David Swincer
swin0030 #2131710

Flinders University

School of Computer Science, Engineering and Mathematics

ENGR7700 Honours Thesis

Development for Landing an Autonomous Quadrotor Drone on an Autonomous Surface Vessel: Mission Control

David Swincer

Supervisor: Associate Professor Karl Sammut

16/10/201

Disclaimer

The following thesis and the project work it describes is solely the work of the author unless specifically stated otherwise by acknowledgement or reference.

Acknowledgements

The author would like to sincerely thank the following persons for the time, advice and guidance they provided without which this project would not have been possible:

- Karl Sammut
- Jonathan Wheare
- Greg Ruthenbeck
- Andrew Lammas

The author also thanks Flinders University for the use of the facilities and budget for this project. Finally, the author would like to thank his project partners, Stephen Keen and Wahyu Prasetyono for the encouragement and teamwork that made this project so enjoyable.

Abstract

Autonomous vehicles have come a long way in the last decades, one of the most obvious cases being that of the Unmanned Aerial Vehicle (UAV). With increasing flight times and decreasing costs, the drone is becoming a commonly used instrument in many fields. On the water, Autonomous Surface Vessels (ASV) have seen similar growth, but now look to the skies to expand their capabilities. In this project, the aim is to autonomously launch and retrieve a UAV from the Flinders University TopCat ASV. The objective of this specific research is to design the high-level control software which provides guidance over the drone's behaviour. Making use of multiple sensors aboard the UAV and ASV, the control software is required to plot the route of the drone, decide when it is launched and retrieved and attempts to ensure the safe operation of the drone at all points. The research will look into the commonly used methods for localizing the drone and control structures for operating it. Finally, the software will be implemented on a physical quadcopter and tested via simulation and practice to ensure its viability.

Table of Contents

1. Introduction

With the increase in capability and availability of high-speed computers in compact packages, the development of autonomous vehicles is a rapidly growing field. While in the past the size of a computer capable of handling the complex algorithms required for an autonomous robot to process its environment may have prevented it from being carried on board a moving platform, this is no longer the case. As a result, technology developers are seeing the possibilities for autonomous robots to be used in a massive spectrum of applications. These range from the mundane and repetitive to the dangerous and unpredictable tasks that humans cannot, will not or would prefer not to undertake.

The Autonomous Surface Vessel (ASV) is one facet of this field of rapid growth, with many new designs and applications being attempted. Some of the uses that ASVs are being created for include search and rescue missions, coastline mapping, ecological surveys, intelligence gathering and defence applications. Any autonomous platform is expensive and complex, and as such most are expected to be able to not only perform one task but multiple roles. With such a variety of expectations placed upon them, the ASVs of the future need to be capable in activities beyond the scope of simply a boat on the water.

The Flinders University Autonomous Surface Vessel Project run through the Centre for Maritime Engineering, Control and Imaging is one such programme that would benefit from expanding the capability of its ASV. Built upon the Wave Adaptive Modular Vessel (WAM-V) platform, this ASV sometimes referred to as the “TopCat” is a catamaran research platform carrying a plethora of sensor and communication arrays. These include radar, lidar, a directional Global Positioning System (GPS) and multiple cameras. Some of its key practical uses are to provide critical ecological and topological monitoring data to organizations like the School of the Environment and provide a testbed for maritime research in the areas of mission planning and control systems. Other specific capabilities are developed to meet the requirements for international competitions such as the Maritime RobotX Challenge which the TopCat has been involved in with success in 2014 and 2016.

One issue that the TopCat ASV faces like many other autonomous systems is the limit of its perception of its environment due to the range of its sensors. The ability to have detailed and accurate information of the world around is essential to the ability of an autonomous system to find the most efficient and reliable method for achieving its goal. Hence, the fact that many of the sensors currently used on the TopCat ASV, such as lidar and cameras, are limited to line of sight from the vessel not only affects the data it can gather but its ability to perform any mission. This is exacerbated by the fact that a boat is small and close to the surface, meaning the effective ranges of it sensors are not as large as desired and can be easily blocked by relatively low obstructions. Some of this issue can be reduced by using pre-loaded maps to allow path and mission planning to be carried out when the ASV can only perceive a small part of the environment but still locate itself in its belief based upon the features it can extract. However, this system has little flexibility and such dependencies rule out the ASV as an option in situations where pre-existing knowledge of the environment is not available or could be rendered useless by occurrences like a natural disaster.

Another autonomous vehicle whose technology has advanced dramatically over the last decade is the Unmanned Aerial Vehicle (UAV). Light, commonly available and increasingly cheap, the UAV sometimes referred to as a drone is fast becoming the sensor platform of choice in a wide variety of situations. A very common application for the multirotor drones of today is gathering visual data from the environment through one or more cameras. Due to the altitudes a UAV can reach, the field of view from these platforms is extensive, making them a tool of choice for surveying wide areas of terrain, inspecting large structures or even creating 3-dimensional maps of buildings.

This kind of information would be of great value to the operations of the ASV as it removes the limitation of data gathering being only from within the line-of-sight of the ASV. For example, vertical look-down real-time images gathered from a UAV flying in the vicinity of the ASV could provide the TopCat's mission planner with important facts about where navigable water course can be found. A UAV can also gather mission goal data such as large-scale images of coastline and ecological features from a perspective and distance that would be impossible from the ASV. This combination of advantages to both the awareness and capability of the ASV make working with a UAV an attractive proposition.

UAVs suffer several major drawbacks such as limited range, flight time and payload. While advances are being made in battery capacity and lighter structural materials, there is no doubt that the requirement of a UAV to fly severely restricts it in ways less felt by the its surface-born counterparts. The TopCat carries batteries capable of operating the ASV for hours on end and propelling it distances in the range of tens of kilometres. This makes the platform ideal for operating in remote locations and capable of spending a lot of time gathering its data. A UAV capable of similar range and endurance would be prohibitively large and expensive to operate for the rewards it could offer.

An alternative being explored in this project is to use a UAV that is launched from the ASV, thus requiring a much-reduced flight time. Instead of having to fly for the duration of the ASV's mission and travel the same distance that it does, the UAV in this situation would only be air-born for the periods in which the ASV required advanced sensing. The obvious challenge to overcome in this scenario is how to launch and retrieve a UAV from the TopCat autonomously, as the ASV needs to capable of full operational ability without the intervention of a human controller.

The focus of the project that this thesis is based upon is to design and implement a UAV which is capable of taking off and landing upon the TopCat ASV. The specific section of this project that this thesis addresses is the mission control algorithm which oversees when the UAV is launched, the mission that it carries out, positioning for attempted landings and emergency safety procedures. The scope of the work in this thesis does not include significant investigation into planning the actual mission that the UAV carries out between launch and retrieval but rather focuses on facilitating the difficult task of landing the UAV on the ASV which is covered in other theses.

The main focus of the research for this project is into the forms of localization and control that are most effective when guiding a UAV. While no one source of data is able to provide all the information required to estimate pose in every situation, a merging of data from various sources such as visual tracking, lidar and GPS has been shown to achieve strong results. In the same manner, the most common forms of control algorithms can be assessed. Although these technologies are established in industry, the knowledge gap to be addressed here is specific to Flinders University. As the first project of this kind, the focus is to evaluate the most effective methods to date and implement them in the specific hardware and firmware system that is both already in place and being developed.

At the beginning of this project, the only prior work that could be built upon was the TopCat ASV. The UAV was custom designed and built for this project, as were all the supporting software systems and control algorithms.

The research question that will be addressed in this work is “What considerations must be evaluated before and during the launch, flight and retrieval of an autonomous UAV from an autonomous ASV and how can these considerations be contained within mission control software that provides safe and reliable oversight of the drone’s operation?”

The aim of this research is to investigate how similar projects have handled the mission control side of launching and retrieving an autonomous UAV from a water-born platform and what factors have been taken in consideration. It also aims to consider a number of control approaches and choose an appropriate model or combination of models based upon the specific needs of this project. The main qualities that this system is expected to display are reliability and low latency. This research aims to choose a suitable software structure and develop the high-level mission control algorithm for the UAV being launched from the ASV. It aims to test and evaluate the ability of this software to safely and reliably control the flight plan of the UAV. Finally, this research aims to allow interaction with other sections of this project to construct a single system able to meet the goals of launching, flying and retrieving the UAV from the ASV.

It is hypothesized that another autonomous vehicle working in collaboration with and sharing sensor data with the TopCat would be step towards improving the ASV’s knowledge of its environment and thus its ability to achieve its mission goals.

It is hypothesized that a suitably scaled UAV launched and retrieved by the ASV would be capable of providing useful mission data to the ASV, thus improving its efficiency and capability.

It is hypothesized that an autonomous UAV launched, flown and retrieved from the TopCat ASV can be safely and reliably controlled at a high level by a state-machine type mission planner that is capable of assessing the operating conditions and state of the ASV and UAV and make independent decisions about the immediate goals of the drone.

The designed software is initially tested using control flow analysis tools to highlight any inadequacies or shortcomings in the control structure. The individual sections of the software that evaluate the operating conditions are tested against field data gathered from both the ASV and UAV. Further testing is completed on the combined control algorithm by simulation of random conditions that the control structure might be exposed to. The mission objectives and actions that the high-level planner requests of the low-level drone control are tested in the field to ensure that the UAV is capable of achieving these actions consistently. Test data from the field is used to generate the missions the high-level planner would provide to the UAV.

In the section two, approaches from other researchers working in the same field will be described and their features evaluated. Their capabilities are used to choose a suitable structure or combination of structures that would best suit the goals of this research project. In the next section, the design of the high-level control algorithm and its components is described along with details of how these sections were tested. The results of these tests are then recorded and discussed together with an overview of how the software was revised and improved as a result in section four. Shortfalls in both the testing methods used and the capabilities of the system designed are considered. The continuing issues identified are used as the basis for a discussion of the future works that could be completed in this aspect of the project along with recommendations for the direction of study. Finally, an overview of the main accomplishments of this research are provided in the concluding section five.

2. Literature Review

When planning the control system of the drone, there are several key areas to consider. The first is understanding the location of the UAV in reference to other objects in the world. A means for detecting and localizing pose is required. Another important area that needs to be assessed is how the drone is guided during flight. The final area to be evaluated is the design of the software which controls the UAV during flight and makes decisions about its objectives and behaviour.

2.1 UAV Location System

A range of technologies are being used to calculate the pose of UAVs within their operating environment. These systems include the use of visual tracking, Inertial Navigation System (INS) and Global Positioning System (GPS) (B B V L & Singh 2016).

2.1.1 Visual Tracking

One of the most common issues found with traditional pose tracking options is that they rely on data from an external beacon or source which is not always available (Anitha & Kumar 2012; Kong et al. 2014). To gain full autonomy, the use of visual guidance has been exploited to contain the tracking of the UAV to on-board systems (Ajmera et al. 2015; Anitha & Kumar 2012; Araar, Aouf & Vitanov 2017; Saripalli, Montgomery & Sukhatme 2002) . In Ajmera's and Araar's research, the camera was mounted aboard a UAV and used to estimate the pose of the UAV relative to designated symbols. One of the major advantages of taking this approach is that the pose estimate is always relative to the distinguishing ground symbols. This can be exploited when landing on a moving platform which is a high-level objective of this project. It tends to be a robust system that tolerates considerable noise once a fix has been acquired (Araar, Aouf & Vitanov 2017). Night time visual guidance can be achieved by using illuminated beacons on the ground and algorithms implemented which negate the need to calibrate the camera (Anitha & Kumar 2012).

At the same time, this approach with an on-board camera has the negative side effect of requiring that the pose of the UAV is estimated off-board due to the high image processing costs (Ajmera et al. 2015). This leads to losses in latency and requiring a constant radio link to the base station. Another major problem observed with this technique is that it relies on the use of specific patterns on the landing area which can be undesirable. The system also struggles by itself to recover a pose estimate if it is applied at a distance from the landing platform (Saripalli, Montgomery & Sukhatme 2002). In addition, depending upon the quality of the camera, the altitude from which a pose estimate can be calculated is a limiting factor (Ajmera et al. 2015).

Other researchers take the approach of using natural landmarks to localize the UAV (Cesetti et al. 2009; Conte 2009). The obvious advantage of this method is that it does not require an artificial beacon for the UAV to estimate its pose. However, this approach requires the environment is at least somewhat previously mapped so that the vision from the UAV can be correlated with known landmarks.

Research shows that image processing algorithms are always costly and high definition cameras are heavy to mount on a UAV (Kong et al. 2014). A means to mitigate this issue is to mount the camera looking upwards from the ground where heavy cameras and larger processors are suitable. A number of researchers have tried this method with varying degrees of success (Abbeel, Coates & Ng 2010; Pebrianti et al. 2010; Tang et al. 2016; Wang et al. 2006). Abbeel et al. were able to use multiple ground based cameras to locate the position of the UAV at an altitude of forty metres with an accuracy of twenty-five centimetres. However, the negative of this approach is that the field of view from the upward looking camera is small and another means of localization is required when the drone is outside the camera's vision. This issue can be reduced however by the use of an omnidirectional camera lens (Kim et al. 2014) or visual servoing (Lee, Ryan & Kim 2012).

As summarized by Kong in his 2014 survey on the topic, the best use for visual-based UAV localization is during the high precision landing phase. During normal operations however, high precision localization is not essential and the difficulties encountered locating the UAV in the world space with a vision approach are numerous.

2.1.2 Inertial Navigation System with Onboard Sensors Tracking

The sole use of the inertial measurement unit (IMU) to determine the pose of a UAV is not considered viable by most researchers. Instead, they tend to combine the data received from the IMU with other sensors to gain a more accurate understanding of the surroundings.

One approach is to use the natural landmarks in the UAV's operating environment to localize the drone (Anthony & Detweiler 2017). In this specific instance, the UAV was operating in row crops and used a laser scanner in conjunction with its IMU to be able to determine pose within a single crop row. More precise than uncorrected GPS and well-suited to the out-door environment, the laser scanner is capable of operating in a variety of lighting conditions. Although the UAV uses GPS to augment its navigation systems, the precision comes from the ability to estimate the shapes in the environment allowing accurate agricultural readings. This approach allows the UAV to operate in GPS restricted environments which is a desirable quality. However, the project does have several shortcomings including the loss of precision when the drone is at altitude and its reliance on known regular landmarks which are not common in a marine environment. In addition, the use of lidar is prohibitively expensive in certain situations.

The data fusion of the IMU with an ultrasonic sensor has also been attempted (Santos et al. 2017). This study revealed similar results to (Anthony & Detweiler 2017) in that the IMU and ultrasonic sensor provided precise local pose estimates but in the larger world required another reference sensor. This was provided in the particular project by a depth camera. The main advantage of this system over others is that it is cheap. However, it has not been proven outdoors and the research did not address the problem of localization in the relatively feature free marine environment.

One of the most difficult tasks to accomplish in UAV technology is the ability to operate efficiently in both the indoor and outdoor environment (Tomic et al. 2012). However, another approach which is potentially capable of achieving this integrates a laser scanner and stereo cameras with the IMU data to navigate accurately in GPS denied environments as well as open areas (Tomic et al. 2012). The result is a drone that is capable of moving through small unmapped spaces by fusing data in three dimensions. While not aimed at solely outdoor flying, this research supports the view that vision-based tracking is one of the most accurate localization techniques at close range.

Similar research shows that this success is not just the result of the use of stereo cameras (Achtelik et al. 2011). In addition, Achtelik's research was carried out using entirely on-board processing, decreasing latency in the control loop to speeds which approached the response ability of the UAV platform. A monocular camera also has a weight advantage over the stereo camera and lidar system used by (Tomic et al. 2012). By filtering and correlating the visual and IMU data, a Simultaneous Localization and Mapping (SLAM) algorithm could be run on the UAV and used to control the motion of the vehicle with full autonomy. One of the key outcomes from this research was the stability this approach achieved in unmapped environments. Similar to a number of other approaches that focus on fusing sensor data with IMU data however, there was little evidence of the system being tested at height in an open environment.

2.1.3 Global Positioning System Tracking

To overcome the shortcomings while retaining the advantages of the visual based pose estimation, some researchers chose to combine the visual tracking with alternative location systems such as GPS. While GPS lacks precision and when uncorrected can only give pose estimates with an accuracy of approximately 0.5 to 2 metres (Kong et al. 2014), its capability in otherwise landmark free environments is desirable.

This property is highlighted in an approach that uses both the visual and GPS guidance systems (Hermansson, Skoglund & Schön 2010). This merging recognizes the broad-scale ability of GPS to track the pose of the UAV in the wider environment as well as its failing to provide the precision required to consistently lad the UAV on the target. In complementary fashion, the vision-based system has high precision at close range but loses this property when the UAV is at a distance or outside the field of view. The result of this union was a stable system that operated well in the field environment similar to that which is the focus of this project. Restricting factors in this research included the quality of the camera that could be mounted on the drone due to weight constraints and the fact the target was stationary.

Another project which used a similar approach was (Pinto, E. et al. 2014; Pinto, Eduardo et al. 2014) which combined an upward looking visual guidance system with GPS location to provide both wide scale localization and close-range precision for landing. The results from this research was particularly impressive as it was carried out in the marine environment which is the ultimate goal for this project. In addition, it showed the effectiveness with which an upward looking camera operated without the need to add that weight to the UAV's payload.

2.2 UAV Control System

2.2.1 State Machines

Although the pose of the drone is a critical piece of information when planning how an autonomous drone is to behave, there are many other pieces of information which must be considered. To be able to make sense of the flow of control and behaviour in the UAV, a control structure must be implemented.

