Department of Computer Science

Submitted in part fulfilment for the degree of BSc.

Autonomous Landing of a UAV on a UGV for Extending Mission Flight Time

Alistair Foggin

2 May 2024

Supervisor: Dr. Alan Millard

Acknowledgements

I would like to give special thanks to my supervisor Alan Millard for all the help and support he provided throughout this project.

Contents

List of Figures

List of Tables

Executive Summary

The aim of this project was to begin investigating the potential of inter-robot collaboration between Unmanned Aerial Vehicles (UAVs) and Unmanned Ground Vehicles (UGVs) in extending mission flight time. In particular, this was done by developing an algorithm to land a quad-copter UAV on a simplified representation of the Clearpath Husky UGV in simulation and then testing the effectiveness in various environmental conditions to determine the range of optimal flight conditions. The motivation behind this precision landing is that the primary limiting factor in autonomous UAV missions is battery life. When UAVs are running low on power, the usual method is to fly back to the original takeoff location to land and recharge before taking off again to continue the mission. This effectively halves the range that a mission could cover. The motivation for the precision landing algorithm is to use the UGV as a portable recharging station for the UAV to land on so that it does not have to return as far to recharge before continuing the mission.

There are a few primary components of the algorithm that were developed. The first is controlling the UAV to navigate to the GPS coordinates of the UGV. GPS waypoint navigation is a very straightforward problem that has an implemented solution in the vast majority if not all flight control systems and hence is not the priority of this project. The primary issue here is rather communication with the UGV to determine those GPS coordinates. Across large regions, there may be limited connectivity, but by using the last known location, the UAV will end up flying closer and connectivity should be restored to determine more accurate coordinates.

The next issue is the precision landing aspect, which was the primary focus of development in this project. The foundation of precision landing is pose estimation which is determining the position and orientation of the UAV and landing target on the UGV relative to each other, primarily through computer vision. There are several approaches that use different types of cameras or multiple cameras either on the UAV or on the target. The approach taken here was to use a standard monocular camera on the UAV with a fiducial marker (distinct black and white markers that are easily identifiable) on the target. There is a fundamental issue in using a single fiducial marker with a single camera for pose estimation which causes the wrong pose to occasionally be calculated, so the markers were used to identify when the

UAV could see the target, then the true UAV position from the simulator was used with some artificial noise to emulate the calculations.

The flight control part of the system used the pose estimate to calculate the target velocity. This velocity then was used in addition to the offset from the target centre to calculate the necessary velocity to send to the PX4 flight controller which would control the individual rotors of the UAV. All this was done in the Gazebo Garden simulator using the Robot Operating System as the communication system between components.

To test the effectiveness and limits of this algorithm, many experiments were carried out looking at specific variables that could affect the flight of the UAV. The variables tested were the magnitude of pose estimation errors, the speed of the UGV, the mean wind speed, and the starting altitude of the UAV. Each variable was tested against a range of values, with each value having multiple runs of the experiment. Each configuration of a single variable was compared against the other samples and against a baseline using a variety of statistical tests to determine significant differences. A combination of all the variables based on reasonable values found in the previous experiments was also tested.

The results of these experiments show that the algorithm is effective up to where the pose estimation error has a mean of 4cm and a standard deviation of 3cm. The target can move at 1m/s corresponding to the Clearpath Husky's max speed, and the mean wind speed is 10m/s. This was tested at an altitude of 14m, but a large range of other altitudes may also be possible.

Fundamentally, this project has developed a simple landing algorithm and shown that it is effective in a large range of environmental conditions that can be used with a moving UGV, laying the foundations for the portable charging station to be mounted which can extend mission flight times. The primary further research that is necessary is that of pose estimation, and there are several possible methods that can be investigated.

There were no legal, social, ethical, professional or commercial issues in this project. All development and testing has been done within simulation to avoid safety issues, and there were no human participants. All external work used was cited and the project was purely for furthering research with no commercial investment.

1 Introduction

The use of Unmanned Aerial Vehicles (UAVs) has been increasingly widespread in the public and commercial sectors over the last few years expanding into many fields including mapping, agriculture, and infrastructure inspection to mention a few. The technologies that go into UAV flight are increasingly capable with regards to range, speed, camera resolution and more. However, a major limiter is still battery life, particularly for multi-copter UAVs, which leads to reduced flight times and limits the scope of missions that UAVs would be used for. Unmanned Ground Vehicles (UGVs) are also an active field of research in running autonomous missions, but they have their own limitations in manoeuvrability and types of mission.

There is potential for missions that involve collaboration between multiple types of robots, and that is what this project is starting to explore. In particular, investigating the use of a UGV as a target for the UAV to land on as it could easily be turned into a portable charging station. This project will explore the intricacies of collaboration between the UAV and the UGV in developing and testing a precision landing algorithm. This will involve communication, position estimation and movement controllers, etc. As this is a very unconstrained problem in which many environmental variables and varying sensor noise which all affect the way the UAV will fly, the testing will be used to determine some reasonable limits of these conditions in which the algorithm will reliably, accurately, and precisely land on the UGV. Any additional limitations on capabilities will be discussed in detail. The goals of how the project will investigate the UAV-UGV collaboration and communication in relation to precision landing are summarised as follows:

1. Set up a simulation environment
2. Select a pose estimation method
3. Develop an algorithm to land the UAV on the UGV
4. Determine the optimal range of environmental conditions in which the algorithm is successful in landing the UAV on the UGV
5. Identify key areas to be researched further

2 Literature Review

2.1 Overview of Robotics

Robotics is a broad reaching field that is rapidly progressing, particularly in recent years. Initially in the 1960s, industrial robots were developed for factory work [1] but the field has since developed into three key areas: manipulators [2], mobile robots [3], and biology-inspired robots. The development has been driven by human need, but has been expanding both scope of capability and variability of environment. Of particular interest is the field of mobile robotics and how research into UAVs and UGVs has developed.

2.1.1 Unmanned Aerial Vehicles

UAVs have a lot of potential in autonomous missions in a large range of fields. Initial development was in military applications, but now a lot of development and research is for civilian purposes including public safety, disaster relief, remote sensing, planetary exploration and intelligent agriculture [4]. UAVs vary greatly in both size and methods of flight. These range from having fixed wings with flaps like an aeroplane to having multi-copters which use multiple propellers on arms for vertical lift. Each has their own advantages in manoeuvrability and range, where fixed wings can easily fly long distances but must have continuous forward motion where multi-copters have very precise motion control, but cannot reach the same distances.

2.1.2 Unmanned Ground Vehicles

UGVs vary in also cover a large range of potential missions. NASA's Perseverance rover [5] for exploring the planet Mars is a prime example of this. There are many different types of of UGV but wheeled UGVs are most common as they are very simple to make [3]. The Clearpath Husky [6] is an example of one of these designed as a research platform. Legged robots are now coming into use with the now ubiquitous Boston Dynamics Spot robot [7].

2.1.3 Specific Applications

The many fascinating applications of robotics include agricultural automation, disaster relief and planetary exploration to name a few. Robots are able to do things that are difficult or even impossible for humans whether that is very precise movements, going into dangerous or unreachable areas, or doing repetitive manual labour.

As farms are growing larger to accommodate the growing world population, it is increasingly difficult for farmers to keep up with demand, so there has been and continues to be much research in robotics to assist in ways ranging from automated planting and harvesting, irrigation, and monitoring of plant health [8]. Collaboration between UAVs and UGVs has also received much interest in precision farming [9] and more generally, path planning [10]. Multi-robot systems, particularly with multiple types of robot provide a much larger scope of potential mission, and also provide increased performance and reliability over single-robot systems [11]. The major factors that are involved in development of such systems are collaboration, communication, and mapping and localisation [12].

Research into the use of UAVs in disaster relief has been starting in recent years investigating how they can be used in different types of disaster in preparation, assessment and response [13]. While some research has been done and UAVs will be very beneficial, there is still a lot more research left to do [14]. As humanity is looking to explore space, robotics is essential to explore planets and moons as it is very challenging and dangerous to send people. NASA's latest efforts in exploring mars using the Perseverance rover and the Ingenuity helicopter [15] clearly demonstrate how robots can benefit research and exploration beyond our planet as well as how UGVs and UAVs can work together in challenging environments.

2.2 UAV Energy Requirements

Battery life and energy usage is a major consideration when it comes to electric powered UAVs. As they get heavier, they need more battery capacity to be able to fly for longer. However, as batteries with larger capacity are put on UAVs, they inherently have more weight added which means it is a balance between how much benefit the additional capacity provides compared to how much it hinders the flight. Overall, work is done in the development of UAVs to make them lighter, and more energy efficient, but that can only go so far, so replenishing the battery is a necessity for longer missions. Three possible methods have been proposed by various researchers which are battery replacement, wireless charging, or solar power [4]. Solar power is primarily for for fixed wing UAVs and has a large set of challenges of its own to provide enough power to keep flying for an extended period of time. One major work that was done to swap batteries used a motion tracking system to manage the landing and positioning, but demonstrated an effective system to enable a UAV to keep flying [16]. However, the use of external tracking systems limits the scope of mission range.

A couple more recent works have done similar methods of battery replacement by having stations for the UAV to land [17][18]. Other works have developed systems of wireless charging by induction [17][19]. All of these methods do help enable UAVs to have extended mission flight time, but require precision landing, and are stationary charging and replacement stations. There is another proposal of wireless power transfer to UAVs during their flight while inspecting power lines [20]. Experiments were being conducted on a small scale on whether charge could be received from the electromagnetic field around the power lines. This research could be taken further, but still limits UAVs to be near power lines to charge.

2.3 Localisation

A key necessity for mobile robots is to have an understanding of the environment and how they fit into it. In outdoor environments, GPS systems provides generally accurate location information. However this is only accurate to a few metres [21]. Real-Time Kinematic GPS (RTK-GPS) [22] provides major improvements to this. Other methods of localisation include Simultaneous Localisation and Mapping (SLAM) and Fiducial markers for pose estimation.

2.3.1 Simultaneous Localisation and Mapping (SLAM)

A major part of safety is obstacle avoidance and path planning. Simultaneous Localisation and Mapping (SLAM) is a part of robotics that takes on the task of both building up a representation of the environment that the robot can understand, while also identifying where in the environment the robot is located. For UAVs, this is a bigger challenge than it is for

UGVs as there is the additional dimension of altitude. The main way in which this can be done is through vSLAM (Visual SLAM) of which there are many proposed methods, a lot of which are tested in the application of landing a UAVs [23]. Two major ones that were investigated further were ORB-SLAM3 [24] and OV^2-SLAM [25]. ORB-SLAM3 was generally more accurate while computationally slower than OV^2-SLAM. The reduced computational cost of OV^2-SLAM was deemed less important than the better accuracy of ORB-SLAM3. These techniques all produce point clouds (a collection small environmental landmarks and their associated position) and use them for localisation for the UAV which can then be used in path planning. However, this is for the fine detail navigation or for when GPS is unavailable. When doing large scale UAV missions, GPS should be used for the global plan while vSLAM and other obstacle avoidance methods should be used on a smaller scale.

