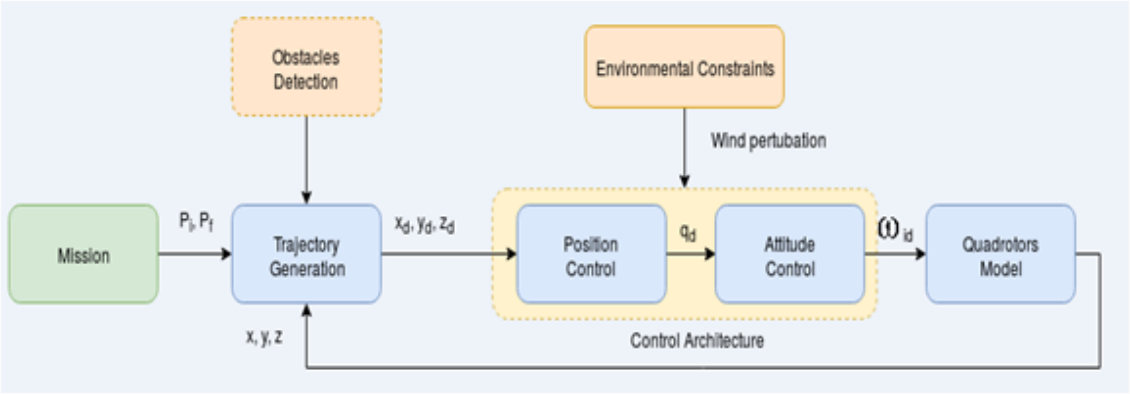

Figure 3.1: Multi-layer Optimal Navigation System

3.3 Trajectory Tracking control

3.3.1 Control Architecture

Quadrotors have been shown to be a differential system with 4 at outputs. These at outputs are the inertial position of the vehicle, x, y, and z, and the yaw angle ψ. All of the quadrotors states can be written as a function of these four flat outputs and their derivatives. These states include the position, velocity, and acceleration of the vehicle's center of mass, as well as the orientation, rotational velocity, and rotational acceleration of the vehicle. [6]

$$\chi = \begin{bmatrix} x \\ y \\ z \\ \psi \end{bmatrix} \tag{3.1}$$

Assuming zero yaw angle for simplicity, then the mapping from the at outputs to the position, velocity, and acceleration of the quadrotors center of mass expressed in the inertial coordinate frame is trivial, as shown in Equation 3.2

$$\begin{aligned} \begin{bmatrix} x & y & z \end{bmatrix}^T &= \begin{bmatrix} \chi_1 & \chi_2 & \chi_3 \end{bmatrix}^T \\ \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^T &= \begin{bmatrix} \dot{\chi_1} & \dot{\chi_2} & \dot{\chi_3} \end{bmatrix}^T \end{aligned} \tag{3.2}$$

The mapping from the flat outputs to the quadrotors quaternion orientation can be derived using fundamental quaternion principles and the inherent properties of a standard quadrotors. A quaternion orientation can be formulated as a rotation about some axis $\widehat{n}$, as shown in Equation 3.3 This representation will be useful in defining the vehicle orientation in terms of the flat outputs.[6]

$$q = \begin{bmatrix} cos(\frac{\theta}{2}) \\ \widehat{n} sin(\frac{\theta}{2}) \end{bmatrix} \tag{3.3}$$

Since a typical quadrotor can only produce a thrust force along its body z axis, this axis will always be aligned with the inertial acceleration vector with an arbitrary yaw angle. The final orientation is then described by a yaw rotation about the inertial acceleration vector, which is collinear with the body z-axis. For the derivation of the orientation equations from the flat outputs, the normalized thrust vector in the body frame and the inertial frame will be defined as F^B and F^I respectively. The normalized body frame thrust vector is always $[0; 0; 1]^T$ and the normalized inertial frame thrust vector is defined in Equation 3.4

$$\widehat{F}^I = \frac{1}{\sqrt{\ddot{x}^2 + \ddot{y}^2 + (\ddot{z}^2 - g)}} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} - g \end{bmatrix} \tag{3.4}$$

The quadrotor's orientation can be found by calculating the quaternion rotation required to match the direction of the body frame thrust vector to that of the inertial frame thrust vector and correcting for the yaw angle. The rotation vector $\widehat{n}$ is solved for in Equation 3.5 [6]

$$\begin{aligned}\widehat{F}^B.\widehat{F}^I &= ||\widehat{F}^I||||\widehat{F}^B||cos(\theta) = cos(\theta)\\ \widehat{F}^B \times \widehat{F}^I &= ||\widehat{F}^I||||\widehat{F}^B||sin(\theta\widehat{n}) = sin(\theta\widehat{n})\\ \widehat{n} &= \frac{\widehat{F}^B \times \widehat{F}^I}{sin(\theta)} = \frac{\widehat{F}^B \times \widehat{F}^I}{\sqrt{1-cos^2(\theta)}} = \frac{\widehat{F}^B \times \widehat{F}^I}{\sqrt{1-\widehat{F}^{B^T}.\widehat{F}^I}}\end{aligned} \tag{3.5}$$

The equations for the half angle cosine and sine are shown in Equation 3.6 and Equation 3.7, respectively.

$$cos(\frac{\theta}{2}) = \sqrt{\frac{1}{2}(1+cos(\theta))} = \sqrt{\frac{1}{2}(1+\widehat{F}^{B^T}.\widehat{F}^I)} \tag{3.6}$$

$$sin(\frac{\theta}{2}) = \sqrt{\frac{1}{2}(1-cos(\theta))} = \sqrt{\frac{1}{2}(1-\widehat{F}^{B^T}.\widehat{F}^I)} \tag{3.7}$$

The rotation vector and half-angle sine and cosine can now be substituted into Equation 3.3, resulting in the quaternion rotation without a yaw correction. This quaternion will be denoted $\widetilde{q}$ and is shown simplified in Equation 3.8 The quaternion orientation $\widetilde{q}$ can be corrected for yaw with Equation 3.9, resulting in the final vehicle orientation q. [6]

$$\widetilde{q} = \frac{1}{\sqrt{2(1+\widehat{F}^{B^T}.\widehat{F}^I)}}\begin{bmatrix}1+\widehat{F}^{B^T}.\widehat{F}^I\\ \widehat{F}^B \times \widehat{F}^I\end{bmatrix} \tag{3.8}$$

$$q = \widetilde{q} \otimes \begin{bmatrix}cos(\frac{\psi}{2})\\ 0\\ 0\\ sin(\frac{\psi}{2})\end{bmatrix} \tag{3.9}$$

Finally, a possible conversion from Euler angles to quaternion can be performed using the following equation:

$$q_d = \begin{bmatrix}cos(\frac{\varphi_d}{2})cos(\frac{\theta_d}{2})\\ sin(\frac{\varphi_d}{2})cos(\frac{\theta_d}{2})\\ cos(\frac{\varphi_d}{2})sin(\frac{\theta_d}{2})\\ -sin(\frac{\varphi_d}{2})sin(\frac{\theta_d}{2})\end{bmatrix} \tag{3.10}$$

3.3.2 Optimal Controllers Design

The trajectory tracking controller consists of two parts, an inner control loop namely attitude controller and an outer one namely the position controller as depicted in Figure 3.2. The outputs of the attitude controller are the desired angular speeds. The position controller generates the desired position quaternion value q_d and the desired trust T_d for the attitude controller.

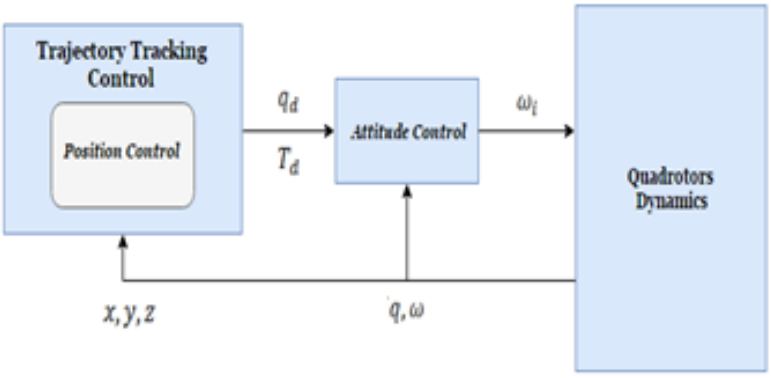

Figure 3.2: Block diagram of the prposed control structure

The state space model of the simplified quadrotors attitude and position is given by Equation 3.11 and Equation 3.12, respectively. The attitude state vector is $x_A = \begin{bmatrix} \overline{q} & \Omega \end{bmatrix}$ and the position state vector is $x_P = \begin{bmatrix} p & \dot{p} \end{bmatrix}$. [3]

$$\dot{x_A} = \begin{bmatrix} 0_{3\times3} & 0.5I_{3\times3} \\ 0_{3\times3} & 0_{3\times3} \end{bmatrix} x_A + \begin{bmatrix} 0_{3\times3} \\ J^{-1} \end{bmatrix} \begin{bmatrix} \tau_{ux} \\ \tau_{uy} \\ \tau_{uz} \end{bmatrix} \tag{3.11}$$

$$\dot{x_P} = \begin{bmatrix} 0_{3\times3} & I_{3\times3} \\ 0_{3\times3} & 0_{3\times3} \end{bmatrix} x_P + \begin{bmatrix} & 0_{3\times3} & \\ g & 0 & 0 \\ 0 & -g & 0 \\ 0 & 0 & \frac{1}{m} \end{bmatrix} \begin{bmatrix} u_{xd} \\ u_{yd} \\ T_d \end{bmatrix} \tag{3.12}$$

For both controllers the LQR command is used because it's a method to find the optimum solution for a problem of minimization that assures the system stability in close-loop, in addition its calculation is easy.

The Equation 3.13 represents the quadratic cost function to minimize:

$$J(x,u) = \int (x^2(t)Q(t) + u^T Ru(t))dt \tag{3.13}$$

Controller	K1	K2
Attitude	[0.02 0.02 6.25]	[0.16 0.16 0.25]
Position	[5.6 5.6 3.16]	[0.15 0.15 0.9]

Table 3.1: LQR Control Parameters

R and Q are weight matrix used respectively in order to increase or to diminish the effect of the states and the entrances of individual form and are select for the designer in agreement with the required performance.

The optimum input is defined as $U = -Kx$ With $K = R^{-1}B^TP$ and P is the solution to the Ricatti differential equation given by:

$$A^TP + PA - PBR^{-1}B^TP + Q = 0 \tag{3.14}$$

The obtained control parameters for the LQR controller for the attitude and position control are shown on Table 3.1

The Lyapunov function V_A used to design a backstepping attitude controller is given by Equation 3.15, where q_{e0} is the quaternion error and e_2 denotes angular velocity error $\eta_d - \eta$.

$$V_A = |q_{e0}| + \frac{1}{2}e_2^Te_2 \tag{3.15}$$

The derivative of the Lyapunov function V_A is expressed in Equation 3.16

$$\dot{V}_A = -\frac{1}{2}sign(q_{e0})q_{e13}^T\eta_d + e_2(\eta_d - \eta) \tag{3.16}$$

Equation 3.17 shows the desired derivative of the Lyapunov function $\dot{V}_A$, which is negative as long as c_{1A} is a positive constant and matrix c_{2A} is a positive definite matrix.

$$\dot{V}_A = -\frac{1}{2}c_{1A}q_{e13}^Tq_{e13} - e_2^Tc_{2A}e_2 \tag{3.17}$$

Assuming the virtual control η_d is expressed by Equation 3.18 then the control law τ is expressed by formula Equation 3.19

$$\eta_d = sign(q_{e0})c_{1A}q_{e13} \tag{3.18}$$

$$\tau = (c_{2A}e_2 + \dot{\eta}_d)J + \tau_{ext} \tag{3.19}$$

The Lyapunov function V_P used to design the trajectory tracking backstepping controller is given by Equation 3.20, where $e_1 = p_d - p$ is the position error and $e_2 = v_d - \dot{p}$ is the velocity error.

$$V_P = \frac{1}{2} e_1^T e_1 + \frac{1}{2} e_2^T e_2 \tag{3.20}$$

Given the desired velocity as Equation 3.21 and the desired derivative of the Lyapunov function $\dot{V}_P$ as Equation 3.22 that is negative as long as c_{1P} and c_{2P} are positive definite matrices, then the control law u_P is derived from Equation 3.23

$$v_d = c_{1P} e_1 + \dot{p}_d \tag{3.21}$$

$$\dot{V}_P = -e_1^T c_{1P} e_1 - -e_1^T c_{2P} e_1 \tag{3.22}$$

$$\ddot{p} = e_1(I - c_{1P}^2) + e_2(c_{1P} + c_{2P}) + \ddot{p}_d \tag{3.23}$$

Basically a multi-objective optimization problem has more than one objective function, in engineering problems usually two or more objectives, to be optimized.

The multi-objective optimization problems may also have one or more constraints including inequality, equality and/or variable bounds to be satisfied. However in real engineering applications usually more than one constraint is involved in the problem. A general formulation of a multi-objective optimization problem is defined as follows: [7]

$$\begin{array}{cc} min/max f_m(x) & \text{m} = 1,2,...,\text{M} \\ s.t. g_j(x) & \text{j} = 1,2,...,\text{J} \\ h_k(x) & \text{k} = 1,2,...,\text{K} \\ x_i^L \leq x_i \leq x_i^U & \text{i} = 1,2,...,\text{n} \end{array} \tag{3.24}$$

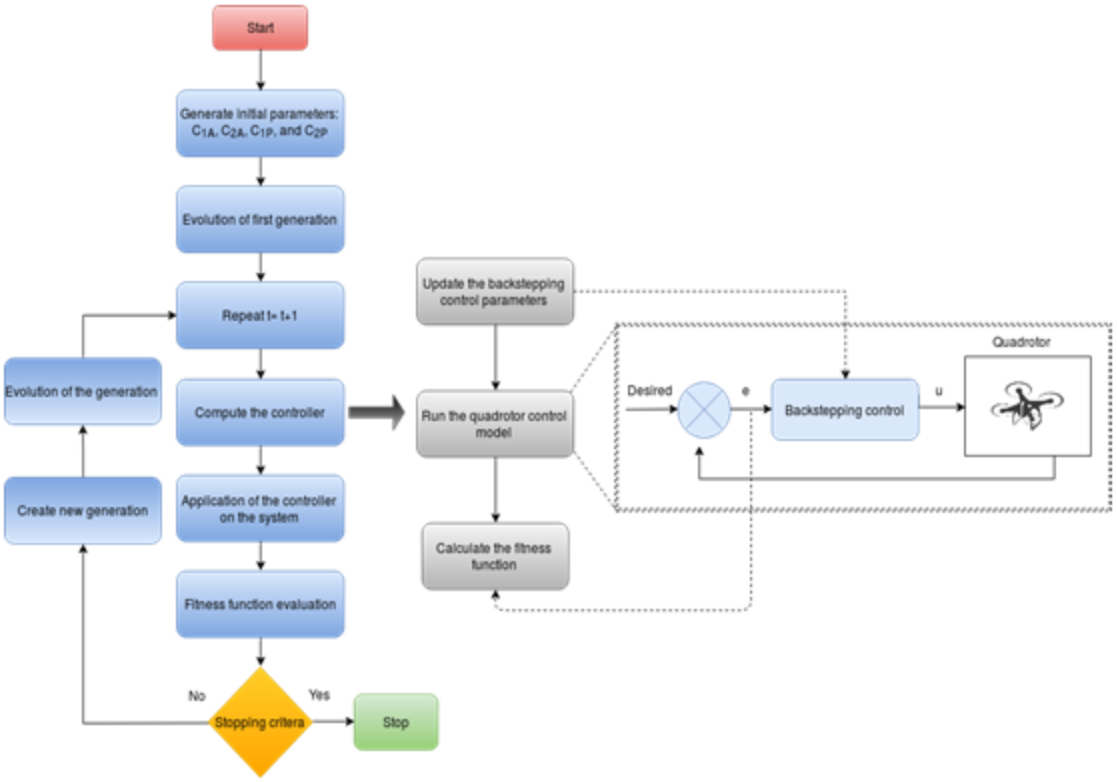

Figure 3.3: MO-GA Optimization Algorithm

In the present study a double loop control strategy is used to stabilize the attitude and track the desired position. The coefficients matrices c_{1A}, c_{2A}, c_{1P} and c_{2P} are the control parameters and need to be positive to satisfy stability criteria. In conventional backstepping method, these parameters are selected randomly or properly but not sure to be optimal. To overcome this drawback, this research adopts the MO-GA algorithms for selecting the optimal value of the backstepping control parameters. The MO-GA is utilized offline to determine the backstepping controller parameters as shown in Figure 3.3.

The performance of the controller varies according to adjusted parameters. Since the optimal backstepping (OBS) control aims to improve the control performance yielded by a backstepping controller and optimize the energy consumption, it keeps the simple structure of the backstepping controller. As this controller is based on a two sub-controllers (Attitude and position) the problem is then divided into two OBS controllers: the attitude controller with eight (8) control parameters and the position controller with four (4) control. All the control parameters need to be selected simultaneously so that each subsystem is asymptotically stable. The obtained values must ensure the optimal energy and an acceptable behavior of the system time response.

Controller	c1	c2
Attitude	[0.8 0.8 0.5]	[0.1 0.1 1.2]
Position	[0.2 0.2 3]	[0.1 0.1 1.5]

Table 3.2: OBS Control Parameters

In this chapter, an integral absolute error (IAE) is utilized to judge the performance of the controller. IAE criterion is widely adopted to evaluate the dynamic performance of the control system. The index IAE is expressed as follows:

$$IAE = \int_0^t |e(t)|dt \tag{3.25}$$

The obtained control parameters for the OBS controller using the MO-GA for the attitude and position control are shown on Table 3.2

3.4 Trajectory Optimization and Obstacles Avoidance

To determine the optimal reference trajectory, typically a optimization within the control space is performed subject to some constraints placed within the output space and the state space. These constraints include physical constraints, actuator constraints and obstacle avoidance constraints.

From the differential flat equations the problem can be reposed to allow optimizations to occur within the output space as opposed to the control space. This is beneficial because constraints such as obstacle avoidance occur in the output space, hence the computation time for constraint handling is reduced. The problem is posed as follows: [5]

$$\begin{aligned} min\Phi for \quad & t \in [0,T] \\ y(t) \quad & \\ & s.t.c_y(y) \leq 0 \\ & x_0 - h_(y(0)) = 0 \\ & y_T - y(T) = 0 \end{aligned} \tag{3.26}$$

Where the inequality constraints are now expressed as a function of the output $c_y(y)$ and the state is now a function of the output obtained from the differential flatness $h_1(y)$.

The objective function,Φ, is a quantitative measure of the optimality of the trajectory, which, can be approximated by a measure of the running costs. Assuming running costs are proportional to average velocity then the objective function can be defined as:

$$\Phi = \int_0^T \sqrt{P_1\dot{x}^2 + P_2\dot{y}^2 + P_3\dot{z}^2}dt \tag{3.27}$$

Where P_1, P_2 and P_3 are weighting factors [5].

To prevent UAVs from destroying themselves and surrounding objects (other vehicles, personnel, and infrastructures), a collision avoidance mechanism should be generally incorporated into the control design [5].

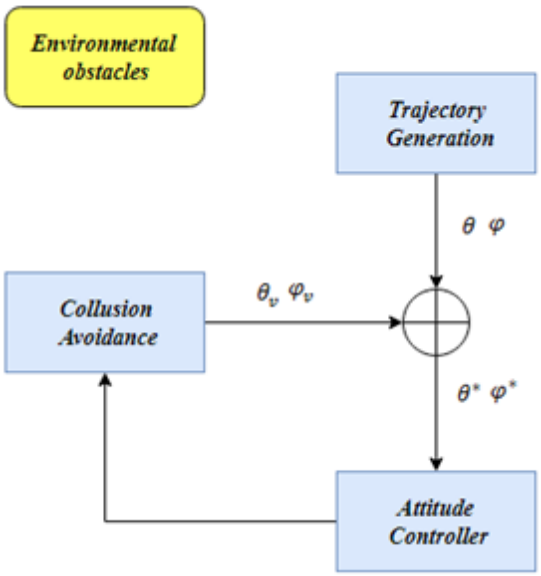

Figure 3.4: Obstacle avoidance algorithm

As illustrated in Figure 3.4, the primary idea of this technique is to modify the roll and the pitch angles while keeping the separation between the UAV and the obstacle. The essence of this technique can also be mathematically expressed as follows:[4]

$$\begin{aligned} \phi^* &= \phi + \phi_v \\ \theta^* &= \theta + \theta_v \end{aligned} \tag{3.28}$$

Where ϕ^* and θ^* denote the ultimately desired roll and pitch angles after modification, ϕ_v and θ_v represent the reactive obstacle avoidance terms for roll and pitch angles of each UAV.

3.5 Simulation Results

In this section the simulation results related to the quadrotors optimal trajectory generation and control discussed in the other sections is shown.

For all the simulation cases both controller (LQR and OBS) are applied and compared. The controller was simulated at a rate of 200 Hz which makes it suitable for a real implementation. All the necessary limitations over the actuators and energy consumption were taking into consideration.

Five scenarios for different situations of the quadrotors drone have carried out as follow:

1. **Scenario 1** The quadrotors starts from an initial position and maintain its position in another points until the battery is discharged.
2. **Scenario 2** In this case the quadrotors is tracking a circular path in the presence of an external wind gust disturbance.
3. **Scenario 3** For this scenario the quadrotors is facing a sudden engine failure for a few seconds.
4. **Scenario 4** During this case the quadrotors is scanning a long range area at a constant altitude.
5. **Scenario 5** The quadrotors UAV is facing a circular obstacle so it had to avoid it then continue the desired mission.

3.5.1 Scenario 1

As mentioned before, in this case the quadrotors from a point $P_0(x, y, z) = (0, 0, 0)$ and travel to another point $P_1(x, y, z) = (1, 1, 1)$ and hold it position until the battery is discharged. This first scenario was chosen to test the optimal trajectory generation and the energy optimization.

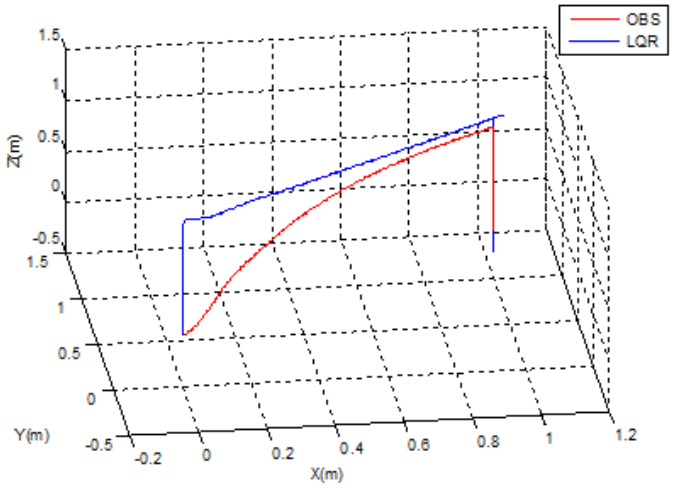

Figure 3.5: Trajectory for scenario 1

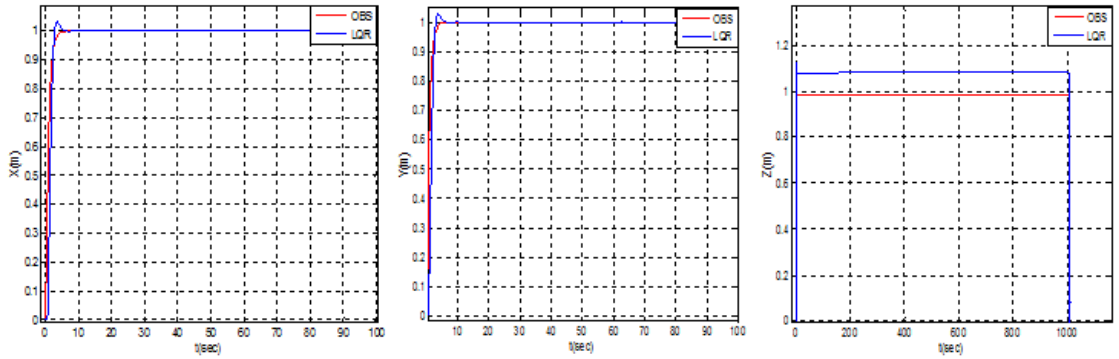

Figure 3.6: Position Tracking for scenario 1

Figure 3.5 Shows the obtained results for the trajectory generation of the LQR and the OBS controller where it is clear that the OBS controller is following a more optimal 3D path and presents a more accurate control when comparing to the LQR controller (overshoots in the X-Y plan). For the emergency landing due to the battery discharging the LQR controller is more energy saving technique due to the low magnitude in the control inputs (see Figure 3.6) but both controllers were able to execute a safe emergency landing, and prevent the quadrotors from the destruction.

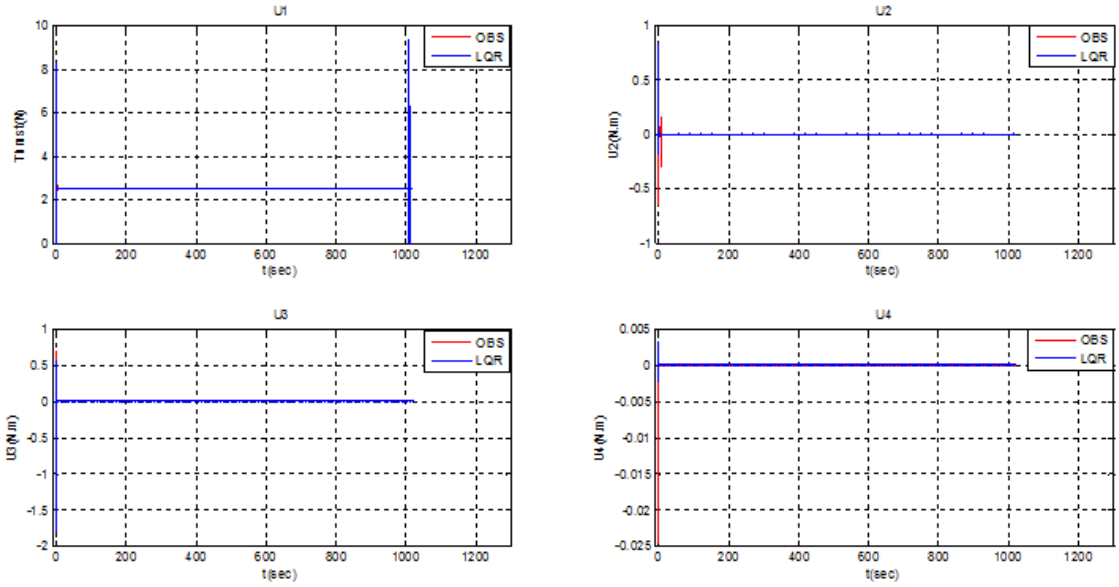

Figure 3.7: Control Inputs for scenario 1

3.5.2 Scenario 2

In this scenario the quadrotors UAV is sacked, in a first case, to follow a circular path of a 1m diameter and at 1m of altitude. In the second case the quadrotors is tracking the same previous path but in the presence of a wind gust perturbation. This kind of scenarios allows testing the robustness of the controllers. The obtained results for the first case are shown in Figure 3.8

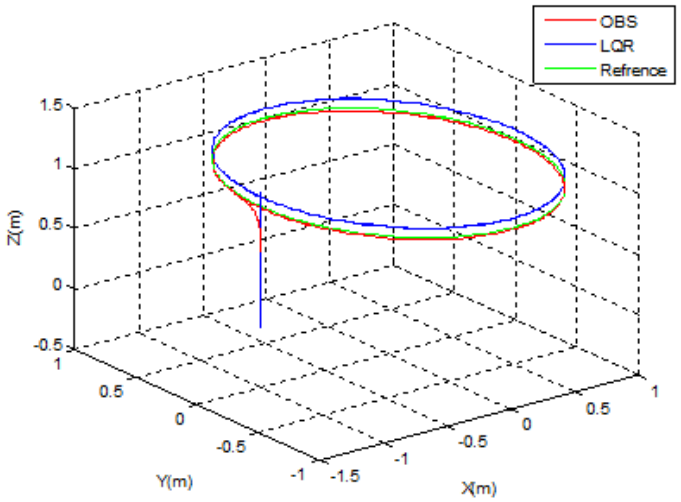

Figure 3.8: Trajectory tracking for scenario 2

From Figure 3.8 both controller were able to track the desired path but the OBS controller is estimated to be more accurate.

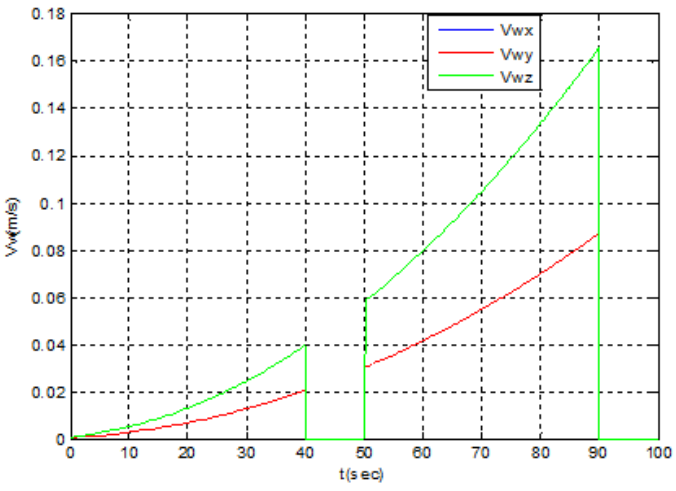

Figure 3.9: Wind gust velocity profile

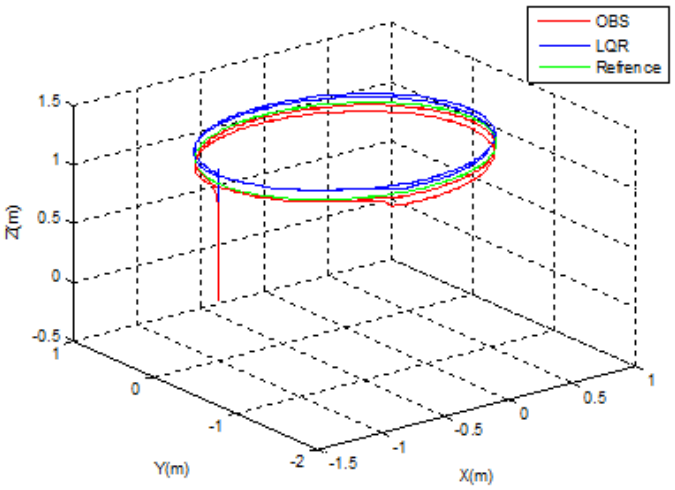

Figure 3.10: Trajectory Tracking in the presence of a wind gust

In the second case the quadrotors is facing a wind gust. The perturbation is applied twice a time for a 40 sec during the simulation, a recovery period of 10 sec is given to the quadrotrs to stabilize. Figure 3.9 illustrates the wind gust velocity profile used in the simulation.