In the case of (Araar, Aouf & Vitanov 2017), a "landing procedure" needed to be implemented. This control structure was designed to make decisions about the rate of descent and as the UAV approached the target. The information considered revolved around how well centred the UAV was over the camera and how well the fiducials were being detected. This was implemented in a state machine type behaviour. A significant issue with this process however was the lack of flexibility it had in the latter landing stages. However, it was a simple and clear how the UAV would operate and was capable of recovering the UAV in the case the camera lost track of the UAV pose.

The main goal of the work of (Tomic et al. 2012) was to implement a fully autonomous UAV. Hence, the cognition process involved a number of state machines more complex than those in (Araar, Aouf & Vitanov 2017)'s research. Each state machine had a specific high-level task and was called to control the actions of the UAV until it reached that goal. This structure proved reliable in complex environments where unexpected situations were encountered. However, the full details of how this type of control was implemented is not revealed.

2.2.2 Petri Nets

While a state machine gives the ability to plan the behaviour of a system, it lacks the ability to demonstrate that the state machine will behave as intended. In complex systems, it becomes essential to determine the properties of those systems when any input is provided.

One system that can analyse algorithms for safety is the Petri network (Zhou, DiCesare & Desrochers 1989). Using a mathematical algorithm to construct the state machine allows an equation to be formulated which has the same properties as the state machine. Any set of inputs can be supplied to this equation, allowing the designer to test the behaviour of the system and detect unsafe properties such as loops or dead-ends.

The ability to apply this type of control structure to real-time vehicle control problems is demonstrated by (Dotoli & Fanti 2004). The problem the research solved involved multiple vehicles which could move independently of one another. This behaviour can be likened to that of the properties of the UAV system which has features such as wind speed and pose which change completely independently as well. The petri network described by (Dotoli & Fanti 2004) is capable of providing a structure and hierarchy which determines safe real-time behaviour. One advantage of this approach is that it allows for a modular control structure which can be implemented to represent the different tasks as used by (Tomic et al. 2012).

3. Methodology and Results

3.1 Evaluation of Conditions

Before an effective mission planner could be designed, it required the input of timely and accurate data regarding the state of the UAV, ASV and the environment around them. The major factors that were considered for evaluation had to be readily available from the sensors on the two vehicles or have related data available which could be manipulated to provide useful information regarding their state. The critical information areas extracted from the system are the wind speed and direction, the pitch of the ASV, the battery status of the UAV and the global geographic location of the UAV. In combination with known operational parameters gained through testing and pre-loaded data from external sources, this information could be used by the mission planner to attempt to choose the safest course of action.

3.1.1 Wind Speed and Direction

Due to their light weight, multirotor UAVs are susceptible to even moderate winds. While the stabilizing functions built into the quadcopter’s flight controller is capable of handling the drone in gusts of wind, it quickly became apparent in test flights that the autonomous landing capability of the drone was decreased in the presence of even a steady breeze. In addition to this, the intended location of the UAV’s landing position on the TopCat ASV is aft and below the level of the main sensor array which contains expensive GPS, radar, lidar and cameras, creating a hazard when the UAV is in close proximity to the ASV. All these factors indicated that wind speed and direction are an important consideration when designing the mission planner.

The first tests considered what magnitude of winds the UAV was able to withstand during normal flight. Using wind speed estimates from the Bureau of Meteorology, assessment was made of the quadcopter’s performance in varying conditions over several test flights. It was found that while the quadcopter could withstand moderate sustained winds and light gusts, it struggled to maintain its course in sustained moderate winds and frequent gusts. This feedback suggested that at least two different measurements of the wind speed ought to be considered when deciding whether the conditions were safe for flying.

Speed	Average Wind Speed	Gust Wind Speed
0-5 m/s	Stable flight. Performed well when moving between waypoints with little or no sign of impedance.	Stable flight. Drone would pause/rock occasionally when struck with wind but swiftly recover
5-10 m/s	Mostly stable flight. Drone would pause occasionally when in autonomous mode.	Mostly stable flight. Drone would be moved off course but recover and continue.
10-15 m/s	Not Tested (Hazardous)	Unstable flight. Drone would turn into wind and lose altitude rapidly when hit with strong gusts.

Table 1. Estimation of the performance of the drone in varying wind conditions in manual flight and autonomous mode.

1) A running average of the wind speed which would give data on the sustained speed of the wind.
2) A record of the maximum wind speed detected.

The other main consideration when taking off from and landing on the TopCat was that the ASV was facing into the wind. There were two reasons for choosing this orientation. Firstly, when at all possible, the ASV was expected to remain stationary during the launch and retrieval process although this may change in the future. A stationary surface vessel retains is position best when it is facing into the wind and can use thrust from its motors to hold itself place. The second reason was due to the proposed location of the landing pad for the drone being below and aft the main sensor array. In this position, it was considered a collision between the vehicles was less likely if the prevailing winds were in a direction that would blow the UAV away from the ASV's main array.

The wind data was available from two different sources. The TopCat ASV possessed an anemometer and direction sensor which published the wind speed in metres per second and wind heading in degrees relative to the heading of the boat on a dedicated ROS topic. The UAV also had a wind sensor which provided the quadrotor's airspeed in two axes. This data could be used in combination with the heading and ground speed of the drone to calculate the wind speed. The ASV's wind data was chosen for several reasons.

- The wind speed was required to be measured before the drone was necessarily launched. In its landing position, it was possible the accuracy of the wind data it gathered would be impaired due to protecting features on the ASV. In contrast, the wind speed sensor on the ASV remained in the same exposed position at all times.
- For safe take-off and landing, the wind direction relative to the heading of the ASV needed to be measured. This data was available immediately from the ASV's wind sensor but would need to be calculated from the UAV's data. This would require a transformation from the quadrotor's frame of reference into that of the ASV. In addition, further inaccuracies would be introduced by the need to subtract ground speed from the quadrotor's air speed.
- The mission-planning software was located on the ASV. It was more convenient to use the data on the wired connection from the local wind sensor than the radio connection from the UAV which could be intermittent.
- The wind direction and speed at the landing point was more important than at the UAV as that was where the launch and retrieval would take place. The wind speed and direction at the location and altitude of the UAV was possibly different from that at the ASV.

To calculate a running average of the wind speed experienced by the ASV, a ROS node was created which subscribed to the ASV's wind data topic. The node read the latest wind data each second and recorded that data into a one hundred and twenty element array to hold wind speed from the last two minutes. The oldest data was removed each time a new measurement was taken. The wind speeds in this array were averaged to determine the mean wind speed. The maximum wind speed from the last two minutes of data was also found from this array.

Using the information displayed in Table 1, it was decided that if the wind speed average from the previous two minutes was less than 8 m/s, the conditions were safe for flying. If a wind speed of greater than 10 m/s had been recorded in the last two minutes, the conditions were deemed unsafe for flying. The results from these comparisons were combined and a Boolean published to the rest of the system stating whether the wind conditions allowed flight or not.

Subsequent tests as the project went on showed that the UAV's landing stage was much more susceptible to wind than the normal flight modes, requiring an update to conditions that were considered safe for flying.

Speed	Average Wind Speed	Gust Wind Speed
0-5 m/s	Mostly stable flight. At higher wind speeds, drift would become harder to counter if UAV started to the windward of landing area.	Stable flight. Drone would pause/rock occasionally when struck with wind but swiftly recover
5-10 m/s	Unstable flight. Landing system unable to correct fast enough.	Unstable flight. Landing system sometimes recover but not always
10-15 m/s	Not Tested (Hazardous)	Not Tested (Hazardous)

Table 1. Estimation of the performance of the drone in varying wind conditions when trying to land autonomously.

As a result of this data, the wind speed requirements were tightened to ensure that the UAV did not take off or attempt to land in conditions that the landing guidance system would not handle. The new restrictions for take-off were set to require an average wind speed of less than 5 m/s and a maximum gust speed of 7 m/s over the previous two minutes.

The direction of the wind was also monitored to ensure that take-off and landing were not attempted at times when the ASV was not facing into the wind. The ASV was considered to be facing into the wind if the heading was within 15° on either side of the bow of the ASV. This margin of error was considered reasonable due to the fact the area behind the proposed landing pad location on the ASV is entirely free of obstructions.

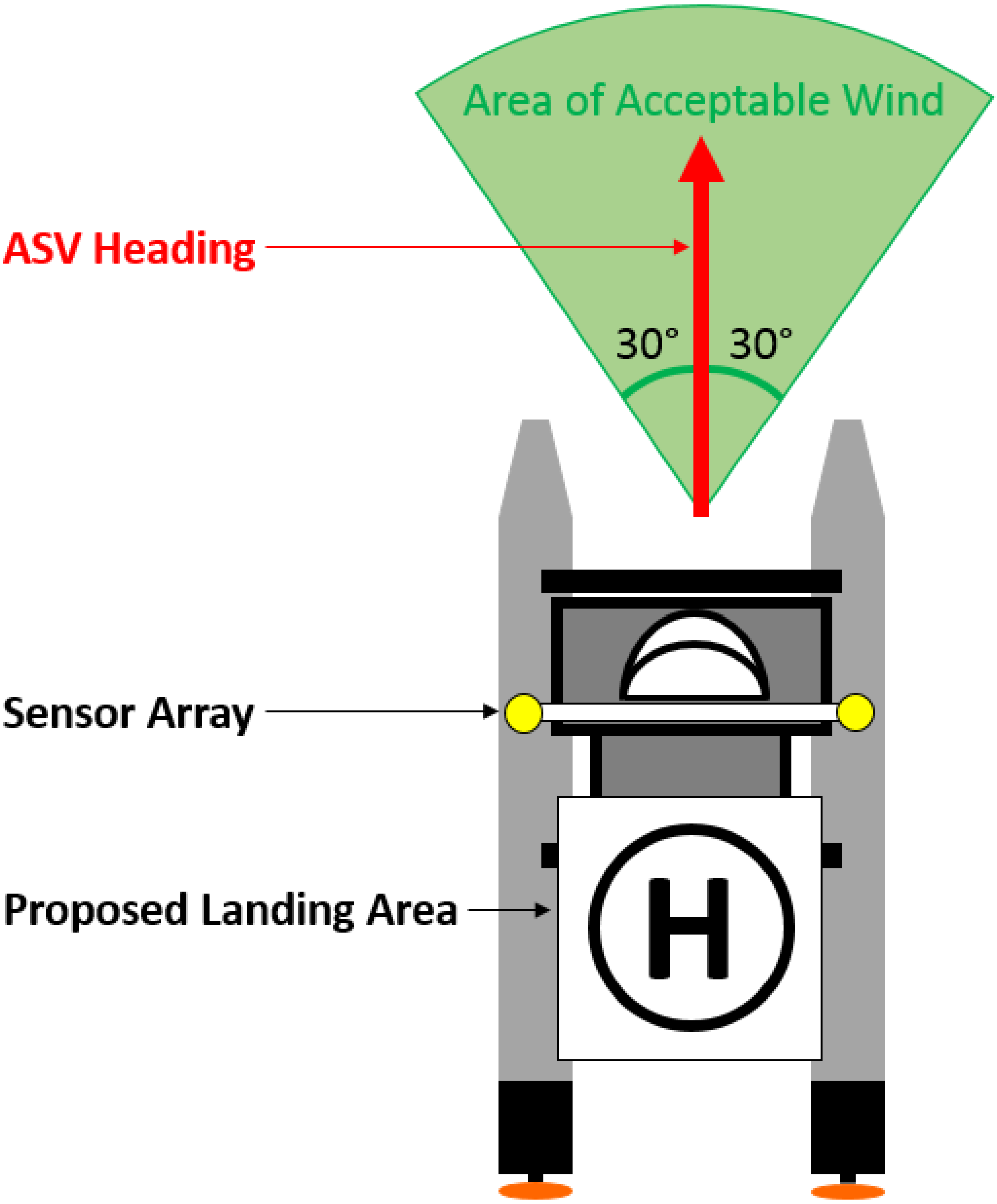


Figure 1 Diagram of the ASV with angles of wind suitable for landing and take-off shown in green.

Comments were left in the ROS wind node's code where a communication could be inserted to alert the ASV system that the drone was looking to take off or land and request the ASV halt its current task and face into the wind.

3.1.2 ASV Pitch

The ASV pitch was considered an important indicator of the water surface conditions. In rough weather or when the ASV is travelling at high speed, it is likely the ASV will tilt at angles that are unsafe for the UAV to attempt take-off and landing. Due to the late production of the landing pad and the latest landing gear for the UAV, which was not a major focus for the project, the safe angles at which the drone could land while reliably gripping the wire netting surface were not rigorously tested.

Despite this, a placeholder ROS node was created to maintain a peak value record of the maximum pitch from the ASV's IMU data. When pitch angles were detected that exceeded a set threshold, the conditions were declared unsafe for flying for two minutes. If no pitch angles were encountered over the set threshold in the following two minutes, the pitch conditions were deemed to be at a safe level once again. The maximum acceptable pitch allowed was set at 20 degrees, although this was largely an arbitrary value.

3.1.3 UAV Battery Status

Another key indicator that the mission planner had to monitor at all times was the status of the UAV's battery. This included the voltage level and the on-board flight controller's percentage estimate of the battery's remaining capacity. This information was published by the UAV on a MAVROS topic.

A ROS node was created to monitor this topic and alert the main mission controller when the battery capacity reach 40%. A second warning was also sent when the battery voltage level reached 10.25 Volts.

It was decided not to set a hard limit on what value was considered "safe to fly". The task of deciding the correct response to the battery level was left to the main mission planner as the reaction to a low battery situation differed depending upon the mode and stage of the UAV through its mission.

3.1.4 GPS Location and No-Fly Zones

There are many locations in which there are restrictions on the flying of UAVs, especially in metropolitan areas where airports and helipads are common. As the mission planner had control over when the UAV was to be launched, steps needed to be taken to ensure that the ASV was not in a no-fly area when the drone was launched and that the drone did not enter such an area during flight.

Many UAV manufacturers for the public include software restrictions in their products that prevent the user from flying into designated no-fly areas. These restrictions are called geo-fences and provide a level of certainty that the drone will fly within the regulations of the local district. However, the quadcopter that was custom built for this project did not contain any such data as it was a custom build. The ability to add geofences to the flight controller was possible through the Mission Planner software package.

While maps showing where the restricted airspaces are located can be found easily on the internet, the ownership of machine-readable no-fly maps and geofences is held by proprietary companies. Due to the difficulties and costs to find such maps, a different approach was taken. As most restricted airspaces can be simplified or expanded to generally take on the shape of a circle, it was decided to create an internal database of airports and no-fly area locations. These locations were stored in Universal Transverse Mercator (UTM) coordinates which was the local frame of the UAV in the mission planner. The data came from a number of sources including reference data in the mission planner software and Google Earth maps of the local area the UAV was being tested in.

Using the Civil Aviation Safety Authority's "Can I fly there?" web application, the maximum distance from the central coordinate to the edge of each no-fly zone was calculated using Google Maps.

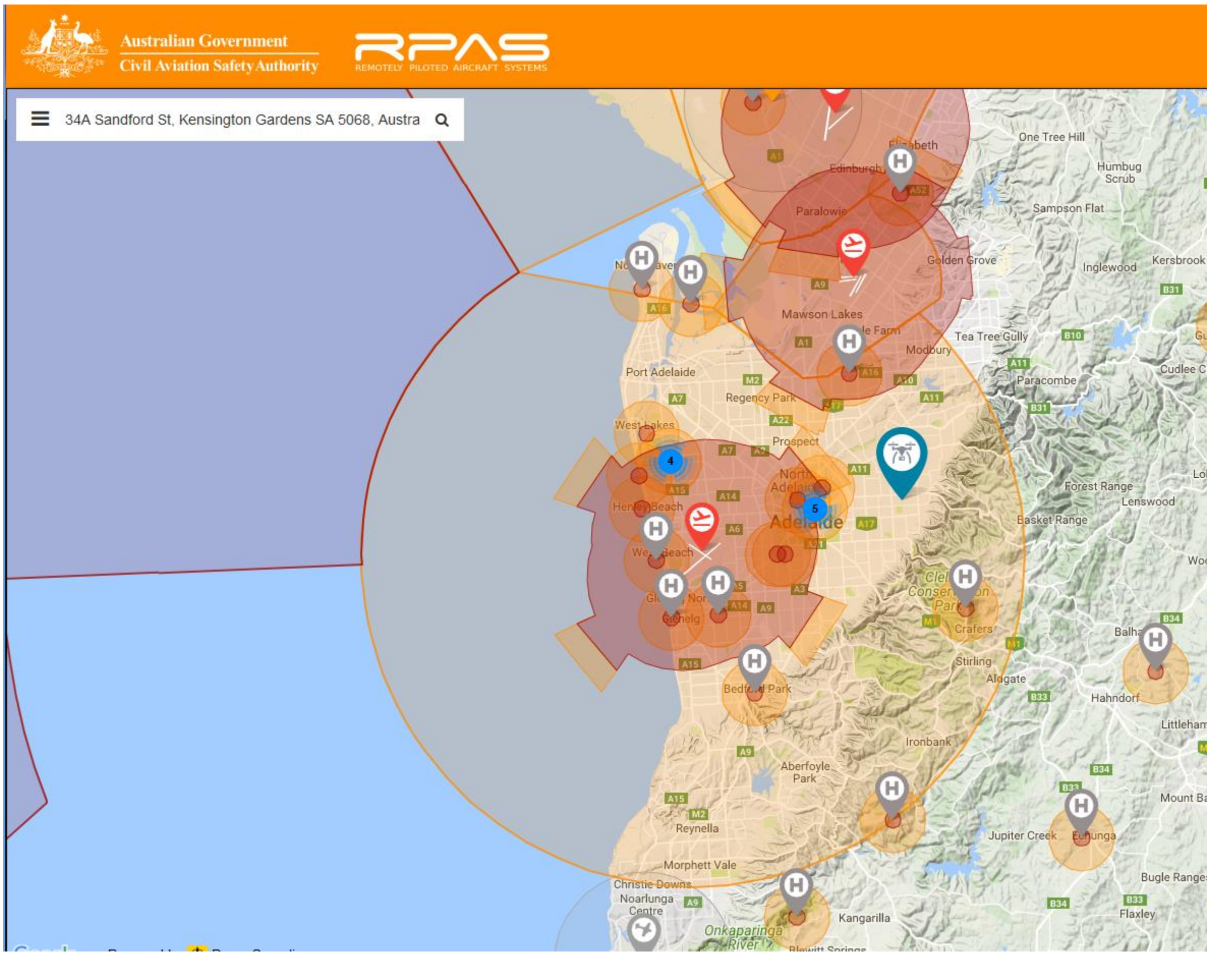


Figure 2 Screen capture from the CASA "Can I fly there?' web application showing some of the local no-fly zones that were used to prevent illegal UAV movement.

