Project Code: AAE03a

THE HONG KONG POLYTECHNIC UNIVERSITY

BEng (Hons) in Aviation Engineering

2021/2022

AAE4002 Capstone Project Interim Report

Autonomous Landing of UAV on Mobile Platform

LEUNG Cheuk Hei (18048844D)

WAN Chun Yan (18048531D)

YEUN Wai Ki (18046033D)

Academic Supervisor:	Prof. CY Wen
Industrial Supervisor (if any):	
Date of Submission:	28/12/2021

Table of contents

Chapter 1. Introduction

1.1 UAV Background and Development

Unmanned aerial vehicle (UAV) is an aircraft which does not require an onboard human pilot and is usually controlled by remote control or autopilot. The implementation of UAV in different fields has been wide and fruitful in the past decade. It becomes an important tool in business (Herwitz et al., 2004), government (Girard et al., 2004), mapping (Nagai et al., 2009), and search-and-rescue (Waharte & Trigoni, 2010). Stepping in the 21st century, benefited from the development of smart technologies and enhanced electrical power systems, the market of the small quadcopter, which is the helicopter with four rotors, has grown exponentially. The growth of quadcopters accelerates the implantation of UAV in different fields, which a part of the UAV functions is being autonomous.

For the basic structure of a UAV, the concept is shown in Figure 1. Processing units, sensors, actuators, and energy supply are the essential components of a UAV. If the UAV is under remote control, communication module is required. If the UAV can be operated by autopilot, the communication module is not mandatory, the flight control of the UAV would only depend on the computing power of the processing unit. The processing unit act as the brain of the whole system. It is responsible to control different parts of the UAV to implement flight control, such as grabbing data from sensors, managing the outputs of the actuators. (Alvika et al., 2014)

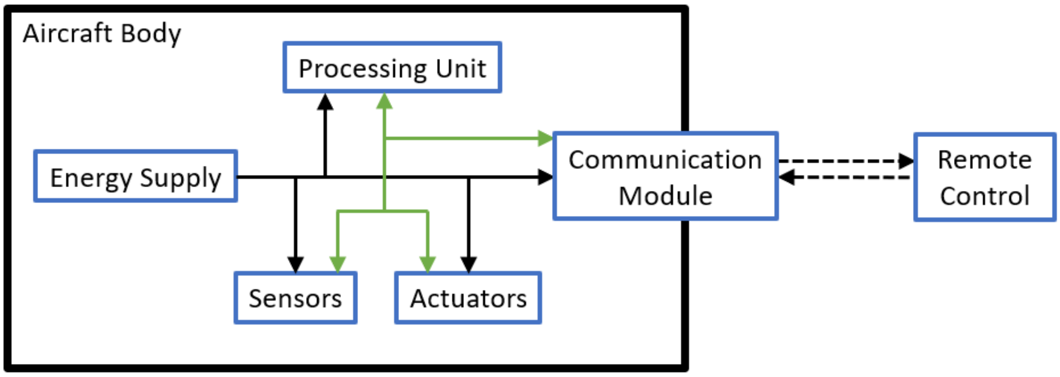


Figure 1. Basic Physical Structure of UAV

1.2 Autonomous Landing

Flight controls of an aircraft consist of serval stages, including take-off, ascent, cruise, descent, and landing. UAV may also have these stages of flight controls. Similar to the large, manned aircraft, the most challenging and dangerous part in-flight controls is landing. Given that accidents often happen during landing and hence raise high risks and reliability issues. As only limited amount of time and space are allowed for a landing procedure, precise sensing and accurate controls are required for landing otherwise the drone may have physical damages together with safety concerns. In accordance with the historical data, human factors have been the main elements contributing to the causes of aircraft incidents. Therefore, it is believed that manual flight controls should be replaced by autonomous landing to eliminate the incidents due to improper human handling.

Autonomous landing can be developed to have a better performance on facing different types of environments like indoor and outdoor, terrain, wind disturbances, visibility. For which is difficult to control the drone manually like the poor connections during remoting the drone by the controller, and no localization system on the platform, has not been developed and discovered. In these circumstances, the UAV becomes a way to explore some new places, beneficial to scientific and environmental works. Then, this feature may save the manpower on controlling the drones and become a more reliable approach without human interventions.

Furthermore, the autonomous landing will become trending feature for the new-invented drones or even large aircrafts as it has a number of usages with great demand. The autonomy of flight controls consists of decisions making on their own and taking actions accordingly based on self-analyzed data obtained from the sensors. Even to recognize and solve tackles themselves. Therefore, with more mature development on such features, the autonomous landing of UAV may soon be widely spread to civil or even households' applications.

1.3 Mobile Platform

In the automation of UAV, a more precise landing can be achieved by removing the human factor. With autopilot or autonomous landing function, UAV can land with high performance under different conditions. Despite landing on a fixed landing platform, the autonomous landing can be carried out on a moving platform. This requires the UAV to trace and follow the moving platform in the landing procedure.

The autonomous landing on mobile platform would extend the operation range and increase the choice of landing position of the UAV. The mobile platform can be a moving car or a sailing ship. The application can lower the difficulties of landing on a moving station. Also, the autonomous process allows the UAV to land on the mobile platform without network and only relies on its own process power. Therefore, the UAV can extend its operation range and land on mobile platform without external command.

1.4 Current Problem

There are some general problems of operating a UAV. Due to the characteristic of UAV being light and maneuverable, it is usually short in range. For instance, many light drones are usually equipped with a lithium polymer rechargeable battery which can only sustain for 10-30 minutes flying time only. This results in UAV having to rely on recharging stations throughout the operation area and limited flight performance due to the small power supply.

Another shortcoming of a light drone is its dimensions. Considering the distribution of weight and balance together with the center of gravity, the allowed size for a payload shall not be larger than the diameter of the UAV frame, especially for the multi-motors drone. The payload should be hung at the center of gravity of the UAV. Hence it is afraid that the UAV is still not a good transport for cargo delivery. Also, the space for employing the onboard computer may lead to weaken the computational power, and then decline the ability of trajectory planning and data processing.

Therefore, depending on the mission type, this could massively decrease the flexibility of operation. Such shortcomings of light UAV may be the obstacles on the further development and invention.

There are some problems on implementing the autonomous landing of a UAV. Most of the consumer-grade UAV use the Global Positioning System (GPS) for determining its own location. The civil GPS receiver can achieve positioning accuracy between 1 meter and 30 centimeters. However, the accuracy of GPS is not good enough to implement precise autonomous landing of a small quadcopter. Also, if the UAV only relies on the GPS in the autonomous landing process, it cannot recognize the environment and the danger surrounding the landing platform. To increase landing accuracy and environment awareness, image positioning can be adopted.

For the image positioning, the processing unit of the UAV equipped with normal camera requires to convert to a 2-dimension image into depth data. It has a high requirement on the computing power of the processing unit leading the high energy consumption and shortens the flying time of the UAV. To obtain depth data with lower power consumption, depth camera can be equipped, or use the specialized maker, i.e., ArUco Marker, which will be introduced in the following section.

