MINISTRY OF EDUCATION
TECHNICAL UNIVERSITY
OF CLUJ-NAPOCA, ROMANIA

Field of System Engineering

PhD THESIS
– ABSTRACT –

Development and Testing of a Multi-agent Search and Localization System Using Drones

PhD Student:
Daniel-Dumitru TIMIȘ

PhD Supervisor:
Prof. Eng. Eva-H. DULF, PhD

Examination committee:

Chair: Prof. Eng. **Vlad MUREȘAN,** PhD - Technical University of Cluj-Napoca;
PhD Supervisor: Prof. Eng. **Eva H. DULF,** PhD - Technical University of Cluj-Napoca;
Members:
- Prof. Eng. **Radu-E. PRECUP**, PhD - Polytechnic University of Timișoara;
- Prof. Eng. **Monica LEBA**, PhD – University of Petroșani;
- Prof. Eng. **Liviu-C. MICLEA**, PhD - Technical University of Cluj-Napoca.

– Cluj-Napoca –
2024

CONTENTS

1. INTRODUCTION

Robots are expected to become indispensable in today's rapidly evolving society. In recent years, the increase in hardware capabilities and performance, the development of complex software, and the need for these entities in everyday applications have led to a rapid evolution of autonomous robots. Whether we mean software robots or physical robots, they are one of the main factors accelerating the evolution of human society.

The current state of technology has allowed researchers to design and conduct experiments on different types of unmanned vehicles. One of these are Unmanned Aerial Vehicles (UAVs) or commonly known as drones. With great flexibility, easy deployment, excellent maneuverability and with many commercial solutions available, they are used in a wide range of applications. These applications include disaster management, search and rescue, construction and civil industry, traffic monitoring, agriculture, and many others. In addition, the potential for expanding the use of drones into other areas is possible due to their flexibility and market opportunities.

The challenge for this study was to design and develop a system of drones capable of working together to identify and locate people in critical situations. Every year around the world numerous persons are reported missing due to natural or man-made disasters. Such a system, capable of being deployed anywhere and in any conditions to manage the disaster, can save many lives and reduce the risk of injury to the rescuers. Basically, drones are assigned a search mission in a geographically well-determined area and, using the sensors with which they are equipped, can locate people or groups of people. They can also be used to carry a range of first-aid necessities, and based on the environmental information gathered, they can help planning the rescue mission, that minimizes any risk to the rescue teams.

This thesis aims in the first phase to study and identify how such a system can be built, what are the solutions, problems, and current directions of development in this field. Based on the collected information, it is proposed to develop and test such a search and localization system in a controlled environment.

The first part of this thesis (Chapters 1 and 2) conducts a detailed literature survey in which more than 100 research articles in the field of drones are evaluated. Current solutions and methods used to integrate UAVs in various applications such as search and rescue, construction monitoring, agriculture, traffic monitoring, transportation and delivery are presented. The methods and results that the literature proposes as solutions for these integrations and future trends in this direction were evaluated. In addition to the study of applications, multi-agent drone systems were also evaluated (Chapter 3), an effort that focused on understanding how such a system is designed and what are its most important components. Starting from the results of the above analysis, the development principles for a solution that satisfies the purpose of this thesis were formulated.

The second part of the paper is divided into 4 case studies applied on the theme of drones, in order to evaluate their performance and to identify the best way to integrate them in a real application.

The first study (Chapter 7) focused on flight stability control structures of a quadcopter. Several modern techniques such as fractional PID controllers and adaptive control structures were evaluated. **The second study (Chapter 8)** focused on the development and testing of the distributed control system infrastructure, to which a few drones need to connect to collaborate. The software applications needed to accomplish this objective were defined and designed, and the way in which a drone can be controlled using this structure was evaluated.

The third study (chapter 9) shows how using image processing, objects of certain shapes and colors can be identified. With the help of this identification drones can fly in a predefined search area in order to locate the target. **The last study (Chapter 10)** focuses on the design of a distributed drone system, which using Particle Swarm Optimization (PSO) algorithm. The algorithm will compute the future position of all drones with the aim of identifying the previously defined target object. The study analyzes the simulated and experimental results of this system.

Chapters 11 and 12 present a series of general discussions of the results obtained the conclusions of the thesis and the personal contribution to this field.

2. STATE OF THE ART

Throughout history, mankind has faced a series of natural and man-made disasters. In such circumstances, a rapid and well-coordinated response is needed to save as many lives as possible after such an event strikes a populated area. This is where the concept of Search and Rescue (SAR) was born and began to evolve. SAR is a concept in which an organized team or organized teams of searchers or rescuers are dispatched to an area where a dangerous event has occurred and where one or more people have been put at risk because of it. These teams attempt to identify and locate the victims and use their efforts, experience, and knowledge to bring them back to safety.

At the very beginning of history, search and rescue operations were exclusively dedicated to maritime expeditions. The concept is first encountered in history, during antiquity in ancient Greece or ancient Rome, where coastal communities were often involved in actions to rescue shipwrecked sailors near the shore. The first documented example of SAR action took place in 1656, when a Dutch merchant ship 'Vergulde Draeck' was shipwrecked off the Australian coast. Three separate rescue teams ventured out to save the surviving sailors, but without success [1]. The first organization in history with the primary purpose of saving lives was founded by Sir William Hillary in 1824 as the Royal National Lifeboat Institution (RNLI). Sir Hillary, who lived on the Isle of Man, saved many lives from shipwrecks and his efforts laid the foundation for the first organization dedicated to the rescue of shipwrecked sailors [2]. From the 20th century, and more specifically during World War II, with the development of aviation and the increasing frequency of air accidents in remote areas, the need for specialized groups to undertake rescue actions increased. On this basis, things evolved very rapidly and by the end of the 20th century there were already dedicated teams of professionals around the world whose main area of activity was search and rescue. Emerging areas such as flood rescue, fire rescue, mountain rescue or other rescue actions following natural or man-made disasters developed. For example, in Romania, the mountain rescue organization (Salvamont) was established in the 1960s through a series of government decrees and voluntary initiatives. It represented at that time a response to the growing number of people engaging in outdoor activities such as hiking, skiing, and mountaineering, especially in the Carpathian Mountains [3]. To support the development effort of this sector and to standardize it globally, the International Search and Rescue Advisory Group (INSARAG) was established in 1991 within the United Nations in 1991. The purpose of this organization is to standardize cross-border SAR operations, ensuring a coordinated response to large-scale disasters.

Search and rescue has evolved from a volunteer community effort to a sophisticated, technology-driven field essential for disaster management. Modern SAR operations are characterized by advanced technologies such as GPS, drones, and artificial intelligence, and are supported by structured organizations and international cooperation frameworks. As the frequency and complexity of disasters continue to increase, the role of SAR will continue to be enhanced to improve the ability to save lives.

The component of drone integration in this field, is an intensely studied one, which is enjoying real and applied solutions to SAR needs. In this regard, this paper had as a first objective, to conduct a detailed survey of the current state of the literature, to understand how drones are being used and integrated, not only in the SAR field, but also in other industries. The study of the literature was based on 3 directions: classification of UAVs, applications of drones in different industries of interest and a more current concept that of multi-agent UAV systems or "Swarm of drones".

2.1. Unmanned Aerial Vehicles – UAV

In order to give a complete overview of the subject of UAVs and before detailing applications and functionalities, a classification of flying objects needs to be made. The term UAV is too generic, it can take different forms and have different deployment solutions. This is possible mainly due to the high interest in recent years from both the private and scientific sectors.

Several classification criteria can be defined, such as scope, shape, size, range, aerodynamics, maneuverability, and structure. The most common UAV or drone is the quadcopter or multirotor UAVs in general, due to their maneuverability and ease of integration into different applications [4].

2.1.1. UAV Classes

The main criterion by which drones can be categorized refers to the altitude at which they can fly. Three classes of UAVs are defined [5] [6]:

- Low Altitude UAVs (LAU)
- High Altitude UAVs (HAU)
- Satellites

The first class refers to UAVs that have a limited range and operate at low altitude. The most common type is multi-rotor UAVs. This class should be easy and fast to operate over short and medium distances. The second class operates at higher altitudes and needs a pre-planned route to operate. They have a longer range, and often take the form of unmanned aircraft. The third class operate in space and are actually operational commercial or military satellites. One of their purposes is surveillance and communications coverage for LAU and HAU.

At the moment there is no system to integrate the three classes into a real application, but studies and research are being carried out.

2.1.2. Aerodynamics

It is worth mentioning the classification of UAVs according to their shape or aerodynamics. They are available in many shapes and variants. The main evaluation criteria are to classify them according to their landing or take-off capability. Thus, 2 types of flying vehicles can be defined: horizontal take-off and landing (HTOL) or vertical take-off and landing (VTOL) [7].

Figure 1 shows the classification structure of UAVs based on their aerodynamics. The HTOL category is mainly composed of fixed-wing aircraft or a classic airplane. Remote control systems can be installed on these types of aircraft to transform them into UAVs.