Figure 3.10 and Figure 3.11 introduce the results obtained during the trajectory tracking in the presence of the wind gust perturbation. The LQR and OBS controller were able to stabilize the quadrotors and accomplish the missions but with a better accuracy for the OBS controller in the X-Y plan and a good one of the LQR controller in the altitude hold.

For the control inputs shown in Figure 3.12 the OBS controller presents a more aggressive maneuvers in each time the perturbation starts of finish but with a less magnitude when compared to the LQR control inputs.

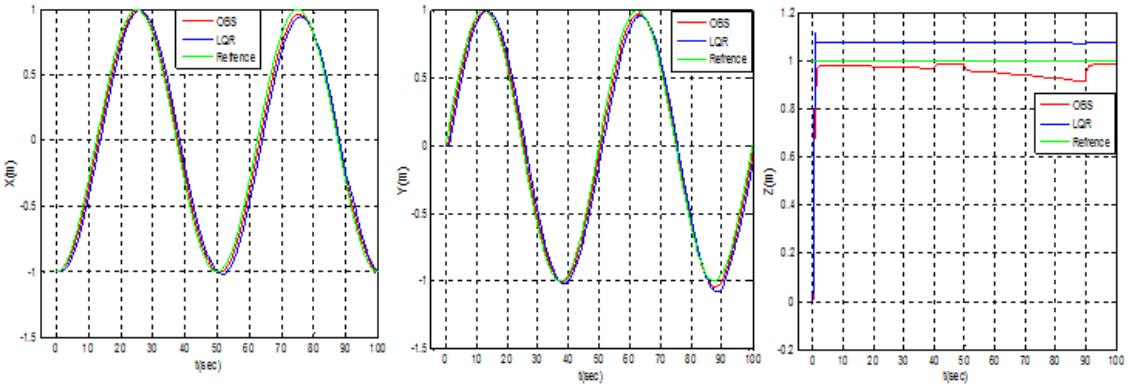

Figure 3.11: Position Tracking for scenario 2

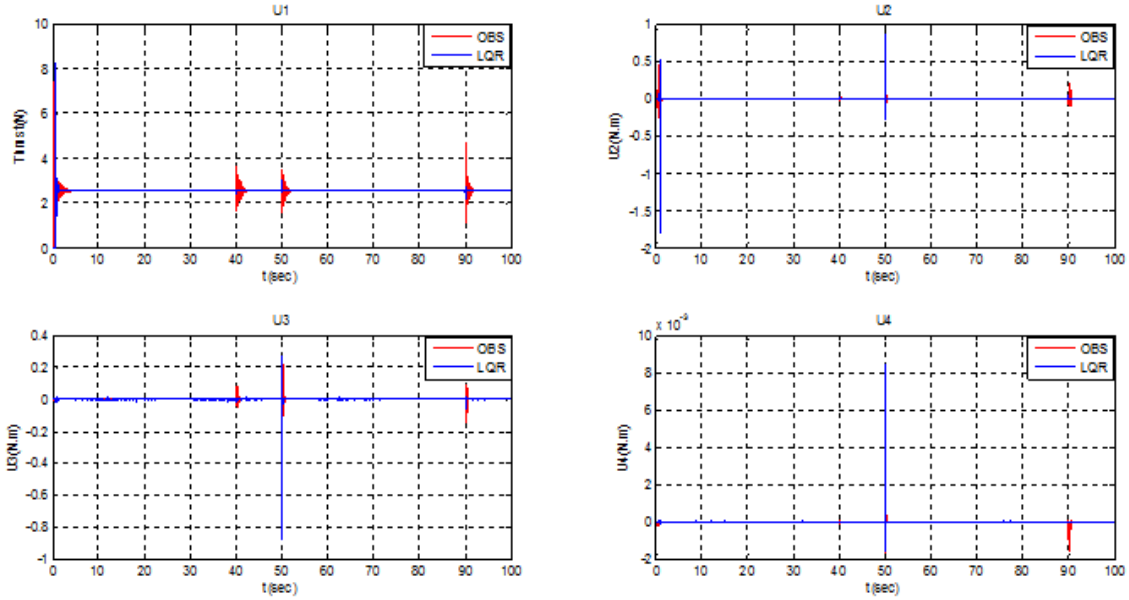

Figure 3.12: Control Inputs for scenario 2

3.5.3 Scenario 3

This scenario is dedicated to simulate the attitude stability of the quadrotors after a sudden engine failure of the number 01 engine. The fault starts from the 15 sec and still for 3 sec, the UAV is supposed to be at 10m of altitude.

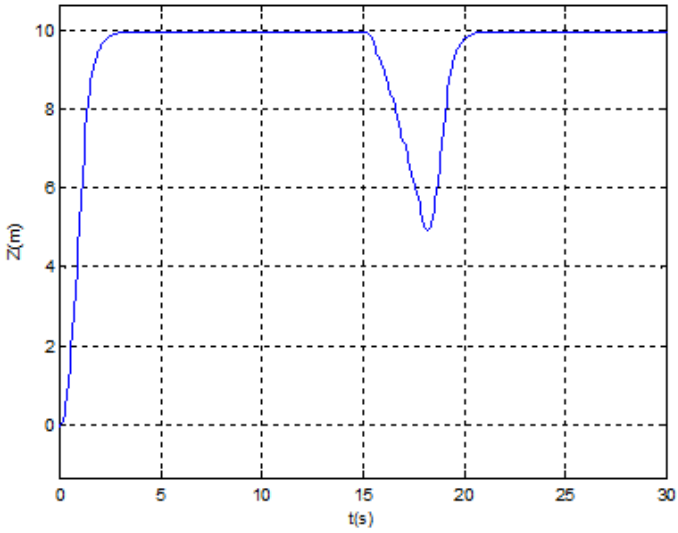

Figure 3.13: Altitude response for scenario 3

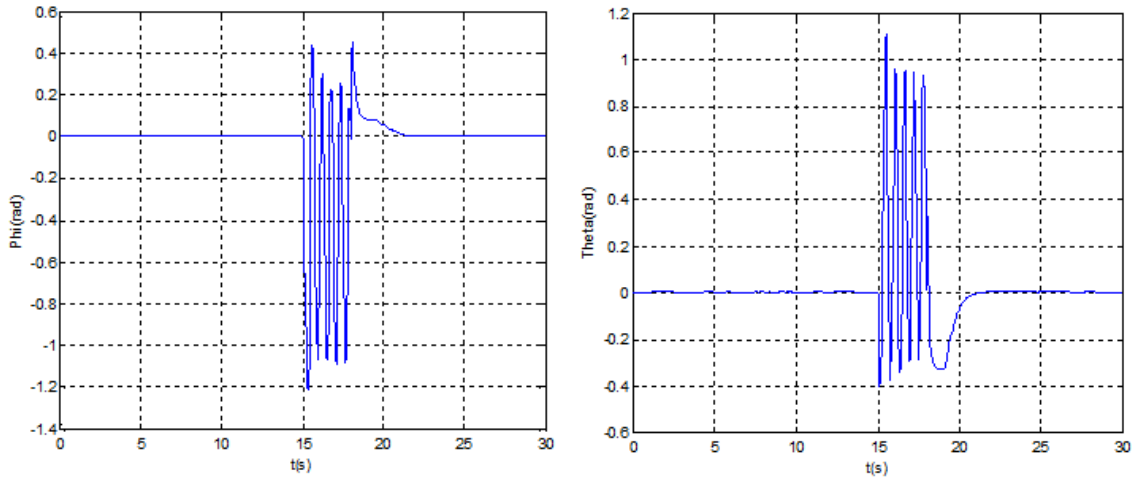

Figure 3.14: Attitude response for scenario 3

For the scenario the LQR controller was not able to execute this task, and the quadrotors was not able the recover its altitude so it crashes.

All the results are using only the OBS controller. Figure 3.13 indicates the altitude response of the quadrotors, while Figure 3.14 and Figure 3.15 show the attitude response and control inputs respectively.

From the obtained results it can be noticed that the quadrotors was able to recover its altitude and stabilize after the sudden engine failure even with some lose of the altitude (go down to less than 6 m).

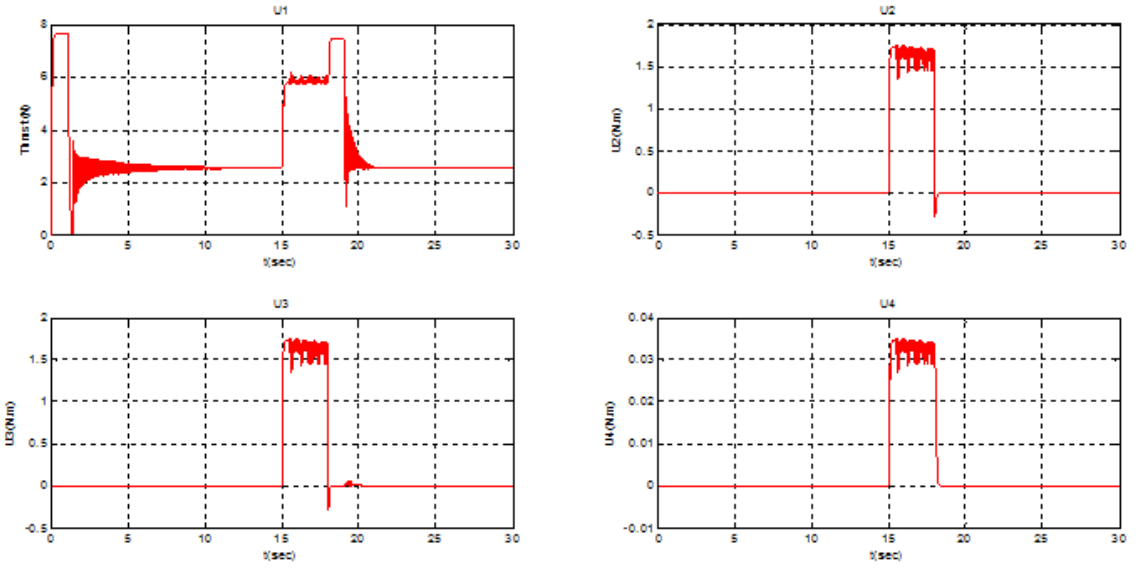

Figure 3.15: Control Input for scenario 3

As mentioned before the OBS controller was the only controller to execute this task, this was due to its ability of big attitude degrees tracking (up to 1 rad as shown in Figure 3.14) using an aggressive control inputs with high rate changes (Figure 3.15) that cannot be generated using a linear LQR controller.

3.5.4 Scenario 4

For this scenario the quadrotors is scanning a large area by tracking a rectangular shape (1000 m * 1000 m) at 10m of altitude, the goal behind this scenario is to test the energy optimization of the used controllers during the trajectory tracking.

Figure 3.16 and Figure 3.17 show the obtained 3D trajectory tracking. It is clear that the OBS controller presents a more accurate results compared to the LQR controller, this is due to the high attitude degree (up to 1 rad) used by the OBS controller during the curving movements (See Figure 3.18) but with less control inputs magnitude (about 3N) during rectilinear movements (Figure 3.19).

The total energy consumption is estimated to be less for the OBS controller compared to the LQR controller.

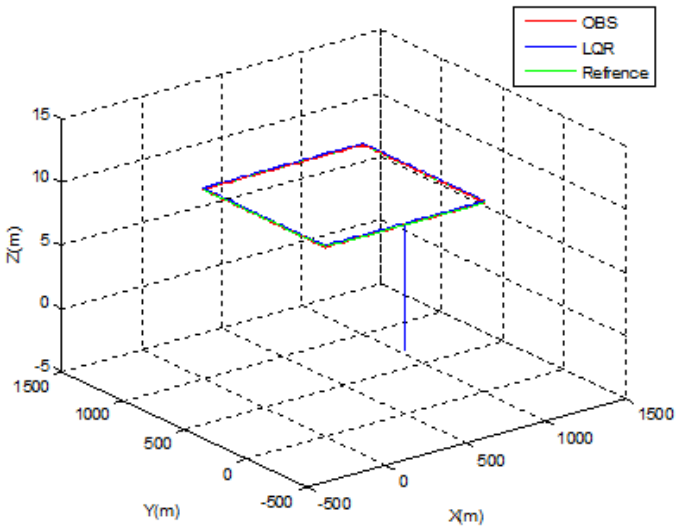

Figure 3.16: Trajectory tracking for scenario 4

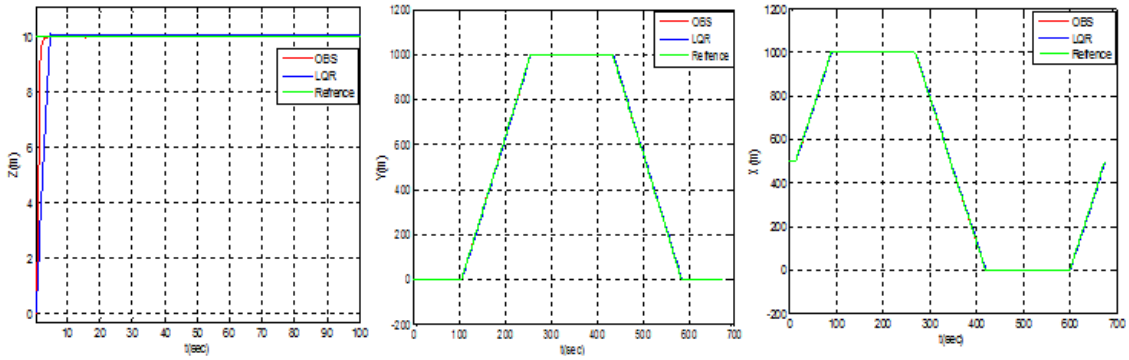

Figure 3.17: Position tracking for scenario 4

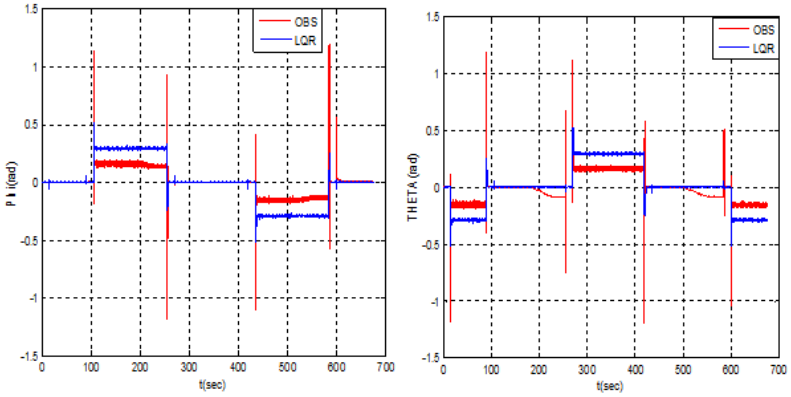

Figure 3.18: Attitude response for scenario 4

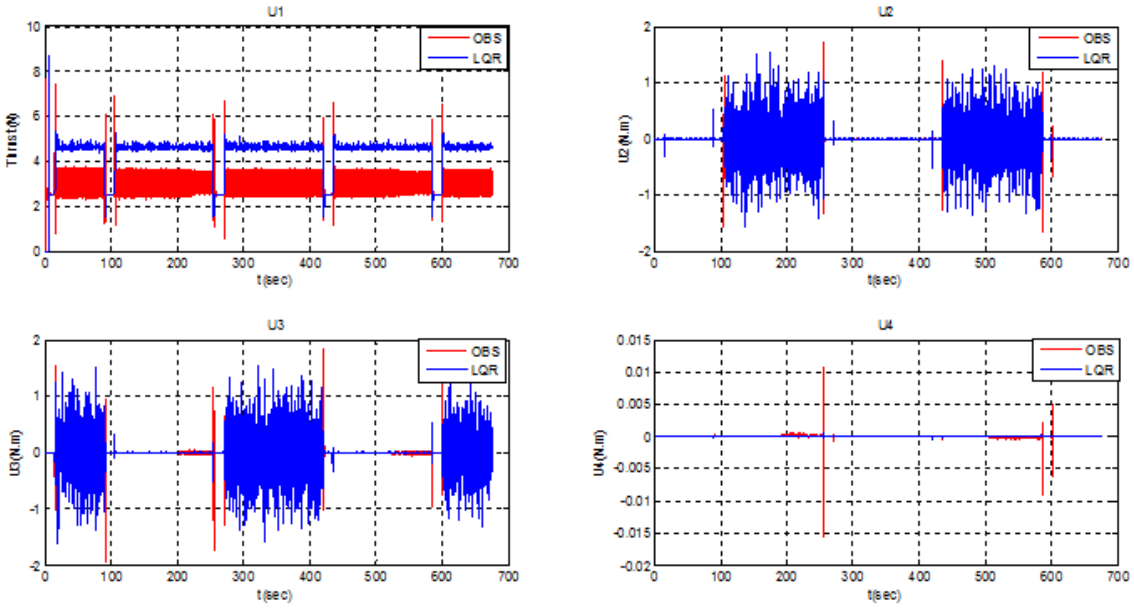

Figure 3.19: Control Inputs for scenario 4

3.5.5 Scenario 5

For this last scenario the quadrotors is facing a circular obstacle after starting from an initial point $P_0(x, y) = (50, 35)$ and traveling to another point $P_1(x, y) = (51, 65)$ in the horizontal plan.

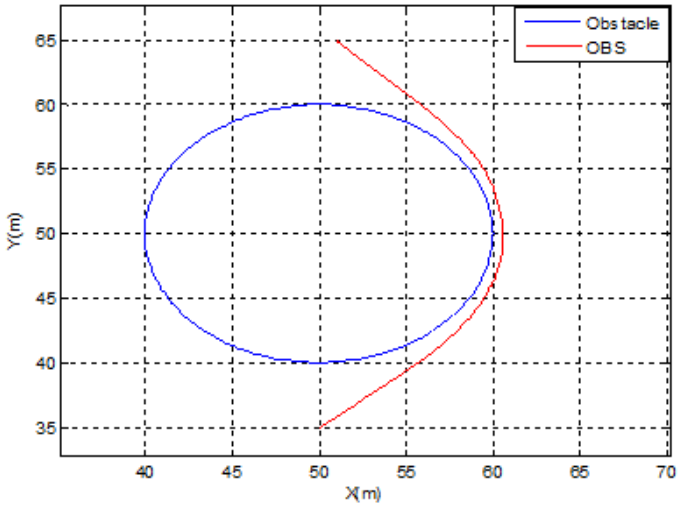

Figure 3.20: Obstacle avoidance

It is important to mention that the LQR controller was not able to execute this task, so all the results are using only the OBS controller.

From Figure 3.20 it can be noticed that the quadrotors was able to track an optimal trajectory to reach the desired destination, and avoid the collision with the obstacle.

Figure 3.21 and Figure 3.22 present the obtained attitude and the control inputs. Those results reflect the high performance of the OBS controller to avoid the obstacle with high accuracy and a minimum of energy during a short time.

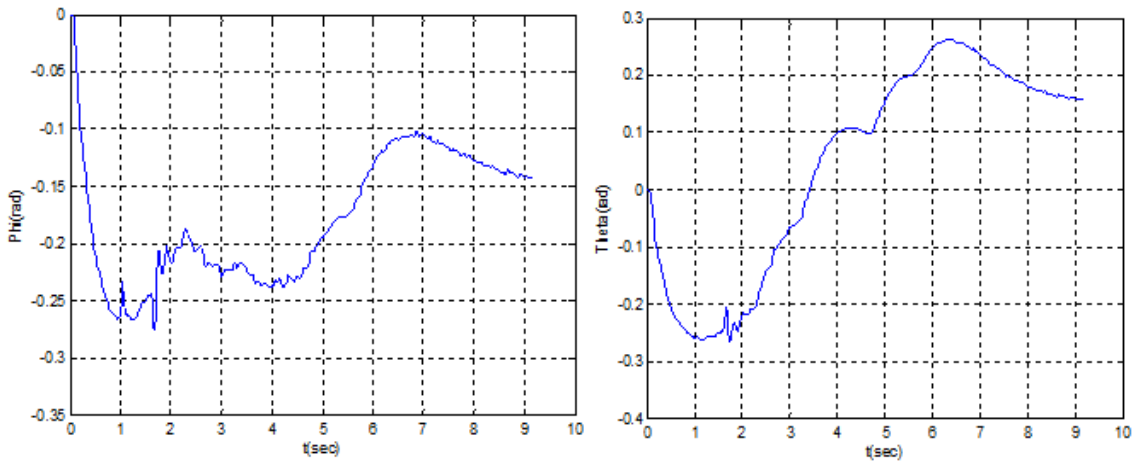

Figure 3.21: Attitude response for scenario 5

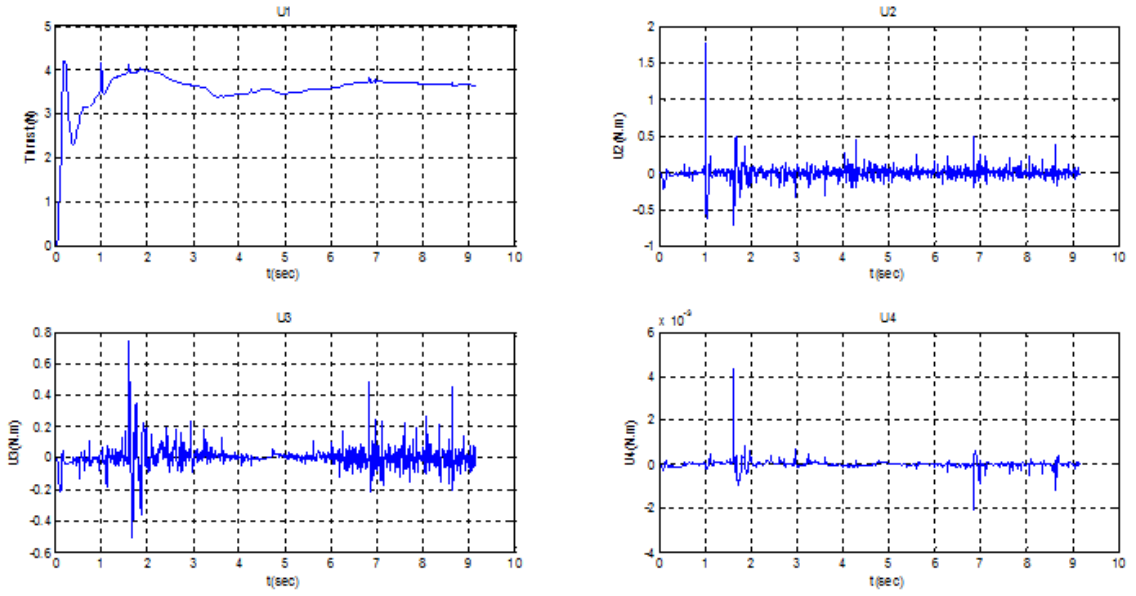

Figure 3.22: Control Inputs for scenario 5

3.5.6 Control Strategies Comparison

In this section a comparison between the different control strategies is made, the obtained results are shown in Table 3.3

Scenario/Controller	**Fitness Function (IAE)**	**Battery Consumption (%)**	**Mission Time (sec)**
Scenario 1 LQR OBS	 0.0867 0.0234	 100 100	 1020 1020
Scenario 2 Case 1: LQR OBS Case 2: LQR OBS	 0.0775 0.0295 0.1215 0.0718	 30.01 28.40 26.08 24.22	 100 100 100 100
Scenario 3 LQR OBS	 - -	 - 8.4	 - 30
Scenario 4 LQR OBS	 0.1026 0.0145	 35.74 33.40	 675 675
Scenario 5 LQR OBS	 - -	 - 5.6	 - 9.26

Table 3.3: Control Strategies Comparison

From Table 3.3 it is clear that the OBS controller is the only controller that was able to execute all the proposed scenarios (The LQR controller was not able to avoid the obstacle and to recover the stability after an engine failure).

During all the scenarios the energy optimization of the controllers is improved, since the consumed energy is estimated to be optimal compared to the mission time (The quadrotors is able to still for 17 min at a hovering movement before the emergency landing – mission 1 -) spatially for mission 4 where the quadrotors was able to scan a large area with only 35

For the controllers tracking accuracy, the OBS is clearly the most accurate controller with a very interesting IAE coefficients (less than 0.1 during all the scenarios) even when the UAV is facing a wind perturbation (0.1215 for LQR and 0.0718 for OBS) which proof the controllers robustness.

3.6 Conclusion

In this chapter the problem of the quadrotors optimal trajectory tracking and control was studied using a new scheme based on a multi-layer optimal system.

A combination of a double loop control structure with an inner attitude loop and an outer position loop based on the differential flatness approach is used in order optimizes the output spaces and gives them a direct relation with the states and their derivatives.

Many scenarios were proposed in order to test the performances and the robustness of the controllers used within this work, all the obtained results were judged to be satisfactory since the quadrotors UAVs have successfully followed the generated path and eliminate the effects of the external disturbances, but the OBS controller still the most effective controller to use since it is optimal, accurate and robust.

Chapter

4 Multi-Agents Quadrotors Formation Control

Contents

4.1 Introduction

In the last recent years, multi-agents formation control problems have been widely investigated among the research community. Compared with a single UAV, a group of collaborative UAVs can fulfill more difficult tasks and accomplish complex objectives. Different strategies and architectures have been proposed in the literature, such as behavior-based [62], virtual structure [18], potential field [19] and leader-follower [27, 31, 32]. In the centralized leader-follower (L-F) scenario, one of the agents designated as "leader" possesses the reference motion to be tracked by the other agents "followers". In order to act cooperatively, the leader spreads its states among the rest of the swarm by means of proper communication link, Thus, any mere failure of the leader will lead to failure in the whole mission. .

In a formation control, quadrotors are not physically coupled. However, their relative motions are strongly constrained to keep the formation. In order to achieve formation control, consensus algorithms for multi-agent systems have been extensively studied in the literature [13, 27, 30, 31, 40, 64, 68, 74, 77]. Many control mechanisms were used to hold the formation topology. In effect, the theory of multiple UAVs formation control can be found in [31]. References [30, 64] propose a second-order consensus algorithm to follow a predetermined external reference, while [27, 64, 68] describes the formation control problem as a position control problem to be solved. The precedent control techniques were able to maintain the formation, whereby an estimation of the position for the leader as well as the followers was required. On the other hand, an attitude control technique is used for spacecraft formation vehicles such as in [74, 77] where robust attitude coordinated control is used. For quadrotors, reference [40] proposes a transformation control technique to convert the position control to an attitude control problem, formation attitude stability is then assured using a backstepping controller.

One of the recent investigated problems within the leader-follower scenario is the leader election. For most cases the leader is supposed to be a particular agent chosen at the beginning of the task, this event is called static leader election. In [9] a novel leader election method is proposed using an adaptive / reliable network structure. The work in [61] proposes a distributed leader election without direct inter-agent communication. For online leader election, the authors in [25] propose a fully-decentralized adaptive strategy able to periodically select online the best leader among the neighbors of the current leader online.

In order to operate safely and to accomplish mission tasks, one of the important criteria required for the UAVs is the ability to avoid collisions with other member of swarms and environmental obstacles. A survey of UAVs obstacles avoidance is presented in [56]. Paper [38] proposes two efficient algorithms: conflict detection (CD) algorithm and conflict resolution (CR) algorithm for cooperative multi-UAV collision avoidance system. The work in [76] proposes modified tentacle formation flight and collision avoidance algorithm for multiple UAVs in unstructured environments, while [11] developed an autonomous navigation and avoiding obstacles along

the trajectory without any pilot inputs in outdoor environment. In [58], the authors presented a directional collision avoidance with obstacles in swarming applications through the implementation of relative position based on cascaded PID position and velocity controllers, while [18] deals with a behaviour-based decentralized control strategy for UAV swarming by using artificial potential functions and sliding mode control technique. While the previous cited papers were able to deal with the obstacle avoidance problem within a swarm of UAVs, no one has optimized the generated trajectory.

This chapter introduces a new framework for multi-agents quadrotors formation that overcomes the drawbacks of standard leader-followers formation and provides better performance. As such, the designed framework includes the trajectory generation, formation control, obstacles avoidance and leader election algorithms for distributed leader-followers architecture. To determine the optimal reference trajectory the differential flatness method is used, optimization within the control space is performed subject to some constraints placed within the output space and the state space. These constraints include physical constraints, actuator constraints and obstacle avoidance constraints.

A consensus-based attitude control is used for the formation control; this allows the formation topology to be maintained with a minimum of a sharing data, and makes the controller more robust to any external disturbances. Furthermore both of trajectory tracking and formation control algorithms are based on a double loop control structure with backstepping/SMC controller.

Moreover, this paper introduces a new leader-follower paradigm in which: i) the agents can elect the new leadership in order to adapt to a leader sudden failure or a variation in the graph topology, ii) the swarm is able to track this reference with the smallest possible error/delay.

This chapter is organized as follows: section 4.2 gives a brief background over multi agents system. The formation control architectures are described in section 4.3. section 4.4 introduces the consensus dynamics and the formation control design using SMC controller. Trajectory generation and obstacles avoidance algorithms are given in section 4.5. section 4.6 presents the leader election algorithm, while section 4.7 discusses the simulation results with many proposed scenarios. Finally in section 4.8, conclusion as well as future recommendations are given.

4.2 System Presentation

Our proposed approach is based on these requirements:

- **R1**: ***Multi-agents***. Multi-agent systems are systems composed of multiple interacting computing elements, known as agents. In order to take advantage of the decentralized characteristic of the multi-agent system, the agent should be given some degree of autonomy. When talking about autonomy, we mean the agent can collect information by interacting with other agents or the environment they are in, then make decision according to these information.

- **R2**: ***Formation control***. In the proposed approach the swarms have a specific topology such as: rectangle, diamond, etc. A consensus-based attitude control is used for the formation control; this allows the formation topology to be maintained with a minimum of a sharing data, and makes the controller more robust to any external disturbances.

- **R3**: ***Leader election***. Centralized formations have often one leader of the swarm, however if it is crashed then the entire mission is cancelled. In order to deal with this problem the swarm lunch an election to find the right drone to be the new leader.

- **R4**: ***Obstacles avoidance***. As expected, the earth is not flat. Many obstacles like mountains, can confuse the drones trajectory and their topology. For this reason, if an obstacle is detected the formation may be splitted into two sub-formations with two leaders. As a result, the topology have to be switched according to the two groups of swarms.

According to Figure 4.1, this framework includes a swarm of drones under a distributed L-F formation (**R1**). In the investigated problem, only the leader is aware of the mission task and formation topology (**R2**). An optimal path between the departure P_i and arrival points P_f is generates by the leader. Otherwise, all the UAVs collect information from other vehicles and the environment, the decisions include special movements, period of trajectory planned etc. Then the leader transfers the desired position r_d to control command for autopilot to achieve the arrival point. Finally the desired formation topology is maintained through the formation control.