2.3.2 Fiducial Markers

Another method of localisation is through the use of fiducial markers which are artificial landmarks that are easily recognisable to a camera system. A systematic review of existing markers was carried out by Kalitzakis et al. [26]. The authors identified a large number of markers that had developed over the years, then selected four key ones to thoroughly test. The markers tested were the ARTag, AprilTag [27], ArUco [28], and STag [29]. The experiments looked into the detection rates and the mean error in the calculations with different camera conditions, particularly looking at how well each one deals with various distances, viewing angles and motion blur. ARTag had the lowest computational requirements but had very unreliable detection rates. The other three all were able to reliably produce accurate results, although motion blur does cause significant errors. They can all be used to estimate the pose of the camera through co-planar features [30]. The general way to do these calculations is to compute a homography (a matrix to transform points on one plane to matching points on another plane) between points on the marker and the points on the image. The rotation matrix and translation vector of the camera can then be extracted from that, giving the camera pose.

Other tags were have also been developed to the range of distances from which it can be detected. Krogius et al. [31] used AprilTags as the base to develop flexible markers of different arrangements of black and white squares. Of particular interest is their recursive marker that places smaller tags inside of larger ones as that enables detection from a wider range of distances. The Fractal ArUco marker [32] was developed in a very similar way but based on the ArUco tag instead of AprilTag. They both have their own advantages. Both have more robust detection algorithms. The Fractal

ArUco tag has a particularly good advantage of being resilient to partial occlusion.

There is an important issue to be aware of with regards to planar pose estimation when using a single fiducial marker which occurs regardless of algorithm when calculating the extrinsic parameters (rotation and translation) of a camera using co-planar points (all of the 3D points lie within the same 2D plane). At certain viewing angles, a different rotation matrix and position may be calculated that is incorrect, but still has a low error when projecting the points back onto the image. The reason for this ambiguity is explained by Schweighofer and Pinz [33] by showing that the error functions used for pose estimation have two minima. As the distance between the camera and the marker increases, so does the ambiguity as the lack of perspective leads to difficulty in estimating the rotation of the camera. Image noise is another limiting factor that creates instability in calculating an accurate pose.

2.4 Precision Landing

There has also been a decent amount of research into precision landing, both on stationary targets and on moving targets. Wang et al [34] used AprilTags [27] to design a landing pad that can be detected at different scales and distances before designing an algorithm to land on the target. Testing showed an average error of 10cm from the target centre. Similar methods have been taken up by other researchers in this area. Zhu et al [35] developed a series of increasingly small black and white squares within each other which enable positional landing, but there is limited orientation information. This target was then put on a car in simulation to test a landing algorithm. It was was also tested in the real world and the maximum error was 20cm. Another paper explored the challenges of landing a UAV out at sea for marine operations [36] which used a combination of ArUco tags and lights on the target to help identify the target in varying visibility. This has the added challenge of a changing orientation of target due to the waves in the ocean, but managed to successfully achieve an average landing error of 8.18cm with a maximum error of 27.73cm. All of these papers have very similar methodologies with varying types of targets and similar algorithms for landing. They provide a great base upon which this project can investigate landing on a UGV which can have movement and has potential to go over rough ground causing unexpected orientations of the target. The target can also be selected with the advantages and shortcomings described in previous research.

3 Design and Implementation

3.1 Problem Formulation

The problem being addressed by this project is to research the challenge of precisely landing a UAV on an UGV as a stepping stone to investigating the potential of using a portable UAV charging station mounted on a UGV, to extend UAV flight time for autonomous missions. Such missions include detailed mapping, search and rescue, and remote sensing to mention a few [4]. This challenge can be broken down into individual chunks that can be individually addressed and broken down further.

3.1.1 Communication

When multiple autonomous robots are collaborating, communication is a major factor that must be taken into account as unknown environments may have limited connectivity. Some communication is necessary, but when distances are large, that communication should be minimal to help ensure that messages that are sent between the robots are properly received. When the UAV and UGV are closer together, more communication is possible. This project however still aims to keep communication minimal while still enabling collaboration. Alternative options are discussed which may use more communication but are still feasible.

3.1.2 Pose Estimation

The next challenge is pose estimation. This involves exploring the methods of calculating where the UAV and UGV are in relation to each other in space. This is a key piece of information that is used in the landing algorithm. Usage of fiducial markers for this has had extensive research [26] as discussed previously, but still has limitations with an unconstrained camera [33]. Some of these are explored below, along with some proposed alternative methods of pose estimation.

3.1.3 Control System

The final piece of the puzzle is the actual control of the UAV to land on the target. This should take into account the state of the UAV in the environment, tackling its position and speed in the environment, dealing with environmental factors that affect flight such as wind speed and direction, and dealing with the difficulties of a moving target. All of these challenges are explored with their possible solutions along with the implemented solution that is tested against these different factors.

3.2 Simulation

To ensure safety in the development process of any autonomous robotic system, a simulator must be used. PX4 [37], the UAV flight control system being used, supports several different simulators including Gazebo [38] (formerly known as Ignition Gazebo), Gazebo Classic, jMAVSim and more, each of which has its own advantages and disadvantages. For the latest release of PX4 (1.14) the two recommended simulators are Gazebo Garden and Gazebo Classic, and the former specifically for ROS2 Humble. The advantage of using Gazebo over other simulators designed specifically for UAVs is that Gazebo has the flexibility to incorporate other types of robot.

3.2.1 UAV

To be able to tackle the challenge of any UAV control system, a standard system should be selected to both build on other research and to enable further research to build on this project. In particular, PX4 [37] was used as the UAV flight stack as it is one of the major open source control system, and the Robot Operating System 2 (ROS2 Humble [39]) was used as it is the industry standard for robotics research. PX4 provides a wide variety of automated procedures for landing, navigation and missions, while also provided precise control of position and velocity both in local coordinates and in global GPS coordinates. A communication interface is provided to enable integration with the ROS2 which provides a layer of communication standards to enable communication between components on the same robot and between different robots collaborating in the same network.

3.2.2 UGV

The Clearpath Husky [6] is a potential UGV to be used as a landing target. To create a UGV as the landing target, an SDF file [40] (Simulation Description Format which is used by Gazebo to describe scenes and robots) was used which contained an abstract representation of the Husky. The UAV should be able to identify the target location when in sight, and be able to request the target UGV GPS location from a distance. The UGV representation consisted of a block to roughly match the shape along with the landing target on top which is a visual marker for the UAV to spot. The marker could easily be swapped out to test different marker types. To implement the movement, a plugin was created which can be set the velocity of the target which can vary over time and is used to test the capabilities of the UAV's detection and landing algorithm. Were the system to be extended to work outside of the simulator, a marker or target could easily be mounted to the Husky with limited difference to the landing process.

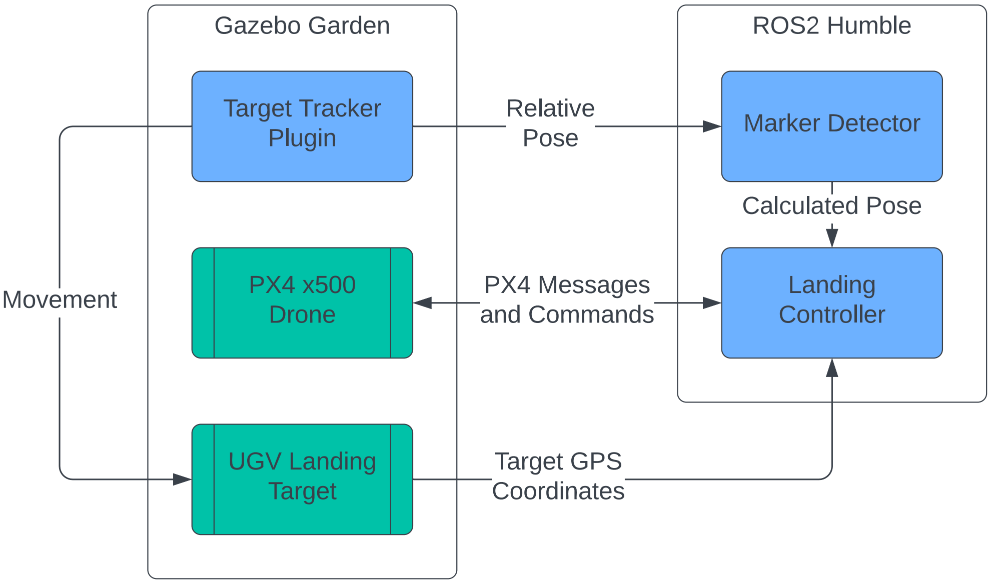


Figure 3.1: The architecture of the system where the plain blue blocks are the developed components, and the turquoise blocks correspond to the simulated UAV and UGV

3.3 Target Detection

The main challenge in precision landing is identifying where the target is relative to the UAV. Usage of GPS sensors on both the UAV and the UGV provide general coordinates which the UAV can use for initial navigation. However, once the UAV is within a certain range of the target, it can no longer rely on GPS to provide the necessary accuracy [21]. The simplest method that is commonly used in robotics and computer vision is the use of fiducial markers as discussed in the literature review. For the purposes of this project, ArUco [28][41] tags were initially used as the majority of the library was included in OpenCV [42], the Computer Vision library which was used due to it being the industry standard. Using the points of the located fiducial markers, there are algorithms to calculate the pose of the camera relative to the target.

3.3.1 Simulator Plugin

As the system is being developed in the simulator, it is possible to extract the pose information as the ground truth. To do this, the Target Tracker Gazebo plugin in figure 3.1 was made which takes the position of the target and the position of the UAV and publishes the relative pose of the UAV. The plugin also contains the capabilities to control the velocity of the target to enable testing of the landing algorithm against different variations of a moving target.

The use of the published pose of the UAV is two-fold. One is for logging pose throughout the landing process which can be used in the analysis of the system. The other purpose is for use by the landing algorithm instead of a calculated pose. Artificial noise can be added to emulate the small errors in results that may arise from various estimation algorithms. An additional limitation to help emulate vision based pose estimation algorithms is to only publish when the target is visible to the camera. To do this, there is a marker detector node (an executable that can communicate through the various ROS communication methods) that reads the true location from the Gazebo plugin, and monitors the camera feed to detect the marker. When the marker is visible and is detected, the pose is then published on a new topic which the landing controller then receives. This provides the baseline to develop and test the landing algorithm.

3.3.2 Fiducial Markers

One of the most straightforward fiducial markers to use for the target is a plain ArUco tag as shown in figure 3.2. It is loaded into the target SDF file to be put in Gazebo. The detection algorithm is already implemented into OpenCV which both locates the bounds within the image and calculates the 3D pose of the camera. However, there are limits to this marker. Specifically in how well it deals with occlusion since it is no longer identifiable if any part of the marker is hidden from the camera. This is an issue because

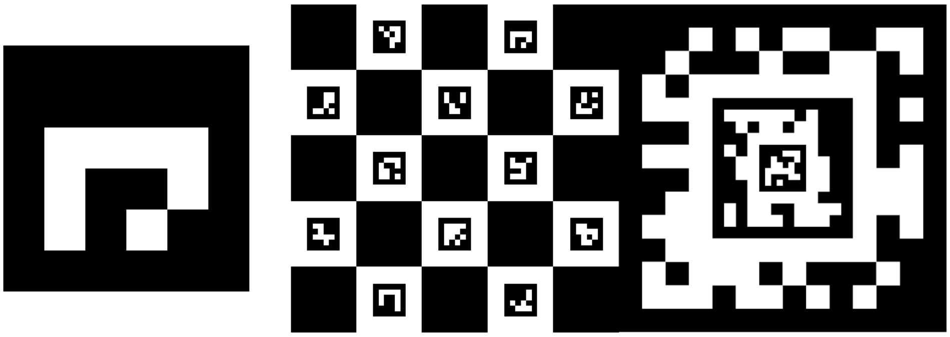


Figure 3.2: The three types of fiducial markers used in the project. The regular ArUco tag on the left, the ChArUco board in the centre, and the Fractal ArUco marker on the right.

it will disappear when the UAV gets close to land. Also, as the UAV flies towards it, it tilts in the direction it flies which angles the camera away from the target, inhibiting the detection. Part of this latter problem can be dealt with by putting the camera on a gimbal as done here [34]. There are other markers that are designed to deal with some of these issues.