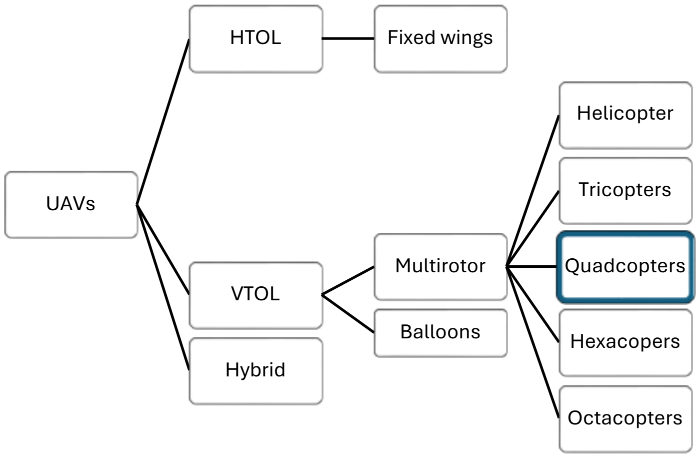

Figure 1: Classification of drones according to aerodynamics

The second category presented (VTOL) consists mainly of multi-rotor systems capable of vertical take-off. The most widespread UAV or multi-rotor drone nowadays is the quadcopter, due to its robustness and its ability to be easily operated. Balloons can also be added to this section. Both weather and surveillance balloons are autonomous and able to take off and land vertically. These types of UAVs can be used as terminal stations or communication terminals in certain applications, as they can stay in the air as long it is needed and can carry heavier equipment, compared to multi-rotors, with low power consumption. The third category is hybrid. This type of aircraft is capable of taking off vertically or horizontally, depending on the needs of the application.

2.1.3. Size and operating distance

Size and range are two important characteristics by which UAVs can be categorized. Both will define the type of application and its complexity. Some applications will require smaller UAVs with limited range, while other applications will require heavier UAVs with long range. The authors in [7] and [8] categorize UAVs according to their weight and size. The classification presented in these 2 articles is applied to the whole spectrum of UAVs.

Table 1 classifies quadcopters intended for the civil sector based on weight and range. This table can be used as a reference when building an application dedicated to a specific sector. Also, legislation must be confronted and respected to develop any type of UAV systems applied for a real need.

In Europe and the USA, the Nano category is the only one in the table below for which no special certification, registration or training is required to operate such an aircraft, according to the European Union Aviation Safety Agency (EASA) [9] and the Federal Aviation Administration (FAA) [10].

Category	Weight	Range	Type
Nano	< 250 g	5 km	Fix-wing, multirotor
Micro	< 2 kg	25 km	Fix-wing, multirotor
Light with small range	< 20 kg	40 km	Fix-wing, multirotor
Light with medium range	< 50 kg	100 km	Fix-wing, multirotor
Small or medium UAVs	< 150 kg	150 km	Fix-wing

Table 1: Classification of UAVs by mass and range

2.2. Single UAV applications

This section conducts a detailed literature review to identify in which industries and how UAVs are integrated. Numerous works assessing the applications in which UAVs can be integrated were analyzed to begin with. Based on this assessment, the most important areas that deserve a detailed analysis in order to understand how UAVs can be integrated and what are the results of this integration have been identified.

Over the past 15 years, researchers have focused on developing a stable drone capable of autonomous flight. In [11] the authors reviewed the most relevant research in the UAV field. The paper shows that 31% of the reviewed papers focused on hardware development for quadcopters or drones and 28% on control and modeling. Hardware and control strategy is the first necessity in building an application that integrates UAVs. These two topics represent the main pillars developed by the researchers to achieve a robust drone capable of successfully operating in various flight missions. This first need is highlighted by 59% of the main effort dedicated to this area. The remaining 41% of the work considered higher goals for UAVs, such as 17% - route planning, 12% - mapping and inspection, 7% - teleoperation and 5% collision avoidance.

The purpose of this section is to expand the research and focus only on the most relevant and important applications. **Search and rescue (or disaster management), infrastructure construction and inspection, precision agriculture, transportation and delivery of goods, real-time traffic monitoring and surveillance** are the applications where researchers' interest and creation of a real solution are higher. Therefore, this section will focus on the topics outlined above, by conducting a broad and detailed survey of papers where the main research focus is the integration of UAVs in one of the areas listed above.

This chapter presents the importance of drones in key sectors of today's economy and society. Based on this analysis, it can be concluded that these systems are extremely important, but solutions are still under development, which represents a real opportunity for further research. This section has been based on applications that require the use of a single drone, but the areas mentioned above can be improved by integrating a multi-agent drone system (or a swarm of drones), capable of interacting and working together to solve tasks and thus improving key sectors of the economy.

2.3. Intelligent UAV swarms

For this section, an extensive literature review on the topic of drone swarms was conducted. The research aims to understand how multiple UAVs can be effectively integrated into a collaborative structure (or swarm) to increase the performance and usability capabilities of a single agent in different applications. It was mentioned in the introduction that this study was preliminary research, for this PhD thesis, which aims to implement a collaborative system with multiple UAVs for a search and rescue application capable of identifying people in distress.

The first step was to analyze the most relevant and updated research works on the topic of multi-agent UAVs or UAV swarms. Following this reading, the present study aims to identify how such systems are structured, which are the most important components, and which are the most relevant aspects to be further analyzed.

Based on the analysis conducted as part of this chapter, a pattern can be observed in terms of the key aspects or components of a UAV swarm. All the articles mentioned the importance of communication, which plays a fundamental role and is a base layer for a robust and efficient system. Secondly, formation control and the importance for the swarm to maintain shape, avoid collisions and obstacles in the environment, and plan the flight path of the whole structure. Finally, the swarm control algorithm, which is the center of the swarm, is responsible for ensuring that the entire structure acts as a unified entity capable of achieving common goals.

This section reviews the state-of-the-art based on articles discussing one of the following key ideas: UAV swarm **communication**, drone **collaboration** (formation management, task management and trajectory planning) and **swarm control algorithms.**

2.4. Conclusion of the literature review

In the coming era of artificial intelligence and remotely operated applications, the use of UAVs is almost inevitable. The UAV market is continuously expanding and is expected to grow considerably over the next decade due to the growing need for remote work with drones and the continuous R&D being invested in the sector. By conducting an extensive analysis of the top five sectors or industries in which these systems can be integrated or applied, this paper has identified the status and current solutions for UAV applications. In all of the five industries that were analyzed: Search and Rescue, Construction, Agriculture, Transportation, and Traffic Monitoring, the use of drones is contributing to their improvement.

The most vital component or aspect that influences the quality of such an application is its ability to identify the surrounding environment using image processing or other sensors such as LiDAR and spectral or thermal cameras. For image processing there is a large set of algorithms such as TensorFlow, YoLo or Meshlab, combined with convolutional neural networks (CNN), have proven to be reliable tools for identifying the environment or other key shapes (people, buildings, infrastructure, or obstacles). Finding a way to integrate the processed image information with the data collected from the other sensitive sensors mentioned above is essential for the development of an autonomous UAV capable of navigating in a real environment. Current research continues to focus on achieving this milestone and substantial progress has been made in this direction.

Multi-agent UAV systems, in which a swarm of drones can be deployed to achieve a common goal, is growing. These solutions can be integrated into any of the industries mentioned above and enhance the capabilities of a single drone application. To realize such a system, it is important to provide three vital components: communication, flight formation control and swarm control. Communication between the swarm components is vital and should be as stable as possible to ensure the overall collaboration of the agents. In addition, communication with the base is also important, but not vital, as this system should operate autonomously. Today's 5G technology is a reliable and affordable tool to help improve this functionality. The formation control component is the coordination algorithm used to ensure that all agents maintain the desired swarm shape, avoiding collision between each agent with surrounding obstacles. Swarm control is responsible for ensuring that the common goal is achieved by implementing advanced algorithms such as Particle Swarm Optimization, Genetic Algorithms, Ant Colony Optimization, Model Predictive Control, and many others.

In conclusion, this study highlights the state-of-the-art of UAV applications in general, presenting the current technologies, algorithms, results and future trends of this research topic. The real solutions showed that, from a technical point of view, these systems are almost capable of being deployed in real applications. The time horizon for this to happen also depends on how and when legislation in this area will be regulated.

3. Working hypothesis /objectives

The aim of this research is to design a search and rescue system based on a swarm of 3 or more drones, capable of locating lost and/or critically distressed people. From the detailed bibliographical study carried out in the above chapters, it can be seen that more and more industries are studing the integration of UAVs, and many solutions have already been identified in this direction. At the same time the interest in designing a multi-agent system, consisting of several UAVs or intelligent swarm of UAVs is growing.