A ROS node was written subscribing to the drone's local position which was published on a MAVROS topic. Each second, the latest position of the drone was used to calculate the distance from the UAV to all the stored no-fly zone central points. The distance calculation was straightforward due to the UTM projection assuming a flat surface within each local region. As the scope of operations intended for testing the UAV were within a single UTM zone, airspace restrictions outside this zone were not considered. If the UAV's position was outside the maximum radius of every no-fly zone, the mission was permitted to continue but otherwise was aborted to ensure the drone either remained on the ASV or returned to a safe home position immediately.

The nodes used for the evaluation of conditions were tested alongside the main mission planner as shown in section 3.5.

3.2 UTM and GPS Conversions

One of the most important decisions to be made was which location system ought to be used as the main frame of reference for the UAV mission planner. The two main frames of reference considered were Global Positioning System (GPS) or UTM.

GPS had the major advantages of being a very universal system that is used by both the ASV and UAV. In addition, it is compatible with many other devices, making for easy comparison between the latitude and longitude of other locations. For example, the drone accepts waypoint files using coordinates in the GPS system only. On the other hand, GPS gives the location of the vehicle in degrees from geometric reference points. This system is difficult to use in close confines where changes in latitude and longitude do not linearly convert to distances.

The UTM system provided a cartesian coordinate system which assumes the ground is flat across the area of each of its grids. The units are in metres and allow for easy calculation of coordinates and thus waypoint planning.

The advantages of both systems were too beneficial to choose just one system, so it was decided to use both types of measurement. This raised the question of how to handle conversion between these frames of reference. Mathematical conversion was considered using various assumptions about the curvature and shape of the earth. However, none were found which provided a simple algorithm with sufficient accuracy. With the intention to be able to locate the UAV within half a metre, uncertainties introduced by the mathematics which could by tens of metres were unacceptable. A solution was found through using the proj4 library to convert between the UTM and GPS coordinates.

Using sample code from the TopCat ASV's mission planner as a basis for correct syntax, two ROS nodes were written to handle the conversion between GPS and UTM, one for each direction of conversion. Each node was connected to other nodes that needed its abilities by a service call rather than a normal topic.

3.3 Home Location Choice

In the event of an emergency or the definitive failure of the drone to complete its mission, the UAV must have a home location to which it knows it can travel and land safely. In most land-based UAVs, this location is the GPS position that the drone was launched from or where the flight controller was activated. When command of the UAV is lost through communications breakdown for example, this is the point to which the UAV will attempt to return.

In the situation of the UAV working in conjunction with the ASV, the normal home position is not suitable for a number of reasons. First of all, the home position at the point the UAV is first activated or launched is most likely to be on the launch platform on the back of the ASV. The ASV cannot be guaranteed to remain in the same location throughput the entirety of either its mission or the UAV's mission. As a result, this home location is entirely unsuitable as it is most likely a marine surface which the quadcopter cannot landing upon. In addition, the ASV is likely to travel considerable distance before the UAV might be launched and having a home position outside the range of the drone would become a major issue.

To overcome this issue, prior to the launch of the mission planner, a number of possible home locations need to be stored in the memory of the UAV in GPS format. These home locations need to be open spaces of flat ground without obstruction from above. They would preferably be close to the waterway the ASV is operating from and without tall obstacles in between. Coastal beaches for oceanic operations or the banks of estuaries and rivers would often be locations of choice. Accurate maps of the area the UAV will be operating in are essential for generating safe home positions.

A ROS node was written which calculated the distance between the location of the UAV and each of the preloaded potential home points. The closest point was chosen as the home location. If the home location changed, an alert was sent to the UAV's immediate control system with the coordinates of the new home position. This data was then intended to be used to update the home position set in the drone's flight controller and allow it to move to the best choice of home location.

This node was tested using data from a test flight in which the drone was flown in a straight path up an oval. The plot of the path the UAV took is shown in the following figure.

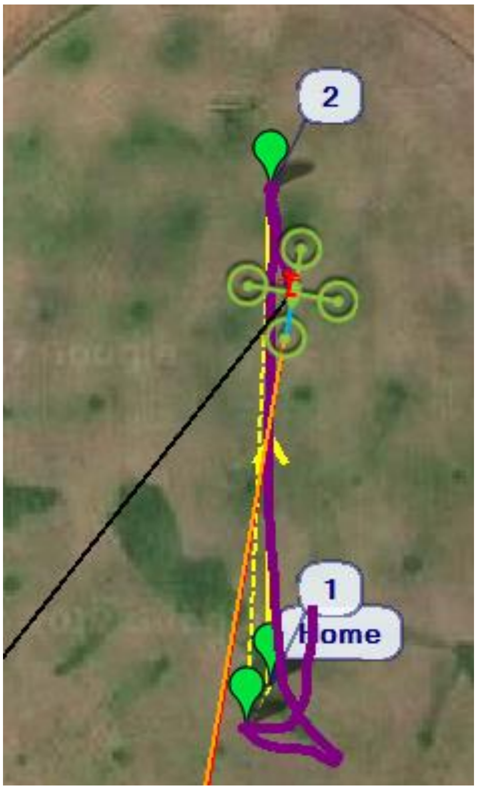


Figure 3 Movement of the UAV shown in purple moving up the field from waypoint 1 to waypoint 2

Two possible GPS home positions were set in the ROS node on either side of the course of the drone which can be seen in the following figure. The drone began it flight with its home positions set to the bottom left most green marker labelled as “Home”.

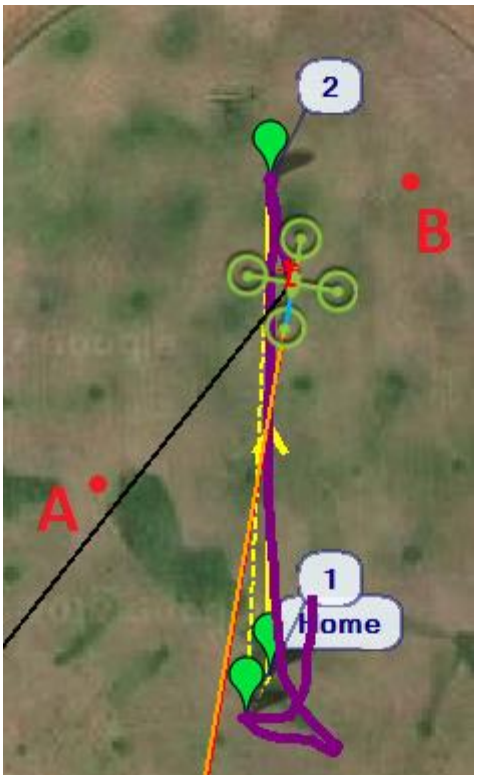


Figure 4 Potential home positions A and B shown in red relative to the course of the UAV.

As the drone flew between the two waypoints, the home position of the drone updated twice, as it neared different potential waypoints along it course and calculated they were closer than its current home position. The points at which the drone changed home positions are shown in the following figure.

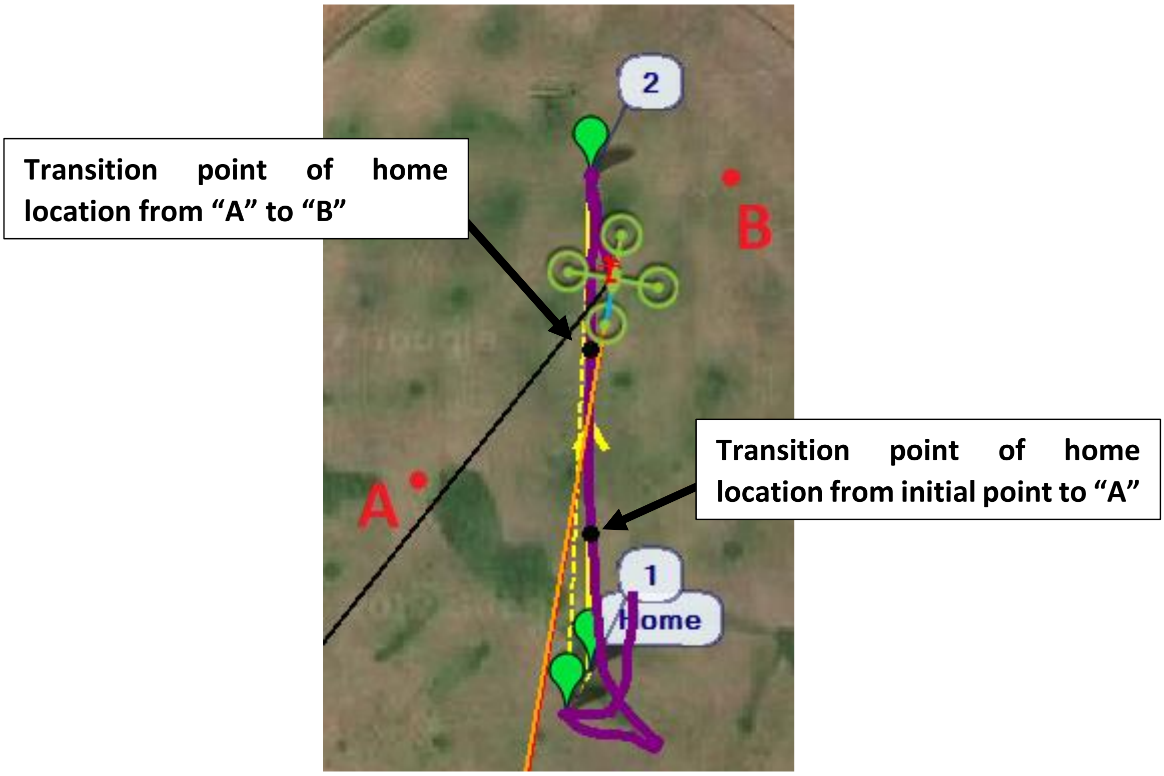


Figure 5 Position of the locations along its path where the UAV changed home positions shown in black, choosing the closest potential waypoint.

While being able to update the home position on the UAV to minimize ground distance to the home position was a major step, the distance that the drone was required to travel through the air and energy it had to expend was realized to be the more important factor. When considering this difference, wind speed and direction became major factors in choosing the correct home location.

Using the wind data from the ASV, a vector calculation was done to add a weight to any distance that needed to be travelled in the opposite direction to the wind and the cost of travelling with the wind was decreased. The process used to do this essentially warped the distances between the UAV and potential home locations to make it appear as if home locations downwind of the UAV were closer than they actually were while upwind locations were moved further away. The magnitude of this manipulation was dependent on the windspeed. This caused the drone to choose home locations which it could travel to more easily with the wind even if there was closer potential home location upwind. An example of the affect this improvement had on the where home locations were changed is shown in the following figure.

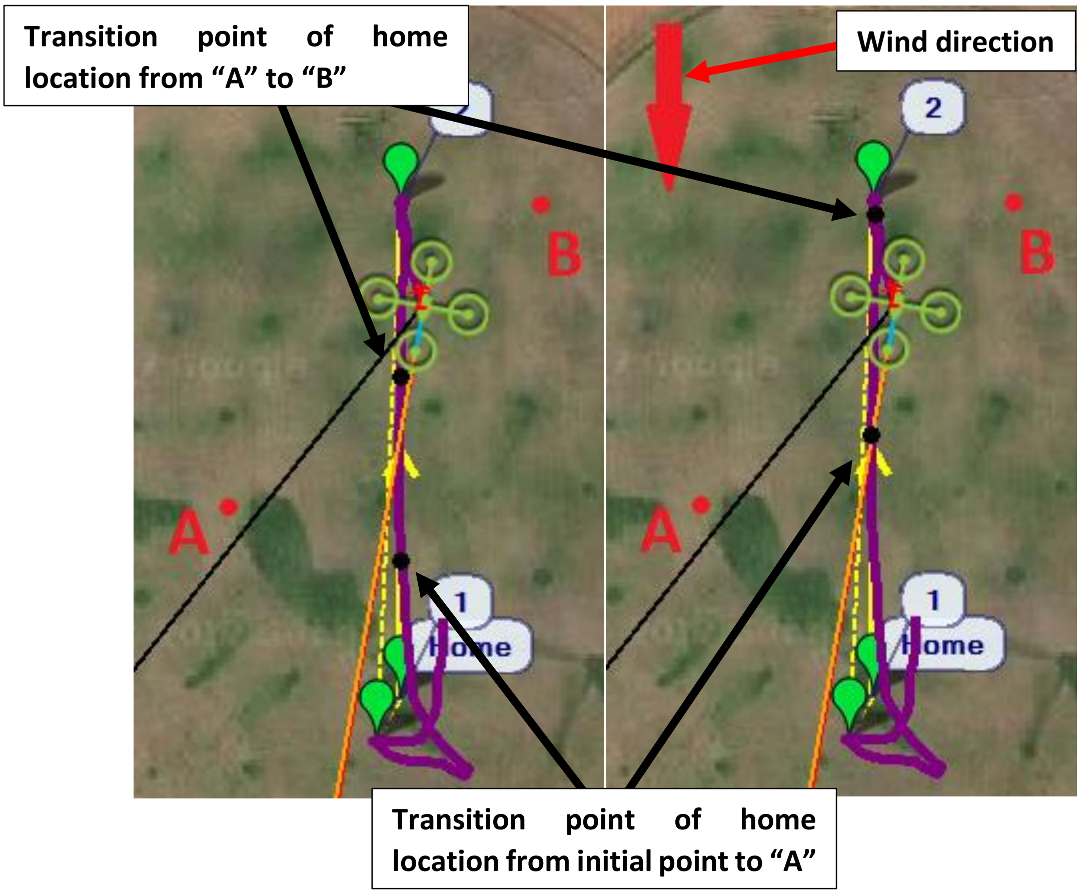


Figure 6 Comparison of the home position update points before and after wind compensation was included. On the left, without wind compensation and on the right with wind compensation.

As can be seen in the previous figure, the UAV travelled closer to the "A" potential home position than its original home position but did not update the home position due to the wind direction almost blowing it straight back towards the original home position, keeping it the option of lowest cost in the event of an emergency. However, as the drone moved further along its course, it finally became more efficient to travel to "A" than the original home position and updated the UAV's home position accordingly.

3.4 Mission Path Planning

3.4.1 Main Mission Path

Although the path that the UAV flew over during its mission was not the main focus of the project, some thought was put into choosing a flight plan which could serve as the basis for a realistic mission that the UAV could be expected to do for the ASV.

The first concept considered was to create a vector field around the boat which could be used to give velocity commands to the quadcopter. The centre of the vector field would be the position above the ASV where the landing algorithm could find it and guide it down the final approach.

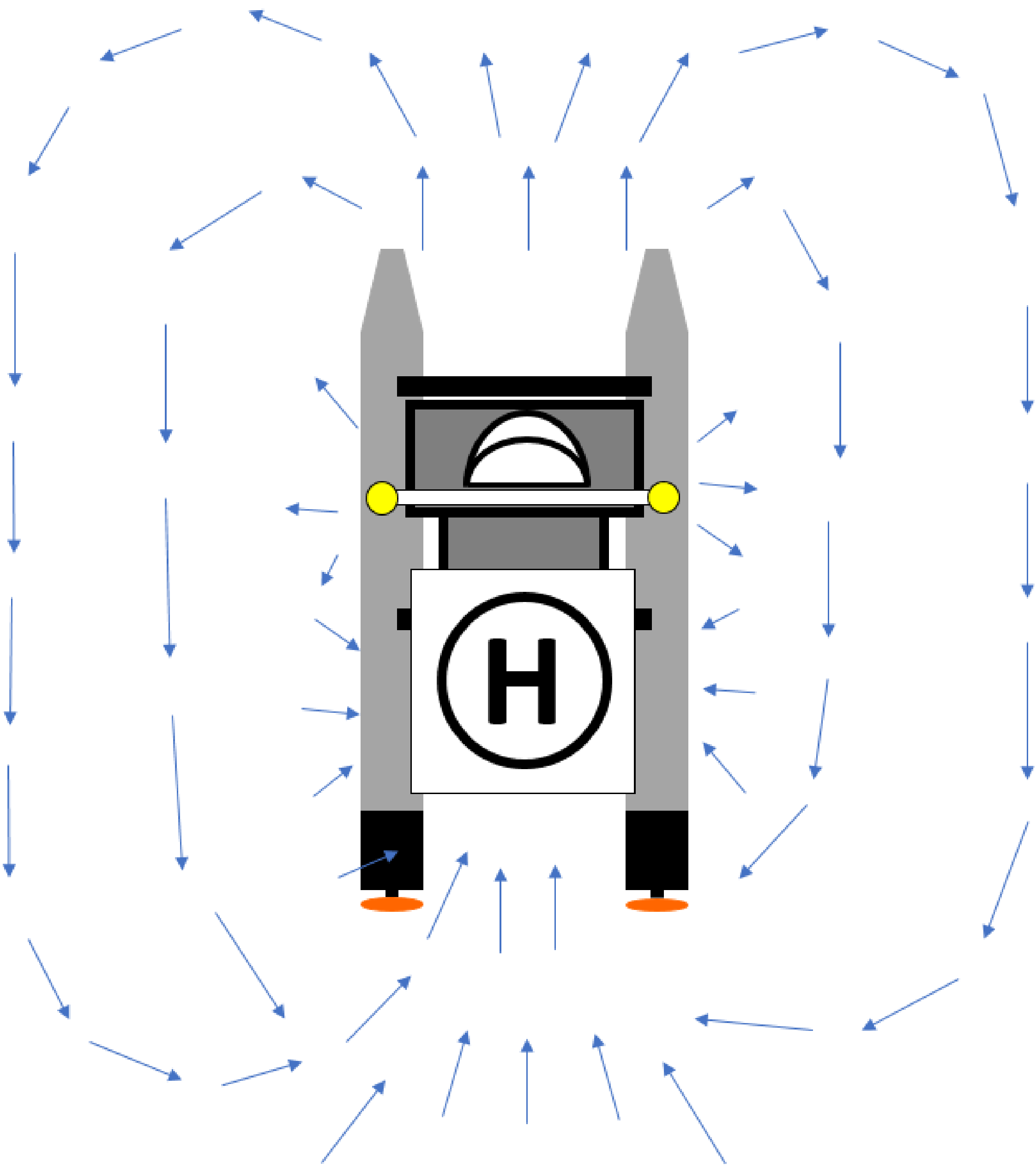


Figure 7 Vector velocity field concept for directing the UAV into position over the ASV. Each vector shows the velocity command the UAV would be given if it were located at the base of the vector.