During the mission, if a failure occurs to the leader, the followers start a leader election procedure (**R3**) in order to choose the new leader, and switch their topology depending on the remaining number of UAVs. Moreover, if an obstacle is detected (**R4**), the leader order the followers to split the topology into two formations with two leaders in order to avoid the obstacle, and then coming back to the initial topology.

4.3 Formation Control

The predefined trajectory for the group of UAVs is called “formation trajectory”. It represents the common interest of the group of quadrotors. Considering the formation trajectory, we give the definition about the formation task as follows:

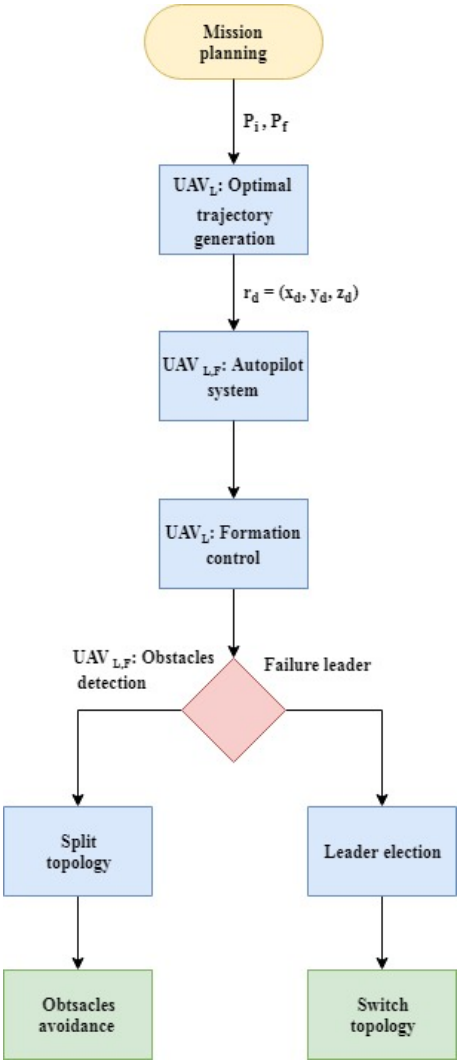

Figure 4.1: Multi-agents framework for distributed leader-followers formations

Definition 3. *Formation task: A formation task for a multi-quadrotor system with L-F architecture is represented by a desired formation trajectory (given to the leader(s)) and a desired geometric pattern (desired inter-distance and orientation between neighboring quadrotors) for the group of quadrotors.*

In the above definition, the formation task is rigid if the desired geometric pattern is fixed. Otherwise, the formation task is flexible. The formation task describes the desired integral behaviors of the multiple quadrotors. The objective of the formation control is to accomplish the formation task, i.e., the quadrotors keep the desired pattern and track a given trajectory by using the formation controllers.

The formation control strategies are based on the formation structures. According to the different formation structures, we detail the following three formation control strategies, i.e., the hierarchical centralized, decentralized and distributed formation control strategies.

4.3.1 Hierarchical centralized formation control

4.3.1.1 Centralized control without leader in the formation

This formation control strategy is considered as a quantitative extension with respect to the single quadrotor navigation problem. The formation is achieved through the planning of the desired trajectory of each quadrotor. The planed trajectories of the quadrotors should be tracked perfectly thus the collisions are avoided during the formation. In this case, a central decision maker exists to generate the desired trajectories for the agents. This formation control strategy is considered as, which is commonly based on a central trajectories generator.

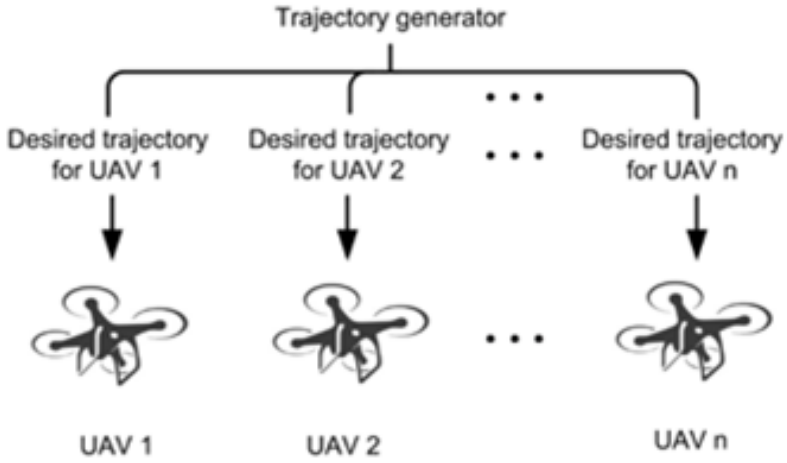

Figure 4.2: Centralized control without leader in the formation

This method is a completely centralized method, which is then simple and quite easy to implement because no interaction between UAVs is considered. However, a wide bandwidth communication channel and fast processors are required to guarantee the performance of the system. The centralized method is known as an efficient and simple implementation approach. This formation control structure can be depicted in Figure 4.2

4.3.1.2 Hierarchical centralized formation control with L-F configuration

As shown in Figure 4.3, the desired formation trajectory is given to some of the quadrotors in the group, designed as leaders of the group. The position and orientation states of the leaders are transferred to the followers through communication modules.

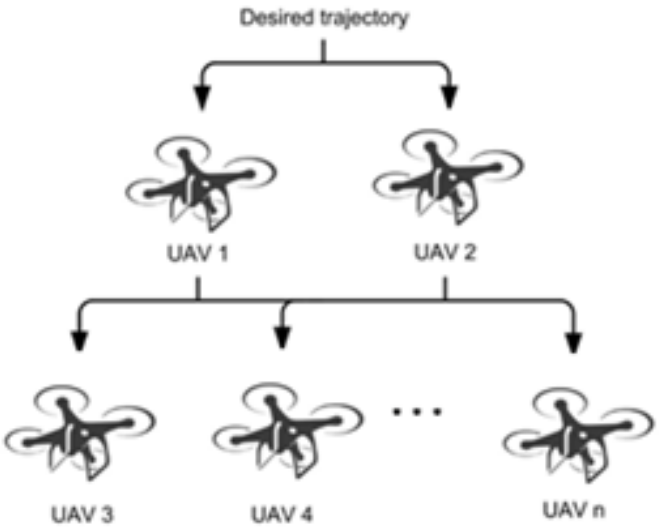

Figure 4.3: Centralized formation control with L-F configuration

This strategy reduces the burden of the central component. The calculation and data transmission are shared by some low-level components. The collision avoidance do not need to be considered explicitly, if the controllers are well-designed.

4.3.2 Decentralized formation control

As shown in Figure 4.4, for this formation structure, the leader does not exist. The desired position or trajectory is predefined for each quadrotors. The navigation of the quadrotors is achieved based on the ''navigational feedback". Every quadrotors should have the ability of sensing/detecting other UAVs around them for the collision avoidance.

The dashed lines in Figure 4.4 represent the possible detections between the UAVs. The idea is to keep some inter-distances between the UAVs, specially when the system is in the presence of disturbance.

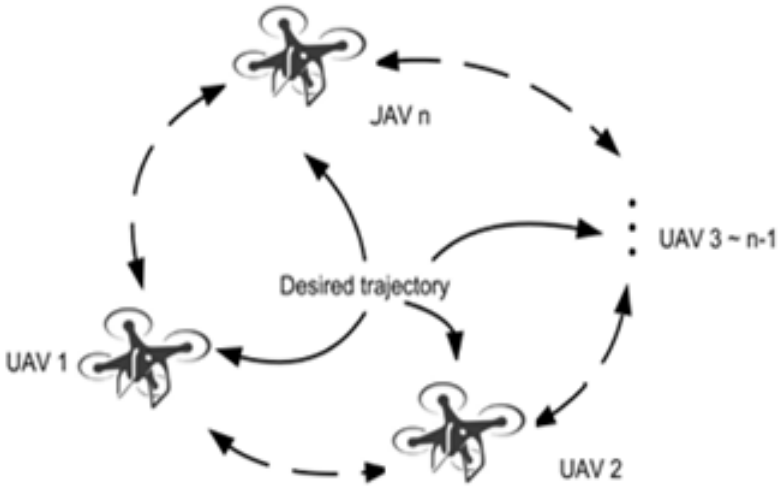

Figure 4.4: Decentralized formation control

4.3.3 Distributed formation control

The distributed formation control structure evolves from the decentralized structure. The terminology "distributed" is usually used in the domain of computer science. In multi-robot systems, the term "distributed" is used, when the communication issues are added between the robots.

Distributed control is related to the areas of decentralized control and of large scale systems. Distributed control strategies have been proposed to include communication issues into the decentralized control design framework. Such extensions concern the communication among subsystems, local controllers, and communication in the feedback loop.

4.3.4 Decentralized/Distributed formation control with L-F configuration

An example of the L-F decentralized formation structure is depicted in Figure 4.5 The red quadrotors represent the leaders (In some special formation task, the leaders are changeable), while the others are followers. The difficulty of this formation structure is that the followers have no knowledge about the formation trajectory. They only depend on the states of their neighbors (positions and velocities) in order to accomplish the formation task. Therefore, the interactions are important for the followers, not only for the reason of collision avoidance but also for the reason of formation. The proposed leader-follower formation has the following novelties comparing to the existing works.

- Decentralized formation control: no single centralized decision maker (e.g. external trajectory generator) exists. The quadrotors do not have any global knowledge.
- Multiple and changeable leaders: the number of leaders may be greater than one, the statue (leader or follower) of the agent is changeable.
- Interactions between leaders and followers: the leader(s) can be affected by their neighboring followers.

Note 1. *In the investigated leader-follower formation problem, only the leader is aware of the formation task, the remaining UAVs interact with each other or with the leaders through a rigid or switching topology.*

Note 2. *The rigid or switching topology does not correspond to the rigid or flexible formation. Although a formation is rigid, it may have rigid or switching topology.*

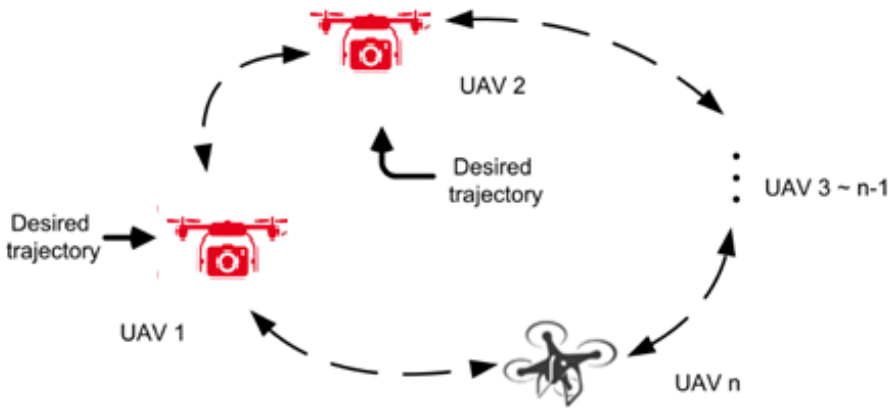

Figure 4.5: Distributed formation control with L-F configuration

4.4 Network Dynamics

A common objective for UAV swarms is to reach agreement on one or more of their states. For example, agreement on UAV position achieves rendezvous, or, if agreeing on a virtual position, formation flight can be acquired with known position offsets from the virtual position. Velocity agreement is another attractive property for formation flight. For distributed surveillance, bearing agreement is often desired.

If agreement is required over a network of UAVs without all-to-all communications, a distributed approach is necessary. A protocol which is particularly adept at distributed agreement is the consensus protocol, which is detailed below, and runs within the network dynamics of the UAV swarm.

4.4.1 Consensus Dynamics

Consider $x_i(t) \in R$ to be the i-th node's state at time t on which agreement is required for all nodes. The continuous-time consensus dynamics is defined over the graph $G = (\nu, \varepsilon)$ as:

$$\dot{x_i}(t) = \sum_{\nu_i, \nu_j \in \varepsilon} (x_j(t) - x_i(t)) \tag{4.1}$$

Thus, to update the i-th node's state, only the relative state of node i's neighbor's state is required. In a compact form with $x(t) \in R$, the collective dynamics is represented as:

$$\dot{x_i}(t) = -Lx_i(t) \tag{4.2}$$

With L being the graph Laplacian matrix of the underlying interaction topology, described in the previous subsection. For a connected graph G, the network dynamics will converge to an agreement on the state, that is $x_1(t) = x_2(t) = \cdots = x_n(t) = \alpha$, for some constant α , for all initial conditions. Further, the slowest convergence of the dynamics is determined by $\lambda_2(L)$ which is a measure of graph connectivity.

4.4.2 Leader-Follower Consensus Dynamics

Definition 4.3. The L-F consensus of system 4.3, is said to be achieved if, for each UAV $\in \nu$,

$$\begin{array}{ll} \lim_{t\to\infty} ||x_i - r(t) - d_{i0}|| = 0 & \\ \lim_{t\to\infty} ||\dot{x_i} - \dot{r}(t)|| = 0 & \text{where } i = 1, ..., n \end{array} \tag{4.3}$$

for some initial conditions $x_i(0)$, $i = 1, \ldots, n$ Therefore, the desired position of UAV $i \in \nu$ evolves according to $x_i^d(t) - r(t) = d_{i0}$ and $\dot{x_i}^d(t) - \dot{r}(t) = 0$, then, we obtain:

$$\begin{aligned} x_i^d(t) &= d_{i0} + r(t) \\ \dot{x_i}^d(t) &= \dot{r}(t) \end{aligned} \tag{4.4}$$

Let us make a sum of the relative position state vectors. Note that we drop the explicit expression of time in the expressions for the sake of simplicity.

$$\begin{array}{ll} \sum_{j\in N_i}(x_i - x_j - d_{ij}) & \text{if } i \text{ is a follower} \\ \sum_{j\in N_i}(x_i - x_j - d_{ij}) + x_i - r - d_{i0} & \text{if } i \text{ is a leader} \end{array} \tag{4.5}$$

The inter-distance is given by $d_{ij} = d_{i0} - d_{j0}$. Then, equations Equation 4.5 can be rewritten as follows:

$$\begin{array}{ll} \sum_{j\in N_i}((x_i - r - d_{i0}) - (x_j - r - d_{j0})) & \text{if } i \text{ is a follower} \\ \sum_{j\in N_i}((x_i - r - d_{i0}) - (x_j - r - d_{j0})) + x_i - r - d_{i0} & \text{if } i \text{ is a leader} \end{array} \tag{4.6}$$

We introduce the available desired trajectory for each UAV as follows:

$$\begin{array}{ll} \bar{x}_i^d = \frac{1}{|N_i|}\sum_{j\in N_i}((x_j + d_{ij}) & \text{if } i \text{ is a follower} \\ \bar{x}_i^d = \frac{1}{|N_i+1|}(\sum_{j\in N_i}((x_j + d_{ij}) + r + d_{i0}) & \text{if } i \text{ is a leader} \end{array} \tag{4.7}$$

We then observe that $\bar{x}_i^d$ is available for UAV i. We rewrite the equation Equation 4.7 in matrix form for all the quadrotors as follows:

$$\begin{bmatrix} x_i - \bar{x}_i^d \\ \vdots \\ x_n - \bar{x}_n^d \end{bmatrix} = (\widetilde{G} \otimes I_2) \begin{bmatrix} x_i - x_i^d \\ \vdots \\ x_n - x_n^d \end{bmatrix} \tag{4.8}$$

where $\widetilde{G}$ represents the normalized interaction matrix. We know that $\widetilde{G}$ is invertible if the graph of the multi-UAV system is connected with at least one leader. Therefore, if each UAV can precisely track the desired trajectory $\bar{x}_i^d(t)$, the formation task is achieved. Its time derivative $(\dot{\bar{x}}_i^d(t))$ can be obtained, which are in terms of the attitudes of the neighbors. Note that d_{ij} is constant in a rigid formation task. In the literature, for instance, where a leaderless multi-agent system is considered, the proposed consensus algorithm leads to a normalized Laplacian matrix. In this thesis, since an L-F configuration is considered, a normalized interaction matrix is defined by:

$$\widetilde{G} = (G^D + G^L)^{-1}.G \tag{4.9}$$

4.4.3 Controller Design

In this section a control formation scheme is proposed in order to maintain the centralized Leader-followers formation.

The objective of the formation controller is to achieve the desired leader-follower formation configuration in X-Y plane, after following the leader in the Z direction to either same or different height. This formation shape is maintained via keeping a constant distance d and angle α between each follower and the leader.

$$\begin{aligned} d_x &= -(X_L - X_F)\cos\psi_L - (Y_L - Y_F)\sin\psi_L \\ d_y &= (X_L - X_F)\sin\psi_L - (Y_L - Y_F)\cos\psi_L \end{aligned} \tag{4.10}$$

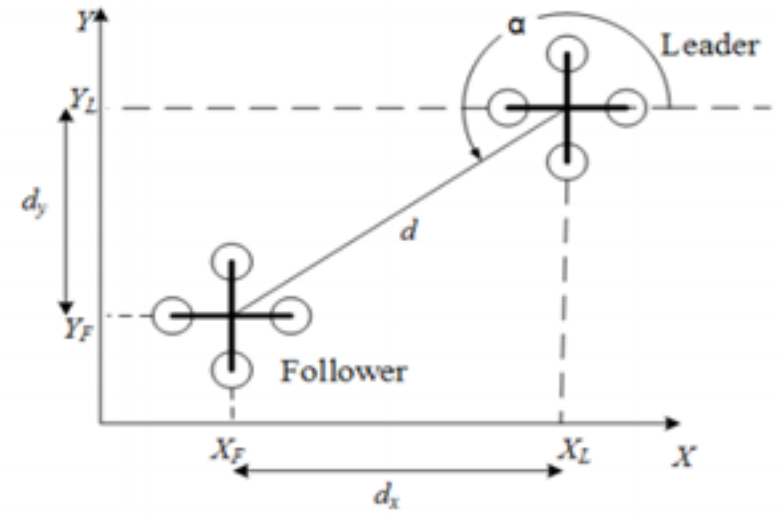

Figure 4.6: Leader-followers formation architecture

with d_x and d_y are the X and Y coordinates of the actual distance d as shown in Figure 4.6 The proposed control algorithm is shown in Figure 4.7 applying SMC in its design to keep the formation even in perturbed and uncertain environment. Formation control errors of x; y and z should satisfy the following conditions:

$$\begin{aligned} \lim_{t\to\infty} \parallel e_x \parallel = \parallel d_x^d - d_x \parallel = 0 \\ \lim_{t\to\infty} \parallel e_y \parallel = \parallel d_y^d - d_y \parallel = 0 \end{aligned} \tag{4.11}$$

Where d_x^d and d_y^d are the desired distance between the leader and follower in the X and Y coordinates respectively.

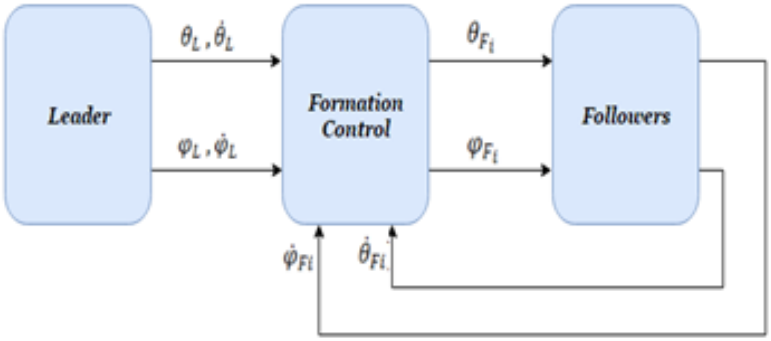

Figure 4.7: Leader-followers formation controller

To achieve this goal, first-order sliding mode controller is used to minimize this error. First, a time varying surface $S(t)$ is defined by the scalar equation $s(e;t) = 0$, where:

$$s(e,t) = (\frac{\mathrm{d}}{\mathrm{d}y} + \lambda)^{n-1} e \tag{4.12}$$

The second-order tracking problem can be transferred to a first-order stabilization problem, thus:

$$\begin{aligned} \dot{s} &= \ddot{e} + \lambda \dot{e} \\ \frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}y} s^2 &\leq -\eta \mid s \mid \end{aligned} \tag{4.13}$$

Equation 4.13 is a Lyapunov candidate function chosen for the control law u to maintain scalar $s = 0$. This function states that s^2 is the squared distance to the sliding surface, where η is a positive constant.

The formation can be then controlled for each follower using the following equations:

$$\begin{aligned} \ddot{X}_{Fi} &= \ddot{X}_L + \lambda_x(\dot{X}_L - \dot{X}_{Fi}) \\ \ddot{Y}_{Fi} &= \ddot{Y}_L + \lambda_y(\dot{Y}_L - \dot{Y}_{Fi}) \end{aligned} \tag{4.14}$$

Finally, the position control problem is transformed to an attitude control, which means that a direct estimation of the attitude can be used to control the swarm:

$$\begin{aligned} \theta_{Fi} &= \theta_L + \lambda_\theta(\dot{\theta}_L - \dot{\theta}_{Fi}) \\ \phi_{Fi} &= \phi_L + \lambda_\phi(\dot{\phi}_L - \dot{\phi}_{Fi}) \end{aligned} \tag{4.15}$$

where λ_θ and λ_ϕ are the formation control gains, with $\lambda_\theta > 0$ and $\lambda_\phi > 0$.

4.5 Trajectory Optimization and Obstacles Avoidance

As in ([14]), this section the quadrotors swarm is facing much type of obstacles. The leader mission is to reach the desired position and avoid the collision with obstacles from a part and the collision between the agents from the other part. For all the simulated cases we consider only the 2D obstacles in the X-Y horizontal plan. The altitude is maintained constant during the entire mission. The algorithm used for the swarm obstacle avoidance is described in algorithm 4.1

Algorithm 4.1: Obstacles Avoidance

```
1  begin
2      Path(P_i(i); P_f(i))alli ∈ V
3      P_i ← P
4      while Dist(P_f; P_i) > d_admissible do
5          P ← DrawLine(x_g; x_i)
6          if Pexist then
7              x_i ← SelectOptimalPath(P_i, P)
8              P ← AddPath(P, x_i)
9              if Obtacle_detected then
10                 x_i ← SelectOptimalPath(P_i, P)
11                 ifSelectOptimalPath(P_i, P)
12                 switch decentralized/distributed formation
13                 x_i ← SelectOptimalPath(P_i, P)
14                 P ← AddPath(P, x_i)
15                 switch line formation
16                 x_i ← SelectOptimalPath(P_i, P)
17                 P ← AddPath(P, x_i)
18             end
19         end
20         i ← i + 1
21     end
22 end
```

algorithm 4.1 starts with initial positions of all the swarm UAVs, and the only the final position of the leader, the agent's final position is then estimated depending on the formation topology. The mission objective is that the leader reaches its final destination, which means that the distance between the starting and final position converge to zero. The inter-distance between the swarm agents is also supposed to be respected what-ever the formation topology is.

The optimal path between the starting the final position is a straight line, if exists then the leader and the followers will track it. If any obstacles detected then the function SelectOptimalPath will generates the nodes to avoid the obstacles for the leader, which then generates the follower's path.

A decentralized/distributed formation is generated by the leader for the followers if any optimal path can be tracked by the followers. The leader then designed a new leader for the other teams, which then track its path and respect the separation distance with its teammates.

The switching topology to line formation is called whenever no optimal path can be generated for the followers. The swarm then switches its topology to line, avoids the obstacles, and comes back to its initial topology if no furthered obstacles are detected.

4.6 Leader Election

This section simulates the scenario whenever a sudden failure occurs to one of the swarm agents. During such cases, the failed agent makes an emergency landing to prevent crashes, and the rest of the swarm agents have to switch their topology. Furthermore a leader election procedure is required whenever the failed agent is a leader. The algorithm is executed as indicated in algorithm 4.2.

First of all, a set of N drones are identified from 1 to N. The desired topology (T) depends on the drones number. Each topology has its related paths ($P(T)$). The head of the drones list (A_1) is considered as a leader L. The leader is supposed to have the knowledge about: the drones IDs, the desired topology (T_{ex}) and path (P_{ex}) and all the possible paths and topologies (P and T).

Before the mission, the leader sends a ready message to all the drones to ask them if they are ready for the mission or not (line 11). Once all the drones are ready (line 16), the leader tracks the reference path and maintains the formation topology (line 19).

During the mission, a test loop is required to test the state of each drone. If a failure is detected in one of the drones (line 22) an emergency land occurs to prevent the crash. Two possible cases can take place depending on the failed agent. If the failed agent is a leader, the election procedure starts (line 27), and the agent with better performances (Y) is elected as a leader (line 28-32). Whereas, if the failed agent is a follower, it will be removed from the drones list (line 34). The desired topology will be switched in both cases depending on the number of remaining drones (return to line 5).

4.7 Simulation Results

In this section the simulation results related to the quadrotors formation control discussed in the other sections are shown.

Many scenarios have carried out depending on the formation control and the different constraints that can occur during a mission. For all the next scenarios 4 quadrotors UAVs are used (Figure 4.8), the communication link between all the UAVs is supposed to be assured. The aim is that all the UAVs can keep switch their formation.

An overview of the simulated cases can be found bellow:

Algorithm 4.2: Leader Election

Input: N: Number of agents, A: list of N agents where each one has an ID from 1 to N, $T = \{T(N), T(N-1), ..., T(1)\}$: list of N topology according to the agent's number, $P(T)$: list of N paths according to each topology.

```
1  begin
2      idL = 1;
3      L = A_idL;
4      while true do
5          T_ex = T(N);
6          P_ex = P(T_ex);
7          L = [T_ex, T, P_ex, P, A, idL];
8          ALLReady = 0;
9          for i ← 1 to N 1 do
10             if i ≠ idL then
11                 send(i, ready = false) by L;
12                 wait(i, ready(i)) = true by L;
13                 ALLReady = ALLReady + 1;
14             end
15         end
16         if AllReady = N − 1 then
17             for i ← 1 to N 1 do
18                 if i ≠ idL then
19                     send(P_ex(i), i) by L;
20                 end
21             end
22             Failure = FailureDetect(A);
23             if Failure ≠ −1 then
24                 if Failure = idL then
25                     IdY = 0;
26                     for j ← 1 to N 1 do
27                         if i ≠ idL then
28                             State_i = election;
29                             idY = max(Y(A_i), IdY);
30                         end
31                     end
32                     L = A_idY;
33                     idL = idY;
34                 end
35                 A.remouveElement(A_Failure);
36                 N = N − 1;
37             end
38         end
39     end
40 end
```

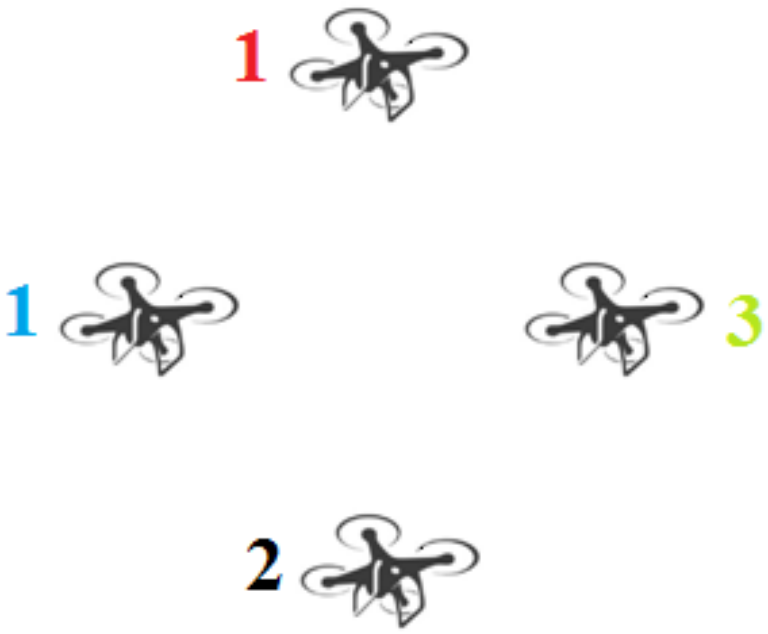

Figure 4.8: Leader-followers formation topology

1. **Scenario 1:** All the 3 UAVs start from different points and track a desired path while keeping a diamond formation with one leader and three followers.

2. **Scenario 2:** In this case the UAVs are tracking the same path of Scenario 1 but in the presence of an external wind disturbance on the leader and followers.

3. **Scenario 3:** For this scenario a decentralized control formation is studied. The four UAVs are forming a rectangular topology at a predefined altitude after staring from different point; an extra edge controller is used to reach the designed formation (see Appendix B).

4. **Scenario 4:** During a mission a sudden engine failure occurs to the leader, and the followers are sicked to elect a new leader of the swarm.

5. **Scenario 5:** The last scenario simulates different cases of the presence of external obstacles. The swarm continues its path to the desired position, and avoids the collision with the obstacles or between agents.

4.7.1 Scenario 1: Centralized L-F Formation

As mentioned before, in this scenario 4 quadrotors UAVs (1 leader and 3 followers) are used, in a diamond formation. The leader begins its route from an initial position $P_i(x_0, y_0, z_0) = [0; 0; 0]^T$ by reaching the required height z first, then track the required path in X-Y plane. The leader's mission is to travel to the point $P_f(x_0, y_0, z_0) = [0; 10; 10]^T$. The initial positions of the three followers are at

$F_1(0) = [-3; 0; 0]^T$, $F_2(0) = [-1; -1; 0]^T$ and $F_3(0) = [3; -4; 0]^T$ respectively, and their desired formation distances with respect to the leader are $d^d_{F_1} = [-2; -2; 0]^T$, $d^d_{F_2} = [0; -4; 0]^T$ and $d^d_{F_1} = [2; -2; 0]^T$. This first scenario was chosen in order to test the controller and its ability to hold the swarm formation while tracking a desired path.