The main component of this work is the multi-agent system or drone swarm. In the literature survey it was identified that such a system must contain 3 main elements: communication, formation control and swarm-wide control. Starting from these 3 fundamental components the study proposes the following solution:

1. **Communication**: through a Wi-Fi local area network with a coverage limited to the area used for validating the results. This isolated network will ensure the elimination of delays.
2. **Formation control**: it does not require a proper formation but rather the design of a structure at each agent level to prevent collision with another agent by imposing a force "field" that does not allow 2 drones to be in the same position at the same moment in time. To do this, the drones will adjust their next move according to the current position of the other agents.
3. **Swarm control**: PSO algorithm is proposed to position each agent in the swarm. Based on the current position of each drone and the target location, the algorithm will calculate and transmit to each drone the next position so that the whole system will achieve its set goal. Additionally, the control component the whole swarm information (position, video image, cost function result) and stores it for analysis. This component is also the interface between the system and the operator. It can at any time request remote manual control of any drone in the swarm.

Based on the scope of this research and the specifications of the multi-agent system the following components and development directions are identified:

1. **UAV control**: the development of a control system for a single UAV, which will be replicated for all agents of the distributed system. Several control methods will be examined in order to identify the most robust control method. Based on the experimental results suitable hardware and software will be created or chosen.
2. **Swarm control server**: implementing a server to connect each agent in the system, using a local internet network. The server must be able to retrieve information (video and positioning) from each agent and store it. Through this component, the human operator can interact with the whole system or with a single element of the system by manual control.
3. **Video identification**: developing a method for video identification of a predefined target. Design and validation of image processing software for the identification of the defined template (a geometric figure of a given color).
4. **Search algorithm implementation**: search algorithm based on the Particle Swarm Optimization (PSO) algorithm to position each agent in the searching space. The cost function that the algorithm must minimize being the smallest distance of a drone from the template shape, identified through the image processing algorithm.

For each of the above components an independent experiment will be carried out, which implements a solution to the defined problem and validates it by experimental results.

4. General methodology

Each component will first be tested individually, and when the results are satisfactory, the whole control system will be validated on a real scenario.

1. **UAV control**: is validated using an experimental stand for static tests. If the static tests are satisfactory, the solution will also be validated in real flight in a controlled environment.
2. **Swarm control server**: the first validation is the communication between the server and a drone. If the solution guarantees automatic connection at application startup and a stable connection (without any communication loss) then the same test will be

repeated for 3 agents. The system is considered ready to fly when all 3 agents show a stable connection to the server. The server is only valid if the drones can be manually controlled by the human operator and the video image from each agent can be played back.

3. **Video identification**: solution initially validated in the laboratory by static tests. If the identification shows an error of less than 5% and a processing time of less than 0.1 seconds, the solution is considered stable and can be integrated into the system.
4. **Search algorithm implementation**: the solution will be proposed for validation when all of the above components fulfill the required conditions. In the first phase the algorithm will be validated manually, by statically positioning the drones at the position calculated by the algorithm. If this test passes, then the algorithm will be validated with the equipment in flight, but with reduced computation and positioning speed in order to allow possible errors to be identified and corrected by the human factor.

5. Single drone control

The classical PID and fractional PID structures were chosen for this evaluation because they showed the best simulated results. In addition, they were coupled with an adaptive algorithm, which aimed to increase the robustness of the system and to ensure the best possible rejection of perturbations and physical variations of the system.

The system dynamics are shown in Figure 2. From this it can be understood that the motion of the quadcopter is directly influenced by the control of the motor pairs. For example, for a forward motion, the speed of motors M11 and M12 must be decreased by the same value, and the speed of motors M21 and M22 must be increased by the same value. Thus, the drone will rotate around the Y-axis, which will cause the system to move in that direction. The same principle applies to the other motors. The more powerful the tuning structure, the speed of each motor is adjusted independently of the inter-connection between them, to reduce the effect of coupling between the inputs.

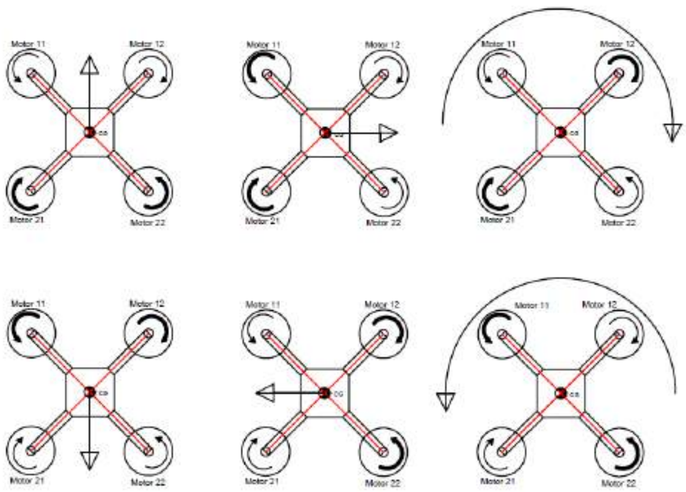

Figure 2: Quadcopter dynamics

Figure 3: The experimental prototype

Using the experimental stand presented in Figure 3 and the control program described above, 2 comparative studies of advanced control methods will be carried out in order to identify the best solution for flight control of the quadcopter prototype. Based on the static tests results, the initial control parameters for the drone flight will be determined.

Methods analyzed:

- Comparison between a classical PID with anti-saturation mechanism and a fractional regulator.
- Comparison between a fractional PID controller and an adaptive tuning method for adjusting the fractional controller parameters.

5.1. The first comparative study

The angular positions for X (roll) and Y (pitch) axes were evaluated to determine the stability of the quadcopter on the experimental prototype. For both classical and fractional PID controller, the PSO method was used.

The following conditions were imposed $\omega_{gc} = 20\ rad/sec$ and $\gamma_m = 100\ deg$. Running the algorithm to determine the parameters of the regulator resulted in the following values:

Controller type	K_p	K_i	K_d	λ	μ
PID	4,65	2,2	0,15	1	1
FO – PID	4,98	1,8	0,5	0,89	0,6

Table 2: Controllers parameters for the first study

For the PID controller the result was very close to 0, and for the FO-PID, the result returned by minimizing the cost function was 0.295, which represents the value of the 3rd evaluated equation, which failed to be fully satisfied, but the results are satisfactory.

The obtained controllers were implemented on the micro-controller presented above by applying discretized transfer functions. The fractional order regulator was discretized using the "Crone" approximation method [12], [13]. Both methods ran with a frequency of 250 Hz.

Figure 4 shows the time response of the physical model when a short mechanical impulse is applied to the Y-axis (pitch). The classical PID responds faster due to its derivative component. In less than 1 second, the angular velocity variation is counteracted. The only disadvantage is that the other axes are also disturbed. With the fractional controller, the adjustment time is short, about 1 second, and the main advantage is that interactions are eliminated.

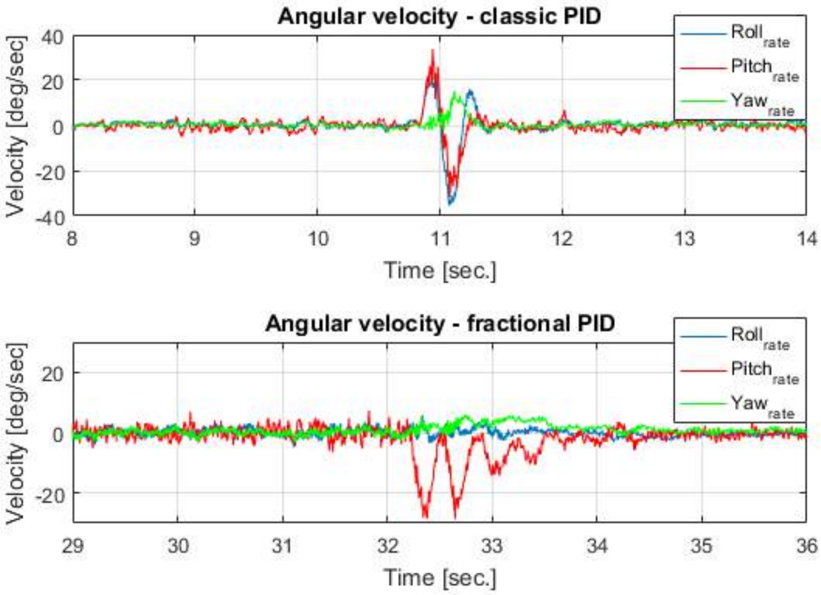

Figure 4: Disturbance rejection response

5.2. The second comparative study

Starting from the results obtained in the above study, it was concluded that a simple controller, even a fractional one, cannot respond adequately to unstable flight conditions. In these situations, the model evolutions due to the nominal rpm of the engines to reach a certain altitude and most importantly, due to unpredictable environmental conditions require the analysis of an adaptive structure. Such a structure is intended to continuously identify the current system model and adjust its tuning parameters according to these changes. Thus, environmental variations and disturbances can be rejected, leading to a stable system.