1.5 Project Contribution

The autonomous landing of UAV on mobile platform would have different applications in different industries. The first example is the mobile charging station. After conducting automated missions such as surveying, UAV can land on the moving charging station and carry on the next mission in another location without any human operation. The second example is the offshore landing. UAV can be operated in the middle of the ocean and carry-on small cargo delivery. Although the UAV is under the environment without network, it can still land through the autonomous landing function without human remote control. These are two examples of the application of autonomous landing of UAV on mobile platform.

1.6 Objectives and Organization

This project aims to solve these problems by implementing a complete system architecture for UAV to land autonomously on moving platform under various conditions. Our implementation consists of a vision-based target detection via ArUco marker; location measurement; and integrated control. Of which the landing is accomplished by the following parts:

1. Presumptions: *(i)* Other ways such as GPS module had already guided the UAV

to the proximity of the landing target; *(ii)* The landing platform, although moving, is a cooperative party in the system and will travel in a path without aggressive movement.

2. UAV identifies the ArUco marker using vision module for position estimation.
3. Flight controller approaches the platform based on the position estimated.
4. Descend when the angle between the UAV and platform is within the error.

The remaining part of the report is allocated as follows. Chapter 2 may first review serval literature and then summarize some key points for our project. Chapter 3 is going to showcase our work scope, main tasks, and problems to be handled, and selected tools and approaches to addressing the problems. Then, Chapter 4 will present the preliminary results with discussion of what we have obtained at this moment. Lastly, Chapter 5 is the project management which explains the current progress and the overall schedule, and the plan for the upcoming project period.

Chapter 2. Literature Review

2.1 Comparing different landing techniques for UAVs

Landing a UAV is a complicated task. A number of autonomous landing technologies have been developed. A paper done by (Alvika et al., 2014) reviewed the autonomous landing techniques. Typical aerial vehicle uses Global Positioning System (GPS) and inertial navigation sensors (INS) in the landing system. However, aircraft cannot rely on the GPS only during the landing process, as the height measurement from GPS is not accurate enough. As a result, other range sensors are required such as radar altimeter. Therefore, GPS cannot be used in UAV precise landing but can only be applied to the general navigation of the UAV.

The article suggested that vision-based landing will be more applicable in the autonomous landing of UAV. The vision-based landing is suitable for the situation where the landing pad is located at an unknown position or non-stationary such as moving cars and sailing ships. The vision-based landing relies on object detection or pattern recognition techniques. It can be achieved by several approaches. The first one is image segmentation; the processing unit would partition the image. Partitions are assigned a label and can be used to identify lines, boundaries, and patterns of the objects. The second one is monocular vision which increases the field of view that is close to human vision. The third one is stereo vision. It can conduct the 3-dimension information from the images and obtain the distance between the UAV and the target objects. By controlling the distance with the target objects, the UAV can implement precise landing through the vision-based approaches.

This shows that image positioning has greater advantages than GPS and other landing techniques in the precise UAV autonomous landing.

2.2 Example of autonomous landing system

Autonomous landing can be applied to the automated drone charging system which does not require manual control in the operation. According to (Jiang, 2019), the infrastructure of autonomous landing system is the combination of GPS navigation, ArUco marker localization, and position control of the UAV. The flow of the system is divided into three sections. When the drone is detected as low battery, the onboard processing unit will start the landing process. The UAV will first fly to the charging station using GPS navigation. The second stage is implementing the accurate landing by using ArUco Marker detection. The algorithm will ensure the drone hovering right above the center of the marker by repeating the second stage of the process. To counteract the errors caused by wind, the system would keep aligning the maker until the UAV is under target altitude. The adjustment of the positioning is the third stage of the system. At the end of the task, the UAV would descend to the ground once the maker is within the acceptable range. The charging system would start after the landing of the UAV.

2.3 ArUco marker and Embedded ArUco (E-ArUco) marker

An ArUco marker is a marker with predefined size and proportions so that the relative rotation and distance can be extracted from the camera image with relative ease when compared to other visual pose estimations with much uncertainty of an unknown object. Its internal binary coding facilitates the reliability and error correction needed for a precise UAV landing autonomously. It consists of a black boundary and an internal matrix that is built a base of the N x N cells entries with a unique identifier (ID). It is designed in such a way to speed up the detection process.

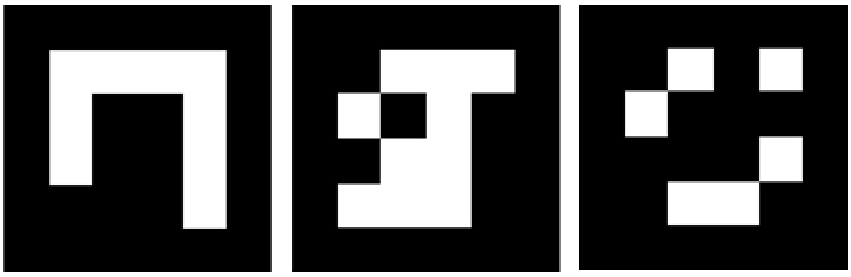


Figure 2. From left to right, 4x4 ArUco markers with IDs of 10, 100, 200.

The ArUco marker is a vital part of the UAV's vision-based position estimation tool. For a precise UAV landing utilizing visual sensory information, robust and continuous detection of the marker is essential for the pose estimation throughout the approach at different altitudes, which means the marker should be detectable within the angle of view of the onboard camera during the entire process when attempting a landing. However, such requirement is easier said than done for the following reason: a large marker is preferred so that it can be detected from a long-range, but such marker will exceed the camera field of view by the time the UAV approaches close enough for a landing shut down.

In the Conference Paper done by (Artur et al.), an e-ArUco design is proposed. A smaller ArUco marker is placed within another larger marker at the center of the outer marker, with the two having different IDs. The larger marker is used in the early stage of the approach for rough positioning, while the smaller marker is used when the distance between the onboard camera and the target is too close for the ArUco marker to be seen by the camera. The inner ArUco marker is placed at the center of the larger marker, and to be seen as a single black block of the larger marker. As a result, the marker ID selection should be well noted in such a way that the larger marker has a black block at the center, and the small marker has more black blocks than the white ones. With their different size, so as the detection range. E-ArUco marker can combine the minimum detection distance of the smaller marker and the maximum detection distance of the larger marker to enrich the range of detection.

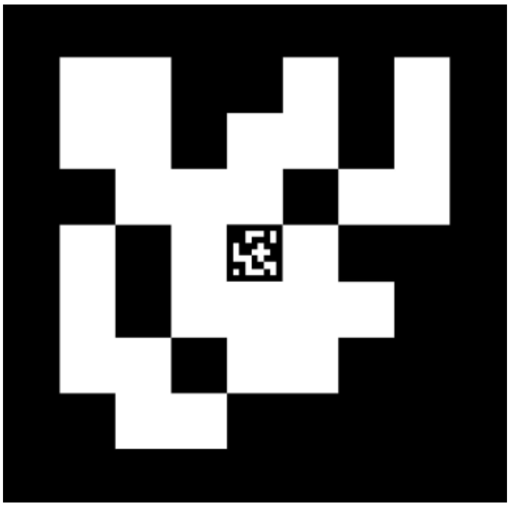


Figure 3. Example of a 7x7-within-7x7 e-ArUco marker.