The advantage of this concept was that once the equation of the three-dimensional vector field was calculated, a position could always be fed into it and a velocity to move would be the returned. While this idea promised to be simple to implement once the equation was established, the resulting ability it provided to the UAV was very inflexible. From any location, the only objective the quadrotor would follow would be to move into the landing position and wait there. The other major problem with this approach was that while the projected path of the UAV at any point could be considered safe, there was little way to control what ground it would cover before arriving back at the landing platform.

An alternative approach was to use spline curves to generate a smooth approach trajectory for the UAV to follow as it approached the ASV. However, this idea had the same limitations as the vector field in that it was very inflexible and would not allow the UAV to do anything practical. In addition, early tests showed that should the UAV lose the plot of the spline curve, it would have difficulty in reacquiring it.

A major change in approach came when initial testing of the quadrotor platform showed that it performed well when following a simple waypoint file rather than having its position or velocity directly controlled by the software. This allowed a basic path to be plotted before the flight and followed automatically as soon as the drone was placed in self-guide mode. The advantages this gave the UAV included the ability to set the precision with which it followed the desired path as well as better control over the altitude of the UAV at each point in the flight. In addition, several options regarding the heading of the UAV in relation to each waypoint became available which was of significance later in the project. In contrast, field testing showed the raw velocity commands were more difficult to use for long range movement of the drone due to a high sensitivity to overcorrecting the position of the drone.

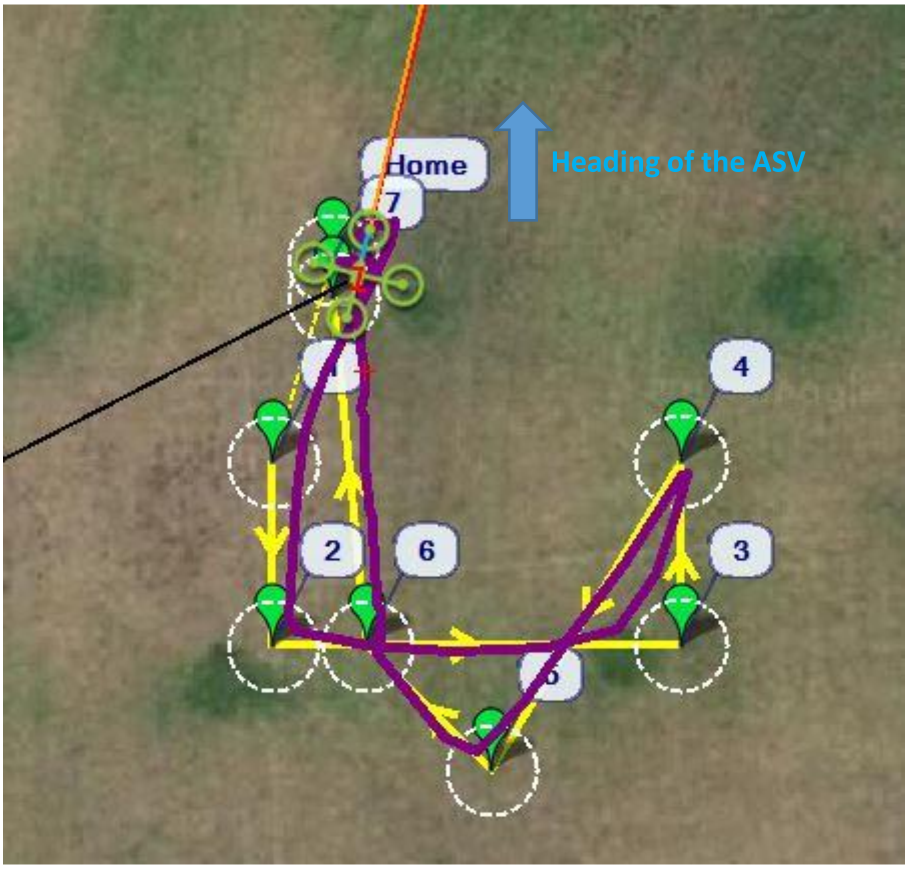


Figure 8 Testing of the capability of the UAV to follow a waypoint file simulating a mission. Green points represent GPS waypoints, yellow direction arrows show the movement of the UAV and white circles represent the tolerance area which the drone needed to achieve for each waypoint. Path of UAV in purple.

The above figure shows a field test of the UAV following a mission waypoint mission file that could be generated seconds before the release of the UAV from the ASV. The tolerance of the distance within which the drone had to achieve each waypoint was set to one metre during this test, but could be tightened to allow for more precise tracking. In this mission, ASV location was set to be the Home location and it as assumed that the ASV remained stationary and with the same heading. In this situation, the UAV was set to a mode where the yaw of the drone did not change during the entirety of the flight. As a result, the drone reached the end of its mission in a loiter above the position of the ASV with the same heading as the expected heading of the ASV. This is the desired status for the close range visual landing algorithm which assumes the heading of the drone and ASV are close.

Later work was completed to ensure that the yaw of the drone could be corrected for changes in the heading of the ASV. This was done by setting the UAV to alter its yaw to point to the next waypoint. This feature was exploited by plotting a set of approach waypoints to the final landing that ensured it would consistently face the same direction as the ASV. Rather than make these waypoints a part of the mission plan which was not dynamic once the drone was launched, these final approach points were constantly being updated to allow the drone the ability to “find” the ASV after it had completed its mission.

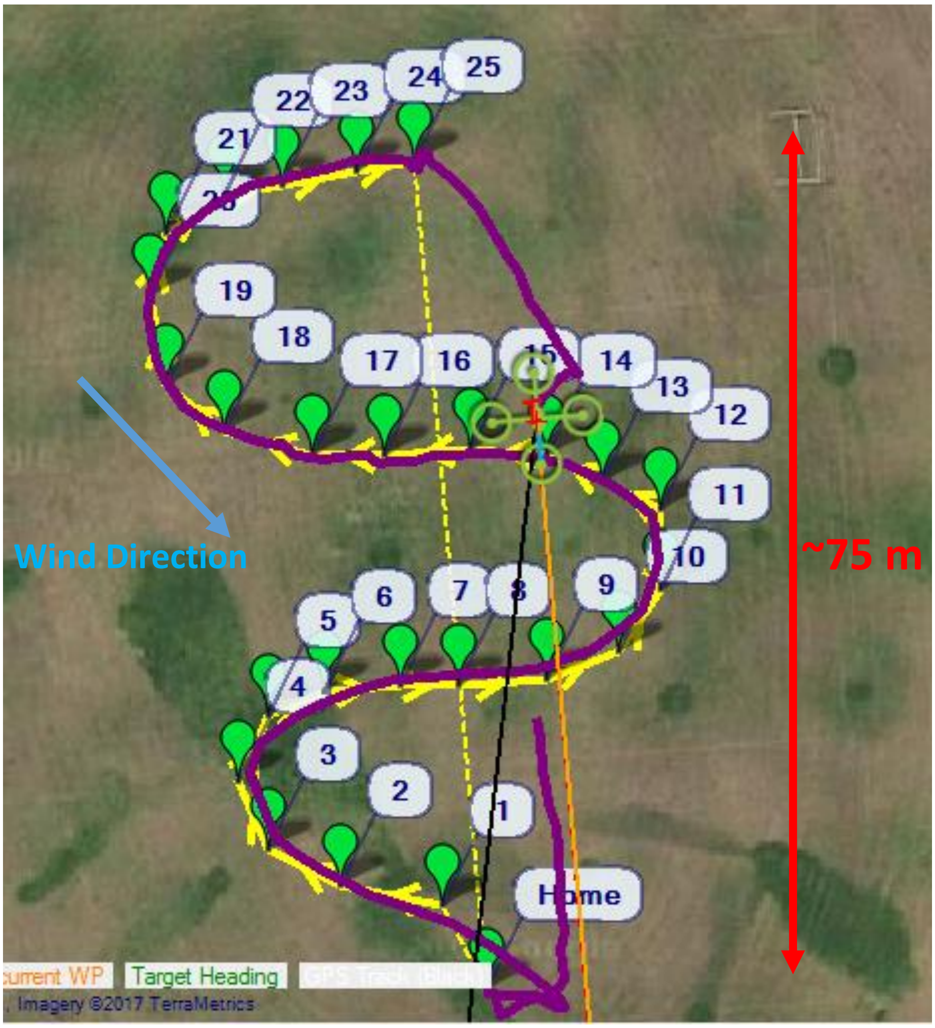


Figure 9 Testing of the capability of the UAV to follow a longer mission with a large set of waypoints. Green points represent GPS waypoints and yellow direction arrows show the movement of the UAV. Path of UAV in purple.

In the test conducted in above figure, the UAV heading was set to follow the direction of the waypoint it was flying towards, resulting in the UAV yaw following a smooth curve closely approximating the yellow arrows from the waypoint file. This demonstrated the capability of future waypoints to control the heading of the drone as well as the ability of the ASV to store a larger number of waypoints and follow them accurately over a significant distance.

Another important result from this test was the way the ASV handled itself in stronger wind conditions while completing a heading varying mission. At waypoints 16 and 17 it can be seen that the path of the UAV has kinks in it where the heading of the drone changed. This was a default behaviour or the flight controller during stronger winds, to pause the progress of the drone and face it into the wind. As can be seen from the flight path, the UAV quickly corrected itself once the wind gust was passed and carried on its course. At these points however, the UAV rapidly lost approximately five metres of altitude while countering the wind gusts. This observation resulted in setting the minimum mission operating ceiling at 10 metres with the exception of periods where the UAV was landing or taking off.

With the results of these tests in mind, a ROS node was created to generate the waypoint file the UAV would follow between the launch and landing. The initial path chosen was a 20 metre by 40 metre course immediately behind the ASV. The choice to fly behind the ASV was thought prudent to ensure the UAV did not come close to the sensor array located on the ASV. It was assumed for simplicity at this stage that the flight area in the close vicinity was an obstacle free space, a not unreasonable assumption in many marine environments.

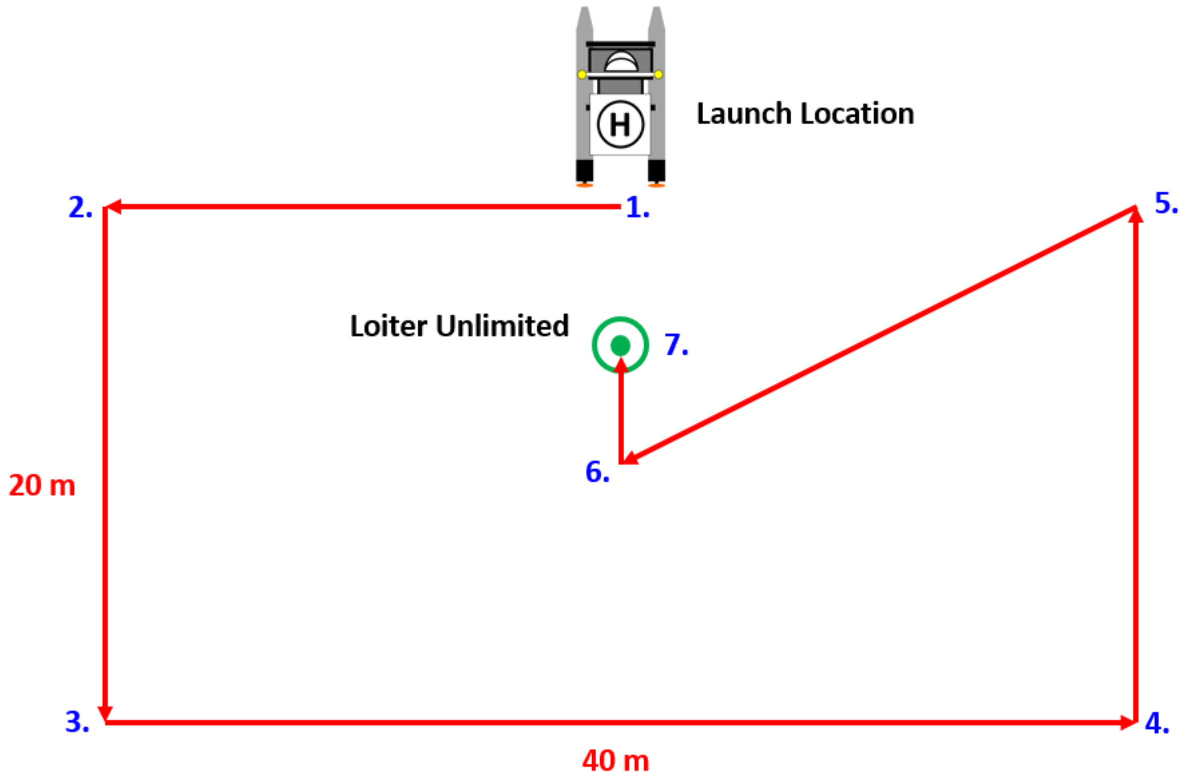


Figure 10 Initial flight path of drone plotted in GPS coordinates behind the ASV from its known location.

The flight path shown in the figure above was constructed using 7 GPS waypoints. The initial 6 waypoints were written as fly-through waypoints, meaning the UAV only needed to approach within a designated distance of the waypoint before approaching the next one. The final waypoint was set to be a loiter unlimited, keeping the UAV in its final pose while the mission planner switched from mission mode to the landing approach. The intention of the waypoints 6 and 7 was to place the drone such that it was approaching the ASV from the rear by the end of the mission with a heading the same as the ASV, assuming it had not moved.

Testing of this planner revealed an issue where the path plotted did not consider the heading of the ASV before the launch. An example of where this could be an issue is shown in the diagram below where the path of the drone would be directly over the sensor array of the ASV after take-off. While the altitude of the waypoints would likely ensure no collision occurred, this is not entirely certain as demonstrated by the altitude drops noted during high winds. In addition, the final approach of the drone to the ASV would be from the wrong heading.

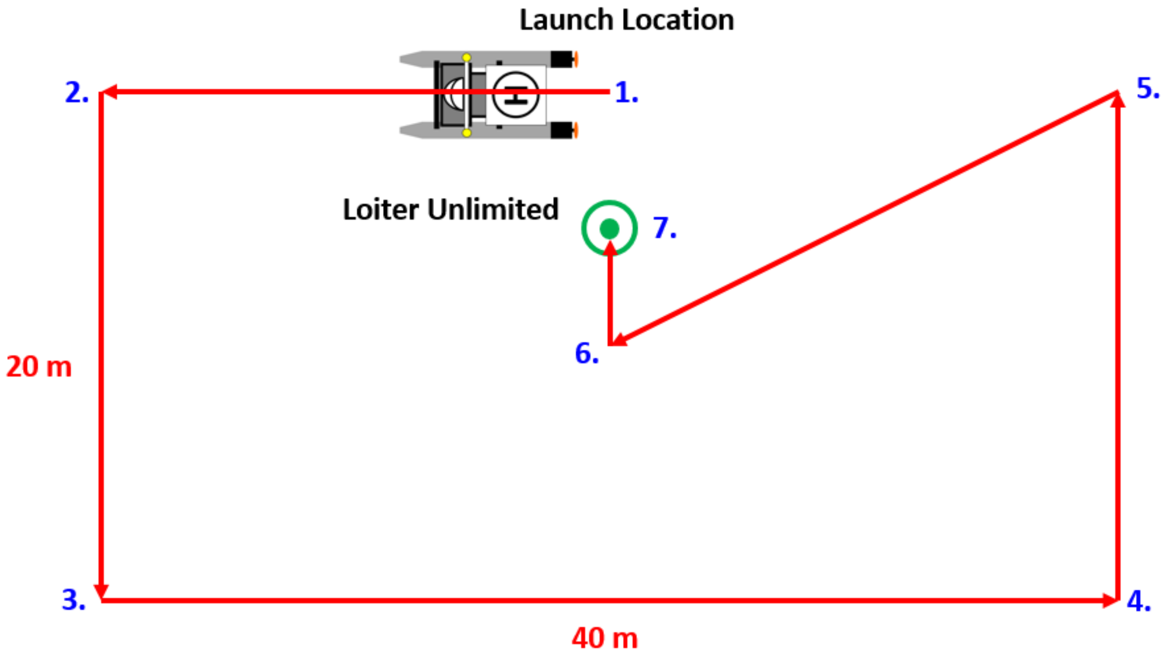


Figure 11 Danger of a collision between UAV and ASV sensor array if the heading of ASV is not taken into account.