This study performed a comparison between a classical PID and a FO-PID, around which an adaptive structure was built. For this stage, a comparative study between a non-adaptive and an adaptive structure was carried out.

The model has been applied for the X and Y axis; it is described by the schematic from Figure 5. The "System FO Controller" block implements the controller model, either PID or FO-PID. It receives the tuning values from the "PSO Controller Designer" block, which implements the PSO algorithm to determine these parameters. This block receives from the "System Parameters Estimator" the physical parameters of the process by continuously identifying them based on the input and response of the controlled system. "Quadcopter Axis" represents a quadcopter axis.

As for the first analysis, constraints were imposed on the tuning structure. Thus, the required performance must respect cut-off frequency = 20 rad/sec, phase edge φm = 100° and it is required to be robust, i.e. the phase variation (or its derivative) must be zero. The design specifications were chosen to increase the overall stability of the system (large phase edge) and to ensure a fast-settling time (large gain edge).

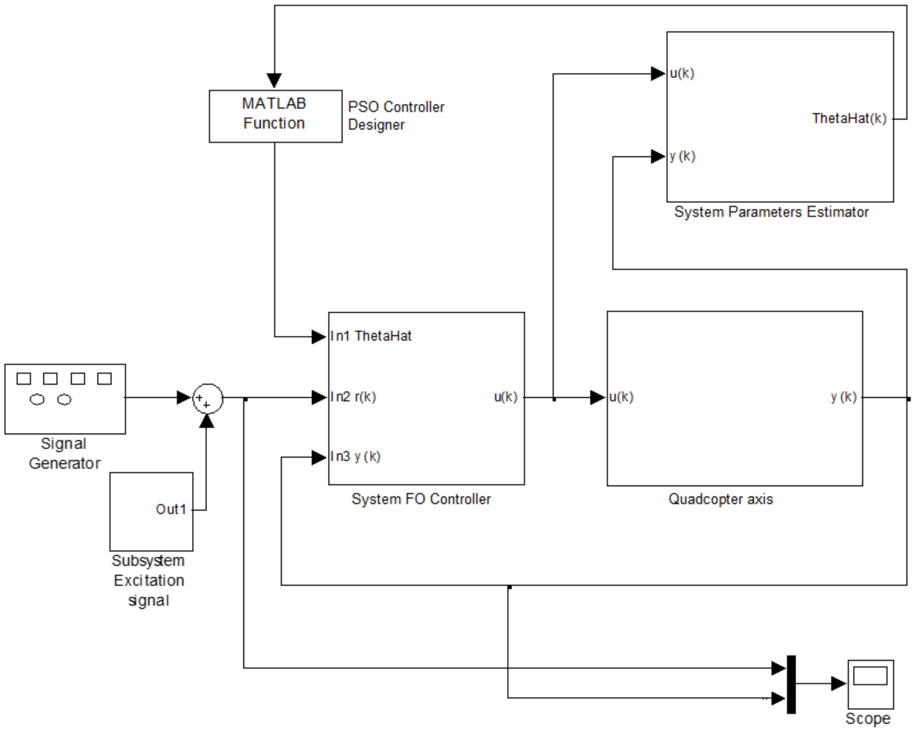

Figure 5: Adaptive control strategy proposed for quadcopter axis control

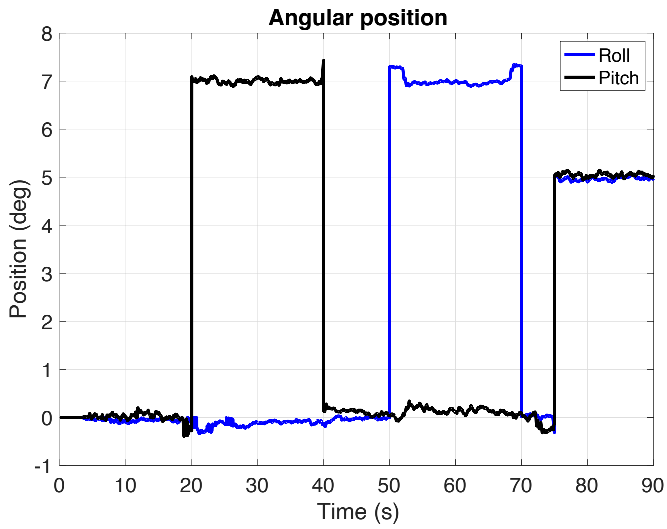

Figure 6: Closed-loop adaptive structure feedback for FO-PID

The obtained experimental results applied to the physical model, with integer-order and fractional-order adaptive PID controllers, are shown in Figure 6. These results aim to demonstrate how the system responds to a given reference applied on the X(pitch), Y(roll) axis and how they affect each other. The reference points in both cases were 7 degrees. The results clearly showed that both control strategies are capable of tracking the reference, although the integer-order controllers exhibit some large oscillations due to the interactions of the system axes.

Controller type	K_p	K_i	K_d	λ	μ
PID	4,65	2,2	0,15	1	1
FO – PID	5,0501	0,2758	0,0097	0,6747	0,5085

Table 3: Initial tuning parameters for adaptive PID and FO-PID controllers

5.3. Conclusions

On the experimental prototype all the 4 methods satisfied the required conditions, and the PSO algorithm, demonstrated that it can very simply and quickly find a set of good values for the required specifications.

The adaptive algorithm applied on the prototype proved to be the best performing, and that the FO-PID controller provided a much better response in terms of control effort, than the classical controller.

Both on the prototype and especially for real flight experiments, the adaptive control component has proven its importance. In particular, this need was observed by the introduction of a manual throttle adjustment function, which emphasizes this need for adaptive control. Here we can conclude that for such a process it is necessary to implement an adaptive control structure in order to ensure minimum flight performance and to identify process variations, such as meteorological ones.

In order to integrate this prototype into a wider application, such as Search and Rescue or aerial monitoring, altitude and geographic position control is needed to maintain a fixed point of flight in 3D space. So additional equipment has to be introduced for that, such as barometer for altitude and GPS for geographical positioning. In addition to these hardware components, new control loops to ensure correct positioning need to be introduced into the control algorithm.

In conclusion, for this study, using the Arduino UNO to implement a simple flight control of a quadcopter can be easily realized, it is just that this control model requires much more effort from the operator and even learning how the model can be controlled. To extend the scope of the application and to improve the control performance it is necessary to implement the control algorithm on other hardware, which is able to implement adaptive control algorithms (at least elementary), and which is able to provide a positioning control.

6. Drone remote control

The final objective of this thesis is to build a multi-agent application, in which each agent is able to identify, using a video camera, a specific target object. In order to achieve this, when choosing the hardware, we have to take into account that it is able to perform image processing operations, can control the drone's flight and can communicate with the monitoring server.

Once the control solution for the drone is in place, it is necessary to create the actual monitoring application, which is a distributed system consisting of a drone, a server and a graphical user interface for operation. The operating application and the server can run on the same component, but as stand-alone applications.

On the basis of the results obtained in the previous chapter and the above, this study has defined the following objectives:

1. Identifying a hardware and software solution.
2. Identify a hardware solution capable of capturing images and processing them. This component must also be able to connect to a server connected to the Internet via a local area network.
3. Flight performance evaluation of the chosen control solution.
4. Developing a server and a communication protocol, capable to exchange data bidirectionally with the drone.
5. Developing a simple monitoring application to interface the human operator with the drone. The application should be able to read and display flight data (GPS, altitude and angular position) and send simple flight commands (moving in space using buttons for the 6 directions of movement)