2.4 Data filtering

The approach of this project utilizes computer vision in a crucial way in the autonomous landing procedure. It is important for the data obtained to be accurate so that the system can produce the appropriate response to the changing situation observed. Should the information obtained be false, the system will not be able to produce the correct response. When flying above the target, the motion blur introduced by the UAV movement and the error of camera ArUco detection could be problematic to the position and velocity estimation of the target.

Kalman Filter (KF) can be implemented in the algorithm so that the position and velocity data can be smoothed out and the error in which can be corrected. (Miguel-A. et al., 2013) The KF is a feedback control estimation, with the filter estimating the result at a time and the feedback is obtained from the measurements. Which the measurements would usually be noisy.

In a KF algorithm, a time equation is constructed for projecting the state for the next time step as a priori estimate, and a measurement equation for the feedback taken, and an improved posterior estimate. To put it simply, a time equation to predict the result and a measurement equation to update the correction. (Greg & Gary, 1995)

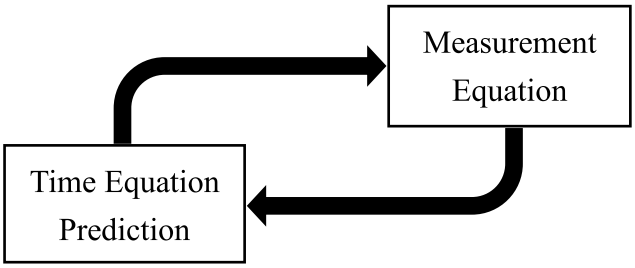


Figure 4. An ongoing cycle of Kalman Filter. The time equation predicts an estimation ahead of time; The measurement equation corrects such estimation with the measured data at the time.

2.5 Algorithms in explorations and Path planning

As UAV is applied to handle tasks in dangerous environments to save and secure the labor force and make their lives more convenient nowadays. Nevertheless, as the same as human worker, flying drones also need to face many challenges in the working site including low visibility, narrow spaces, dusty, and high temperature. Even if the operator remoting the UAV in another place, they may also need to deal with the poor wireless connection for communication and navigation problems. An autonomous path planning algorithm should be exploited for UAV implanting missions in different hazardous environments.

In order to resolve the above-mentioned challenges, the path planning algorithm should characterize the following features and expectations. Given that the environment is unknown and there are unpredicted hazards, the algorithm should be capable to provide real-time expedition data for path planning to safeguard the clearance of the pathway. Also, the algorithm must have high reliability and capability in the hardware drone. The runtime for calculating the paths should be short and have high energy efficacy to complete the entire operations within the battery capacity.

In the paper by (Moletta, 2020), a master student at KTH Royal Institute of Technology, designed an optimized version of a path planning algorithm specific for vertical tunnel exploration. In accordance with the above requirements, the Autonomous Exploration Planner (AEP) has been proposed after some reviews. This approach is operated by choosing the best node and generating a pathway without collision with obstacles. It is a dynamic approach that the pathway is still being modified by time after it is built, while this may lead to heavy demand on the onboard computer and longer mission time. Thus, an optimized version of this approach has been made by adjusting parameters and modification of programs, namely shaft-AEP (s-AEP).

After a number of experiments and implementations, the exploration time taken to complete the whole tunnel pathway of s-AEP is shorter than AEP's. Although the s-AEP takes a longer length of trajectory but performs faster and less consumption of energy in both exploration and path planning in most of the scenarios.

2.6 Collision-free path generation

Making use of UAV in modern life such as goods delivery, search and rescue, and surveillance is crucial for small, unmanned flying drone development in the future. Path planning is an essential element when UAV is flying without human interventions. Indeed, for the developed areas especially our living areas, which are intricated and dynamic, makes trajectory generations not that easy. Hence, obstacle avoidance will be an important feature in the algorithms of path planning to safeguard the drones and payloads and complete the task without surrounding disturbance. A success path planner should be able to draw feasible trajectories without suffering any collision in a fast timely manner, noting that both movable and static objects will change over time.

In the paper of (Sanchez-Lopez et al., 2019), a real-time path planning solution in 3D for a UAV to generate paths feasibly and free of collisions in a complicated dynamic environment. Following is the graph showing the planning solution suggested in the paper.

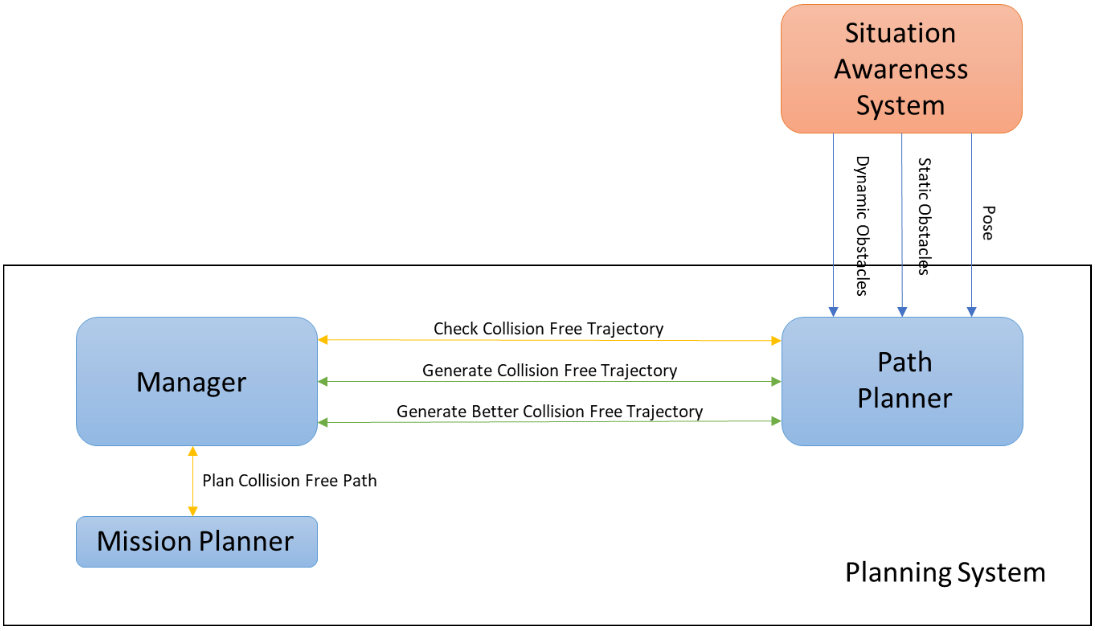

Situation Awareness System
Dynamic Obstacles
Static Obstacles
Pose
Manager
Check Collision Free Trajectory
Generate Collision Free Trajectory
Generate Better Collision Free Trajectory
Path Planner
Plan Collision Free Path
Mission Planner
Planning System

Figure 5. Path Planning Solution

Referring to figure 5, the entire path planning solution consists of two main parts: Manager and path planner. There are three types of formats for internal and external communication: blue arrows are the data streams; green arrows are the services that represent requesting or replying tasks; amber arrows are actions that represent the enforcement on replacing the infeasible tasks. The function of the planner part is to generate a free-collision path contributing to the predicted pose of the drone and estimates the existence of the obstacles in the environment. The situation awareness system is responsible for giving the essential data streams for computing the collision-free path which is the reply of the planner. The manager acts as the agency of the mission planner, receiving the request from it and returning actions with the most optimum pathway without suffering collisions. Then the manager makes a request to the path planner and then activates the checking action, continuously calling the better reply from the path planner.