To correct this, the path planner was altered to use the heading of the ASV as well as its GPS position to create an angle dependent path as demonstrated in the image below.

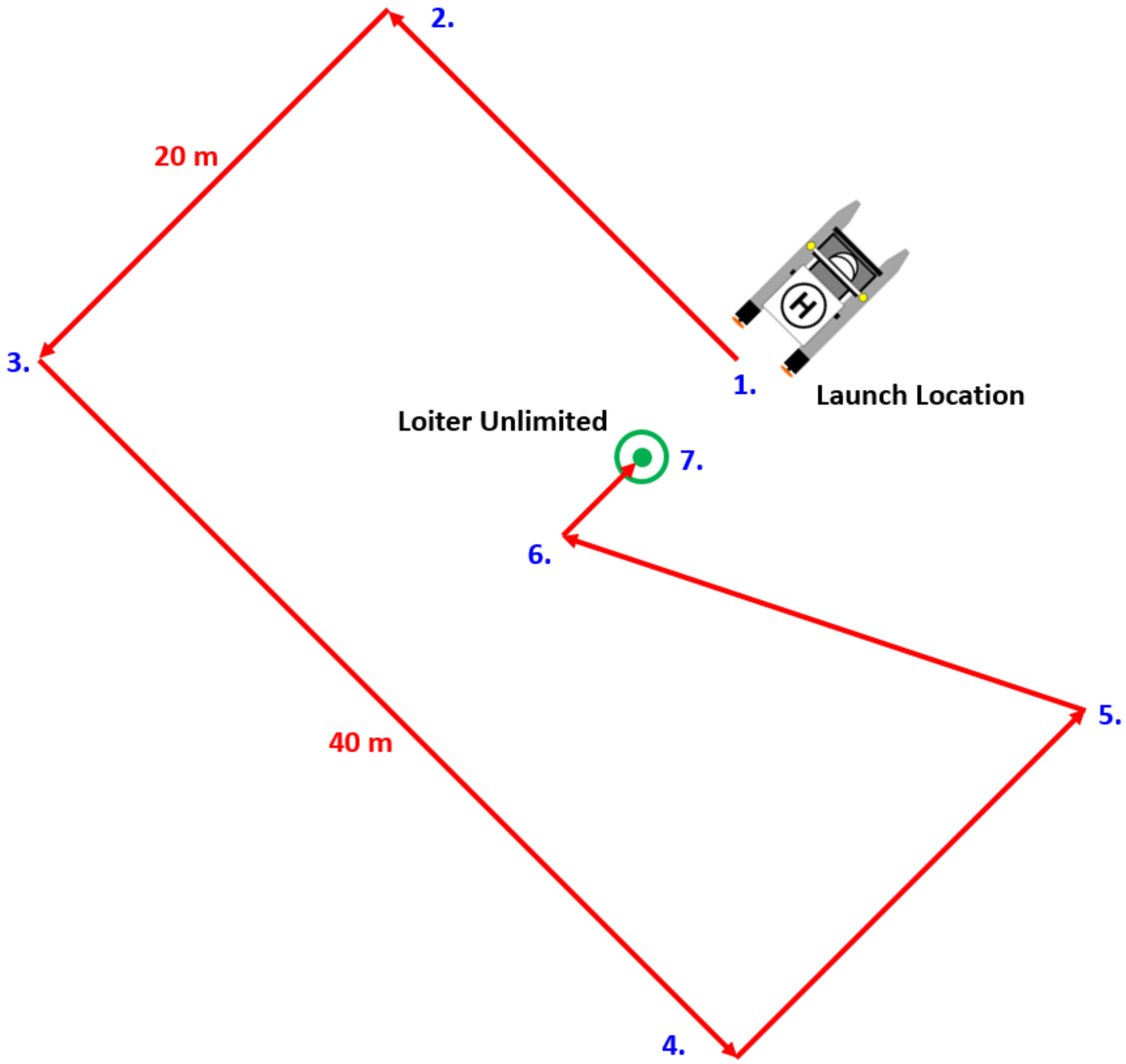


Figure 12 Path plotting using the location and heading of the ASV to create and course that is consistently safe and provides correct approach heading.

This algorithm was capable of generating the waypoint file used in the flight test shown in Figure 8. The drone was clearly able to follow this waypoint file and showed stable behaviour as it approached the landing target in any orientation from the correct direction and with the right heading.

Several features of the path plan were noted which could be improved. First of all, the ASV mounted several obstacle-detecting sensors including lidar and radar which monitored the area in front of the vehicle. While the assumption that the airspace immediately around the ASV would be unobstructed holds true in most cases, it would be better to fly in a region where this could be ascertained prior to the flight. In addition, a likely scenario where the UAV could provide useful data to the ASV would involve the drone flying ahead of the ASV to gather data on the terrain to be traversed. For this reason, the option was added to generate a similar waypoint file in front of the ASV as shown in the diagram below.

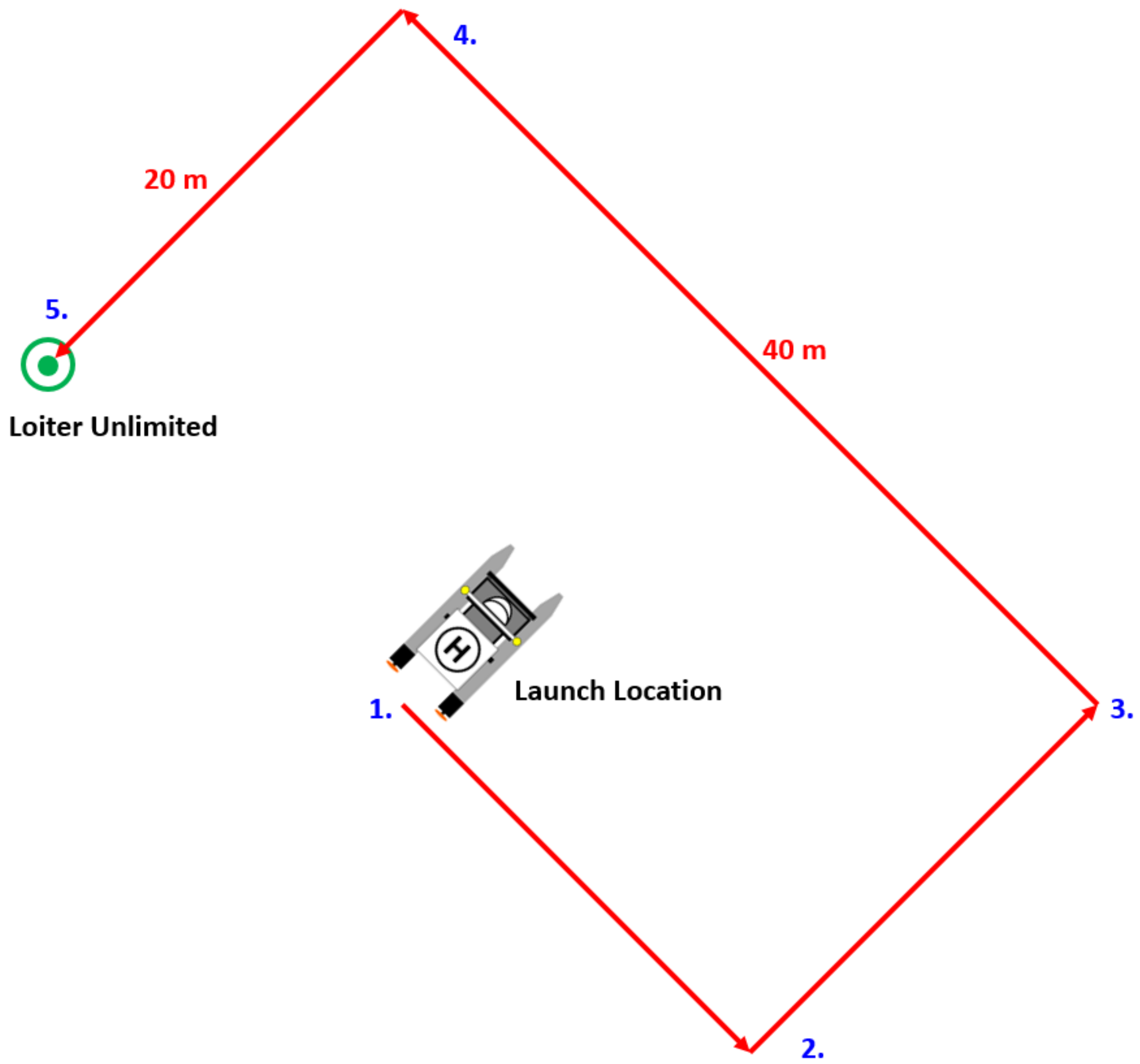


Figure 13 Path plotting for the UAV using the location and heading of the ASV to plot a trajectory ahead of the ASV in range of its sensors.

The final approach waypoints were removed at this stage because it was realized these waypoints would be out of date if the ASV had moved from its launch position. Instead, the last few waypoints were dynamically generated for the UAV as it approached the ASV as will be shown in the next section.

An issue in the development arose at this point with regards to the timing of the waypoint file generation node. The ROS node was designed to write a waypoint file every time it received an update from the ASV regarding its location and heading. As the waypoint file was always over-written to save memory space and ensure the UAV only needed to be loaded with one possible file, the file was in the process of being changed for a significant time slice. This was a danger to the UAV as it was possible to be passed an incomplete waypoint file and could also interfere with immediate drone control software which needed ready access to the file at unspecified times. To overcome this, it was decided to only write to the waypoint file once a second to lessen the risk of conflicts over access to the file. This solution raised another issue however, where the pose of the ASV used when generating the waypoint file was not the most recent but the oldest in the input buffer. Due to the slow processing of input values which came at a frequency faster than 1 Hertz, the node's waypoint files used ASV poses which quickly fell behind reality. A solution was implemented to reduce the buffer size of the input data to a single value to ensure that whenever a waypoint file was written, it used the most recent pose data. All poses received between file writes were discarded.

As the TopCat ASV was unavailable, the path plotting algorithm was tested using data from a UAV flight simulating the movement of an ASV. The plot of the pseudo ASV is shown in the diagram below along with a sample of the waypoint files that were generated along the way, showing where the UAV would fly if launched at that point. The image on the left shows the path the ASV would follow as a waypoint mission displayed in the Mission Planner software. On the right side is shown the plot of the software generated waypoint mission the UAV would follow if launched at a point on the path of the ASV.

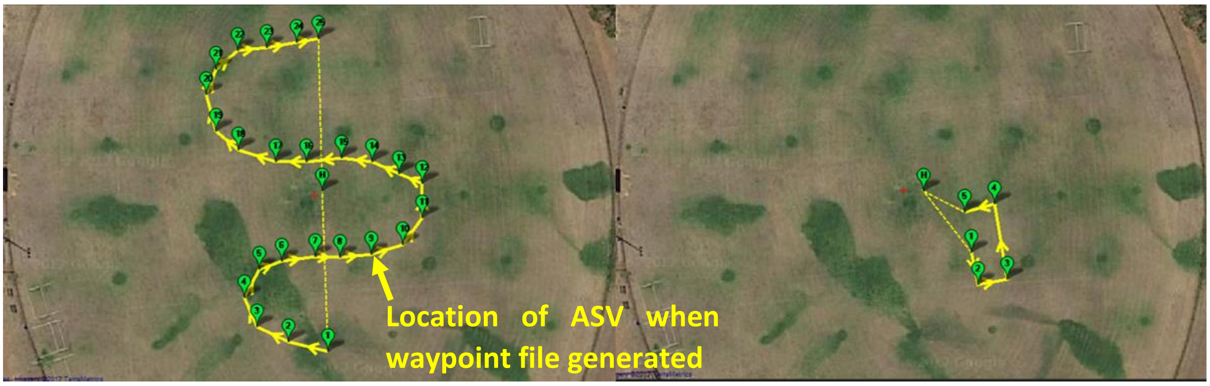


Figure 14 Automatic UAV mission waypoint file generation based upon pose of pseudo-ASV. Left: The simulated path of the ASV. Right: The UAV mission waypoint file generated at the time the ASV was passing through approximately waypoint 9.

Further examples of the automatically generated waypoint files can be found in Appendix A.

3.4.2 Final Approach Trajectory

As mentioned previously, the final approach section of the pre-flight generated waypoint file where the UAV was lined up with the heading and approached the rear of the ASV was removed in the later stages of the project. This allowed flexibility for the ASV to continue its operations and change pose throughout the flight of the UAV because the it did not have to be in the same position it was in when the UAV was launched.

To ensure the UAV approached the ASV from the rear without flying over the ASV, a node was created which generated a spline curve from the location of the UAV to a point above and behind the ASV. A number of different equations for generating smooth splines were tested in MATLAB and an equation which used the Kronecker Tensor was chosen. Using a similar technique as the waypoint file generator for the general mission, a shaping point was calculated to keep the spline from ever coming too close to the ASV's sensor array.

As shown in the figure below, the shape of the curve as a result of this node were suitable for the UAV's approach trajectory.

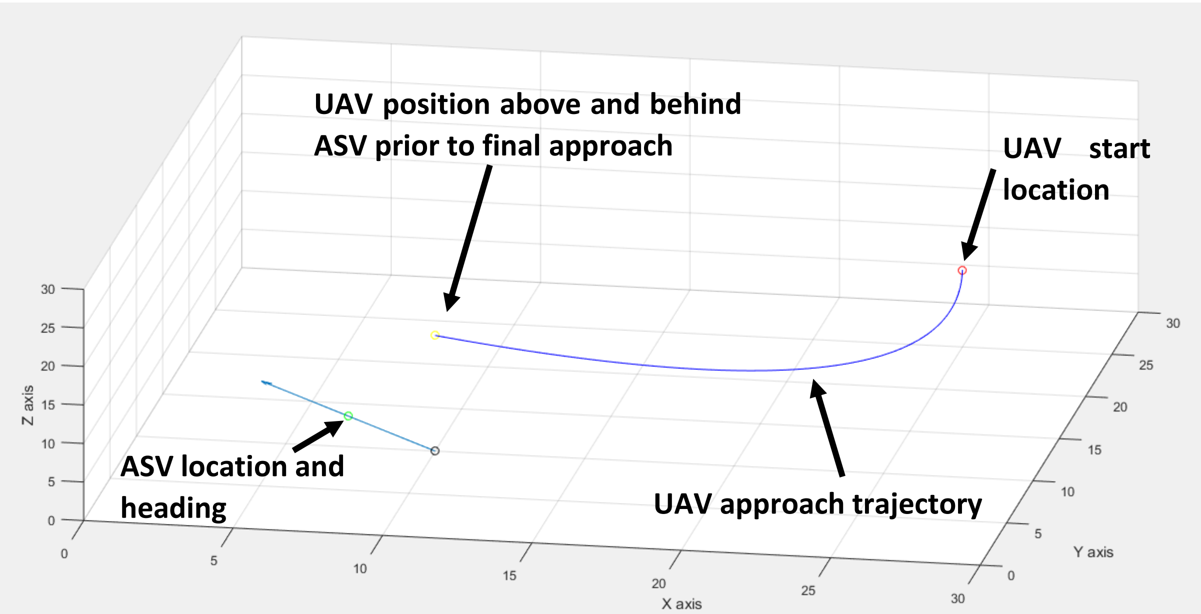
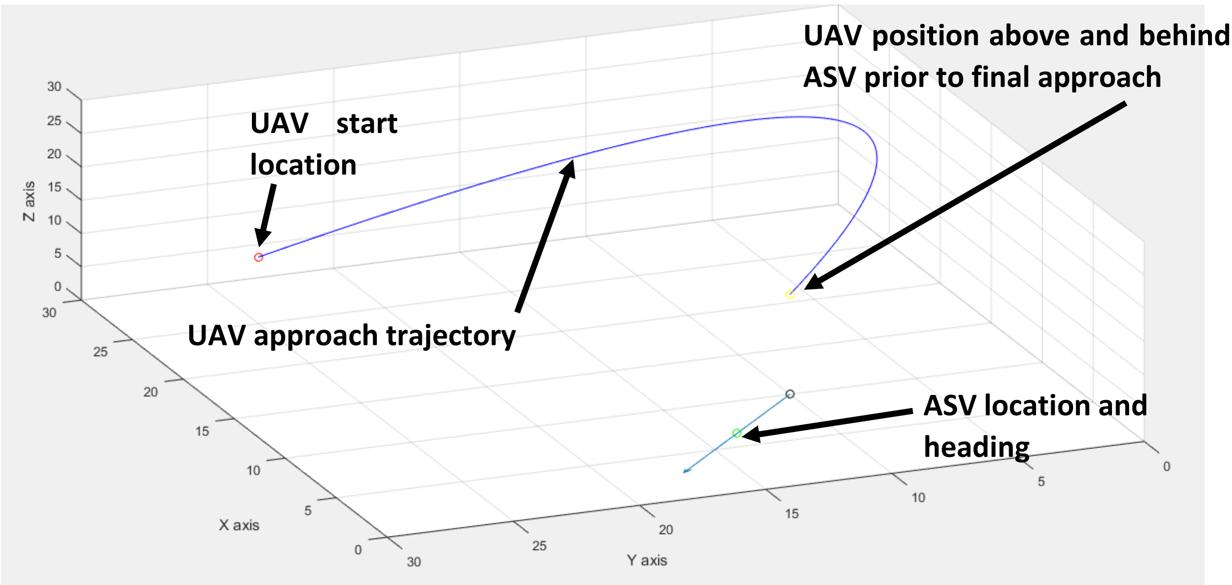


Figure 15 Approach trajectory of the UAV from the rear of the ASV to avoid collision with sensor array.