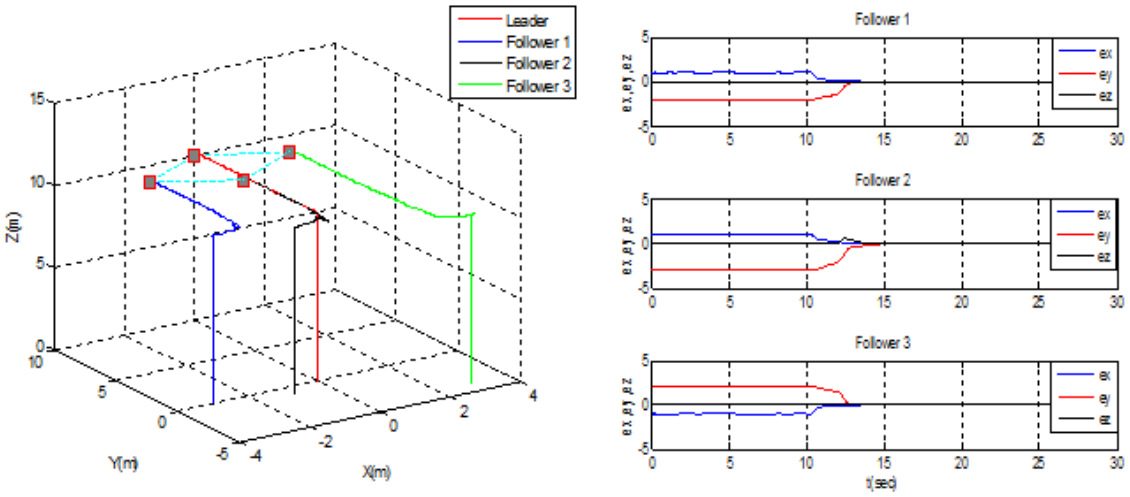

Figure 4.9: Centrelized L-F Formation

Figure 4.9 shows that the required formation is achieved with high accuracy, and that errors in all coordinates x, y and z were converged to zero in only 13 sec.

4.7.2 Scenario 2: Centralized L-F Formation with Disturbance

In this scenario the UAVs swarm is tracking the same path as *scenario*1 but with the presence of an external wind gust disturbance from the 20 to 25 sec over the X,Y and Z axes. The wind gust velocity over the three coordinates is illustrated in Figure 4.10 The wind speed is between -1 and 1m/s.

This kind of scenarios allows testing the robustness and effectiveness of the controller.

From Figure 4.10 and Figure 4.11 it is clear that all the quadrotors were able to maintain their stability, as well as the desired formation during the wind gust disturbance. The formation errors converged to zero after just 1 sec from the end of the disturbance. The swarm agents continue then their desired path while maintaining the same altitude.

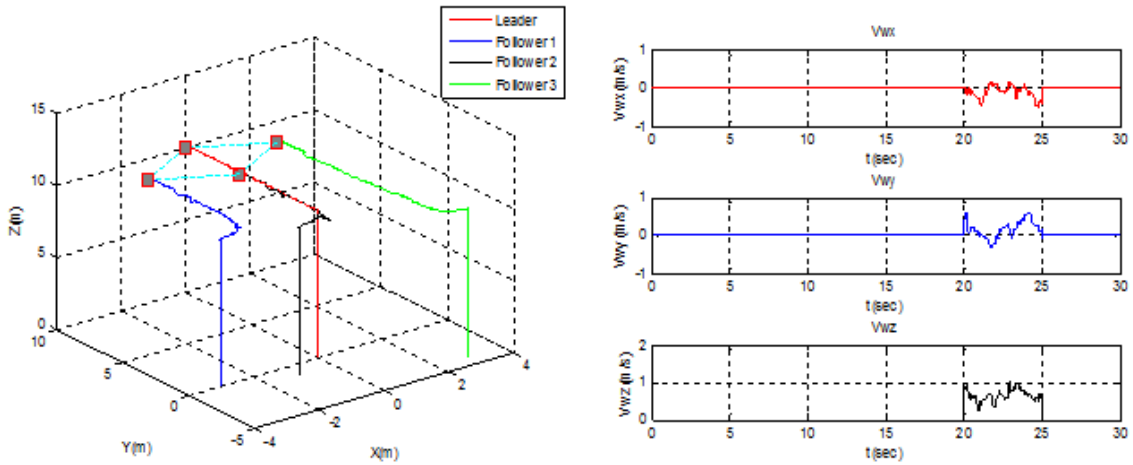

Figure 4.10: Centralized L-F Formation with disturbance

4.7.3 Scenario 3: Decentralized Formation with Extra Edge

This scenario aims to test the decentralized control formation, where no leader is designed, and all the four quadrotors have knowledge about the desired path to follow. The quadrotors UAVs from 1 to 4 start from the initial positions $F_1(0) = [10; -10; 0]^T$, $F_2(0) = [-10; -10; 0]^T$, $F_3(0) = [-10; 10; 0]^T$ and $F_4(0) = [10; -10; 0]^T$ respectively, and meet each other at an altitude $z = 10m$, while keeping a rectangular separation of 10 m . An extra edge controller is added to ansure the convergence over the meeting points.

The obtained results shown in Figure 4.12 proofs the effectiveness of the extra-edge designed controller, where all the quadrotors were able to reach the desired meeting points and keep the separation distances. The formation errors between the UAV 1-2 and UAV 3-4 converge to zero in only 6 sec. The altitude error is kept to zero during the entire mission because of the synchronization of the altitude control for all the swarm agents.

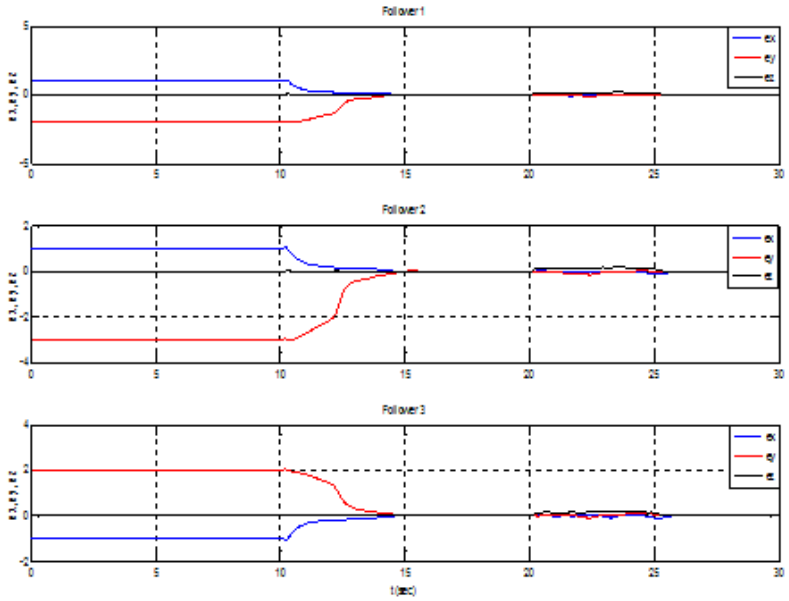

Figure 4.11: Followers Trajectory Tracking Errors

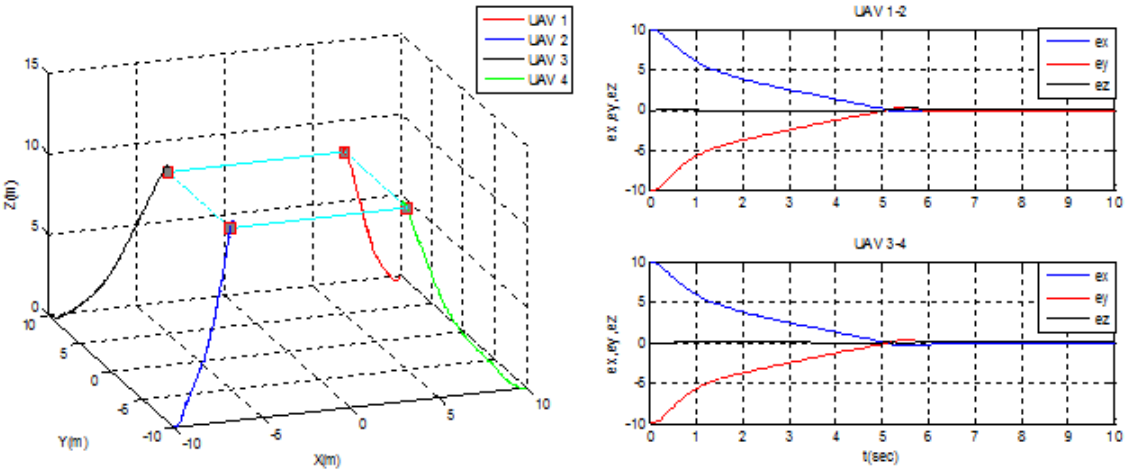

Figure 4.12: Decentrelized Formation with Extra Edge

4.7.4 Scenario 4: Leader Election

4.7.4.1 Case 1

Let's suppose that $Follower1$ is elected as new leader.

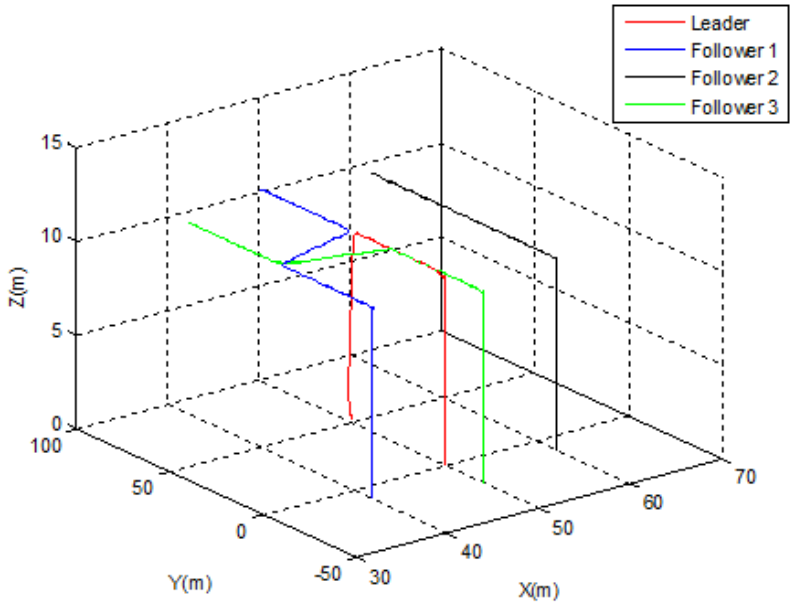

Figure 4.13: Leader election case-1-

As shown in Figure 4.13 the swarm was able to switch its topology formation from diamond to triangular just after the leader emergency landing. The position's switching between the new leader and $Follower2$ was made only in 5 sec, while the separation distance (10 m) between the new leader and $Follower3$ was highly respected during the switching operation Figure 4.14. The new triangular topology was completely achieved after t = 30 sec.

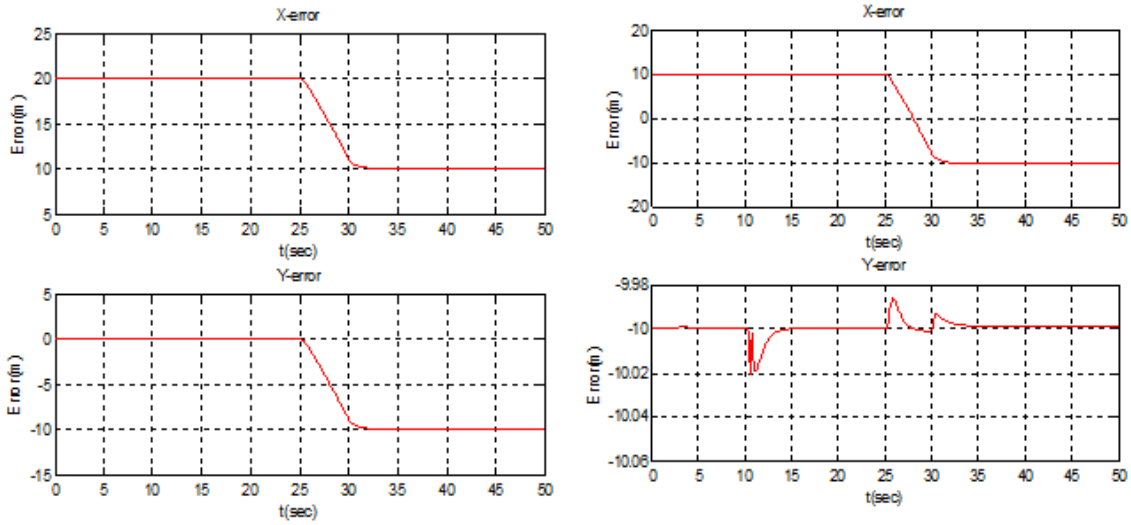

Figure 4.14: Leader election case-1- Control inputs

4.7.4.2 Case 2

In this case, *Follower*2 is supposed to be elected as new leader. Figure 4.15 and Figure 4.16 illustrate the simulation case. The obtained results are similar to those obtained in the first case. The triangular topology was achieved in t = 30 sec, and the position's switching between the new leader and *Follower*3 was made in 5 sec. The separation distance between the new leader and *Follower*1 was also maintained during the switching operation (Figure 4.16).

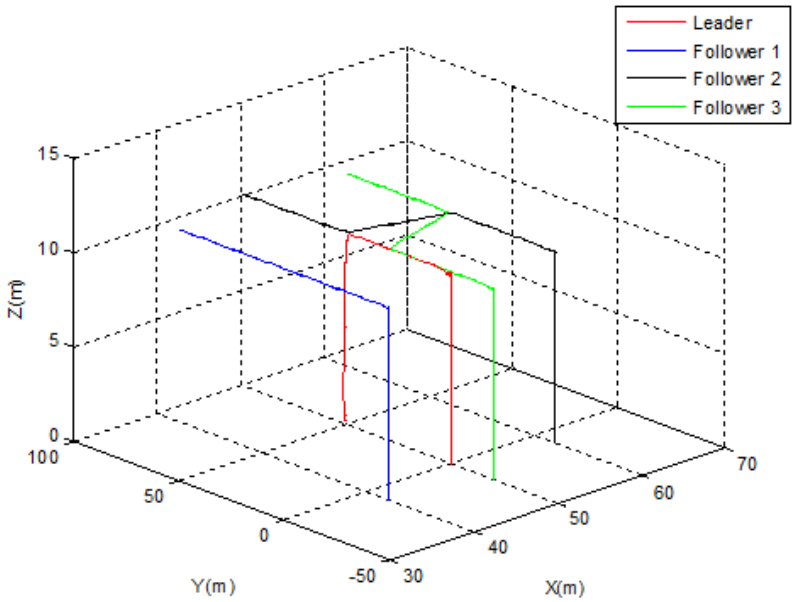

Figure 4.15: Leader election case-2-

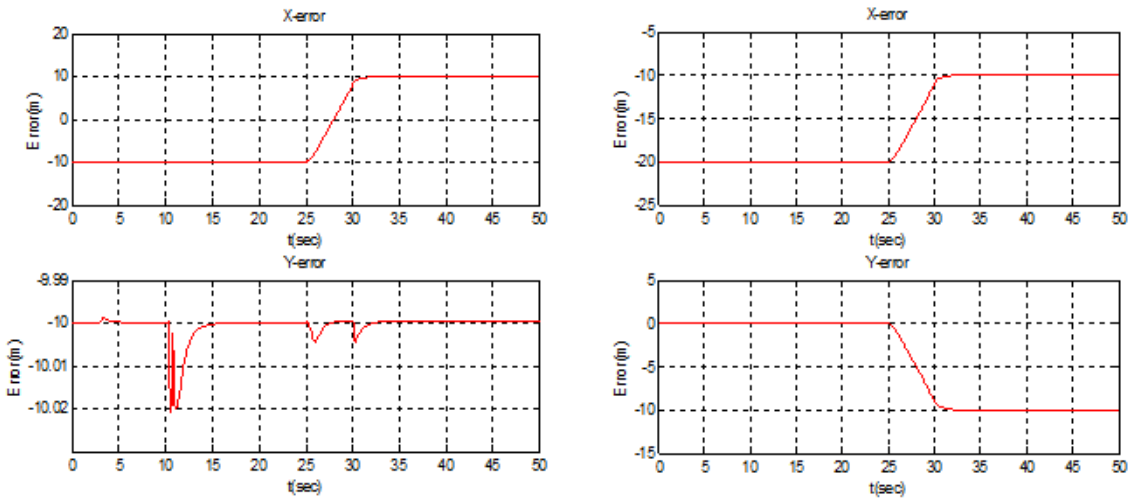

Figure 4.16: Leader election case-2- Control inputs

4.7.4.3 Case 3

For this last case, *Follower*3 is supposed to be elected as new leader. As shown in Figure 4.17 *Follower*3 is positioned behind the leader, thus no need to switch positions. Figure 4.18 demonstrates that the separation distance was maintained with *Follower*3 and *Follower*1 with high accuracy.

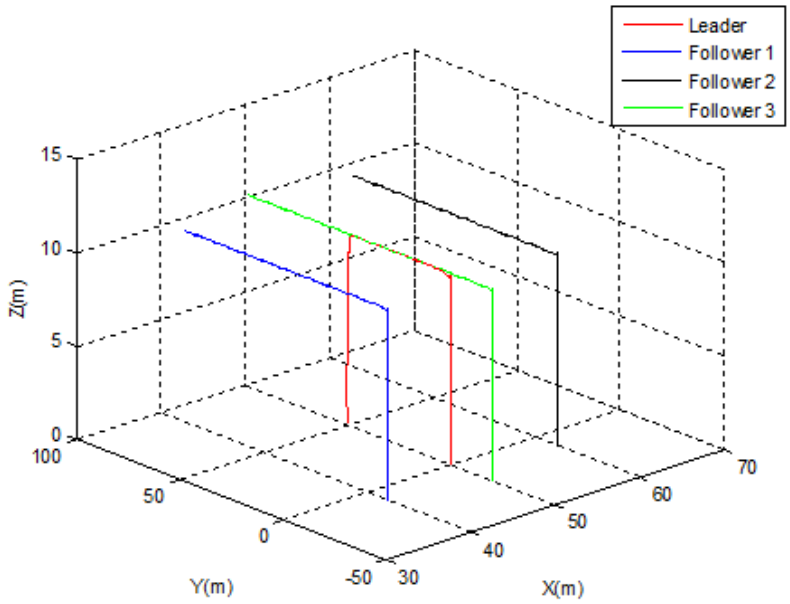

Figure 4.17: Leader election case-3-

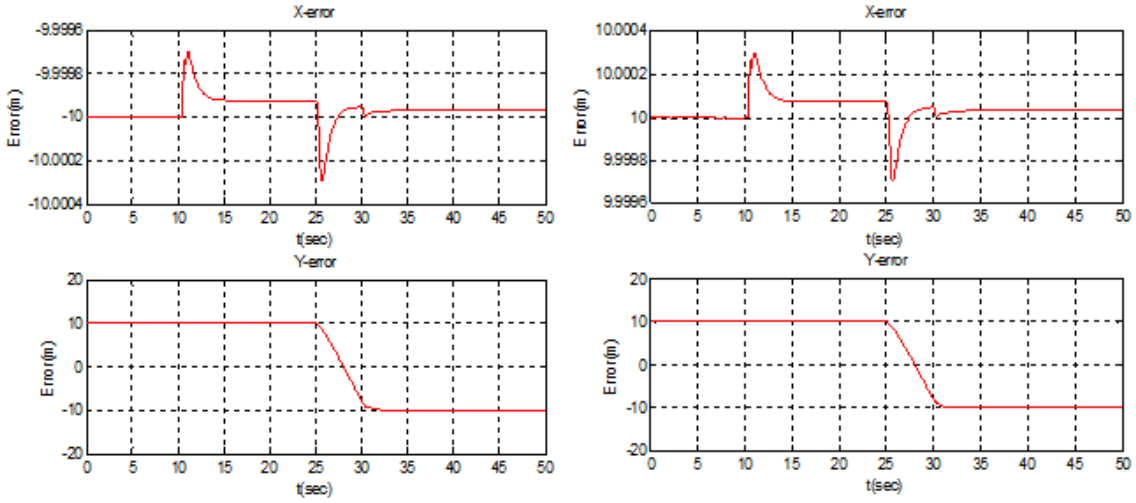

Figure 4.18: Leader election case-3- Control inputs

4.7.5 Scenario 5: Obstacles Avoidance

4.7.5.1 Case 1

In this case the UAVs swarm start from the following positions: $F_1(0) = [50; 65]^T$, $F_2(0) = [60; 75]^T$, $F_3(0) = [50; 85]^T$ and $F_4(0) = [40; 75]^T$. The mission of the leader is to achieve the desired point $P_f(x_0, y_0) = [50; 15]^T$ and avoid the circular obstacle (R = 10 m) located at $O_1(x_0, y_0) = [50; 50]^T$. The separation is 10 m between the agents. The swarm is supposed to hold the diamond formation.

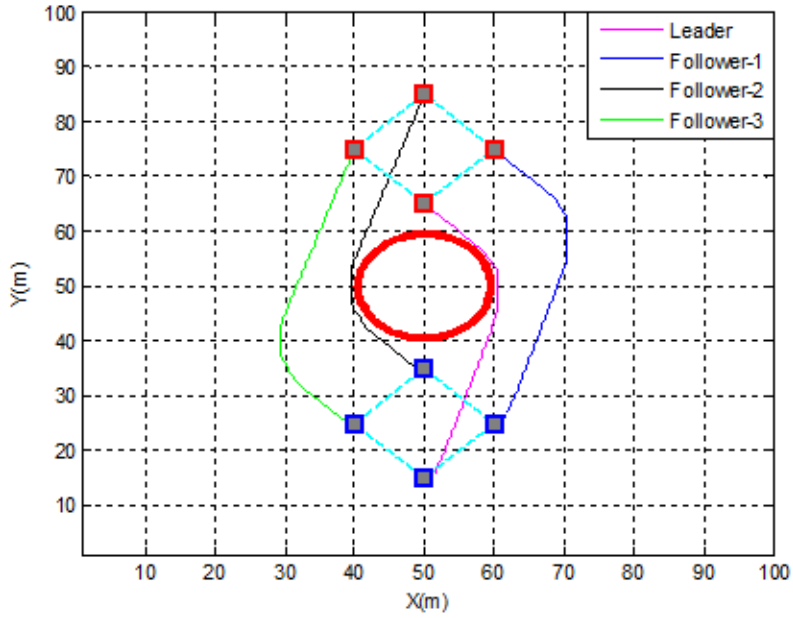

Figure 4.19: Obstacle avoidance case-1-

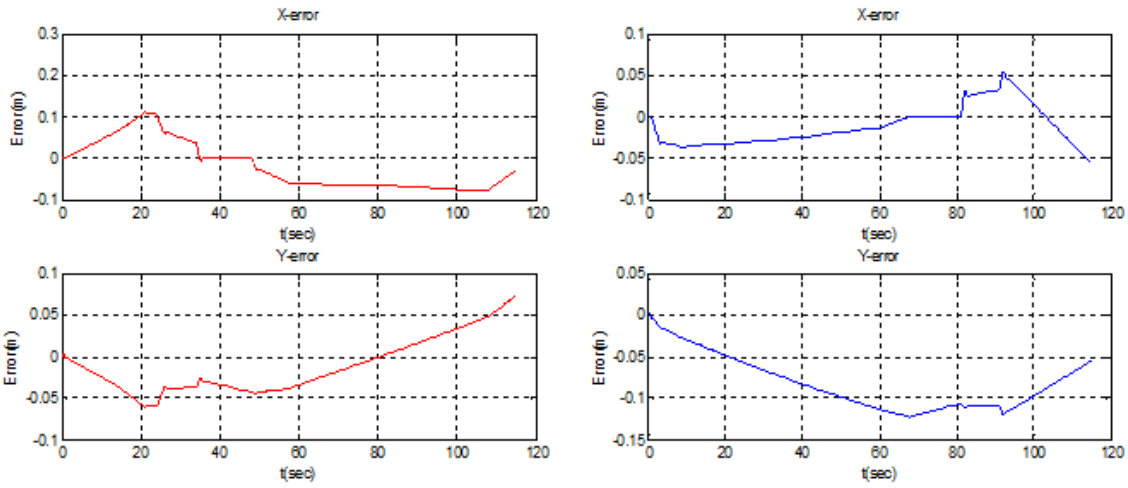

Figure 4.20: Obstacle avoidance case-1- tracking error

From Figure 4.19 it can be noticed that the swarm was able to avoid the circular obstacle using the decentralized/distributed formation in order to optimize the energy consumption.

The swarm was divided into two teams with two Leaders, one formed by the initial leader and follower 1 and the second formed by follower 2 (new leader) and follower 3.

Fig Figure 4.20 presents the formation error of team 1 and team 2 respectively, it can be noticed that the separation distance (10 m) between the two UAVs was respected with high accuracy in both x and y directions. This reflects the high performance of the formation controller.

4.7.5.2 Case 2

In this case the UAVs swarm start from the following positions: $F_1(0) = [50; 75]^T$, $F_2(0) = [60; 85]^T$, $F_3(0) = [50; 95]^T$ and $F_4(0) = [40; 85]^T$. The mission of the leader is to achieve the desired point $P_f(x_0, y_0) = [50; 30]^T$ and avoid two circular obstacles (R = 10 m) located at $O_1(x_1, y_1) = [60; 63]^T$ and $O_2(x_2, y_2) = [60; 37]^T$. The separation is 10 m between the agents. The swarm is supposed to hold the diamond formation.

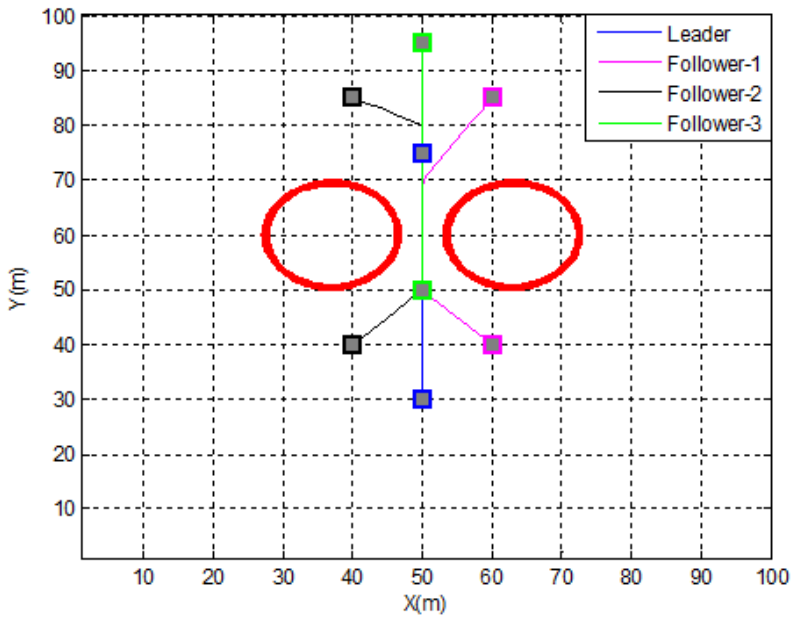

Figure 4.21: Obstacle avoidance case-2-

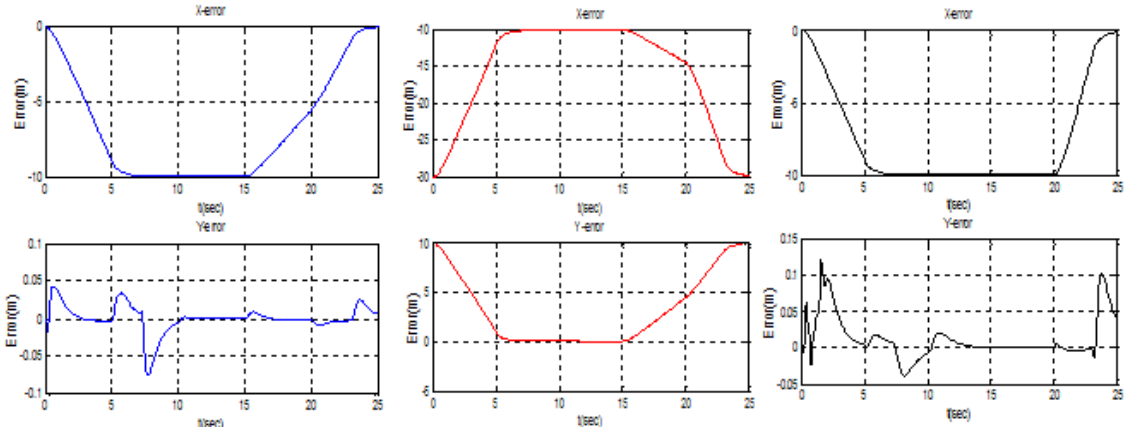

Figure 4.22: Obstacle avoidance case-2- tracking error

In such case that the swarm cannot passes between the obstacles since the distance between the obstacles is only 6 m. The optimal solution for this problem is to switch the formation topology from diamond to linear, and then comes back to the initial topology if no further obstacles are detected. Figure 4.21 shows the case study scenario.

From Figure 4.22 that the inter-distance between the Leader- Follower 1, Follower 1- Follower 2 and Follower 2- Follower 3 was respected with high accuracy. The position switching was made in just 5 seconds from the diamond to the line formation and in about 10 second to come back to the diamond formation.

4.7.5.3 Case 3

This case is an extension of the second case where a new line obstacle is added, and the swarm is sicked to maintain its linear formation and avoid the new obstacle which is situated at $O_3(x_3, y_3) = [40 : 60; 35]^T$. The obtained results are shown in Figure 4.23

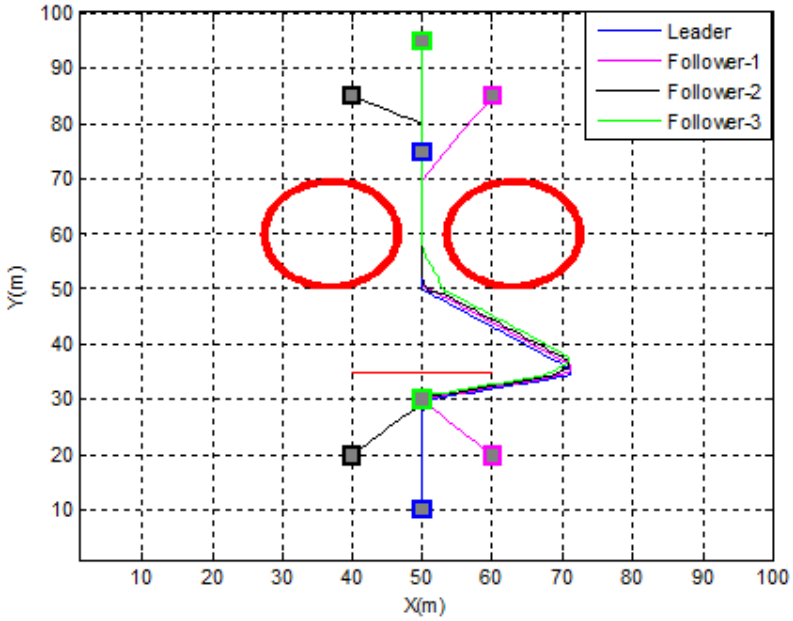

Figure 4.23: Obstacle avoidance case-3-

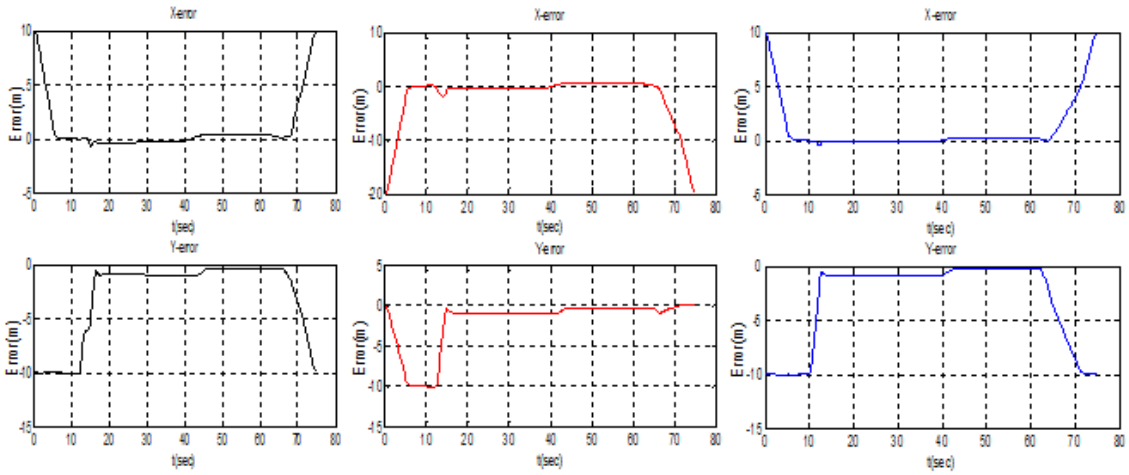

Figure 4.24: Obstacles avoidance case-3- tracking error

As illustrated in Figure 4.23 the swarm avoid the two circular obstacle the as case 2, but this time the leader detect the presence of the new line obstacle so the swarm maintain its linear formation until the point $(x, y) = [50; 30]^T$ where the swarm start it switching back to the diamond formation.