Chapter 3. Methodology

The landing system contains three modules: *(i)* An ArUco-based Location Estimation module; *(ii)* An UAV Action Planning module; *(iii)* An UAV Flight Control System module. The architecture of the whole system is provided below. The modular design allows a different operation to be switched on and off independently for testing, and ease the modification for the module algorithms.

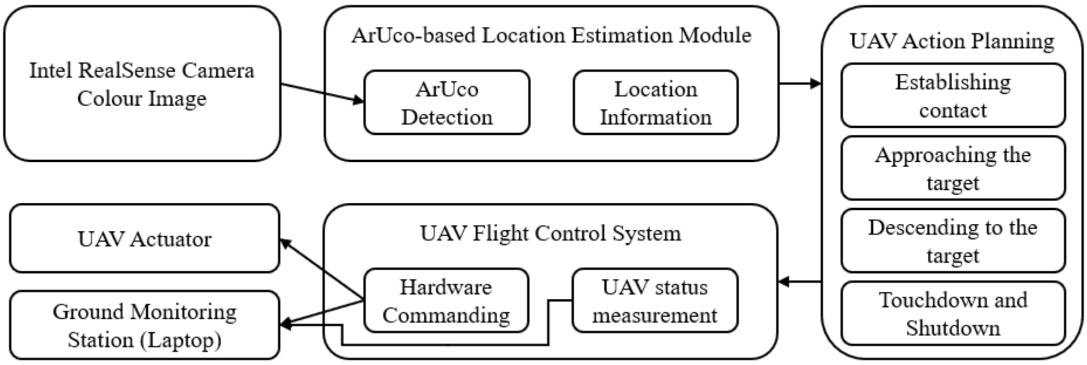


Figure 6. The system architecture

3.1 ArUco-based location estimation

With the library of ArUco, the system can get the position with respect to the camera and the orientation of the marker. Using ArUco marker can minimize the effort of object recognition in vision-based target tracking, and reduce the computational power needed when compared to other vision-based systems such as Artificial Intelligence. With the distance be between the UAV and marker, and the orientation of the marker known. Since the camera is always pointing downward, we can obtain a polar location of the marker, with reference to the camera frame:

1. An angle between the y-axis of the camera and the marker on the x-y plane.
2. Distance between the camera and the marker on the x-y plane.
3. The height between the camera and the marker.

The calculation of all frames feed from the camera (~30Hz) is desired for the most accurate result. Those relative position measurements are based on the visual data which may come with some noise. To reduce the error of the measurement, Kalman Filter is used to optimize the prediction as follows:

$$x_t(k+1) = A_t x_t(k) + \omega(k) \tag{1}$$

$$z_t(k) = C_t x_t(k) + \upsilon(k) \tag{2}$$

$x_t(k)$ is the state at time k, $z_t(k)$ is the measurement at time k, A_t is the transition matrix, $\omega(k)$ and $\upsilon(k)$ are zero-mean Gaussian random vectors representing uncertainty.

3.2 UAV Action Planning

The landing includes different actions throughout the process. This module aims to facilitate different works at different stages of the mission. At the initial stage of the mission, the UAV is assumed to be hovering above the proximity of the target and waiting for further action. A total of four actions can be done by the UAV during the mission and they are selected based on the state of the UAV at the time:

a. Establish contact with the target

b. Approaching the target

c. Descending to the target

d. Touchdown and Shutdown

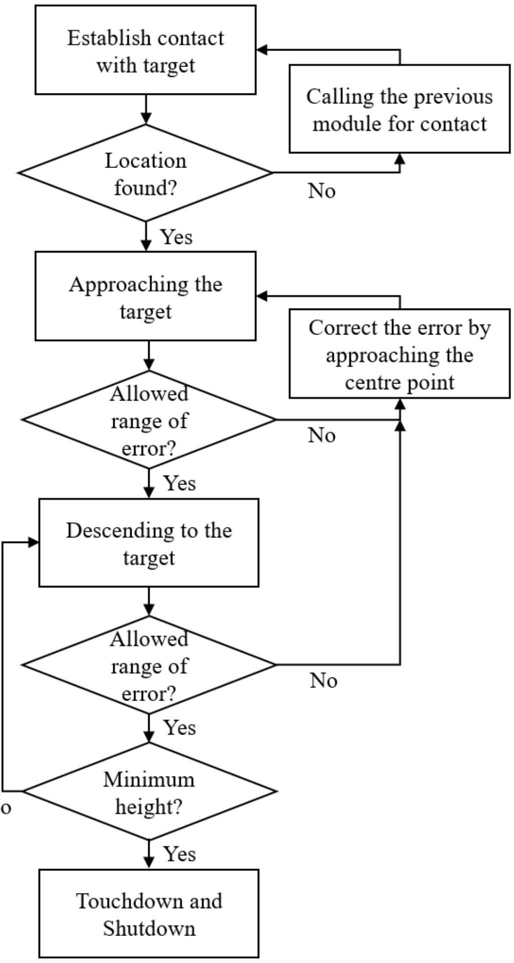


Figure 7. UAV action logic flowchart.

3.2.1 Establish contact with the target

The functionality of the state of establishing contact with the target is to receive the location of the ArUco marker and retrieve the coordinate of it in the camera frame. Then, the orientation and distance from the drone to the marker will be obtained for the approach. When the target marker is lost from the camera frame, the UAV would ascent to a higher altitude to have a greater view with the same FOV so that the marker can be found and continue the process. If not, the UAV may abort the mission and restart the assumed previous mission of GPS guidance for renavigation to the landing platform.

3.2.2 Approaching the target

When the ArUco maker enters the field of view (FOV) of the onboard camera, the UAV would establish contact with the target marker. The UAV would start approaching the marker horizontally. If the maker is not detected in the allowed range of error in the whole landing process, the horizontal approaching process would be run again until the maker appears within the proposed projection area. Once the marker entered the allowed range of error of the camera, which is within a certain incline angle from the camera, the UAV would proceed the descending process.