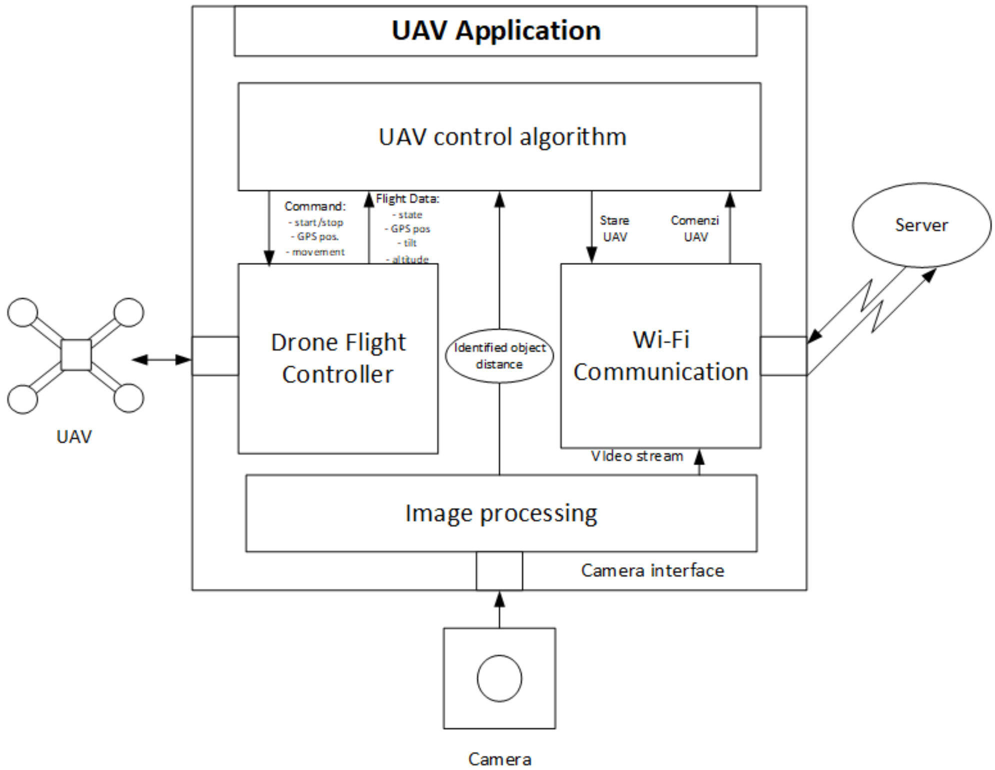

Figure 7: Integrated UAV control application diagram

Figure 7 shows the UAV control application diagram. It is not mandatory, that it consists of a single hardware component, it can consist of several dedicated components, but only one of them will realize the integral control and communicate with the server. At the top of the proposed structure is the main control program of the drone. It is connected to the server via a TCP-IP communication protocol and will exchange information with it. The server can send for example the next position of the drone and it will receive what is its current position. Based on the information received from the server, the control algorithm will process it and forward it to the flight control component. At the same time, it can request current GPS positioning data, for example, and forward it to the server.

A Raspberry Pi 4B was chosen, which has a dedicated port for interfacing with a video camera used for image processing. For the stability control of the drone, the PixHawk 4 controller was chosen together with the PX4 firmware, which offers superior flight control performance. An M9N type auxiliary GPS module was also chosen, which has a positioning error of 1 meter. The flight controller will be connected to the Raspberry PI using a UART serial port.

6.1. Drone control performance

The first step to set the drone to fly is to install the control program on the flight controller board. There are several variants available for PixHawx 4, but the best performing is PX4, version 1.14. Using the QGroundControl application, this program was installed, and the quadcopter further configured. It is necessary to calibrate the sensors, define the number of motors, define and measure the battery and install the radio remote control for manual control of the drone.

The hardware and software performance of this flight controller has been validated by manual control using a remote controller. They were evaluated directly in flight, without the need for static stand testing. Through this test it was observed the performance of the model in maintaining a fixed position in the air with minimal oscillations that do not cause a longitudinal movement. The second test evaluated how the model responds to a series of angular positioning commands in the X and Y axis.

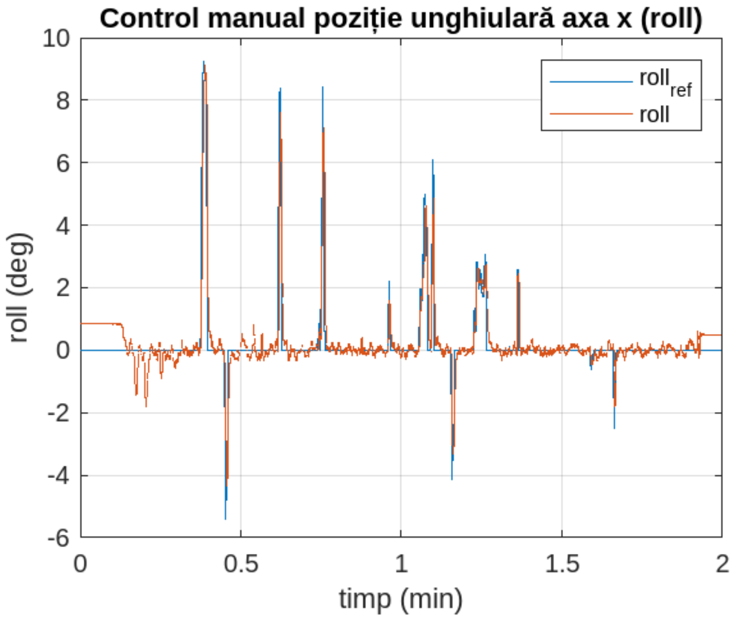

Figure 8: Manual control test results angular position X-axis (roll)

As can be seen in Figure 8, the position around the X-axis is quite stable. During the first test period, a maximum tilt down to about -2˚ is observed. The amplitudes are small, even negligible, which shows that the model can maintain a fixed posture. In conclusion, this type of flight controller is one that meets the required performance criteria and can be easily integrated into any application involving quadcopters.

6.2. Control application design

The PX4 flight controller can be easily interfaced with the Raspberry Pi 4 via the serial port. For communication they use a communication protocol called MAVLink (Micro Air Vehicle Link). It is open source and widely used for integrating UAVs with companion computers.

The most important advantage of this library besides the asynchronous implementation, is that it handles the communication with the flight controller by itself. Thus, the companion, controlling program can set the board in OFFBOARD mode, and set flight references for the drone. There are three types of reference that can be given:

1. **Manual control:** which can be used to directly set the angular position values on each axis and to adjust the speed of the motors. There are basically 4 commands that can be applied here. This type of control is similar to the native radio control.
2. **NED position control:** NED is an acronym for North, East and Down. Using this functionality, a future position can be transmitted in these coordinates. For example there is the requirement to move the drone forward by 2 meters (north), right by 4 meters (east) and up by -10 meters (- for the opposite direction, up) then these values will be transferred to the drone and it will fly to the new position from the current position of the drone.
3. **GPS position control:** by directly transmitting latitude, longitude and altitude, using the GPS sensor, the drone will fly to the indicated coordinates.

These control modes are ideal for defining multiple control times. For example, if a drone is to be positioned by the operator to a certain position, then the operator can choose one of the first 3 modes. For a positioning application, which will do this automatically based on an algorithm, the easiest option is to use the last mode, where the position is transmitted directly.

6.3. Deployment of the server and operating application

The server and the operating application are 2 totally different entities that can run on different platforms or systems. The operating application connects to the server using the TCP/IP protocol. It can either run on the same device, in which case the IP of the server is the IP of the computer, or it can connect from another device on the network by accessing the IP of the server. For example, it can run on a tablet or a cell phone.

For this application Java was chosen as the programming language for the server and the monitoring application. In Java, the Spring framework can be used, which facilitates the creation of client-server applications, by embedding the communication server itself and by using predefined functions to connect to the server functions.

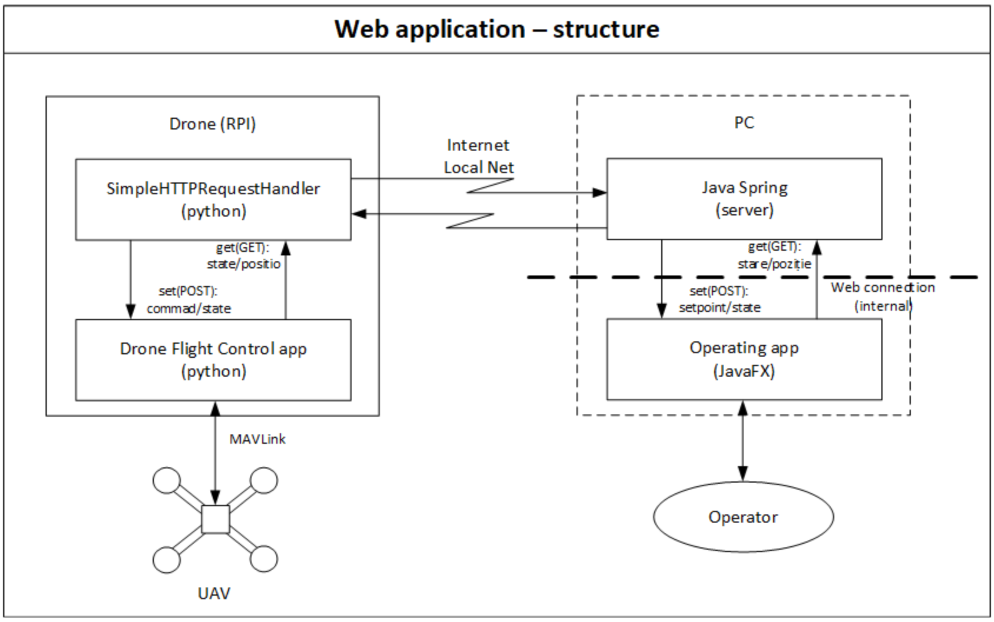

Figure 9: Web application structure

Figure 9 shows the block diagram of the web application. The drone control algorithm was presented in the previous section. Its internal functions are connected to an object that handles the communication between it and Java, SimpleHTTPRequestHandler in this case. This HTTP object, implemented at the Raspberry Pi level, handles communication between Java Spring and the drone command program. Basically, it represents a web interface between the 2 programs.

6.4. Conclusions

In conclusion, the present study has succeeded in creating the communication infrastructure for the final search and rescue application. This infrastructure allowed to evaluate the flight performance of the chosen controller for the drones in the system.