From Figure 4.24 it is clear that the swarm was able switch to the linear formation while maintaining the inter-distance between the UAVs. The new line obstacles is detected and avoided, and the switching to the diamond formation is executed with high accuracy.

4.8 Conclusion

This chapter studied the problem of multi-agents UAVs formation control. A new framework for distributed leader-followers quadrotors formation that overcomes the drawbacks of the standard L-F formation and provides better performances was proposed.

For the formation control, a consensus-based attitude control was used to hole the formation with a minimum of a sharing data, and makes the controller more robust to any external disturbances. Moreover the agents were able to elect the new leadership in order to adapt to a leader sudden failure or a varying in the graph topology due to external obstacles. The combination of a double loop control structure based on backstepping/SMC controllers was applied to track the reference trajectory, maintain the formation strategy and avoid collisions.

Many scenarios were proposed in order to test the performances and the robustness of the designed framework; all the obtained results were judged to be satisfactory. The quadrotors UAVs have successfully tracked the reference with the smallest possible error/delay, and eliminate the effects of the external wind disturbances.

Chapter

5 SAR Operations Using Single UAV

Contents

5.1 Introduction

SAR operations are one of the applications that UAVs could support, especially in the area of catastrophe assistance. In a typical scenario, after a disaster such as an earthquake, volcanic activity or an aircraft crash, the authorities need to have a quick evaluation of catastrophe situation and a precise estimation of damages. UAVs will be used to scan the disaster area and collect information about any possible survivals, and send back their positions to the base station.

SAR operations are considered as critical missions, a considerable organizational capability to aid as fast as possible the disaster victims is required. An overview for the use of UAVs for SAR applications could be found in [36, 46]. The growing recognition of the potential of using UAVs is supported by an increasing number of utilities spanning in many areas, including human detection and localization [52,70], mission planning [6, 66], networking and data processing [50], collaborative SAR missions [1, 57], assisted UAV task allocation and trajectory planning [20, 21], etc. Many types of the SAR missions are defined depending on the environment and the situation of the targets. Survivals could be on mountain in an avalanche events [45], in marine [4, 23, 54], facing a fire [49], or even in a closed building [53]. Reference [42] describes the design a robotic Autonomous Unmanned Aerial System (AUAS), namely the ROLFER (Robotic Lifeguard for Emergency Rescue) system, which aims to immediately provide rescue and lifesaving services. However the previous cited works describe the different parts of a SAR system, but no one has presented a full designed autonomous system that takes into consideration all the contributed parts (i.e. mission planning, autopilot system, search strategy and targets geo-localization).

Various algorithms and techniques have been proposed for searching targets, Reference [67] studies different search strategies based on greedy heuristics, potential-based algorithms and partially observable Markov decision, while [26] proposes a bio-inspired self-organized search strategy. In reference [28] search strategies provided by the international maritime organization and international civil aviation organization, and published annually in IAMSAR (International Aeronautical and Maritime Search and Rescue) manual are used. Despite the development of many heuristics and probabilistic problem-solving techniques for SAR search problem, published procedures still deliver approximate solution and mostly fail to cover the integrity of searching area and include the probability of detection/observation model, questioning their real expected relative efficiency.

This chapter is organized as follow: section 5.2 introduces a presentation of the designed SAR system. section 5.3 presents the mission planning with the different SAR strategies and their path patterns while section 5.4 discusses the sensor coverage area and target detection algorithm. Simulations and SAR strategies comparison are in section 5.5. Finally section 5.6 a conclusion is given.

5.2 System Design

This chapter aims to design a SAR system using low cost quadrotors to execute such missions, since SAR operations are often facing many challenges such as the quality of the sensor data, the energy limitation and environmental hazards .. Quadrotors UAVs are chosen because they are capable to hover vertically, forward and back which is so benefits to reduce the need of a high quality embedded camera, and deal easily with any type of environmental obstacles.

The designed SAR system is subdivided to three main sub-systems as illustrated in Figure 5.1

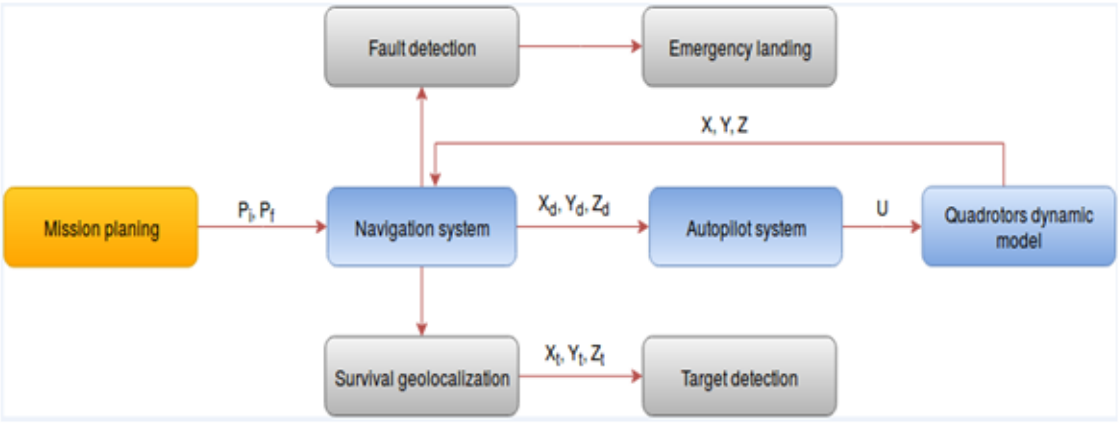

Figure 5.1: SAR System

1. **Mission planning** The aim of this section is to define the SAR strategy to be followed, then generates the coordinate of the initial and final point (P_i and P_f) of every leg of the strategy path.

2. **Navigation system** This sub-system is the heart of the hall system because it have many functions, the first function is to provide the autopilot sub-system with the optimal path to track, using an optimal trajectory generation algorithms that generates the optimal coordinates (X_d, Y_d, Z_d) for each time step between the path leg. The sub-system also uses a survivals geo-localization algorithm to calculate the target coordinates (X_t, Y_t, Z_t) based on the probability of detection and the sensor range of the embedded camera. Finally the system monitors any failures that can occur such as a divergence from the desired path, an engine failure, or the battery energy consumption, by an emergency landing operation that saves the quadrotors from a sudden crash, and informs the base station by its position which means a mission failure.

3. **Autopilot System** This sub-system have to track the path generated by the navigation sub-system, this is done due to a double loop control strategy of the position and attitude tracking based on an optimal backstepping controller that assures the energy optimization, the path tracking accuracy and the quadrotors stability.

5.3 Mission Planning

As shown in Figure 5.2 during a typical scenario, quadrotors UAVs will be used to deploy an area of interest perform sensory operations to collect evidence of the presence of a victim, and report their collected information to a remote ground station or rescue team. For surface and aircraft facilities to search effectively, search patterns and procedures must be pre-planned so ships and UAVs can cooperate in coordinated operations with the minimum risks and delay. Standard search patterns have been established to meet varying circumstances.

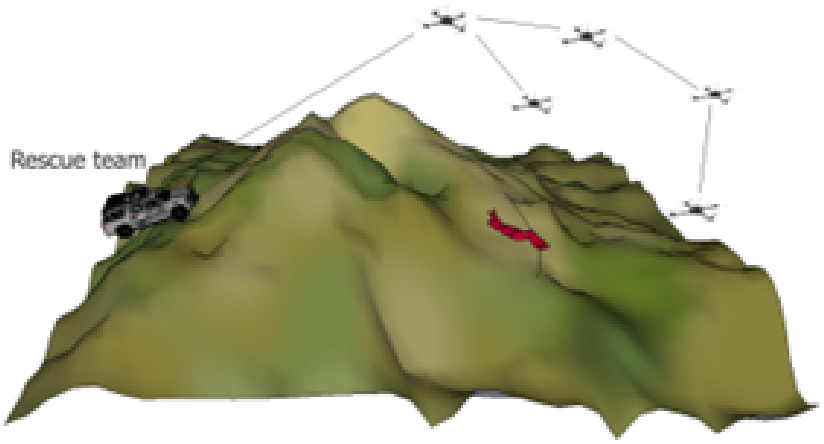

Figure 5.2: SAR Operation Using Quadrotors UAVs

Several search strategies have been recognized and set as standards by the International Maritime Organization and International Civil Aviation Organization. These are published annually in IAMSAR (International Aeronautical and Maritime Search and Rescue) manual. These are designed to provide simple and effective visual search patterns that can be applied in various situations.

Before defining the search pattern it is important to determine the search area by creating boundaries, because we have often a huge open space with trees, fields, rivers, streams. From this perspective, the subject could be anywhere. Sending UAVs out randomly would be a waste of resources and time. So it will be necessary to establish a datum, or geographic reference, for the area to be searched.

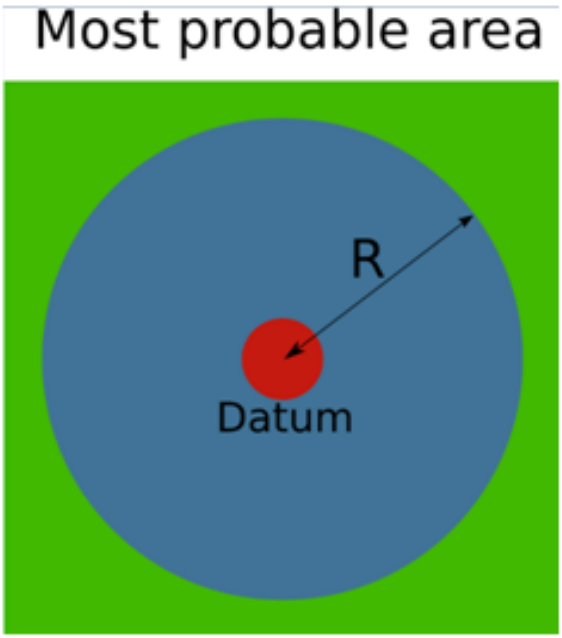

Figure 5.3: Searching Surface and Probabilistic Areas

As shown in Figure 5.3, lets A be the searching surface with the most probable area. Depending on the reported position and time of the SAR incident, and the estimated surface movements of the distressed craft or survival craft, we need to define a circle with a radius R_a (depending on drift) , where the center of the circle is the Last Knowing Position (LKP) and R_a is giving in Equation 5.1 as follow:

$$R_a = \frac{\sqrt{A_g}}{2} \tag{5.1}$$

The standard search patterns for a desired area are as follow:

- Parallel Track and Creeping line Search
- Expanding Square Search
- Sector Search
- Contour Search

The following section discusses the main characteristics of the previous standard searching patterns and the conditions under which they are being utilized.

5.3.1 Parallel Track and Creeping line Search

Both parallel track and creeping line search are based on covering (sweeping) the search area by maintaining parallel tracks as shown in Figure 5.4 The only difference between these two is in the orientation of the search legs, i.e. if the search legs are parallel to the long sides (major axis) or short sides (minor axis) of the designated search area.

Parallel track or creeping line search strategy are generally used when one or a combination of the following conditions exists:

- The search area is large.
- Uniform coverage is required, i.e. the search area has to be divided into sub-areas for assigning individual search units to search for the target (survival person) simultaneously.
- Search is conducted over water or flat terrain.
- The location of the target is uncertain, i.e. equal probability of the target being anywhere in the search area.

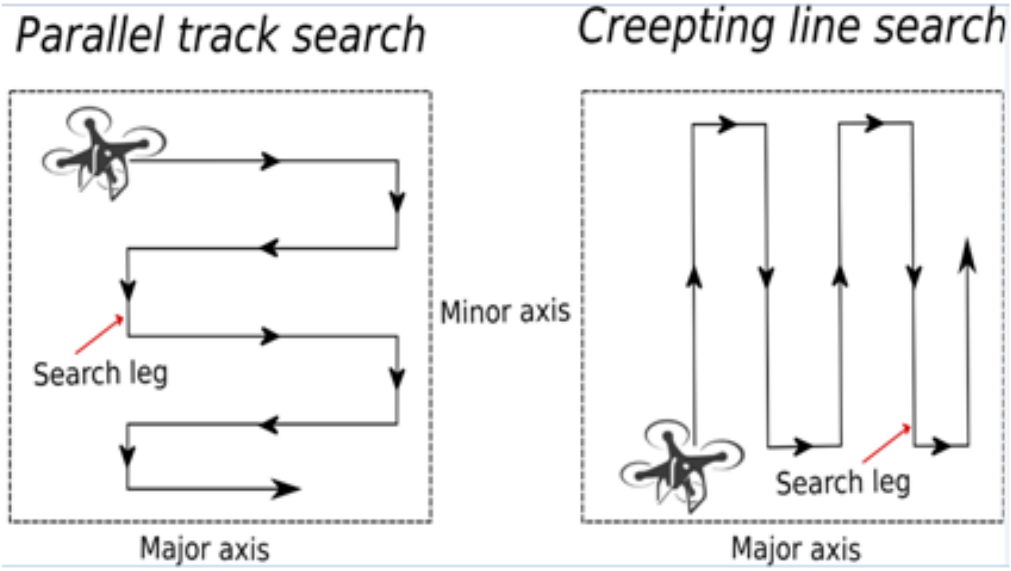

Figure 5.4: Parallel track search and Creeping line search pattern

5.3.2 Expanding Square Search

For this search strategy the pattern begins at the center of the designated search area and expands outward in concentric squares as shown in Figure 5.5 Expanding Square search strategy is generally used when one or a combination of the following conditions exists:

- The search area or search section is relatively small.
- Uniform coverage is required, i.e. search area has to be divided into subareas for assigning individual search units to search for the target simultaneously.
- Search is conducted over water or flat area.
- Most effective when the location of the target is known/predicted within close limits, i.e. a more concentrated search strategy.

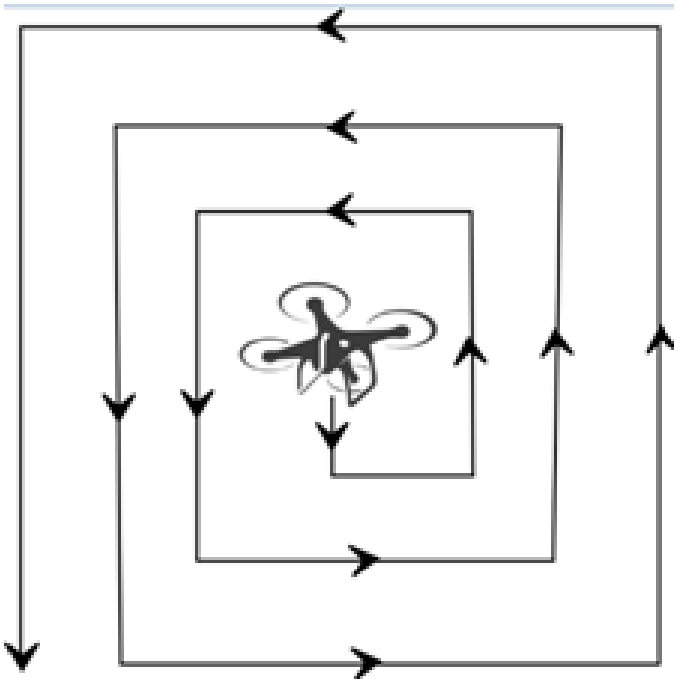

Figure 5.5: Expanding Square Search Pattern

5.3.3 Contour search

The search pattern of this search strategy may be described as a downward spiral motion as shown in Figure 5.6 This search strategy is applied around mountains or-valleys with sharp changes in elevation as the other search patterns are not practical in these circumstances.

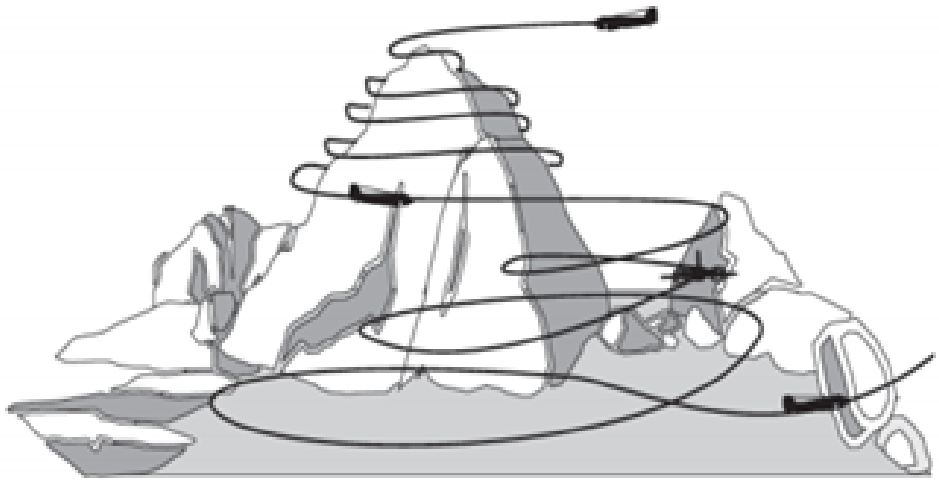

Figure 5.6: Contour Search Pattern

5.4 Navigation System

5.4.1 Sensor Coverage

The sensor's ability to detect a victim in the prevalent conditions will significantly affect the maximum altitude at which the UAVs can operate.

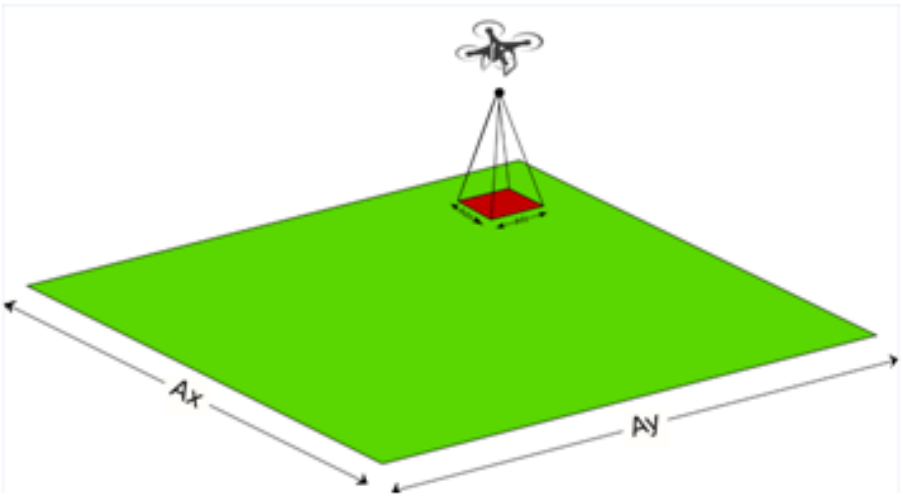

Figure 5.7: Sensor Coverage

Let's consider that the sensing areas on the ground cover surface $A = A_x * A_y$ (green area in Figure 5.7) a set of $A_d(h)$ cells, where the surface $A_d = A_{dx} * A_{dy}$ (red area in Figure 5.7) depends on the altitude of the UAV. The dimension of the sensing area increases with the height of the UAVs. UAVs at high altitude have a greater sensing coverage than UAVs at low altitude but they also have lower sensing resolution (smaller detection probability).

Height (m)	α_h	β_h
50	0.243599	0.000000
100	0.028369	0.000000
150	0.026099	0.046211
200	0.001110	0.046745

Table 5.1: Observation Model

Lets also consider a sensing model accounting for false positive and false negative:

- The probability of sensing a target at height h in the presence of the target $= 1 - \beta_h$
- The probability of not sensing a target at height h in the presence of the target $= \beta_h$
- The probability of not sensing a target at height h with no target $= 1 - \alpha_h$
- The probability of sensing a target at height h with no target $= \alpha_h$

With α_h $(0 \leq \alpha_h \leq 1)$ and β_h $(0 \leq \beta_h \leq 1)$ represent the false alarm and missed detection probabilities and vary as a function of the height h of the quadrotors. The values accorded to α_h and β_h at the different searching heights h are summarized in Table 5.1

For this work a fixed searching high of 100m above the earth is used in order to maintain an acceptable values of α_h and β_h that avoid any possibility of missed detection and minimize a false alarm.

5.4.2 Target Detection

algorithm 5.1 presents the searching algorithm used depending on the searching strategy, where the UAV scan the hall area and report any target detection to the base station. The mission is completed whenever all the targets are detected (if the number is limited) or the hall area is scanned, it also can be canceled if any fault is occurred, for all the cases the base station is reported about the mission situation.

Algorithm 5.1: Target Detection

```
Initialization
P_i(X_i, Y_i, Z_i) = Initial location
A_d(A_dx, A_dy) = Detection Area
A(A_x, A_y) = Searching Area
Update(Searching Strategy)
while true do
    P = P'
    GetObservation(P)
    if target detected then
        Report Base Station
    end
    Searching Strategy (P)
    P'=NextMove(P)
end
```

5.5 Simulations Results

The first part of this section is concerned to test the ability of the quadrotors to track the desired generated trajectory of the different SAR strategies patterns, while the second part compares the different SAR strategies over a case study of survivals SAR operation. Several search areas are designed to demonstrate the quadrotors ability of scanning different shapes, detect any possible survivals, and then report their position to the base station or support team.

Three types of search areas are designed for the SAR strategies in order to simulate different scenarios: a flat, mountainous and a hybrid area for the study case, all the areas are supposed to be with a total surface of 1000m * 1000m.

For the mountainous simulation environment, an exponential function is introduced to emulate mountains which are the main threats of the UAVs in this experiment (Equation 5.2):

$$z(x,y) = \sum_{i=1}^{n} h_i * exp(-(\frac{x - c_{xi}}{g_{xi}})^2 - (\frac{y - c_{yi}}{g_{yi}})^2) \tag{5.2}$$

This function gives altitude of the given coordinate (x,y) in the simulated terrain. Where, h_i controls height of peak i; c_{xi} and c_{yi} mark central position of peak i; g_{xi} and g_{yi} control peak gradient in x and y orientation respectively.

5.5.1 Expanding Square Search (ESC)

For this search strategy the quadrotors is supposed to scan a flat surface from a departure point P_0 located in the middle of the covered area with $P_0(X, Y, Z) = (500, 500, 0)$ then track the reference trajectory generated by strategy.

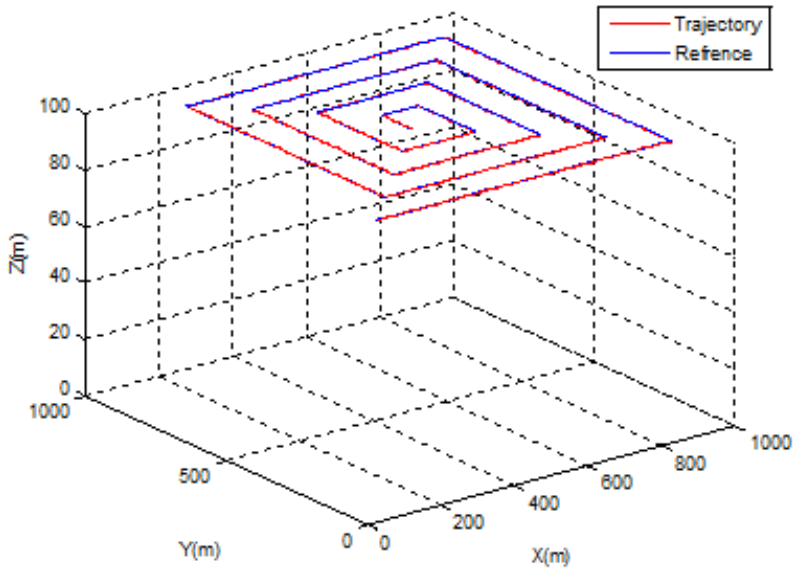

Figure 5.8: ESC Trajectory Tracking

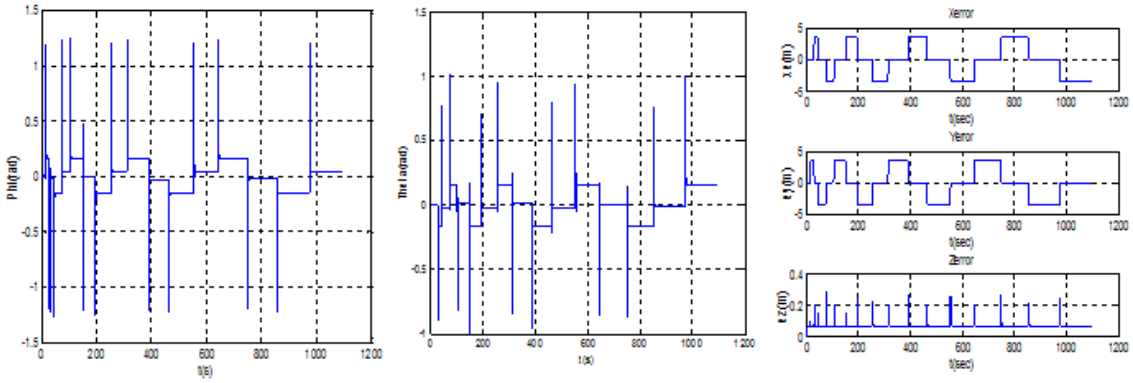

Figure 5.9: ESC Control Outputs

As shows Figure 5.8 the quadrotors UAV was able to pass over all the prescribed way points, and track the reference path using straight lines without any oscillations and high accuracy.

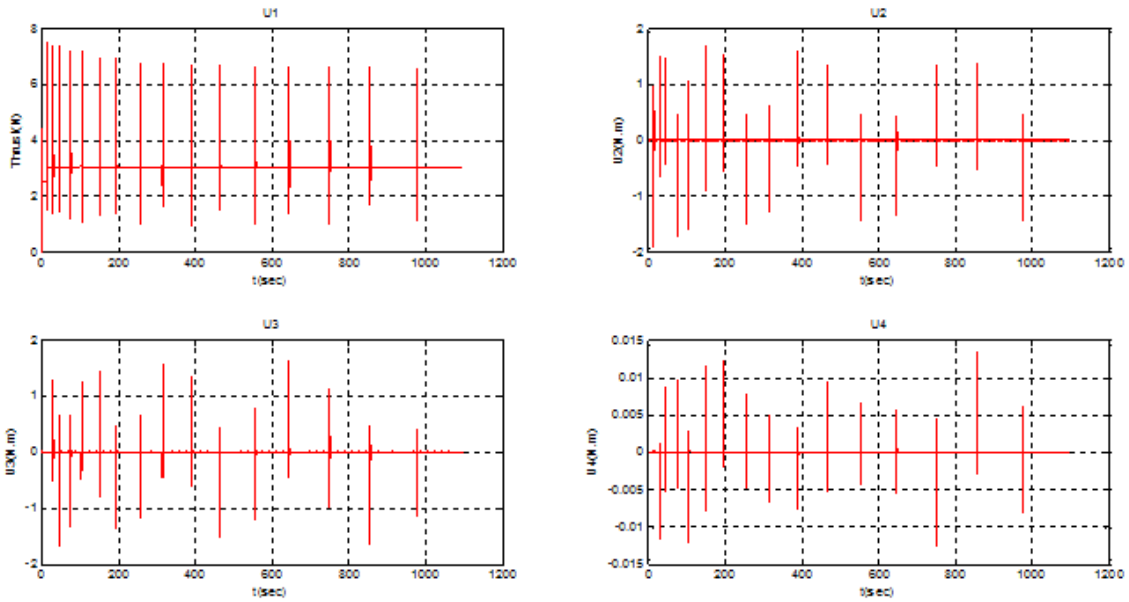

Figure 5.10: ESC Control Inputs

The control outputs are shown in Figure 5.9 where a good accuracy of the path tracking is presented specially for the altitude hold. High attitude degrees (up to 1 rad) are used during curving movements when passing from a way point to another.

Figure 5.10 shows that the control signals remain within the limits of the power delivered by the actuators. These results show that all the constraints and the assumptions considered during the modeling and the control design are successfully respected.

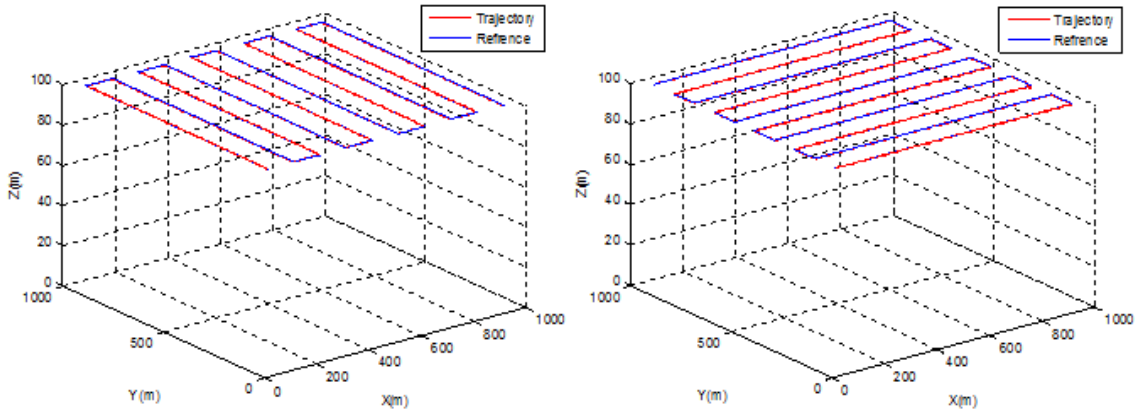

Figure 5.11: CLS Trajectory Tracking PT Trajectory Tracking

5.5.2 Parallel Track (PT) and Creeping line Search (CLS)

For this strategy the quadrotors is supposed to scan the same flat area used in the precedent strategy but this time from two different departure points: $P_{0C}(X, Y, Z) = (50, 50, 0)$ for the CLS and $P_{0P}(X, Y, Z) = (950, 50, 0)$ for the PT strategy. The obtained results for both CLS and PT 3D trajectory tracking are shown in Figure 5.11.