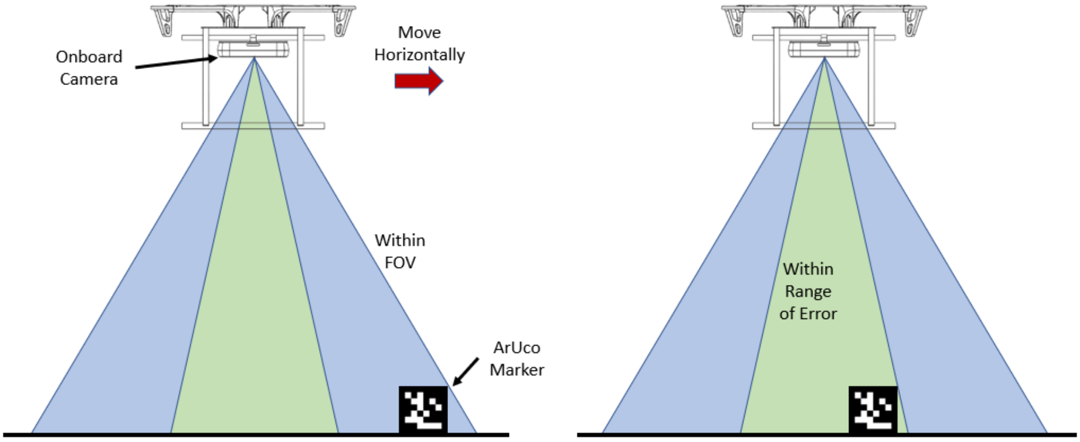


Figure 8. *UAV Approaching Process*

3.2.3 Descending to the target

The descending process of the UAV starts when the ArUco maker enters the allowed range of error. The allowed range of error will decrease with the height to increase the alignment of the landing position and the ArUco maker. During the descending, the maker may deviate from the allowed range. As a result, the approaching process will be run again for the calibration. When the UAV reaches the target minimum height, it will process the touchdown procedure.

3.2.4 Touchdown and Shutdown

The touchdown of the UAV is detected by the flight control unit. When the drone can land on the platform stably, the flight control unit detects that the drone stopped descanting even is continuing the descent controls. Then the flight control units may send a message of mission completion to the action planner and it terminates the mission, at the same time the flight control unit shut down the propellers.

3.3 Hardware Information

For the UAV used in this project, the onboard computer of the quadcopter should be able to process the ArUco maker recognition and trajectory generation. The hardware list of the UAV is listed as follows.

Table 1. List of Hardware

Hardware	Model
Onboard Computer	LattePanda Alpha 864s
Autopilots	Holybro Pixhawk 4 Mini
Motors	SunnySky A2212 980KV
Transmitter	RadioLink R9DS
Buck Regulator	Matek Systems UBEC DUO
Frame	ZMR F450
Onboard Camera	Intel RealSense Depth Camera D455

The camera used in this project is different from a normal photographic camera. This Intel depth camera is designed for depth detection and has high accuracy. The depth error is as low as 2% at a 4-meter distance. The low deviation can help the onboard computer to accurately determine the distance between the camera and the target marker. The accuracy of the autonomous landing thus can be enhanced.

The high landing gear is adopted for the quadcopter. There are two reasons for not using the normal height landing gear. The first reason is our camera are designed to face downward. The high landing gear can prevent causing damage to the camera due to collision. The second reason is to maintain the focus of camera on the target maker. When the camera is too close to the ArUco maker, it would lose focus and would not be able to track the objective marker. This may cause error to the landing. The high landing gear can keep the distance between the camera and the marker. It can ensure the UAV keeps tracking the marker and increase the accuracy of the landing. The assembly concept of the UAV used in this autonomous landing project is shown below.

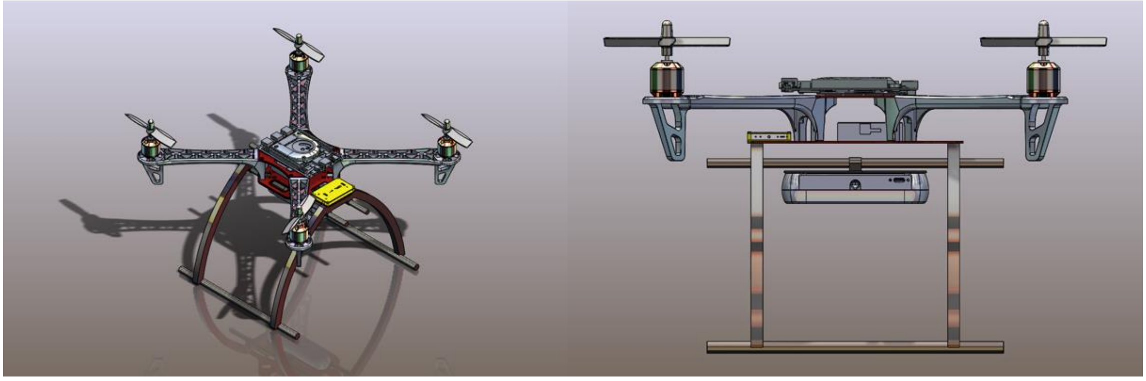


Figure 9. Assembly Concept of the UAV

Chapter 4. Preliminary Results

4.1 Location estimation

Acting as the backbone of the location estimation module, ArUco marker detection is the first one in the line for development. The first accomplishment is a single ArUco detection in space, with its axis being drawn to indicate its x, y, and z-axis. During the first test, the default ArUco dictionary is being used.

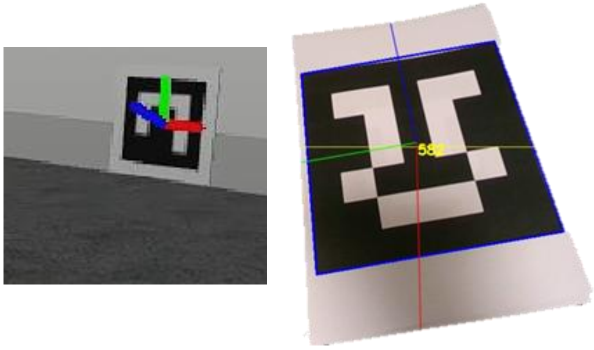


Figure 10. Simple single ArUco marker detection. On the left is a virtual simulation in Gazebo; On the right is a real-world detection using the Intel RealSense Camera.

During the single ArUco marker detection test runs, it is obvious that the larger the marker is, the longer the distance it can be stay detected by the camera. However, when the camera approaches the marker, the marker took up a lot of space in the view. As a result, when the marker moved away from the center, part of the marker would be outside of the camera view angle. Which results in a loss of tracking, or a distortion of the resulted detection data. Either of that is not desirable.

After some research, which is mentioned in the above chapters, it is decided that an E-ArUco marker should be tested for a better close-range performance. At the same time, a different way of reading the result data is being tested in order to be prepared for the upcoming integration between different modules.