One of these other markers is the ChArUco board which is technically a combination of markers. It is a black and white chequerboard with smaller ArUco tags in every white square. The computer vision algorithm can then use this arrangement of markers to make estimations for where the rest of the board could be when occluded if only a few of the smaller ArUco tags are visible. Its primary use is camera calibration but it can use the projected points along with the visible points to make a pose estimation.

The third type of marker used is a Fractal ArUco marker [32] which builds on research into previous fiducial markers. From the outside in, it has decreasing sizes of black and white squares arranged to be tracked from a much larger range of positions and to deal with occlusion. It has the same maximum range of detection as a standard ArUco marker, but the camera can get much closer so long as it can see a certain proportion of the inner corners. This can then be used to project the estimated locations of hidden vertices and from there calculate the camera pose. In the end the final method used in testing was to have this marker be located within the image, and if it is located, then the marker detector node would publish the pose from the Gazebo plugin due to the limitations of planar pose estimation discussed previously.

3.4 UAV Control Algorithm Overview

Once the building blocks were in place, the algorithm could then be developed. As landing is only part of any autonomous mission, there should exist a framework in which it can link to any generic mission. To do this, a simple state machine was created to identify key stages that are relevant to landing. First, the UAV must take off, which in this case is controlled by my system, but in any real-world mission would be done by the mission planner.

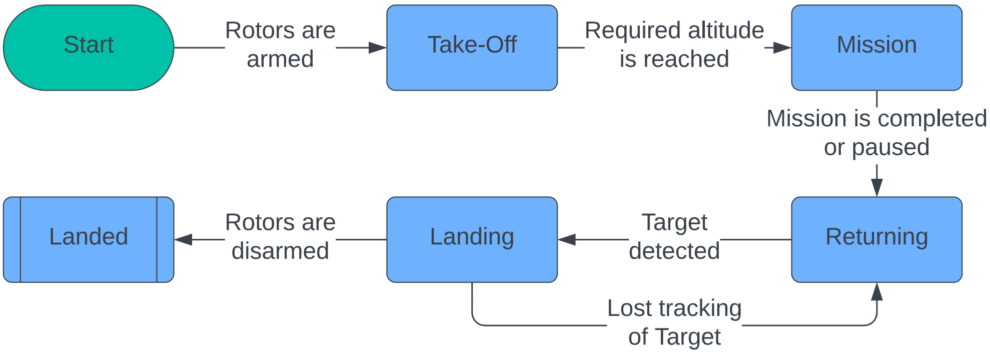


Figure 3.3: State machine explaining the process from take-off to landing of how the system works.

The system starts in the take-off state in which it tells the UAV to rise to an altitude of ten metres (the specific altitude can be adjusted if necessary) before then switching to the mission state where it will carry out the mission specifics. A sample mission that was used in testing was to orbit a point near the take-off location for an arbitrary amount of time (two seconds for testing purposes). The primary focus of the landing starts once the mission is completed or paused due to a necessity to land and recharge. At this new returning stage, the UAV must navigate to the landing latitude and longitude coordinates. In standard consumer UAV systems, the home waypoint usually corresponds to the take-off location. However, in this system the UAV needs to navigate to the UGV which is likely to have moved since take-off.

Once the UAV is close enough to receive or calculate its pose relative to the landing target on the UGV, it switches to the landing state where the system sends a continuous stream of messages of target velocity to the UAV flight controller as explained below. Once the UAV detects the system has landed, it disarms the motors and switches to the landed state. Once recharged the UAV can repeat the process. It should also be noted that during the descent, if the UAV loses sight of the target for an extended period of time while still over a metre above the target, it will revert to navigating to the UGV's GPS coordinates until it can see the target again.

It is also important to note that in the simplified simulation, the coordinates are x and y offsets from the takeoff location rather than GPS coordinates as the focus of the algorithm is on the landing state rather than the returning state.

3.5 System Architecture and Coordinate Systems

Three main components were developed to run the system in the Gazebo simulator. There is a stand-alone Gazebo Plugin which controls the movement of the target. It also provides the location of the UAV relative to the target, which it publishes on a ROS2 topic. The origin of the coordinate system is centred on the target, while the XYZ axes correspond to ENU (East, North, and Up) corresponding to the standard Gazebo and ROS2 coordinate system [43].

The next component, which is more directly involved in the autonomous system developed in this project is the marker detector. It is a ROS2 node that receives the UAV pose from the aforementioned plugin and processes the incoming images from the downward facing UAV camera. With every message received containing the UAV location from the plugin, it converts the coordinate system from ENU to NED (North, East, and Down) to match with the PX4 global coordinate system. To convert from ENU to NED, there is a simple transformation explained in the PX4 documentation [37] on the ROS2 User Guide page. It is a $\frac{\pi}{2}$ rotation around the z axis followed by a π rotation around the x axis.

The marker detector's primary job is to provide the relative location to the landing controller. The method that was used for testing was to publish the location from the simulator only when the Fractal ArUco marker was visible and being detected. The code is in place to calculate the position of the UAV from the marker, but due to the limitations mentioned previously, it was impractical to use and test. The pose is then published to a new ROS2 topic which is subscribed to by the landing controller.

The landing controller is where the autonomy of the system is defined, both in the state machine described above, and in the specific controls below. It receives the state of the UAV which includes key information such as the velocity and orientation. It then uses that, along with the UGV coordinates and the relative pose between the UAV and the target to drive the autonomous control signals which are sent to the flight controller within PX4.

The control messages are sent out on ROS2 topics which are then received by the uXRCE-DDS agent which facilitates transferring messages to the uXRCE-DDS client which converts them to uORB messages (the publisher/subscriber messaging system used by PX4). The uORB messages are received and processed by the PX4 flight controller according to the MAVLink messaging protocol designed for controlling UAVs. The first command is to arm the motors in preparation to take off and the next message sent on startup is to switch to off-board control, which enables the precise control to come from the landing controller as opposed to sending general commands to have the flight controller itself decide the precise commands. This is followed by either position or velocity commands.

3.6 Return Navigation

Navigating to GPS coordinates is a relatively solved problem which most autopilot systems such as PX4 already have implemented. For the sake of simplicity in the project implementation, this is done by passing the local position relative to the world origin to PX4 in place of real GPS coordinates.

This is where part of the collaboration comes in where the UGV publishes its coordinates on a ROS2 topic to which the UAV will be subscribed. In a real system, this would be actual GPS coordinates rather than the simplified location. The main issue with communication is that at great distances, the connection may be lost. However, a consistent stream is not needed as the UAV can still navigate to the last known location of the UGV. By navigating to the last known location, the UAV will end up closer to the UGV especially since the UAV is likely to be faster. Once closer, the new location should be picked up, enabling it to navigate even closer. This is not an entirely foolproof system, so further research is possible, and should be carried out to implement fail-safes. However, this system works on the assumption that the UAV can keep sufficient communication to navigate within range before carrying out the precision landing.

3.7 Precision Landing

Once the UAV is in range to be able to calculate or receive its pose relative to the target, it starts the precise navigational control. The initial component of the navigation is to calculate the vector from the UAV to the target which is just the relative pose reversed. Then using the target pose where the x and y components are the distances North and East respectively, the horizontal velocity is just the distances scaled by some speed factor c which can be tuned. By default the scaling factor is 0.5. The vertical speed is conditional on the horizontal euclidean distance. This means that when the UAV is at a higher altitude, it can start descending from farther away. However, as the UAV's altitude is closer to the target, it needs to remain horizontally closer to the centre of the target. When within this range, the altitude decreases at a speed of 0.5m/s. There will come a point when the UAV is very close to target and will lose the pose. At this stage, the UAV will retain its current velocity, and continues to lower until it has detected that it has landed.

The second component it needs to deal with is the possibility of a moving target. The previously required velocity is relative to the target and needs to be added to the target's current velocity which needs to be calculated separately. The target velocity relative to the UAV is found by calculating the difference between the current location of the target from the previous location and divided by the time since the last calculated pose was received. This relative velocity is then added to the UAV's current velocity u_d to calculate the target's true velocity u_t relative to the ground. This x and y components of this target velocity is then added to the x and y components of the previously requested velocity before being sent to the flight controller. This then allows the UAV to keep up with the target and still do the fine adjustments to land in the centre. This does require the target to maintain a consistent velocity that can have some variation, but not too much. The resulting equations developed to be used by landing controller are shown below where p_0 is the previous pose of the target relative to the UAV, p is the current target pose, t is the time since the last pose calculation, and u_d is the current velocity of the UAV and c is the speed multiplier.

$$d = \sqrt{x^2 + y^2}$$

$$f(z) = \begin{cases} 0 & \text{if } d > z \\ 0.5 & \text{otherwise.} \end{cases}$$

$$u_t = \left(\frac{p - p_0}{t}\right) + u_d$$

$$\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} u_{t_x} + cp_x \\ u_{t_y} + cp_y \\ f(p_z) \end{bmatrix}$$

4 Testing Methodology

4.1 Environmental Variables

The testing of the landing algorithm will be done to determine the approximate range of conditions in which the UAV is capable of landing. A baseline of how well the UAV performs in perfect conditions will be determined by running the simulation repeatedly. Then each main environmental condition will be individually investigated to determine the bounds in which the UAV performs sufficiently well. The conditions being investigated are as follows: the standard deviation of the normally distributed error added to the UAV pose calculations in the x and y directions, the speed of the target, and the mean wind speed. For each of these, three possible values are selected to be tested, and multiple sample runs will be simulated for each value. Once the environmental conditions with an acceptable landing error are determined, then ten sample runs will be simulated with these conditions, and the results can be compared to the previously found baseline.

4.2 Logging and Metrics

The way in which these sample experiments are measured is by logging aspects of the landing process. Specifically, the logging starts when the target is first visible and the landing controller first switches to the landing state. Then every frame of the simulation saves a new entry containing the time since it switched to the landing state, and the x, y and z coordinates relative to the target. Once the UAV has properly landed, then logging completes. Each log file will be associated with the configuration for which it was run. The key part of each log entry will be the final entry when it landed as this will be the resulting landing location.