From Figure 5.11 it is clear that the quadrotors was able to track the way points with high accuracy in both cases and scan all the desired area.

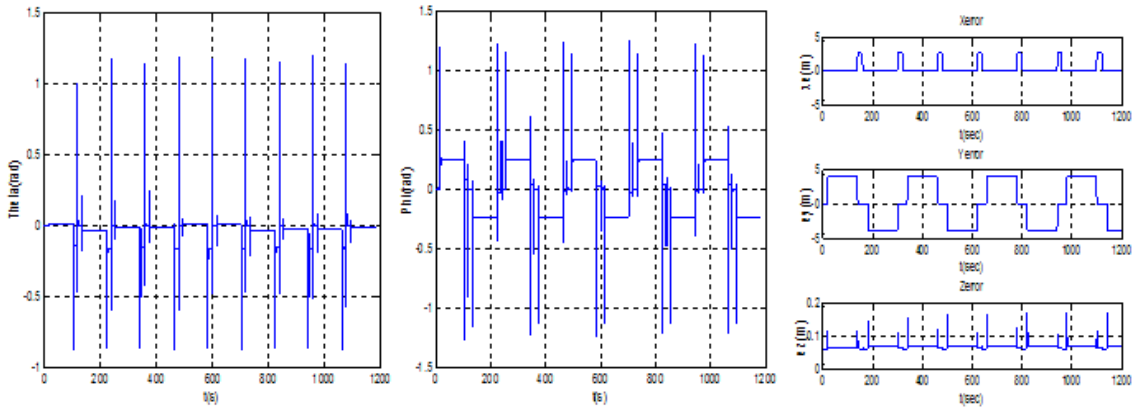

Figure 5.12: CLS Control Outputs

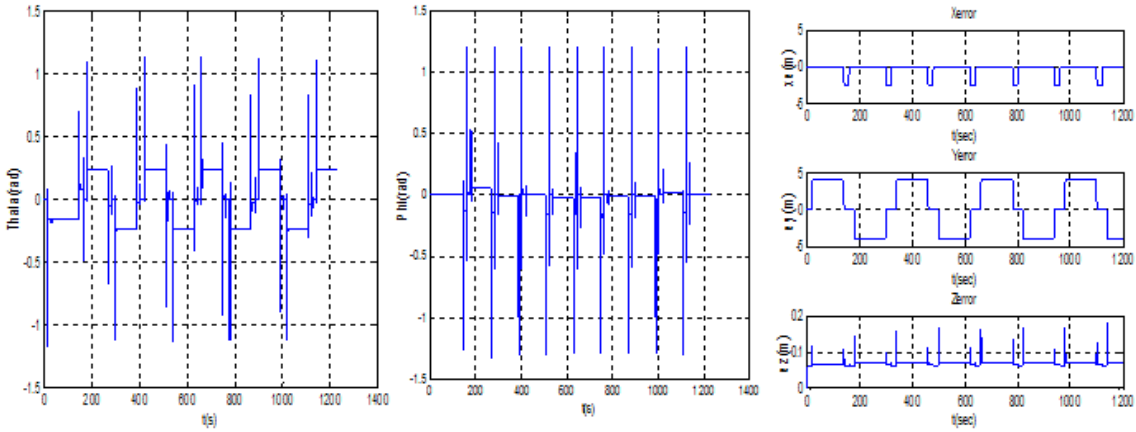

Figure 5.13: PT Control Input

Figure 5.12 and Figure 5.13 despite the control outputs, where the controller was able to maintain the quadrotors stability during the path tracking and spatially on curving corners when changing the waypoints.

The obtained control inputs illustrated in Figure 5.14 Figure 5.15 for the CLS and PT strategies are the same and remain within the desired functional range of the quadrotors engines.

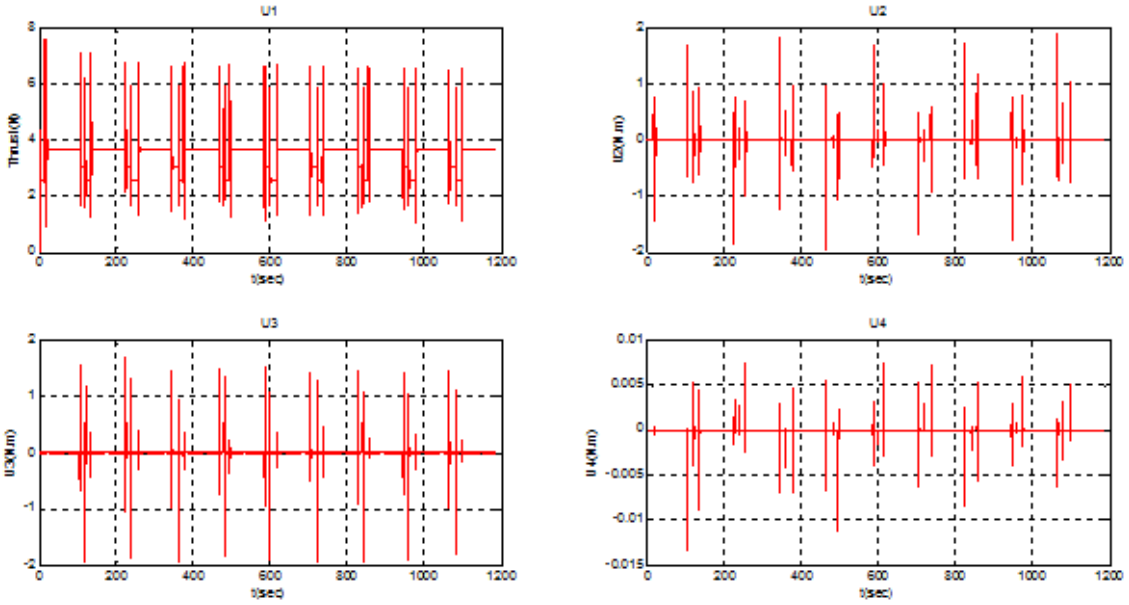

Figure 5.14: CLS Control Outputs

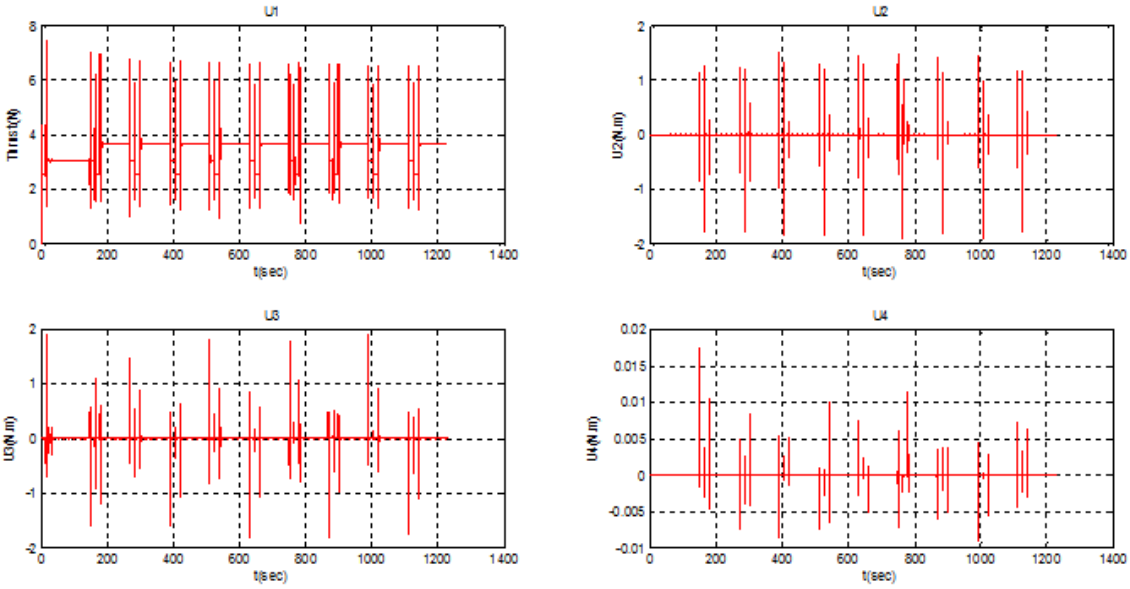

Figure 5.15: PT Control Outputs

A comparison between the different SAR strategies for a flat area scanning is shown in Table 5.2

From the obtained results it is clear that the controller was able to optimize the energy consumption and with high accuracy for all the strategies, but the PT strategy is supposed the best since it offers a more complete covered area with an acceptable searching time and controller path tracking accuracy.

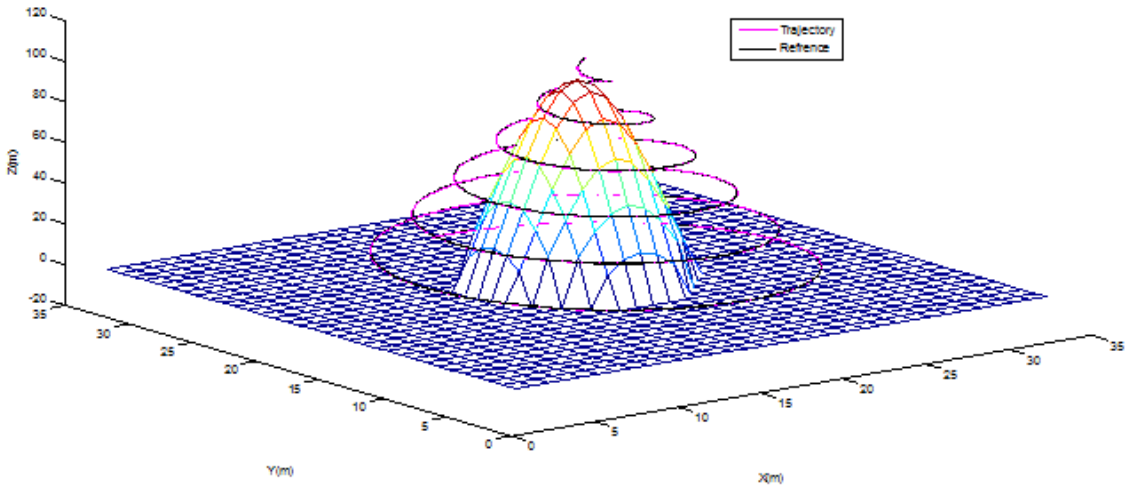

Figure 5.16: CS Trajectory Tracking

Strategies	Search Time (min)	Battery Consumption (%)	Covered area (%)	Fitness (IAE)
ESC	17	76.31	80	0.1584
PT	20	81.22	100	0.1534
CLS	20	80	100	0.8161

Table 5.2: Flat Area SAR Strategies Comparison

5.5.3 Contour Search (CS)

This strategy is often used when scanning a mountain area because of its spiral path that creates a contour. A virtual mountain with a 100m of altitude is simulated, where the quadrotors is starting from a departure point P_0X,Y,Z = 50 , 50, 0 and track the desired path around the mountain. Figure 5.16 Figure 5.17 Figure 5.18 illustrate the simulation environment as same as the obtained trajectory generation and tracking of the quadrotors. The obtained results are shown in 5.3

Strategies	Search Time (min)	Battery Consumption (%)	Covered area (%)	Fitness (IAE)
Spiral	6	13.44	-	0.0046

Table 5.3: CS Strategy Simulation Results

Figure 5.16 shows that the quadrotors was able to track the generated path and cover the entire mountain with a very high accuracy, and a minimum of battery consumption (Table 5.3) in less than 6 min.

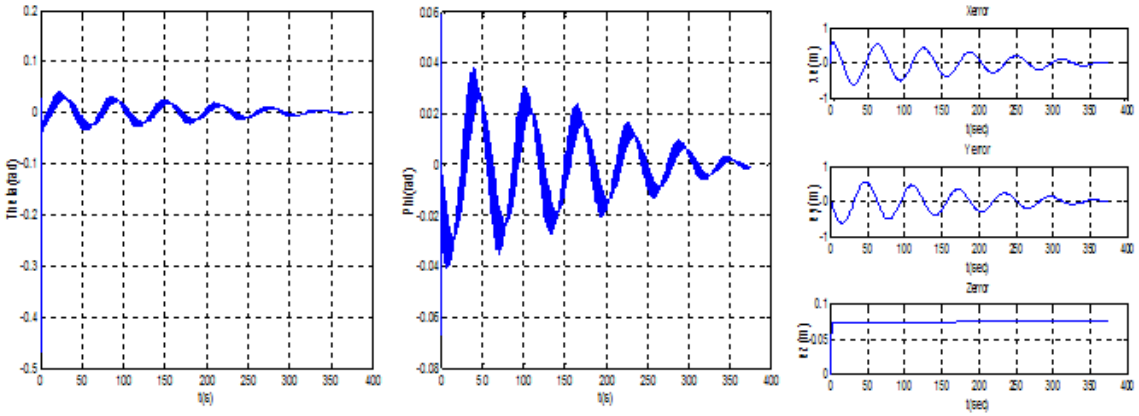

Figure 5.17: CS Control Outputs

The control outputs dissipated in Figure 5.17 are validating the controller effectiveness and optimally where a high accuracy is reached with low attitude degrees. The obtained control inputs shown in Figure 5.18 oscillate with low frequency supported by the quadrotors engines.

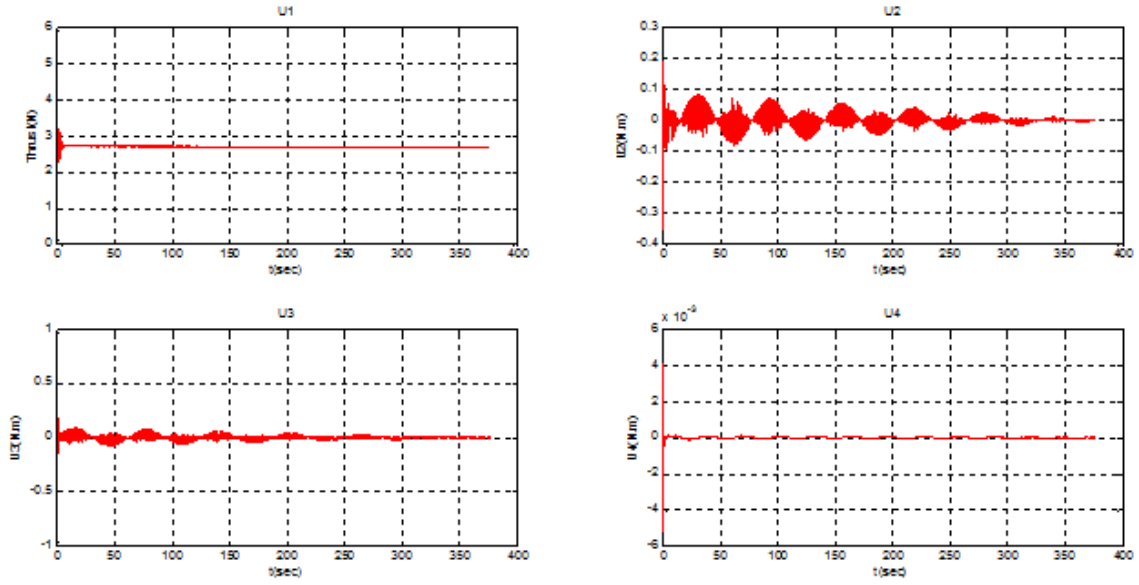

Figure 5.18: CS Control Inputs

5.5.4 SAR Strategies Comparison

In this section the study case is to scan a hybrid flat and mountainous area and search for possible survivals. The surface is of 1000m * 1000m and contains two mountains of 100 m of altitude. All the previous SAR strategies are applied in order to find 10 persons (red dots in Figure 5.19) that are randomly distributed all over the area.

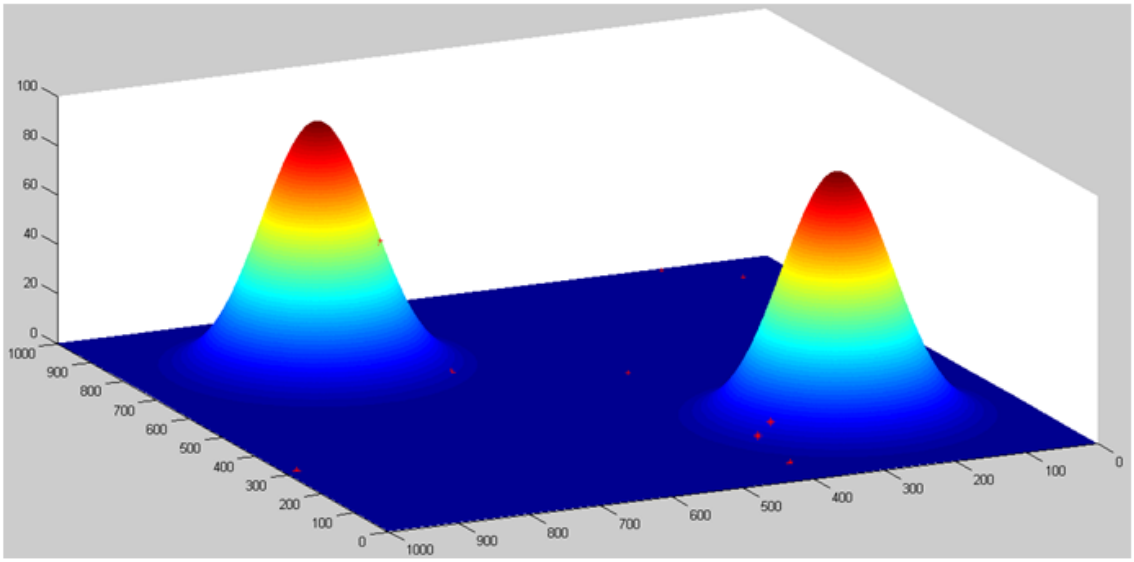

Figure 5.19: SAR Study Case Environment

A comparison between the different SAR strategies is made, the obtained results are shown in Figure 5.20

From Figure 5.20 it is clear that only the CLS and PT strategies were able to cover the desired area and detect all the 10 survivals but after a long time (about 25 min). For the ESC strategy only 70 of the survivals was detected in less than 20 min. For the CS strategy, 60 of the survivals are detected in only 9 min, which is a very interesting time when comparing with the other strategies for the same percentage. The explication behind these results is that the CS is local concentrated searching strategy that covers the surfaces with a high detection probability which means a less searching time, where the other strategies are more generalized searching strategies that cover the entire area and detect all the possible survivals but with a long searching time.

The departure searching point is also an other comparison parameters that must be cited, because generally it is so critical from a point and not from the other due to the complicated nature of the SAR operations.

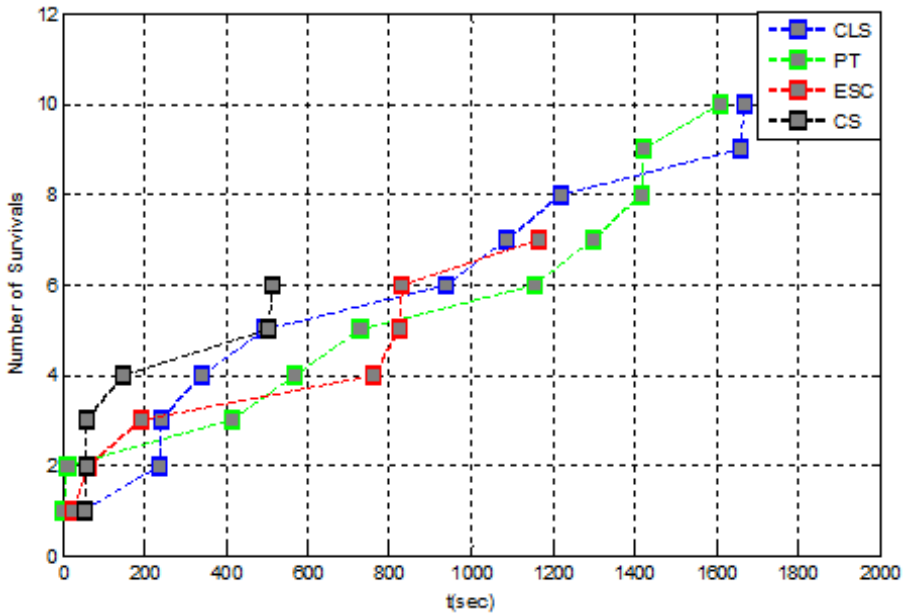

Figure 5.20: SAR Strategies Comparison

5.6 Conclusion

This chapter studied the application of Search and rescue operations using a quadrotors drone. Many SAR strategies defined in the IAMSAR manuals were applied. The obtained results of the trajectory generation, and the path tracking were juged to be satisfactory, since the quadrotors has successfully followed the desired SAR strategy generated path within the functional range and with respect to all the limitations.

A study case of an area scanning for possible survivals geo-localization and detection was simulated. Different SAR strategies were applied and compared, the final results give an overview of the advantages and the drawbacks of every strategy.

The next step for this research will focus over the use of a UAV swarming as a multi-agents system to reduce the research time and improve the efficiency. A combination of the different SAR strategies can also applied to benefit of the advantages of every strategy.

Chapter

6 SAR Operations Using Multi-UAVs

Contents

6.1 Introduction

Due to the limited flight time of such UAVs (a few ten minutes) and the fact that search and rescue missions are time critical, the use of multiple UAVs rather than one UAV plays an important role. This also raises the demand for reliable wireless networking between the UAVs and communication to the base stations.

Many works were published in the field of SAR operation using Multi-UAVs such as [34] where a SAR system using quadrotors with an embedded camera is presented. Paper [43] proposes the use of multi-agents flooding algorithm for the SAR operations in an Unknown terrain. For path planning of the agents swarm, the reader can refer to [7] for a multi-target approach of a multi-agents SAR drones. SAR operation can also be executed by a team of heterogeneous robots; an online collaborative mission planning is explained in [5].

In the other hands coordination and cooperation strategies between the different agents can be found in [10, 55, 59, 63], several subject were treated such as cooperative path finding, task allocation and distributed decision and control systems. References [2, 12, 51] focus over search strategies and collaborative searching within multi-agents system. For this propose an optimal hybrid PSO-GA algorithms for swarm control is given in [22].

International civil authorities and emergency services start to give an important part to SAR operation using UAVs due to the difficulties in adequately managing crises, The European Union ICARUS project on this subject is introduced in [24]. The ICARUS project proposes to equip first responders with a comprehensive and integrated set of unmanned search and rescue tools, to increase the situational awareness of human crisis managers, such that more work can be done in a shorter amount of time.However the previous cited works describe the different parts of a SAR system, but no one has presented a full designed autonomous system that takes into consideration all the contributed parts (i.e. mission planning, autopilot system, search strategy and targets geo-localization).

The objective of this work is to develop an efficient search and rescue methodology for employing a swarm of UAVs in order to efficiently locate survival targets within the search region and rescue them.

This chapter is organized as follow: section 6.2 introduces the multi-UAV SAR system design. section 6.3 explains the mission planning and the different multi-UAVs searching strategies, while section 6.4 is concerned by tasks allocation. Simulations and full scenario of SAR operation using the designed system are in section 6.5. Finally, in section 6.6 a conclusion is given.

6.2 System Design

For the proposed SAR application, a system of homogeneous quadrotors UAVs is designed. All the UAVs can coordinate and cooperate their actions in order to complete a desired mission. The human interaction is supposed to be minimized since all the UAVs are considered to work in a decentralized/distributed architecture, and failure of one UAV will not affect the whole mission. Each UAV is capable of performing its mission independently due to the embedded sensors. During the SAR operation, all the UAVs are able to share their collected data (e.g., locations, states, and images) with a direct communication among themselves and with the control station. The architecture of the designed system is shown in Figure 6.1.

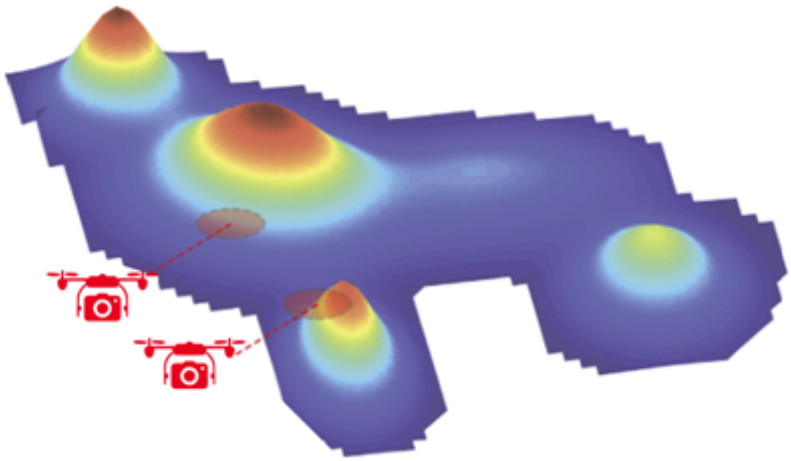

Figure 6.1: SAR Scenario

SAR operation is executed using a small group of pre-defined swarm flying patterns. The flying patterns are based on IAMSAR standard search strategies that guarantees a successful completion of the mission. The initial number of UAVs in the pattern is decided (by ground control station) prior to launch, and each UAV is given the pattern, its position in the pattern, and the dictionary of allowable alternative patterns. The communication link between the UAVs is assured via keeping a limited distance in order to keep all agents within the communication rang. Information about the UAV status and the collected data must be exchanged between UAVs and the control station. Task re-allocation is used to modify the search in the event of a UAV loss, this allows an efficient and automatic reconfiguration of the UAVs array, in the case of a UAV malfunction.

Each UAV has the capability to detect and avoid the obstacles in its environment. The swarm is also robust over the external disturbances and the bad weather conditions, in the case of failure to track the search pattern due to wind, the mission is then cancelled and the base control station is informed.

For a complete mission, the UAVs are divided into two teams depending on the allocate mission, one for search and the second for rescue as shown in Figure 6.2 Each UAV is equipped with inboard sensors that are adequate with the mission requirements. The goal of the mission is to track the search strategy pre-defined pattern and locate the survival person using on board sensors. Once the target is identified, the survival geographic position is sent to the other swarm agents and to the ground control station, using a direct communication link or via the team leader which play the role of a 'relay' that transmit the information to the control ground station. Once the survivals are positioned, the rescue team starts its mission to feed the survivals with firs aids, water and food.

Rescue operation represents the second leg of the SAR operation. Depending on the number of the used UAVs, the rescue can start after completing the search operation (all the survivals positions are known, they are gathered then in form of collections), or it can be real time operation, where a rescue UAV is dedicated to every detected survival, which required then a lot number of rescue UAVs.

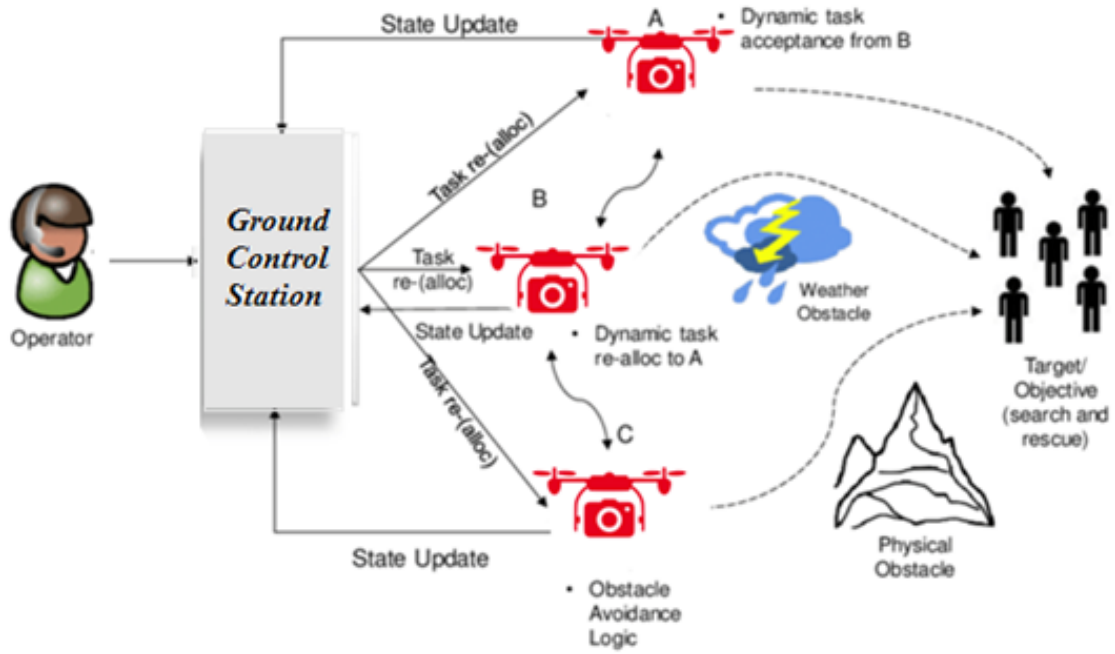

Figure 6.2: SAR Operation Description And Challenges

Having distributed architecture ensures that the UAVs can coordinate with each other directly and not necessarily all the traffic refer to the base station. An autonomous control for navigation and detection with distributed control for collision avoidance of the UAVs is also introduced. Though, if a centralized structure is not applicable or it does not improve the system performance (e.g., pre-planning), decisions are made in a distributed manner. SAR missions are summarized into the following phases:

1. **Pre-planning**: The human operator defines the search region at the control base station. The optimal flight paths for all UAVs are computed to reduce the required time to search the area. Generated plans including the way-points are sent to individual UAVs.

2. **Searching**: The UAVs autonomously follow their predefined way-points while scanning the ground. The detection, collision avoidance, and data transfer are active at this phase.

3. **Detection**: Upon detection of a target, the detecting UAV transmit the survival geographic position to the leader/ground control station.

4. **Rescue**: The rescue team starts to provide the survivals with the necessary equipments.

To summarize, here are some contributions and the corresponding features that distinguish our work from others:

- For a multi-UAV system to be robust, it needs to be fault tolerant, such that the failure of one device does not deem the mission unsuccessful.

- For search and rescue scenarios, time is critical and we need also to satisfy some requirements in terms of mission time and energy optimization.

- In this work low cost, small scales UAVs with a maximum payload are used, which impose then to respect the UAV's load limitations. To address these challenges, we have divided the UAVs into two teams: Search team and rescue team. The UAVs of the search team are equipped with high quality sensors and communication devices, to perform the detection mission and coordinate their actions. In the other hands, the rescues UAVs have no detection sensors in board, so they can carry instead a first aid kit, water or food for the survivals.