Not all points on the trajectory were required to be followed perfectly to maintain a safe distance from the ASV. Instead, several waypoints were plotted which the UAV flew through, roughly following the trajectory generated but still arriving at the same end point. The trajectory planning did generate a smooth but narrow trajectory if the UAV was located in near to the front of the ASV, a shown in the image below.



This possible problem that would arise as a result of this trajectory was the UAV could be flown across the path of the moving ASV. Due to time restrictions, a correction for this flaw was not implemented.

The final stage of the mission flight path control was the approach to the landing pad. At the beginning of this phase, a command was sent to the landing control software to switch on the camera and guidance algorithms in preparation for landing. There were two main objectives to be achieved to ensure that the UAV was in the correct pose to begin a safe vision guided landing. First of all, the UAV had to be hovering in a position where the camera could detect the drone. In addition, the heading of the drone needed to be in the same direction as that of the ASV. The reason for this requirement was because the correction software that would guide the drone down to the landing pad gave instructions to the drone using a frame where the UAV was oriented towards the front of the ASV.

To ensure the UAV was moved to a position inside the field of view of the camera, the final mission waypoint that was generated was set directly above the camera at an altitude of 8 metres. Field testing showed that the camera was able to reliably detect and correct the UAV in this position. Upon achieving this waypoint, the UAV entered a precision loiter mode where it remained until either the final waypoint was updated due to the ASV location changing or the vision guidance software locked onto the UAV and began controlling it for approach.

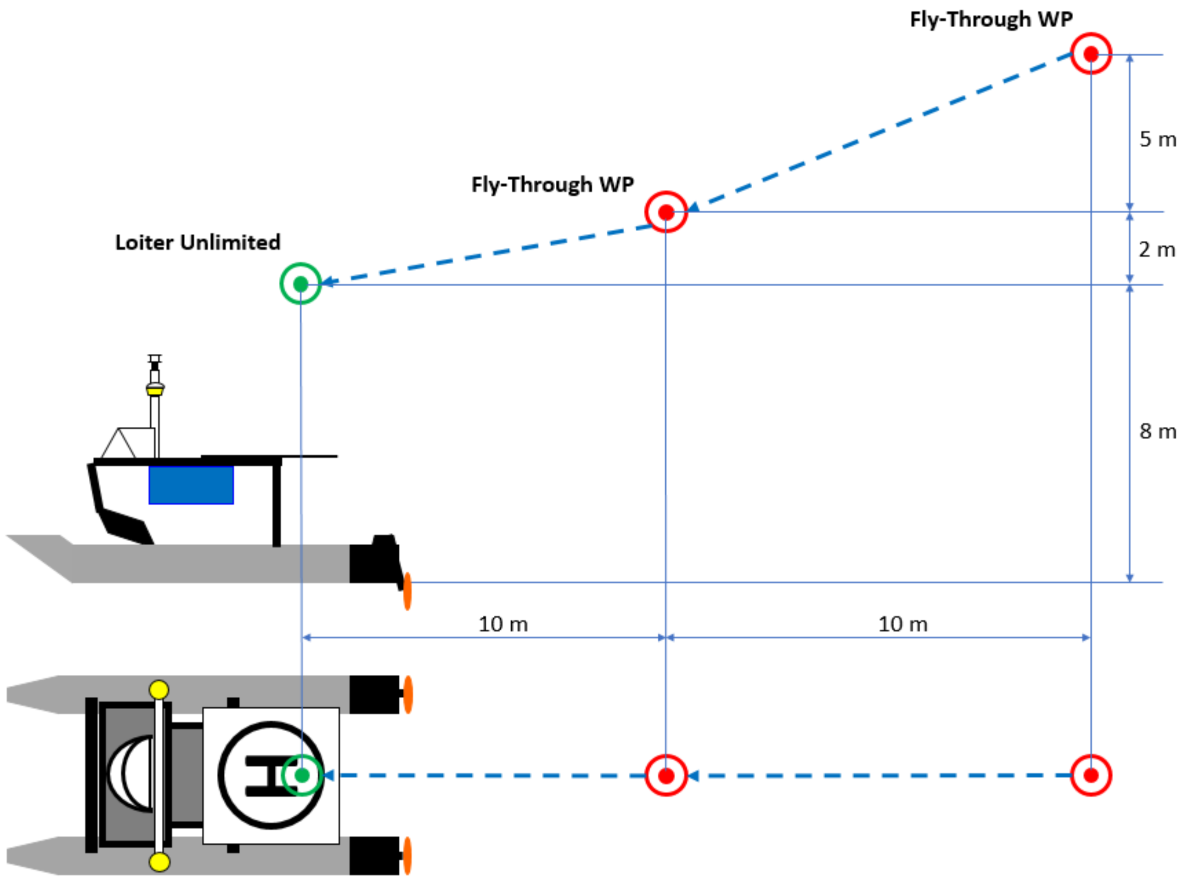


Figure 16 Final approach waypoints for the drone to follow based upon the pose of the ASV. These were dynamically updated to ensure the heading of the UAV was the same as the ASV.

To ensure the UAV was oriented in the same direction as the ASV, several initial approach waypoints were also generated. Using a setting on the drone's PixHawk flight controller, the UAV was commanded to maintain a heading towards the next waypoint of the mission. As the waypoints were oriented in a straight line behind the boat, the orientation of the drone remained always towards the front of the ASV when it reached the final waypoint above the camera. Field testing showed the UAV was capable of maintaining this heading regardless wind direction.

The UAV flight controller allowed a tolerance to be set on the waypoints which gave a distance within which the UAV had to pass from the waypoint for that position to be achieved. Unfortunately, this tolerance was the same across every waypoint the UAV followed and could not be set for individual waypoints. While a high tolerance was acceptable for the mission waypoints, the final waypoint was preferred to be fairly small to ensure the UAV remained in a tight location while the camera locked onto it. If the tolerance was set too tight however, the main mission waypoints took longer to achieve as the drone had to stabilize and position itself with greater precision as each waypoint was passed. A compromise of an accepted radius of 0.5 metres was found to be a suitable choice. At the chosen altitude of 8 metres above the ground level, the drone remained in the field of view of the camera when at the extreme of this tolerance around the final waypoint while the main mission waypoints remained were still able to be achieved rapidly.

3.5 Mission Control

The mission control node of the ROS system was designed to implement a Moore model finite state machine. The Moore model was chosen as it generates the output of the system based purely on the state the machine is rather than the input to the machine as well as the state. This was important to the application where some of the states could have been preceded by a range of different states, but the commands given to the drone were to be the same regardless of its history.

The structure of the ROS node used a while loop to continuously run a switch statement until the node was shut down externally. This was done to allow the node to run even while the UAV was not in operation but could be activated immediately by an external command. The switch statement activated the separate methods that defined the behaviour of each of the states. Each state used globally provided information from the sensors to provide commands to the drone control software and set the next state.

3.5.1 State Machine Design

The state machine had eight main states which defined what the general objectives of the UAV ought to be.

The first state was the Pre-flight Check Mode state where the conditions and requirements to fly were assessed. Most of the calculations for these conditions were already completed in external nodes as described in earlier sections. The state simply combined the results of whether the drone was safe to fly in all these conditions and if so transitioned to the launch mode. If any conditions were unfavourable, the state remained the same, checking the conditions until it was safe to fly. The battery level was required to be above 80%. Critical errors detected from the UAV would result into a hard abort state being chosen as was the case in almost all the states.

Entrance Paths		Exit Paths	
Previous State	**Entrance Conditions**	**Next State**	**Exit Conditions**
Mission Complete	Authorized to fly	Take-off	Ready for take-off
Pre-flight Checks	Not ready for take-off	Pre-flight Checks	Not ready for take-off
		Hard Abort	Critical Failure

Table 3. Pre-flight Checks Mode state transition table

The next state was the Take-off Mode in which the command to complete an automated take-off was sent. The state remained constant until the drone achieved a successful take-off. The flying conditions were again monitored in this state and returned the UAV to pre-flight state if they became unsafe for flying and the UAV had not been launched. However, if the drone was in the process of taking off, the take-off was continued due to inability to be certain the landing platform had not moved. A soft abort would be engaged which would attempt to take the drone to a safe location and immediately land it. In the case of the critical failure a hard abort would be engaged.

Entrance Paths		Exit Paths	
Previous State	**Entrance Conditions**	**Next State**	**Exit Conditions**
Take-off	Ready for take-off	Take-off	Waiting for Take-off to Complete
Pre-flight Checks	Waiting for Take-off to Complete	Pre-flight Checks	Take-off Failed – not airborne
		Mission	Take-off Complete
		Soft Abort	Take-off Failed – airborne
		Hard Abort	Critical Failure

Table 4. Take-off Mode state transition table

The next state was the Mission Mode in which the command to enter autonomous mode and follow preloaded waypoints was sent to the UAV. The state remained constant until the drone successfully finished flying through the pre-allocated waypoint file generated by the mission path planer. The flying conditions were monitored in this state and changed the mission state to soft abort state if they became unsafe for flying or the battery level fell beneath 40%. This would force the UAV to quit the mission and attempt a landing as soon as possible. In the case of a critical failure a hard abort would be engaged. If the UAV successfully accomplished the mission provided in the waypoint file, the state moved into the landing approach.

Entrance Paths		Exit Paths	
Previous State	**Entrance Conditions**	**Next State**	**Exit Conditions**
Take-off	Take-off Complete	Approach	Mission Finished
Mission	Waiting for Mission to Complete	Mission	Waiting for Mission to Complete
		Soft Abort	Mission Failed
		Hard Abort	Critical Failure

Table 5. Mission Mode state transition table

The next state was the Approach Mode in which the command to enter a guided state and follow a dynamically updated set of waypoints leading to the ASV was sent to the UAV. The state remained constant until the drone reached its position above the camera on the landing platform and the visual guidance software locked onto it. The flying conditions were monitored in this state and but a soft abort would not be engaged if it detected it was unsafe to fly as the drone was approaching for landing already. In the case of a critical failure a hard abort would be engaged. If the visual guidance software was able to detect and begin guiding the UAV, the state transitioned into the landing state.

Entrance Paths		Exit Paths	
Previous State	**Entrance Conditions**	**Next State**	**Exit Conditions**
Approach	Camera Not Locked	Approach	Camera Not Locked
Mission	Mission Finished	Landing	Camera Locked
		Hard Abort	Critical Failure

Table 6. Approach Mode state transition table

The next state was the Landing Mode in which the UAV was moved down to the landing pad by the visual guidance system. The state remained constant until the drone successfully landed. The flying conditions were monitored in this state and a soft abort would be engaged if it detected it was unsafe to fly as the drone was not permitted to by close to the ASV in unstable flying conditions. In the case of a critical failure, or the battery reaching a voltage of 10.25 volts, a hard abort would be engaged. If the visual guidance software guided the UAV to a successful landing, the mission state moved to Mission Complete Mode.

Entrance Paths		Exit Paths	
Previous State	**Entrance Conditions**	**Next State**	**Exit Conditions**
Approach	Camera Locked	Mission Complete	Landing Complete
Landing	Landing Incomplete	Landing	Landing Incomplete
		Soft Abort	Landing Failed
		Hard Abort	Critical Failure

Table 7. Landing Mode state transition table

The final main loop state was the Mission Complete Mode in which the UAV waited for the UAV to be activated by an external command. Conditions were not monitored in this state as the UAV would remain on the landing platform for the duration of this mode. When authorization to fly was provided by an external input representing the ASV mission planner requesting a flight, the state would transition to Pre-flight Check Mode.

Entrance Paths		Exit Paths	
Previous State	**Entrance Conditions**	**Next State**	**Exit Conditions**
Mission Complete	Awaiting Authorization	Mission Complete	Awaiting Authorization
Landing	Landing Complete	Pre-flight Checks	Authorized to fly

Table 8. Mission Complete Mode state transition table

In addition to the states used during the normal operation of the UAV, there were also two states used to handle errors or failures in the operation of the system. The first was a Soft Abort Mode state. In this state, regardless of the previous state, the UAV was commanded to stop its current objective and attain an altitude of 10 metres to avoid collision with any obstructions. Once the altitude was achieved, the UAV would be commanded to fly directly to the first waypoint of the landing approach waypoints and from there the state would be changed to attempt a landing. This option was used in situations where any minor system error was encountered, the weather conditions were considered too unsafe for flying or the battery level had become too low for the objective to be completed.

Entrance Paths		Exit Paths	
Previous State	**Entrance Conditions**	**Next State**	**Exit Conditions**
Take-off	Take-off Failed – airborne	Soft Abort	Soft Abort Failure
Mission	Mission Failed	Approach	Soft Abort Success
Landing Mode	Landing Failed		
Soft Abort	Soft Abort Failure		

Table 9. Soft Abort Mode state transition table

The second error handling state was the Hard Abort Mode state. This state was intended to be used in situations where a critical system malfunctioned or the UAV was deemed unfit for landing on the ASV until it had been inspected or repaired. Recovery of the UAV to the ASV was not possible once this state was entered. While in this state, the UAV would attempt to ascend to an altitude of 10 metres before engaging the flight controller's inbuilt return to home function. This would cause the UAV to automatically fly to the currently chosen home position and perform a normal unguided landing. The mission planner would publish to the system the location of the UAV if it was available and the last loaded home position as that would be the UAV's most likely location if radio contact was lost.

Entrance Paths		Exit Paths	
Previous State	**Entrance Conditions**	**Next State**	**Exit Conditions**
Pre-flight Checks	Critical Failure	Hard Abort	Publish Location and Loaded Home
Take-off	Critical Failure		
Mission	Critical Failure		
Approach	Critical Failure		
Landing	Critical Failure		
Soft Abort	Critical Failure		
Hard Abort	Publish Location and Loaded Home		

Table 10. Hard Abort Mode state transition table

The complete state transition diagram is illustrated in the figure below.

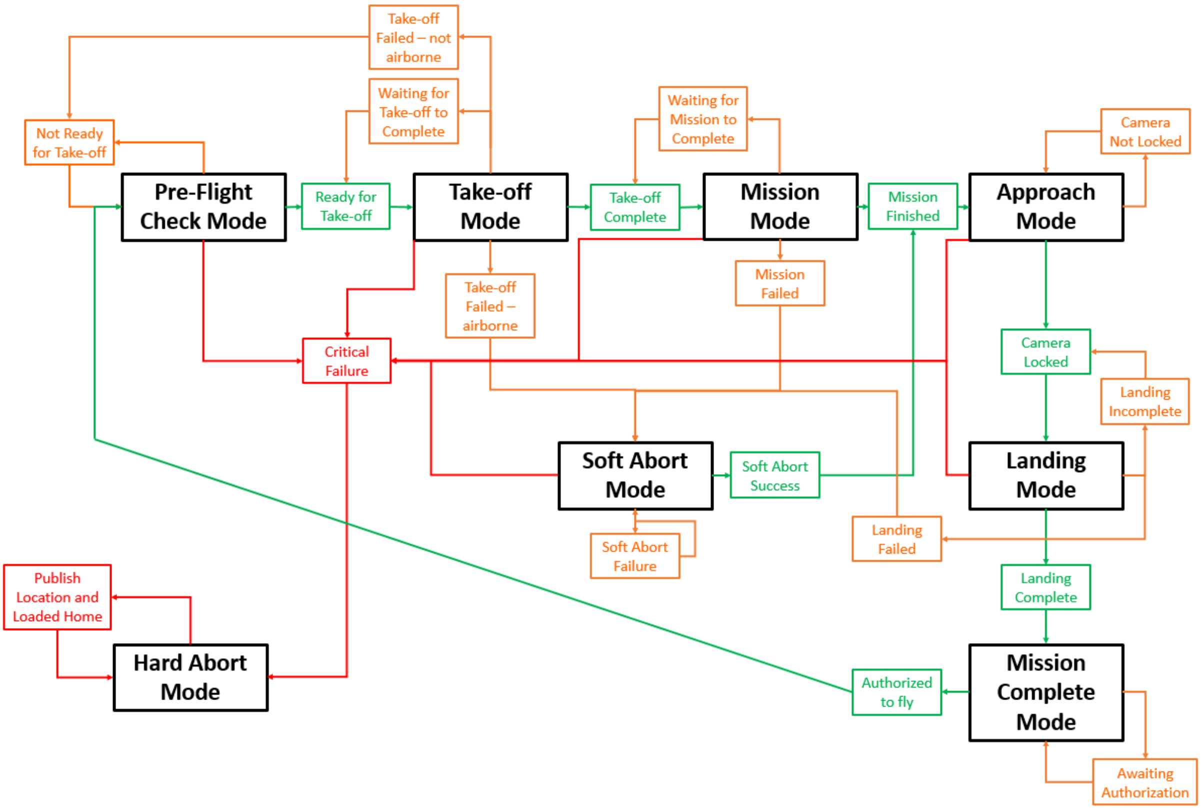


Figure 17 Main mission planner state transition diagram. Black nodes indicate possible states, green nodes show desirable transitions, orange nodes show waiting or error handling transitions and red nodes indicate critical error transitions

3.5.2 State Machine Testing

The mission finite state machine was tested using a separate ROS node which generated input data that could be expected across the sensor topics. This included simulation of the wind speed and direction from the ASV, the ASV pitch, UAV battery status and GPS location. The node would randomly generate data that could be expected to be found in these topics. A set of expected safe transition behaviours was created to test the operation of the mission control algorithm. The node was run in conjunction with operation of the mission control nodes and the number of expected and unexpected transitions were recorded in a sample size of 100 tests. A plot of the results from the test of the final algorithm is shown in the image below.