The main metric obtained from this will be the euclidean distance from the centre of the target using the x and y coordinates. The distribution of distance is highly unlikely to be the normal distribution. While it is possible that x and y are normally distributed, the derived distance will be skewed to the right similarly to the χ (Chi) distribution as it is converted to be a purely positive value calculated using the square root of the sum of squares. Derived distributions are proven to remain normal if it consists of linear combinations of other normal distributions, but the sum of squares of normal variables produces the χ^2 (Chi-Squared) distribution, and the square root turns it into the χ distribution. The sample median and quartiles will be good indicators as to whether the landing was within the acceptable error margin. Another metric that will be used in comparing the baseline with the combination of environmental conditions is the time it takes to land as this may provide some more indication of how each environmental condition affects the landing process.

4.3 Target Limits

The error margin is determined by the motivation behind the project which is using the UGV as a portable charging station for the UAV to land on. As the target is designed with the Clearpath Husky in mind, the dimensions of the entire target is one metre by one metre, but the payload portion which would contain the charging apparatus is within the central half metre by half metre, which should be the only acceptable zone within which the UAV should land. Therefore the acceptable error margin to be used is 0.25 metres from the centre of the target.

4.4 Sampling Technique

Initially the baseline sample was created where the UAV would start landing from an altitude of ten metres above the target with ten runs. Following this, ten runs were collected for each different configuration tested. The pose estimation errors were created by randomly selecting an angle from the uniform distribution, and randomly selecting a distance from the normal distribution centred around the target centre. This does mean that the distance can be negative, but the resulting error vector is the equivalent of rotating the angle by 180° and taking the absolute value of the distance, effectively converting it to the half-normal distribution.

The pose estimation experiments were controlled by setting the standard deviation of the distance error to be 0.02m, 0.05m and 0.10m. The selected target speeds were 0.5m/s, 1.0m/s and 1.5m/s. The Clearpath Husky has a maximum speed of 1.0m/s [6], so it was important to test that and an increased buffer. This allows for other potential robots to be used as the target. The Boston Dynamics Spot robot dog has a max speed of 1.6m/s

[7] and could be a potential moving target to investigate, although there may be more difficulty as it will have variation in vertical height of the target on its back due to it using legs as opposed to wheels. Three mean wind speeds were chosen to be tested corresponding to the Beaufort scale of 3, 5 and 7 from a gentle breeze to near gale speeds: 5m/s, 10m/s and 15m/s. UAVs are unlikely to be flying in extreme winds, but these experiments help provide insight into the performance, especially if the UAV needs to land due to unforeseen extreme weather.

Once experiments had been carried out testing individual parameters, experiments were then to be run combining the three tested parameters at values which have a significant effect but are not prone to causing significant landing discrepancies beyond the acceptable error margin. The error margin was a maximum of 0.25m from the centre of the target as determined by the dimensions of the payload section of the Clearpath Husky.

4.5 Statistical Tests

The primary differences that are being investigated are the differences in distributions between samples of different environmental configurations. To decide which statistical tests to use, it was important to understand the distribution of the metrics used: distance from the target centre, and the time it took to land. As discussed in the choosing of metrics, distance is not normally distributed, and it cannot be assumed that the time to land is. Therefore all of the statistical tests need to be non-parametric.

For comparing samples of differences in the same variable (pose estimation error, target speed, and wind speed), the Kruskal-Wallis test was used to determine if at least one sample had a different median from the others. The null hypothesis is that all samples share the same median, while the alternate hypothesis is that at least one sample has a different median. This was tested twice: once with the baseline and once without. If the null hypotheses is rejected indicating a difference in at least one of the sample medians, then each pair of samples is compared using the Mann-Whitney U test to compare individual distributions. The null hypothesis for this test is that $P(X > Y) = P(X < Y)$ and the alternate hypothesis is $P(X > Y) \neq P(X < Y)$. Comparing all the pairs then gives a sense of which configurations cause a significant difference in distribution. For some of these, it is also useful to compare the variance of different samples using Levene's test. The null hypothesis is that the variances are the same and the alternate hypothesis is that they have different variances. All tests run were two-tailed at 95% confidence.

5 Results, Analysis and Evaluation

5.1 Experimental Observations

When testing the moving target, the UAV struggled to initially locate the target due to the errors in the PX4 position based navigation. There seemed to be an issue where PX4 thought it had navigated to the correct location, but this was not accurate to the true location. Because of this, the altitude was increased to fourteen metres above the target which was enough for the UAV to see the target most of the time despite the navigation errors and start the landing process. Further investigation would be needed to identify the source of the navigation errors in PX4 and to develop a search algorithm around the GPS location to locate the target. Since these new experiments now had a new altitude, a secondary baseline sample of ten experiments was created from this new altitude for subsequent tests to ensure that there was only one variable being compared.

Occasionally certain errors and deviations occur. In some runs, it lost sight of the target and reverted to returning before locating the target again and landing. In other runs, the UAV sometimes failed to land or landed far from the target centre. Some of the issues were addressed by either modifying the algorithm or recommending a lower value of the specific parameter.

5.2 Empirical Results

The box plots in figure 5.1 provide an overview of the distribution of each sample. Based on the results from from the individual experiments, the combined tests used a pose error standard deviation of 0.05m, a target speed of 1m/s and a mean wind speed of 10m/s. The initial altitude was kept at 14m for this.

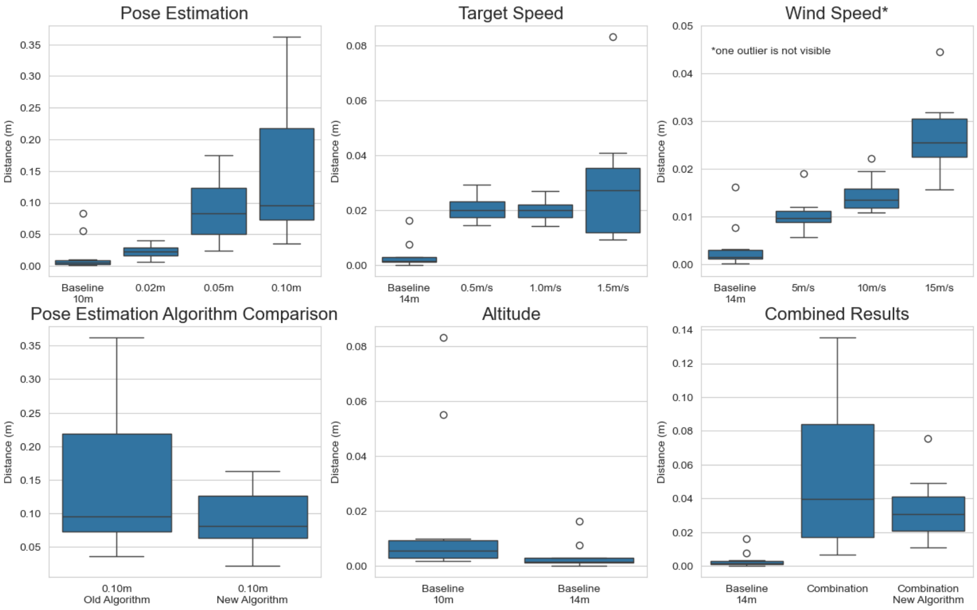


Figure 5.1: A series of boxplots of all the samples of 10 runs each grouped by the primary comparison between samples. The wind speed experiment has been cropped to remove an outlier, where the UAV landed 0.426m from the centre, that made it very difficult to see the distribution.

5.2.1 Pose Estimation

When comparing the distance samples across different pose estimation errors, the Kruskal-Wallis test showed that there was a significant difference in at least one of the sample medians regardless of whether or not the baseline was included in the test. Given this difference, the Mann-Whitney U test was run comparing each pair of samples and that showed that only the 0.05m and 0.10m samples were not statistically different from each other given 95% confidence. However, other combinations were all significantly different. That pair of samples that did not show a significant difference in median, did not show a significant difference in variance using Levene's test. Despite what seems to be a large difference in the box plot, there was insufficient evidence to suggest a different variance between them.

However, there is an issue that occurred during testing that cannot be shown by statistical tests or box plots, and that is the number of failed experiments. This is explored further below, but to summarise, there were 6 failed runs on top of the 10 successful runs for the 0.10m sample. In contrast, the 0.05m sample only had one failed run. The failure occurs when it is about to land but the erroneous pose calculation indicates to the UAV that it is too far from the centre of the target to land. Because of this proportion of errors in the 0.10m sample, the combined runs used 0.05m as the pose error standard deviation. All of the successful landings of this configuration were within 0.2m of the centre which is within the proposed maximum error allowance.

5.2.2 Initial Altitude

After discovering that a higher altitude was necessary for the moving target because of the PX4 navigation errors, a new baseline was created with a new altitude of 14m. This was then compared to the original baseline with an altitude of 10m to see whether altitude made any difference to the distance from the centre of the target once landed. A difference was not anticipated, but after carrying out the Mann-Whitney U test, the sample distances produced a statistically significant difference in distribution suggesting that initial altitude does affect the distance from the target centre.

My hypothesis as to why this is the case is that an increase in altitude leads to more time for the flight path to smooth itself out. There were a couple of runs in the baseline sample where the UAV lost sight of the target during the descent. When this occurred it reverted back to the PX4 position navigation to be above the target at the specified altitude. It would then detect the target and start landing again. This pattern could also be lead to a more unstable flight path with the sudden switching of modes. However, further testing would be necessary to confirm that either of these are the reason. It is also worth noting that the majority of both samples are less than 1cm from the centre, so both starting altitudes produce satisfactory landings.

5.2.3 Target Speed

For testing capabilities of the landing algorithm with a moving target, the target moved at a constant velocity in a single direction at the speeds of 0.5m/s, 1.0m/s and 1.5m/s. Running the Kruskal-Wallis test on the different samples showed that when the baseline was included in comparison to the other samples, there was a significant difference. However, when the samples were compared with each other without the baseline, the test showed no statistically significant difference which seems to indicate that the only configuration that causes a difference is whether the target is moving or not as opposed to the speed of the target. The Mann-Whitney U test comparing each sample to the baseline and the other samples confirmed the results that the only statistically significant sample which had a different distribution was the baseline. Levene's test showed that the only sample with a significantly different variance was the 1.5m/s sample.

Given this, it can be safely concluded that the UAV can consistently land accurately on moving targets with speeds in the tested range, although when it gets up to 1.5m/s, consistency of landing distance starts to change. The majority of all the sample runs all landed within 4cm of the centre of the target with the one outlier just over 8cm away which is still well within the proposed margin of error of 25cm. The max speed of the Clearpath Husky is 1.0m/s [6] which is well within what the algorithm can handle. The flexibility of higher speeds means that the Husky could be potentially swapped out for another UGV with faster speeds if necessary. However, for the scope of the project based around the Husky, the algorithm is very capable.

5.2.4 Wind Speed

The final major parameter to be tested was wind speed. This is an important aspect of the environment to test as it directly affects the flight of the UAV. The highest speed tested was near gale-force winds with a mean of 15m/s which is well above what is usual for UAVs to fly in. When testing the samples, the Kruskal-Wallis test showed significant differences between samples, and the follow-up Mann-Whitney U tests showed that every sample had a significantly different median from every other sample and the baseline. When testing variance with Levene's test, none of the samples had any significant difference in variance. It still was able to successfully land with that wind speed, the distance in the majority of landings were within 5cm meaning that it is very capable of dealing with such wind speeds. Wind was not directly addressed by the algorithm, however PX4 built in control systems to ensure that the requested velocity is true ground velocity by increasing the speed in the direction the wind comes from, counteracting it. The extreme wind speeds of 15m/s did cause an outlier with a distance of over 0.4m from the centre which is outside the proposed error allowance. The majority of the time, it will be fine but given this occurred once within a sample of ten runs, there is potential for it to be more often than preferable, and hence the combined test below uses a lower wind speed of 10m/s.