- Having distributed communication ensures that the UAVs can communicate with each other directly and not necessarily all the traffic is sent through the base station. We also introduce, in our system, an autonomous control for navigation and detection, and a distributed control for collision avoidance of the UAVs. Though, if a distributed structure is not applicable or it does not improve the system performance (e.g., pre-planning), decisions are made in a centralized manner. In the next section we explain our multi-UAV system architecture in more detail.

6.3 Mission Planning

As illustrated in Figure 6.3, during a typical scenario the quadrotors UAVs will be used to deploy an area of interest, perform sensory operations to collect evidence of the presence of a victim, and report their collected information to a remote ground station or rescue team.

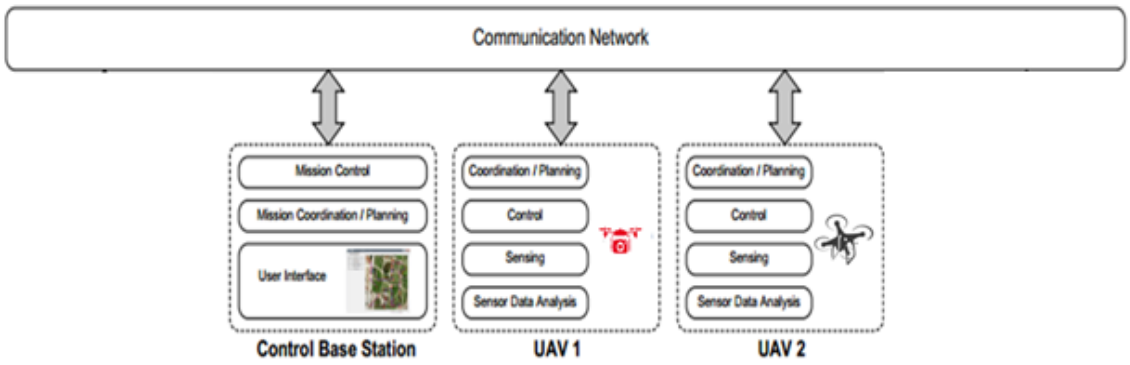

Figure 6.3: SAR System Architecture

This section explains the context of the designed system and presents the different search strategies for Multi-UAV swarms. The proposed mission is that the SAR UAVs scan the desired area, search for any possible survivals, and then rescue them just after a disaster (tsunami, earthquake, hurricane, explosion,...).

There are two main strategies to scan the disaster area using Multi agents UAVs: A centralized/decentralized squadron search strategy by creating a "squadron" of drones that fly in formation, or a distributed independent search strategy that is based on dividing the area into rectangular sub-areas, each sub-area being assigned to a drone. Both strategies are investigated in the next sections.

6.3.1 Squadron Search

The advantage of the "Squadron search" is that the drones fly close to each other (see Figure 6.4) in a centralized or decentralized architecture. Thus, all drones can easily converge to a detected event when one of the drones detects one. However, the drawback of this solution is that the provided substitution network does not cover the whole area to explore: as the drones fly close to each other, they can easily communicate with each other, but the squadron is concentrated in a small area and is isolated from the remainder of the network.

Figure 6.4 shows the exploration search using 6 UAVs (2 rows of 3 drones each). The red path shows the trajectory for drone 1. Depending on the shape of squadron, the separation distance c squadron is estimated in accordance with the width W of the area and the number of UAVs (by assuming that the length of the trajectory is the same for all the drones).

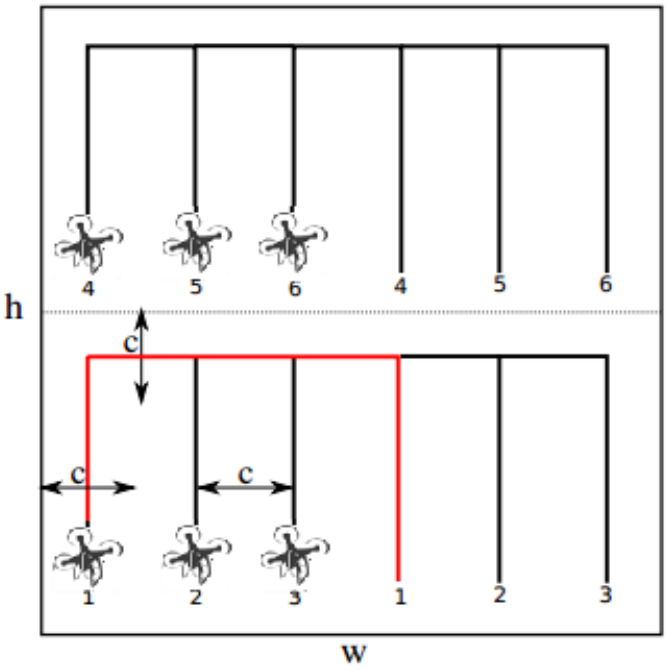

Figure 6.4: Squadron Searching Strategy

The IAMSAR proposes the following searching pattern for the Multi-ships searching (Figure 6.5):

- Parallel sweep: for use by two ships.
- Parallel sweep: for use by three ships.
- Parallel sweep: for use by four ships.
- Parallel sweep: for use by five or more ships.

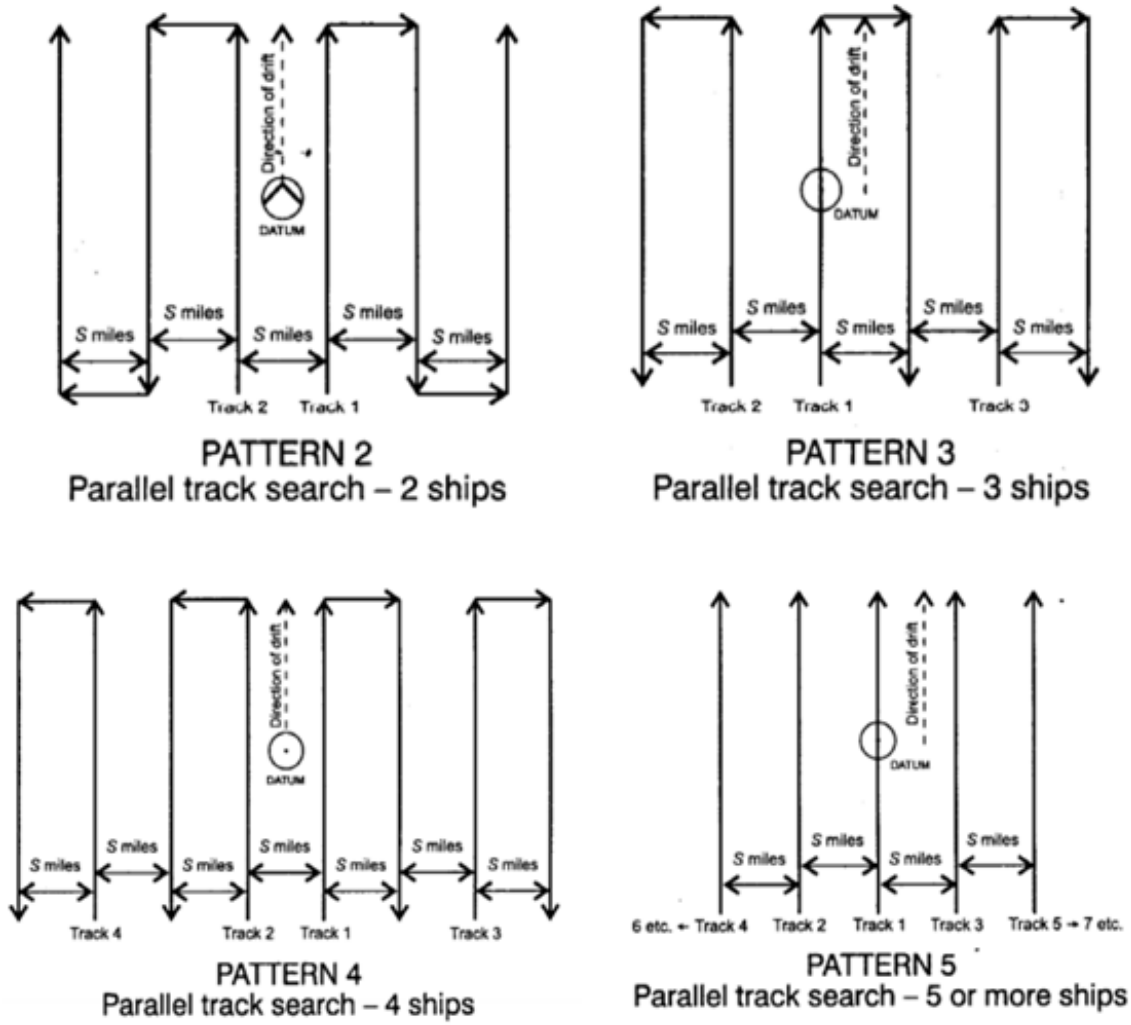

Figure 6.5: Multi-ships IAMSAR Standard Searching Strategy

6.3.2 Distributed Independent Search

This strategy is based on dividing the area into rectangular sub-areas, each sub-area being assigned to a drone (or a swarm of drones). The advantage of the "independent search" is that the substitution network provided by the drones covers a wider surface. Moreover, as for the squadron exploration, this network is very stable (the distances between the drones remain constant). However, the drones are farther from each other, and this implies a larger delay when they need to gather at a given location.

Algorithm 6.1: Tasks Allocation

Input: N: Number of agents, A: list of N agents where each one has an ID from 1 to N, $T = \{T(N), T(N-1), ..., T(1)\}$: list of N topology according to the agent's number, $P(T)$: list of N paths according to each topology.

```
1  begin
2      idL = 1;
3      L = A_idL;
4      while true do
5          T_ex = T(N);
6          P_ex = P(T_ex);
7          L = [T_ex, T, P_ex, P, A, idL];
8          ALLReady = 0;
9          for i ← 1 to N 1 do
10             if i ≠ idL then
11                 send(i, ready = false) by L;
12                 wait(i, ready(i)) = true by L;
13                 ALLReady = ALLReady + 1;
14             end
15         end
16         if AllReady = N − 1 then
17             for i ← 1 to N 1 do
18                 if i ≠ idL then
19                     send(Tasks − Allocation) by L;
20                 end
21             end
22             Failure = FailureDetect(A);
23             if Failure = idL then
24                 IdY = 0;
25                 for j ← 1 to N 1 do
26                     if i ≠ idL then
27                         State_i = election;
28                         idY = max(Y(A_i), IdY);
29                     end
30                 end
31                 L = A_idY;
32                 idL = idY;
33             end
34             A.remouveElement(A_Failure);
35             N = N − 1;
36             send(Tasks − Re − Allocation) by L;
37         end
38     end
39 end
```

6.4 Tasks Allocation

This section discussed the swarm behavior when a failure occurs in one of the agents. The same scenario as the previous sections is used. algorithm 6.1 explains the agent's behavior.

When a failure occurs in one of the swarm agents it stops its mission and executes an emergency landing. The neighbored agents in the communication range are then informed. Depending on the agent health, the task of the failed agent is allocated to the other agents. A re-planning of the task of every allocated agent is occurred and the new path is designed. Switching the scanning strategy is necessary to scan all the remained area.

6.5 Simulations Results

This section is concerned to test the ability of the quadrotors swarm to track the desired generated trajectory of the different SAR strategies patterns, while the second part presents a full case study of a SAR operation using multi-UAV swarm.

Several scenarios for the different multi-agents architectures are simulated to demonstrate the quadrotors swarm ability of scanning different shapes, detect any possible survivals, and then report their position to the base station or support team to start the rescue operation.

The supposed search area is the same as the previous chapters, a comparison between different number of agents is discussed.

6.5.1 SAR Strategies Path Tracking

6.5.1.1 Squadron Search

This section investigates the effects of using multi agents quadrotors in SAR operation. As mentioned before the quadrotors swarm mission is to scan the desired area using a parallel (sweeping) motion strategy. Figure 6.6,Figure 6.7 and Figure 6.8 show the results of the simulated scenario using 2, 5 and 10 quadrotors UAVs with centralized architecture.

Simulation results of the decentralized architecture with 2 and 5 UAVs are shown in Figure 6.9 and Figure 6.10

From the obtained results it is clear that the desired area was scan completely in all the cases. The target pattern was also tracked with high accuracy; the collision between the agents was avoided since the separation distance is maintained.

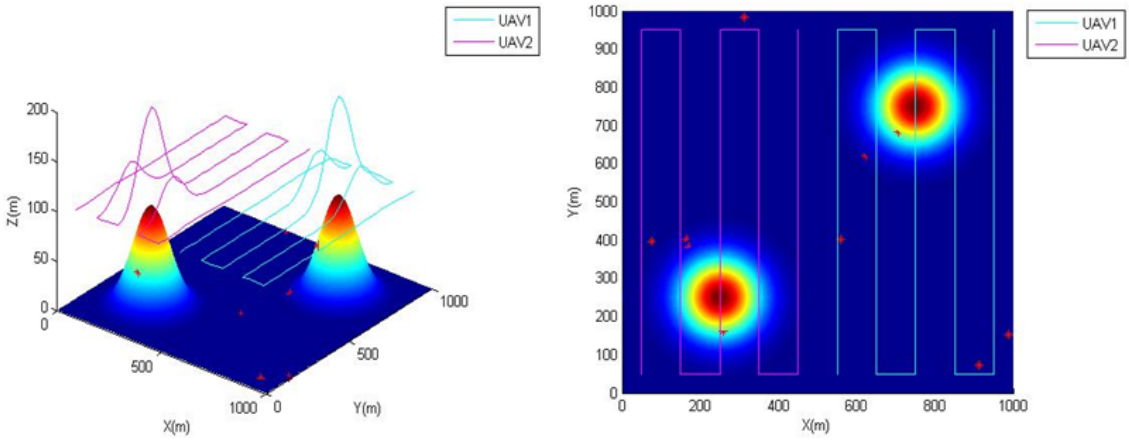

Figure 6.6: Two '2' Agents Decentralized Searching Strategy

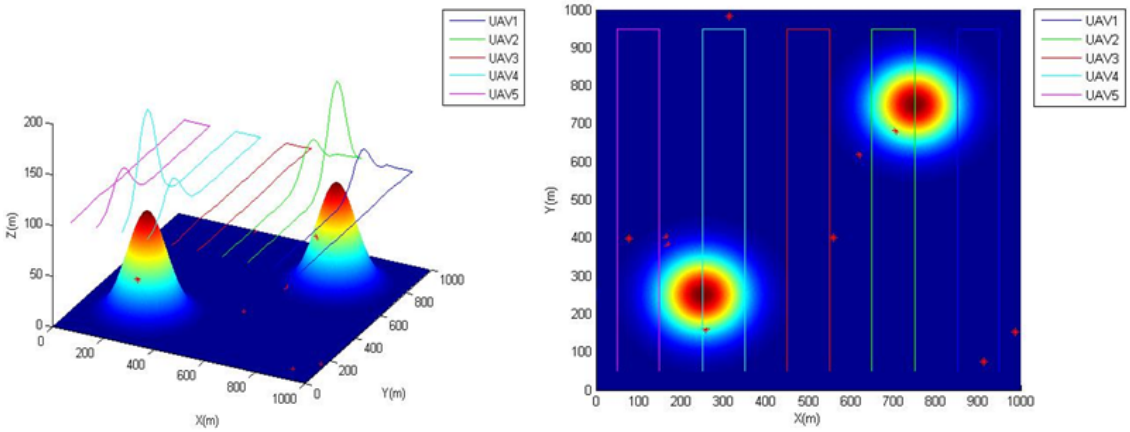

Figure 6.7: Five '5' Agents Decentralized Searching Strategy

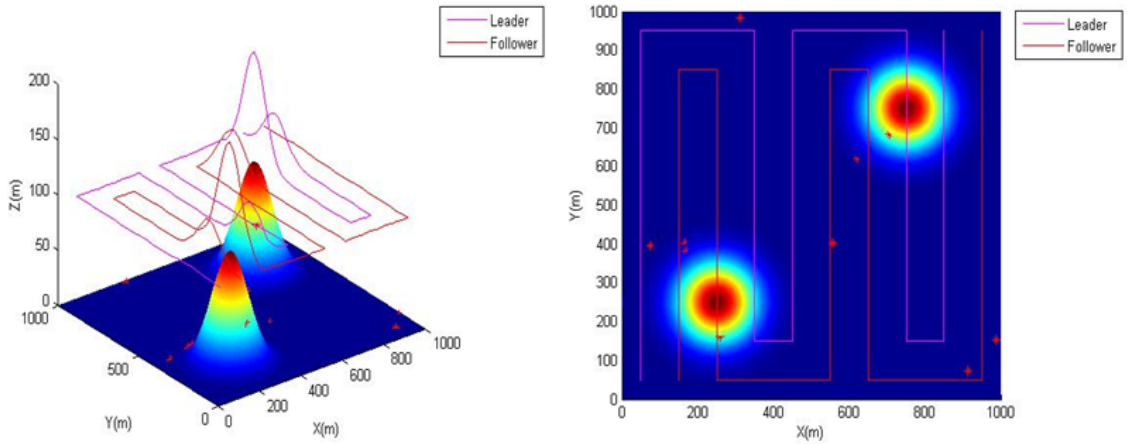

Figure 6.10: Two '2' Agents Centralized Searching Strategy

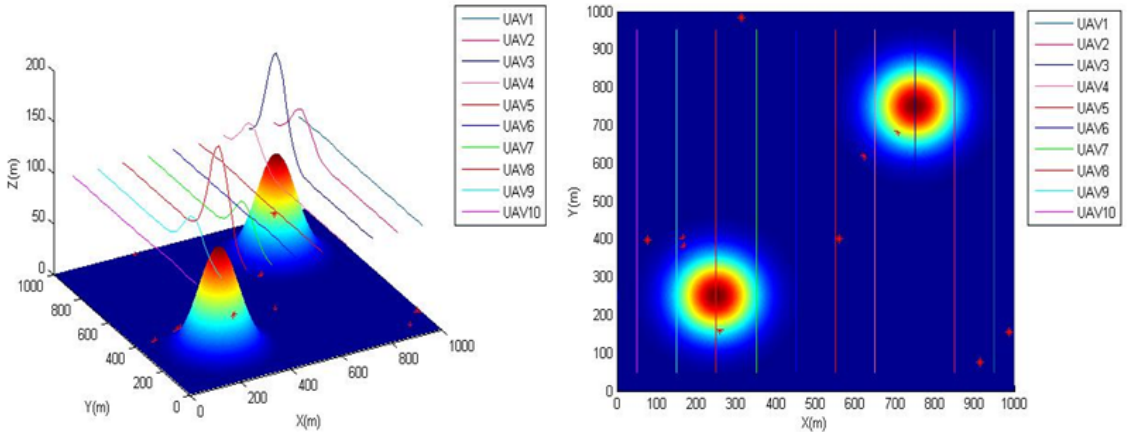

Figure 6.8: Ten '10' Agents Searching Strategy

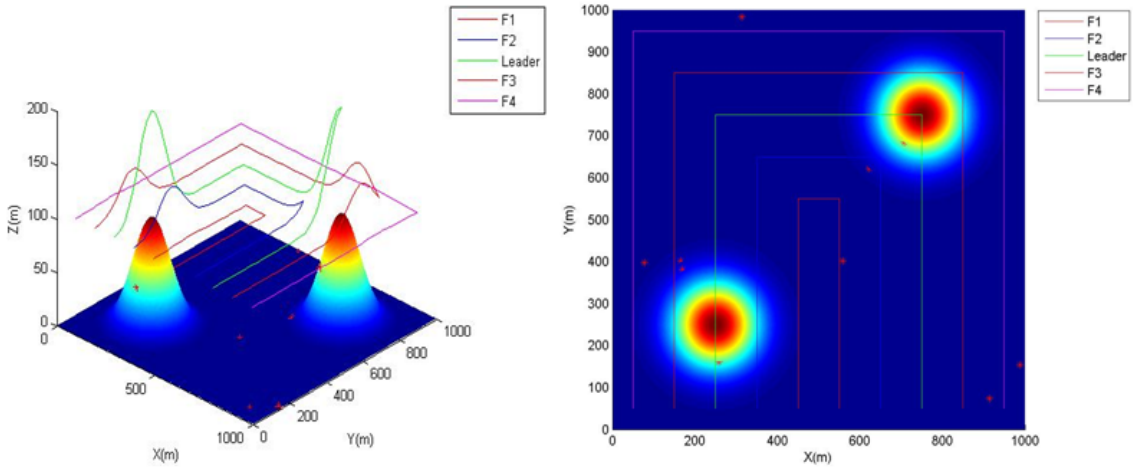

Figure 6.9: Five '5' Agents Centralized Searching Strategy

A comparison between the searching time for the centralized and decentralized architecture using 2, 5 and 10 UAV agents is shown in Figure 6.11.

Figure 6.11 shows that the searching time is decreasing whenever the number of agents is increasing, since the scanned area is divided between the different agents, the searching time is decreasing from 817 sec when using 2 agents to 330 sec with 5 agents and finally to 165 with 10 agents, which means that all the 10 survivals were detected in less than 3 min with 10 quadrotors agents.

It is also important to mention that the number of agents is related to many factors, such as the number of the available agents, the possible starting points, the communication links, the scanned area surface division, and the sensor coverage area. Saturation is reached whenever a limitation over the previous factors exists.

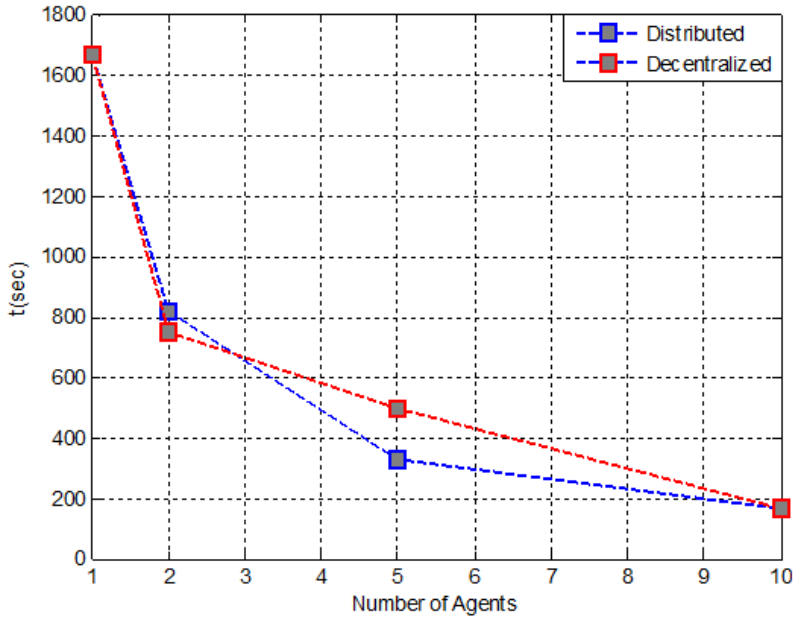

Figure 6.11: Centralized vs Decentralized Searching Strategy

6.5.1.2 Distributed Independent Search

A simulated scenario of an agent failure and task allocation using 5 and 10 agents is shown in Figure 6.12 and Figure 6.13.

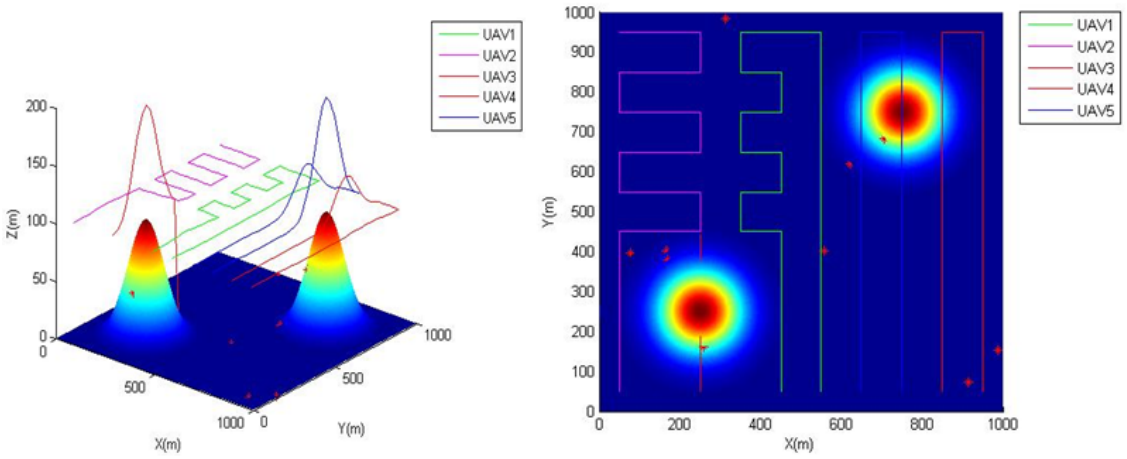

Figure 6.12: Five '5' Agents Searching Strategy With One Agent Failure

From Figure 6.12 and Figure 6.13 it can be noticed that a failure occurred to the agent 2 at Y = 450m, an emergency landing is then happened to prevent the agent from a crash or a collision with the other agents. In the case of 5 agents the failed agent communicates with its neighbored agents (agent 1 and 3) and allocates them with the remained task. The two allocated agents switch their searching strategy and cover the remained area.

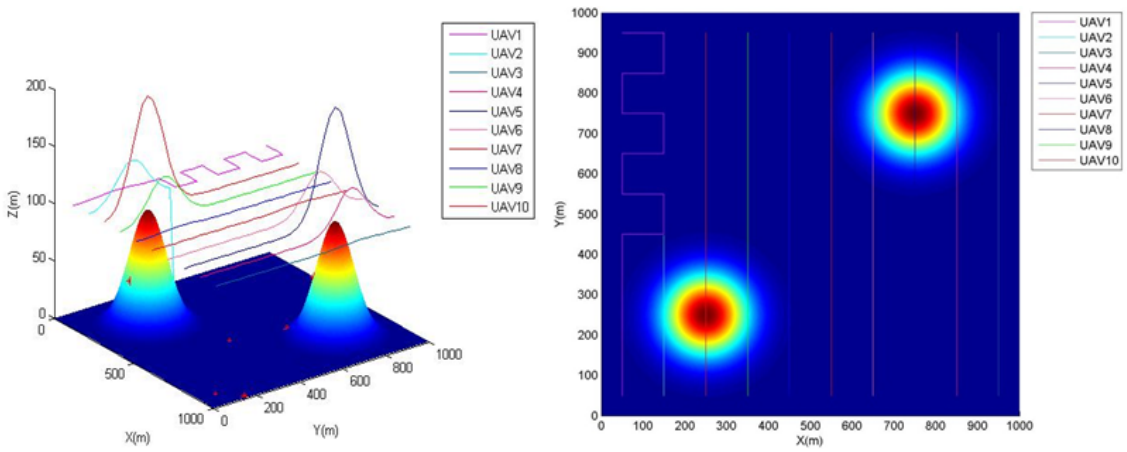

Figure 6.13: Ten '10' Agents Searching Strategy With One Agent Failure

In the case of 10 agents the only the agent number 1 is allocated because of the agent 3 bad health, the allocated agent switches its strategy and successfully complete the mission.

6.5.2 Search Rescue Case Study

In this section a full scenario of search and rescue operation is presented. The mission is then divided into two legs, one for search and survival detection and the second for rescue operation. The same environment is maintained as in the previous scenario but with new methodology.

6.5.2.1 Search Operation

For search and detection mission a decentralized distributed method is used. With 5 UAVs (One leader and four followers), the swarm is scanning the area with a hybrid expanding/contour searching strategy is used. As shown in Figure 6.14 the swarm agents start from points P_1(X,Y,Z) = (250 , 50, 0), P_2(X,Y,Z) = (750 , 50, 0), P_3(X,Y,Z) = (550 , 50, 0), P_4(X,Y,Z) = (450 , 50, 0) and P_5(X,Y,Z) = (500 , 50, 0) for UAV 1, UAV 2, UAV 3, UAV 4 and UAV 5 respectively. The base station is supposed to be at P_b(X,Y) = (500, 0).

UAV 5 is considered as the leader of the swarm since its mission is to coordinate, assist and direct the mission. Only the leader can have the communication with the base station, all the others UAV are supposed to be within the communication range of the leader.

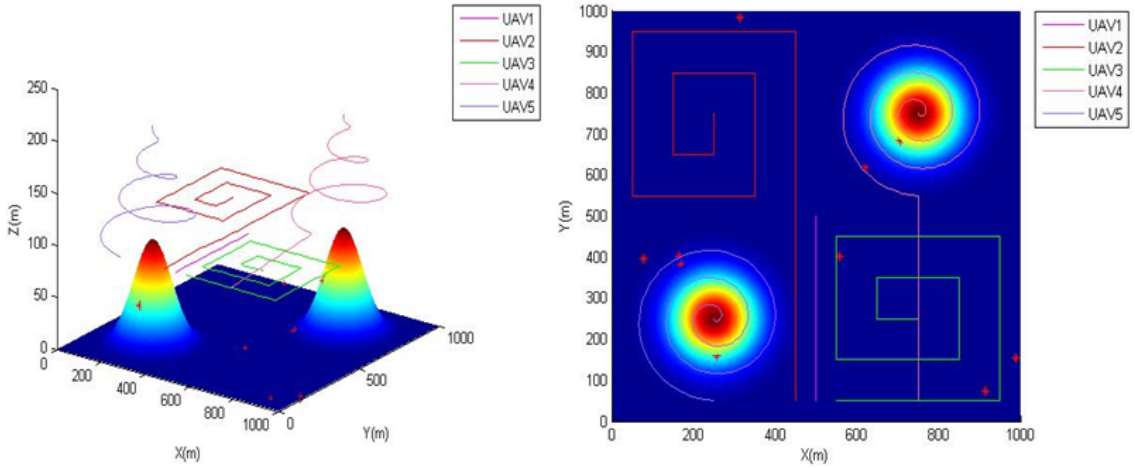

Figure 6.14: Five '5' Agents Distributed Searching Strategy

By the end of the mission the swarm has a meeting point to exchange the information about the searching operation. The meeting points are as follow: P_{m1}(X,Y,Z) = (250 , 250, 0), P_{m2}(X,Y,Z) = (750, 750, 0), P_{m3}(X,Y,Z) = (750, 250, 0) , P_{m4}(X,Y,Z) = (250 , 750, 0) and P_{m5}(X,Y,Z) = (500, 500, 0). Once the survivals positions are collected, the leader send the information to the base station, the rescue operation can then start.

6.5.2.2 Rescue Operation

After receiving the survivals position from the leader of the swarm, an estimation of the survivals collection is done in order to minimize the number of the rescue UAVs. In our case six rescue UAVs denoted R1...R6 are used to rescue six collections of survivals (see Figure 6.15). The collection is defined by the survivals in a range of a circle with a radius of 100m.