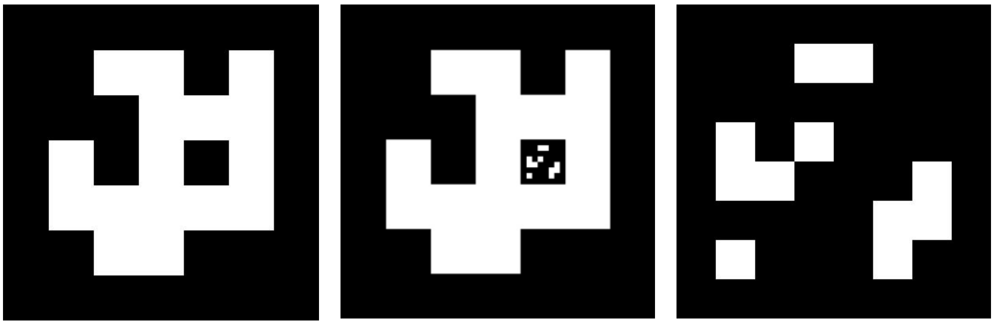


Figure 11. The E-ArUco marker used in the test run. Left: Outer marker (5x5_250, ID:6); Center: Embedded marker; Right: Inner marker (6x6_50, ID:18)

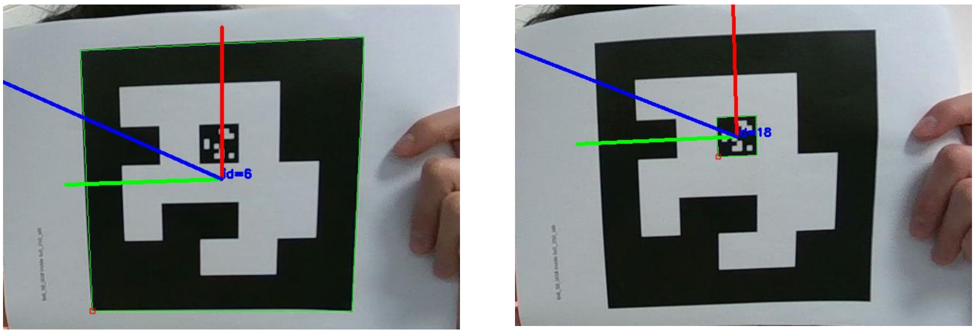
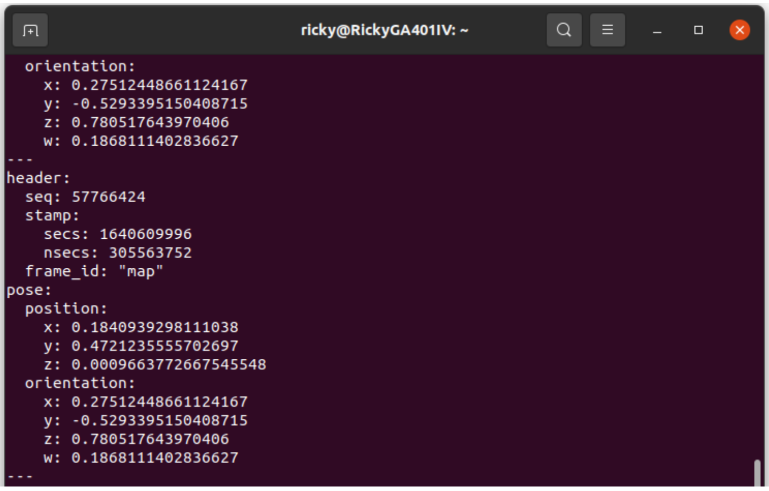


Figure 12. Visualization of the E-ArUco test run



```
  orientation:
    x: 0.27512448661124167
    y: -0.5293395150408715
    z: 0.780517643970406
    w: 0.1868111402836627
---
header:
  seq: 57766424
  stamp:
    secs: 1640609996
    nsecs: 305563752
  frame_id: "map"
pose:
  position:
    x: 0.1840939298111038
    y: 0.4721235555702697
    z: 0.0009663772667545548
  orientation:
    x: 0.27512448661124167
    y: -0.5293395150408715
    z: 0.780517643970406
    w: 0.1868111402836627
---
```

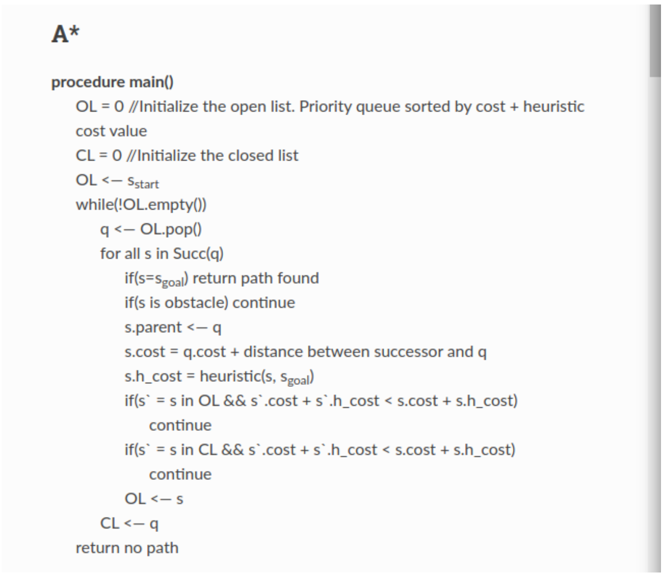
Figure 13. Test run for the rewritten ArUco detection ROS package

During the test run of the rewritten ArUco detection package, the detection of the two markers and the ROS topic publisher are tested with some success. However, there are still optimization and refinement needed for the code. For example, the timing of switching from the large marker to the small marker and reducing the runtime of the program to reduce potential lag. Work would continuously be done on it in order to facilitate the integration with other modules.

4.2 Target approaching

Approaching the target position is the basic technique for the action planning and is required throughout the entire process. Computing a pathway with optimized time efficiency, cost efficiency, energy efficiency, and robustness need an appropriate algorithm as the program to run the planning.

As only 2D movements are involved in the target approaching path, only 2D trajectory generations are needed. The A-star algorithm is introduced which can choose the path having the lowest cost of the aggregate of the actual cost of the actions and the heuristic from the previous node. For the cost of the nodes, it is dependent on the sum of the shortest distance from the start to the current node and the predicted path to the remaining nodes. Thus, the time cost is dependent on the estimation of the future cost on nodes. Following is the logic for the A-star algorithm.



A*

```
procedure main()
    OL = 0 //Initialize the open list. Priority queue sorted by cost + heuristic
    cost value
    CL = 0 //Initialize the closed list
    OL <— $s_{start}$
    while(!OL.empty())
        q <— OL.pop()
        for all s in Succ(q)
            if(s=$s_{goal}$) return path found
            if(s is obstacle) continue
            s.parent <— q
            s.cost = q.cost + distance between successor and q
            s.h_cost = heuristic(s, $s_{goal}$)
            if(s` = s in OL && s`.cost + s`.h_cost < s.cost + s.h_cost)
                continue
            if(s` = s in CL && s`.cost + s`.h_cost < s.cost + s.h_cost)
                continue
            OL <— s
        CL <— q
    return no path
```

Figure 14. Basic Logic for A-star Algorithm.

Then, the algorithm has been implemented. To visualize how the algorithm works, with the aid of ROS tutorials and gits, the algorithm has been formed a package in ROS and pulled into Rviz as below.