The performance is satisfactory. The drone is stable in the static flight position and responds very well to angular positioning references. For linear position control evaluation, the results are equally satisfactory. The drone managed to track the imposed reference. Even if position error is present, this is normal considering that the GPS module was not used to correct the positioning. The position estimation was based only on the internal inertial measurement sensors by integrating the linear accelerations. Estimates that are normally prone to integration errors.

This and the previous study represent the critical component of the whole system, which has been successfully implemented and validated. Using this model, the study can be extended to the following 2 components, the image processing algorithm, and the search algorithm.

7. Image processing

The Raspberry Pi board chosen in the previous study will be used together with a dedicated Raspberry Pi camera, Raspberry Pi Camera 2. The two hardware components will be mounted on the drone structure to capture the in-flight image.

The image processing application will be built using Python's OpenCV library and the data will be statically validated. The algorithm aims to identify a red square and determine the angle and distance that the identified figure is from the center of the image.

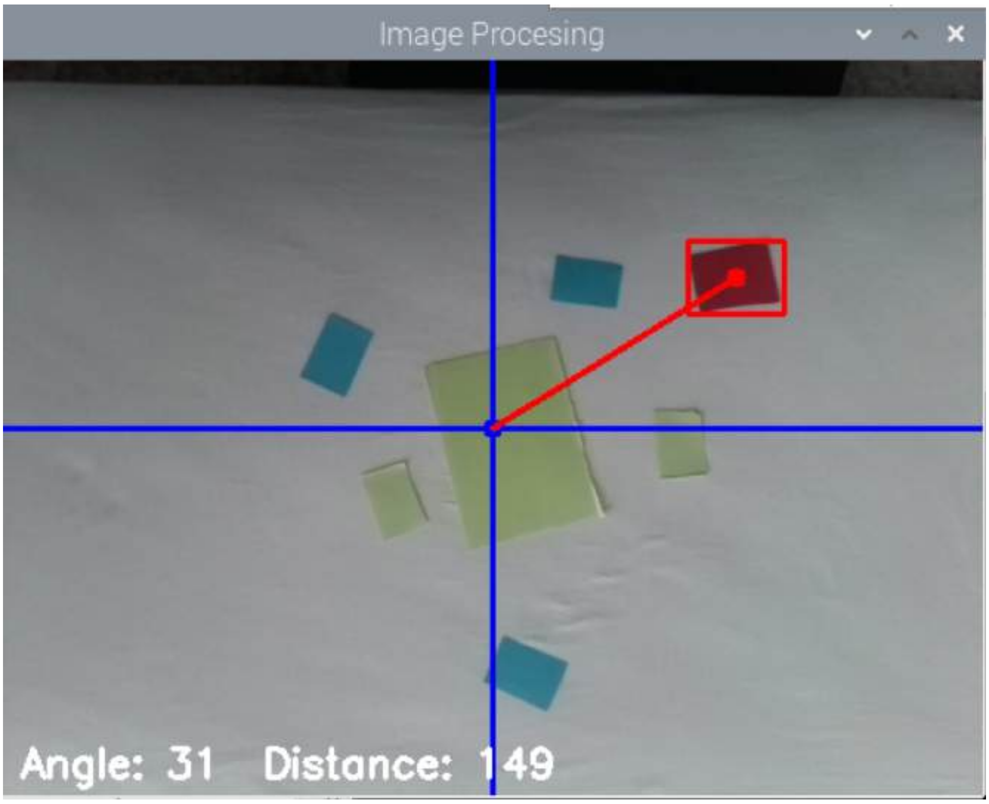

Figure 10: Reference figure identification, final result

Figure 10 shows the final result of the image processing algorithm. It can be seen how the red square is identified, and the distance and angle it makes relative to the center of the image is computed and displayed on the image.

Based on the results obtained and presented above, we can integrate the image processing program with the drone control program. It will run separately as a stand-alone service and will transmit to the Java Spring server the distance and angle it has determined. These two pieces of information will be used to position the quadcopter. For example, if it has identified the shape of an object in the lower left corner of the image (quadrant 4), that means that the drone is moving away from the object, but it also means that it has just identified a possible target. Given that the drone's direction of flight is opposite to the location of the object, the swarm control will update the quadcopter's orientation in the direction opposite to the identified angle relative to the image position. Thus, on the next iteration, the target will appear at the top of the image, between quadrants 1 and 2, at a relative angle of 90˚. The drone will continue its traveling direction towards that point, its position being updated by the control algorithm of the multi-agent system.

In conclusion, the present study succeeded in presenting a simple and fast image processing solution that can be used to identify a target shape. Not only the processed image, but also the computed information (distance and angle) could be used by the control component of the distributed system to fulfill the purpose of this application. This result will then be integrated and adapted in the final solution.

8. Distributed search and localization system

Following the literature review in the first part of this paper, it was found that 3 fundamental components are essential for a drone swarm: communication, formation control and drone control. For communication it is necessary to provide a stable infrastructure and protocol in order to facilitate the exchange of information between all agents of the system. Formation control ensures that the entire structure maintains the required flight posture or shape. This component ensures that each drone is in the correct position and that there are no collisions between them. The actual swarm control that enforces the formation type and directs the system to the common goal or new position. This component is also the link between the swarm and the human factor.

There are several algorithms that can be applied to implement such a system. For example, Particle Swarm Optimization (PSO), which is commonly used in solving cost problems, could be applied to control such a solution. There are other optimization algorithms that can be implemented, but PSO would have direct applicability, being inspired by the way flocks of birds move in order to find food.

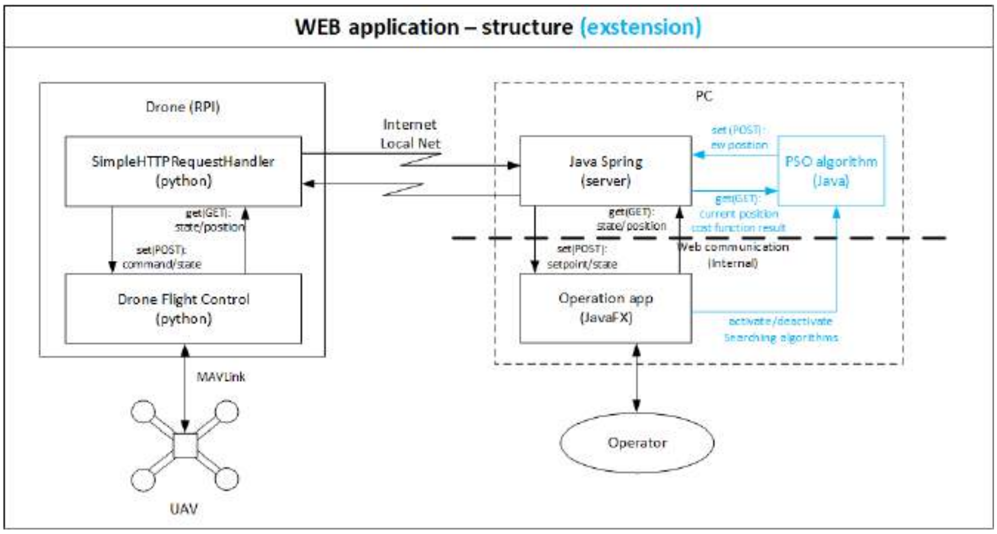

Figure 11: Extended application block diagram for PSO algorithm

Figure 11 shows in blue the PSO algorithm component, which is added to the existing solution. It can be enabled or disabled by the operator. For example, if he wants to reposition a drone, the human operator will deactivate the algorithm, reposition the drone and reactivate the algorithm. Repositioning will also update the position of the agent in the PSO program to work with the newly updated position.

8.1. PSO algorithm – theoretical background

The Particle Swarm Optimization (PSO) algorithm is a nature-inspired optimization technique that mimics the social behavior of organisms, in particular the group behavior of birds or fish. Originally introduced by Kennedy and Eberhart in 1995, PSO belongs to the category of population-based optimization algorithms, in which a group of candidate solutions (called particles) traverses the search space to find the optimal solution. Each particle in the swarm represents a potential solution to the optimization problem and adjusts its position based on two key factors: its previous best position (by evaluating the cost function) and the best position found within the swarm (the position that returns the best result for the cost function). This key aspect makes the PSO search efficiently in the solution space. The behavior of the PSO is governed by a set of equations that dictate how particles update their positions and velocities [14].

$$v_i(t+1) = w \cdot v_i(t) + c_1 \cdot r_1 \cdot (p_i - x_i(t)) + c_2 \cdot r_2 \cdot (g - x_i(t)) \quad (1)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (2)$$

$$p_i = \begin{cases} x_i(t), if\ f(x_i(t)) < f(p_i) \\ p_i\ , else \end{cases} \quad (3)$$

$$g = \begin{cases} x_i(t), if\ f(x_i(t)) < f(g) \\ g\ , else \end{cases} \quad (4)$$

Equations (1 - 4) describe the execution modules of the PSO algorithm, and the variables are defined as follows:

- $v_i(t)$ – the speed of particle i.
- $x_i(t)$ – the position of particle i.
- w – inertial component, speed adjustment factor.
- c_1 – cognitive coefficient, influences the particle's attraction to its best position.
- c_2 – social coefficient, influences the particle's attraction to the best global position.
- r_1 *şi* r_2 – random numbers between 0 and 1.
- p_i – the best personal position of particle i.
- g – global best position.
- $f(x)$ – cost function.