5.2.5 Combined Parameters

As briefly mentioned above, the configuration combining all of the varying environmental conditions used the previous results to select a reasonable combination of parameters to do a stress test. The chosen parameters were pose calculation error standard deviation of 0.05m, a target speed of 1m/s and a wind speed of 10m/s. The altitude at the start of the landing process was 14m. After running the Mann-Whitney U test comparing this sample to the baseline which was also set to 14m altitude, there was sufficient evidence to reject the null hypothesis. The distance therefore is affected by these parameters. However, all the runs of the combination sample were still well within the 25cm maximum error. The mean distance error was 5.4cm with a standard deviation of 4.7cm. The maximum error of the combined sample was 13.5cm, so any parameter configuration within the range from the baseline to these parameters should enable the landing algorithm to be fully functional, even with some buffer to allow for slightly worse environmental conditions.

At this stage, another metric was tested: time taken to land. This is measured from when the UAV first sees the target until it has successfully landed. The mean time taken to land from 14m altitude was 27.7 seconds. The testing of this metric was only started here as it would give an indication to see if further statistical tests were needed to see which parameters affect the time taken to land. However, after running the Mann-Whitney U test on time comparing the baseline sample to the combination sample, there was insufficient evidence to suggest that the median time taken to land was any different. The UAV seems to land equally fast regardless of environmental conditions, which is very good, particularly for the case where it needs to land due to low battery, although further testing would be necessary to see if battery levels affect how well the UAV flies in these conditions.

5.3 Erroneous Results

During the testing of the pose estimation noise, it was noted that the flight path when approaching the target seemed to vary a lot more than expected. There were also instances where the UAV would fail to land with the 0.10m sample due to the calculation saying that the UAV was too far away from the centre of the target in comparison to its altitude. When it is too far away from the centre, the UAV halts its descent and should navigate towards the centre at its current altitude. However, the instances where this fails are where it gets one wrong calculation before losing sight of the target causing the UAV to navigate base on its last seen location, causing it to drift further and further away from the target and never land until the PX4 fail-safe kicks in (a procedure to ensure safety when something goes wrong). This does highlight an issue in the algorithm where it can only detect this error above a certain altitude, and does not know when it occurs and how to deal with it when very close to the target. The original reason for only recalculating above a certain altitude is because the UAV is likely to lose sight of the target at low altitudes anyway. Before putting this into a real-world system, it would be necessary to create more fail-safes to deal with these errors. Beyond fail-safes, there were also some potential changes to the algorithm that could reduce the error rate.

5.4 Algorithm Tweaks and Tests

The first tweak to the algorithm to help reduce the error rate was to increase the zone around the target centre which was an acceptable range to keep descending. This cut the error rate from 6 failures for 10 successful runs, to just a single failure for 10 successful runs. This was in the configuration of only changing the pose estimation noise with setting the standard deviation to 0.10m. The second change was created to attempt to smooth out the flight path when there are large jumps in calculated distance also due to the pose estimation noise. There are two parts of the velocity controller that are affected by this noise. First is just the local offset from the target, and the second is the calculated velocity of the target which is derived from the UAV's velocity and the change in pose calculation. Having both of these parameters affected by the pose calculation noise compounds the effect on the flight of the UAV. The change made was to how the target velocity is calculated. Rather than recalculating it completely every frame which could indicate it is moving fast when it is just an error in the pose, the system takes that newly calculated velocity and averages it with the previously calculated velocity. The hope was that this would keep the target velocity calculations closer to the ground truth.

Now that these changes had been made, they had to be tested to check if they made any difference. The initial test was to get a new sample where the only parameter different from the baseline is the pose calculation noise where the standard deviation is 0.10m from an altitude of 10m to match the previous sample of this configuration with the old algorithm. As mentioned above, this massively reduced the rate of failed landings. The Mann-Whitney U test was then done to compare the medians of these two samples. There was no significant difference shown by this test. Then Levene's test was carried out to compare the variances between the two samples. Despite the box plot seeming to indicate that there is a different variance, there was no significant difference shown by the statistical test.

Following this up, the new algorithm was tested on the combination sample. The new sample was created with all the same parameters as the combined test with the old algorithm to ensure the only variable being tested was the algorithm. The Mann-Whitney U test results indicated there is no difference in median between the old and new algorithm, but there was sufficient evidence from Levene's test to suggest that there was a difference in variance which is also clearly visible in the box plot. From this, it seems to be clear that the new algorithm made a significant different to the distance from the centre of the target once landed. However there is not a clear metric to use which measures the smoothness of the flight path. The graph in figure 5.2 of the distance over time across the samples does give a bit of an indication that it made a difference to the flight path smoothness. After around fifteen seconds from when it started landing where it is now centred over the target, the new algorithm has a maximum distance of 10cm while the old algorithm has a maximum distance of 15cm.

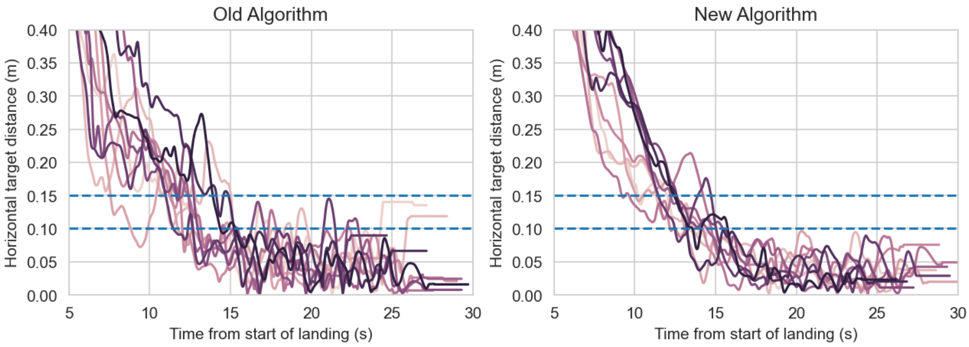


Figure 5.2: The horizontal distance of the UAV from the centre of the target over time for all runs within each sample. The two samples shown are of the same combined environmental configuration just using the two algorithm versions. The horizontal lines highlight the difference in maximum distances for each sample after converging at around 15 second in.

The final test of the new algorithm is to see if it makes a difference in the time it takes to land. The Mann-Whitney U test and Levene's test both show insufficient evidence to suggest that either median or variance respectively have any difference. It can therefore be concluded that the time taken to land is unaffected by which of the two algorithm was used.

5.5 Proposed Limits of Conditions

Given these results from testing, the following parameter limits are proposed for the conditions in which this algorithm will work. The mean error in any pose calculation should ideally be less than 4cm with a standard deviation of around 3cm in the half-normal distribution. This is derived from the absolute value normal distribution where $\mu = 0$ and $\sigma = 5$cm fitting with the pose estimation error samples. It is worth stating that the error is likely to increase with the vertical distance from the target so the proposed limits only apply to the final approach in the last metre or two.

The speed of the UGV can easily be 1m/s matching the max speed of the Clearpath Husky. The maximum wind speed should be 10m/s which is the equivalent of a fresh breeze (5 on the Beaufort scale [44] approximately equivalent to 36kph or 22mph). The altitude when the UAV starts landing should be approximately 14m above the target. Given these parameters, this algorithm should consistently land accurately well within the maximum 25cm distance error.

5.6 Limitations of Fiducial Markers

The primary limitation of using a single fiducial marker to estimate the camera pose is due to a fundamental ambiguity as explained by Schweighofer and Pinz [33] by showing that the error functions used for pose estimation have two minima in the reprojection error. As the distance between the camera and the marker increases, so does the ambiguity as the lack of perspective leads to difficulty in estimating the rotation of the camera. Image noise is another limiting factor in calculating an accurate pose.

The mounting of a camera to the UAV without a gimbal also leads to limited success with pose detection. This is especially pronounced when it comes to using the plain ArUco marker as it is very easily occluded as the UAV changes orientation. Both the Fractal ArUco marker and the ChArUco board do help address this issue by not requiring the entire marker to be visible to be detected. However it does not completely eliminate the issue, particularly if the orientation of the UAV angles the camera away from the target to keep up with a moving target or to counteract the wind. The most straightforward solution to this would be to attach the camera to a gimbal and point it directly at the target once located [34]. By doing this marker tracking, it would minimise occlusion of the target as the UAV flew in different directions to adjust its course, but it would also be able to detect the target from a farther horizontal distance as the camera can be angled forward while the UAV is navigating by GPS looking for the target.

5.7 Proposed Pose Estimation Solutions

Future research into constraining the problem can be done on the following possible solutions to the pose estimation ambiguity. Knowledge of the orientation and headings of the UAV and UGV could constrain the problem.

Using these constrains, the remaining unknown component of the relative pose would be the position leaving just the correct pose.

5.7.1 Temporal Filtering

Another method to constrain the problem would be using previous calculations to restrict possible new pose calculations. This would help provide a better starting point for gradient descent solutions. Another use of the previous calculations is to measure similarity between the old and new calculations and filter out any subsequent calculations with a major difference, indicating a potentially impossible change in location.

5.7.2 Depth Camera

Another possible solution is to use a depth camera to restrict any estimation algorithms to match the same depth gradient as the camera as done in this paper [45]. Due to depth sensors having an innate noise level, the depth points are filtered to be within a certain range of the median depth. The normal of the plane is calculated from the depth which restricts three of the degrees of freedom (DoF) from the total 6 DoF and simplifies the pose estimation. The corners of the marker are calculated using the provided RGB detection algorithm, are then projected onto the plane calculated from the depth providing the necessary information for calculating the correct camera pose.

5.7.3 Multiple Cameras

A similar solution would be to have a stereo camera or multiple cameras mounted on the UAV, or even on the UGV. The known placement of the cameras further constrains the problem and hence helps in determining the pose of the UAV. This particular technique would be a subset of multi-view stereo which is the technique of determining where the points are in relation to the cameras. If the cameras are on the UGV, then the calculated pose would be transmitted to the UAV. The wireless communication should be reliable enough for this when the drone is close enough to start landing.

5.7.4 Target Design

Changes to the target beyond just the pattern of the marker could be made. The simplest would be to add a third dimension to the target such as adding secondary markers that are not co-planar with the primary marker. The additional points would limit the possible poses to be close to the true pose. This may become challenging as they disappear out of view when the UAV gets close, but when it is that close, the issues are minimal and the UAV can maintain a constant horizontal velocity while going down.