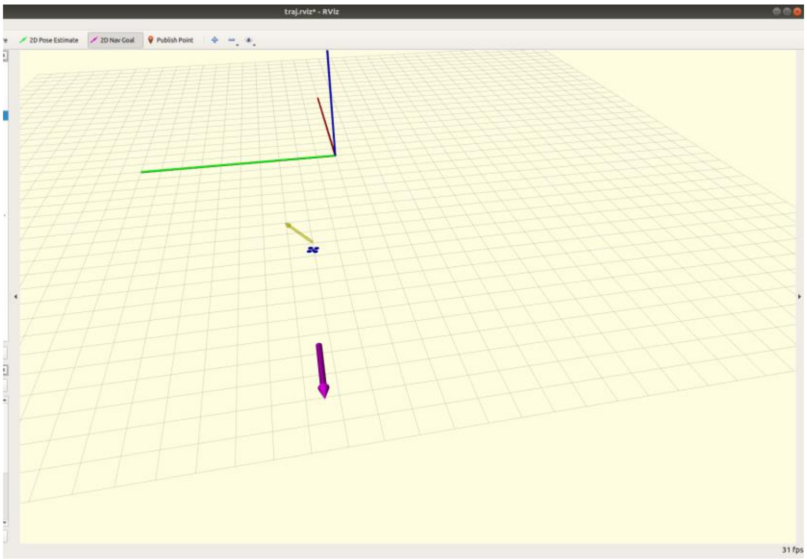
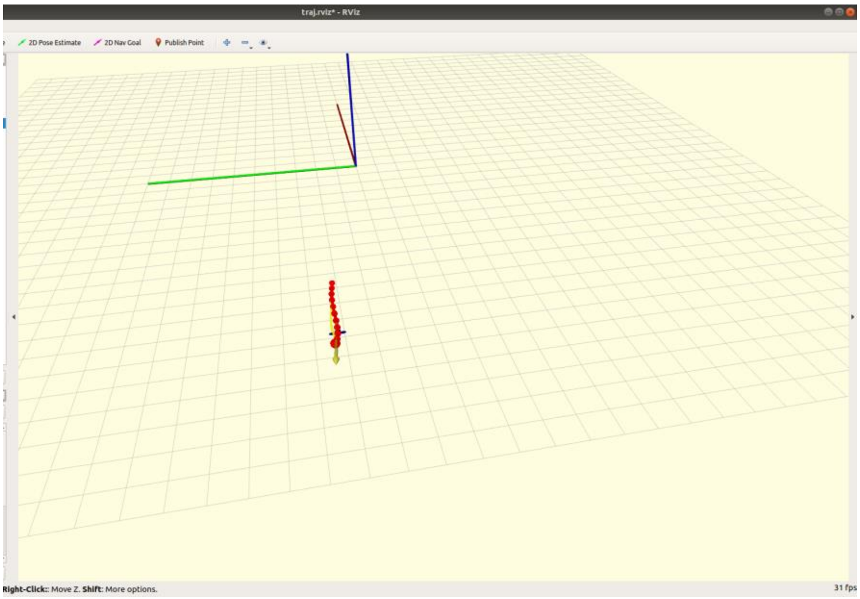


Figure 15. Target point and Direction (Violet Arrow).



Right-Click:: Move Z. **Shift**: More options.

Figure 16. Path Generated (Red Dotted Line).

Referring to the above figure, only a simple pathway can be planned and generated at this stage in which is a static target. It is assumed that the algorithm for object tracking on dynamic objects will be introduced and complied with the developed path planner.

4.3 Virtual testing environment

Before testing in the real environment, all the modules will be tested in the virtual testing environment, i.e., Gazebo. Gazebo provides a real-time 3D simulation environment for robot testing. The autonomous programs and the computer visions can be verified in the software. It also includes a physical engine to simulate gravity, inertia, etc. Gazebo is usually run with ROS. Image recognition, sensor data processing, path planning can be previewed. To simulate different testing environments, customized worlds can be created. In our testing, an ArUco maker acts as the fixed platform, and the quadcopter equipped with a camera act as the UAV used in the project. The algorithm of maker detection, path generation, and autopilots will be tested in the testing world.

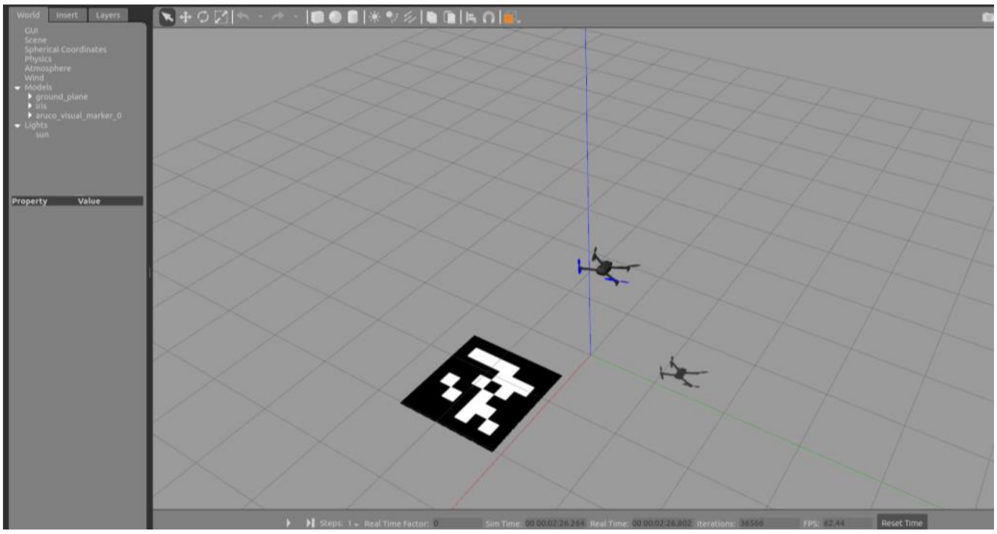


Figure 17. Testing Environment.

In the first stage of simulation, a fixed platform is tested for the basis of the autonomous landing of the UAV. When the algorithm is verified as stable and reliable, the second stage of the simulation will bring the mobile platform into the environment. The ArUco maker will be put on the ground vehicle and an improved algorithm will be generated for the simulation. After the simulation, the testing will be brought to the real environment.