After testing the algorithm on a training cost function, it was found that the rate at which the particle positions vary is too large, and most of the time they end up in the limits of the searching space. To combat this effect a new formula was introduced in the PSO algorithm with the aim of modifying the inertial component w, thus allowing a much slower variation of the angular position [15].

$$w(i) = m - n\frac{1}{p^{best(i)}+1} + q\frac{1}{r^{bestf}+1} \quad (5)$$

where m, n, p and r are subunit values that have been chosen experimentally, and best(i) is the personal best solution of particle i and bestf is the best solution found so far.

8.2. PSO validation on real system

The test aims to validate the search algorithm on the real system using 3 drones. An open search area was chosen, the whole system infrastructure (local area network, server and operating application) was set up. The PSO algorithm was imposed a searching area, which could not exceed the GPS coordinates [46,777159 23,707359] and [46,777449 23,707512].

The evolution of Drone 2 is shown in Figure 12. It has been initialized in the upper-left corner. For this particle it can be observed that its position is updated with a rather large variation. That is, the next computed position is at the opposite extreme of the current position. This behavior may also be an advantage, because the drone covers a larger searching area and may have a chance to identify the target figure. The drone flew past the figure several times, identified it, but did not stop at the figure, because the position imposed by the algorithm was somewhere else.

As it can be observed, the fact that this drone identified the figure at one point made it find the location of the figure. The last position when the algorithm completed its execution was right next to the target.

Figure 12: PSO validation –Drone 2 evolution

The data stream retrieved from Drone 2 is shown in Figure 13. Figure 13 (a) and (b) show the image retrieved at the time of the last iteration, when the algorithm has completed and returned the position of the target object.

Figure 13 (c) and (d) show an example where during the flight the drone passes by the target object (10.9.(c)), identifies it, but moves away from it (10.9.(d)) while moving to the next position. In this case the value provided to the algorithm was about 182, being the last distance successfully identified. This may also be an advantage, as the PSO algorithm determines that it is approaching the desired point. On the other hand, this behavior may lead the current particle (or other particles) to a point of local minimum. This is because the program may think that the current position is close to the solution, even though it may be at the opposite extreme.

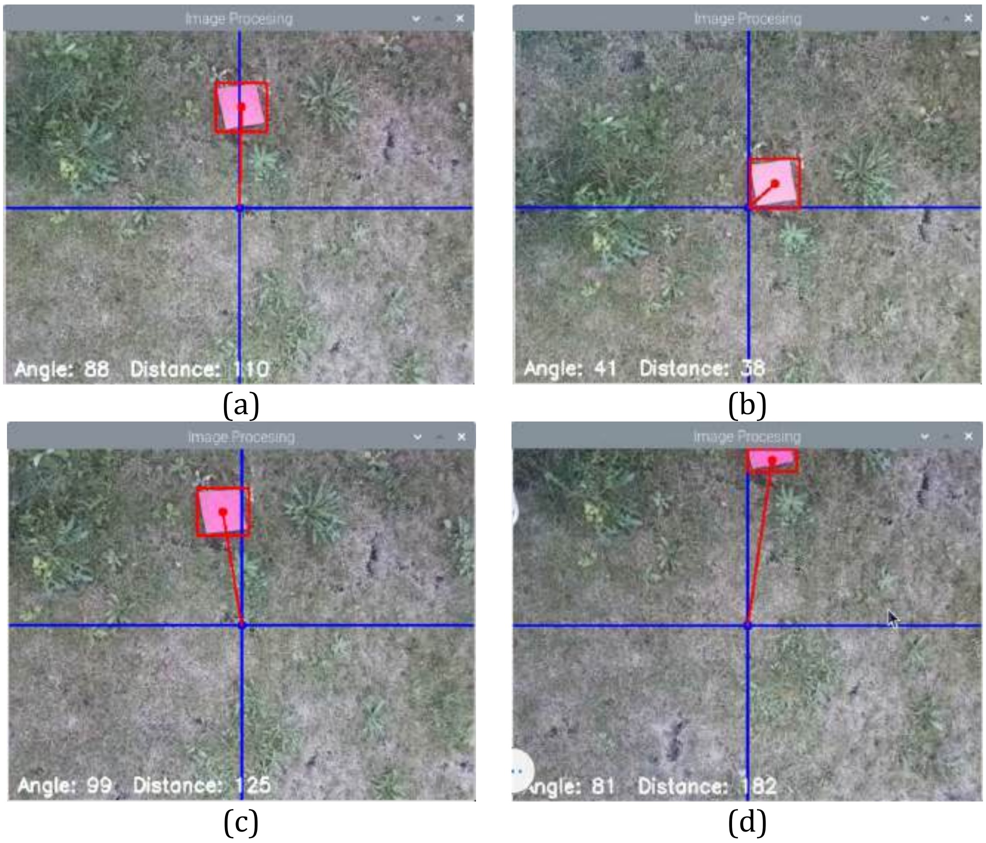

Figure 13: PSO validation – video stream from Drone 2

8.3. Conclusions

The PSO algorithm can be considered suitable for such an application. Although the current solution did not provide the most optimal response (2 drones out of 3 got stuck in a local minimum point), it proved its applicability, satisfying the objective imposed by localizing the target figure (red square).

The main observation is that the present algorithm needs improvement and adaptation for such a system, in order to increase its efficiency. First of all the sequential execution prevents real-time localization. To combat this the PSO program needs to be adapted to work between iterations. As the drone locates a possible target its position should be updated in the direction of that target, without waiting for the drone to arrive at the initially computed future position. This can be done either by decreasing the future position step and increasing the execution speed of the algorithm independent of whether the drone has reached the initially computed position or not. Or by introducing an internal PSO loop, dedicated on each agent, allowing it to update its position in the current search space if it has identified a possible target. At that point the other drones that haven't identified anything yet will continue their execution according to the program in the outer loop, while the other drone will run according to its own inner loop. Results that it will also override the algorithm in the outer loop.

The second observation to be made is that it is also necessary to take into account the angle that is made when the figure is identified. It may represent as a direction, or a vector of the next motion, and the future point be directly influenced by that direction.

Keeping in mind that the searching area most of the time is known, the PSO algorithm should be forced to look for solutions towards the inner part of the area and not towards the extremities. Optimizing the algorithm in this way increases the chances of identifying the imposed target.

It was hypothesized that a PSO algorithm can be used in a search and localization application. This study demonstrates that the system created for this research can fulfill the purpose of the research, namely search and localization. The results gathered provide insight and formulate a future development trajectory that can increase the efficiency and response time of the algorithm.

9. Discussions

The potential of UAVs and their applicability are of very high interest at the moment, and researchers and the business community are doing their best to provide the most stable and reliable solution for integration into any application. The results are already extremely good and promising. Agriculture uses drones not only for crop monitoring, but also for specific actions such as fertilization or disease control. The construction industry is benefiting from the development of tools by integrating UAVs to assess and compare how a project is executed against its blueprint. For disaster management, drones are also enjoying popularity. They can be deployed in hard-to-reach areas to obtain information about the area or provide aid to victims of such disasters. In all of these applications, the man is the main winner, benefiting from safety, lower response and deployment time, and optimizing the cost with which a given activity can be carried out.

Distributed drone systems have started to become popular in the last 8 to 10 years, when a number of research on algorithms (collective intelligence) for managing large populations of robots started to be developed. This sector is still not fully explored and there are many questions that are looking for answers, but the direction is promising. Several agents involved can solve a problem much faster and more efficiently than one.

As it has emerged from the results presented in this thesis, a search and localization system can be developed with some ease and can be applied on a real case. Current technology enables both stable and robust drone control methods and efficient image processing with relatively low cost and high performance. It was also found that an algorithm such as PSO is quite promising to be used as a collective intelligence algorithm for the control of distributed drone systems.

Of course, in addition to these promising and motivating results and directions for future development, there are also hurdles. The main obstacle is the small battery capacity, which limits the flight time of a drone. This impediment can be overcome for the time being by a multi-agent drone system, as one drone can substitute the activity of another drone when the latter runs out of power.

Communication network coverage is also an impediment. The existing telecommunications infrastructure does not provide global coverage. In some areas expensive communications equipment must be deployed to make these systems functional. A future solution is on the horizon, via the internet provided by satellites, to which drones could also connect. This next-generation internet could also ensure global coverage for a distributed drone system.