5.8 Reality Gap

This has all been within the simulator which will have a gap in how well it represents reality, so before this algorithm is put into practice, experiments would have to be done in a controlled, real-world environment. However, there should be a fairly limited discrepancy between the simulation results and experiments done on a real UAV so long as the conditions follow the above proposed limits. The PX4 autopilot simulation models should be fairly accurate. A controlled environment where a person can override the controls if necessary will provide the safety requirements for those kind of experiments. When the UAV is in the returning state, the code should properly switch back to the PX4 navigation before going back to manual landing controls. The primary limit of environmental conditions for the precision landing that should be followed is the the pose estimation error.

5.9 UAV Legislation

When implementing real-world robotic autonomous systems, especially in relation to UAVs, there is legislation that must be followed. In the UK, this is defined by the UK Civil Aviation Authority (CAA) in the Drone and Model Aircraft Code [46]. The rules set out in this include limits on allowed airspace, how close to people one can fly a UAV, and requirements direct line of sight between the pilot (or an assistant observer) and the UAV. This last legislation does pose a challenge for deploying autonomous systems that go beyond line of sight, but for the moment permission can be sought from the CAA for specific uses. Legislation to standardise the process of approval of UAV systems that go beyond visual line of sight [47] is being developed where UAVs would be required to have testing of avoidance systems in various environments before deployment.

6 Conclusion

The overarching goal of this project has been been to explore the potential in collaboration between a UAV and a UGV in autonomous missions, in particular looking at precision landing of the UAV on the UGV. An initial landing algorithm has been developed and tested and the goals set in the introduction have been achieved in both progressing research in this field and highlighting key areas for expanding this research.

6.1 Achievement of Project Goals

The original project goals are repeated here along with the extent to which they were achieved:

1. Set up a simulation environment - A simulation world was created in Gazebo containing a realistic quad-copter UAV model and a simplified representation of a UGV based on the Clearpath Husky.

2. Select a pose estimation method - For determining the UAV pose, the Fractal ArUco marker was placed on the UGV for the UAV to identify when it is in range, and while the marker is visible, the true location is provided by the simulation with some added artificial noise to emulate calculation errors. Other potential markers were explored to calculate the pose directly but ambiguity in the calculations limited the effectiveness of those other methods.

3. Develop an algorithm to land the UAV on the UGV - A simple algorithm was developed to fly the UAV from the end of a mission to the UGV, receive the relative pose estimate and proceed to land precisely on the target mounted on the UGV.

4. Determine the optimal range of environmental conditions in which the algorithm is successful in landing the UAV on the UGV - The algorithm was then tested against a variety of factors that affect the performance of the algorithm to determine the limits of the algorithm's capabilities of landing the UAV within a margin of error on the target. These determined limits as defined in the analysis are a mean pose

estimation error of 4cm and standard deviation of 3cm, a target speed of 1.0m/s and a mean wind speed of 10m/s. Within these conditions, the UAV can effectively land well within the error margin of 25cm from the centre of the target.

5. Identify key areas to be researched further - Key areas of future research has been touched on previously but is further expanded below.

6.2 Future Research

This project has only investigated the start of a wide research field in autonomous robot collaboration. There are several parts of this project that leave plenty more to be researched. The key field that is essential for this project to progress is that of pose estimation. Investigating ways that either allow the UAV to locate the accurate 3D position and orientation of the target or vice versa as discussed in section 5.7 is the foundation that the landing algorithm builds on. Primary options include using a depth or stereo camera on the UAV, creating a new target that has more than just two dimensions, or enabling the target to locate the UAV with multiple cameras and send the pose across.

That final object of investigation requires another field that can be further investigated: inter-robot communication in remote environments. This would be especially difficult in places that do not have cell coverage. This research could include looking into robust wireless communication and networks, search patterns for the UAV to follow to locate the target should communication fail, and methods to communicate errors for human override. In these remote places, there is also the challenge of UGV navigation over rough terrain. The robot would have to understand its environment and its capabilities, and from that identify the best way to follow the UAV.

Finally a key area of investigation particularly relating to the algorithm is identifying when failures occur. When autonomous systems are developed and used, people may not be watching it all the time to identify when errors occur. Investigating how these errors are identified and then rectified is key. However, overall this project has formed the basis of an effective and reliable system that will result in extending mission flight times by allowing the UAV to land on the UGV which in turn will charge the UAV.

A Communication Architecture of PX4 and ROS2

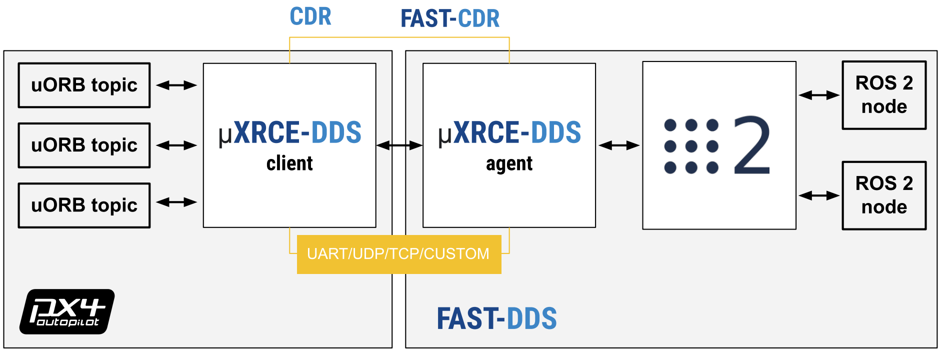
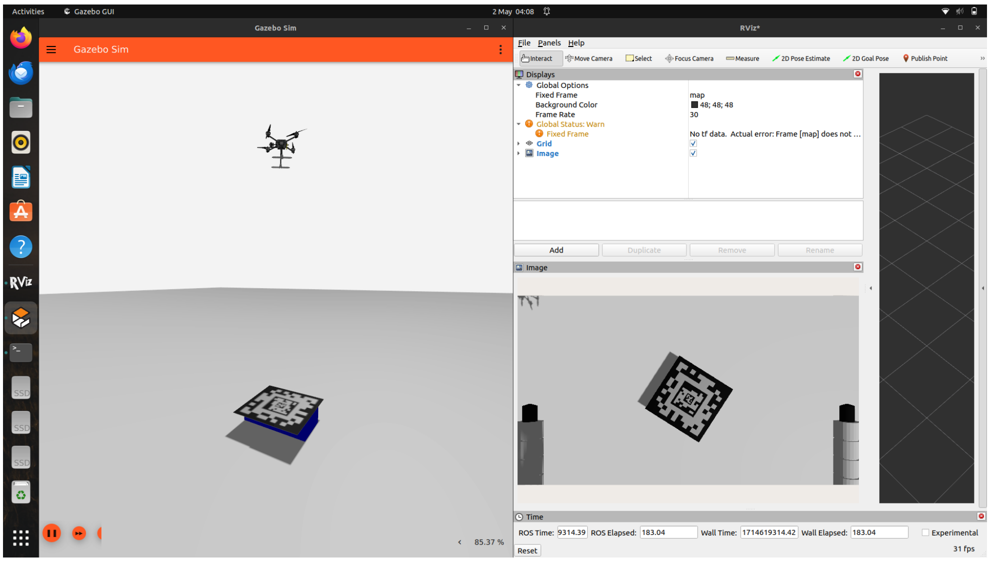


Figure A.1: An overview of the communication architecture between ROS2 and PX4 from the PX4 documentation [37]

B Simulation Screenshot



Global Options
Fixed Frame: map
Background Color: 48; 48; 48
Frame Rate: 30
Global Status: Warn
Fixed Frame: No tf data. Actual error: Frame [map] does not ...
Grid: ✓
Image: ✓

ROS Time: 9314.39 ROS Elapsed: 183.04 Wall Time: 1714619314.42 Wall Elapsed: 183.04

Figure B.1: A screenshot of the simulation with the UAV moving to land on the UGV. The left window shows the Gazebo simulator with the UAV and UGV in the simulated world. The window on the right is RViz2 showing the camera image with the Fractal ArUco marker captured from the UAV.

C Graph of ROS2 Nodes and Topics

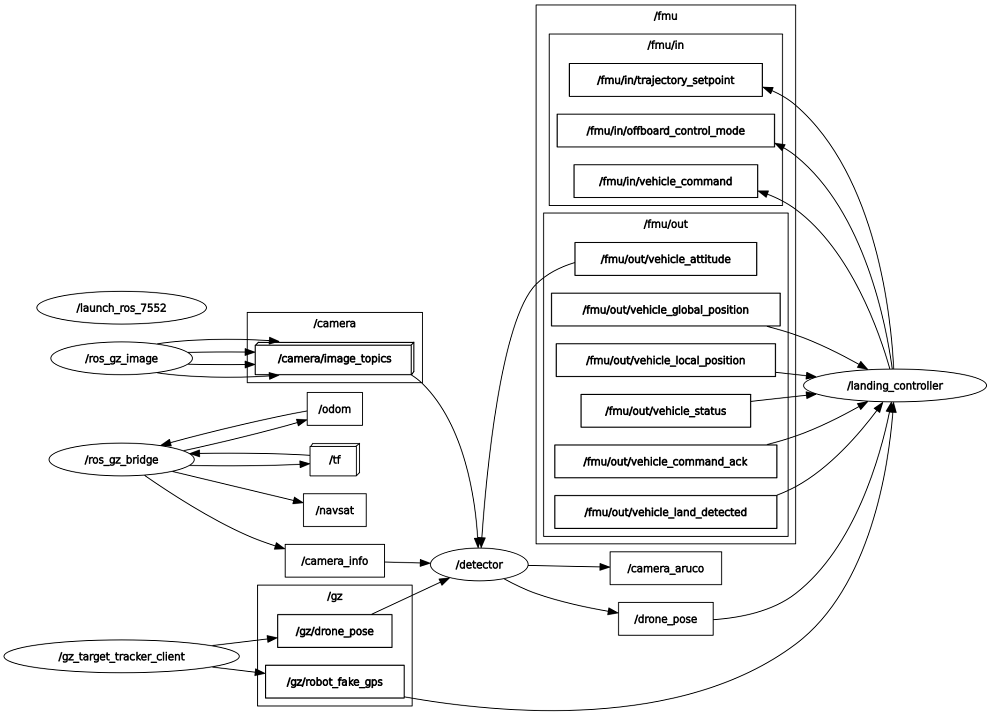


Figure C.1: A graph of how different ROS2 nodes communicate each other across topics (communication channels). The ellipses represent ROS2 nodes where the ones implemented by this project are gz_target_tracker_client, detector, and landing_controller. All of the rectangles are ROS2 topics passing messages between nodes. All of the "fmu" topics go directly to and from PX4, where others may come from Gazebo or other ROS2 nodes.

D Statistical Test Results

Experiment	**With Baseline**	**Without Baseline**
Pose Estimation Error	1.64×10^{-5}	0.000183
Target Speed	0.000156	0.784
Wind Speed	1.44×10^{-6}	3.00×10^{-5}

Table D.1: p-values of Kruskal-Wallis test results for each parameter tested with and without the baseline run from the same altitude. 95% significance means that significant p-values are less than 0.05

Configuration	Mann-Whitney U	Levene's test
Baseline and 0.02m	0.0452	0.486
Baseline and 0.05m	0.00171	0.0257
Baseline and 0.10m	0.000769	0.0524
0.02m and 0.05m	0.00101	0.000286
0.02m and 0.10m	0.000246	0.0287
0.05m and 0.10m	0.427	0.254

Table D.2: p-values of the Mann-Whitney U test and Levene's test between every pair of samples for the pose estimation error with regards to distance.