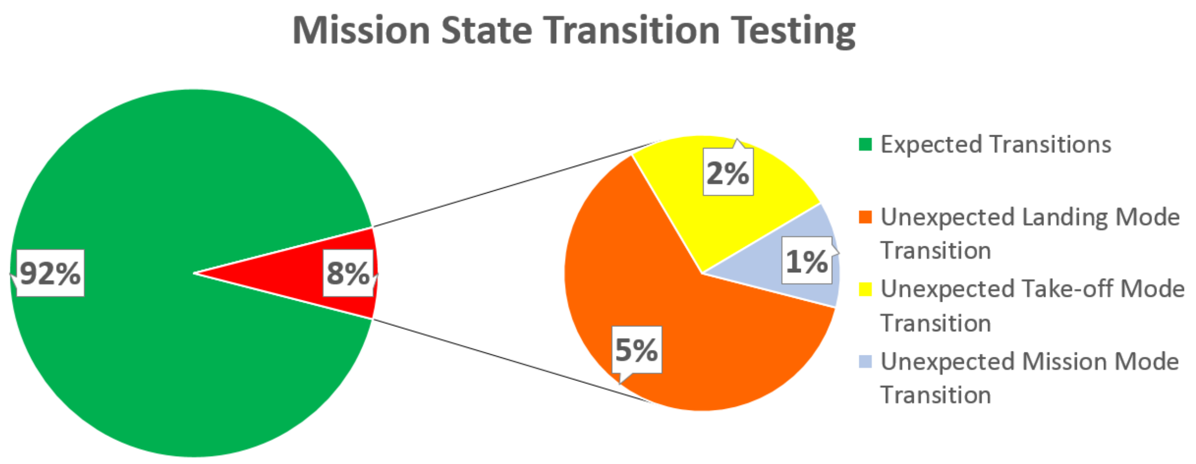


Figure 18 State transition testing results from 100 sample transition scenarios using random inputs

This result demonstrates not only the effectiveness of the state machine to correctly set the UAV mission state based upon the data supplied to the mission planner but also shows the data analysis nodes were working correctly as well.

However, it was found to be difficult to test the mission planner's ability to run a whole mission. This was due to the fact that generating random data and inputting it into the sensors often would provide situations to the mission planner which were not expected to occur in real life. While this could be seen as a weakness on the part of the mission planner, the results from such a test tended to be misleading by presenting an overwhelmingly high number of errors in the mission planner's handling of situations and almost every mission ending in the hard abort mode due to a critical error or condition arising at some point in the mission. Instead, a better method was required to analyse the state machine and determine issues such as the presence of unwanted loops and or cases where the state would reach an unintended dead-end.

A petri net model of the state machine was generated to allow the quantitative assessment of the mission planner. Petri nets are mathematical representations of the flow of the status of a system and can be used to model a system's behaviour. In this case, the critical qualities that were being assessed included whether it this state machine was bounded and did not contain any deadlocks.

The Petri net model was generated in the PIPE (v4.3.0) software and analysed using the built-in tools. The petri network analysed is shown in the diagram below.

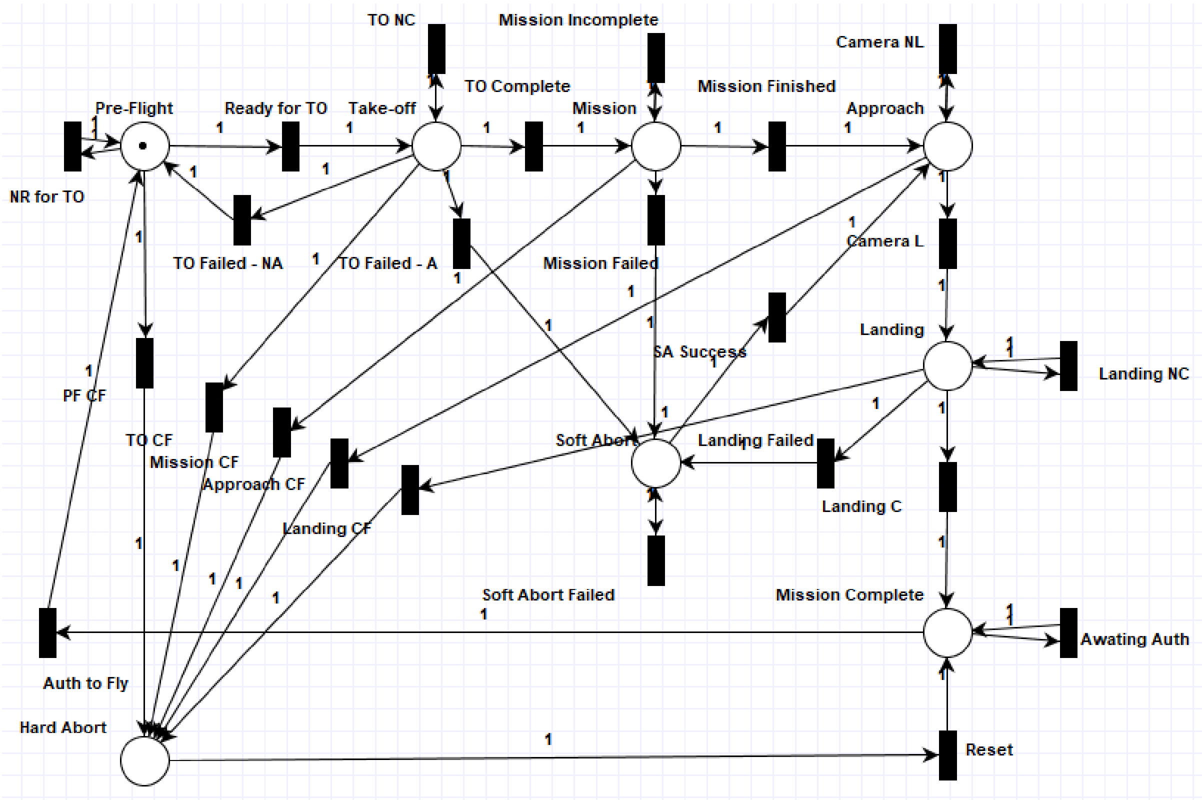


Figure 19 Petri net model of the mission control state machine (Software: PIPE v4.2.3)

The petri net consisted of the same states as given in the state machine description mentioned earlier, and these are represented by the white circles in the above diagram. The black rectangles represent transition conditions. The only difference between the petri net and the mission control algorithm was the need to split the critical failure transitions that result in a hard abort condition into separate transitions as is required for a true state machine.

The results from these tests showed that the initial versions of the state machine did not have stable and safe behaviour. One of the key factors that caused this was the presence of dead-ends where the petri net state was liable to become trapped. In a situation such as this, the UAV could not recover no matter what inputs are provide to it.

There were several clear results from the testing of the petri net. The first was a classification given based upon the general behaviour and characteristics of the net.

Petri net classification results

State Machine	true
Marked Graph	false
Free Choice Net	true
Extended Free Choice Net	true
Simple Net	true
Extended Simple Net	true

Figure 20 Analysis of the Petri net shows it to display state machine behaviour.

As shown in Figure 20, the final Petri net modelled a state machine. This indicated the transition times and behaviour of the network would be exactly as expected from the conventional state machine diagram.

Another important feature that required testing was the terminal state of the machine. This means what the expected behaviour of the machine is if a random set of inputs are applied to the machine. In an unbalanced machine with deadlocks or loops, certain states are more commonly frequented than others as the mode of the machine is more commonly caught in those states. In a balanced net however, all nodes ought to be equally frequented.

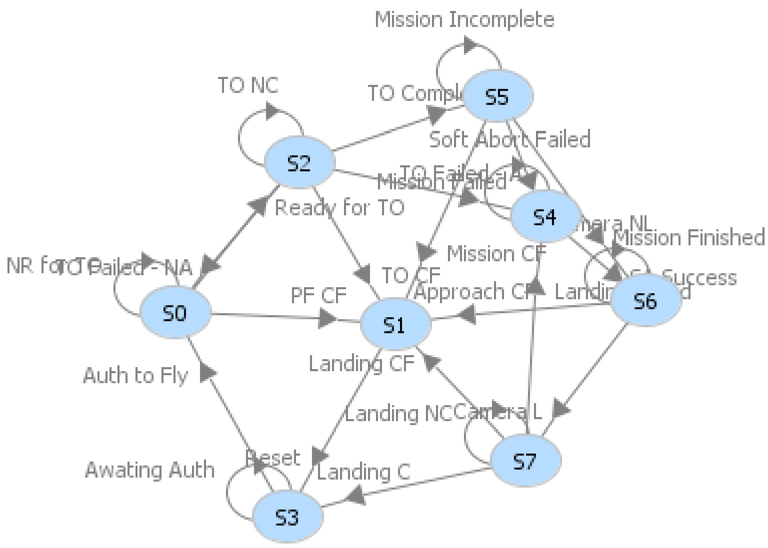


Figure 21 Analysis of the Petri net shows it to display stable behaviour, with all states equally frequented and thus the same colour.

Figure 21 shows that the balanced quality has been achieved in the final iteration of the state machine. This analysis is backed up by a second test which examines whether the net is bounded or not.

P-Invariants

Hard Abort	Pre-Flight	Approach	Mission	Take-off	Landing	Soft Abort	Mission Complete
1	1	1	1	1	1	1	1

The net is covered by positive P-Invariants, therefore it is bounded.

P-Invariant equations

M(Hard Abort) + M(Pre-Flight) + M(Approach) + M(Mission) + M(Take-off) + M(Landing) + M(Soft Abort) + M(Mission Complete) = 1

Figure 22 Analysis of the Petri net shows it to display bounded behaviour.

The bounded status as shown in Figure 22 means the state of the machine cannot reach an unknown value or one outside the set states in the machine.

The final analysis of the Petri net examined three properties. These were whether it was bounded, safe and deadlocked.

Petri net state space analysis results

Bounded	true
Safe	true
Deadlock	false

Figure 23 Analysis of the Petri net shows it to display bounded, safe and non-deadlocked properties.

As can be seen from the results in Figure 23, the final Petri net design is bounded as mentioned before but also safe from unrecoverable loops and did not contain any deadlock conditions.

4. Discussion and Future Work

4.1 Evaluation of Conditions

4.1.1 Wind Speed and Direction

Due to not actually using the ASV's anemometer at any point during the project, it was not ascertained how well the software for measuring wind speed and direction would work. However, tests using simulated wind worked in most situations.

It was observed that the current software reacts very quickly to gusts and changes in the wind speed. While a positive aspect in some regards, this sensitivity also meant the drone did not often achieve a situation where it could complete a full take-off in simulation with random wind data.

Two suggestions would be investigated for future work to correct this. The first would be to adjust the wind simulation software so that rapid and significant changes in wind speed are less uncommon. This could be done by putting a limit on how much the wind speed can change from one reading to the next. Th other option is to make further adjustment to the averaging method so that wind gusts need to last longer before the UAV reacts to them.

Further testing is required to determine the exact wind speeds the UAV can tolerate. The figures provided in this report have only been estimates of the wind speed based upon Bureau of Meteorology reports.

Another area for future work is making use of the air speed sensor on the UAV. By merging this data with the heading of the UAV, a second estimate of the wind direction and speed could be made and compared with that received from the UAV anemometer. Alternatively, during flight this data could be exclusively used by the UAV to determine the wind speed used to calculate the home position. This choice would make sense as the wind velocity at the drone is more significant to this calculation.

4.1.2 ASV Pitch

Similar to wind speed, the pitch measurement was not practically tested and only simulated through software. In later stages of testing, the UAV with redesigned landing gear showed good landing qualities and low likelihood to slide on the landing net despite angles of around 5-10 degrees. However, this was not rigorously evaluated.

It is suggested for future work to test the maximum angle the UAV can tolerate when landing with consideration for the safety of its propellers as too steep an angle can easily cause clipping to occur between landing net and propeller.

4.1.3 UAV Battery Status

The battery status monitoring software simply read the information passed to the ASV across the MAVLINK regarding the battery. This information only provided the percentage level of the battery's capacity and the voltage levels overall and for each cell.

One major issue that arose was that the percentage battery capacity value was found to read as the capacity the battery had available, using the point at which the flight controller was powered as the reference. For example, if the battery had 50% charge when the flight controller was powered up, the battery percentage level would tend to start at 100% and decrease twice as fast as it normally would. Since the battery percentage level was relative, it was not a trustworthy source of charge information. The voltage levels were not linear indications of the battery energy levels, making them more difficult to use.

One solution that ought to be considered for future work would be to only connect full batteries to the flight controller at the start of a mission and not disconnect them for the entirety of the mission. This would ensure the battery percentage is always read relative to a fully charged battery.

An alternative would be to cycle the batteries over a number of typical flight and record the voltage discharge profile of the batteries during this time. By comparing flight time with the voltage levels, it could be possible to ascertain the energy levels in the batteries.

A third option is to investigate installing a coulomb counter on the UAV to determine the energy dissipated during the flight and transmit a more accurate battery percentage to the ASV.

4.1.4 GPS Location and No-Fly Zone

The GPS system was tested by placing the drone in a stationary position and observing by how much the location estimate drifted over time. Observations suggested the drone GPS location could drift by up to a meter a minute but this depended on how obscured the satellite sky was. Considering that the GPS was not to be used for any of the precise final landing manoeuvres, this was deemed acceptable for general use. The UAV was capable of remaining in the upward-looking camera's field of view in calm conditions using GPS position hold.

Real Time Kinematic (RTK) corrections were intended to be implemented using the ASV's GPS position but this was not carried out. This would be a suggested future work as it would improve the ability for the UAV to remain in a steady position and thus improve the visual guidance system's chance to locate the drone.

The world frame in which the drone operated was a Universal Transverse Mercator (UTM) projection taken from the WGS84 coordinates received by the UAV's GPS. This conversion was calculated with accuracies of less than 200 cm using the proj4 library. However, it should be noted that this system is not ideal for all locations in the world. When operating in the extreme north and south of the globe, the flat rectangular projections, which approximate the globe well closer to the equator, become distorted. If operations were considered around the globe, a more flexible world frame would be recommended.

Another improvement that ought to be attempted is to find a method to determine the UTM zone from the GPS data rather than have it hardcoded as is the current status.

The GPS distance calculation to ensure compliance with local airspace legislation was tested with simulation but not in practice. Due to the extra safeguards built into this calculation, the UAV demonstrated its capability to recognize and avoid operating in the no-fly zones of the Adelaide metropolitan region. However, its behaviour was inefficient and in some areas that were safe for flight it did not operate. This was due to the assumption of circular no-fly zones.

In the future, it is recommended that instead of a calculation, a more flexible geo-fence be used to keep the UAV from entering restricted airspace. This is a set of GPS coordinates which set the boundaries of exclusion zones. Depending upon the size of these files, a decision must be made as to whether to store the geofence information on the UAV or ASV. While having the information on the UAV takes up memory space, it would also be available if the UAV loses radio contact with the ASV.

While the GPS worked well in all the field trials and was able to provide the main form of world frame localization for the UAV, the drone did lack in alternative forms of localization. Based upon the research done in the area and the fact there is a camera already mounted on the UAV, it is recommended that visual tracking be incorporated into the localization algorithm to improve its robustness.

4.2 Home Location Choice

The Home location update worked as intended during simulation testing, providing a less costly route in the case of a hard abort. It was observed that the wind direction which was used to calculate the cost was measured from the ASV anemometer. This was not the best option, as there was a possibility the wind experienced by the UAV and ASV would be different in magnitude and direction. In the case that the UAV is being used to scout ahead of the ASV and might be over different terrain, such as the opposite side of a sandhill, this probability becomes very likely. Future work to make use of wind speed data from the UAV would be recommended.

4.3 Mission Path Planning

4.3.1 Main Mission Path

Due to the focus of the project being on facilitating the drone's ability to take-off and land on the ASV, not much effort was placed into developing the trajectory that the UAV followed between launch and retrieval. As a result, the path planner remained as nothing more than plotting coordinates relative to the frame of the UAV at launch which the UAV then followed as waypoints.

Testing of the UAV's capability to follow the waypoints revealed that it was able to complete its missions at a speed of at least 5 metres per second while travelling through the waypoints with a tolerance of 0.5 metres. The UAV was also tested with higher numbers of waypoint and longer routes. The longest of these contained 25 waypoints and was several hundred metres in length which the UAV was able to complete without issue.

One problem was encountered while flying the UAV on waypoint missions heading into winds greater than approximately 5 metres per second. In this event, the UAV would pause its progress, face into the wind and lose between 5 and 10 metres of altitude. When the gust of wind died down, the UAV would resume its normal altitude and continue the mission. The cause of this behaviour has still to be established. As a result, the UAV is set to follow its waypoints at an altitude of at least 15 metres and this is recommended to be followed until the cause for the anomaly is discovered in future work.

The main mission path is currently written periodically to a waypoint file in the ASV's memory. Although it was not observed during testing, there is the possibility that the ASV writes the waypoint file to the UAV at the same time the file is being updated, creating read/write errors. For this reason, a minor part of future work could focus on developing a software interlock so that the high level and low-level control software do not seek to access the waypoint file simultaneously.