Current US and European legislation do not yet approve the widespread use of UAVs for commercial purposes, except in a number of exceptional cases. The commercial solutions, end-user-enabled solutions are currently restricted, mainly on safety reasons. Once robust solutions capable of responding to failures and public safety criteria are created, legislation will start to open up and regulate the large-scale usege of UAVs.

It is precisely the unresolved issues that make this a subject of enormous importance and potential. An industry has started to develop which is starting to employ more and more people every year, and the economic impact it will have will also be significant for the global economy.

10. General conclusions

In conclusion the present thesis has managed in the first phase an extensive bibliographical survey of the most important literature in the field of UAV applications and distributed UAV systems. This information formed a foundation for the thesis further on. Based on the principles and conclusions drawn from this phase, the following studies were built and presented in this thesis.

The first study concludes that modern tuning structures can improve the in-flight stability of a quadcopter. An adaptive tuning structure using fractional PID controller was the novelty. It was found that such a structure can meet the required specifications and be efficient in terms of control effort compared to a classical structure.

The second study aimed to build a distributed application to realize the remote control of a quadcopter. The results obtained were satisfactory. The developed application was stable and could command the movement of a drone in the air. It was observed that the drone follows the imposed reference, returns to the initial location and communication problems were non-existent. This application is the foundation of the distributed system that will be used in search and localization activities.

Image processing to identify a target figure was presented in the third study. In this case the drone had to identify a red square. The results were more than satisfactory. By using the OpenCV library, which was developed and improved in this direction it was possible to identify, track and calculate the angle and distance from the center of the image to the target shape. Later this solution was also analyzed in a real scenario.

Following the testing of the distributed system on a real scenario it was concluded that such a solution is implementable and provides viable solutions. The results were presented in study 4, where the PSO algorithm was applied to determine future GPS positions to reach the target. The most important conclusion and observation from this study is to identify how the performance of such a system can be optimized for faster and better response. The inefficiency was that when the drone was flying from one location to another and passing by the target, it did not stop but on the contrary, in most cases it went out of the visual range of the target. As a solution the following 3 directions were proposed:

1. Increase program execution frequency. For the presented solution the algorithm waited for the repositioning of all the drones, then proceeded to the calculation of the next iteration. By increasing the frequency, the PSO algorithm can adjust the trajectory, by picking up information that the drone picks up during flight. Moreover, in this case the integration step used to compute the next position must be reduced. Thus, the initially calculated trajectory will be segmented into smaller intermediate points.
2. Create an internal PSO loop for each drone. This loop will only intervene if it identifies a possible target. In this case, the drone will work independently of the PSO algorithm, it will take control and correct its trajectory based on the video information it takes. If indeed the found object is really the wanted one then the drone stops and will communicate the result to the main algorithm, which will stop the execution.
3. Introduce a component that forces the PSO to position drones inside the search area, so all future positions are computed as close to the center of the area as possible. This solution would only work for cases where the search area is known.

In conclusion, the present thesis has succeeded in demonstrating that the PSO optimal algorithm, can be applied in search and localization applications. The implemented system has managed to provide at least one solution that confirms this hypothesis.

In addition, the following future directions are proposed to develop and improve the performance of the presented solution:

1. Optimization of the PSO algorithm, by introducing a new strategy to allow the segmentation of flight trajectories in order to identify the target faster.
2. Implement a software component to prevent drone collisions.
3. Validation of the new system on a real system using drones in flight.

REFERENCES

[1] "Wikipedia - Search and Rescue." [Online]. Available: https://en.wikipedia.org/wiki/Search_and_rescue

[2] "RNLI foundation." [Online]. Available: https://rnli.org/about-us/our-history/timeline/1824-our-foundation

[3] "Salvamont Romania - History." [Online]. Available: https://www.salvamontromania.ro/istoric.php

[4] Y. Alghamdi, A. Munir, and H. M. La, "Architecture, Classification, and Applications of Contemporary Unmanned Aerial Vehicles," *IEEE Consum. Electron. Mag.*, vol. 10, no. 6, pp. 9–20, Nov. 2021, doi: 10.1109/MCE.2021.3063945.

[5] B. Alzahrani, O. S. Oubbati, A. Barnawi, M. Atiquzzaman, and D. Alghazzawi, "UAV assistance paradigm: State-of-the-art in applications and challenges," *J. Netw. Comput. Appl.*, vol. 166, p. 102706, Sep. 2020, doi: 10.1016/j.jnca.2020.102706.

[6] A. Gupta, T. Afrin, E. Scully, and N. Yodo, "Advances of UAVs toward Future Transportation: The State-of-the-Art, Challenges, and Opportunities," *Future Transp.*, vol. 1, no. 2, pp. 326–350, Sep. 2021, doi: 10.3390/futuretransp1020019.

[7] G. Singhal, B. Bansod, and L. Mathew, "Unmanned Aerial Vehicle Classification, Applications and Challenges: A Review," Nov. 27, 2018. doi: 10.20944/preprints201811.0601.v1.

[8] M. Hassanalian and A. Abdelkefi, "Classifications, applications, and design challenges of drones: A review," *Prog. Aerosp. Sci.*, vol. 91, pp. 99–131, May 2017, doi: 10.1016/j.paerosci.2017.04.003.

[9] "Open Category - Low Risk - Civil Drones | EASA." Accessed: Apr. 24, 2024. [Online]. Available: https://www.easa.europa.eu/en/domains/civil-drones-rpas/open-category-civil-drones

[10] "How to Register Your Drone | Federal Aviation Administration." Accessed: Apr. 24, 2024. [Online]. Available: https://www.faa.gov/uas/getting_started/register_drone

[11] F. Ahmed, J. C. Mohanta, A. Keshari, and P. S. Yadav, "Recent Advances in Unmanned Aerial Vehicles: A Review," *Arab. J. Sci. Eng.*, vol. 47, no. 7, pp. 7963–7984, Jul. 2022, doi: 10.1007/s13369-022-06738-0.

[12] P. Lanusse, J. Sabatier, and A. Oustaloup, "Fractional Order PID and First Generation CRONE Control System Design," in *Fractional Order Differentiation and Robust Control Design*, vol. 77, in Intelligent Systems, Control and Automation: Science and Engineering, vol. 77. , Dordrecht: Springer Netherlands, 2015, pp. 63–105. doi: 10.1007/978-94-017-9807-5_2.

[13] P. Lanusse, J. Sabatier, and A. Oustaloup, "Extension of PID to fractional orders controllers: a frequency-domain tutorial presentation," *IFAC Proc. Vol.*, vol. 47, no. 3, pp. 7436–7442, 2014, doi: 10.3182/20140824-6-ZA-1003.01053.

[14] J. Kennedy, "Particle Swarm Optimization," in *Encyclopedia of Machine Learning*, C. Sammut and G. I. Webb, Eds., Boston, MA: Springer US, 2011, pp. 760–766. doi: 10.1007/978-0-387-30164-8_630.

[15] E.-H. Dulf, C. I. Muresan, and Daniel D. Timis, "Adaptive fractional order control of a quadrotor," *Dyn. Syst. – Theory Appl.*, pp. 129–138, 2019, doi: https://doi.org/10.34658/9788366287303.

PUBLICATIONS

1. Timiș, D. D. & Dulf, E. H. (2024). Unmanned Aerial Vehicles: State-of-the-Art on Swarm Control, Journal of Control Engineering and Applied Informatics, Vol. 26 No. 4, 69-80, DOI: 10.61416/ceai.v26i4.9270.
2. Timiș, D. D., Muresan, C. I., & Dulf, E. H. (2022). Design and experimental results of an adaptive fractional-order controller for a quadrotor. Fractal and Fractional, 6(4), 204. WOS:000786793700001(Q1), https://doi.org/10.3390/fractalfract6040204.
3. Timis, D. D., & Dulf, E. H. (2020). Software implementation and test of an advanced robust control applied to a quad-copter. In 2020 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR) (pp. 1-4). IEEE., WOS:000643732600075 https://ieeexplore.ieee.org/document/9130018.
4. Dulf, E. H., Timis, D. D., Szekely, L., & Miclea, L. C. (2019). Adaptive fractional order control applied to a multi-rotor system. In 2019 22nd International Conference on Control Systems and Computer Science (CSCS) (pp. 696-699). IEEE. WOS:000491270300117 https://ieeexplore.ieee.org/document/8744692.
5. Dulf, E.H., Muresan, C.I. and Timis, D. D. (2019) Adaptive fractional order control of a quadrotor. In: 2019 International Conference "Dynamical Systems – Theory and Applications" (DSTA) (pp. 129-138). ISBN 978-83-66287-30-3. https://doi.org/10.34658/9788366287303.

ARTICLES UNDER REVIEW

6. Timiș, D. D. & Dulf, E. H. (2024). Unmanned Aerial Vehicles: State-of-the-Art in Applications and Swarm Control, IEEE ACCESS, ISSN 2169-3536.