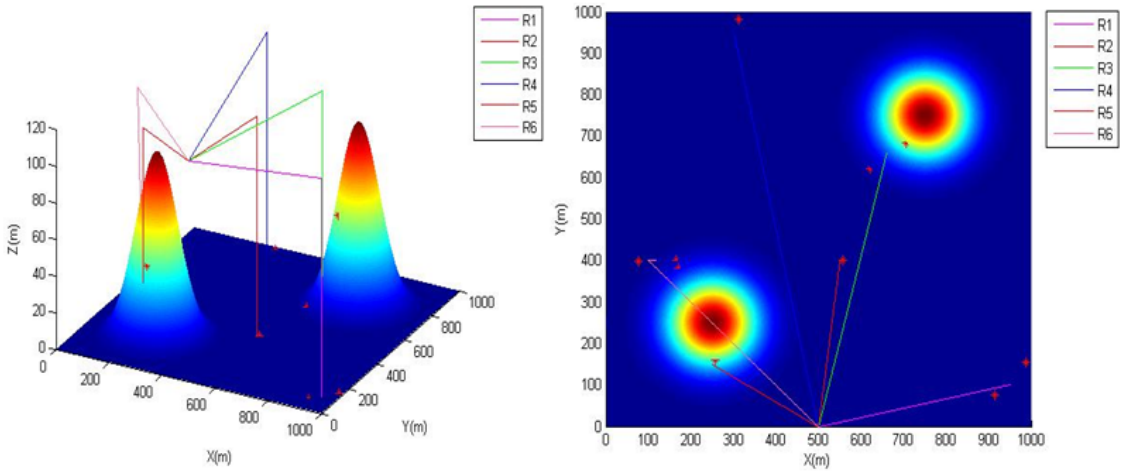

Figure 6.15: Rescue Agents UAVs

Each rescue UAV is equipped by a first aid kit, and it starts from the base station and go directly at the desired collection. All the UAV are flying at a flight level that is situated at 120 m, which means that they are above the mountains, and bellow the searching team, no collisions can occurs !

6.1 presents the rescue time for each collection. All the survivals are rescued in less than 3 minutes, which reflect the efficiency of the operation.

Table 6.1: Rescue Operation Time

Survivals Collection	Time (sec)
1	100
2	70
3	100
4	140
5	60
6	90

6.6 Conclusion

This chapter studied the application of Search and rescue operations using multi-agents quadrotors drones. SAR strategies were developed in accordance with multiple UAVs possibility. All the obtained results for the different strategies were judged to be satisfactory, since the quadrotors swarm has successfully followed the desired generated path, coordinate their movements and collaborate to finish the mission in the case of any agent failure. Both decentralized and distributed search strategies were applied and compared in order to obtain the most adequate method. A conclusion about the factors that affect the search operation and the different requirement is then given.

Finally a full study case of search and rescue operation is presented using five quadrotors UAVs organized in a decentralized distributed method with a hybrid expanding/contour searching strategy. The searching and rescuing time is evaluated to be very promising comparing to the other strategies.

Chapter 7

Conclusion

7.1 General Conclusion

This thesis was concerned with the application of a multi-agents quadrotors swarm in the search and rescue operation.

In a first part, the quadrotors UAV was modeled using a new approach that models all the quadrotors components. The interaction between them and their physical limitations was taken into consideration.

After that, the optimal trajectory tracking and control problem was studied. A new scheme based on a multi-layer optimal system. A combination of a double loop control structure with an inner attitude loop and an outer position loop based on the differential flatness approach is used in order optimizes the output spaces and gives them a direct relation with the states and their derivatives.

Once our quadrotors was modeled and controlled. We were concerned to the problem of the quadrotors swarming formation control. A double loop control structure based on a non-linear optimal backstepping controller was applied in the autopilot system to assure the high accuracy trajectory tracking, the energy optimization and the attitude stability, this was due to the MO-GA that offers a good solution for the multi-objectives optimization of the non-linear coupled systems. Many scenarios were proposed in order to test the performances and the robustness of the designed controllers. The quadrotors UAVs have successfully followed the desired path, hold their formation and eliminate the effects of the external wind disturbances.

As application, the SAR operation was selected. We studied the ability of a single quadrotors to track the SAR strategy pattern , detect the survivals and rescue them. Many SAR strategies defined in the IAMSAR manuals were applied. The obtained results of the trajectory generation, and the path tracking were juged to be satisfactory, since the quadrotors has successfully followed the desired SAR strategy generated path within the functional range and with respect to all the limitations. A study case of an area scanning for possible survivals geo-localization and detection was simulated. Different SAR strategies were applied and compared; the final results give an overview of the advantages and the drawbacks of every strategy.

Finally we have studied the application of Search and rescue operations using multi-agents quadrotors drones. SAR strategies were developed in accordance with multiple UAVs possibility. Both decentralized and distributed search strategies were applied and compared in order to obtain the most adequate method. A conclusion about the factors that affect the search operation and the different requirement was then given.

In order to validate the efficiency of our SAR system, a full study case of search and rescue operation is presented using five quadrotors UAVs organized in a decentralized distributed method with a hybrid expanding/contour searching strategy. The searching and rescuing time is evaluated to be very promising comparing to the other existed strategies.

7.2 Perspectives and Future Works

Although the SAR system using multi-agents quadrotors was proposed, and validated using a full scenario, the implementation of the system and real tests should be considered in further works. Therefore, in this section many perspectives for future works and some open problems are shown as follow:

- For a single quadrotors modeling, a more detail behind the aerodynamics theory should be given, in order to introduce more forces and torques that affect the quadrotors dynamics and then the control.

- A proposition of a quadrotors model that is adequate with SAR operation is so necessary, since the drone model used in this thesis is low cost, with limited capacities. The designed quadrotors must have a big battery capacity, more sensors, high quality camera and be able to carry out more loads to deal with any possible scenario and the most difficult environmental disturbances.

- In the control theory, we propose the design of a non-linear sliding mode optimal controller that reduces the chattering effects presented in the control outputs. It is also possible to use the MO-GA algorithms to find the optimal controller parameters.

- For the obstacle avoidance problem, this thesis was only concerned with the 2D obstacles; an extension of the obstacles to 3D would be more challenged! The developed algorithm is then able to find a 3D optimal trajectory far from the obstacles. After that a study for the swarm obstacle avoidance and a vertical separation should be added.

- SAR operations are wide research area, many scenarios for many possible environments can be simulated and tested. The most important part is that the quadrotors must detect the survivals with a high probability. The way how a drone could detect a survival is an interesting subject to study.

- Even the SAR strategies used in this thesis were so promising in term of searching and rescuing timing and high probability of detection. We suggest to compare those conventional strategies with the bio-inspired searching algorithms such as the ant colony and the particle swarm optimization.

Appendix

A Quadrotors Parameters

A.1 Quadrotors Parameters

Table A.1 presents all the parameters adopted to the quadrotors model used in the simulation.

Table A.1: Quadrotors Parameters

Parameter	**Value**	**Unit**
I_x	0.00065	$kg.m^2$
I_y	0.00065	$kg.m^2$
I_z	0.0014	$kg.m^2$
l	0.125	m
k_T	0.001	$kg.m^2$
k_D	0.00002	$kg.m^2$
m	0.26	kg

Appendix

B Network topology

B.1 Network topology

B.1.1 Graph theory

To represent the network topology of the UAV swarming the graph theory is used. In an undirected graph the communication points in the network are denoted as nodes, and edges are communication links between points. For UAV swarms, the nodes represent UAVs, and the edges correspond to inter-UAVs links, such as wireless communications or relative sensors. It is assumed that all communication links are bidirectional.

Abstractly, an undirected graph $G = (\nu, \varepsilon)$ is defined by a node set ν with cardinality n, the number of nodes in the graph, and an edge set E comprised of pairs of nodes, where nodes ν_i and ν_j are adjacent if $\{\nu_i, \nu_j\} \in \varepsilon \subseteq [\nu]^2$, the set, of two-element subsets of ν. One special family of undirected graphs are tree graphs, denoted by the set Γ, where all two-node pairs are connected by exactly one simple path, that is, a connected graph without cycles. As panning tree of a graph is a tree sub graph that connects all vertices in the graph.

The neighborhood set $N(v_i)$ is composed of the set of nodes adjacent to ν_i. The scalar $d = (\nu_i, \nu_j)$ is the minimum path length, induced by the graph G, between nodes ν_i and ν_j. The adjacency matrix of G is denoted by $G^A = [\omega_{ij}^a] \in R^{n \times n}$, where ω_{ij}^a represents the entry of the i-th row j-th column of matrix G^A with $\omega_{ij}^a = 1$ if $(i, j) \in E$ and $\omega_{ij}^a = 0$ otherwise. The degree δ_i of node ν_i is the number of its

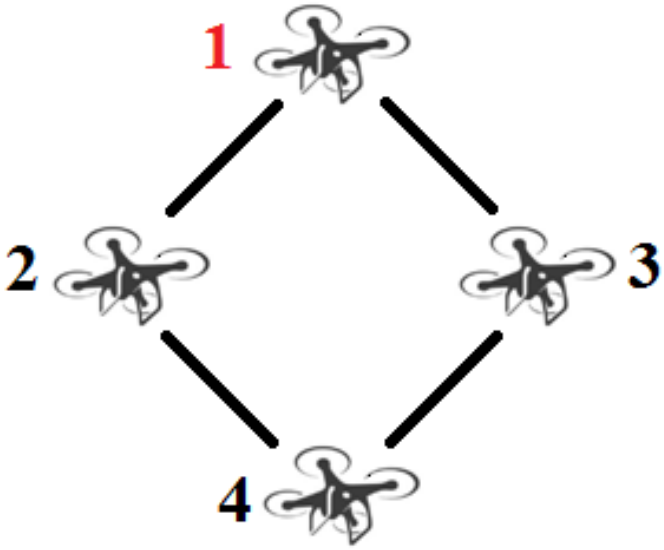

Figure B.1: Graph presentation

adjacent nodes. The degree matrix G^D is a diagonal matrix with δ_i at entry (i, i). We also define the diagonal matrix $G^L = diag\{\omega_1^l, \ldots.., \omega_n^l\}$ representing the status of agent: if $\omega_i^l = 1$ then agent i is a leader. Otherwise if $\omega_i^l = 0$ then agent i is a follower.

The graph Laplacian matrix is defined as $L = G^D - G^A \in R^{n\times n}$. It plays an important role in the dynamics of the network. An important feature of this matrix is that it is a (symmetric) positive semi definite matrix. The spectrum is ordered as $0 = \lambda_1(L) \leq \lambda_2(L) \leq \cdots \leq \lambda_n(L)$. The spectrum of the graph Laplacian matrix displays many features of the network. For example, the number of zero eigenvalues is equal to the number of connected components. Other features that couple the network topology and dynamics are explored in the following subsection.

The interaction matrix G for L-F formation is defined as follow:

$$G = G^D - G^A + G^L = L + G^L \tag{B.1}$$

Figure B.1 shows an example graph with its corresponding graph matrices. An undirected graph $G = (\nu, \varepsilon)$ is defined, with undirected graph $\nu = \{\nu_1, \nu_2, \nu_3, \nu_4\}$ and $E = \{\{\nu_1, \nu_2\}, \{\nu_1, \nu_3\}, \{\nu_2, \nu_4\}, \{\nu_3, \nu_4\}\}$. The adjacent matrix G^A, degree matrix G^D, diagonal matrix ζ^L and Laplacian matrix L are defined as follow:

$$G^A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} G^D = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix} G^L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$
$$L = \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 2 & 0 & -1 \\ -1 & 0 & 2 & -1 \\ 0 & -1 & -1 & 2 \end{bmatrix} \tag{B.2}$$

The interaction matrix G for L-F formation is defined as follow:

$$G = \begin{bmatrix} 3 & -1 & -1 & 0 \\ -1 & 2 & 0 & -1 \\ -1 & 0 & 2 & -1 \\ 0 & -1 & -1 & 2 \end{bmatrix} \tag{B.3}$$

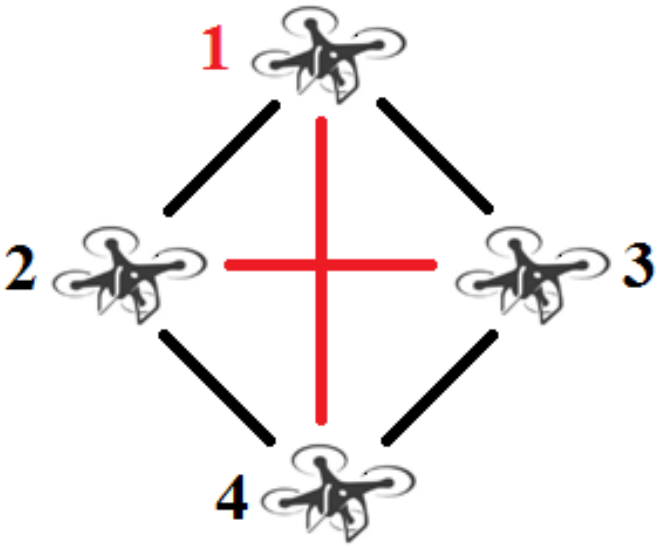

Figure B.2: Extra Edge

B.1.2 Extra edges Dynamics

We assume that two extra edges. As shown in Figure B.2 The added two edges and the same vertices constitute the graph $G_e = (\nu, (\epsilon)_e)$. Then, G_e and $\widetilde{G}$ yield:

$$\widetilde{G} = G + G_e \tag{B.4}$$

Bibliography

[1] A., G. Farinelli, L. Grisetti, S. Iocchi, D. Lo Cascio, and Nardi. Design and evaluation of multi agent systems for rescue operations. In *Conference on Intelligent Robots and Systems*. IEEE, 2003. (Cited on pages 44 and 98.)

[2] Pentland Alex Bruckstein Alfred M. Altshuler, Yaniv. *Swarms and Network Intelligence in Search*. Springer International Publishing, 2018. (Cited on page 116.)

[3] Kehlenbeck Andrew. Optimal quaternion-based control for aggressive trajectory tracking with a micro-quadrotor uav, 2014. (Cited on page 44.)

[4] Salvatore Aronica, Francesco Benvegna, Massimo Cossentino, Salvatore Gaglio, Alessio Langiu, Carmelo Lodato, Salvatore Lopes, Umberto Maniscalco, and Pierluca Sangiorgi. An agent-based system for maritime search and rescue operations. In *WOA*, 2010. (Cited on page 98.)

[5] Zoltán Beck, W. T. Luke Teacy, Alex Rogers, and Nicholas R. Jennings. Online planning for collaborative search and rescue by heterogeneous robot teams. In *AAMAS*, 2016. (Cited on page 116.)

[6] Jean Berger and Nassirou Lo. An innovative multi-agent search-and-rescue path planning approach. *Computers OR*, 53:24–31, 2015. (Cited on page 98.)

[7] Jean Berger, Nassirou Lo, and Martin Noel. A new multi-target , multi-agent search-and-rescue path planning approach. 2014. (Cited on page 116.)

[8] Bouabdallah, Noth S., Siegwart A., and R. Pid vs lq control techniques applied to an indoor micro quadrotor. In *Intelligent Robots and Systems,(IROS).*, pages 2451–2456. IEEE, 2004. (Cited on pages 28 and 44.)

[9] Matthias R. Brust and Bogdan M. Strimbu. A networked swarm model for uav deployment in the assessment of forest environments. *2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, pages 1–6, 2015. (Cited on page 69.)

[10] Changyun and WEI. *Cognitive Coordination for Cooperative Multi-Robot Teamwork*. PhD thesis, Dutch Research School for Information and Knowledge Systems, 2016. (Cited on page 116.)

[11] KY Chee and ZW Zhong. Control, navigation and collision avoidance for an unmanned aerial vehicle. *Sensors and Actuators A: Physical*, 190:66–76, 2013. (Cited on page 69.)

[12] Yu Fan Chen, Mark Cutler, and Jonathan P. How. Decoupled multiagent path planning via incremental sequential convex programming. *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5954–5961, 2015. (Cited on page 116.)

[13] K CHOUTRI, M LAGHA, L DALA, and M LIPATOV. Quadrotors uavs swarming control under leader-followers formation. In *2018 22nd International Conference on System Theory, Control and Computing (ICSTCC)*, pages 794–799. IEEE, 2018. (Cited on page 69.)

[14] Kheireddine Choutri, Mohand Lagha, and Laurent Dala. Distributed obstacles avoidance for uavs formation using consensus-based switching topology. *International Journal of Computing and Digital Systems*, 8, 2019. (Cited on page 80.)

[15] Kheireddine Choutri, Mohand Lagha, Laurent Dala, and Michael Lipatov. Quadrotors trajectory tracking using a differential flatness-quaternion based approach. In *Modeling, Simulation, and Applied Optimization (ICMSAO)*, pages 1–5. IEEE, 2017. (Cited on pages 28 and 44.)

[16] Anezka Chovancová, Tomas Fico, Peter Hubinskỳ, and F Duchoň. Comparison of various quaternion-based control methods applied to quadrotor with disturbance observer and position estimator. *Robotics and Autonomous Systems*, 79:87–98, 2016. (Cited on page 28.)

[17] D. J. Cooke. Optimal trajectory planning and lqr control for a quadrotor uav. 2006. (Cited on page 44.)

[18] Anh Duc Dang, Hung Manh La, Thang Nguyen, and Joachim Horn. Distributed formation control for autonomous robots in dynamic environments. *CoRR*, abs/1705.02017, 2017. (Cited on pages 69 and 70.)

[19] D De Silva. Formation control for unmanned aerial vehicles, 2012. (Cited on page 69.)

[20] Ahmad Din, Meh Jabeen, Kashif Zia, Abbas Khalid, and Dinesh Kumar Saini. Behavior-based swarm robotic search and rescue using fuzzy controller. *Computers & Electrical Engineering*, 70:53–65, 2018. (Cited on page 98.)

[21] Maria Drakaki, Hacer Güner Gören, and Panagiotis Tzionas. An intelligent multi-agent based decision support system for refugee settlement siting. *International journal of disaster risk reduction*, 31:576–588, 2018. (Cited on page 98.)

[22] Haibin Duan, Qinan Luo, Yuhui Shi, and Guanjun Ma. ?hybrid particle swarm optimization and genetic algorithm for multi-uav formation reconfiguration. *IEEE Computational Intelligence Magazine*, 8:16–27, 2013. (Cited on page 116.)

[23] Ehsan and Ebrahimi-Oskoei. Swarm of uavs: Search rescue operation in chaotic ship wakes, 2014. (Cited on page 98.)

[24] Murphy R.R. et al. *Search and Rescue Robotics*. Springer Handbook of Robotics. Springer, Berlin, Heidelberg, 2008. (Cited on page 116.)

[25] Antonio Franchi and Paolo Robuffo Giordano. Online leader selection for improved collective tracking and formation maintenance. *IEEE transactions on control of network systems*, 5(1):3–13, 2018. (Cited on page 69.)

[26] Chen Gao, Ziyang Zhen, and Huajun Gong. A self-organized search and attack algorithm for multiple unmanned aerial vehicles. *Aerospace Science and Technology*, 54:229–240, 2016. (Cited on page 98.)

[27] Khaled A Ghamry and Youmin Zhang. Formation control of multiple quadrotors based on leader-follower method. In *Unmanned Aircraft Systems (ICUAS)*, pages 1037–1042. IEEE, 2015. (Cited on page 69.)

[28] Francois Jan Yvon Gourhant Guillaume Remy, Sidi-Mohammed Senouci. Sar.drones: Drones for advanced search and rescue missions. Technical report, DRIVE Laboratory, University of Bourgogne, France, 2011. (Cited on page 98.)

[29] H., X.F. Liu, Y.S. Wang, and Zhong. Quaternion-based robust attitude control for un-certain robotic quadrotors. *IEEE Trans. Ind. Inform.*, 11(2):402–415, 2015. (Cited on page 44.)

[30] Liang Han, Xiwang Dong, Qingdong Li, and Zhang Ren. Formation tracking control for second-order multi-agent systems with time-varying delays. *2016 35th Chinese Control Conference (CCC)*, pages 7902–7907, 2016. (Cited on page 69.)

[31] Zhicheng Hou. *Modeling and formation controller design for multi-quadrotor systems with leader-follower configuration.* PhD thesis, Compiègne, 2016. (Cited on page 69.)

[32] Zhicheng Hou and Isabelle Fantoni. Leader-follower formation saturated control for multiple quadrotors with switching topology. *2015 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*, pages 8–14, 2015. (Cited on page 69.)

[33] Xing Huo, Mingyi Huo, and Hamid Reza Karimi. Attitude stabilization control of a quadrotor uav by using backstepping approach. *Mathematical Problems in Engineering*, 2014, 2014. (Cited on pages 28 and 44.)

[34] Jürgen, Saeed Scherer, Samira Yahyanejad, and Hayat. An autonomous multi-uav system for search and rescue. In *First Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*, pages 33–38. ACM, 2015. (Cited on page 116.)

[35] Kaddouri, Mokhtari Djamel, Abdelaziz Abdellah, and Benallegue. Attitude optimal backstepping controller based quaternion for a uav. *Mathematical Problems in Engineering*, 2016, 2016. (Cited on page 44.)

[36] Yunus Karaca, Mustafa Cicek, Ozgur Tatli, Aynur Sahin, Sinan Pasli, Muhammed Fatih Beser, and Suleyman Turedi. The potential use of unmanned aircraft systems (drones) in mountain search and rescue operations. *The American journal of emergency medicine*, 36 4:583–588, 2018. (Cited on page 98.)

[37] L., Z.Y. Sun, and Zuo. Nonlinear adaptive trajectory tracking control for a quad-rotor with parametric uncertainty. *J. Aerosp. Eng.*, 229(9):1709–1721, 2015. (Cited on page 44.)

[38] Mingrui Lao and Jun Tang. Cooperative multi-uav collision avoidance based on distributed dynamic optimization and causal analysis. *Applied Sciences*, 7(1):83, 2017. (Cited on page 69.)

[39] Larbi, Meguenni M. A., Meddahi K. Z., Litim Y., and M. Nonlinear observer and backstepping control of quadrotor unmanned aerial vehicle. *International Review of Aerospace Engineering*, 6(5):233–242, 2013. (Cited on page 44.)

[40] Keun Uk Lee, Yoon Ho Choi, and Jin Bae Park. Backstepping based formation control of quadrotors with the state transformation technique. *Applied Sciences*, 7(11):1170, 2017. (Cited on page 69.)

[41] ZX Liu, X Yu, C Yuan, and YM Zhang. Leader-follower formation control of unmanned aerial vehicles with fault tolerant and collision avoidance capabilities. In *Unmanned Aircraft Systems (ICUAS)*, pages 1025–1030. IEEE, 2015. (Cited on page 28.)

[42] Eleftherios Lygouras, Antonios Gasteratos, Konstantinos Tarchanidis, and Anastasios Mitropoulos. Rolfer: A fully autonomous aerial rescue support system. *Microprocessors and Microsystems*, 2018. (Cited on page 98.)

[43] M., M. Klusch, M. Thimm, and Paprzycki. A multi-agent flooding algorithm for search and rescue operations in unknown terrain. In *Conference: German Conference on Multiagent System Technologies*. Springer-Verlag Berlin Heidelberg, 2013. (Cited on page 116.)

[44] M, E Pena, C Vivas, and Rodriguez. Simulation of the quadrotor controlled with lqr with integral effect. *ABCM Symposium Series in Mechatronics*, 5(5):390–399, 2012. (Cited on page 44.)

[45] Mario, Silvagni, Andrea Tonoli, Marcello Enrico Zenerino, and Chiaberge. Multipurpose uav for search and rescue operations in mountain avalanche events. *Natural Hazards and Risk*, 2017. (Cited on page 98.)

[46] Marzena, Szymon Półkaa, Łukasz Ptak, and Kuziora. The use of uav's for search and rescue operations. In *International scientific conference on sustainable, modern and safe transport*, 2017. (Cited on page 98.)

[47] Kamel Mina, Alexis Kostas, Achtelik Markus, and Siegwart Roland. Fast nonlinear model predictive control for multicopter attitude tracking on so(3). In *IEEE Conference on Control Applications (CCA) Part of 2015 IEEE Multi-Conference on Systems and Control*, pages 21–23. IEEE, 2015. (Cited on pages 28 and 44.)

[48] Mohd, Ariffanan, and Mohd Basri. Design and optimization of backstepping controller for an underactuated autonomous quadrotor unmanned aerial vehicle. *TRANSACTIONS OF FAMENA*, XXXVIII(3), 2014. (Cited on page 44.)

[49] Robin R Murphy. Trial by fire [rescue robots]. *IEEE Robotics & Automation Magazine*, 11(3):50–61, 2004. (Cited on page 98.)

[50] Ram G. Lakshmi Narayanan and Oliver C. Ibe. A joint network for disaster recovery and search and rescue operations. *Computer Networks*, 56:3347–3373, 2012. (Cited on page 98.)

[51] Nuno and Miguel. Distributed two-layer algorithms for collaborative multi-uav multi-target tracking, 2013. (Cited on page 116.)

[52] P., P. Doherty, and Rudol. A uav search and rescue scenario with human body detection and geolocalization. In *Conference on Artificial Intelligence*. AI07, 2007. (Cited on page 98.)

[53] P., I. Molina, T. Colomina, and Vitoria. Searching lost people with uavs: The system and results of the close-search project. *Remote Sensing and Spatial Information Sciences*, 2012. (Cited on page 98.)

[54] S. P., L. M. Yeong, S. S. King, and Dol. A review on marine search and rescue operations using unmanned aerial vehicles. *International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering*, 2015. (Cited on page 98.)

[55] H. Van Dyke Parunak, Sven A. Brueckner, J. J. Odell, and Altarum. Swarming coordination of multiple uav ' s for collaborative sensing. 2003. (Cited on page 116.)

[56] Hung Pham, Scott A Smolka, Scott D Stoller, Dung Phan, and Junxing Yang. A survey on unmanned aerial vehicle collision avoidance systems. *arXiv preprint arXiv:1508.07723*, 2015. (Cited on page 69.)

[57] Boualem Rabta, Christian Wankmüller, and Gerald Reiner. A drone fleet model for last-mile distribution in disaster relief operations. *International Journal of Disaster Risk Reduction*, 28:107–112, 2018. (Cited on page 98.)

[58] Rethnaraj Rambabu, Muhammad Rijaluddin Bahiki, and S. M. Ali. Relative position-based collision avoidance system for swarming uavs using multi-sensor fusion. 2015. (Cited on page 70.)

[59] Sarvapali D. Ramchurn, Joel E. Fischer, Yuki Ikuno, Feng Wu, Jack Flann, and Antony Waldock. A study of human-agent collaboration for multi-uav task allocation in dynamic environments. In *IJCAI*, 2015. (Cited on page 116.)

[60] S.H., Y. Wang, and Yang. Quadrotor aircraft attitude estimation and control based on kalman filter. *Control Theory Appl*, 30(9):1110–1115, 2013. (Cited on page 44.)

[61] Iman Shames, André Teixeira, Henrik Sandberg, and Karl Henrik Johansson. Distributed leader selection without direct inter-agent communication. In *2nd IFAC Workshop on Distributed Estimation and Control in Networked Systems, NecSys' 10. Annecy. 13 September 2010-14 September 2010*, pages 221–226, 2010. (Cited on page 69.)

[62] RK Sharma and D Ghose. Collision avoidance between uav clusters using swarm intelligence techniques. *International Journal of Systems Science*, 40(5):521–538, 2009. (Cited on page 69.)

[63] Tal Shima and Steven Rasmussen. Uav cooperative decision and control : Challenges and practical approaches. 2009. (Cited on page 116.)

[64] Ivan Stojkovi. Formation control of robotized aerial vehicles based on consensus-based algorithms. 2017. (Cited on page 69.)

[65] Işık Y. & Korul H. Tosun, D. C. Lqr control of a quadrotor helicopter. Technical report, Faculty of Aeronautics and Astronautics, Anadolu University 26470 Eskisehir, Turkey, 2014. (Cited on page 44.)

[66] Roberto Vettor and Carlos Guedes Soares. Computational system for planning search and rescue operations at sea. In *ICCS*, 2015. (Cited on page 98.)

[67] Sonia Waharte and Agathoniki Trigoni. Supporting search and rescue operations with uavs. *2010 International Conference on Emerging Security Technologies*, pages 142–147, 2010. (Cited on page 98.)

[68] Rui Wang and Jinkun Liu. Adaptive formation control of quadrotor unmanned aerial vehicles with bounded control thrust. *Chinese Journal of Aeronautics*, 30(2):807–817, 2017. (Cited on page 69.)

[69] Wei and Zheng. Trajectory tracking for an autonomous airship using fuzzy adaptive sliding mode control. *Journal of Zhejiang University-Science*, 13(7):534–543, 2012. (Cited on page 44.)

[70] Huafeng Wu, Xiaojun Mei, Xinqiang Chen, Junjun Li, Jun Wang, and Prasant Mohapatra. A novel cooperative localization algorithm using enhanced particle filter technique in maritime search and rescue wireless sensor network. *ISA transactions*, 78:39–46, 2018. (Cited on page 98.)

[71] Y., H. Bouzid, Y. Siguerdidjane, .M Bestaoui, and Zareb. Energy based 3d autopilot for vtol uav under guidance navigation constraints. *Journal of Intelligent and Robotic Systems*, 87(2), 2016. (Cited on page 44.)

[72] Y. and Yan. Attitude regulation for unmanned quadrotors using adaptive fuzzy gain-scheduling sliding mode control. *Aerospace Science and Technology*, 54:208–217, 2016. (Cited on page 44.)

[73] Ye and Yan. Neural network approximation-based nonsingular terminal sliding mode control for trajectory tracking of robotic airships. *Aerospace Science and Technology*, 54:192–197, 2016. (Cited on page 44.)

[74] Jian Zhang, HU Qinglei, WANG Danwei, and XIE Wenbo. Robust attitude coordinated control for spacecraft formation with communication delays. *Chinese Journal of Aeronautics*, 30(3):1071–1085, 2017. (Cited on page 69.)

[75] Li Zhang, and X., Wang X., Lu K., and Y. A survey of modelling and identification of quadrotor robot. *Abstract and Applied Analysis*, 2014, 2014. (Cited on page 28.)

[76] Minghuan Zhang. Formation flight and collision avoidance for multiple uavs based on modified tentacle algorithm in unstructured environments. *PloS one*, 12(8):e0182006, 2017. (Cited on page 69.)

[77] Zhong Zheng and Shenmin Song. Autonomous attitude coordinated control for spacecraft formation with input constraint, model uncertainties, and external disturbances. *Chinese Journal of Aeronautics*, 27(3):602–612, 2014. (Cited on page 69.)