Configuration	Mann-Whitney U	Levene's test
Baseline and 0.5m/s	0.000246	0.543
Baseline and 1.0m/s	0.000330	0.788
Baseline and 1.5m/s	0.000583	0.0136
0.5m/s and 1.0m/s	0.791	0.592
0.5m/s and 1.5m/s	0.623	0.0179
1.0m/s and 1.5m/s	0.571	0.0136

Table D.3: p-values of the Mann-Whitney U test and Levene's test between every pair of samples for the target speed with regards to distance.

Configuration	Mann-Whitney U	Levene's test
Baseline and 5m/s	0.00361	0.00361
Baseline and 10m/s	0.00131	0.00131
Baseline and 15m/s	0.000246	0.000246
5m/s and 10m/s	0.00728	0.00728
5m/s and 15m/s	0.000246	0.000246
10m/s and 15m/s	0.000769	0.000768

Table D.4: p-values of the Mann-Whitney U test and Levene's test between every pair of samples for the wind speed with regards to distance.

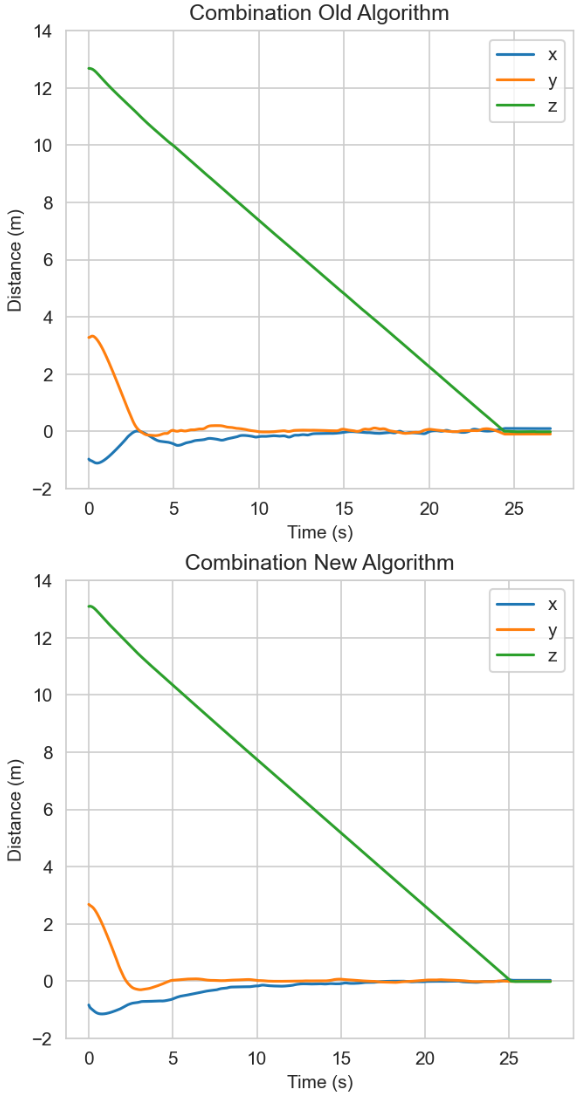
Configuration	Mann-Whitney U	Levene's test
Altitude	0.0257	N/A
Baseline and Combination	0.000769	N/A
Old and new algorithm: Pose	0.473	0.202
Old and new algorithm: Combination	0.677	0.0157

Table D.5: Additional statistical test results (p-values) for specific pairs of samples with regards to distance.

Configuration	Mann-Whitney U	Levene's test
Baseline and Combination	0.384	0.324
Old and new algorithm: Combination	0.384	0.698

Table D.6: Additional statistical test results (p-values) for specific pairs of samples with regards to time. The algorithm and baseline comparisons have the same p value for the Mann-Whitney U test. This is not a mistake, but they just happen to have the same probability.

E Sample Combined Run Trajectories



Combination Old Algorithm

Combination New Algorithm

Figure E.1: Sample runs from using the combined parameters, with one from the old and one from the new algorithm. The graphs show the variation in x, y, and z coordinates over time throughout the landing process.

F Half-Normal Distribution Calculations

Half-Normal Metric Calculations	
Mean	$\mu + \sqrt{\frac{2\sigma^2}{\pi}}$
Standard Deviation	$\sqrt{\left(1 - \frac{2}{\pi}\right)\sigma^2}$

Table F.1: The formulae for calculating the mean and standard deviation of the half-normal distribution given the mean μ and standard deviation σ of the base normal distribution.

G Code Library Citations

Special thanks goes to the developers of the following libraries used in this project: PX4 [37], ROS2 Humble [39], Gazebo [38], ArUco [28][41], OpenCV [42], NumPy [48], Pandas[49], SciPy [50], Matplotlib [51] and Seaborn [52].

Bibliography

[1] E. Garcia, M. A. Jimenez, P. G. De Santos and M. Armada, 'The evolution of robotics research,' *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 90–103, Mar. 2007, Conference Name: IEEE Robotics & Automation Magazine, ISSN: 1558-223X. DOI: 10.1109/MRA.2007.339608. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/4141037 (visited on 29/04/2024).

[2] A. Billard and D. Kragic, 'Trends and challenges in robot manipulation,' *Science*, vol. 364, no. 6446, eaat8414, Jun. 2019, Publisher: American Association for the Advancement of Science. DOI: 10.1126/science.aat8414. [Online]. Available: https://www.science.org/doi/full/10.1126/science.aat8414 (visited on 29/04/2024).

[3] F. Rubio, F. Valero and C. Llopis-Albert, 'A review of mobile robots: Concepts, methods, theoretical framework, and applications,' en, *International Journal of Advanced Robotic Systems*, vol. 16, no. 2, p. 1 729 881 419 839 596, Mar. 2019, Publisher: SAGE Publications, ISSN: 1729-8806. DOI: 10.1177/1729881419839596. [Online]. Available: https://doi.org/10.1177/1729881419839596 (visited on 30/04/2024).

[4] N. Elmeseiry, N. Alshaer and T. Ismail, 'A Detailed Survey and Future Directions of Unmanned Aerial Vehicles (UAVs) with Potential Applications,' en, *Aerospace*, vol. 8, no. 12, p. 363, Dec. 2021, Number: 12 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 2226-4310. DOI: 10.3390/aerospace8120363. [Online]. Available: https://www.mdpi.com/2226-4310/8/12/363 (visited on 14/12/2023).

[5] *Mars 2020: Perseverance Rover - NASA Science*. [Online]. Available: https://science.nasa.gov/mission/mars-2020-perseverance/ (visited on 01/05/2024).

[6] *Husky UGV - Outdoor Field Research Robot by Clearpath*, en-US. [Online]. Available: https://clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/ (visited on 23/04/2024).

[7] *Spot Specifications*. [Online]. Available: https://support.bostondynamics.com/s/article/Robot-specifications (visited on 23/04/2024).

[8] L. F. P. Oliveira, A. P. Moreira and M. F. Silva, ‘Advances in Agriculture Robotics: A State-of-the-Art Review and Challenges Ahead,’ en, *Robotics*, vol. 10, no. 2, p. 52, Jun. 2021, Number: 2 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 2218-6581. DOI: 10.3390/robotics10020052. [Online]. Available: https://www.mdpi.com/2218-6581/10/2/52 (visited on 30/04/2024).

[9] P. Tokekar, J. V. Hook, D. Mulla and V. Isler, ‘Sensor Planning for a Symbiotic UAV and UGV System for Precision Agriculture,’ *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1498–1511, Dec. 2016, Conference Name: IEEE Transactions on Robotics, ISSN: 1941-0468. DOI: 10.1109/TRO.2016.2603528. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7587351 (visited on 01/05/2024).

[10] J. Li, G. Deng, C. Luo, Q. Lin, Q. Yan and Z. Ming, ‘A Hybrid Path Planning Method in Unmanned Air/Ground Vehicle (UAV/UGV) Cooperative Systems,’ *IEEE Transactions on Vehicular Technology*, vol. 65, no. 12, pp. 9585–9596, Dec. 2016, Conference Name: IEEE Transactions on Vehicular Technology, ISSN: 1939-9359. DOI: 10.1109/TVT.2016.2623666. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7728117 (visited on 01/05/2024).

[11] Z. Yan, N. Jouandeau and A. A. Cherif, ‘A Survey and Analysis of Multi-Robot Coordination,’ en, *International Journal of Advanced Robotic Systems*, vol. 10, no. 12, p. 399, Dec. 2013, Publisher: SAGE Publications, ISSN: 1729-8806. DOI: 10.5772/57313. [Online]. Available: https://doi.org/10.5772/57313 (visited on 29/04/2024).

[12] A. Gautam and S. Mohan, ‘A review of research in multi-robot systems,’ in *2012 IEEE 7th International Conference on Industrial and Information Systems (ICIIS)*, ISSN: 2164-7011, Aug. 2012, pp. 1–5. DOI: 10.1109/ICIInfS.2012.6304778. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6304778 (visited on 01/05/2024).

[13] M. Erdelj, E. Natalizio, K. R. Chowdhury and I. F. Akyildiz, ‘Help from the Sky: Leveraging UAVs for Disaster Management,’ *IEEE Pervasive Computing*, vol. 16, no. 1, pp. 24–32, Jan. 2017, Conference Name: IEEE Pervasive Computing, ISSN: 1558-2590. DOI: 10.1109/MPRV.2017.11. [Online]. Available: https://ieeexplore.ieee.org/document/7807176 (visited on 12/12/2023).

[14] S. M. S. Mohd Daud, M. Y. P. Mohd Yusof, C. C. Heo *et al.*, ‘Applications of drone in disaster management: A scoping review,’ en, *Science & Justice*, vol. 62, no. 1, pp. 30–42, Jan. 2022, ISSN: 13550306. DOI: 10.1016/j.scijus.2021.11.002. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1355030621001477 (visited on 11/12/2023).

[15] J. Balaram, M. Aung and M. P. Golombek, 'The Ingenuity Helicopter on the Perseverance Rover,' en, *Space Science Reviews*, vol. 217, no. 4, p. 56, May 2021, ISSN: 1572-9672. DOI: 10.1007/s11214-021-00815-w. [Online]. Available: https://doi.org/10.1007/s11214-021-00815-w (visited on 30/04/2024).

[16] K. Fujii, K. Higuchi and J. Rekimoto, 'Endless Flyer: A Continuous Flying Drone with Automatic Battery Replacement,' in *2013 IEEE 10th International Conference on Ubiquitous Intelligence and Computing and 2013 IEEE 10th International Conference on Autonomic and Trusted Computing*, Italy: IEEE, Dec. 2013, pp. 216–223, ISBN: 978-1-4799-2482-0. DOI: 10.1109/UIC-ATC.2013.103. [Online]. Available: http://ieeexplore.ieee.org/document/6726212/ (visited on 14/12/2023).