The current mission path plan only covers a rectangular section about twenty metres in front and to either side of the ASV. Future work is recommended to focus initially on creating a number of plans for other mission types which the ASV can choose from before giving the command to the UAV to launch. Later stages can focus on being provided with and area, point or landmark to investigate and then dynamically plot a path before launch.

Once the UAV is capable of carrying out a series of different missions and has the ability to change the main mission path of the UAV mid-flight, future work ought to consider improving the integration of mission objectives between ASV and UAV. An important milestone would be to achieve rendezvousing between the two autonomous vehicles. This would enable them to share their respective paths prior to launch and agree on a meeting location which is cost effective for both vehicles. An even more advanced objective would see the UAV capable of adapting its route from this pre-planned meeting point if the ASV had to change course.

The ultimate goal of the future work is to even further merge the operations of UAV and ASV. In the current version, the UAV assumes a free space around itself when planning a path. While this is often the case in the marine environment, there are certain to be exceptions. It is recommended the data from the ASV lidar and radar is made available to the path planner in the form of an occupancy grid, allowing the UAV's trajectory to avoid potential obstacles.

4.3.2 Final Approach Trajectory

Similar to the main mission approach trajectory, the approach trajectory is largely pre-planned. However, this was intended for the purpose of ensuring the direction the UAV approaches from is behind the ASV sensor array and so will not require as much development.

When testing this section of the software, it became apparent that the UAV was operating in a specific "no yaw" mode when approaching the ASV from the rear. This meant the UAV did not turn on its z axis at any point during the flight. This setting works well if the ASV is in the same heading as when the UAV launched because the UAV will land facing in the same direction as the ASV. This is desirable because the current landing algorithm assumes the heading of the UAV is the same as the ASV. If this is not the case, the UAV cannot be recovered.

With this in mind, it was realized it would be very likely that the ASV has moved and changed heading during the flight of the UAV. It was found that the "no yaw" setting could be altered to yaw the UAV so that its heading would be in the direction of the next waypoint. With this is mind, it is essential the UAV approaches in a straight line from the rear of the ASV through a number of waypoints so that its heading will be correct. This choice worked well in the tests. It is suspected though that if the UAV overshoots the final waypoint, it will turn its heading to face that final waypoint from the opposite direction.

To completely remedy this problem, it is recommended for future work that a new visual guidance control transform be written which can guide the UAV irrespective of its heading once it arrives over the landing pad. Alternatively, a command could be written to align the IMU magnetometers so that the UAV and ASV can achieve the same heading. Tests however showed the magnetometer on the UAV is inaccurate.

4.4 Mission Control

Testing on the mission control finite state machine was done first of all through the Petri net. The results were initially unfavourable, showing that the design had a number of unbreakable loops in the landing situation if the drone attempted to land but was unable to. The first state machine did not take into account the number of attempts which had already been made to land. With the inclusion of conditions by which the UAV could escape tasks it was unable to complete, the Petri net became defined as safe.

Once the Petri net analysis showed the state machine was behaving well, the control system was tested with several hundred random data samples. The state transitions were recorded and compared with the expected values. The outcome of this evaluation is shown in Figure 18. The majority of the unexpected transitions occurred in the landing stage.

It is suspected there are two possible causes for this. One is that there is a mismatch in the parameters of what is safe for landing and what isn't safe for landing between the state machine and the desired parameters. Inspection of the code did not reveal any obvious mistakes of this nature although a deeper look could be undertaken. The second possibility is that the simulation data was providing some out of range inputs during the landing phase simulation. This theory is strengthened by the fact that the control system worked completely as expected when recorded data from the ASV was used instead of randomly generate data.

The first major area for future work with the high-level control software is ensuring its compatibility with the low-level control software. While designed to work together, time constraints prevented them from being tested simultaneously in the field. Once the software is integrated correctly with the hardware, it is suggested the finite state machine is further developed to allow it to run dynamic missions which can be altered mid-flight.

A final improvement would be a redesign of the finite state machine so that it is generated through the Robotic Operating System (ROS) Petri net package. In its current state, the Petri net is developed separately from the state machine. This means there is the possibility for differences between the Petri net and state machine behaviour. If the Petri net is generated from the state machine in ROS and then assessed, it is more likely to accurately display the behaviour of the state machine.

5. Conclusion

This research demonstrated the extra capability a UAV could provide to an ASV. With the objective of creating high level control software for a UAV to be launched and retried from an ASV, a number of methods for determining the pose of the UAV were explored and their characteristics assessed. In addition, the control methods used by some of the most successful projects in this field were considered. Software was written to extract data from the ASV and UAV sensors regarding the environment. The objectives of the project were used to develop software that could plan a simple route the UAV would fly, the locations it would use to abort in an emergency and the approach path it would use when landing. The knowledge of the environment and the desired mission plans were controlled by an over-arching control algorithm. This control algorithm was implemented as a state machine and its characteristics evaluated using a Petri net. Simulation and individual component testing results showed significant progress has been made towards the autonomous launch, flight and retrieval of a UAV from the TopCat ASV.

Bibliography

Abbeel, P, Coates, A & Ng, AY 2010, 'Autonomous Helicopter Aerobatics through Apprenticeship Learning', *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1608-39.

Achtelik, M, Achtelik, M, Weiss, S & Siegwart, R 2011, 'Onboard IMU and monocular vision based control for MAVs in unknown in- and outdoor environments', *2011 IEEE International Conference on Robotics and Automation*, pp. 3056-63.

Ajmera, J, Siddharthan, PR, Ramaravind, KM, Vasan, G, Balaji, N & Sankaranarayanan, V 2015, 'Autonomous visual tracking and landing of a quadrotor on a moving platform', *2015 Third International Conference on Image Information Processing (ICIIP)*, pp. 342-7.

Anitha, G & Kumar, RNG 2012, 'Vision Based Autonomous Landing of an Unmanned Aerial Vehicle', *Procedia Engineering*, vol. 38, no. Supplement C, pp. 2250-6.

Anthony, D & Detweiler, C 2017, 'UAV Localization in Row Crops', *Journal of Field Robotics*, vol. 34, no. 7, pp. 1275-96.

Araar, O, Aouf, N & Vitanov, I 2017, 'Vision Based Autonomous Landing of Multirotor UAV on Moving Platform', *Journal of Intelligent & Robotic Systems*, vol. 85, no. 2, pp. 369-84.

B B V L, D & Singh, P 2016, *A survey on design and development of an unmanned aerial vehicle (quadcopter)*, Emerald Group Publishing Limited, Bingley, 20496427.

Cesetti, A, Frontoni, E, Mancini, A, Zingaretti, P & Longhi, S 2009, 'A Vision-Based Guidance System for UAV Navigation and Safe Landing using Natural Landmarks', *Journal of Intelligent and Robotic Systems*, vol. 57, no. 1, p. 233.

Conte, G 2009, *Vision-Based Localization and Guidance for Unmanned Aerial Vehicles*, 0345-7524.

Dotoli, M & Fanti, MP 2004, 'Coloured timed Petri net model for real-time control of automated guided vehicle systems', *International Journal of Production Research*, vol. 42, no. 9, pp. 1787-814.

Hermansson, J, Skoglund, M & Schön, T 2010, *Vision and GPS based autonomous landing of an unmanned aerial vehicle*.

Kim, J, Jung, Y, Lee, D & Shim, DH 2014, 'Outdoor autonomous landing on a moving platform for quadrotors using an omnidirectional camera', *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 1243-52.

Kong, W, Zhou, D, Zhang, D & Zhang, J 2014, 'Vision-based autonomous landing system for unmanned aerial vehicle: A survey', *2014 International Conference on Multisensor Fusion and Information Integration for Intelligent Systems (MFI)*, pp. 1-8.

Lee, D, Ryan, T & Kim, HJ 2012, 'Autonomous landing of a VTOL UAV on a moving platform using image-based visual servoing', *2012 IEEE International Conference on Robotics and Automation*, pp. 971-6.

Pebrianti, D, Kendoul, F, Azrad, S, Wang, W & Nonami, K 2010, 'Autonomous Hovering and Landing of a Quad-rotor Micro Aerial Vehicle by Means of on Ground Stereo Vision System', *Journal of System Design and Dynamics*, vol. 4, no. 2, pp. 269-84.

Pinto, E, Marques, F, Mendonça, R, Lourenço, A, Santana, P & Barata, J 2014, 'An autonomous surface-aerial marsupial robotic team for riverine environmental monitoring: Benefiting from coordinated aerial, underwater, and surface level perception', *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*, pp. 443-50.

Pinto, E, Santana, P, Marques, F, Mendonça, R, Lourenço, A & Barata, J 2014, 'On the Design of a Robotic System Composed of an Unmanned Surface Vehicle and a Piggybacked VTOL', in LM Camarinha-Matos, NS Barrento & R Mendonça (eds), *Technological Innovation for Collective Awareness Systems: 5th IFIP WG 5.5/SOCOLNET Doctoral Conference on Computing, Electrical and Industrial Systems, DoCEIS 2014, Costa de Caparica, Portugal, April 7-9, 2014. Proceedings*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 193-200.

Santos, MCP, Santana, LV, Brandão, AS, Sarcinelli-Filho, M & Carelli, R 2017, 'Indoor low-cost localization system for controlling aerial robots', *Control Engineering Practice*, vol. 61, no. Supplement C, pp. 93-111.

Saripalli, S, Montgomery, JF & Sukhatme, GS 2002, 'Vision-based autonomous landing of an unmanned aerial vehicle', *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 3, pp. 2799-804.

Tang, D, Hu, T, Shen, L, Zhang, D, Kong, W & Low, KH 2016, 'Ground Stereo Vision-Based Navigation for Autonomous Take-off and Landing of UAVs: A Chan-Vese Model Approach', *International Journal of Advanced Robotic Systems*, vol. 13, no. 2, p. 67.

Tomic, T, Schmid, K, Lutz, P, Domel, A, Kassecker, M, Mair, E, Grixa, IL, Ruess, F, Suppa, M & Burschka, D 2012, 'Toward a Fully Autonomous UAV: Research Platform for Indoor and

Outdoor Urban Search and Rescue', *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 46-56.

Wang, W, Song, G, Nonami, K, Hirata, M & Miyazawa, O 2006, 'Autonomous Control for Micro-Flying Robot and Small Wireless Helicopter X.R.B', *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2906-11.

Zhou, M, DiCesare, F & Desrochers, AA 1989, 'A top-down approach to systematic synthesis of Petri net models for manufacturing systems', *Proceedings, 1989 International Conference on Robotics and Automation*, pp. 534-9 vol.1.

Appendices

Appendix A: Waypoint File Generation Testing Results

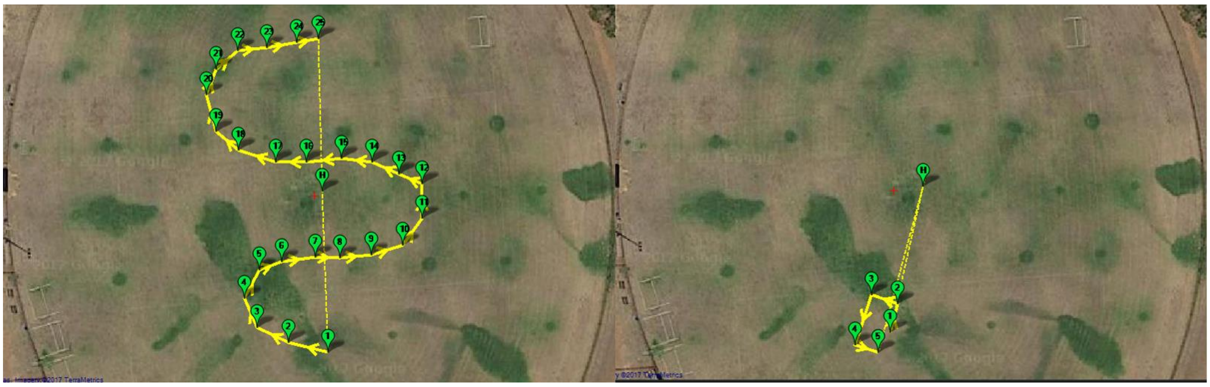


Figure 24 Automatic UAV mission waypoint file generation based upon pose of pseudo-ASV. Left: The path of the ASV. Right: The waypoint file generated at the time the ASV was passing through approximately waypoint 2.

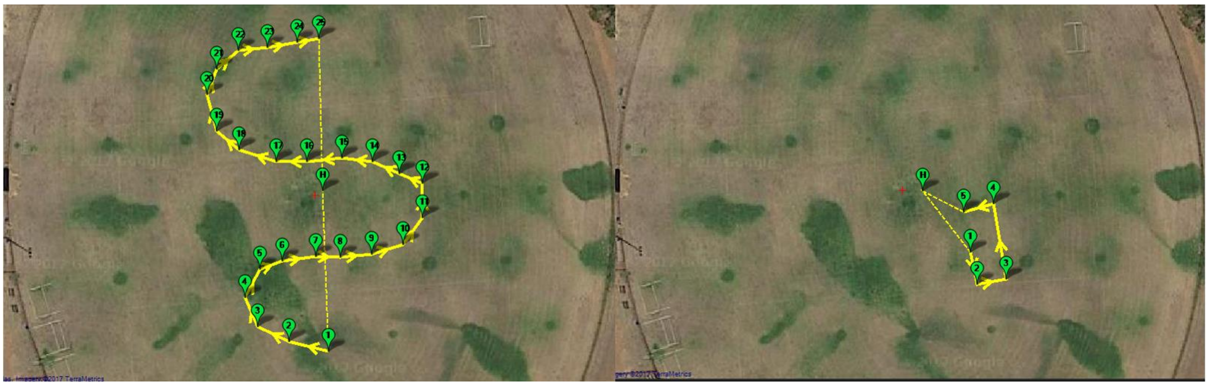


Figure 25 Automatic UAV mission waypoint file generation based upon pose of pseudo-ASV. Left: The path of the ASV. Right: The waypoint file generated at the time the ASV was passing through approximately waypoint 9.

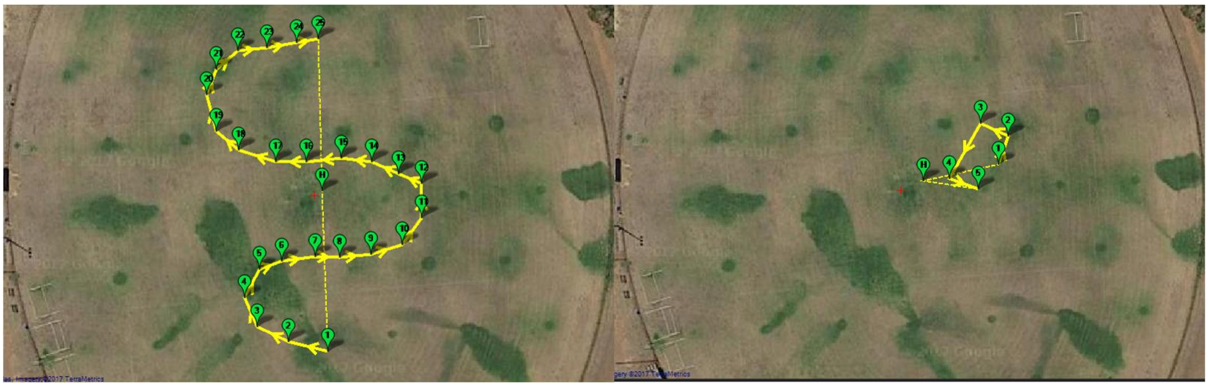


Figure 26 Automatic UAV mission waypoint file generation based upon pose of pseudo-ASV. Left: The path of the ASV. Right: The waypoint file generated at the time the ASV was passing through approximately waypoint 13.

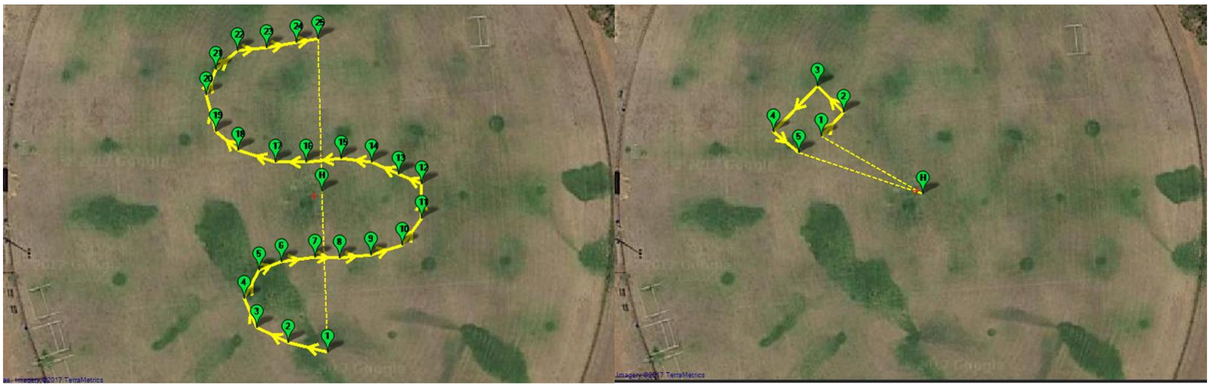


Figure 27 Automatic UAV mission waypoint file generation based upon pose of pseudo-ASV. Left: The path of the ASV. Right: The waypoint file generated at the time the ASV was passing through approximately waypoint 19.

Figure 28 Automatic UAV mission waypoint file generation based upon pose of pseudo-ASV. Left: The path of the ASV. Right: The waypoint file generated at the time the ASV was passing through approximately waypoint 24.