Chapter 5. Project Planning

5.1 *Project modules division*

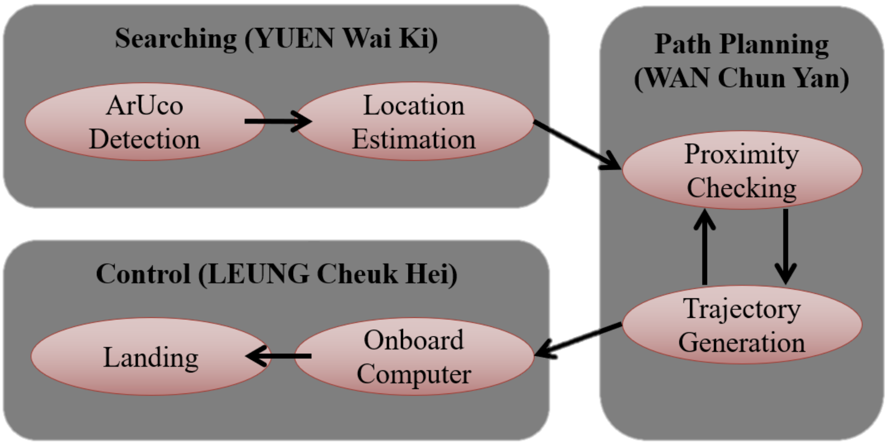
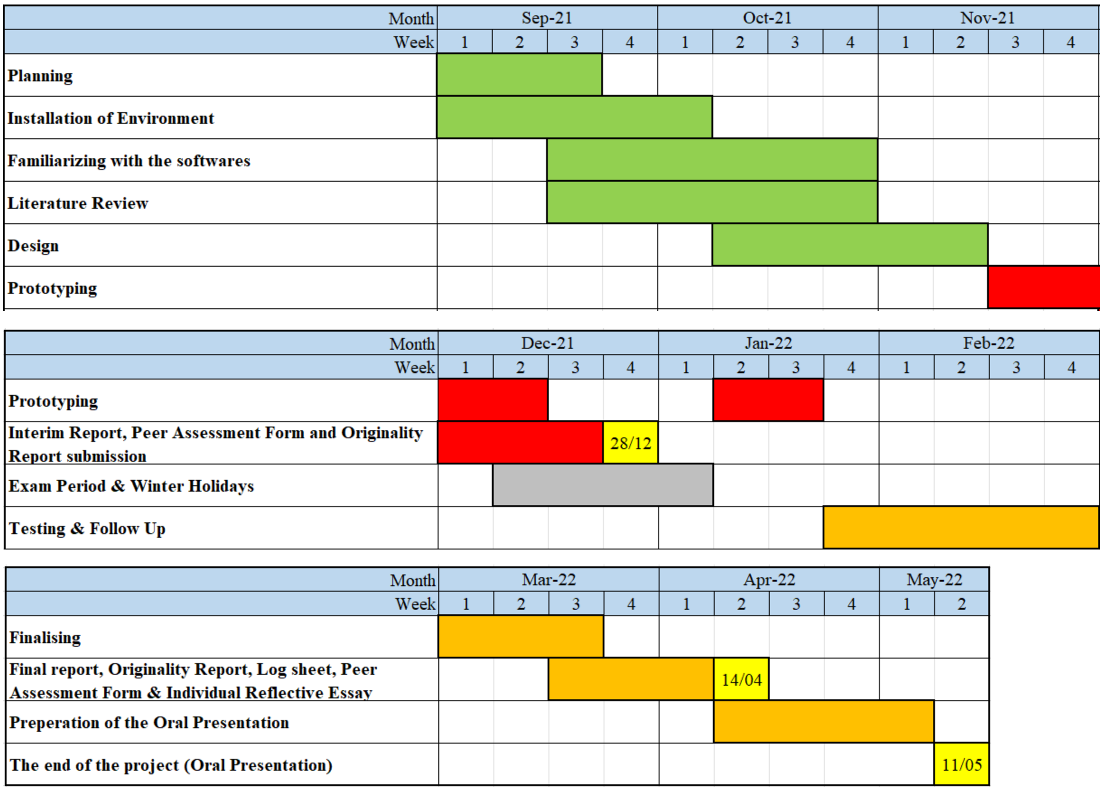


Figure 18. Division of modules for group members. Works are done in parallel.

5.2 *Gantt chart*



Month	Sep-21				Oct-21				Nov-21			
Week	1	2	3	4	1	2	3	4	1	2	3	4
Planning	🟩	🟩	🟩									
Installation of Environment	🟩	🟩	🟩	🟩	🟩							
Familiarizing with the softwares			🟩	🟩	🟩	🟩	🟩	🟩				
Literature Review			🟩	🟩	🟩	🟩	🟩	🟩				
Design						🟩	🟩	🟩	🟩	🟩		
Prototyping											🟥	🟥

Month	Dec-21				Jan-22				Feb-22			
Week	1	2	3	4	1	2	3	4	1	2	3	4
Prototyping	🟥	🟥				🟥	🟥					
Interim Report, Peer Assessment Form and Originality Report submission	🟥	🟥	🟥	28/12								
Exam Period & Winter Holidays		▨	▨	▨	▨							
Testing & Follow Up								🟧	🟧	🟧	🟧	🟧

Month	Mar-22				Apr-22				May-22	
Week	1	2	3	4	1	2	3	4	1	2
Finalising	🟧	🟧	🟧							
Final report, Originality Report, Log sheet, Peer Assessment Form & Individual Reflective Essay			🟧	🟧	🟧	14/04				
Preperation of the Oral Presentation						🟧	🟧	🟧	🟧	
The end of the project (Oral Presentation)										11/05

Figure 19. Gantt Chart of the project. Green: Done; Red: In progress; Orange: Planned; Yellow: Important Dates.

References

Alvika, G., Sujit, P. B., & Srikanth, S. (2014). *A survey of autonomous landing techniques for UAVs*. https://doi.org/10.1109/ICUAS.2014.6842377

Artur, K., Aufar, Z., Tatyana, T., Mikhail, S., & Evgeni, M. (2021). Embedded ArUco: a novel approach for high precision UAV landing.

Girard, A. R., Howell, A. S., & Hedrick, J. K. (2004, 14-17 Dec. 2004). Border patrol and surveillance missions using multiple unmanned air vehicles. 2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601),

Greg, W., & Gary, B. (1995). An introduction to the Kalman filter.

Herwitz, S. R., Johnson, L. F., Dunagan, S. E., Higgins, R. G., Sullivan, D. V., Zheng, J., Lobitz, B. M., Leung, J. G., Gallmeyer, B. A., Aoyagi, M., Slye, R. E., & Brass, J. A. (2004). Imaging from an unmanned aerial vehicle: agricultural surveillance and decision support. *Computers and Electronics in Agriculture*, *44*(1), 49-61. https://doi.org/https://doi.org/10.1016/j.compag.2004.02.006

Jiang, Z. (2019). *An Autonomous Landing and Charging System for Drones* Massachusetts Institute of Technology].

Miguel-A., O.-M., Iván-F., M., & Pascual, C. (2013). Autonomous Landing of an Unmanned Aerial Vehicle using Image-Based Fuzzy Control. *IFAC Proceedings Volumes*, *46*(30), 79-86. https://doi.org/https://doi.org/10.3182/20131120-3-FR-4045.00011

Moletta, M. (2020). Path planning for autonomous aerial robots in unknown underground zones optimized for vertical tunnels exploration. In.

Nagai, M., Chen, T., Shibasaki, R., Kumagai, H., & Ahmed, A. (2009). UAV-Borne 3-D Mapping System by Multisensor Integration. *IEEE Transactions on Geoscience and Remote Sensing*, *47*(3), 701-708. https://doi.org/10.1109/TGRS.2008.2010314

Sanchez-Lopez, J. L., Wang, M., Olivares-Mendez, M. A., Molina, M., & Voos, H. (2019). A Real-Time 3D Path Planning Solution for Collision-Free Navigation of Multirotor Aerial Robots in Dynamic Environments. *Journal of Intelligent & Robotic Systems*, *93*(1), 33-53. https://doi.org/10.1007/s10846-018-0809-5

Waharte, S., & Trigoni, N. (2010, 6-7 Sept. 2010). Supporting Search and Rescue Operations with UAVs. 2010 International Conference on Emerging Security Technologies.