[17] C.-W. Lan, C.-J. Wu, H.-J. Shen, H.-C. Lin, P.-J. Chu and C.-F. Hsu, 'Development of the Autonomous Battery Replacement System on the Unmanned Ground Vehicle for the Drone Endurance,' in *2020 International Conference on Fuzzy Theory and Its Applications (iFUZZY)*, ISSN: 2377-5831, Nov. 2020, pp. 1–4. DOI: 10.1109/iFUZZY50310.2020.9297363. [Online]. Available: https://ieeexplore.ieee.org/document/9297363#citations (visited on 12/12/2023).

[18] T. Addabbo, S. De Muro, G. Falaschi *et al.*, 'An Automatic Battery Recharge and Condition Monitoring System for Autonomous Drones,' in *2020 IEEE International Workshop on Metrology for Industry 4.0 & IoT*, Jun. 2020, pp. 1–5. DOI: 10.1109/MetroInd4.0IoT48571.2020.9138314. [Online]. Available: https://ieeexplore.ieee.org/document/9138314/references#references (visited on 12/12/2023).

[19] D. Raveendhra, M. Mahdi, R. Hakim, R. Dhaouadi, S. Mukhopadhyay and N. Qaddoumi, 'Wireless Charging of an Autonomous Drone,' in *2020 6th International Conference on Electric Power and Energy Conversion Systems (EPECS)*, ISSN: 2642-648X, Oct. 2020, pp. 7–12. DOI: 10.1109/EPECS48981.2020.9304971. [Online]. Available: https://ieeexplore.ieee.org/document/9304971 (visited on 16/12/2023).

[20] M. Simic, C. Bil and V. Vojisavljevic, 'Investigation in Wireless Power Transmission for UAV Charging,' *Procedia Computer Science*, Knowledge-Based and Intelligent Information & Engineering Systems 19th Annual Conference, KES-2015, Singapore, September 2015 Proceedings, vol. 60, pp. 1846–1855, Jan. 2015, ISSN: 1877-0509. DOI: 10.1016/j.procs.2015.08.295. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050915024229 (visited on 15/12/2023).

[21] Z. Tongyu, X. Aihua and S. Rui, ‘Evaluation on user range error and global positioning accuracy for GPS/BDS navigation system,’ in *Proceedings of 2014 IEEE Chinese Guidance, Navigation and Control Conference*, Aug. 2014, pp. 680–685. DOI: 10.1109/CGNCC.2014.7007297. [Online]. Available: https://ieeexplore.ieee.org/document/7007297 (visited on 04/04/2024).

[22] T. Takasu and A. Yasuda, ‘Development of the low-cost RTK-GPS receiver with an open source program package RTKLIB,’ en, 2009.

[23] T. Dutrannois, T.-T. Nguyen, C. Hamesse, G. De Cubber and B. Janssens, ‘Visual SLAM for Autonomous Drone Landing on a Maritime Platform,’ in *2022 International Symposium on Measurement and Control in Robotics (ISMCR)*, Sep. 2022, pp. 1–7. DOI: 10.1109/ISMCR56534.2022.9950582. [Online]. Available: https://ieeexplore.ieee.org/document/9950582 (visited on 16/12/2023).

[24] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel and J. D. Tardós, ‘ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM,’ *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021, Conference Name: IEEE Transactions on Robotics, ISSN: 1941-0468. DOI: 10.1109/TRO.2021.3075644. [Online]. Available: https://ieeexplore.ieee.org/document/9440682 (visited on 16/12/2023).

[25] M. Ferrera, A. Eudes, J. Moras, M. Sanfourche and G. Le Besnerais, ‘OV^{2}SLAM: A Fully Online and Versatile Visual SLAM for Real-Time Applications,’ *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1399–1406, Apr. 2021, Conference Name: IEEE Robotics and Automation Letters, ISSN: 2377-3766. DOI: 10.1109/LRA.2021.3058069. [Online]. Available: https://ieeexplore.ieee.org/document/9351614 (visited on 16/12/2023).

[26] M. Kalaitzakis, B. Cain, S. Carroll, A. Ambrosi, C. Whitehead and N. Vitzilaios, ‘Fiducial Markers for Pose Estimation,’ en, *Journal of Intelligent & Robotic Systems*, vol. 101, no. 4, p. 71, Mar. 2021, ISSN: 1573-0409. DOI: 10.1007/s10846-020-01307-9. [Online]. Available: https://doi.org/10.1007/s10846-020-01307-9 (visited on 04/04/2024).

[27] E. Olson, ‘AprilTag: A robust and flexible visual fiducial system,’ in *2011 IEEE International Conference on Robotics and Automation*, ISSN: 1050-4729, May 2011, pp. 3400–3407. DOI: 10.1109/ICRA.2011.5979561. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/5979561 (visited on 16/12/2023).

[28] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas and M. J. Marín-Jiménez, ‘Automatic generation and detection of highly reliable fiducial markers under occlusion,’ *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, Jun. 2014, ISSN: 0031-3203. DOI: 10.1016/j.patcog.

2014.01.005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320314000235 (visited on 30/04/2024).

[29] B. Benligiray, C. Topal and C. Akinlar, 'STag: A stable fiducial marker system,' *Image and Vision Computing*, vol. 89, pp. 158–169, Sep. 2019, ISSN: 0262-8856. DOI: 10.1016/j.imavis.2019.06.007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0262885619300903 (visited on 27/04/2024).

[30] P. Sturm, 'Algorithms for plane-based pose estimation,' in *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*, vol. 1, 2000, 706–711 vol.1. DOI: 10.1109/CVPR.2000.855889.

[31] M. Krogius, A. Haggenmiller and E. Olson, 'Flexible Layouts for Fiducial Tags,' en, in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Macau, China: IEEE, Nov. 2019, pp. 1898–1903, ISBN: 978-1-72814-004-9. DOI: 10.1109/IROS40897.2019.8967787. [Online]. Available: https://ieeexplore.ieee.org/document/8967787/ (visited on 08/02/2024).

[32] F. Romero-Ramirez, R. Muñoz-Salinas and R. Medina-Carnicer, 'Fractal Markers: A New Approach for Long-Range Marker Pose Estimation Under Occlusion,' *IEEE Access*, vol. PP, pp. 1–1, Nov. 2019. DOI: 10.1109/ACCESS.2019.2951204.

[33] G. Schweighofer and A. Pinz, 'Robust Pose Estimation from a Planar Target,' *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2024–2030, Dec. 2006, Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence, ISSN: 1939-3539. DOI: 10.1109/TPAMI.2006.252. [Online]. Available: https://ieeexplore.ieee.org/document/1717461 (visited on 05/04/2024).

[34] Z. Wang, H. She and W. Si, 'Autonomous landing of multi-rotors UAV with monocular gimbaled camera on moving vehicle,' in *2017 13th IEEE International Conference on Control & Automation (ICCA)*, ISSN: 1948-3457, Jul. 2017, pp. 408–412. DOI: 10.1109/ICCA.2017.8003095. [Online]. Available: https://ieeexplore.ieee.org/document/8003095 (visited on 12/12/2023).

[35] Y. Zhu, H. Pei, L. Wang, Y. Huang and K. Ou, 'A Vision-Based Autonomous Landing Method on Mobile Platform for UAV,' in *2023 42nd Chinese Control Conference (CCC)*, ISSN: 1934-1768, Jul. 2023, pp. 4089–4094. DOI: 10.23919/CCC58697.2023.10240611. [Online]. Available: https://ieeexplore.ieee.org/document/10240611 (visited on 12/12/2023).

[36] A. Moura, J. Antunes, J. J. Martins *et al.*, 'Autonomous UAV Landing Approach for Marine Operations,' in *OCEANS 2023 - Limerick*, Limerick, Ireland: IEEE, Jun. 2023, pp. 1–10, ISBN: 979-8-3503-3226-1. DOI: 10.1109/OCEANSLimerick52467.2023.10244606. [Online]. Available: https://ieeexplore.ieee.org/document/10244606/ (visited on 11/12/2023).

[37] *PX4 User Guide (v1.14)*. [Online]. Available: https://docs.px4.io/v1.14/en/ros/ros2_comm.html (visited on 11/04/2024).

[38] *Gazebo*. [Online]. Available: https://gazebosim.org/home (visited on 01/05/2024).

[39] S. Macenski, T. Foote, B. Gerkey, C. Lalancette and W. Woodall, 'Robot operating system 2: Design, architecture, and uses in the wild,' *Science Robotics*, vol. 7, no. 66, eabm6074, 2022. DOI: 10.1126/scirobotics.abm6074. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.abm6074.

[40] *SDFormat Home*. [Online]. Available: http://sdformat.org/ (visited on 01/05/2024).

[41] F. Romero-Ramirez, R. Muñoz-Salinas and R. Medina-Carnicer, 'Speeded Up Detection of Squared Fiducial Markers,' *Image and Vision Computing*, vol. 76, Jun. 2018. DOI: 10.1016/j.imavis.2018.05.004.

[42] G. Bradski, 'The OpenCV Library,' *Dr. Dobb's Journal of Software Tools*, 2000.

[43] *REP 103 – Standard Units of Measure and Coordinate Conventions (ROS.org)*. [Online]. Available: https://www.ros.org/reps/rep-0103.html (visited on 28/04/2024).

[44] *Beaufort wind force scale*, en. [Online]. Available: https://www.metoffice.gov.uk/weather/guides/coast-and-sea/beaufort-scale (visited on 22/04/2024).

[45] P. Jin, P. Matikainen and S. S. Srinivasa, 'Sensor fusion for fiducial tags: Highly robust pose estimation from single frame RGBD,' in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, ISSN: 2153-0866, Sep. 2017, pp. 5770–5776. DOI: 10.1109/IROS.2017.8206468. [Online]. Available: https://ieeexplore.ieee.org/document/8206468 (visited on 29/03/2024).

[46] U. C. A. Authority, 'The Drone and Model Aircraft Code,' en, 2023. [Online]. Available: https://register-drones.caa.co.uk/drone-code.

[47] U. C. A. Authority, 'Non-Segregated BVLOS,' en, 2020. [Online]. Available: https://publicapps.caa.co.uk/modalapplication.aspx?appid=11&mode=detail&id=9294.

[48] C. R. Harris, K. J. Millman, S. J. v. d. Walt *et al.*, 'Array programming with NumPy,' *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020, Publisher: Springer Science and Business Media LLC. DOI: 10.1038/s41586-020-2649-2. [Online]. Available: https://doi.org/10.1038/s41586-020-2649-2.

[49] W. McKinney, 'Data Structures for Statistical Computing in Python,' in *Proceedings of the 9th Python in Science Conference*, S. v. d. Walt and J. Millman, Eds., 2010, pp. 56–61. DOI: 10.25080/Majora-92bf1922-00a.

[50] P. Virtanen, R. Gommers, T. E. Oliphant *et al.*, 'SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python,' *Nature Methods*, vol. 17, pp. 261–272, 2020. DOI: 10.1038/s41592-019-0686-2.

[51] J. D. Hunter, 'Matplotlib: A 2D graphics environment,' *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007, Publisher: IEEE COMPUTER SOC. DOI: 10.1109/MCSE.2007.55.

[52] M. L. Waskom, 'Seaborn: Statistical data visualization,' *Journal of Open Source Software*, vol. 6, no. 60, p. 3021, 2021, Publisher: The Open Journal. DOI: 10.21105/joss.03021. [Online]. Available: https://doi.org/10.21105/joss.03021.